

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

TIAGO LOPES TELECKEN

**Um Estudo sobre Modelos Conceituais  
para Ferramentas de Definição  
de Processos de Workflow**

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

Prof. Dr. José Valdeni de Lima  
Orientador

Porto Alegre, março de 2004

## CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Telecken, Tiago Lopes

Um Estudo sobre Modelos Conceituais para Ferramentas de Definição de Processos de Workflow / Tiago Lopes Telecken. – Porto Alegre: Programa de Pós-Graduação em Computação, 2004.

118 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2004. Orientador: José Valdeni de Lima

1. Workflow. 2 Ferramenta de Definição. 3 Processo. 4 Modelo Conceitual. I Lima, José Valdeni de. II Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof<sup>a</sup>. Wrana Maria Panizzi

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

Pró-Reitora Adjunta de Pós-Graduação: Prof<sup>a</sup>. Jocélia Grazia

Diretor do Instituto de informática: Prof. Philippe Oliver Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

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

## AGRADECIMENTOS

Agradeço primeiramente a minha esposa Juliane, aos meus pais Carlos e Osana e aos meus irmãos Lucas e Vanessa pelo apoio, incentivo e compreensão.

Agradeço ao meu orientador prof. Valdeni pela atenção, confiança, apoio, conselhos e ensinamentos.

Agradeço também a todos que de alguma forma colaboraram com este trabalho seja lendo, revisando, questionando, sugerindo, discutindo, incentivando, analisando, implementando ou configurando. Em especial vai um agradecimento para os professores Carlos Dal'Col Zeve, Nina Edelweiss e José Palazzo de Oliveira e para os colegas Rafael Dutra de Souza, Montgomery Barroso, Manuele Kirsch Pinheiro, Tharso Borges, Cassiano Maciel, Carlos Viegas, Eversson Rizzon, Nícholas Rodriguez Vidal e Igor Steinmacher.

Agradeço ao CNPq pelo apoio financeiro.

E finalmente a Deus pela vida.

Obrigado a todos.

## SUMÁRIO

|  |           |
|--|-----------|
| <b>LISTA DE ABREVIATURAS.....</b>  | <b>8</b>  |
| <b>LISTA DE FIGURAS .....</b>  | <b>9</b>  |
| <b>LISTA DE TABELAS .....</b>  | <b>10</b> |
| <b>RESUMO.....</b>   | <b>11</b> |
| <b>ABSTRACT.....</b>   | <b>12</b> |
| <b>1 INTRODUÇÃO .....</b>  | <b>13</b> |
| <b>2 O CONTEXTO.....</b>   | <b>16</b> |
| 2.1 Sistemas de <i>workflow</i> .....  | 16        |
| 2.2 Tipos de sistemas de <i>workflow</i> .....   | 18        |
| 2.3 As ferramentas de definição .....  | 18        |
| <b>3 TECNOLOGIAS E MÉTODOS UTILIZADOS NA DISSERTAÇÃO. ....</b>                                 | <b>19</b> |
| 3.1 O process model .....  | 19        |
| 3.2 XML.....   | 21        |
| 3.3 Namespace.....   | 22        |
| 3.4 SVG.....   | 22        |
| 3.5 XLink.....   | 23        |
| 3.6 A ferramenta Amaya .....   | 23        |
| <b>4 MODELOS CONCEITUAIS DE FERRAMENTAS DE DEFINIÇÃO.....</b>                                  | <b>25</b> |
| 4.1 Modelo conceitual da aplicação .....   | 25        |
| 4.2 Especificação dos requisitos de ferramentas de definição .....                             | 27        |
| 4.2.1 Conceitos da área de domínio de workflow .....   | 27        |
| 4.2.2 Os usuários.....   | 27        |
| 4.2.3 O contexto.....  | 27        |
| 4.3 Conceitos do domínio de <i>workflow</i> representados em ferramentas de<br>definição ..... | 28        |
| 4.3.1 O modelo de <i>workflow</i> .....  | 28        |
| 4.3.2 Representação do modelo de <i>workflow</i> .....   | 33        |
| 4.3.3 Definição de processo .....  | 35        |
| 4.3.4 Repositório.....   | 35        |
| 4.4 Funcionalidades de ferramentas de definição .....  | 36        |
| 4.4.1 Visões .....   | 36        |
| 4.4.2 Funcionalidades de edição.....   | 36        |

|  |           |
|--|-----------|
| 4.4.3 Controles da definição de processo.....  | 37        |
| 4.4.4 Regras dos Controles.....  | 37        |
| 4.4.5 Reedição e reaproveitamento de <i>workflows</i> .....                            | 39        |
| 4.4.6 Importar/exportar arquivos .....   | 39        |
| 4.4.7 Gerenciamento do repositório.....  | 39        |
| 4.4.8 Modularização.....   | 40        |
| 4.4.9 Verificação de erros .....   | 40        |
| 4.4.10 Simulação .....   | 41        |
| 4.4.11 Recursos para cooperação.....   | 41        |
| 4.4.12 Adaptabilidade .....  | 42        |
| 4.4.13 Interoperabilidade .....  | 43        |
| <b>4.5 Os modelos de interação de ferramentas de definição.....</b>                    | <b>45</b> |
| 4.5.1 Dispositivos de Hardware.....  | 45        |
| 4.5.2 O <i>software</i> da interface.....  | 45        |
| 4.5.3 Estilos de interação .....   | 46        |
| 4.5.4 O estilo de interação menu .....   | 46        |
| 4.5.5 O estilo de interação formulário .....   | 47        |
| 4.5.6 O estilo de interação manipulação direta.....                                    | 47        |
| 4.5.7 Hipertexto .....   | 47        |
| 4.5.8 Os estilos de interação em ferramentas de definição .....                        | 47        |
| 4.5.9 Padrões .....  | 48        |
| <b>4.6 Tabela de descrição de modelos conceituais de ferramentas de definição.....</b> | <b>49</b> |
| <b>5 O AMAYA WORKFLOW.....</b>   | <b>51</b> |
| <b>5.1 Introdução .....</b>  | <b>51</b> |
| <b>5.2 Especificação de requisitos .....</b>   | <b>51</b> |
| 5.2.1 Conceitos da área de domínio de <i>workflow</i> .....                            | 51        |
| 5.2.2 Os usuários.....   | 51        |
| 5.2.3 O contexto.....  | 51        |
| <b>5.3 Conceitos do domínio de <i>workflow</i> representados no AW.....</b>            | <b>52</b> |
| 5.3.1. O modelo de <i>workflow</i> .....   | 52        |
| 5.3.2 A representação do modelo de <i>workflow</i> .....                               | 56        |
| 5.3.3 A definição de processo.....   | 60        |
| <b>5.4 Funcionalidades do AW.....</b>  | <b>60</b> |
| 5.4.1 Visões .....   | 60        |
| 5.4.2 Funcionalidades de edição.....   | 62        |
| 5.4.3 Controles da definição de processo.....  | 62        |
| 5.4.4 Regras dos Controles.....  | 62        |
| 5.4.5 Reedição e reaproveitamento de <i>workflows</i> .....                            | 64        |
| 5.4.6 Exportação .....   | 64        |
| 5.4.7 Importação .....   | 65        |
| 5.4.8 Gerenciamento do repositório.....  | 65        |
| 5.4.9 Modularização.....   | 67        |
| 5.4.10 Verificação de erros .....  | 67        |
| 5.4.11 Adaptabilidade .....  | 68        |
| 5.4.12 Simulação .....   | 69        |
| 5.4.13 Recursos para cooperação.....   | 69        |
| 5.4.14 Interoperabilidade .....  | 70        |
| <b>5.5 Modelo de interação.....</b>  | <b>71</b> |
| 5.5.1 Dispositivos de hardware.....  | 71        |

|  |            |
|--|------------|
| 5.5.2 A interface .....  | 71         |
| 5.5.3 Interações nos Menus .....   | 72         |
| 5.5.4 Interações nas Visões .....  | 73         |
| 5.5.5 Interações da visão gráfica .....  | 73         |
| 5.5.6 Interações na visão da estrutura .....   | 74         |
| 5.5.7 Interações na visão AW .....   | 75         |
| 5.5.8 Interações nas barras de ferramentas.....  | 75         |
| 5.5.9 Inserção de links.....   | 75         |
| 5.5.10 Interações nos formulários do item de menu Attributes.....                      | 75         |
| 5.5.11 Interações nos arquivos de configuração.....                                    | 77         |
| 5.5.12 Padrão de interação .....   | 77         |
| <b>5.6 Tabela de descrição do modelo conceitual AW .....</b>                           | <b>77</b>  |
| <b>5.7 Itens do modelo AW ainda não implementados no protótipo AW.....</b>             | <b>79</b>  |
| <b>6 A REPRESENTAÇÃO AW .....</b>  | <b>81</b>  |
| <b>6.1 Os elementos.....</b>   | <b>81</b>  |
| 6.1.1 Elemento <i>Workflow</i> .....   | 82         |
| 6.1.2 Elementos e atributos gráficos.....  | 82         |
| 6.1.3 Atributos comuns a vários elementos.....   | 83         |
| 6.1.4 Elemento Begin.....  | 84         |
| 6.1.5 Elemento Connector.....  | 89         |
| 6.1.6 Elemento End.....  | 89         |
| 6.1.7 Elemento Task .....  | 89         |
| 6.1.8 Elemento SuperTask .....   | 91         |
| 6.1.9 Elemento MultiTask.....  | 92         |
| 6.1.10 Elemento ForkJoinTotal .....  | 93         |
| 6.1.11 Elemento JoinPartial.....   | 93         |
| 6.1.12 Elemento JoinInteractive .....  | 94         |
| 6.1.13 Elemento ForkConditional.....   | 94         |
| 6.1.14 Elemento EForkConditional .....   | 94         |
| <b>6.2 Resumo da representação AW.....</b>   | <b>94</b>  |
| <b>6.3 Análise da representação .....</b>  | <b>97</b>  |
| 6.3.1 O <i>software</i> Amaya e a tecnologia XML.....                                  | 97         |
| 6.3.2 As representações <i>process model</i> e XPDL.....                               | 97         |
| 6.3.3 A representação SVG .....  | 98         |
| 6.3.4 XLink.....   | 99         |
| 6.3.5 Namespace.....   | 99         |
| 6.3.6 Funcionalidades do AW .....  | 99         |
| <b>7 ANÁLISE E COMPARAÇÃO DOS SISTEMAS EXISTENTES.....</b>                             | <b>100</b> |
| <b>7.1 Análise comparativa de modelos conceituais de ferramentas de definição.....</b> | <b>100</b> |
| <b>7.2 Sistemas existentes .....</b>   | <b>102</b> |
| 7.2.1 Foro model designer.....   | 102        |
| 7.2.2 ILOG <i>workflow</i> modeler.....  | 104        |
| 7.2.3 Ultimus designer .....   | 106        |
| <b>7.3 Análise comparativa do modelo conceitual do Amaya <i>Workflow</i>.....</b>      | <b>109</b> |
| <b>8 CONCLUSÃO .....</b>   | <b>111</b> |
| <b>8.1 Trabalhos futuros.....</b>  | <b>112</b> |
| <b>8.2 Publicações .....</b>   | <b>112</b> |

**REFERÊNCIAS..... 114**

## LISTA DE ABREVIATURAS

|       |  |
|-------|--|
| AW    | Amaya Workflow   |
| DTD   | Document Type Definition                                 |
| EBNF  | Extended Backus-Naur Form                                |
| HTML  | HyperText Markup Language                                |
| HTTP  | Hypertext Transfer Protocol                              |
| OSF   | Open System Foundation                                   |
| RDF   | Resource Description Framework                           |
| SVG   | Scalable Vector Graphics                                 |
| URL   | Uniform Resource Locators                                |
| W3C   | World Wide Web Consortium                                |
| WfMC  | Workflow Management Coalition                            |
| WIDE  | Workflow on Intelligent Distributed database Environment |
| WPDL  | Workflow Process Definition Language                     |
| WWW   | World Wide Web   |
| XLink | XML Linking Language                                     |
| XML   | eXtensible Markup Language                               |
| XPDL  | XML Processing Description Language                      |

## LISTA DE FIGURAS

|   |     |
|---|-----|
| Figura 2.1: Arquitetura da tecnologia de <i>workflow</i> .....  | 16  |
| Figura 2.2: Arquitetura detalhada de um sistema de <i>workflow</i> .....  | 17  |
| Figura 3.1: Diagrama de classes do <i>process model</i> .....   | 19  |
| Figura 3.2: Símbolos da representação gráfica do <i>process model</i> .....   | 20  |
| Figura 3.3: Processo modelado através da representação gráfica do <i>process model</i> .....                                    | 21  |
| Figura 3.4: Exemplo de código SVG.....  | 23  |
| Figura 3.5: Tela do Amaya.....  | 24  |
| Figura 4.1: Relacionamentos entre a especificação dos requisitos, o modelo conceitual e os componentes do <i>software</i> ..... | 26  |
| Figura 5.1: Diagrama de classes do modelo de <i>workflow</i> proposto pela WfMC.....  | 52  |
| Figura 5.2: Diagrama de classes do <i>process model</i> .....   | 53  |
| Figura 5.3: Diagrama de classes do modelo AW .....  | 53  |
| Figura 5.4: Diagrama da entidade pacote dos modelos WfMC e do AW.....   | 54  |
| Figura 5.5: Estrutura de elementos e atributos da representação AW .....  | 57  |
| Figura 5.6: Exemplo de código da representação AW .....   | 58  |
| Figura 5.7: Exemplo de código da representação estruturada.....   | 59  |
| Figura 5.8: Exemplo de código da representação estruturada.....   | 60  |
| Figura 5.9: Visões do AW.....   | 61  |
| Figura 5.10: Estrutura de documentos, processos e pacotes de um <i>workflow</i> .....   | 66  |
| Figura 5.11: <i>Profiles</i> do AW .....  | 69  |
| Figura 5.12: Interface do AW.....   | 72  |
| Figura 7.1: Tela do <i>Foro model designer</i> .....  | 104 |
| Figura 7.2: Tela do <i>ILOG workflow modeler</i> .....  | 106 |
| Figura 7.3: Tela do <i>Ultimus designers</i> .....  | 109 |

## LISTA DE TABELAS

|  |     |
|--|-----|
| Tabela 4.1: Tabela de descrição de modelos conceituais de ferramentas de definição..     | 50  |
| Tabela 5.1: Comparação entre os modelos.....   | 55  |
| Tabela 5.2: Tabela de descrição do modelo conceitual AW .....                            | 79  |
| Tabela 6.1: Atributos do elemento <i>image</i> utilizados nos elementos AW .....         | 83  |
| Tabela 6.2: Elementos e atributos da representação AW.....                               | 95  |
| Tabela 7.1: Tabela de requisitos de modelos conceituais de ferramentas de definição.     | 103 |
| Tabela 7.2: Tabela de descrição do modelo conceitual do <i>Foro model designer</i> ..... | 105 |
| Tabela 7.3: Tabela de descrição do modelo conceitual do <i>ILOG workflow modeler</i> ... | 108 |
| Tabela 7.4: Tabela de descrição do modelo conceitual do <i>Ultimus designers</i> .....   | 101 |

## RESUMO

A crescente necessidade de padronização, eficácia e gerenciamento de processos têm aumentado o interesse das mais diferentes organizações pela tecnologia de *workflow*. Entretanto, a rápida propagação da tecnologia de *workflow*, ocorrida na última década, fez com que muitos sistemas desenvolvidos nesta área fossem disponibilizados e difundidos sem uma padronização clara.

Analisando-se especificamente a fase de construção do *workflow*, nota-se que há muitas divergências. Conceitualmente, ainda não há modelos nem metodologias de modelagem amplamente aceitos e consolidados. Quanto às implementações, ainda não existe uma especificação clara e detalhada das ferramentas de definição. A consequência deste panorama é a baixa interoperabilidade e a baixa padronização dos conceitos, funcionalidades e interações das ferramentas de definição.

Contudo, muitos esforços estão sendo feitos para solucionar o problema da interoperabilidade e padronização. A área de *workflow* como um todo já começa a apontar fortes tendências de padronização dos conceitos, funcionalidades e interações de seus sistemas. Detectar e avaliar tais tendências são os focos de estudos desta dissertação.

Mais precisamente, o objetivo desta dissertação é fornecer informações e métodos que possam ajudar desenvolvedores e usuários de ferramentas de definição a: compreender, descrever e avaliar os conceitos, funcionalidades e interações deste tipo de ferramenta.

Para tanto, é mostrado um método de descrição de modelos conceituais de ferramentas de definição. O referido método é resultado de uma pesquisa sobre o modelo conceitual deste tipo de ferramenta.

Com base nas informações pesquisadas, foi desenvolvido o modelo conceitual de uma ferramenta de definição chamada de Amaya *Workflow* (AW). Para mostrar a viabilidade de implementação do modelo conceitual AW e concretizar em um *software* os estudos desenvolvidos durante esta dissertação, foi implementado o protótipo Amaya *Workflow*.

Por fim, é mostrado um panorama geral das funcionalidades, conceitos e interações das principais ferramentas de definição existentes. Com base neste panorama e nos estudos anteriores, é descrito um método de avaliação de modelos conceituais de ferramentas de definição.

**Palavras-chave:** workflow, ferramenta de definição, processo, modelo conceitual

## A study about conceptual models for process definition tools

### ABSTRACT

The growing necessity for standardization, effectiveness and management of processes has increased the interest of several organizations for the workflow technology. However, the fast propagation of this technology in the last decade has brought the development of many incompatible systems and tools.

Analyzing more specifically the construction phase of a workflow, one notices that the tools developed have many divergences. Conceptually, there are not yet models nor modeling methodologies widely accepted or consolidated. Regarding the implementation, a clear and detailed specification of definition tools still does not exist. The consequence of this panorama is low interoperability and low standardization of the concepts, functionalities and interactions of definition tools.

However, many efforts are being made to solve this problem. Trends for a greater standardization of the concepts, functionalities and interactions of workflow systems as a whole have already started. To detect and to evaluate such trends is the primary focus of this dissertation.

More precisely, the objective of this dissertation is to provide information and discuss methods that can help developers and users of definition tools to understand, describe and evaluate the concepts, functionalities and interactions of this type of tool.

For that, a method for describing conceptual models of definition tools is shown. This method is a result from a research on the conceptual model of this type of tool.

Based on the researches, a conceptual model of a definition tool called Amaya Workflow (AW) was developed. To show its viability and to materialize it, the Amaya Workflow prototype was implemented.

Finally, it is shown a general panorama of the functionalities, concepts and interactions of the main existing definition tools. Based on this panorama and the previous studies, a method for evaluating conceptual models of definition tools is described.

**Keywords:** Workflow, process definition, process, conceptual models

# 1 INTRODUÇÃO

A crescente necessidade de padronização, eficácia e gerenciamento de processos têm aumentado o interesse das mais diferentes organizações pela tecnologia de *workflow* (GEORGAKOPOULOS et al.,1995; SHETH et al., 1996). Todo este interesse fez com que muitos modelos, métodos e sistemas fossem criados em torno do *workflow* (JOOSTEN; BRINKKEMPER, 1996). Entretanto, a rápida propagação da tecnologia de *workflow*, ocorrida na última década, fez com que muitos sistemas desenvolvidos nesta área fossem disponibilizados e difundidos sem uma padronização clara (JOOSTEN, 1994).

Segundo Georgakopoulos et al.(1995), um projeto de *workflow* pode ser dividido em três etapas principais: a construção do *workflow*, a execução do *workflow* e a redefinição de processos.

Na fase de construção do *workflow*, um processo organizacional é modelado em um formato que pode ser manipulado por sistemas de *workflow* e compreendido por seus participantes. Para poder ser modelado, um processo precisa ser compreendido através da análise de documentos, de entrevistas e do uso de metodologias e ferramentas próprias para este fim (ZSCHORNACK, 2003; BARESI et al., 1999). Após ser suficientemente compreendido o processo deve ser modelado em uma **definição de processo** que segue as regras de um **modelo**. Um modelo inclui uma série de conceitos e relacionamentos que abstraem os aspectos mais relevantes do processo que está sendo analisado. A utilização de um *software* chamado de **ferramenta de definição de processos de workflow** ou simplesmente de **ferramenta de definição** é essencial para uma boa eficácia da modelagem do processo. Este tipo de *software* auxilia na visualização, compreensão, modelagem e remodelagem de processos. A ferramenta de definição faz uma interface que tem de um lado os usuários e do outro uma máquina de *workflow*. Os usuários poderão modelar um processo seguindo um modelo ou uma abstração amigável e de alto nível. Já, a máquina de *workflow* irá receber a definição de processo em uma linguagem de baixo nível facilmente manipulável por sistemas computacionais.

Durante a fase de execução do *workflow*, a definição de processo feita anteriormente é interpretada e executada por um sistema. Nesta fase uma ou mais instâncias da definição de processo são executadas e podem interagir com usuários e outros aplicativos.

Já na fase de redefinição de processos, as instâncias de *workflow* são avaliadas e as definições de processos são analisadas e redefinidas caso seja necessário.

Analisando-se especificamente a fase de construção do *workflow*, nota-se que há muitas divergências. Conceitualmente, ainda não há modelos nem metodologias de modelagem amplamente aceitos e consolidados. Quanto às implementações, ainda não existe uma especificação clara e detalhada das ferramentas de definição. A situação mais comum é um fabricante disponibilizar uma solução proprietária e fechada onde a sua ferramenta de definição é compatível somente com máquinas de *workflow* e outras ferramentas do mesmo fabricante. A consequência deste panorama é a baixa interoperabilidade e a baixa padronização dos conceitos, funcionalidades e interações das ferramentas de definição.

Contudo, muitos esforços estão sendo feitos para solucionar o problema da interoperabilidade e padronização. A área de *workflow* como um todo já começa a apontar fortes tendências de padronização dos conceitos, funcionalidades e interações de seus sistemas (WFMC, 1999; SHETH et al., 1996). Detectar e avaliar tais tendências foram os focos de estudos desta dissertação.

Mais precisamente, o objetivo desta dissertação foi fornecer informações e métodos que possam ajudar desenvolvedores e usuários de ferramentas de definição a: compreender, descrever e avaliar os conceitos, funcionalidades e interações deste tipo de ferramenta.

Para tanto, foi feito um levantamento sobre as principais características dos modelos conceituais de ferramentas de definição. O referido levantamento de informações resultou em um método de descrição de modelos conceituais de ferramentas de definição. O **modelo conceitual** de uma aplicação (SOUZA et al., 1999; HEUSER, 1991) é utilizado na engenharia de *software* para definir os **conceitos** que envolvem a aplicação, as **funcionalidades** oferecidas pela aplicação e as possíveis **interações** com esta aplicação. Os usuários utilizam o modelo conceitual para compreender as características do *software* que irão utilizar. Já os desenvolvedores utilizam o modelo conceitual como uma importante etapa da definição e implementação de um *software*.

Com base nas informações pesquisadas sobre os modelos conceituais de ferramentas de definição, foi desenvolvido o modelo conceitual do Amaya *Workflow* (AW). O AW é um modelo conceitual de uma ferramenta de definição que possui os requisitos básicos e comuns aos modelos conceituais das principais ferramentas disponíveis. São requisitos básicos como, por exemplo, conformidade com os padrões da WfMC e disponibilização de uma visão gráfica dos processos de *workflow*. São diferenciais do AW: o uso de padrões abertos e recursos como, por exemplo, edição cooperativa de processos, adaptabilidade da ferramenta, interoperabilidade com padrões XML e interoperabilidade com padrões da Web (o protocolo http, o protocolo WebDAV e WebServices).

Para mostrar a viabilidade de implementação do modelo conceitual AW e concretizar em um *software* os estudos desenvolvidos durante esta dissertação, foi implementado o protótipo Amaya *Workflow* (TELECKEN, 2002; TELECKEN, 2004). O protótipo foi construído como uma extensão do *software* Amaya. O *software* Amaya é um editor e *browser* de documentos XML oficial do *World Wide Web Consortium* (W3C). O protótipo AW não implementa todas as funções definidas em seu modelo conceitual, porém é funcional e está pronto para o uso.

Por fim, é mostrado um panorama geral das funcionalidades, conceitos e interações das principais ferramentas de definição existentes. Com base neste panorama e nos estudos anteriores, é descrito um método de avaliação de modelos conceituais de ferramentas de definição. Tal método é utilizado para avaliar o AW e três modelos conceituais de ferramentas existentes. Foram avaliadas mais detalhadamente as ferramentas: *Foro model designer*, *Ilog workflow modeler* e *Ultimus designer*.

A presente dissertação está organizada da seguinte forma: o capítulo 1 introduz o assunto abordado na dissertação. Os capítulos 2 e 3 mostram uma revisão bibliográfica do contexto e das tecnologias onde estão inseridos os estudos desta dissertação. O capítulo 4 faz um levantamento sobre as principais características dos modelos conceituais de ferramentas de definição. Também é mostrado no capítulo 4 um método de descrição de modelos conceituais de ferramentas de definição. O capítulo 5 descreve o modelo conceitual do AW. O capítulo 6 descreve a estrutura de representação de processos utilizada no AW. O capítulo 7 apresenta um método de avaliação de modelos conceituais e faz uma avaliação do AW e de outros sistemas existentes. No capítulo 8 são mostradas as conclusões.

## 2 O CONTEXTO

Este capítulo descreverá o contexto em que estão inseridos os estudos apresentados nesta dissertação.

### 2.1 Sistemas de *workflow*

*Workflow* pode ser definido como “um conjunto coordenado de atividades (seqüenciais ou paralelas) que são interligadas com o objetivo de alcançar uma meta comum” (CASATI et al., 1996). Já a automação de um processo de *workflow* tem por objetivo criar um sistema que agilize e gerencie um conjunto de atividades manuais e automáticas que visam alcançar uma meta comum.

De uma maneira mais formal a *Workflow Management Coalition* (WfMC, entidade cujo objetivo é definir os padrões da área de *workflow*)(WFMC ,2004) define a automação de um *workflow* como: "a automação de um processo de negócios, por inteiro ou em parte, durante o qual documentos, informações e tarefas são passadas de um participante para outro respeitando-se um conjunto de regras procedurais" (WFMC, 1996).

Segundo a WfMC (WFMC ,2004; WFMC, 1996) a estrutura de um projeto de automação de *workflow* deve seguir a arquitetura mostrada na figura 2.1.

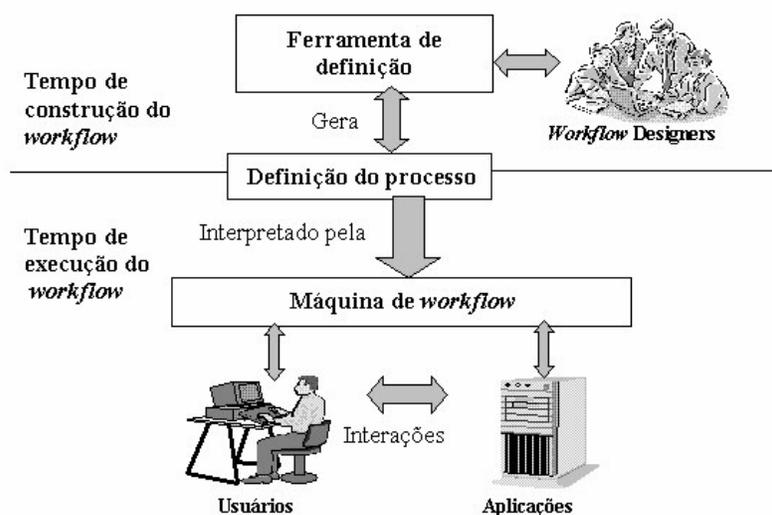


Figura 2.1: Arquitetura da tecnologia de *workflow*

Resumidamente, esta arquitetura determina que existem dois tempos funcionais: o tempo de construção e o tempo de execução. No tempo de construção, o *workflow* é modelado. Já, no tempo de execução, o *workflow* é executado.

Através de uma ferramenta de definição, os *designers* de *workflow* podem gerar uma definição de processo. A definição de processo deve conter todas as informações que uma máquina de *workflow* necessita para gerenciar, controlar e executar um *workflow*.

Após ser gerada, a definição de processos pode ser enviada para uma máquina de *workflow*. Primeiramente, a máquina de *workflow* irá interpretar a definição de processos. Em seguida irá controlar, gerenciar e executar o *workflow* descrito na definição de processos.

A figura 2.2 mostra mais detalhes da arquitetura proposta pela WfMC (WFMC, 1996).

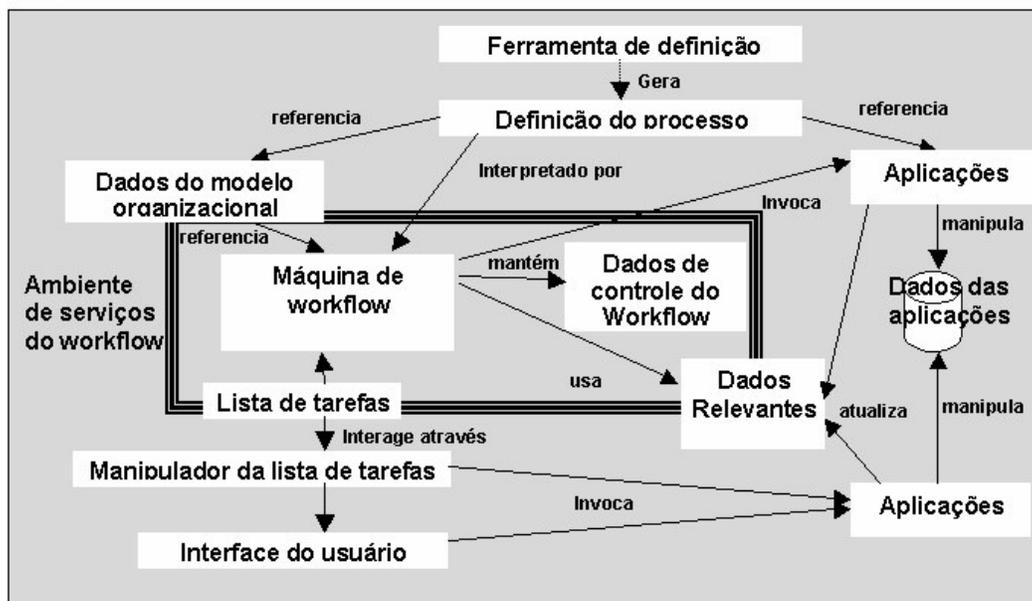


Figura 2.2: Arquitetura detalhada de um sistema de workflow

A arquitetura da figura 2.2 mostra que a definição de processo pode referenciar as aplicações e estruturas da organização. Após ser modelada, a definição de processos pode ser interpretada por uma máquina de *workflow*. Este *software* é responsável pelo controle, gerenciamento e execução do *workflow* modelado. A máquina de *workflow* pode:

- utilizar/referenciar dados da organização;
- invocar aplicações externas;
- manter os dados de controle do *workflow* (são os dados necessários para controlar e manter o funcionamento do *workflow*);
- usar os dados relevantes do *workflow* (são os dados obtidos pela máquina de *workflow* através de recursos externos); e
- manter uma lista de trabalho (uma lista de tarefas que é associada a cada participante do *workflow*).

O usuário interage com o *workflow* através de sua interface. Ele vê as tarefas que deve fazer em sua lista de trabalho e as executa invocando ou utilizando aplicações externas sempre que for necessário. Entre o usuário e a lista de trabalho há um *software* de intermediação chamado de manipulador da lista de trabalho.

As aplicações externas podem ter suas próprias bases de dados e interagem com o *workflow* atualizando os dados relevantes e executando tarefas ao serem invocadas pelos usuários, pela máquina de *workflow* ou pelo manipulador da lista de trabalho. São exemplos de aplicações externas: editores de texto, servidores de e-mail, editores gráficos, etc.

## 2.2 Tipos de sistemas de *workflow*

Segundo Georgakopoulos et al.(1995) e Mccready (1992) os sistemas de *workflow* podem ser classificados nos seguintes tipos:

- ***workflow* Ad Hoc:** são *workflows* pouco estruturados aconselháveis para a modelagem de tarefas não estruturadas. Envolvem coordenação, colaboração e decisão humana durante o seu fluxo. Podem sofrer grandes transformações durante sua execução;
- ***workflow* administrativo:** possuem características semelhantes aos *workflows* de produção, entretanto são menos exigentes e não tem a necessidade de acessos múltiplos a sistemas de informação. São utilizados para processos simples e estruturados como os procedimentos necessários para se aprovar um relatório. A ordenação e coordenação dos processos deste *workflow* podem ser automatizadas;
- ***workflow* de produção:** são *workflows* estruturados, com processos de informações complexos. Existe a necessidade de acesso a múltiplos sistemas de informações. A ordenação e coordenação dos processos deste *workflow* podem ser automatizadas. Este tipo de *workflow* automatiza processos de negócios complexos, similares, por exemplo, a linha de montagem de um carro. Suporta grandes volumes de informações, acompanhamento de tarefas e documentos compartilhados.

## 2.3 As ferramentas de definição

As definições de processos são documentos complexos, que contém muitas informações e que devem seguir uma rígida sintaxe. Por isso, na fase de construção de *workflow* é essencial a utilização de uma ferramenta que auxilie a análise, modelagem e codificação de definições de processos.

Este tipo de ferramenta é denominado pela WfMC (WFMC, 1996) como ferramenta de definição. Tais ferramentas foram o foco de estudos desta dissertação e serão detalhadas a partir do capítulo 4.

### 3 TECNOLOGIAS E MÉTODOS UTILIZADOS NA DISSERTAÇÃO.

Este capítulo descreve as tecnologias e métodos utilizados ou referenciados durante a presente dissertação.

#### 3.1 O process model

Nesta dissertação foram utilizados o modelo e a representação gráfica do *process model*. O *process model* (CASATI et al., 1995) é um modelo de *workflow* desenvolvido no projeto WIDE (CASATI et al., 1996; WIDE, 2002). Através deste modelo é possível definir vários aspectos de um fluxo de trabalho (*workflow*). O diagrama de classes do *process model* é mostrado na figura 3.1.

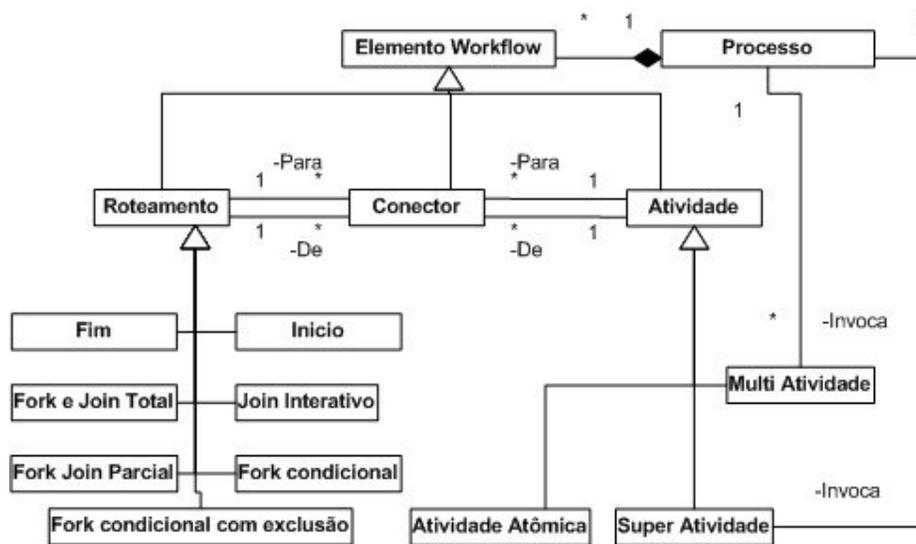


Figura 3.1: Diagrama de classes do process model

No *process model*, um processo é formado por elementos de *workflow*. Existem três tipos de elementos de *workflow*: roteamento, conector e tarefa. As entidades roteamento e tarefa também são formadas por diferentes subtipos, conforme mostra a figura 3.1.

Para o *process model* foi desenvolvida uma representação gráfica. A representação gráfica utiliza símbolos e textos para descrever as tarefas envolvidas em um *workflow* e especificar os mecanismos de disparo e término suas ações.

A representação gráfica do *process model* (CASATI et al., 1995; SIZILIO, 2000) é formada por um conjunto de símbolos que representam: o início do *workflow*, o fim do *workflow*, as tarefas, as conexões entre os elementos, a modularização das tarefas, os diferentes tipos de junções (*joins*) e os diferentes tipos de separações (*forks*). Tais símbolos são mostrados na figura 3.2

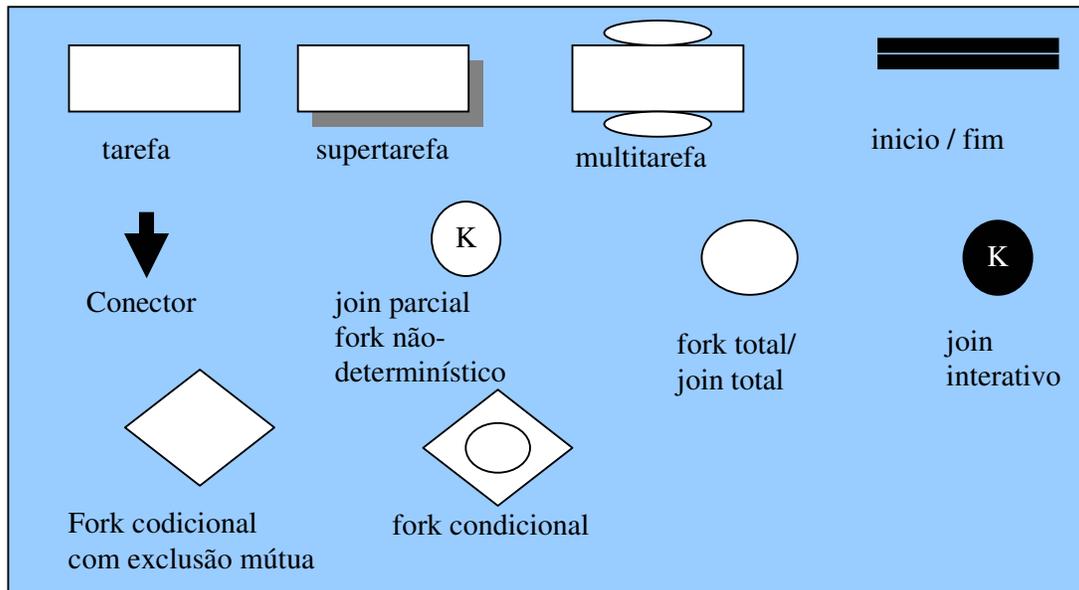


Figura 3.2: Símbolos da representação gráfica do process model

A tarefa é a unidade fundamental de trabalho do *workflow*. É uma atividade, identificada por um nome que pode ser executada por um dos atores definidos.

As conexões entre as tarefas definem os possíveis caminhos a serem tomados a partir das tarefas que foram interligadas. Existem três tipos possíveis de conexões: causalidades, os forks e os joins. Na conexão do tipo causalidade, o término de uma determinada tarefa habilita automaticamente a tarefa posterior.

Nas conexões definidas como *forks*, uma determinada tarefa não é seguida por apenas uma tarefa, mas sim por múltiplas tarefas sucessoras. Assim, existem diferentes tipos de *forks*, são eles: total, não-determinístico, condicional e condicional com exclusão mútua.

No *fork* do tipo total, todas as atividades que seguem a tarefa são habilitadas com o seu término. No *fork* do tipo não-determinístico, um número qualquer de atividades (mas não todas) é habilitado a prosseguir. No *fork* do tipo condicional, após o término da atividade, somente as atividades sucessoras que cumprirem certos requisitos irão prosseguir. Finalmente, no *fork* condicional com exclusão mútua, as atividades sucessoras são avaliadas e somente uma delas é escolhida para ser habilitada.

Nas conexões definidas como *join*, uma determinada tarefa é precedida por múltiplas tarefas diferentes. Existem diferentes tipos de *joins*, são eles: *join* total, *join* parcial e *join* interativo. O *join* total somente habilita a tarefa sucessora se todas tarefas predecessoras forem completadas. O *join* do tipo parcial define que a tarefa sucessora pode ser executada assim que um certo número de tarefas antecessoras for completado (não todas). Finalmente, o *join* do tipo interativo define que a tarefa sucessora será

executada a cada término de um número definido de tarefas. Assim, a tarefa pode ser executada todas as vezes que uma tarefa anterior for completada.

Existem marcas que delimitam o começo e o término de uma instância de *workflow*. Com o símbolo de *início*, habilita-se a atividade sucessora a este, e com a marca de *fim* termina-se a instância, interrompendo qualquer atividade que porventura ainda esteja em execução.

Consegue-se modularizar o *workflow* através do elemento *supertarefa*. O elemento *supertarefa* reúne várias tarefas num só elemento. Seria como uma *função* em uma linguagem de programação estruturada.

Por último, tem-se o elemento multitarefa, que reúne uma série de tarefas que podem ser executadas em paralelo. Deve-se definir quantas instâncias da multitarefa serão geradas. Bem como o número de atividades que devem ser concluídas para que toda a multitarefa esteja concluída.

A figura 3.3 mostra um *workflow* modelado através da representação gráfica do *process model*.

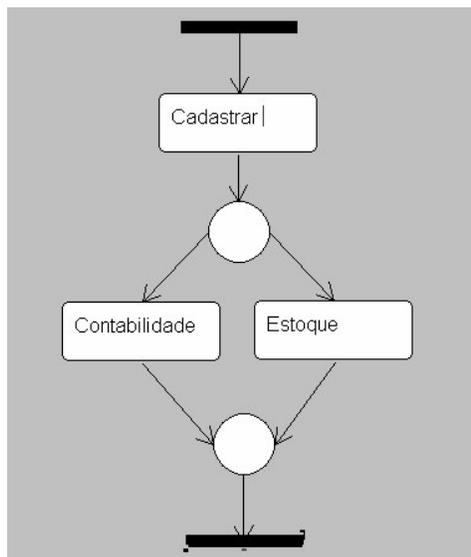


Figura 3.3: Processo modelado através da representação gráfica do *process model*

O *workflow* da figura 3.3 descreve os procedimentos para se registrar uma venda. Inicialmente a venda deve ser cadastrada. Depois, paralelamente é feito o registro da venda no setor da contabilidade e no controle de estoque. Quando os dois registros acabarem termina também o processo.

### 3.2 XML

A tecnologia XML (BRAY et al., 2004) é um conjunto vasto e crescente de módulos que oferecem serviços, ferramentas e padrões úteis para as mais diversas áreas. Em especial o seu uso na estruturação de documentos é uma prática amplamente difundida que oferece a quem o utiliza um crescente número de ferramentas e uma comunidade de profissionais com ampla experiência.

Os benefícios do XML na estruturação de documentos já são amplamente utilizados em definições de processos. A própria WfMC definiu uma linguagem XML (o XPDL) (WfMC, 1999) como sendo o seu modelo de *workflow* padrão. Outros importantes modelos de *workflow* que foram definidos como linguagens XML são: a XLang da Microsoft (THATTE, 2001), a WSFL da IBM (LEYMAN, 2001), a BPML da BPMI (BPMI, 2003), o WSCI da Sun/BEA (W3C, 2002) e a BPEL4WS (LEYMANN; ROLLER, 2002) que é a junção ou evolução da XLang e WSFL.

Muitas ferramentas de definição já utilizam uma linguagem XML como o seu modelo de *workflow*. É o caso do Biztalk Orchestration Designer que utiliza o XLang (THATTE, 2001) e do IBM's MQ Series *Workflow* (IBM, 2001) que utiliza um modelo baseado no WSFL.

A maioria das ferramentas de definição disponíveis exporta ou importa as suas definições de processo para um ou vários formatos XML.

Os principais benefícios de se utilizar, importar ou exportar definições de processos no formato XML são uma grande interoperabilidade com:

- sistemas de *workflow* existentes. Os principais sistemas de *workflow* disponíveis são capazes de importar, exportar ou interpretar documentos escritos em pelo menos uma linguagem XML;
- sistemas que não são específicos da área de *workflow*. Os documentos XML são um formato padrão de troca de dados e tem a sua disposição uma ampla gama de produtos e serviços;

### 3.3 Namespace

A especificação Namespace (BRAY; HOLLANDER; LAYMAN, 1999) é um padrão XML que determina como duas ou mais representações XML podem ser colocadas em um mesmo documento. Esta especificação é uma recomendação oficial do *World Wide Web Consortium* (W3C)(W3C, 2004).

O namespace é utilizado em definições de processos para incluir nestes documentos atributos e elementos de linguagens externas. É o caso da inclusão de elementos RDF em definições de processos da linguagem PSL. Conforme é relatado em (SCHLENOFF, 2000), os elementos RDF em documentos PSL definem recursos que são utilizados em *workflows*.

O namespace também é utilizado para colocar elementos de uma definição de processo dentro de um documento XML de uma linguagem externa. É o caso de elementos XLang colocados em documentos WSDL (*Web Services Description Language*). Conforme está relatado em (THATTE, 2001), o WSDL é um protocolo de descrição de serviços de rede. Dentre outros serviços, o WSDL descreve Web Services (através da XLang).

### 3.4 SVG

A especificação *Scalable Vector Graphics* (SVG) (W3C, 2001) é uma linguagem XML que descreve gráficos 2D. Esta especificação tornou-se em 4 de setembro de 2001 uma recomendação oficial do *World Wide Web Consortium* (W3C).

Diferentemente de formatos como JPEG e PNG que representam um gráfico descrevendo-o *pixel a pixel* o SVG representa um gráfico através de vetores. Vetores representam figuras geométricas através de parâmetros e atributos.

A figura 3.4 mostra um exemplo de código SVG e ao lado o gráfico descrito neste código:

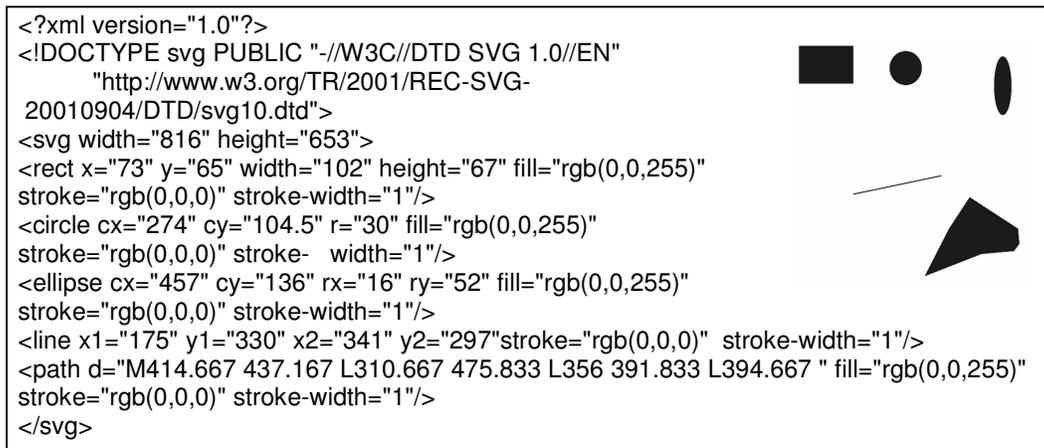


Figura 3.4: Exemplo de código SVG

Junto com suas figuras geométricas, a representação vetorial do SVG permite que figuras (JPEG, por exemplo) e textos sejam adicionados a um gráfico. Os textos adicionados podem ser copiados e manipulados como figuras ou editados como textos comuns. Os textos, portanto, podem ser manipulados por scripts.

Devido ao seu modo de representação, um gráfico SVG é escalonável. Ele permite que uma área em particular de um gráfico seja aproximada sem que haja uma degradação da imagem. Tal característica também permite que um gráfico seja reproduzido em diferentes resoluções sem que haja uma perda de qualidade da imagem.

Ferramentas de definição como, por exemplo, o ILOG *workflow modeler* (ILOG, 2001) utilizam o SVG como um formato de exportação. Estas ferramentas geram documentos SVG que descrevem gráficos que representam um *workflow*.

### 3.5 XLink

A especificação XLink (XML *Linking Language*) (DEROSE; MALER; ORCHARD, 2001) é uma linguagem XML que descreve ligações (*links*) entre recursos. Esta especificação é uma recomendação oficial do World Wide Web Consortium (W3C).

O XLink é utilizado por algumas ferramentas de definição e proposto por vários autores (DODDS et al., 2001) como um bom padrão para ligar elementos de definições de processos localizados em diferentes documentos.

### 3.6 A ferramenta Amaya

O Amaya (W3Cb, 2004) é um *browser* e editor de documentos, desenvolvido pelo W3C que permite a publicação de documentos na Web. Os documentos editados pelo Amaya podem ser do tipo texto, HTML, XHTML, folhas do estilo do CSS, expressões de MathML, e desenhos de SVG. Em especial destaca-se o editor SVG. Através de uma paleta de elementos gráficos, este editor permite que um usuário edite figuras geométricas que seguem o padrão SVG. Sua interface é do tipo "WYSIWYG" (a aparência do documento durante a edição é semelhante à aparência do documento final). A figura 3.5 mostra uma tela do Amaya.

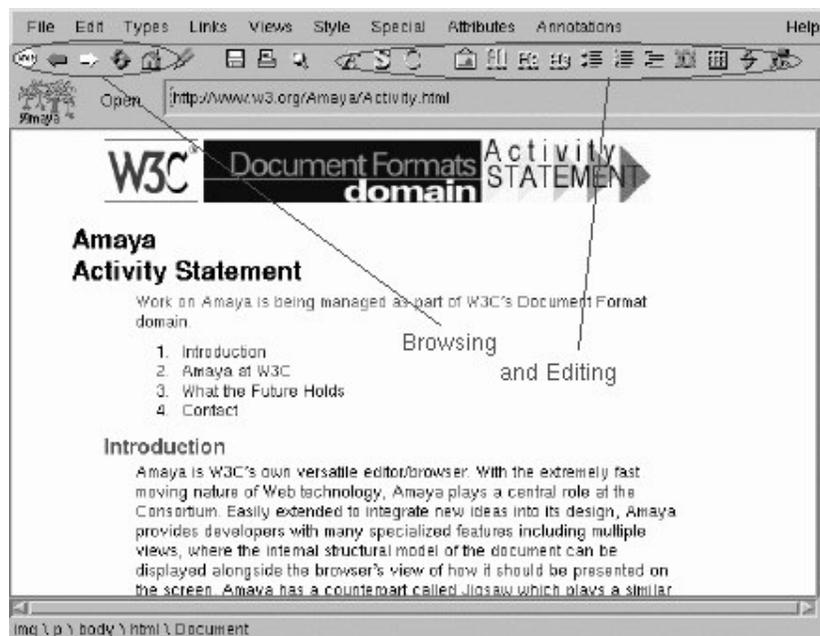


Figura 3.5: Tela do Amaya

Também se destacam no Amaya o suporte as tecnologias: RDF(W3C, 1999), Xpointer, anotação colaborativa (projeto Annotea) (KOIVUNEN, 2001), XLink, protocolo http (que permite a edição de documentos via Internet), protocolo WebDAV (que permite a edição colaborativa de documentos via internet) e namespace.

Um documento no Amaya é representado por sua estrutura lógica, essencialmente hierárquica, que se chama “árvore abstrata”. Essa estrutura segue algumas regras, uma espécie de formato de documento, que ajuda (ou obriga) seus usuários a produzirem documentos daquele tipo. Para documentos HTML o Amaya segue as regras da DTD do HTML, para documentos SVG o Amaya segue a DTD do SVG.

O protótipo Amaya *Workflow* descrito nesta dissertação foi desenvolvido como uma extensão do Amaya.

## 4 MODELOS CONCEITUAIS DE FERRAMENTAS DE DEFINIÇÃO

### 4.1 Modelo conceitual da aplicação

O **Modelo Conceitual da Aplicação** é a visão ou percepção que o usuário tem do sistema. A idéia de modelo conceitual da aplicação também é chamada na engenharia de *software* de *modelo conceitual do usuário*, *virtualidade*, *modelo de usabilidade*, *metáfora de interface* entre outras denominações (LEITE; SOUZA, 1999; SOUZA et al., 1999).

O Modelo Conceitual da Aplicação define (HEUSER, 1991; LEITE; SOUZA, 1999; SOUZA et al., 1999):

- quais são os **conceitos** ou entidades do domínio estão representados na aplicação;
- o que o usuário pode fazer com a aplicação. São as **funcionalidades** da aplicação; e
- como ele pode interagir com a aplicação (interação usuário-sistema). É o **modelo de interação**.

Para melhor definir ou interpretar um modelo conceitual é fundamental um conhecimento prévio da **especificação dos requisitos** do *software*. Os seguintes itens fazem parte da especificação de requisitos de uma ferramenta de definição:

- os conceitos envolvidos no domínio de *workflow*;
- os usuários da aplicação (suas necessidades e preferências); e
- o contexto onde a aplicação será inserida.

As informações sobre a arquitetura dos componentes do sistema, a estrutura de dados e os algoritmos utilizados são omitidos do modelo conceitual. O foco do modelo conceitual está no usuário. O seu objetivo é definir para o usuário as características do sistema que ele irá utilizar.

O modelo conceitual também é utilizado por desenvolvedores como uma importante etapa da definição e desenvolvimento de um *software* (SOUZA et al., 1999). Com base em uma análise de requisitos, o modelo conceitual é utilizado para especificar as características de um *software*. Numa fase seguinte, os conceitos e funcionalidades do modelo conceitual são utilizados na definição e/ou implementação da arquitetura e componentes do *software*. Já o modelo de interação do modelo conceitual é utilizado na definição e/ou implementação da interface do usuário. Este ciclo pode ser feito várias vezes. Nestes casos, o resultado do processo de desenvolvimento de *software* é um protótipo que a cada ciclo é melhorado. A figura 4.1 mostra os relacionamentos entre a

especificação de requisitos, o modelo conceitual e os componentes do *software* ou protótipo (SOUZA et al., 1999).

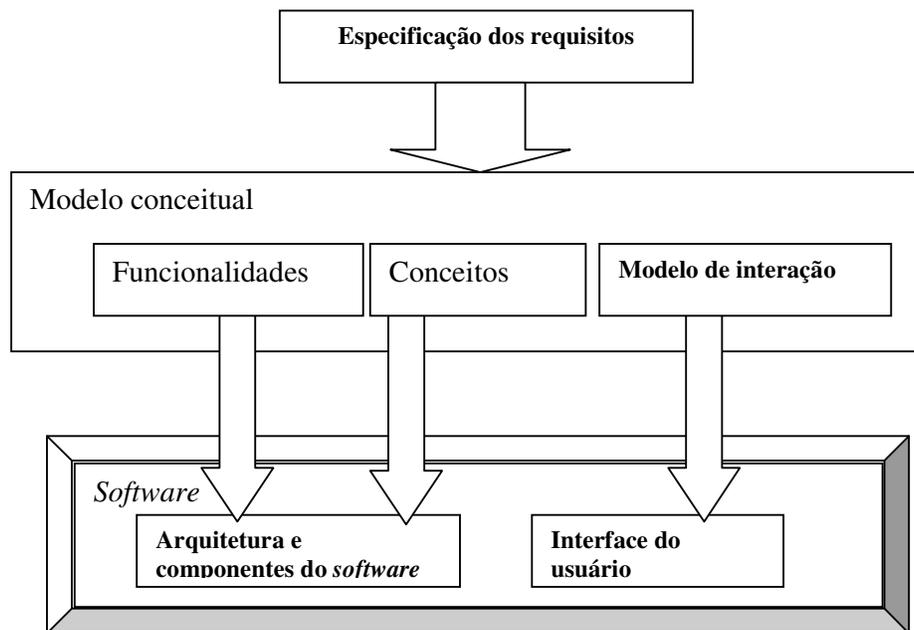


Figura 4.1: Relacionamentos entre a especificação dos requisitos, o modelo conceitual e os componentes do *software*

O protótipo AW foi desenvolvido segundo esta metodologia. A especificação de requisitos do AW, a descrição do modelo conceitual do AW e a descrição do protótipo AW são apresentadas nos demais capítulos desta dissertação.

O objetivo deste capítulo é fornecer informações que possam ajudar desenvolvedores e usuários de ferramentas de definição a:

- compreender os conceitos e o funcionamento deste tipo de ferramenta;
- descrever o modelo conceitual de uma ferramenta já existente; e
- descrever o modelo conceitual de uma nova ferramenta.

Para tanto serão expostos e analisados: os principais itens da **especificação dos requisitos** destas ferramentas; os principais **conceitos** do domínio de *workflow* representados em ferramentas de definição; as suas principais **funcionalidades** e as diferentes opções de **modelos de interação** deste tipo de ferramenta. Por fim será mostrado um método e uma tabela que resume os itens que devem ser descritos em modelos conceituais de ferramentas de definição.

## 4.2 Especificação dos requisitos de ferramentas de definição

### 4.2.1 Conceitos da área de domínio de workflow

São os conceitos mostrados no capítulo 2 desta dissertação. As seguintes bibliografias também fornecem informações sobre a área de domínio de *workflow*: (GEORGAKOPOULOS et al.,1995; WFMC, 1996).

### 4.2.2 Os usuários

O foco do modelo conceitual é os usuários. Portanto, torna-se fundamental conhecer o perfil, as necessidades e as preferências destes usuários.

Em ferramentas de definição de processos de *workflow*, os principais usuários são os **analistas de *workflow***. Estes profissionais utilizam a ferramenta para modelar um processo, analisar um processo e gerar uma definição de processo que será enviada para uma máquina de *workflow* ou outro *software*. Geralmente, um analista de *workflow* experiente possui bons conhecimentos sobre o domínio de *workflow*, sobre análise de processos, sobre a modelagem de *workflow*, sobre as tecnologias de *workflow* e está familiarizado com recursos de informática. Porém, muitas vezes o analista de *workflow* pode não ter experiência ou conhecimento em uma das áreas citadas. Nestes casos a ferramenta deve ser adaptada a estes diferentes perfis.

Outro eventual usuário é o **analista do domínio**. O analista do domínio é o usuário que tem conhecimento sobre a área de domínio onde a tecnologia de *workflow* será utilizada (ensino, automação de processos de escritório, comércio eletrônico, etc). Normalmente este usuário desconhece a tecnologia de *workflow*. Porém ele participa do processo de modelagem de *workflow* auxiliando o analista de *workflow*. O analista de *workflow* muitas vezes desconhece a área de domínio dos processos que deverão ser modelados. Por isso, é importante o trabalho em conjunto dos analistas de *workflow* e dos analistas do domínio.

O analista de *workflow* precisa ter acesso e compreender bem todas as funcionalidades e potencialidades do sistema. Ele é o usuário que irá manusear diretamente a ferramenta. Já o analista do domínio precisa apenas compreender as representações (esquemas, desenhos, textos) dos processos e os conceitos envolvidos nestas representações. Compreendendo bem as representações dos processos, o analista do domínio poderá analisar o processo e avaliar se o mesmo atende as necessidades do domínio.

### 4.2.3 O contexto

O contexto de um aplicativo é formado por todos os fatores externos ao aplicativo que de alguma forma influenciam o comportamento do *software* e a percepção que o usuário tem do aplicativo (CASATI et al., 1996; GEORGAKOPOULOS et al.,1995).

O contexto onde é aplicada uma ferramenta de definição é dado principalmente:

- pelas características dos tipos de processos que devem ser modelados;
- pela tecnologia de hardware e *software* que irá automatizar os processos já modelados;

- pela estrutura organizacional da instituição/empresa onde será aplicada a tecnologia de *workflow*;
- pelas aplicações externas que podem interagir com a ferramenta; e.
- pelas especificidades da área de domínio onde será aplicada a tecnologia de *workflow*.

### **4.3 Conceitos do domínio de *workflow* representados em ferramentas de definição**

#### **4.3.1 O modelo de *workflow***

Um modelo de *workflow* tem a função de abstrair um determinado conjunto de entidades relevantes à realidade de um fluxo de trabalho.

O modelo de *workflow* adotado por uma ferramenta de definição determina quais são os conceitos e recursos de *workflow* que poderão ser usados e modelados na ferramenta.

Ao definir o modelo de *workflow*, as entidades definidas pela WfMC devem ser levadas em consideração. A seguir, serão citadas algumas entidades de *workflow* que fazem parte de vários modelos de *workflow*. Em especial fazem parte do meta-modelo definido pela WfMC (WfMC, 1999).

##### **4.3.1.1 Atividade**

É a entidade que representa as unidades de trabalho ou as ações de um processo.

As atividades podem representar ações que interagem com aplicativos externos, com a própria máquina de *workflow*, com máquinas de *workflows* externas e com pessoas. Além da sua ação, a atividade pode fornecer as seguintes propriedades: dados de entrada, dados de saída, participantes, recursos, momentos em que as ações são disparadas, eventos que disparam as ações, eventos que são disparados pelas ações, tratamentos em caso de erros ocorridos em tempo de execução, versão, prioridade, etc. Alguns modelos definem diferentes tipos de atividades sendo que para cada tipo são associadas diferentes informações.

Existem ainda atividades que representam um conjunto de atividades e invocam processos ou grupos de atividades.

##### **4.3.1.2 Roteamento**

É a entidade que representa os roteamentos ou fluxos das atividades.

Os roteamentos podem representar uma seqüência simples de atividades, bifurcações de diferentes tipos, sincronização de diferentes tipos, *loops*, diferentes tipos de início (roteamento que determina o início ou uma entrada para o *workflow*) e diferentes tipos de fim (roteamento que determina o fim ou uma saída do *workflow*). Através de uma única entidade, alguns modelos representam um fluxo complexo que seria formado por um conjunto de outras entidades de roteamento. Alguns tipos de roteamento permitem que o usuário ou a máquina de *workflow* defina em tempo de execução do *workflow* algumas de suas propriedades.

Três tipos importantes de propriedades das entidades de roteamento são:

- as propriedades que definem as condições que um fluxo do *workflow* precisa ter para poder passar por um determinado roteamento (estas propriedades também são chamadas de condições do elemento de transição ou regras para a transição);
- as propriedades que definem sob que condições uma atividade pode ser inicializada (estas propriedades também são chamadas de pré-condições da atividade);
- as propriedades de roteamento que definem sob que condições uma atividade pode ser considerada terminada (estas propriedades também são chamadas de pós-condições da atividade).

Um estudo mais detalhado sobre tipos de roteamentos utilizados em sistemas de *workflow* é mostrado no artigo de Aalst et al.(2000).

#### 4.3.1.3 Participante

É a entidade que representa as informações sobre o ator ou método de escolha do ator que deverá executar ou ser responsável por uma determinada atividade. A máquina de *workflow* utiliza estas informações para definir em tempo de execução quem será o ator que irá executar ou se responsabilizar por uma atividade.

Pode-se informar o ator diretamente no modelo, pode-se definir papéis que em tempo de execução serão assumidos pelos atores apropriados ou pode-se informar as condições e características que o ator deve ter para poder executar uma tarefa. Os atores podem ser pessoas, sistemas automatizados, grupo de pessoas, departamento ou organização.

As características dos atores podem conter informações sobre suas aptidões, função na organização, permissões de acesso, disponibilidade, entre outras. Além de informações sobre as características dos atores o modelo também pode representar informações sobre o método de escolha do ator.

#### 4.3.1.4 Aplicativos

É a entidade que representa as informações sobre um aplicativo que poderá executar uma atividade automática.

O modelo poderá representar as seguintes informações sobre os aplicativos: os parâmetros de entrada que devem ser enviados ao aplicativo quando este for chamado, os parâmetros de saída que o aplicativo irá devolver a máquina de *workflow* após executar suas ações, o nome, uma identificação única e uma descrição.

#### 4.3.1.5 Dados relevantes ao processo de workflow

É a entidade que representa as variáveis ou dados criados e utilizados na definição do processo de *workflow*. Estes dados são persistentes e seu escopo é definido no modelo (Geralmente o escopo é o processo ou o pacote onde é definido).

Os dados relevantes ao processo de *workflow* são criados e utilizados na própria definição do processo, portanto um modelo deve ter meios de representar:

- os tipos de dados que poderão ser criados: É a representação da definição dos tipos de dados das variáveis do modelo. Alguns modelos já definem tipos fixos pré-definidos e todos dados devem ser destes tipos pré-definidos, outros permitem que os tipos de dados sejam definidos na própria definição do processo. Os tipos possíveis vão desde os básicos como *strings* e números inteiros, até tipos complexos formados por um conjunto de tipos básicos ou tipos de dados especificados em documentos externos. Sempre que necessário também são informados os possíveis valores que podem ser assumidos por variáveis de um determinado tipo;
- a criação dos dados: É a representação da criação de uma variável. Na sua criação a variável recebe um nome e a especificação do seu tipo de dado;
- a atribuição de valores aos dados: é a representação da atribuição e alteração dos valores de uma variável já criada. Pode-se atribuir ou alterar o valor de uma variável através de interações com os usuários, com aplicativos e com a máquina de *workflow*.

#### 4.3.1.6 Aspectos Temporais

São as entidades ou propriedades que representam os aspectos temporais do processo.

Podem ser representados no modelo referências :

- aos instantes de tempo: é um instante no tempo que pode ser expresso em diferentes unidades (ano, mês, dia, hora, minuto, etc);
- ao intervalo de tempo: é um intervalo entre dois instantes de tempo pré-definidos;
- à duração: é um intervalo de tempo cujos instantes inicial e final não são pré-definidos.

#### 4.3.1.7 Prioridade

São as entidades ou propriedades que representam os diferentes níveis de prioridade de execução de uma entidade (geralmente processo ou atividade).

Há sistemas que permitem que seja definido se a execução de uma entidade é prioritária ou não. Outros permitem a definição de diferentes níveis de prioridade. A prioridade pode ser definida através de uma expressão de prioridade que é testada e determinada em tempo de execução e ainda pode contar com a intervenção do usuário.

#### 4.3.1.8 Versão

São as entidades ou propriedades que registram a versão de uma entidade.

São informações do modelo que registram dados sobre a versão de uma entidade. Tipos de informações sobre versões são: o nome da versão, descrição da versão e a data em que a versão foi criada.

#### 4.3.1.9 Extensão

É a entidade do modelo que representa informações adicionais não previstas no modelo.

Os modelos podem determinar a existência de entidades "adaptáveis". Inicialmente estas entidades não têm uma função específica. Porém, o modelador do processo poderá usar e valorar a entidade extensão conforme a sua necessidade caso queira colocar alguma informação na definição do processo que não foi prevista no modelo.

#### 4.3.1.10 Processo

É a entidade que representa um conjunto de atividades que visam alcançar uma meta. Em uma entidade processo estão contidas as informações necessárias para a automação de um processo de *workflow*.

A entidade processo pode conter as seguintes entidades: recursos, participantes, dados relevantes ao processo de *workflow*, atividades, roteamentos, transações, exceções, versão, prioridade, outros processos, entre outros.

Além destas entidades um processo pode conter informações gerais sobre o processo como, por exemplo: identificador único, nome, responsável, criador, descrição do processo, versão, parâmetros de entrada e parâmetros de saída para o caso de processos parametrizáveis.

#### 4.3.1.11 Pacote

É a entidade que representa um pacote que pode conter vários processos e informações compartilhadas por todos processos do pacote.

O pacote pode conter:

- vários processos;
- informações sobre o pacote tais como: identificador único, nome, responsável, criador e descrição do pacote;
- entidades compartilhadas e utilizadas por todos processos do pacote tais como: recursos, participantes e dados relevantes ao processo de *workflow*.

#### 4.3.1.12 Repositório

É a entidade que representa um grande container que conterá vários pacotes e processos de *workflow*.

O repositório pode conter:

- vários pacotes;
- informações sobre o repositório tais como: identificador único, nome, responsável, descrição do repositório;
- conceitos compartilhados e utilizados por todos pacotes do repositório tais como recursos, participantes e dados relevantes ao processo de *workflow*.

**OBS:** As próximas entidades não estão diretamente explicitadas no modelo da WfMC. Mas podem ser deduzidas implicitamente deste modelo ou fazem parte de outros projetos importantes da área (CASATI et al., 1996; EDER; LIEBHART, 1995; ILOG, 2001).

#### 4.3.1.13 Eventos

É a entidade que representa algum acontecimento observável. Difere-se de atividades pelo fato das atividades estarem associadas a um intervalo de tempo enquanto o evento ocorre em um determinado instante de tempo. O momento em que um evento é observável chama-se de ocorrência do evento.

Pode-se modelar os seguintes tipos de eventos:

- internos ocorridos no escopo das atividades e demais elementos do *processo*;
- externos que de alguma forma influenciam o *processo*.

Cada tipo de evento citado pode conter vários sub-tipos.

#### 4.3.1.14 Recursos

É a entidade que representa os recursos que devem ser oferecidos ou informados aos atores humanos que executarão uma atividade. Um recurso é um aplicativo, documento ou informação que pode ajudar o ator a executar sua atividade.

Os recursos podem ser agrupados em containeres que podem ser referenciados ou utilizados em várias atividades.

#### 4.3.1.15 Transação

É a entidade que representa os recursos que o modelo possui para a determinação das transações de um processo.

Há modelos que permitem a representação de somente um nível de transação. Outros permitem a existência de n níveis de transações. Formando assim uma hierarquia de transações.

Há modelos que fornecem detalhes sobre cada transação. Estes detalhes fornecem informações como: a importância da transação para o processo, características da transação, os procedimentos que devem ser feitos em caso de compensação da transação, os procedimentos que devem ser feitos em caso de aborto da transação, as condições que determinam o aborto ou compensação da transação, entre outras.

#### 4.3.1.16 Exceções

É a entidade que representa os recursos que o modelo possui para a determinação das exceções e situações incomuns que podem acontecer em um processo. Tais entidades muitas vezes também informam como tratar estas exceções.

As principais exceções representadas em modelos de *workflow* estão relacionadas a a seguir :

- temporais;
- internas relacionadas com o funcionamento da máquina de *workflow* (por exemplo, os estados das instanciações do processo e suas atividades, valores incorretos atribuídos a variáveis do *workflow*, etc);
- externas à máquina de *workflow* como, por exemplo, os eventos relacionados com os participantes e os aplicativos externos.

Os tipos de tratamentos representados podem ser instruções que indicam ações que vão desde simples mensagens enviadas aos atores até a execução de definições de processos alternativos, a re-execução de parte do processo, a mudança dos estados das atividades ou a mudança dos estados do *workflow*.

#### 4.3.1.17 Informações gráficas

São as entidades ou propriedades gráficas do modelo de *workflow*.

Quando um modelo será representado graficamente, as propriedades gráficas do modelo precisam estar descritas na definição do processo. Portanto, precisam fazer parte do modelo. As principais propriedades gráficas encontradas em modelos de *workflow* foram:

- **figura:** é uma referência a uma figura ou ícone que irá representar um conceito de *workflow* em uma visão do processo. Por exemplo, uma referência para uma figura no formato *gif* ou *jpg* ou ainda um ícone no formato *xpm*;
- **forma:** é uma forma geométrica que irá representar um conceito de *workflow* em uma visão do processo. Por exemplo, círculo, linha, triângulo, elipse, etc;
- **tamanho:** são as dimensões que a forma ou figura de um conceito de *workflow* vão ter em uma visão do processo. Por exemplo, altura, largura ou diâmetro ocupado pela forma ou figura na visão do processo;
- **cores:** são as cores que a forma ou figura terá em uma visão do processo. Por exemplo, as cores das bordas de uma figura ou ainda as cores internas de uma forma;
- **localização:** é a localização da forma ou figura de um conceito dentro em uma visão do processo. Por exemplo, as coordenadas cartesianas x e y do canto superior esquerdo de uma forma ou figura.

**OBS:** Além destas entidades, que são entidades mais gerais, existem algumas entidades específicas para determinadas aplicações. No caso de uma aplicação necessitar, estas entidades específicas também devem ser analisadas e inseridas no modelo de *workflow*.

#### 4.3.2 Representação do modelo de *workflow*

A representação do modelo de *workflow* é a estrutura de dados e/ou símbolos que representará as entidades e propriedades definidas no modelo de *workflow*.

As representações encontradas em ferramentas de definição de processo podem ser divididas em: representações gráficas, representações textuais e representações de baixo nível (CASATI et al., 1995).

Geralmente, representações gráficas descrevem um *workflow* através de elementos gráficos em um grafo. Normalmente, os nodos representam elementos do modelo de *workflow* como, por exemplo, as atividades. Os nodos do grafo são conectados por um arco orientado que indica desta maneira a seqüência, paralelismo ou sincronia das atividades ou demais elementos do *workflow*. Informações complementares podem ser colocadas no modelo através de textos (CASTROIANNI, 1995).

As representações textuais descrevem um *workflow* através de textos que seguem a sintaxe de uma determinada gramática. Algumas representações textuais podem ser

interpretadas por sistemas automatizados como máquinas de *workflow* que em muitos casos compilam a representação textual em uma representação de baixo nível.

Portanto pode-se dizer que representações textuais oferecem especificações de um modelo de *workflow* em um nível intermediário enquanto as representações gráficas apresentam especificações em um alto nível (NICOLAO, 1998).

Já as representações de baixo nível são interpretadas apenas por sistemas computacionais. Estas representações descrevem um *workflow* através de linguagens de baixo nível.

#### 4.3.2.1 Representações estruturadas.

As representações de modelos de *workflow* também podem ser divididas entre estruturadas e não estruturadas. As representações não estruturadas definem estruturas livres. Já as representações estruturadas definem estruturas de dados e símbolos que devem seguir uma determinada sintaxe. Por ser o tipo de representação mais importante e utilizado em ferramentas de *workflow* a presente dissertação focalizará as representações estruturadas.

Em representações estruturadas um *workflow* é descrito através de:

- **elementos:** são as unidades de informações destas representações. Um elemento pode conter outros elementos formando assim uma hierarquia de elementos. Os elementos também podem conter atributos e elementos básicos como textos (caracteres) ou imagens;
- **elementos básicos:** um elemento básico é formado por um texto ou símbolo. A sintaxe da seqüência de elementos básicos dentro de um elemento pode ser livre ou seguir um vocabulário/linguagem específico e pré-determinado;
- **atributos:** são usados para fornecer informações adicionais sobre os elementos. O seu conteúdo pode ser livre ou um vocabulário/linguagem específico e pré-determinado;
- **vocabulários/linguagens:** são os vocabulários ou linguagens utilizados nos conteúdos dos elementos e seus atributos. Podem ser desde tipos simples como imagem, texto livre, booleanos, números e datas, até linguagens mais complexas como uma EBNF (*Extended Backus Naur Form*), um vocabulário XML ou uma linguagem de programação.

Numa representação estruturada as informações sobre uma entidade do modelo de *workflow* podem estar explicitamente representadas em:

- um atributo;
- um único elemento;
- um elemento raiz juntamente com seus atributos e elementos filhos;
- vários elementos que são interligados por referências mutuas;

Algumas informações sobre uma entidade do modelo de *workflow* podem não estar explicitamente colocadas na representação do modelo. Porém, podem ser implicitamente deduzidas através de convenções do contexto da representação do modelo. A idéia de se deduzir informações vem do próprio comportamento humano. Segundo (KENT, 1978) "a ambigüidade das palavras é resolvida pelos humanos pelo

entendimento do contexto". Autores como (BALZER; GOLDMAN; WILE, 1978) (LOH, 1991) dizem que o uso do contexto simplifica o processo de representação, as representações tornam-se mais concisas, pois apenas parte da representação fica explícita. As demais informações ficam implícitas e devem ser extraídas do contexto. Isto permite direcionar esforços apenas para as informações explicitadas, o que torna a definição de processo mais legível e compreensível.

Os fatores do contexto da ferramenta que mais influenciam a representação de um modelo de *workflow* são as características e convenções da máquina de *workflow*, da organização que utiliza a máquina de *workflow* e dos processos que serão modelados. Com um bom conhecimento do contexto, as representações do modelo podem ser otimizadas.

Se por exemplo, em um determinado contexto, as variáveis relevantes de *workflow* só puderem ser do tipo *string*. Não é necessário que o usuário defina quais são os tipos de variáveis possíveis de serem utilizadas (pois só é possível o tipo *string*) e ao valorar uma variável o usuário não precisa explicitar o seu tipo (pois são todas do tipo *string*). Neste caso uma boa representação do modelo seria uma representação em que não fosse necessário informar os possíveis tipos de dados, (por convenção só existe o tipo de dado *string*) nem os tipos de dados de cada variável (por convenção são todas do tipo *string*).

Outro exemplo de convenção é a posição dos elementos dentro de uma definição do processo. Em algumas representações, a disposição dos elementos passa implicitamente várias informações.

### 4.3.3 Definição de processo

A definição de processo é o documento que armazena a descrição de um *workflow*. A estrutura de dados e símbolos da definição de processo é definida em uma representação do modelo de *workflow* (WFMC, 1996).

Uma definição de processo pode estar num formato que pode ser **interpretado** pela ferramenta de definição. Neste caso, a ferramenta de definição consegue interpretar e gerar visões da definição de processo.

Além da definição de processo, que é interpretada pelo sistema, existem as definições de processo que são **exportadas** ou **importadas** pelo sistema (a exportação e importação serão detalhadas adiante).

### 4.3.4 Repositório

Um **repositório** é um container de várias definições de processos (WFMC, 1999).

Um repositório pode ser **temporário**. Neste caso, fazem parte do repositório as definições de processo abertas durante uma seção. Quando a ferramenta ou seção é fechada as informações sobre o repositório são perdidas.

Outros repositórios podem ser **persistentes**. Existem duas opções de repositórios persistentes: (i) as definições de processos que pertencem a um repositório são definidas **logicamente** e as informações sobre este repositório são armazenadas em um documento persistente que poderá ser recuperado posteriormente; e (ii) as definições de processo de um repositório são **armazenadas** em um mesmo container como, por exemplo, um diretório ou uma tabela de um banco de dados.

## 4.4 Funcionalidades de ferramentas de definição

Funcionalidades da ferramenta são os recursos que a ferramenta oferece aos seus usuários. As principais funcionalidades encontradas em ferramentas de definição são citadas a seguir (CASTROIANNI, 1995; ILOG, 2001; IBM, 2001; SHETH et al., 1996; WFMC, 1999).

### 4.4.1 Visões

São funcionalidades que permitem ao sistema disponibilizar para o usuário uma visualização da definição de processo no formato definido por uma representação do modelo de *workflow*.

Embora algumas ferramentas disponibilizem apenas uma visão, o mais desejável é que sejam disponibilizadas aos usuários **várias visões** da definição de processo. Cada visão deve mostrar as informações contidas em uma definição de processo segundo uma representação do modelo de *workflow*. Através das visões os usuários podem analisar e manipular um *workflow* modelado. Qualquer mudança feita em qualquer visão deverá ser atualizada e **sincronizada** na definição de processo e nas demais visões. Desta forma o usuário pode escolher a visão que lhe é mais compreensível para analisar ou alterar a definição de processo.

Cada visão do modelo define os elementos e atributos de uma representação que poderão ser visualizados/manipulados. Os valores dos elementos e atributos não mostrados em uma visão podem ser acessados por interfaces próprias para este fim e/ou através de outras visões.

### 4.4.2 Funcionalidades de edição

São recursos de edição de representações gráficas e textuais.

Os principais recursos de edição encontrados foram:

- **copiar, recortar e colar** elementos do *workflow*;
- **desfazer e refazer** ações;
- recursos de **impressão**;
- recursos de **zoom** que permitam o afastamento ou aproximação da visão do modelo;
- **barra de rolagem** nos eixos x e y;
- recursos que centralizem o *workflow* e **adaptem o zoom** e a barra de rolagem de forma que toda a *definição do processo* seja mostrada na tela;
- criar um **fundo quadriculado** na tela para facilitar o design gráfico;
- recursos que **localizam palavras** ou elementos dentro de uma *definição de processos*;
- **barra de status** onde informações sobre a ferramenta ou sobre o modelo são passadas ao usuário conforme o contexto da modelagem;
- **lentes**, este recurso gráfico permite que uma determinada área da visão (a área sob a lente) seja mostrada sob algum efeito de distorção ou ampliação de forma que os

elementos sob a lente sejam mostrados de forma ampliada ou com mais detalhes que os demais elementos da visão;

- **algoritmos que re-arranjam** a distribuição, forma e tamanho dos elementos do *workflow* segundo critérios como: minimizar o cruzamento de conectores, maximizar a simetria, padronizar o tamanho e a direção dos conectores, padronizar o tamanho dos nodos e distribuir de forma uniforme os elementos (CASTROIANI, 1995);
- **filtros**. São funcionalidades do sistema que permitem que o usuário aplique filtros sobre as visões. Com este recurso o usuário poderá ver em uma definição do processo somente os elementos que possuem as características que ele deseja analisar.

#### 4.4.3 Controles da definição de processo

São os recursos da ferramenta que permitem a manipulação de uma definição de processo.

As entidades e propriedades do modelo de *workflow* são representadas através de elementos e atributos. Portanto, a ferramenta deve proporcionar recursos que permitam a manipulação de elementos e atributos.

Para **manipular os elementos** foram encontrados recursos que permitem a:

- seleção de um elemento. Após selecionar um elemento o usuário poderá realizar sobre este elemento as demais manipulações;
- seleção de vários elementos. Ao selecionar vários elementos o usuário poderá executar, simultaneamente, sobre todos os elementos selecionados as demais manipulações;
- visualização dos elementos;
- adição de elementos;
- exclusão de elementos;
- alteração de um elemento.

Para **manipular os atributos** dos elementos foram encontrados recursos que:

- selecionam um ou mais atributos;
- mostram os atributos de um elemento;
- adicionam atributos aos elementos;
- alteram o valor dos atributos de um elemento;
- excluem os atributos dos elementos.

#### 4.4.4 Regras dos Controles

São recursos que controlam, monitoram ou corrigem os controles feitos sobre a definição de processo. Estas regras têm por objetivo manter a definição de processo correta e coerente durante sua manipulação.

No momento em que o usuário manipula um atributo ou elemento o sistema poderá detectar esta ação e reagir com novas ações automáticas. As principais regras sobre as manipulações do modelo encontradas em ferramentas de definição foram:

**A. verificar e reagir a erros de modelagem que estão sendo feitos pelo usuário.**

Muitas ferramentas reagem quando o usuário tenta modelar uma configuração incorreta de elementos ou atributos. As reações podem ser:

- mandar um aviso ao usuário;
- desfazer a ação do usuário fazendo com que o modelo volte ao seu estado anterior;
- desfazer a ação do usuário e executar novas ações que tornem válida a configuração de elementos e atributos do modelo.

São exemplos deste tipo de controle as situações onde o usuário tenta adicionar um elemento em um lugar da definição de processo onde o elemento não pode ser colocado. Nestes casos o sistema exclui o elemento adicionado. Em alguns casos, após excluir o elemento recém adicionado pelo usuário o sistema pode inclui-lo em um local da representação mais apropriado;

**B. recursos e restrições para prevenção de erros.**

São recursos que limitam a liberdade da ação do usuário a fim de prevenir que o mesmo cometa erros. São recursos como desabilitar botões e itens de menu que não podem ser usados em um determinado momento da modelagem; apresentar uma interface que só permita ao usuário inserir valores válidos; etc;

**C. estabelecer relacionamentos entre elementos.**

O usuário poderá estabelecer relacionamentos entre elementos do modelo. Os relacionamentos poderão ser tanto relacionamentos hierárquicos, onde um elemento está contido em outro, quanto relacionamentos associativos, onde um elemento é associado a outro. O estabelecimento de um relacionamento é uma atividade relativamente complexa, composta por várias manipulações simples (adição, alteração e exclusão de elementos e atributos). Ao detectar que o usuário quer estabelecer um relacionamento, o sistema pode realizar automaticamente as manipulações que compõem o estabelecimento deste relacionamento. Tal ação pode poupar esforços do usuário e o impede de cometer erros.

No caso de um relacionamento associativo, o sistema poderá automaticamente ajustar os valores dos atributos dos elementos para que se referenciem mutuamente da forma correta. Já no caso de um relacionamento hierárquico além de ajustar os atributos dos elementos pai e filho o sistema pode colocar o elemento filho no local apropriado (dentro do elemento pai);

**D. manter a coerência do modelo e de seus relacionamentos.**

Muitas vezes quando o usuário manipula um elemento ou atributo o sistema dispara uma série de ações. As ações irão adaptar a representação do modelo a nova configuração de elementos. Estas adaptações têm como objetivos: ajustar a visualização dos elementos, poupar trabalho do usuário realizando automaticamente algumas manipulações nos elementos, manter a coerência dos relacionamentos e manter a coerência lógica e visual do modelo. Este tipo de controle pode ocorrer, por exemplo, quando o usuário exclui um elemento associado a outro. O sistema então exclui os elementos associados ao elemento excluído pelo usuário. Outro

exemplo ocorre quando o usuário move um elemento dentro da representação e o sistema move junto com o elemento todos os seus elementos filhos.

#### 4.4.5 Reedição e reaproveitamento de *workflows*

São funcionalidades que permitem a reedição e o reaproveitamento de definições de processos modeladas a priori.

O usuário deverá ter a possibilidade de salvar o corrente estado da definição de processo e recarregá-lo novamente com a mesma configuração. Isto significa que as estruturas de dados da definição de processo devem ser armazenadas em um formato persistente para que possam ser recarregados mais tarde (CASTROIANNI, 1995). Fazem parte desta funcionalidade os recursos: **abrir**, **fechar**, **salvar** e **recarregar** uma definição de processo que está num formato que pode gerar visões.

#### 4.4.6 Importar/exportar arquivos

Exportar é o recurso que converte uma definição de processo que está num formato em que a ferramenta consegue gerar visões para um formato externo. Importar é o recurso que converte uma definição de processos que está em um formato externo para uma definição de processos que está num formato em que a ferramenta consegue gerar visões.

Assim como as visões, os arquivos exportados estão em um formato definido por uma representação do modelo de *workflow*, entretanto existem as seguintes diferenças entre visões e arquivos exportados:

- os arquivos exportados são gerados a partir de uma definição de processo e em seguida são armazenados em um formato persistente. Já as visões são geradas a partir da definição de processo e mostradas ao usuário. Em nenhum momento uma visão é armazenada em um formato persistente;
- através de uma visão se pode manipular uma definição de processo. Já através de um arquivo exportado não se pode manipular uma definição de processo;
- diferentemente das visões, os arquivos exportados após serem gerados não são sincronizados com as mudanças ocorridas na definição de processo original;
- diferentemente das visões, os arquivos exportados não podem ser manipulados pela ferramenta.

#### 4.4.7 Gerenciamento do repositório

São funcionalidades que permitem o gerenciamento de um repositório de definições de processos.

Em um repositório, cada definição de processo pode conter a descrição: de um ou vários processos, de um ou vários pacotes e das relações hierárquicas entre processos e pacotes. Um processo pode estar relacionado hierarquicamente com processos e pacotes definidos em definições de processos diferentes.

Cabe ao gerenciador do repositório reconhecer as suas definições de processos, os seus processos, os seus pacotes e as relações hierárquicas entre pacotes e processos. O gerenciador do repositório deverá proporcionar aos usuários da ferramenta os seguintes recursos:

- a **visualização dos processos** e pacotes de um repositório bem como a visualização da relação hierárquica entre estes elementos;
- o **acesso a um processo** (abrir o documento que contém um processo através da visualização da hierarquia de processos);
- a visualização, inclusão e exclusão dos documentos do repositório (**manipulação de documentos**);
- a **execução de procedimentos** que envolvem vários documentos;
- a **referência a pacotes externos**.

#### 4.4.8 Modularização

São funcionalidades que permitem a expansão e o colapso de processos em um número indefinido de níveis.

A expansão de um processo permite que o usuário visualize todo o conteúdo de um processo-filho que era apenas referenciado em seu processo-pai. O colapso de um processo permite que o usuário esconda o conteúdo de um processo deixando no processo-pai apenas uma referência ao processo-filho.

No conteúdo de um processo podem ser encontrados sub-processos que por sua vez podem conter outros sub-processos. A ferramenta deve fornecer aos usuários uma maneira ágil de navegar pela hierarquia de processos, possibilitando a visualização de um processo em diferentes níveis de detalhes.

Outros recursos desta funcionalidade permitem que o usuário visualize, insira, exclua ou altere a relação hierárquica entre os processos e a relação entre os processos e os pacotes. É a edição da hierarquia.

#### 4.4.9 Verificação de erros

São funcionalidades que permitem a verificação automática de erros de sintaxe e semântica existentes em uma definição de processo.

Esta funcionalidade disponibiliza ao usuário da ferramenta um recurso que analisa a sintaxe e a semântica de uma definição de processo. A verificação de erros também retorna ao usuário um relatório contendo os erros encontrados. Para facilitar a localização e correção de erros podem ser oferecidos aos usuários os seguintes recursos para:

- **destacar** os elementos e atributos que contém erros através de cores diferenciadas ou outros recursos gráficos;
- disponibilizar relatórios que possuem **links** em cada erro. Ao ser acionado um link de um erro os elementos que contém os correspondentes erros são selecionados.

Os tipos de erros detectados vão desde erros de sintaxe até erros semânticos e lógicos. Por exemplo, podem ser detectados elementos que nunca poderão ser alcançados durante a execução de um *workflow* ou elementos que podem ocasionar um *loop* infinito durante a execução do *workflow*.

#### 4.4.10 Simulação

São funcionalidades que permitem a simulação do comportamento do *workflow* modelado ainda na sua fase de construção.

Esta funcionalidade simula a execução de um *workflow*. Desta forma os *workflow* designers podem analisar o comportamento de um *workflow* sem que o mesmo seja realmente executado. Os principais recursos de simulação encontrados em ferramentas de definição foram:

- **modelagem da simulação.** Este recurso permite que cenários sejam montados. Tais cenários descrevem ambientes reais;
- **animação gráfica;**
- **armazenamento dos dados** colhidos durante a simulação;
- **geração de relatórios** (tabelas, gráficos, etc) sobre os dados colhidos durante a simulação;

Uma característica importante para as ferramentas de simulação é a **interoperabilidade** com os padrões definidos pela WfMC. Se há uma compatibilidade com os padrões da WfMC os dados sobre simulações poderão ser reutilizados por qualquer sistema que seja capaz de interpretar tais padrões.

#### 4.4.11 Recursos para cooperação

São funcionalidades que facilitam a modelagem cooperativa das definições de processos.

A modelagem de processos de *workflow* é uma atividade complexa que pode envolver uma equipe multidisciplinar formada principalmente por analistas de *workflows* e analistas de domínio. Para facilitar o trabalho de equipes, as ferramentas podem oferecer recursos para a edição cooperativa de definições de processos (PINHEIRO, 2002). A edição cooperativa de definições de processos é uma área de pesquisa ampla. Uma análise profunda dos recursos de cooperação destas ferramentas foge ao escopo deste trabalho. Entretanto, estudando as ferramentas existentes e a área de edição cooperativa de documentos, foram encontradas as seguintes funcionalidades que poderão auxiliar na edição cooperativa de modelos de *workflow* (OTSUKA, 1997):

- **comunicação:** são os recursos que permitem a comunicação síncrona e assíncrona entre os usuários. A comunicação pode ser feita através de recursos como *e-mails*, fóruns de discussões, sistemas de anotações, *chats*, etc;
- **percepção:** são recursos que fornecem informações sobre o que já foi feito, o que está sendo feito no momento e o que será feito no processo de edição de *workflows*;
- **coordenação:** são recursos que visam a coordenação das atividades a serem realizadas durante a edição de processos. Entre outras soluções pode-se utilizar mecanismos de controle de mudanças e distribuição de versão e a própria tecnologia de *workflow* para coordenar a edição de processos de *workflow*;
- **compartilhamento:** são os recursos que disponibilizam aos usuários uma área comum onde todos podem acessar documentos, objetos e informações sobre a edição de processos;

- **edição cooperativa:** são os recursos que permitirão que o usuário acesse e edite uma definição de processos.

#### 4.4.12 Adaptabilidade

Adaptabilidade é a funcionalidade que permite a adaptação da ferramenta às necessidades e preferências de seus usuários.

Uma ferramenta de definição deve atender as necessidades e preferências de seus usuários, porém estas necessidades e preferências variam muito de usuário para usuário. Até mesmo as necessidades de um único usuário variam ao longo do tempo. Por isso é importante que estas ferramentas possuam recursos que permitam a sua adaptação.

As adaptações discutidas neste trabalho ocorrem após a ferramenta ser instalada e configurada em uma organização. São adaptações feitas geralmente por seus usuários (Analistas de *workflow*) ou pelo próprio sistema (neste caso o termo mais adequado é adaptatividade do sistema). Estas adaptações são executadas através de uma interface ou arquivo de configuração. As adaptações já alteram o sistema ou o seu comportamento a partir do momento em que são feitas. Não há a necessidade de se re-compilar o sistema ou outro processo típico da fase de desenvolvimento de um *software*. Os principais recursos deste tipo de adaptabilidade encontradas em ferramentas de definição de *workflow* foram:

- **adaptação dos símbolos** (ícones ou figuras geométricas) que representam as entidades de *workflow* em visões com representação gráfica. Através deste recurso cada usuário ou organização pode escolher os símbolos que lhes são mais convenientes para representar as entidades de *workflow* em representações gráficas;
- **configuração das funcionalidades** da ferramenta. Muitas das funcionalidades da ferramenta podem ser parametrizáveis possuindo interfaces próprias para a sua configuração;
- **adaptação dos itens da interface.** Pode-se escolher entre mostrar ou esconder itens de menus, menus *pull-down* e barras de ferramentas. Ao esconder um dos itens citados o mesmo não aparecerá na interface com o usuário o que poderá deixar a interface mais clara e menos poluída. Por outro lado, ao esconder um item da interface, o usuário não terá acesso às funcionalidades e facilidades acionadas a partir dos itens de interface escondidos. Usa-se o recurso de esconder um item da interface quando uma funcionalidade não é utilizada ou quando o usuário prefere uma interface mais clara e com um menor número de funcionalidades. Também é possível re-arranjar a distribuição, hierarquia e localização dos menus e itens de menu.
- **adaptação da interface.** Adaptações relacionadas com a aparência da interface. São adaptações como: tamanho e tipo de letra dos títulos de menus, cor de janelas, cor dos formulários, cor de fundo das visões, etc;
- **adaptação e integração da ferramenta com os dados da empresa.** Na maior parte dos processos é necessária a modelagem dos aplicativos e participantes de uma organização. Pode-se modelar estes elementos em todas *definições de processo* de forma independente. Entretanto, algumas ferramentas estão integradas com os dados da organização. A ferramenta acessa um banco de dados ou arquivo com os dados da organização e mostra ao usuário uma interface já adaptada a estes dados. O usuário só poderá escolher aplicativos e participantes que constam nesta base de

dados. Quando a base de dados ou o arquivo é alterado a ferramenta automaticamente adapta-se as mudanças e altera sua interface. Pode-se alterar a base de dados com informações sobre aplicativos e participantes tanto a partir da ferramenta de definição quanto a partir de outros aplicativos da organização.

#### 4.4.13 Interoperabilidade

São as funcionalidades do sistema que mantém a ferramenta e suas *definições de processo* compatíveis com os principais padrões de *workflow* e de sistemas de informação existentes.

A análise da interoperabilidade de uma ferramenta pode ser dividida em três sub itens:

- a interoperabilidade do modelo de *workflow* adotado pela ferramenta;
- a interoperabilidade da representação do modelo; e
- a interoperabilidade tecnológica da ferramenta.

##### 4.4.13.1 Interoperabilidade do modelo

O modelo de *workflow* adotado pela ferramenta deverá ser compatível com os principais padrões e conceitos da tecnologia de *workflow*.

A WfMC é a organização responsável por definir os principais conceitos e padrões de *workflow*. Por tanto, o principal item a ser avaliado é a compatibilidade das entidades do modelo analisado com as entidades definidas pela WfMC (WfMC, 1999).

Para ter uma boa interoperabilidade cada entidade do modelo analisado precisará ser mapeada para as entidades do modelo da WfMC. Caso o mapeamento completo não ocorra o modelo perde interoperabilidade. As informações não mapeadas não poderão ser colocadas em uma *definição de processo* do padrão da WfMC. Em casos mais extremos a impossibilidade de um mapeamento pode indicar que o modelo analisado não é um modelo de *workflow*. Estes modelos podem não conter os fundamentos mínimos para que a WfMC e a comunidade da área o considere um modelo de *workflow*.

O padrão definido pela WfMC prevê principalmente as entidades já comentadas no capítulo 4.3.1 (atividades; roteamento; participante; aplicativos; dados relevantes ao processo de *workflow*; extensão; processo e pacote)

Entidades como, por exemplo, transação e exceções não são referenciadas no padrão da WfMC. Contudo são utilizadas e referenciadas nos principais sistemas e projetos da área (EDER; LIEBHART, 1995; CASATI et al., 1996). Para manter a compatibilidade com a WfMC estes sistemas e projetos propõem várias técnicas de mapeamento destas entidades para as entidades da WfMC (WfMC, 1999). Muitos realizam o mapeamento completo. Porém, existem outros que realizam um mapeamento parcial o que ocasiona uma perda de informações quando a definição do processo é exportada para o formato da WfMC.

##### 4.4.13.2 Interoperabilidade da representação

O principal item a ser avaliado neste tipo de interoperabilidade é a **compatibilidade da representação do modelo com os padrões WPDL e XPD**L definidos pela WfMC.

A EBNF WPDL (*Workflow Process definition Language*) é o formato padrão de representação de *workflows* da WfMC e o XPDL é a versão XML da WPDL.

Outro item importante a ser avaliado é a compatibilidade da representação do modelo com a tecnologia XML. O XML é o padrão de transferência de dados mais utilizado pelos principais sistemas da área. Vários *modelos de workflow* são representados por um vocabulário XML (THATTE, 2001; LEYMAN, 2001; BPMI, 2003) (W3C, 2002).

O XPDL é compatível com a tecnologia XML e é a alternativa de representação de processos de *workflow* mais indicada pela WfMC (WFMC, 1999).

Uma ferramenta de definição é compatível com o padrão XPDL se for capaz de importar e exportar arquivos no formato XPDL.

Portanto, para uma representação ter uma boa interoperabilidade é preciso que todos seus elementos e atributos possam ser mapeados para atributos e elementos da representação XPDL (e vice-versa).

O XPDL possui em seu modelo o conceito “Extensão”. Ele é representado pelo elemento “<ExtendedAttribute>”. Através do “ExtendedAttribute” pode-se representar em XPDL informações não previstas neste modelo.

Quando o “ExtendedAttribute” é utilizado no mapeamento de um elemento para o formato XPDL, pode-se ter como resultado uma representação em XPDL válida. Porém, outros sistemas que foram desenvolvidos para compreender uma definição de processo escrita em XPDL não interpretarão corretamente o “ExtendedAttribute” sem que sejam modificados para isto. Caso modificações no sistema não sejam feitas os “ExtendedAttribute” são ignorados.

Portanto o uso dos “**ExtendedAttributes**” no mapeamento de elementos diminui a interoperabilidade do sistema.

Uma ferramenta de definição pode representar, importar e exportar vários formatos de representação. Quanto maior o **número de formatos** que podem ser utilizados/gerados pela ferramenta maior é a sua interoperabilidade. Além do XPDL, destacam-se os formatos: XLang da Microsoft (THATTE, 2001), WSFL da IBM (LEYMAN, 2001), BPML da BPMI (BPMI, 2003), WSCI da Sun/BEA (W3C, 2002) e a BPEL4WS (LEYMANN; ROLLER, 2002) que é a junção ou evolução da XLang e WSFL.

#### 4.4.13.3 Interoperabilidade tecnológica da ferramenta.

Além da compatibilidade da ferramenta com os principais padrões de *workflow* existentes é importante que a ferramenta seja compatível com outras tecnologias como, por exemplo:

- protocolo HTTP: permite que a ferramenta utilize recursos de um servidor de http. Através do http os usuários poderão acessar e editar uma definição de processos via Internet utilizando a *World Wide Web* (WWW);
- protocolo WebDAV: este protocolo estende o protocolo http e fornece aos seus usuários recursos para acessar e editar cooperativamente uma definição de processos via Internet;

- protocolo SMTP: permite que a ferramenta utilize recursos de um servidor de SMTP. Através do smtp a ferramenta pode, por exemplo, enviar e receber *e-mails* automaticamente;
- protocolo FTP: permite que a ferramenta utilize recursos de um servidor de FTP. Através do FTP a ferramenta pode disponibilizar aos seus usuários recursos de acesso, compartilhamento e manipulação de arquivos;
- Web Services: permite que a ferramenta utilize recursos de um servidor de Web Services. Através do Web Services a ferramenta poderá utilizar recursos ou sistemas remotos;
- XML: Permite que a ferramenta utilize os recursos da tecnologia XML. A tecnologia XML inclui uma ampla gama de aplicativos, padrões e linguagens (BRAY et al., 2004) que podem ser utilizadas pela ferramenta ou aplicadas sobre suas *definições de processos*.

Esta não é uma lista exaustiva de tecnologias que podem ser compatíveis com ferramentas de definição. A lista de tecnologias que devem ser compatíveis com a ferramenta de definição depende muito do domínio e do contexto onde a mesma será aplicada. Cabe aos desenvolvedores e usuários analisar cada contexto e definir que tecnologias melhorariam a usabilidade da ferramenta neste contexto.

## 4.5 Os modelos de interação de ferramentas de definição

Os modelos de interação descrevem os recursos do sistema que possibilitam a interação entre o usuário e a ferramenta. Os principais objetivos do modelo de interação são: mostrar ao usuário quais são as funcionalidades da aplicação oferecidas pelo sistema e como ele pode interagir (fisicamente e através de sua percepção) com cada uma delas (SOUZA et al., 1999; CASTROIANNI, 1995).

### 4.5.1 Dispositivos de Hardware

Os dispositivos de hardware são os dispositivos físicos com os quais os usuários interagem com o sistema realizando atividades motoras e perceptivas. Os dispositivos de hardware estão divididos em:

- dispositivos de entrada: são os dispositivos de hardware pelos quais o usuário envia informações ao sistema;
- dispositivos de saída: são os dispositivos de hardware pelos quais o sistema envia informações ao usuário.

Em ferramentas de definição de *workflow* os principais dispositivos de entrada encontrados foram o mouse e o teclado. E os principais dispositivos de saída encontrados foram o monitor e a impressora.

### 4.5.2 O *software* da interface

Além dos dispositivos de hardware, uma interface do usuário também necessita de recursos de *software*. Estes recursos de *software* são chamados de *software* da interface. Os *softwares* da interface devem realizar as seguintes funções:

- controlar os dispositivos de hardware;

- construir e gerenciar os elementos da interface com os quais o usuário poderá interagir;
- gerar os símbolos e mensagens que representarão as informações que o sistema quer passar para o usuário;
- interpretar as interações dos usuários.

O *software* da interface é formado por um conjunto complexo de rotinas, bibliotecas e sub sistemas. Porém, toda esta complexidade deve ser transparente para o usuário que deverá interagir com o sistema através de um determinado paradigma ou **estilo de interação**.

#### 4.5.3 Estilos de interação

Existem diferentes tipos de interfaces pelas quais o usuário pode interagir com o sistema. Na literatura da área estas diferenças são chamadas de estilos de interação. Os principais estilos de interação encontrados em ferramentas de definição são descritos a seguir.

#### 4.5.4 O estilo de interação menu

A interface orientada por menu permite que o usuário selecione, através do mouse ou teclas alfanuméricas, as funções oferecidas pelo sistema. As funções do sistema são oferecidas ao usuário na forma de um menu de opções mostradas no monitor. Os menus podem formar uma hierarquia de menus onde um item de menu pode ser o título de um sub menu com vários sub itens.

Os principais tipos de menus utilizados em ferramentas de definição de processos estão citados abaixo:

- ***pull-down***: são menus geralmente localizados na parte superior da tela, seus títulos são sempre visíveis, a sua localização dentro da janela não varia. São utilizados para acessar as funções mais importantes do sistema;
- ***pop-up***: são menus que aparecem no local onde está o cursor. Seus itens variam de acordo com o contexto e configuração do sistema no momento em que foi chamado;
- **barra de tarefas**: são menus cujos títulos são geralmente ícones. A sua localização pode ser variável. São utilizados para acessar as funções mais utilizadas pelo usuário.

O esforço do usuário neste tipo de interface concentra-se mais no reconhecimento das funções do que na recordação de comandos e funções. Para minimizar os esforços do usuário os seguintes itens devem ser bem planejados:

- os textos e símbolos que serão mostrados nos títulos e itens dos menus;
- a apresentação, distribuição, agrupamento e hierarquia de menus.

Em sistemas com muitas funcionalidades, um mau planejamento dos itens acima pode deixar os usuários perdidos ou com dificuldades de navegar pelas inúmeras estruturas de menus.

Exemplos deste tipo de estilo de interação são os menus dos *softwares* Microsoft Word e Microsoft Excel.

#### 4.5.5 O estilo de interação formulário

A interface no estilo formulário permite que o usuário envie informações ao sistema através de uma tela que lembra um formulário tradicional de papel. Geralmente, são interfaces de fácil aprendizado e podem prover aos usuários facilidades como: a prevenção e correção de erros de digitação, a verificação dos dados digitados, recursos que visem aumentar a produtividade e minimizar o esforço físico e cognitivo dos usuários, etc.

São exemplos de estilo de interação do tipo formulário as interfaces de inserção de dados em um banco de dados e interfaces de configuração de alguns sistemas como o formulário de configuração de impressões do *software* Microsoft Word.

#### 4.5.6 O estilo de interação manipulação direta

A interface no estilo manipulação direta permite que o usuário interaja diretamente com os dados ou representações dos elementos da aplicação. Na manipulação direta não há a necessidade de se digitar ou escolher um comando. Os comandos da aplicação são as ações que o usuário desempenha diretamente com os elementos da interface.

As interfaces gráficas que utilizam a metáfora da área de trabalho, como o Microsoft Windows, são exemplos de estilo de interação do tipo manipulação direta. A manipulação direta do Microsoft Windows permite que os usuários interajam com o gerenciador de arquivos do sistema operacional. A interação é feita através da manipulação de ícones que representam arquivos, diretórios, discos e outros componentes computacionais. O usuário envia comandos ao sistema através de ações como as de arrastar e soltar os ícones utilizando geralmente o *mouse*.

#### 4.5.7 Hipertexto

Na interface do estilo hipertexto, um documento pode conter trechos chamados de nós. Os nós estão ligados a outros trechos (também chamados de nós) de outros documentos ou do próprio documento. Os nós são interligados por elos (links). Através dos elos o usuário pode recuperar ou acessar o conteúdo de um nó destino a partir de um nó origem.

São exemplos de estilo de interação do tipo hipertexto alguns sistemas de ajuda (*Help*) e a interface de navegação por documentos da World Wide Web (WWW).

#### 4.5.8 Os estilos de interação em ferramentas de definição

Embora existam exceções, geralmente em uma única ferramenta de definição são utilizados os quatro estilos de interação já citados: menus, formulários, manipulação direta e hiperdocumentos.

O estilo de interação “menus” é geralmente utilizado para acionar as diversas funcionalidades da ferramenta. Normalmente, na parte superior da ferramenta, um menu *Pull-Down* provê acesso a maioria das funções do sistema.

Também são disponibilizados alguns menus *pop-up* e barras de ferramentas, dentre as quais se destaca a barra de ferramentas para construção de representações gráficas de um processo. Através desta barra de ferramentas o usuário insere elementos em uma visão com representação gráfica. Para inserir um elemento gráfico na visão, geralmente o usuário clica em um item da barra de tarefas e depois na área de trabalho da visão.

O estilo de interação “formulário” é geralmente utilizado para se adicionar, excluir ou alterar dados dos atributos ou sub-elementos de um elemento. A importância e utilização deste estilo aumentam quando os valores que devem ser preenchidos nos atributos e sub-elementos são mais complexos, mais rígidos, tendo que obedecer a várias regras ou até mesmo seguir uma determinada gramática. Este estilo de interação é o mais apropriado para o preenchimento de valores complexos por poder proporcionar ao usuário facilidades como:

- a prevenção de erros oferecendo ao usuário uma interface onde só é possível preencher o formulário com valores válidos;
- a verificação e correção de erros no preenchimento de dados do formulário;
- o preenchimento automático de campos do formulário, minimizando assim o esforço físico e cognitivo do usuário.

Os formulários também são muito utilizados em algumas interfaces que visam configurar funcionalidades da ferramenta.

O estilo de interação “manipulação direta” é utilizado em representações gráficas da definição de processo. Estas representações geralmente são grafos onde:

- os nodos são ícones ou figuras geométricas que representam elementos do *workflow* como, por exemplo, as atividades;
- os arcos são figuras geométricas que representam as associações entre os elementos.

O grafo é mostrado no monitor através de uma visão e poderá ser manipulado pelo usuário através do mouse ou teclado.

Ao manipular um nodo ou arco que representa um elemento de *workflow* o usuário está manipulando este elemento na definição.

Normalmente, através do mouse o usuário pode definir o tamanho e local onde um elemento será criado. Pode-se selecionar, mover e redimensionar um elemento além de determinar relacionamentos entre elementos criando-se arcos orientados entre os nodos do grafo. Em algumas manipulações com o mouse é necessário o acionamento simultâneo de teclas especiais do teclado.

Usualmente, através do teclado o usuário pode excluir um elemento selecionado ou alterar textos mostrados na representação gráfica. Os textos mostrados no grafo geralmente representam um atributo de um elemento que esteja próximo ou em baixo do texto.

O estilo de interação “hipertexto” é utilizado em algumas interfaces que permitem que o usuário navegue pela hierarquia de processos de um *workflow*. Geralmente um processo raiz possui elementos que são nós para outros processos. Os usuários podem navegar pelos nós dos processos clicando em seus links. Desta maneira o usuário pode facilmente acessar toda a hierarquia de processos de um *workflow* ou alguns de seus detalhes tais como atributos de elementos, comentários ou arquivos anexados contendo informações sobre um elemento.

#### **4.5.9 Padrões**

Um padrão é um conjunto de regras, normas e recomendações que foram propostas com o objetivo de uniformizar e manter a consistência da aparência e comportamento de interfaces gráficas.

Um padrão não define como a interface é implementada nem como seria a sua arquitetura. O padrão apenas define aspectos relacionados com a interação usuário-sistema. Isto implica em definir a aparência e o comportamento dos tipos de interface com o usuário.

Definir a aparência consiste em determinar uma padronização da visualização de cada um dos recursos de interface (menus, formulários, mensagens, botões, janelas, etc) e da combinação destes recursos. Definir o comportamento consiste em determinar uma padronização das reações e modificações da aparência dos recursos de interface durante o processo de interação usuário-sistema.

Esta padronização e uniformização fazem com que a interface seja interpretada mais facilmente pelo usuário. Se todo menu *pull-down* possuir o mesmo comportamento, o usuário saberá mais facilmente como interagir com um menu *pull-down* qualquer.

São exemplos de padrões: o padrão Motif da Open System Foundation (OSF) (Open Group, 2004), os padrões OpenLook e Swing da Sun Microsystems e o padrão Windows da Microsoft.

A adoção de um padrão em ferramentas de definição é um requisito importante, pois fornece ao sistema uma maior padronização. O melhor padrão para cada sistema será o padrão que melhor se adapte ao contexto da ferramenta e as preferências e costumes dos usuários.

## **4.6 Tabela de descrição de modelos conceituais de ferramentas de definição**

Foi elaborado nesta dissertação um método de descrição de modelos conceituais de ferramentas de definição. Tal método consiste do preenchimento da tabela de modelos conceituais de ferramentas de definição conforme as instruções da tabela 4.1. Esta tabela sintetiza os itens que devem ser descritos/definidos em um modelo conceitual de ferramentas de definição. Para cada item é mostrado o que deve ser definido. Os itens da tabela foram explicados em detalhes ao longo do capítulo 4, sendo que, para cada item foi dedicado um sub-capítulo. Os sub-capítulos descrevem e destacam em negrito o que deve ser definido em seu correspondente item.

A tabela de modelos conceituais pode dar uma visão geral e concisa sobre modelos conceituais de ferramentas de definição. Porém, também pode servir como um guia para descrições mais detalhadas.

TABELA 4.1 - Tabela de descrição de modelos conceituais de ferramentas de definição

| <b>Itens do modelo</b>             | <b>O que descrever em cada item</b>  |
|------------------------------------|--|
| <b>Conceitos</b>                   |  |
| Modelo de <i>workflow</i>          | Citar os conceitos de <i>workflow</i> definidos no modelo de <i>workflow</i> adotado pela ferramenta, os principais itens são: atividade, roteamento, participante, aplicativos, dados relevantes, aspectos temporais, prioridade, versão, extensão, processo, pacote, repositório, exceções, informações gráficas, eventos, transações e recursos |
| Representação do modelo            | Citar as representações do modelo adotadas pela ferramenta   |
| Definição de processo              | Citar as representações que definem o formato das definições de processos que podem ser interpretadas, importadas ou exportadas pela ferramenta.   |
| Repositório                        | Citar características gerais do repositório da ferramenta (caso a ferramenta utilize este conceito)  |
| <b>Funcionalidades</b>             |  |
| Visões                             | Citar as características gerais de cada visão. Citar se são sincronizadas.   |
| Funcionalidades de edição          | Citar as funcionalidades de edição disponibilizadas. As principais funcionalidades são: copiar, recortar, colar, desfazer ações, refazer ações, impressão, zoom, barra de rolagem, localizar palavras, barra de status, algoritmos de re-arranjo, filtros, lentes, fundo quadriculado e adaptação do zoom.   |
| Controles da definição de processo | Citar se possui as funcionalidades: manipular os elementos; manipular os atributos;  |
| Regras dos Controles               | Citar se possui as funcionalidades: verificar e reagir a erros; prevenção de erros; estabelecer relacionamentos; manter a coerência  |
| Reedição e reaproveitamento        | Citar se possui as funcionalidades: abrir, fechar, salvar e recarregar   |
| Exportação                         | Citar os formatos que podem ser exportados   |
| Importação                         | Citar os formatos que podem ser importados   |
| Gerenciamento do repositório       | Citar se possui as funcionalidades: manipulação de documentos, execução de procedimentos, referenciar pacotes externos, visualização dos processos, acesso a um processo   |
| Modularização                      | Citar se possui as funcionalidades: navegar pela hierarquia, editar hierarquia   |
| Verificação de erros               | Citar se possui as funcionalidades: verifica sintaxe, verifica semântica, relatório de erros, destaca erros, links para erros  |
| Recursos para cooperação           | Citar recursos de: comunicação, percepção, coordenação, compartilhamento, edição cooperativa   |
| Simulação                          | Citar os recursos de: modelagem da simulação, animação gráfica, armazenamento dos dados, geração de relatórios e interoperabilidade  |
| Adaptabilidade                     | Citar se possui as funcionalidades: adaptação dos símbolos, configuração das funcionalidades, adaptação dos itens, adaptação da interface, integração com os dados da empresa  |
| Interoperabilidade                 | Citar informações sobre a interoperabilidade do modelo, a interoperabilidade da representação e a interoperabilidade tecnológica.  |
| <b>Interação</b>                   |  |
| Dispositivos de hardware           | Citar dispositivos de entrada e de saída   |
| Estilos de Interação utilizados    | Citar os estilos de interação utilizados na ferramenta   |
| Padrão de interação                | Citar o padrão de interação utilizado na ferramenta  |

Com isso, um método que pode auxiliar a compreensão, criação e descrição de modelos conceituais de ferramentas de definição foi disponibilizado a usuários e desenvolvedores. Tal método foi a principal contribuição e o objetivo deste capítulo.

## 5 O AMAYA WORKFLOW

### 5.1 Introdução

Com base nos estudos do capítulo anterior, foi definido durante esta dissertação o modelo conceitual de uma ferramenta de definição. A ferramenta chama-se Amaya *Workflow* (AW). Uma descrição do seu modelo conceitual é mostrada neste capítulo.

Também foi implementado um protótipo com as características do modelo conceitual proposto. Ainda não foram implementadas todas as características descritas no modelo conceitual. Porém o protótipo AW (TELECKEN, 2004; TELECKENb et al.,2002; TELECKEN et al., 2002) é funcional e está pronto para o uso.

Este capítulo está constituído da seguinte forma: primeiramente é descrita a especificação de requisitos do AW; a seguir são descritos os conceitos, funcionalidades e interações do modelo conceitual proposto. Finalizando, o modelo conceitual proposto é descrito conforme a tabela de descrição de modelos conceituais de ferramentas de definição proposta no capítulo 4. Nesta tabela também são destacados, e a seguir comentados, os itens do modelo conceitual AW que ainda não foram implementados no protótipo AW.

### 5.2 Especificação de requisitos

#### 5.2.1 Conceitos da área de domínio de *workflow*

São os conceitos definidos no capítulo 2 desta dissertação.

#### 5.2.2 Os usuários

O Amaya *Workflow* (AW) é uma ferramenta de definição que auxiliará analistas de *workflow* e analistas de domínio na definição e análise de processos em um ambiente que poderá ser distribuído e colaborativo. Poderão utilizar a ferramenta desde usuários experientes (na tecnologia de *workflow* e no uso da ferramenta) até iniciantes, pois o AW disponibilizará recursos que adaptarão as funcionalidades da ferramenta as características do usuário.

#### 5.2.3 O contexto

Poderão ser modelados na ferramenta diversos tipos de processo para as mais diversas organizações. A limitação do tipo de processo que poderá ser modelado na ferramenta é a limitação que é proposta no modelo da WfMC (WFMC, 1999). Ou seja,

o AW foi projetado para ser tão abrangente e genérico quanto o padrão previsto pela WfMC para este tipo de ferramenta. Tal limitação exclui alguns tipos de processos que atendem a certos nichos de aplicações, mas inclui, segundo a WfMC, os processos de *workflow* do tipo AdHoc, administrativo e de produção mais utilizados.

Uma análise mais completa sobre a abrangência do modelo da WfMC é feita nos artigos de Aalst (2003) e de Casati et al. (1996). Nestes artigos são apontados recursos e limitações do modelo da WfMC. Os referidos artigos também dão uma boa noção das potencialidades e deficiências do modelo da WfMC e conseqüentemente do modelo do AW.

O AW será compatível e poderá interagir com tecnologias padrões da WfMC e do XML, além de protocolos da web como o http e o WebDAV.

Por fim, a característica comum a todas tecnologias e padrões utilizadas no AW é que são todas abertas e de livre distribuição. Portanto, o AW também é um *software* de livre distribuição e de código aberto.

## 5.3 Conceitos do domínio de *workflow* representados no AW

### 5.3.1. O modelo de *workflow*

O modelo de *workflow* do AW foi baseado principalmente no meta-modelo da WfMC e no modelo *process model* definido no projeto WIDE.

A figura 5.1 mostra o diagrama de classes do modelo de *workflow* proposto pela WfMC (WfMC, 1999).

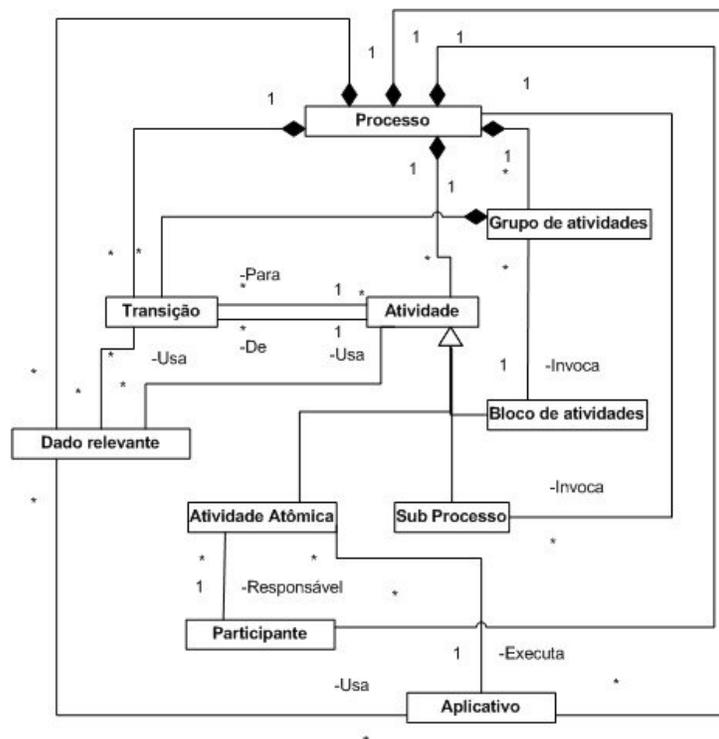


Figura 5.1: Diagrama de classes do modelo de *workflow* proposto pela WfMC

O *process model* foi introduzido no capítulo 3.1. O diagrama de classes do *process model* é mostrado na figura 5.2 (CASATI et al., 1996):

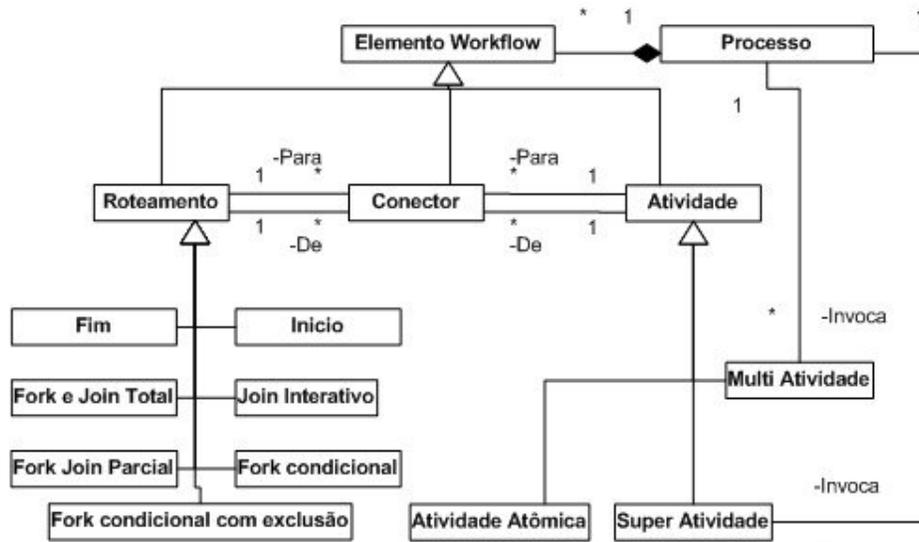


Figura 5.2: Diagrama de classes do *process model*

Já o diagrama de classes do modelo AW é mostrado na figura 5.3.

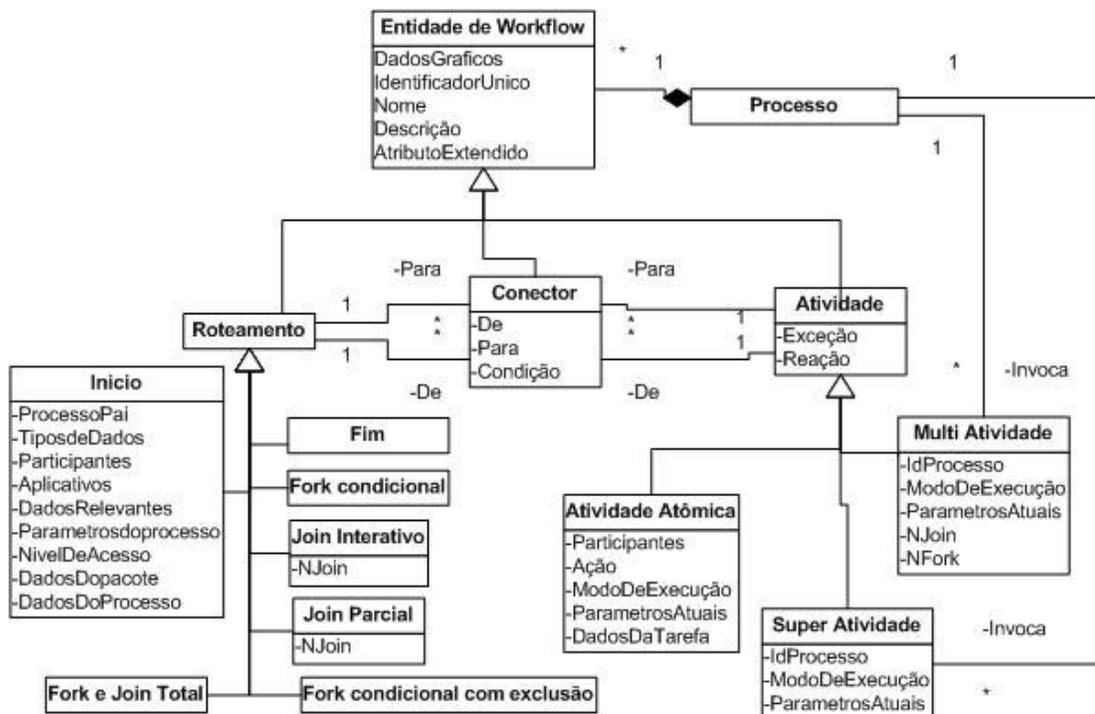


Figura 5.3: Diagrama de classes do modelo AW

Ao analisar o modelo AW, nota-se que sua estrutura de entidades é muito semelhante à estrutura do *process model*. O modelo AW tem as mesmas entidades de

*workflow* que as entidades do *process model*. O que vai diferenciá-los são as propriedades de cada entidade e os relacionamentos entre estas propriedades. Em sua maioria, tais propriedades têm a mesma função que entidades do modelo da WfMC.

As entidades de *workflow* do AW possuem a mesma semântica que as entidades do *process model*. A exceção é a entidade início. As demais entidades possuem apenas as funções determinadas no *process model*. Já a entidade início acumula as funções previstas no *process model* e a função de conter informações sobre o processo, conforme será detalhado adiante.

Por fim o modelo da WfMC e o AW também representam a entidade de *pacote* a qual foi modelada a parte para tornar o diagrama mais claro. A figura 5.4 mostra um diagrama de classes da entidade *pacote* dos modelos AW e da WfMC (WfMC, 1999).

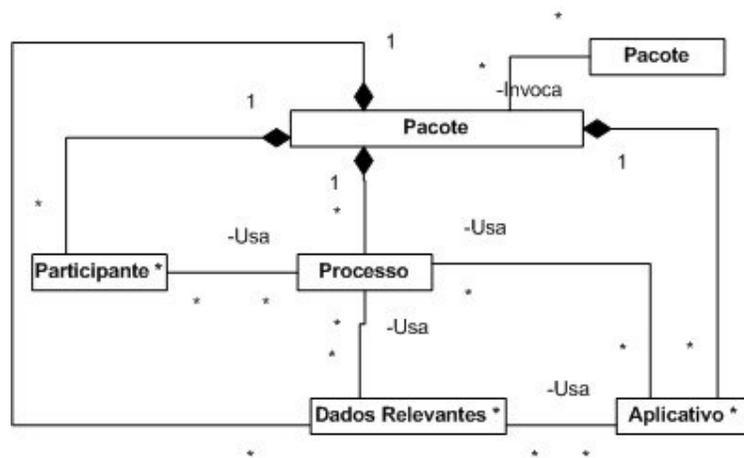


Figura 5.4: Diagrama da entidade pacote dos modelos WfMC e do AW.

\* Entidades que podem ser redefinidas na entidade processo.

### 5.3.1.1 Análise dos modelos AW, process model e da WfMC

A tabela 5.1 compara as principais entidades dos três modelos e mostra suas equivalências. As colunas mostram as entidades de cada modelo. As linhas mostram como uma entidade é definida em cada modelo. Em fundo branco estão as entidades de cada modelo. Quando o fundo de uma célula é cinza significa que o modelo da sua coluna não definiu o tipo de entidade da sua linha. Porém no fundo cinza, são informados recursos alternativos que podem representar uma entidade não definida.

Tabela 5.1: Comparação entre os modelos

| Tipos de Entidades                        | Entidade do Modelo AW                              | Entidade do <i>process model</i>  | Entidade do Modelo da WfMC            |
|---|--|---|---------------------------------------|
| <b>Roteamento</b>                         | Conector   | Conector  | Transições                            |
|   | Início   | Início  | Combinação de atividades e transições |
|   | Fim  | Fim   |                                       |
|   | Join parcial                                       | Join parcial  |                                       |
|   | Fork e join total                                  | Fork e join total   |                                       |
|   | Join interativo                                    | Join interativo   |                                       |
|   | Fork condicional                                   | Fork condicional  |                                       |
|   | Fork condicional com exclusão mutua                | Fork condicional com exclusão mutua   |                                       |
| <b>Atividades</b>                         | Multi Tarefa                                       | Multi Tarefa  |                                       |
|   | Super Tarefa                                       | Super Tarefa  | Sub Processo                          |
|   | Tarefa   | Tarefa  | Atividade Atômica                     |
|   | Combinação de atividades e elementos de roteamento | Combinação de atividades e elementos de roteamento                              | Bloco de Atividades                   |
| <b>Principais Informações do workflow</b> | Aplicativo   | Informações contidas nos modelos Organizacional e Informacional do projeto WIDE | Aplicativo                            |
|   | Participante                                       |   | Participante                          |
|   | Dado Relevante                                     |   | Dado Relevante                        |
| <b>Pacote</b>                             | Pacote   |   | Pacote                                |
| <b>Grupo de Atividades</b>                | Combinação de processos interligados               | Combinação de processos interligados  | Grupo de Atividades                   |
| <b>Processo</b>                           | Processo   | Processo  | Processo                              |

#### A) WfMC X Process model

Comparando-se os modelos *process model* e da WfMC constata-se que as entidades *processo* e *conector* são comuns aos dois modelos.

Ambos tem *atividades* de vários tipos. Os tipos *tarefa* e *super tarefa* são comuns aos dois modelos. O modelo da WfMC não possui a entidade *multi tarefa*, porém pode-se simular o comportamento de uma *multi tarefa* no modelo da WfMC através de uma combinação de *atividades* e *transições* da WfMC. De forma análoga o *process model* não possui a entidade *blocos de atividade*, porém pode-se simular o comportamento de um *bloco de atividades* no modelo WIDE através da combinação de *atividades* e *elementos de roteamento*. Um *bloco de atividades* invoca um *grupo de atividades* que no *process model* pode ser simulado como uma combinação de *processos* interligados.

O modelo da WfMC não tem a maioria das entidades de *roteamento* do *process model*. Estes roteamentos são feitos no modelo da WfMC por *atividades* ou combinação de *atividades* e *transições*.

Já o *process model* não possui as entidades de informações do *workflow* (*aplicativo*, *participante*, *dado relevante*, etc). No projeto WIDE estas informações são armazenadas em outros modelos (no modelo organizacional e no modelo informacional) (CASATI et al., 1996). Conforme mostra a tabela 5.1, pode-se mapear as entidades do *process model* para as entidades do modelo da WfMC. Em (BARESI et al., 1999) este mapeamento é detalhado.

## B) AW

Para se definir o modelo AW partindo-se do *process model* deve-se:

- adicionar na entidade início do process model as propriedades: aplicativos, dados relevantes, participantes, tipos de dados, parâmetros do processo, dado do processo, dado do pacote, nível de acesso e pacotes. As propriedades adicionadas possuem a mesma semântica que entidades do modelo da WfMC. A correspondência entre as propriedades do modelo AW e as entidades do modelo da WfMC com a mesma semântica é mostrada na tabela 6.2 do capítulo 6;
- adicionar em todas entidades de workflow a propriedade dados gráficos: esta propriedade foi introduzida no AW com a finalidade de armazenar as informações gráficas de cada elemento de workflow. Os valores da propriedade dados gráficos definirão a aparência gráfica das suas entidades de workflow;
- adicionar em todas entidades de workflow a propriedade erro: esta propriedade foi introduzida no AW para armazenar informações sobre erros de modelagem de cada elemento;
- adicionar em todas entidades de workflow a propriedade atributo estendido: esta propriedade tem a mesma função que a entidade extensão do modelo da WfMC;
- adicionar nos diferentes tipos de atividades as propriedades dados da tarefa, modo de execução e parâmetros atuais: as propriedades adicionadas possuem a mesma semântica que entidades do modelo da WfMC. A correspondência entre as propriedades do modelo AW e as entidades do modelo da WfMC com a mesma semântica é mostrada na tabela 6.2 do capítulo 6.

Com as características acima descritas o AW une em um único modelo a riqueza de representação de elementos de roteamento do *process model* com a riqueza de representação de informações de *workflow* do modelo da WfMC. E ainda pode armazenar informações gráficas capazes de descreverem a aparência proposta na representação gráfica do *process model*.

Conforme mostra a tabela 5.1, assim como o *process model* é compatível com o modelo da WfMC, o modelo AW é compatível com ambos. Pode-se mapear as entidades do modelo AW tanto para o *process model* quanto para o modelo da WfMC.

### 5.3.2 A representação do modelo de *workflow*

Foram desenvolvidas três representações do modelo AW: a representação AW, a representação estruturada e a representação gráfica.

#### 5.3.2.1 A representação AW

A representação AW descreve o modelo AW através do padrão XML. Esta representação é o formato padrão de armazenamento de informações da ferramenta AW. É através de definições de processo que estão neste formato que o AW pode gerar visões e documentos exportados. O AW também disponibiliza uma visão da representação AW.

A estrutura de elementos e atributos da representação AW é mostrada na figura 5.5.

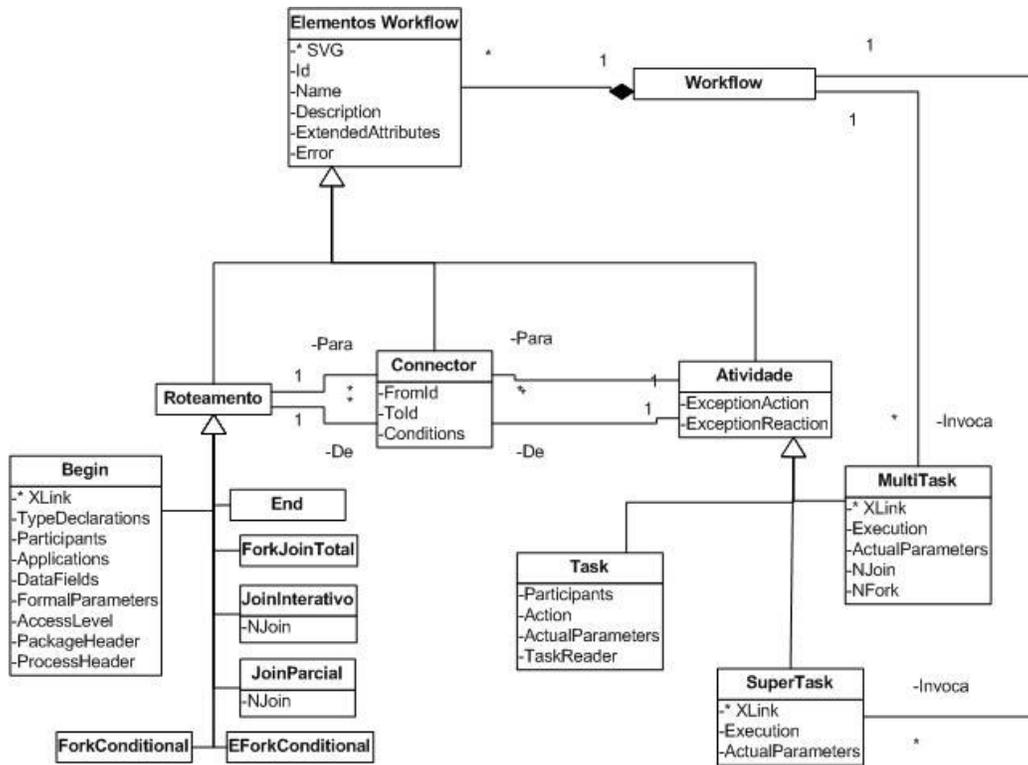


Figura 5.5: Estrutura de elementos e atributos da representação AW

Neste diagrama de classes, o atributo precedido por \* pode ser um ou vários atributos de outras linguagens XML. Estes atributos são adicionados na representação AW através do padrão Namespace. Os atributos especiais são:

- **\*XLink**: atributos da linguagem XLink que descrevem um *link* entre processos;
- **\*SVG**: atributos da linguagem SVG que descrevem a aparência gráfica dos elementos.

A figura 5.6 mostra um *workflow* segundo a representação AW.

Informações mais detalhadas sobre a sintaxe e a semântica da representação AW são mostradas no capítulo 6.

```

publish.svg <2>
File Edit Types Links Views Style Special Attributes Annotations Workflow Help
1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <!-- Created by amaya 6.1, see http://www.w3.org/Amaya/ -->
3 <svg xmlns="http://www.w3.org/2000/svg">
4   <wbegin stroke="black" fill="black" stroke-width="5" id="element" y="36px"
5     x="318px" width="90px" height="6px" wWork=". /editdocument.svg"
6     xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink"
7     xlink:href="editdocument.svg"/>
8   <wfjtotal stroke="black" fill="white" id="element1" cy="97px" cx="363px"
9     r="25px"/>
10  <wtask stroke="black" fill="white" id="element2" rx="5px" y="145px"
11    x="168px" width="130px" height="50px" wName="Choose address"/>
12  <wconnector stroke="black" fill="none" id="connector"
13    points="363,39 363,72 370,59 363,72 355,59" wFromID="element"
14    wToID="element1"/>
15  <wconnector stroke="black" fill="none" id="connector1"
16    points="363,122 298,170 312,168 298,170 303,156"
17    wFromID="element1" wToID="element2"/>
18  <wconnector stroke="black" fill="none" id="connector2"
19    points="363,122 432,167 425,153 432,167 417,166"
20    wFromID="element1" wToID="element3"/>
21  <wconnector stroke="black" fill="none" id="connector3"

```

Figura 5.6: Exemplo de código da representação AW

### 5.3.2.2 A representação estruturada

A representação estruturada é utilizada em uma das visões do AW.

A estrutura de elementos e atributos da representação estruturada é a mesma da representação AW (ver figura 5.5). A semântica e os nomes dos atributos e elementos também são os mesmos para as duas representações. A diferença é que a representação AW utiliza apenas textos que estão distribuídos conforme o padrão XML. Já a representação estruturada utiliza textos, recursos gráficos e endentações para mostrar a estrutura de seus documentos. Nesta visão fica mais clara a hierarquia entre os elementos e atributos de um documento. A figura 5.7 mostra um *workflow* segundo a representação estruturada do AW.

```

<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20001102//EN"
"http://www.w3.org/TR/2000/CR-SVG-20001102/DTD/svg-20001102.dtd">
<!-- Created by amaya 6.1, see http://www.w3.org/Amaya/ -->
svg xmlns="http://www.w3.org/2000/svg"
  wbegin stroke=black fill=black stroke-width=5 id=element y=36px x=318px width=90px height=6px wWor
    xlink:href=editdocument.svg
  switch
    foreignObject width=15px height=20px y=225px x=185px
      div dir=ltr
        p
          Publish
  wftotal stroke=black fill=white id=element1 cy=97px cx=363px r=25px
  wtask stroke=black fill=white id=element2 rx=5px y=145px x=168px width=130px height=50px wName=C
  wconnector stroke=black fill=none id=connector points=363,39 363,72 370,59 363,72 355,59 wFromID=
  wconnector stroke=black fill=none id=connector1 points=363,122 298,170 312,168 298,170 303,156 w
  wconnector stroke=black fill=none id=connector2 points=363,122 432,167 425,153 432,167 417,166 w

```

Figura 5.7: Exemplo de código da representação estruturada

### 5.3.2.3 A representação gráfica

A representação gráfica do AW é um grafo onde os nós são símbolos que representam entidades de *workflow* como atividades, *forks* e *joins*. Já os arcos orientados do grafo representam conectores que determinam a seqüência ou paralelismo dos elementos em um *workflow*. A representação gráfica do AW é semelhante semântica e sintaticamente com a representação gráfica do *process model* mostrada no capítulo 3.1.

A representação gráfica do AW é utilizada em uma visão gráfica do AW.

A figura 5.8 mostra um *workflow* segundo a representação gráfica do AW.

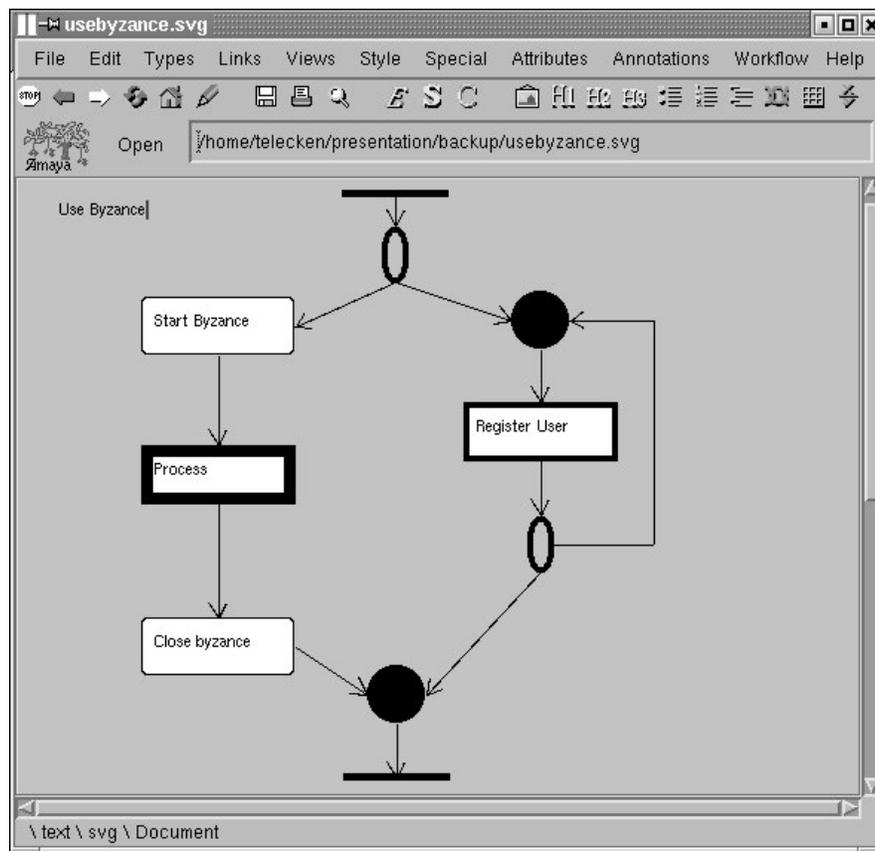


Figura 5.8: Exemplo de código da representação gráfica

### 5.3.3 A definição de processo

As definições de processo do AW são armazenadas no formato definido pela representação AW. É através de definições de processos que estão neste formato que o AW pode gerar visões de um processo de *workflow*. O AW também é capaz de gerar definições de processos que estão no formato definido pela representação XPD. As definições de processo no formato XPD são geradas através de um processo de exportação.

## 5.4 Funcionalidades do AW

### 5.4.1 Visões

O AW disponibiliza três visões aos seus usuários: gráfica, estruturada e AW. Cada visão mostra a definição de processo conforme as representações: gráfica, estruturada e AW respectivamente. Através das visões o usuário pode ver e analisar três diferentes representações de uma definição de processo. Através das três visões o usuário também poderá editar a definição de processo. As visões do AW são sincronizadas. Ao editar uma visão o usuário estará executando uma operação na definição de processo que atualizará as demais visões. A figura 5.9 mostra três visões de um mesmo processo. Também se pode notar que nas três visões está selecionado o mesmo elemento.

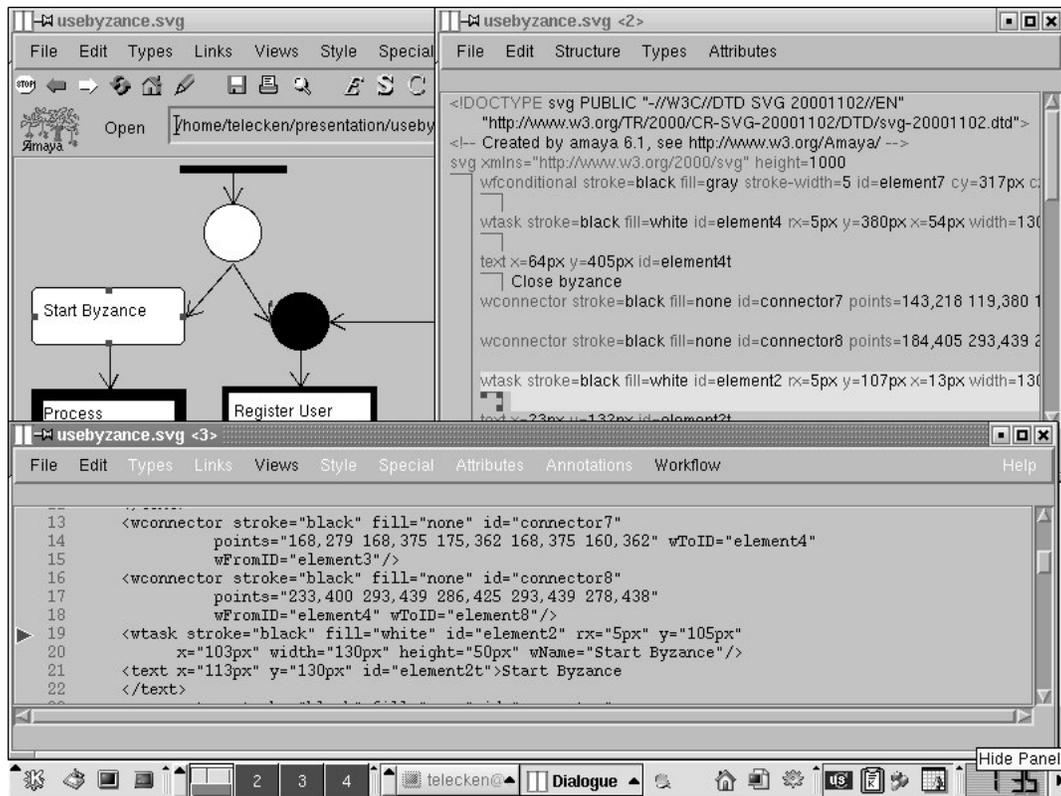


Figura 5.9: Visões do AW

A seguir mais detalhes sobre cada visão.

#### 5.4.1.1 Visão gráfica

A visão gráfica mostra o conteúdo da definição de processo segundo o formato definido na representação gráfica do AW. Nesta visão poderão ser utilizados os diversos recursos de edição gráfica do AW. Tais recursos serão detalhados no capítulo 5.5.

#### 5.4.1.2 Visão estruturada

A visão estruturada mostra o conteúdo da definição de processo segundo o formato definido na representação estruturada. Nesta visão poderão ser utilizados vários recursos de edição estruturada do AW. Tais recursos serão detalhados no capítulo 5.5.

#### 5.4.1.3 Visão AW

A visão AW mostra o conteúdo da definição de processo segundo o formato definido na representação AW. Como a estrutura da definição de processo também está definida pela representação AW, a visão AW mostra exatamente o código fonte da definição de processo. Nesta visão o usuário tem liberdade de editar qualquer texto do código e poderá utilizar os recursos de edição de texto do AW. Tais recursos serão detalhados no capítulo 5.5.

### 5.4.2 Funcionalidades de edição

São disponibilizadas no AW as seguintes funcionalidades de edição:

- copiar, recortar e colar elementos e atributos das representações de *workflow*;
- desfazer e refazer ações;
- recursos de impressão;
- recursos de *zoom* que permitam o afastamento ou aproximação de uma visão;
- barra de rolagem nos eixos x e y;
- recursos que localizam palavras ou elementos dentro de uma definição de processos;
- barra de status;
- algoritmos que re-arranjam automaticamente a distribuição, forma e tamanho dos elementos do *workflow*.

### 5.4.3 Controles da definição de processo

Os controles da definição de processo são as operações que podem ser feitas sobre uma definição de processo através das visões e interfaces da ferramenta.

Para manipular os elementos de uma definição de processo o AW possui recursos que permitem a:

- seleção de um elemento qualquer. Após selecionar um elemento o usuário poderá realizar sobre este elemento as demais operações;
- seleção de vários elementos. Ao selecionar vários elementos o usuário poderá executar simultaneamente sobre todos os elementos selecionados as demais operações;
- visualização de elementos;
- adição de qualquer elemento;
- exclusão de qualquer elemento;
- alteração de um elemento.

Para manipular os atributos de uma definição de processo o AW possui os seguintes recursos:

- selecionar um atributo;
- mostrar os atributos de um elemento;
- adicionar atributos aos elementos;
- alterar o valor dos atributos de um elemento;
- excluir os atributos dos elementos.

### 5.4.4 Regras dos Controles

O AW possui as seguintes regras dos controles da definição de processo:

A. **verificar e reagir a erros de modelagem que estão sendo feitos pelo usuário:**

- se o usuário inserir um elemento *Connector* que não relaciona dois elementos válidos, o AW exclui automaticamente o *Connector* inserido incorretamente;
- um *Connector* do AW é uma seta formada por vários pontos. Se o usuário exclui um ponto desta seta o AW redesenha a seta sem o ponto excluído;
- há um número mínimo de pontos que um *Connector* deve ter. Se o usuário excluir um ponto de um *Connector* que possui apenas o número mínimo de pontos, o conector inteiro é excluído pelo AW;
- se o usuário executar alguma alteração no código da *definição do processo* que deixe o código "mal formado", segundo os padrões XML, o sistema da uma mensagem de erro. A mensagem informa para o usuário o erro ocorrido e a linha do código onde o erro ocorreu. Após, o elemento que está mal formado é destacado (o código deste elemento é pintado de vermelho);
- quando o usuário exclui o elemento *Begin*, o sistema envia uma mensagem pedindo a confirmação da exclusão. Se o usuário confirmar a exclusão, a exclusão é terminada. Se o usuário não confirmar, a exclusão é desfeita. O *Begin* é um elemento que pode armazenar muitas informações sendo muito importante no modelo. Este procedimento procura evitar a exclusão involuntária do elemento *Begin*;

#### **B. recursos e restrições para prevenção de erros:**

- quando já existe um elemento *Begin* na representação, o AW desabilita os recursos que permitem a inserção de outro elemento *Begin*. Não pode existir mais de um elemento *Begin* em uma *definição de processos* válida;
- muitas interfaces para o preenchimento de atributos só permitem que o usuário adicione valores válidos aos atributos. São por exemplo recursos onde o usuário só precisa escolher um valor de uma lista de valores válidos. Outras interfaces apesar de não impedirem que o usuário cometa erros podem diminuir a ocorrência de erros através de interações mais "amigáveis" e intuitivas;
- as visões gráfica e estruturada não permitem que o usuário altere o nome dos elementos e atributos. Os atributos e elementos que podem ser adicionados são somente os elementos e atributos previstos no modelo. Só é possível excluir um elemento ou atributo inteiro. Não existem exclusões parciais. Estas restrições previnem que o usuário torne a *definição de processo* "mal formada";

#### **C. estabelecer relacionamentos entre elementos:**

- ao adicionar um *Connector* em uma representação gráfica, o AW detecta se existem elementos próximos as extremidades do *Connector*. Se existirem elementos próximos às extremidades do *Connector* o AW captura os atributos *Ids* destes elementos e os coloca como valores dos atributos *FromId* e *ToId* do *Connector*. Assim é criado automaticamente um relacionamento entre os elementos da extremidade do *Connector* e o próprio *Connector*;

#### **D. manter a coerência do modelo e de seus relacionamentos:**

- se o usuário excluir um elemento de um processo de *workflow* o AW também exclui os *Connectors* que estavam relacionados com este elemento. O motivo desta exclusão é que após o elemento ser excluído os *Connectors* continuarão apontando para o elemento que não existe mais na representação;

- movimentar um ponto da extremidade de um *Connector* pode tornar a aparência gráfica da *definição de processo* incoerente. Se um ponto de uma extremidade do *Connector* for movimentado, o AW desfaz a ação do usuário fazendo com que o ponto movimentado volte a sua localização original;
- movimentar ou redimensionar um elemento (exceto o *Connector*) pode tornar a aparência gráfica da *definição de processo* incoerente. Se um elemento é movimentado ou redimensionado. O AW automaticamente redefine as posições dos pontos das extremidades dos *Connectors* que estavam relacionados com o elemento movimentado ou alterado. Deste modo a coerência gráfica do modelo é mantida.

#### 5.4.5 Reedição e reaproveitamento de *workflows*

O AW fornece recursos que permitem o usuário abrir, salvar, fechar e recarregar uma definição de processo. Numa definição de processo podem ser armazenadas todas as estruturas de dados necessárias para que o AW possa salvar e posteriormente recarregar um *workflow* modelado. O formato de armazenamento é definido pela representação AW.

A funcionalidade "Abrir definição de processo" permite que o usuário acesse o conteúdo de uma definição de processo modelada a priori. Após ser aberta uma definição de processo pode ser editada.

A funcionalidade "Salvar definição de processo" permite que o usuário salve em um formato persistente o atual estado das estruturas de dados do processo que está sendo modelado.

Quando o usuário utiliza a funcionalidade "Fechar definição de processo" o AW fecha as visões que estiverem abertas e salva a definição de processo se a mesma ainda não foi salva.

A funcionalidade "Recarregar definição de processo" permite que o usuário feche e abra em seguida a atual definição de processo.

#### 5.4.6 Exportação

O AW pode exportar para os seguintes formatos:

- **XPDL:** toda definição de processo válida pode ser exportada para o formato XPDL. Para exportar para o formato XPDL o AW primeiramente verifica se o *workflow* modelado é um *workflow* válido. Se for um *workflow* inválido a exportação é suspensa e uma mensagem é enviada ao usuário informando os erros encontrados. Se for um *workflow* válido o AW gera um documento XPDL com a descrição do *workflow* modelado. O arquivo XPDL gerado pode ser enviado para uma máquina de *workflow* padrão da WfMC;
- **SVG:** qualquer *workflow* modelado pode ser exportado para o formato SVG. Para executar esta exportação o AW gera um documento no formato SVG que descreve a aparência da representação gráfica de um *workflow* modelado. Um documento SVG pode ser interpretado por qualquer visualizador SVG como, por exemplo, o Batik SVG e Adobe SVG Viewer;
- **postscript:** qualquer *workflow* modelado pode ser exportado para o formato *Postscript*. Para executar esta exportação o AW gera um documento no formato

*Postscript* que descreve a aparência da representação gráfica de um *workflow* modelado. Um documento *Postscript* pode ser interpretado por qualquer visualizador *Postscript* como, por exemplo, o GSView e o Adobe Acrobat Distiller.

### 5.4.7 Importação

O AW pode ler um documento no formato XPDL e gerar uma definição de processo no formato da representação AW. O documento no formato AW poderá então, gerar visões e ser manipulado pelo AW. Este procedimento chama-se importação de arquivo XPDL. Caso sejam encontrados erros no arquivo XPDL a importação é suspensa e uma mensagem de erro é mostrada ao usuário.

### 5.4.8 Gerenciamento do repositório

Neste capítulo, primeiramente será apresentado como o AW organiza documentos, processos e pacotes e após serão mostradas as funcionalidades do AW para gerenciar estes itens.

#### 5.4.8.1 Organização de documentos e processos no Amaya Workflow

No AW um repositório é representado por um diretório que conterà todos os documentos que descrevem um *workflow*. Cada documento do AW é uma definição de processo que pode conter apenas um processo. Assim se um *workflow* for formado por vários processos estes processos serão armazenados em vários documentos (um documento para cada processo).

Os processos de um *workflow* formam uma hierarquia de processos. Um processo pode conter um segundo processo que por sua vez pode conter outros processos e assim por diante. Entretanto, esta hierarquia é apenas lógica. Todos documentos que pertencem a um mesmo *workflow* devem estar em um mesmo diretório. O que determina a hierarquia dos processos/documentos são referências contidas em cada processo. Os processos possuem referências para seus processos-filhos e uma referência para o seu processo-pai.

Em especial existe um processo chamado **processo raiz**. Este processo não referencia um processo-pai. Todos os demais processos são descendentes do processo raiz. O documento que contém o processo raiz é o único documento do diretório que conterà informações sobre o pacote e é chamado de **documento raiz**. As informações sobre um pacote são similares as informações sobre um processo. A diferença está no escopo das informações. As informações sobre um processo são compartilhadas e acessadas somente pelo próprio processo. Já as informações sobre um pacote podem ser compartilhadas e acessadas por todos processos do diretório.

A maneira como o usuário manipula as informações de um processo e a maneira como o usuário manipula as informações de um pacote também são similares. O cuidado que o usuário deve ter é apenas saber se as informações que ele quer inserir devem ser acessadas somente por um processo ou por todos os processos. Sabendo isto basta o usuário colocar as informações no processo raiz ou num outro processo.

O usuário pode transformar um processo raiz em um processo comum e vice-versa. Para transformar um processo raiz em um processo comum basta o usuário colocar no processo raiz uma referência a um processo-pai. Já para transformar um processo comum em um processo raiz basta o usuário retirar do processo comum a referência ao

processo pai. O que caracteriza um processo raiz é a ausência de uma referência ao processo pai.

Tanto na transformação processo raiz – processo comum quanto na transformação contrária não são perdidas informações nem são necessárias outras adaptações.

A figura 5.10 mostra uma estrutura de documentos, processos e pacotes de um *workflow* feito no Amaya *Workflow*. Nesta figura as setas mostram as referências que um processo faz para os seus processos filhos.

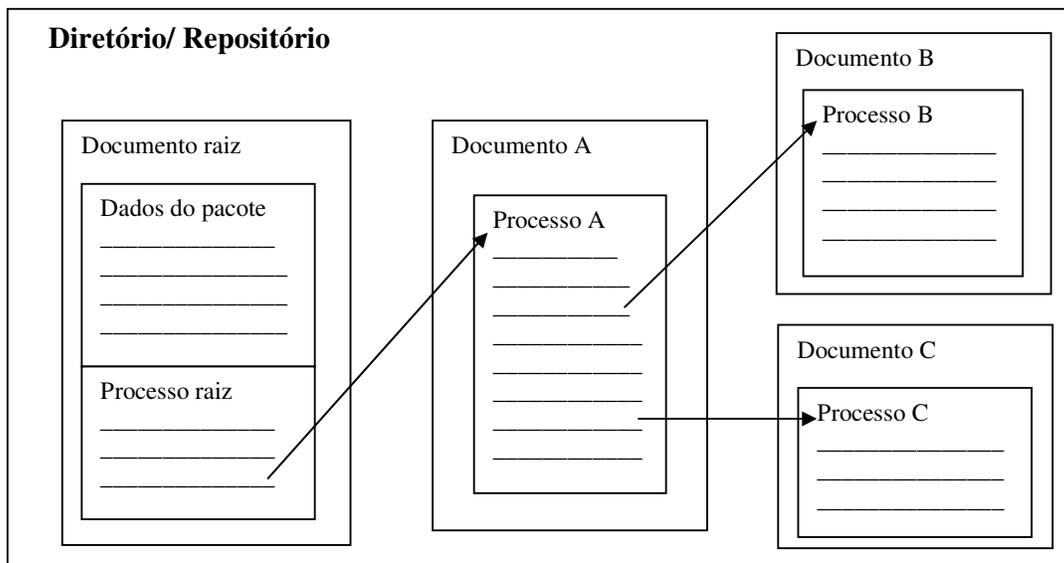


Figura 5.10: Estrutura de documentos, processos e pacotes de um *workflow*

O processo raiz não possui informações próprias. O processo raiz utiliza informações do pacote. Já um processo comum utiliza informações próprias e do pacote.

#### 5.4.8.2 Recursos de gerenciamento do repositório

Um *workflow* pode estar todo contido em um único processo e conseqüentemente em um único documento. Neste caso o usuário não precisa utilizar os recursos da funcionalidade de gerenciamento do repositório, pois as demais funcionalidades da ferramenta são suficientes para a realização de qualquer operação disponibilizada pelo sistema.

Contudo se um *workflow* for formado por vários processos e conseqüentemente por vários documentos, o usuário pode utilizar os seguintes recursos de gerenciamento do repositório:

- **verificação de erros do pacote:** este recurso verifica (i) se existe um e somente um documento raiz, (ii) se todas referências para processos-filhos e processos pais estão corretas, e (iii) se todos processos do diretório são descendentes diretos ou indiretos do processo raiz. Se forem encontrados erros o sistema mostra os erros ao usuário. Caso contrário o sistema mostra uma mensagem dizendo que a estrutura e as referências do pacote estão corretas;
- **geração de um arquivo XPD L para um pacote com vários processos:** inicialmente esta função faz uma verificação de erros do pacote. Se forem encontrados erros o sistema mostra os erros ao usuário e suspende a geração do arquivo XPD L. Se não forem encontrados erros o sistema monta um arquivo XPD L

que contém numa só *definição de processo* os dados do pacote e de todos os processos do *workflow* de um determinado diretório.

Um processo pode referenciar processos que estão em pacotes externos. Os pacotes externos podem estar na mesma máquina do corrente pacote ou em máquinas externas que podem ser acessadas via rede interna, internet, etc. As referências a pacotes externos são feitas através de atributos do AW que possuem a sintaxe semelhante a elementos da representação XPDL. A função de verificação de erros do pacote do AW não verifica se as referências externas estão corretas.

#### 5.4.9 Modularização

Para definir a hierarquia de processos de um *workflow* e permitir a fácil navegação do usuário por esta hierarquia o AW utilizou *links* nos processos. Os *links* de um processo apontam para o seu processo-pai e para seus processos-filhos. É através destes *links* que o usuário pode navegar pela hierarquia de processos.

O usuário pode acessar os processos-filhos do atual processo dando um duplo clique nos elementos *SuperTask* ou *MultiTask*. Estes elementos possuem *links* que apontam para um processo-filho. Quando um elemento *SuperTask* ou *MultiTask* é clicado, o AW fecha o atual processo e carrega o processo-filho solicitado.

Quando um *SuperTask* ou *MultiTask* é inserido em uma representação o AW mostra ao usuário uma interface. Nesta interface deve ser informado o caminho de um processo-filho do atual processo.

Para alterar o processo-filho apontado por um *SuperTask* ou *MultiTask*, é necessário que o usuário altere o valor do atributo *href* do atual elemento. O valor do *href* é o caminho de um processo-filho.

Para acessar o processo pai do atual processo, basta o usuário dar um duplo clique no elemento *Begin*. Este elemento possui um *link* que aponta para o processo-pai do atual processo. Quando um elemento *Begin* é clicado o AW fecha o atual processo e carrega o processo-pai.

Quando um elemento *Begin* é inserido em uma representação, o AW mostra ao usuário uma interface. Nesta interface deve ser informado o caminho do processo-pai do atual processo.

Para alterar o processo-pai apontado pelo *Begin*, é necessário que o usuário altere o valor do atributo *href* do *Begin*. O valor do *href* é o caminho do processo-pai.

#### 5.4.10 Verificação de erros

O AW fornece ao usuário um recurso que verifica os erros de modelagem de um processo e retorna ao usuário mensagens informando os erros encontrados. O AW verifica erros de sintaxe e lógicos.

Os recursos de verificação de sintaxe incluem: (i) a verificação de que o código é "bem formado" segundo o padrão XML; (ii) a verificação de que o código está de acordo com a estrutura de elementos e atributos da representação AW; e (iii) a verificação de que os valores dos atributos estão corretamente preenchidos.

Os recursos de verificação de lógica e semântica incluem principalmente a verificação de que os elementos estão referenciando-se corretamente. As referências entre os elementos devem ser somente referências para *Ids* de elementos que estão no

processo e o conjunto das referências não deve formar grupos isolados de elementos nem configurações de elementos que não estejam previstos no modelo *process model*.

Após verificar os erros, o AW retorna uma mensagem ao usuário com uma lista dos erros encontrados. Caso nenhum erro seja encontrado, uma mensagem dizendo que não foram encontrados erros no *workflow* modelado é mostrada ao usuário. Caso erros ocorram, além da mensagem com a lista de erros, a visão gráfica destaca os elementos que contém erros pintando-os de vermelho. Se o usuário acessar o atributo *Error* de um elemento pintado de vermelho o usuário vai verificar que este atributo foi preenchido pelo AW com uma descrição dos erros referentes a este elemento. Com isto, o usuário pode identificar com maior eficácia os elementos com erros e a descrição dos erros de cada elemento.

Para acionar a função de verificação de erro, o usuário precisa chamar esta função através de um menu. A função de verificação de erro pode ser chamada a qualquer momento pelo usuário. O próprio AW também chama esta função automaticamente sempre que o usuário solicita que o atual *workflow* seja exportado para o formato XPDL. Neste caso o AW só exporta para o formato XPDL se não for encontrado nenhum erro nos processos envolvidos.

No capítulo 5.4.4, que descreve a funcionalidade "Regras dos controles", são descritos alguns erros que podem ser cometidos pelo usuário e que são monitorados pelo AW. Estes erros são corrigidos ou destacados no momento em que o usuário comete o erro. Este recurso também pode ser considerado um recurso de verificação de erros. Porém com características diferenciadas tais como a monitoração e a reação imediata à ocorrência do erro.

#### 5.4.11 Adaptabilidade

O AW permite que os símbolos que representam as entidades de *workflow* na visão gráfica da ferramenta sejam trocados por outros símbolos. A possibilidade de troca dos símbolos permite que cada usuário ou organização possa escolher os símbolos que lhes são mais convenientes para representar entidades de *workflow* na visão gráfica.

Para trocar a aparência gráfica de um símbolo sempre que o mesmo é inserido em qualquer *definição de processo*, deve-se alterar a figura correspondente no diretório de configuração. Este diretório contém as figuras que são adicionadas como símbolos dos elementos em uma definição de processo. Ao substituir uma figura neste diretório, será trocada a aparência de um símbolo de *workflow* nas definições de processo.

As funcionalidades impressão, anotações e serviços da internet (HTTP, WebDAV, proxy, etc) podem ser adaptadas através de interfaces de configurações.

A interação do usuário com a ferramenta também pode ser adaptada. Através de interfaces de configuração o usuário poderá mudar a aparência de quase toda interface da ferramenta: estilo dos botões, fontes utilizadas em menus e mensagens, linguagem da interface (inglês, francês, espanhol, etc), cores da interface (cor de fundo, cor das fontes, cor dos botões, etc), definição de teclas de atalho, entre outras opções.

Uma importante adaptação da interação usuário-ferramenta que o AW disponibiliza é a possibilidade de se definir quais os menus, itens de menus, paletas e botões poderão ser acessados pelos seus usuários. Qualquer item da interface pode ser escondido ou mostrado para o usuário. A definição de quais itens aparecerão ou não em uma interface é feita em um arquivo de configuração. Quando o usuário altera um item no arquivo de

configuração ele poderá ver o resultado fechando e abrindo a ferramenta. Não há a necessidade de recompilar a ferramenta ou qualquer outro processo típico da fase de desenvolvimento do *software*.

Esconder itens da interface deixa a mesma mais clara e diminui a probabilidade de ocorrerem erros. Contudo ao esconder um item da interface, o usuário passará a não ter acesso às funcionalidades e facilidades acionadas a partir dos itens de interface escondidos. Caberá aos usuários analisar a melhor configuração para a sua interface.

Para ajudar os usuários a escolher e adaptar a sua interface é disponibilizado o recurso de se definir e escolher "perfis de usuário" (*profiles*). A definição dos perfis é feita através de um arquivo de configuração. A escolha e ativação de um perfil pode ser feita através de uma interface de configuração. O AW disponibiliza previamente três perfis: iniciante, intermediário e avançado. Cada um destes perfis é destinado a usuários com uma determinada experiência e habilidade em utilizar o AW.

A figura 5.11 mostra telas do AW com dois *profiles* diferentes.

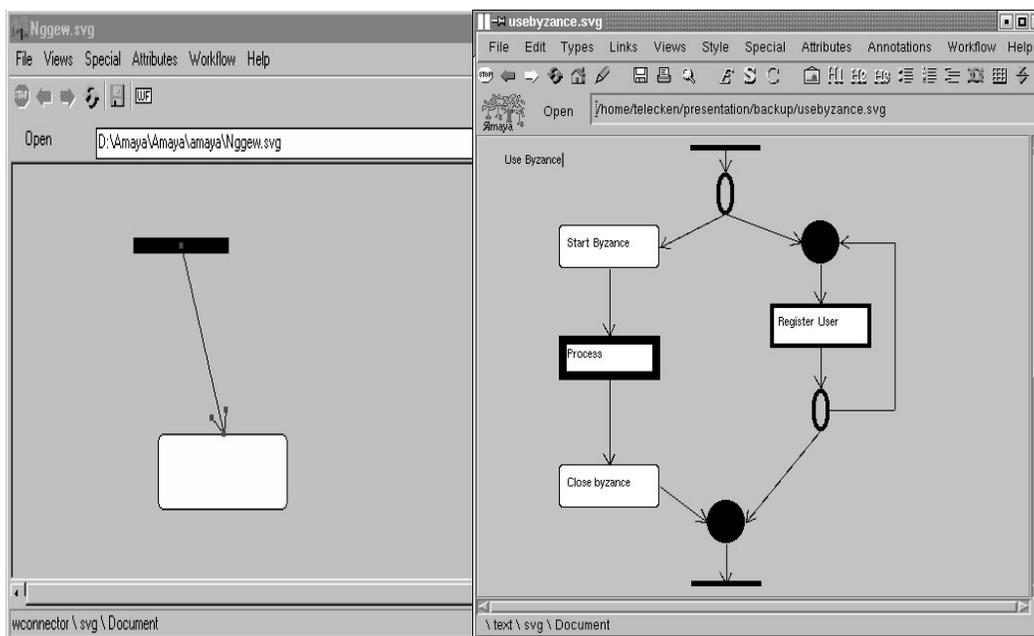


Figura 5.11: *Profiles* do AW

### 5.4.12 Simulação

O AW permite que cenários de simulação sejam modelados.

Os dados de simulação que podem ser modelados no AW estão definidos no modelo da WfMC (WFMC, 1999). Através do módulo de exportação do AW, os dados de simulação podem ser exportados para o formato XPDL. Desta forma documentos gerados pelo AW poderão ser utilizados em ferramentas externas que interpretam XPDL e que possuam recursos de simulação de *workflows*.

### 5.4.13 Recursos para cooperação

O AW possui recursos que permitem a edição cooperativa de *definições de processos* via Internet. São recursos do Amaya que podem ser utilizados no AW para a edição de *workflow*.

O AW é um cliente do protocolo HTTP. A definição de processo que está sendo editado no AW pode ser salva em um servidor HTTP distante. E uma definição de processo que está em um servidor HTTP distante pode ser carregado no AW para que seu usuário a edite. Desde que tenham acesso a internet, várias pessoas podem carregar e salvar uma mesma *definição do processo*. A segurança sobre quem poderá ou não acessar um arquivo fica por conta do servidor que poderá restringir o acesso aos arquivos através de senhas e *logins* que poderão ser informados através de uma interface do AW.

Além de recursos tradicionais de clientes HTTP o AW também possui os recursos de um cliente do protocolo WebDAV. O WebDAV é uma extensão do protocolo HTTP que provê recursos para a edição cooperativa de documentos. Um servidor WebDAV disponibiliza aos seus usuários um diretório onde devem ser colocados documentos destinados à edição cooperativa.

O AW pode acessar este diretório e utilizar os recursos do protocolo WebDAV descritos a seguir. Ao acessar uma *definição do processo* o usuário poderá trancá-la (*lock*) se achar necessário. Quando uma *definição do processo* está trancada somente quem a trancou poderá modificá-la e salvar as modificações no servidor. Os outros usuários podem acessar a *definição do processo* enquanto a mesma permanecer trancada. Mas receberão um aviso de que o arquivo está trancado e não poderá ser modificado. Assim que julgar necessário o usuário poderá destrancar uma *definição do processo*. Após ser destrancada, os demais usuários poderão acessar e alterar a *definição do processo*.

O AW também é um cliente de um servidor Annotea (KOIVUNEN, 2001). Um servidor Annotea permite que anotações sejam feitas em uma *definição do processo*. As anotações não modificam o código fonte do arquivo onde foi feita a anotação e são vistas por todos clientes AW registrados no servidor Annotea. No servidor Annotea podem ser formados grupos onde cada grupo vê somente as suas anotações.

As anotações ficam armazenadas no servidor Annotea. Quando um usuário Annotea acessa uma *definição do processo* com comentários o servidor Annotea manda para o seu cliente (o AW) além da *definição do processo* as anotações. Quando recebe a *definição do processo* e as anotações o AW remonta a *definição do processo* inserindo nela as anotações. Assim o usuário já vê a *definição do processo* com as anotações. O usuário pode a qualquer momento escolher se quer ver ou esconder as anotações de uma *definição do processo*.

Para inserir uma anotação o usuário escolhe o ponto da *definição do processo* onde quer inserir um comentário e o envia para o servidor. Os comentários também podem ser inseridos dentro de outros comentários e em uma *definição do processo* trancada pelo WebDAV. Somente quem inseriu um comentário pode alterá-lo ou excluí-lo.

#### **5.4.14 Interoperabilidade**

##### **A. Interoperabilidade do modelo**

As entidades de *workflow* representadas no AW são compatíveis com as entidades definidas no meta-modelo da WfMC e no *process model*. Tal compatibilidade é demonstrada no capítulo 5.3 e pode ser sintetizada na tabela 5.1.

##### **B. Interoperabilidade da representação**

A representação interna do modelo de *workflow* do AW pode ser mapeada para representação XPD. Uma das funcionalidades do AW é um módulo capaz de exportar e importar *definições de processos* que estão no formato XPD.

O mapeamento da representação AW para a representação XPD é feito sem restrições ou perda de informações (tanto na importação quanto na exportação). No mapeamento não foi necessário o uso dos *extended attributes*. Tais características garantem a interoperabilidade do modelo AW com a representação XPD.

O AW também exporta a sua representação gráfica para as representações SVG e *Postscript*. Portanto, é compatível com visualizadores e demais *softwares* SVG e *Postscript*. Tal recurso garante ao AW uma interoperabilidade com dois dos principais padrões de visualização, descrição e impressão de documentos.

### C. Interoperabilidade tecnológica

O AW é compatível com as seguintes tecnologias:

- protocolo HTTP: o AW é um cliente HTTP que pode utilizar os recursos de um servidor de HTTP. Através do HTTP os usuários poderão acessar e editar uma definição de processos via Internet utilizando a *World WIDE Web* (WWW);
- protocolo WebDAV: o AW é um cliente WebDAV que pode utilizar os recursos de um servidor WebDAV. Através do WebDAV pode-se acessar e editar cooperativamente uma definição de processos via internet;
- *Web Services*: o XPD é compatível com a tecnologia de *Web Services* e pode descrever chamadas a serviços do tipo *Web Services*. O AW pode exportar um arquivo no formato XPD. O XPD pode fazer chamadas a *Web Services*. Portanto, o AW também é compatível com os *Web Services*;
- XML: O AW (i) é um *software* capaz de manipular a estrutura de qualquer documento que segue o padrão XML (característica herdada do Amaya); (ii) é capaz de gerar visões através de um documento que segue o padrão XML; (iv) utiliza e interpreta vários tipos de documentos XML (SVG, XLink, RDF, Namespace, entre outros) e (v) que pode gerar documentos nos formatos (XML) XPD, SVG e AW. Tais características dão ao AW uma grande interoperabilidade com a tecnologia XML.

## 5.5 Modelo de interação

### 5.5.1 Dispositivos de hardware

Na interação do usuário com o AW, o mouse e o teclado são utilizados como dispositivos de hardware para a entrada de dados. Já como saída de dados são utilizados o monitor e a impressora. O usuário envia dados ao sistema através do mouse e do teclado e tem a percepção dos estados do sistema através do monitor. A impressora é utilizada apenas para colocar no papel a aparência gráfica de alguma visão do sistema.

### 5.5.2 A interface

A interface do sistema é formada pelos elementos descritos a seguir.

Na parte superior da interface existe um menu *Pull-Down* formado por vários itens e outros sub-menus que dão acesso a todas as funcionalidades do sistema.

Abaixo do menu *Pull-Down* há uma barra de tarefas formada por ícones que dão acesso as funcionalidades mais utilizadas no sistema. Esta barra é chamada de "barra de ícones".

Abaixo da barra de ícones há um formulário do tipo texto que contém o caminho do arquivo que está aberto na atual seção do AW.

Abaixo do formulário com o caminho do atual arquivo está uma grande área de trabalho onde é mostrada uma visão da definição de processos. Esta área de trabalho possui barras de rolagem verticais e horizontais.

Abaixo da área de trabalho fica uma barra de *status* que exibe mensagens diversas sobre o atual estado do sistema.

Os itens desta interface são mostrados na figura 5.12.

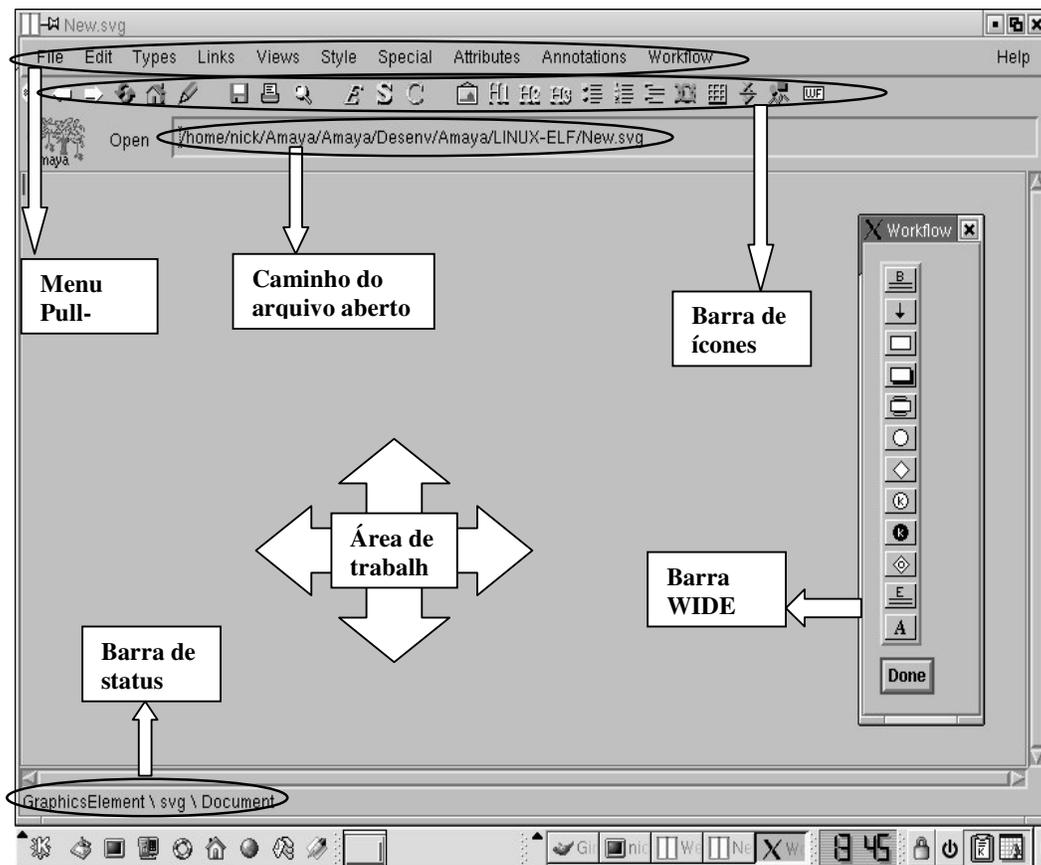


Figura 5.12: Interface do AW

### 5.5.3 Interações nos Menus

Para acionar uma funcionalidade do sistema o usuário deve escolher a funcionalidade desejada no menu *pull-down* ou na barra de ícones.

Quando um item do menu *pull-down* ou da barra de ícones é acionado, pode ocorrer uma das seguintes situações:

- o AW executa uma operação que não exige outras interações do usuário e pode retornar uma caixa de mensagem. Caso seja retornada uma caixa de mensagem ao usuário o mesmo deve clicar um botão para confirmar que leu a mensagem e desativar a caixa;
- o AW chama um ou mais formulários em seqüência para que o usuário insira mais dados necessários a execução de uma determinada funcionalidade. Após obter todos os dados necessários o AW executa a funcionalidade solicitada;
- é ativada e mostrada ao usuário uma barra de ferramenta. Neste caso, após ativar a barra de ferramentas o usuário poderá acionar as funções desta barra de ferramenta. Toda barra de ferramentas possui um botão que a desativa. Esse botão pode ser clicado sempre que o usuário quiser ocultar a barra de ferramentas.

O usuário pode querer que determinada ação seja executada sobre um determinado elemento ou sobre um determinado documento. Para que uma ação seja executada sobre um elemento, o mesmo deve estar selecionado em qualquer uma das visões do AW. Após ser selecionado, o usuário deve chamar a correspondente função dos menus. Para que uma ação seja executada sobre um documento o mesmo precisa estar ativo na atual seção do AW.

Terminada a execução da funcionalidade escolhida pelo usuário, pode-se escolher uma nova funcionalidade seguindo-se os mesmos passos.

#### 5.5.4 Interações nas Visões

Além de interagir com o AW através de menus e formulários pode-se realizar interações através das visões do sistema. A seguir serão mostrados detalhes das interações de cada visão do AW.

#### 5.5.5 Interações da visão gráfica

Na visão gráfica o usuário poderá interagir com o sistema através do estilo de interação "manipulação direta". Nesta visão um processo é representado por um grafo onde:

- os nodos são ícones ou figuras geométricas que representam elementos do *workflow* como, por exemplo, *atividades*, *joins* e *forks*;
- os arcos são setas que representam os *conectores*.

O grafo mostrado ao usuário utiliza as mesmas metáforas de representações e o mesmo comportamento do grafo proposto na representação gráfica do *process model*.

Através do mouse, o usuário pode definir o local onde um elemento será criado, pode-se selecionar, mover e redimensionar a aparência gráfica de um elemento.

Os símbolos da representação gráfica do *process model* podem ser adicionados na visão, através de uma barra de ferramenta chamada de "barra WIDE". A barra WIDE é acionada a partir da barra de ícones e contém vários botões. Cada botão da barra WIDE corresponde a um símbolo da representação gráfica do *process model*.

Para adicionar um símbolo basta o usuário clicar no correspondente botão da barra WIDE e em seguida clicar na área de trabalho. Os elementos são adicionados com tamanho fixo, porém se o usuário achar necessário ele pode mudar o tamanho dos elementos.

Para excluir um símbolo basta selecioná-lo com o mouse e em seguida pressionar a tecla "*delete*".

Para mover um elemento deve-se selecionar o elemento com o mouse, apertar a tecla "*control*", pressionar o botão esquerdo do mouse e arrastar o elemento até o local desejado.

Para redimensionar um elemento deve-se selecionar o elemento com o mouse, apertar a tecla "*control*", pressionar o botão direito do mouse e redimensionar o elemento até o tamanho desejado com o botão direito do mouse pressionado.

Cada elemento possui uma lista de atributos onde são armazenadas informações sobre o *workflow* que está sendo modelado. Muitos atributos são valorados automaticamente pelo sistema conforme é feita a distribuição gráfica dos elementos. É o caso dos atributos *ToID* e *FromID* do elemento *Connector* e os atributos gráficos de todos elementos.

Já outros atributos, devem ser valorados pelos usuários (caso haja necessidade os atributos definidos automaticamente pelo editor também podem ser editados pelos usuários). Para preencher e editar estes atributos o usuário precisa selecionar um elemento e em seguida deve clicar no item "*attributes*" do menu *pull-down*. Ao clicar neste item, um menu será mostrado ao usuário com todos os atributos do elemento selecionado. Para alterar algum atributo basta o usuário selecionar o atributo desejado e em seguida preencher ou alterar um formulário que ajudará o usuário a editar o valor deste atributo.

Quando o atributo *Name* de uma *Task*, *SuperTask* ou *MultiTask* é preenchido, o seu valor é mostrado na visão gráfica como um texto que fica sempre localizado dentro de um retângulo que representa o referido elemento.

Para navegar de um processo para outro foi utilizada a interação do tipo "hipertexto" conforme já foi explicado no capítulo 5.4.9 (que fala sobre a funcionalidade modularização do AW).

Outros recursos do AW que controlam, monitoram ou corrigem as interações do usuário já foram detalhadas no capítulo 5.4.4 que descreve as regras dos controles da definição de processo. Estas regras têm por objetivo manter o modelo e sua representação corretos e coerentes durante sua manipulação.

### 5.5.6 Interações na visão da estrutura

A visão da estrutura do AW é uma visão textual da representação interna do modelo que utiliza alguns recursos gráficos. O código do processo é mostrado de uma forma estruturada que destaca a hierarquia dos elementos através de símbolos gráficos, cores e disposição dos nomes dos elementos, dos nomes dos atributos e dos valores dos atributos.

Nesta visão não podem ser acrescentados novos elementos. Pode-se apenas selecionar, alterar e excluir elementos.

Para excluir um elemento basta selecioná-lo e em seguida utilizar a tecla "*delete*". Ao selecionar um elemento são selecionados automaticamente todos seus filhos. Não é possível selecionar só uma parte de um elemento. Os elementos são sempre selecionados inteiramente. Assim, ao excluir um elemento seus filhos também são excluídos. Nenhum elemento é parcialmente excluído.

Na visão da estrutura também se pode usar o menu *attributes* para inserir, excluir e alterar atributos de um elemento.

Sem utilizar o menu *attributes* não é possível acrescentar nem excluir atributos. Também não é possível alterar o nome do elemento. A única operação possível utilizando-se apenas a visão estruturada é alterar os valores dos atributos.

Para alterar o valor de um atributo basta colocar o cursor do mouse sobre o valor de um atributo e altera-lo através de inserções e exclusões de caracteres.

### 5.5.7 Interações na visão AW

A AW é uma visão textual que mostra o código fonte da definição de processo. O código é mostrado exatamente do modo como ele é armazenado pelo sistema.

Nesta visão o usuário tem liberdade de editar qualquer texto em qualquer lugar do código. Por editar texto deve-se entender inserir, excluir e alterar caracteres. O código nesta visão é tratado como um simples texto.

Caso a alteração feita pelo usuário deixe o código fora da sintaxe do modelo do AW, o sistema informa a ocorrência de erro. Este procedimento já foi detalhado no capítulo 5.4.4 que descreve as regras dos controles da definição de processo.

### 5.5.8 Interações nas barras de ferramentas

O AW possui barras de ferramentas que permitem a adição de elementos nos documentos do AW. As barras de ferramentas mais importantes do AW são:

- a barra de ícones. Esta barra possui itens de menu que apontam para as principais funções do AW e que chamam as demais barras de ferramentas;
- a barra WIDE. Esta barra possui itens de menu que permitem a inserção de elementos da representação AW em uma definição de processo.

### 5.5.9 Inserção de links

Para adicionar um *link* em um documento AW é necessário:

- seleccionar o elemento que será a origem do link a ser adicionado (um elemento *Begin*, *SuperTask* ou *MultiTask*);
- clicar no ícone da barra de ícones que aciona a função que insere ou altera *links*;
- preencher um formulário que indicará o caminho do destino do *link* a ser inserido.

### 5.5.10 Interações nos formulários do item de menu Attributes

Um item do menu *pull-down* que merece especial atenção é o item *attributes*. Como já foi dito o item *attributes* é utilizado para valorar os atributos dos elementos do AW. O *attributes* sempre mostra uma lista com os atributos do elemento que está seleccionado na área de trabalho. Clicando-se em um atributo desta lista é mostrado um formulário que auxiliará o usuário a preencher o valor deste atributo.

Os valores destes atributos são bastante variados e em alguns casos podem ser bastante complexos. Portanto a qualidade destes formulários é essencial para um preenchimento eficaz dos valores de atributos de um *workflow*.

Os tipos de valores de atributos do AW e suas interações são mostrados a seguir:

- A. **caracteres alfanuméricos:** podem ser preenchidos com quaisquer caracteres. Sem muita rigidez devem seguir a sintaxe de uma linguagem natural que seja compreendida por seus usuários. No preenchimento deste tipo de atributo a interação é feita através de um formulário que permite que o usuário informe em um único campo um grupo de caracteres alfanuméricos.

Exemplos deste tipo de atributo são: o *Name*, o *Description* e o *Error* da representação AW;

- B. **números inteiros:** podem ser preenchidos com números inteiros. No preenchimento deste tipo de atributo a interação é feita através de um formulário que permite que o usuário informe em um único campo um número inteiro.

Exemplos deste tipo de atributo são os atributos *Njoin* e *Nfork* da representação AW;

- C. **ID:** podem ser preenchidos com caracteres alfanuméricos. Neste caso não podem ser utilizados alguns caracteres como o "espaço" e caracteres de pontuação. Outra restrição destes atributos é que o seu valor deve ser único em um documento. No preenchimento deste tipo de atributo a interação é feita através de um formulário que permite que o usuário informe em um único campo um grupo de caracteres alfanuméricos.

Este é o caso do atributo ID que está presente em todos os elementos da representação AW;

- D. **referência a Ids:** podem ser preenchidos com um id de um elemento existente no atual documento. No preenchimento deste tipo de atributo a interação é feita através de um formulário que mostra ao usuário uma lista com todos os possíveis *ids* que podem ser selecionados. Após observar a lista o usuário deverá escolher um elemento desta lista. A lista de *ids* é formada dinamicamente no momento em que é solicitado o preenchimento do valor de um atributo deste tipo. O AW percorre todo o documento procurando os *ids* cadastrados até o momento. Após estes *ids* são mostrados ao usuário.

Este é o caso dos atributos *FromID* e *ToID* do elemento *Connector* da representação AW;

- E. **escolha:** podem ser preenchidos por um valor dentre um grupo de valores pré-estabelecidos. No preenchimento deste tipo de atributo a interação é feita através de um formulário que permite que o usuário escolha um valor de uma lista de valores pré-definidos.

Este é o caso do atributo *AcessLevel* do elemento *begin* da representação AW;

- F. **URL:** podem ser preenchidos por uma URL que aponta para o destino de um link. No preenchimento deste tipo de atributo a interação é feita através de um formulário que permite que o usuário informe uma URL em um único campo. O usuário também pode utilizar o recurso de apertar um botão e navegar pela estrutura de arquivos e diretórios do computador local. O estilo de interação da navegação por esta estrutura é a "manipulação direta".

Este é o caso do atributo *href* do elemento *Supertask* da representação AW;

- G. **valor complexo fixo:** devem ser preenchidos por valores que seguem uma sintaxe mais complexa. No preenchimento deste tipo de atributo a interação é feita através

de um formulário que permite que o usuário informe e escolha vários valores em vários campos utilizando vários recursos de interface.

Este é o caso dos atributos *PackageHeader* e *TypeDeclarations* do elemento *begin* da representação AW;

- H. **valor complexo dinâmico:** devem ser preenchidos por valores que seguem uma sintaxe mais complexa. No preenchimento deste tipo de atributo a interação é feita através de um formulário que permite que o usuário informe e escolha vários valores em vários campos utilizando vários recursos de interface. A diferença dos valores complexos fixos para os valores complexos dinâmicos é que as interfaces para os valores fixos são interfaces fixas e pré-definidas. Já as interfaces dinâmicas são definidas dinamicamente. A interface é formada no momento em que é solicitado o preenchimento do valor de um atributo deste tipo. O AW percorre todo o documento procurando todas as informações necessárias para formar uma interface adequada a este atributo. São formadas listas de *ids*, criados campos para serem informados valores, criadas listas de escolha, etc. Após ser formada, a interface é mostrada ao usuário que irá escolher e informar os valores necessários.

Este é o caso dos atributos *Application* e *Participant* do elemento *begin* da representação AW.

**OBS:** Maiores detalhes sobre os valores de cada atributo da representação AW são mostrados no capítulo 6.

### 5.5.11 Interações nos arquivos de configuração

O AW possui alguns arquivos de configuração que podem ser utilizados pelos usuários conforme mostra o capítulo 5.4.11 que descreve a adaptabilidade do AW. São arquivos textos cujo conteúdo pode ser editado pelos usuários. A edição destes arquivos não faz parte da interface do AW. Para editá-los o usuário precisará utilizar um editor de texto externo. O resultado das mudanças feitas nos arquivos de configuração pode ser percebido pelo usuário na próxima vez que o AW for aberto.

Além de arquivos texto existem arquivos binários com figuras que representam símbolos de *workflow*. Estes arquivos podem ser substituídos pelos usuários. Novamente a edição e troca dos arquivos não fazem parte da interface do AW, para editar estes arquivos o usuário terá que utilizar editores gráficos externos. Assim como nos arquivos texto as mudanças feitas no arquivo binário serão percebidas na próxima vez que o AW for aberto.

### 5.5.12 Padrão de interação

O padrão de interação do AW é o padrão Open Motif (Open Group, 2004) da Open System Foundation (OSF).

## 5.6 Tabela de descrição do modelo conceitual AW

A Tabela 5.2 descreve o modelo conceitual da ferramenta AW conforme a *tabela de descrição de modelos conceituais de ferramentas de definição* descrita no capítulo 4.

Além das informações sobre o modelo conceitual AW, são mostradas na tabela 5.2 informações sobre o protótipo Amaya *Workflow*. As informações sobre o protótipo devem ser interpretadas da seguinte forma:

- os itens marcados com "\*" são os itens do modelo conceitual do AW que ainda não foram implementados no protótipo Amaya *Workflow*;
- os itens marcados com "#" são as funcionalidades e conceitos implementados no Amaya que foram reaproveitados no protótipo AW;
- os demais itens são conceitos e funcionalidades que foram criados ou adaptados ao protótipo AW durante a presente dissertação.

Tabela 5.2: Tabela de descrição do modelo conceitual AW

| <b>Itens do modelo</b>              | <b>Modelo conceitual do Amaya Workflow</b>   |
|-------------------------------------|--|
| <b>Conceitos</b>                    |  |
| Modelo de <i>workflow</i>           | Atividade, roteamento, participante, aplicativos, dados relevantes, aspectos temporais, prioridade, versão, extensão, processo, pacote, repositório, exceções, informações gráficas. |
| Representação do modelo             | 1 Gráfica(WIDE), 1 XML (AW), 1 Texto+gráficos (Estruturada)  |
| Definição de processo               | Interpretada (AW); exportada (XPDL); importada (XPDL)*   |
| Repositório                         | Um diretório é interpretado como um repositório  |
| <b>Funcionalidades</b>              |  |
| Visões                              | 3 Sincronizadas.(WIDE, AW, Estruturada)  |
| Funcionalidades de edição #         | Copiar, recortar, colar, desfazer ações, refazer ações, impressão, zoom, barra de rolagem, localizar palavras, barra de status, algoritmos de re-arranjo*.                           |
| Controles da definição de processo# | Manipular os elementos; manipular os atributos.  |
| Regras dos Controles                | Verificar e reagir a erros; prevenção de erros; estabelecer relacionamentos; manter a coerência.   |
| Reedição e reaproveitamento #       | Abrir, fechar, salvar e recarregar.  |
| Exportação                          | (XPDL , SVG , PostScript #)  |
| Importação                          | (XPDL*)  |
| Gerenciamento do repositório        | Manipulação de documentos, execução de procedimentos(Verificar erros, gerar XPDL)*, referenciar pacotes externos   |
| Modularização                       | Navegar pela hierarquia, editar hierarquia #   |
| Verificação de erros                | Verifica sintaxe, verifica semântica, relatório de erros, destaca erros.   |
| Simulação                           | Modelagem da simulação e interoperabilidade com os padrões da WfMC.  |
| Recursos para cooperação #          | Comunicação (Annotea*), percepção (WebDAV*), coordenação (WebDAV*), compartilhamento (http, WebDAV*, Annotea*)   |
| Adaptabilidade #                    | Adaptação dos símbolos*, configuração das funcionalidades, adaptação dos itens, adaptação da interface.  |
| Interoperabilidade                  | Modelo: Compatível com o modelo da WfMC<br>Representação: Mapeamento para o XPDL sem o uso do ExtendedAttribute<br>Tecnológica: HTTP#, WebDAV*, Web Services, XML#                   |
| <b>Interação</b>                    |  |
| Dispositivos de hardware #          | Entrada: Teclado, mouse.<br>Saída: Monitor, impressora   |
| Estilos de Interação utilizados     | Menus, formulários, hiperlink, manipulação direta  |
| Padrão de interação #               | Open Motif   |

Uma análise comparativa entre o modelo conceitual do AW e os modelos conceituais de sistemas existentes é feita no capítulo 7. Mais detalhes sobre a representação AW são mostrados no capítulo 6.

## 5.7 Itens do modelo AW ainda não implementados no protótipo AW

Conforme mostra a tabela 5.2, os principais itens previstos no modelo AW e ainda não implementados no protótipo AW são: a importação de arquivos XPDL, a implementação de algoritmos de distribuição automática de elementos, a adaptação de

símbolos, a implementação de um servidor WebDAV e a implementação de um servidor Annotea (para que funcionem em conjunto com o AW).

Também não estão implementadas algumas funcionalidades que facilitam o preenchimento dos valores de alguns atributos. Não estão desenvolvidas as interfaces que preenchem valores do tipo: escolhas, valores complexos fixos e valores complexos dinâmicos. As funcionalidades de verificação de erros ainda não verificam a sintaxe dos valores destes três tipos de valores. Atualmente os valores do tipo: escolhas, valores complexos fixos e valores complexos dinâmicos são preenchidos com interfaces semelhantes as interfaces de preenchimento de valores do tipo alfanumérico. Uma descrição detalhada dos tipos de valores e suas interfaces são apresentadas no capítulo 5.5.2.9. Este capítulo fala sobre as interações nos formulários do item de menu *Attributes*.

Por fim também é necessário tornar os nomes dos atributos e elementos do protótipo iguais aos nomes previstos no modelo. No momento os nomes dos elementos e atributos do protótipo são precedidos de W (WTask, Wconnector, etc).

## 6 A REPRESENTAÇÃO AW

Este capítulo descreve uma das contribuições desta dissertação: a representação AW. A representação AW foi utilizada tanto no protótipo AW quanto no modelo conceitual AW. Portanto é importante para a compreensão destes dois itens.

### 6.1 Os elementos

A representação AW possui uma sintaxe do padrão XML. Portanto é formada por uma hierarquia de atributos e elementos. O elemento raiz desta representação é o *workflow*. Todos os demais elementos do modelo são filhos do elemento *workflow* e irmãos entre si. Cada filho do elemento *workflow* representa um símbolo da representação gráfica do *process model* (CASATI et al., 1995). As relações entre os elementos do modelo AW e os símbolos definidos na representação gráfica do *process model* são mostradas abaixo:

- elemento *Begin* representa o símbolo *início* do *process model*;
- elemento *End* representa o símbolo *fim* do *process model*;
- elemento *Task* representa o símbolo *tarefa* do *process model*;
- elemento *Connector* representa o símbolo *conector* do *process model*;
- elemento *MultiTask* representa o símbolo *multi tarefa* do *process model*;
- elemento *SuperTask* representa o símbolo *super tarefa* do *process model*;
- elemento *ForkJoinTotal* representa os símbolos *fork* e *join total* do *process model*;
- elemento *JoinPartial* representa os símbolos *join parcial* do *process model*;
- elemento *JoinInteractive* representa o símbolo *join interativo* do *process model*;
- elemento *ForkConditional* representa o símbolo *fork condicional* do *process model*;
- elemento *EForkConditional* representa o símbolo *fork condicional com exclusão mútua* do *process model*.

Todos os elementos acima são irmãos e podem ser colocados dentro do elemento *workflow* em qualquer ordem. A seqüência ou paralelismo das atividades de um *workflow* é determinado pelos atributos dos elementos, não importando a ordem destes elementos dentro da árvore hierárquica.

Não foi incluído na representação AW um elemento que representasse o símbolo *fork parcial* do *process model*. Este símbolo ou um correspondente com funções semelhantes não foi encontrado em outros modelos. Até mesmo modelos que foram

modelos derivados do *process model* não adotaram este símbolo por não considera-lo relevante (BARESI et al., 1999).

A seguir será mostrado em detalhes cada elemento da representação AW bem como os atributos e seus possíveis valores. Ao fim de cada elemento serão dadas referências sobre onde atributos semelhantes foram utilizados na representação XPDL da WfMC e no *process model* do projeto WIDE. Caso o atributo tenha sido proposto por esta pesquisa, é dada uma justificativa sobre a inserção deste atributo no modelo AW.

### 6.1.1 Elemento *Workflow*

O elemento raiz da representação AW é o elemento *Workflow*. Dentro do elemento *Workflow* estão todos os demais elementos modelados. Como este elemento contém todos os demais ele poderá representar as entidades *processo* e *pacote* do modelo.

Se o documento que estiver sendo modelado for o *documento raiz*, o elemento *Workflow* representa a entidade *pacote*. Neste caso o elemento *Workflow* conterá elementos que representarão as informações sobre o pacote.

Se o documento que estiver sendo modelado não for o *documento raiz* o elemento *Workflow* representa a entidade *processo*. Neste caso o elemento *Workflow* conterá elementos que representarão informações sobre um processo.

O elemento *Workflow* não possui atributos.

### 6.1.2 Elementos e atributos gráficos

Todos os elementos que são filhos do elemento *workflow* possuem uma representação gráfica. A aparência gráfica destes elementos é definida na representação gráfica do *process model* (CASATI et al., 1995). Tal aparência é descrita na representação AW por atributos da linguagem SVG que são inseridos na representação AW através do padrão Namespace.

A aparência de cada elemento do AW pode ser descrita através de um elemento SVG e seus atributos. Os atributos do elemento SVG correspondente podem ser colocados na representação AW como atributos dos elementos de *workflow* utilizando-se exatamente a mesma sintaxe.

Também é adicionado ao elemento de *workflow* um atributo do SVG chamado *type*. Assim como os demais atributos, o *type* é adicionado através do Namespace. O valor deste atributo é o nome do elemento SVG que descreve a aparência do elemento de *workflow* do AW. Assim pode-se adicionar em um elemento AW o nome do elemento do SVG que o descreve graficamente. O atributo *type* foi proposto nesta pesquisa.

O elemento *Connector* do AW tem a aparência de um conjunto de pontos interligados por segmentos de reta que formam uma seta.

Tal aparência pode ser descrita através de atributos do elemento *line* do SVG. Por isso o elemento *Connector* do AW possui os mesmos atributos gráficos que o elemento *line* do SVG.

Um atributo importante do *line* é o atributo *points*. O atributo *points* determina as coordenadas dos pontos por onde a seta que representa o *Connector* deverá passar.

Os demais elementos têm a aparência de uma imagem com moldura retangular. A imagem é do tipo “*raster*” (nos formatos jpg, gif ou png) e o seu conteúdo mostra um símbolo de *workflow* proposto no *process model*.

Tal aparência pode ser descrita em SVG através de atributos do elemento *image* do SVG. Portanto os elementos *Begin*, *End*, *JoinPartial*, *JoinPartial*, *ForkJoinTotal*, *JoinInterativo*, *ForkConditional*, *EForkConditional*, *MultiTask*, *SuperTask* e *Task* do modelo AW possuem os mesmos atributos gráficos do elemento *image* do SVG.

Os atributos do elemento *image* utilizados nos elementos AW são os seguintes:

Tabela 6.1: Atributos do elemento *image* utilizados nos elementos AW

|              |  |
|--------------|--|
| X            | Coordenada X do canto superior esquerdo do retângulo que contém a figura.        |
| Y            | Coordenada Y do canto superior esquerdo do retângulo que contém a figura.        |
| Width        | Largura do retângulo.  |
| Height       | Altura do retângulo.   |
| Fill         | Cor de fundo do retângulo.   |
| Stroke       | Cor da borda do retângulo.   |
| Stroke-width | Largura da borda do retângulo.   |
| Href         | Caminho do arquivo que contém a imagem que representa um símbolo do modelo WIDE. |

#### Referências:

Os atributos gráficos dos elementos do AW foram definidos na especificação da linguagem SVG.

### 6.1.3 Atributos comuns a vários elementos

Há na representação AW atributos que são comuns a todos os elementos filhos do elemento *Workflow*. São eles:

- A. **atributo Id:** visa identificar de forma única o elemento dentro de um *workflow*. Através do atributo *Id* os elementos do *workflow* podem se referenciar. O *Id* é um atributo obrigatório em todos elementos. O valor do atributo *Id* pode ser preenchido por qualquer seqüência de caracteres alfanuméricos;
- B. **atributo Name:** visa identificar um elemento dentro de um *workflow*. O *Id* é destinado a identificar um elemento para sistemas automatizados tendo como principal característica a unicidade da identificação. Já o *Name* é destinado a identificar um elemento para pessoas. Tem como principal característica a clareza da identificação (os identificadores *Name* não precisam ser únicos). O valor do atributo *Name* pode ser preenchido por qualquer seqüência de caracteres alfanuméricos;
- C. **atributo Description:** este atributo descreve um elemento fornecendo informações ou observações sobre o seu elemento. O valor do *Description* é preenchido por uma seqüência de caracteres alfanuméricos que formam um texto que deve ser compreensível por analistas de *workflow* e analistas do domínio;
- D. **atributo ExtendedAttributes:** este atributo representa a entidade *extensão* do modelo de *workflow*. Inicialmente este atributo não tem uma função específica, porém o analista de *workflow* poderá usar e valorar o atributo *ExtendedAttributes*

conforme a sua necessidade caso queira colocar alguma informação na *definição do processo* que não foi prevista no modelo. O valor do atributo *ExtendedAttributes* deve seguir uma parte da sintaxe do XPDL. Os valores que podem ser colocados no atributo *ExtendedAttributes* do AW são os mesmos que podem ser colocados dentro do elemento *ExtendedAttributes* do XPDL. Abaixo um exemplo de valor que pode ser colocado no atributo *ExtendedAttributes*.

```
<ExtendedAttribute Name="CoordX" value="120">
```

- E. **atributo Error:** o valor deste atributo é preenchido pelo próprio AW e poderá conter mensagens de erros de modelagem. Maiores detalhes sobre o uso do atributo *error* podem ser vistos no capítulo 5.4.10 que fala sobre a funcionalidade “verificação de erros”.

### Referências

Atributos semelhantes aos atributos *Id*, *Name*, *Description* da representação AW foram primeiramente propostos na representação XPDL da WfMC e no modelo *process model*.

Um elemento semelhante ao atributo *ExtendedAttributes* da representação AW foi primeiramente proposto na representação XPDL da WfMC.

O atributo *Error* foi proposto por esta pesquisa e adicionada na representação AW para possibilitar a implementação da funcionalidade “verificação de erros”.

Os atributos gráficos utilizados na representação AW foram primeiramente propostos na definição da sintaxe do SVG (W3C, 2001).

**OBS:** Para uma maior concisão do texto, a descrição dos próximos elementos não irá citar os atributos *Id*, *Name*, *Description*, *ExtendedAttributes*, *Error* e gráficos. Porém todos estes elementos possuirão estes atributos.

#### 6.1.4 Elemento Begin

O elemento **Begin** representa uma entidade de *roteamento* que indica o início da execução de um processo de *workflow*. Este elemento na representação AW tem funções semelhantes ao símbolo *início* da representação gráfica do *process model*. Em um *workflow* modelado corretamente deve sempre existir um e somente um elemento *Begin*.

Uma das diferenças entre o uso do elemento *Begin* no AW e do símbolo *início* na representação gráfica do *process model* é o fato do elemento *Begin* conter tanto atributos com informações sobre o próprio elemento *Begin* quanto atributos com informações sobre as entidades *pacote* ou *processo*. Se o arquivo que está sendo modelado for o arquivo raiz, as informações dos atributos do *Begin* estarão relacionadas com o pacote. Agora, se o arquivo que está sendo modelado não for o arquivo raiz as informações dos atributos do *Begin* estarão relacionadas com um processo.

O elemento que representa as entidades *processo* e *pacote* é o elemento *workflow*. Entretanto, os atributos com informações sobre o processo ou pacote estão dentro do elemento *Begin*. Foi escolhida esta configuração para que os usuários possam manipular as informações sobre o processo/pacote através de um elemento que possua uma

representação gráfica no *process model*. A representação gráfica do *process model* não prevê uma representação gráfica para as entidades *processo* e *pacote*.

Dentre todos os elementos com representação gráfica no *process model*, o *Begin* foi escolhido como o elemento que conterá informações sobre o pacote/processo. Foi feita esta escolha devido ao fato do *Begin* possuir a propriedade de sempre existir em um *workflow* válido e ser único neste *workflow*. Os atributos do *Begin* são:

- A. **atributo href**: este atributo contém uma referência ao documento que contém o processo-pai do atual processo. O fato deste atributo estar ou não preenchido determina se o corrente documento é um documento raiz ou um documento comum.

Se o atributo *href* estiver preenchido significa que o corrente processo possui um processo pai na hierarquia de processos do *workflow*. Portanto este documento é um documento comum.

Se o atributo *href* não estiver preenchido significa que o corrente processo não possui um processo pai na hierarquia de processos do *workflow*. Portanto este documento é um documento raiz.

A qualquer momento o usuário pode transformar um documento raiz em um documento comum e vice-versa. Em qualquer uma das duas transformações não são perdidas informações nem é necessária outra alteração além de se atribuir ou não um valor ao atributo *href*.

O valor do atributo *href* é preenchido com uma URL válida que aponta para o documento que contém o processo pai do corrente processo;

- B. **atributo TypeDeclarations**: este atributo contém uma lista que descreve os tipos de dados que poderão ser usados no pacote/processo que está sendo modelado.

O valor do atributo *TypeDeclarations* deve seguir uma parte da sintaxe do XPDL. Os valores que podem ser colocados no atributo *TypeDeclarations* são os mesmos que podem ser colocados dentro do elemento *TypeDeclarations* do XPDL. Abaixo um exemplo de valor que pode ser colocado no atributo *TypeDeclarations*.

```
<TypeDeclaration Id="Credito" Name="Credito">
    <ExternalReference location="C:\inf\credito.wsdl"xref="Credito"/>
</TypeDeclaration>
<TypeDeclaration Id="CardTipo" Name="CardTipo">
    <ExternalReference location="C:\inf\order.xsd"
        xref="cardTipo" namespace="orderschema/Order"/>
</TypeDeclaration>
```

- C. **atributo Participants**: este atributo representa a entidade "participantes" do modelo de *workflow*. O atributo *Participants* contém uma lista que descreve os participantes que poderão ser referenciados no pacote/processo que está sendo modelado. A descrição do participante é a descrição de um papel que deverá ser assumido por um ator/pessoa no momento da execução do *workflow*.

O valor do atributo *Participants* deve seguir uma parte da sintaxe do XPDL. Os valores que podem ser colocados no atributo *Participants* são os mesmos que podem ser colocados dentro do elemento *Participants* do XPDL. Abaixo um exemplo de valor que pode ser colocado no atributo *Participants*.

```

<Participant Id="ConexaoBD">
  <ParticipantType Type="SYSTEM"/>
  <Description>Acessa a Base de dados</Description>
</Participant>

```

- D. **atributo *Applications***: este atributo representa a entidade *aplicativos* do modelo de *workflow*. O atributo *Applications* contém uma lista que descreve os aplicativos que poderão ser referenciados no pacote/processo que está sendo modelado. Na descrição do aplicativo são informados também os tipos de dados que o aplicativo precisa receber de entrada para ser acionado e o tipo de dados que o aplicativo retornará para o *workflow* que chamou o aplicativo. Na declaração dos tipos de dado dos parâmetros de entrada e saída de uma aplicação podem ser referenciados os tipos de dados declarados no atributo *TypeDeclarations*.

O valor do atributo *Applications* deve seguir uma parte da sintaxe do XPDL. Os valores que podem ser colocados no atributo *Applications* são os mesmos que podem ser colocados dentro do elemento *Applications* do XPDL. Abaixo um exemplo de valor que pode ser colocado no atributo *Applications*.

```

<Application Id="transformData">
  <FormalParameters>
    <FormalParameter Id="orderIn" Index="1" Mode="IN">
      <DataType>
        <BasicType Type="STRING"/>
      </DataType>
    </FormalParameter>
    <FormalParameter Id="order" Index="2" Mode="OUT">
      <DataType>
        <DeclaredType Id="Order"/>
      </DataType>
    </FormalParameter>
  </FormalParameters>
</Application>

```

- E. **atributo *DataFields***: este atributo representa a entidade *dados relevantes ao processo* do modelo de *workflow*. O atributo *DataFields* contém uma lista que descreve as variáveis que poderão ser utilizadas no pacote/processo que está sendo modelado. Na declaração dos tipos de dados das variáveis podem ser referenciados os tipos de dados declarados no atributo *TypeDeclarations*.

O valor do atributo *DataFields* deve seguir uma parte da sintaxe do XPDL. Os valores que podem ser colocados no atributo *DataFields* são os mesmos que podem ser colocados dentro do elemento *DataFields* do XPDL. Abaixo um exemplo de valor que pode ser colocado no atributo *DataFields*.

```

<DataField Id="orderNumber" IsArray="FALSE">
  <DataType>
    <BasicType Type="INTEGER"/>
  </DataType>
  <InitialValue>0</InitialValue>
  <Length>0</Length>
</DataField>
<DataField Id="status" IsArray="FALSE">
  <DataType>
    <DeclaredType Id="OrderStatus"/>
  </DataType>
</DataField>

```

- F. **atributo FormalParameters:** este atributo contém uma lista que descreve os parâmetros que deverão ser passados para o processo que está sendo modelado quando o processo for chamado por um processo pai. Na declaração dos tipos de dados dos parâmetros podem ser referenciados os tipos de dados declarados no atributo *TypeDeclarations*.

O valor do atributo *FormalParameters* deve seguir uma parte da sintaxe do XPDL. Os valores que podem ser colocados no atributo *FormalParameters* são os mesmos que podem ser colocados dentro do elemento *FormalParameters* do XPDL. Abaixo um exemplo de valor que pode ser colocado no atributo *FormalParameters*.

```

<FormalParameter Id="orderString" Index="1" Mode="IN">
  <DataType>
    <BasicType Type="STRING"/>
  </DataType>
</FormalParameter>
<FormalParameter Id="returnMessage" Index="2" Mode="OUT">
  <DataType>
    <BasicType Type="STRING"/>
  </DataType>
</FormalParameter>

```

- G. **atributo AccessLevel:** este atributo determina o nível de acesso do processo. Um processo pode ser público ou privado. Se for público o processo pode ser acessado por sistemas e aplicações externas. Se for privado o processo só pode ser acessado por processos do próprio pacote. O atributo *AccessLevel* pode ser preenchido pelos valores "PUBLIC" ou "PRIVATE" que representam respectivamente os modos de acesso público e privado.

- H. **atributo PackageHeader**: este atributo contém informações sobre o pacote que não foram informadas nos demais atributos do *Begin*. São informações gerais sobre o pacote como, por exemplo, versão do pacote, data de criação do pacote, empresa ou pessoa responsável pelo pacote, lista dos pacotes externos que poderão ser referenciados pelo pacote corrente, etc.

O valor do atributo *PackageHeader* deve seguir uma parte da sintaxe do XPDL. Os valores que podem ser colocados no atributo *PackageHeader* são os valores dos elementos *ConformanceClass*, *ExternalPackages*, *PackageHeader*, *RedefinableHeader* e *Script* do XPDL. Abaixo um exemplo de valor que pode ser colocado no atributo *PackageHeader*.

```
<PackageHeader>
  <XPDLVersion>0.09</XPDLVersion>
  <Vendor>UFRGS, Inc</Vendor>
  <Created>6/18/2003 7:27:18 PM</Created>
</PackageHeader>
<ConformanceClass GraphConformance="NON_BLOCKED"/>
<Script Type="text/javascript"/>
```

- I. **atributo ProcessHeader**: este atributo contém informações sobre o processo que não foram informadas nos demais atributos do *Begin*. São informações gerais sobre o processo como, por exemplo, versão do processo, data de criação do processo, responsável pelo processo, etc.

O valor do atributo *ProcessHeader* deve seguir uma parte da sintaxe do XPDL. Os valores que podem ser colocados no atributo *ProcessHeader* são os valores dos elementos *ProcessHeader* e *RedefinableHeader* do XPDL. Abaixo um exemplo de valor que pode ser colocado no atributo *ProcessHeader*.

```
<Created>6/18/2003 7:25:17 PM</Created>
<Description>Analisa pedidos de compra </Description>
```

## Referências

Atributos ou elementos semelhantes aos atributos *TypeDeclarations*, *Participants*, *Applications*, *DataFields*, *AccessLevel*, *PackageHeader* e *ProcessHeader* da representação AW foram primeiramente propostos na representação XPDL da WfMC.

Um elemento semelhante ao *FormalParameters* da representação AW foi primeiramente proposto na representação XPDL da WfMC e no *process model*.

Um atributo semelhante ao atributo *href* foi proposto primeiramente pela especificação XLink (DEROSE; MALER; ORCHARD, 2001) da W3C. A especificação XLink define atributos que descrevem *links* entre elementos ou documentos.

### 6.1.5 Elemento Connector

O elemento **Connector** representa uma entidade de *roteamento* que conecta os demais elementos indicando assim o fluxo ou a seqüência de elementos em um *workflow*. Este elemento na representação AW tem características, comportamentos e funções semelhantes ao símbolo "conector" do *process model*.

Os atributos do elemento *Connector* são:

- A. **atributo FromID:** este atributo identifica o elemento que antecede o elemento *Connector* no fluxo do *workflow*. O valor do atributo *FromID* é preenchido com o *Id* do elemento que antecede o elemento *Connector*.
- B. **atributo ToID:** este atributo identifica o elemento que sucede o elemento *Connector* no fluxo do *workflow*. O valor do atributo *FromID* é preenchido com o *Id* do elemento que sucede o *connector*.
- C. **atributo Conditions:** este atributo contém informações que indicarão se o fluxo do *workflow* poderá passar do elemento indicado no *FromID* para o elemento indicado no *ToID*. O *Conditions* deverá conter uma expressão que deverá retornar o valor TRUE ou FALSE. Se a expressão retornar TRUE, ao ser analisada em tempo de execução, o fluxo do *workflow* poderá passar pelo atual *Connector*. Caso o retorno da análise da expressão seja FALSE, o fluxo do *workflow* não poderá passar pelo atual *Connector*.

O valor do atributo *Condition* deve seguir uma parte da sintaxe do XPDL. Os valores que podem ser colocados no atributo *Condition* são os valores do elemento *Condition* do XPDL. Abaixo um exemplo de valor que pode ser colocado no atributo *Condition*.

```
<Condition>OrdemTipo == "Credito"</Condition>
```

### Referências

Atributos e elementos semelhantes aos atributos *FromID*, *ToID* e *Conditions* do AW foram primeiramente propostos na representação XPDL da WfMC e no *process model*.

### 6.1.6 Elemento End

O elemento **End** representa uma entidade de *roteamento* que indica o fim ou a saída de um processo de *workflow*. Em todo *workflow* modelado deve existir pelo menos um elemento *End*. Também é permitida a existência de vários *End* em um mesmo *workflow*.

O elemento *End* possui somente os atributos comuns a todos elementos do *workflow*.

### 6.1.7 Elemento Task

O elemento *Task* representa a entidade *atividade* do modelo de *workflow*. É o elemento que representa as unidades de trabalho ou as ações do processo de *workflow*. O *Task* no modelo AW tem características semelhantes ao símbolo "tarefa" do *process model*.

Na sua modelagem, o elemento *Task* poderá referenciar aplicativos, participantes e variáveis declaradas no "atual processo" ou no "atual pacote". "Atual processo" é o processo que contém o elemento *Task* que está sendo modelado. "Atual pacote" é o pacote declarado no arquivo raiz do "atual processo".

Os atributos do elemento *Task* são:

- A. **atributo Participant:** este atributo informa qual participante será responsável pela tarefa. O valor deste atributo é um *Id* de um participante declarado no atributo *Participant* do elemento *Begin*.
- B. **atributo Action:** este atributo invoca um aplicativo declarado no atributo *Application* do elemento *Begin*. Ao invocar um aplicativo são passados os parâmetros de entrada e saída que o aplicativo necessita para executar alguma ação.

O valor do atributo *Action* deve seguir uma parte da sintaxe do XPDL. Os valores que podem ser colocados no atributo *Action* são os valores do atributo *tool* do XPDL. Abaixo um exemplo de valor que pode ser colocado no atributo *Action*.

```
<Tool Id="checkData" Type="APPLICATION">
  <ActualParameters>
    <ActualParameter>Informacao</ActualParameter>
    <ActualParameter>Estado</ActualParameter>
  </ActualParameters>
</Tool>
```

- C. **atributo ExceptionAction e ExceptionReaction:** estes atributos representam a entidade exceção do modelo de *workflow*. Quando é detectada uma exceção o *workflow* deve disparar uma ação que irá tratar esta exceção. O atributo *ExceptionAction* descreve sob quais condições uma exceção ocorre. Já o atributo *ExceptionReaction* contém as ações que deverão ser executadas para tratar a exceção.

O valor do atributo *ExceptionAction* deve seguir uma parte da sintaxe do XPDL. Os valores que podem ser colocados no atributo *ExceptionAction* são os valores do elemento *Condition* do XPDL. É o mesmo tipo de valor que pode ser colocado no atributo *Condition* do elemento *Connector* do AW.

O valor do atributo *ExceptionReaction* deve seguir uma parte da sintaxe do XPDL. Os valores que podem ser colocados no atributo *ExceptionReaction* são os valores do atributo *tool* do XPDL. É o mesmo tipo de valor que pode ser colocado no atributo *Action* do elemento *Task* do AW;

- D. **taskHeader:** este atributo contém informações complementares sobre a atividade que está sendo modelada.

O valor do atributo *TaskHeader* deve seguir uma parte da sintaxe do XPDL. Os valores que podem ser colocados no atributo *Action* são os valores dos atributos *Deadline*, *BlockActivity*, *Finish Mode*, *Icon*, *Limit*, *Priority*, *SimulationInformation* e *StartMode* do XPDL. Abaixo um exemplo de valor que pode ser colocado no atributo *TaskHeader*.

```

<StartMode><Automatic/></StartMode>
<FinishMode><Automatic/></FinishMode>
  <Deadline Execution="SYNCHR">
    <DeadlineCondition>4 Days</DeadlineCondition>
    <ExceptionName>timeoutException</ ExceptionName >
  </Deadline>

```

## Referências

Elementos semelhantes aos atributos *Participants* e *Actions*, do AW foram primeiramente propostos na representação XPDL da WfMC e no *process model*.

Atributos semelhantes aos atributos *ExceptionAction* e *ExceptionReaction*, do AW foram primeiramente propostos no *process model*.

A representação XPDL da WfMC prevê a representação da entidade exceções de uma maneira muito simplificada. No AW foi implementada uma solução semelhante a proposta do projeto WIDE. Foi tomada esta decisão pelo fato da proposta do projeto WIDE ser uma solução mais completa. Porém o comportamento das "exceções" do *process model* pode ser simulado numa representação XPDL através de um conjunto de elementos do XPDL (BARESI et al., 1999).

Os valores do atributo *TaskHeader* foram primeiramente propostos na representação XPDL da WfMC.

### 6.1.8 Elemento SuperTask

O elemento **SuperTask** viabiliza a implementação de modularizações. Este elemento na representação AW tem características, comportamentos e funções semelhantes ao símbolo *super tarefa* do *process model*. A existência de super tarefas também é prevista na representação XPDL da WfMC.

Os atributos do elemento *SuperTask* são os seguintes:

- A. **atributo href:** este atributo contém uma referência ao arquivo que contém o processo-filho do atual processo na hierarquia do *workflow*.  
O valor do atributo *href* é preenchido com uma URL válida que aponta para o arquivo que contém um processo-filho do corrente processo;
- B. **atributos ExceptionAction e ExceptionReaction:** possuem as mesmas funções e comportamentos que os atributos *ExceptionAction* e *ExceptionReaction* do elemento *Task*;
- C. **atributo Execution:** este atributo determina se o processo filho será executado em modo síncrono ou assíncrono. No modo síncrono, o processo pai dispara o processo filho, espera o processo filho ser executado e só depois que o processo filho é finalizado o processo pai continua a sua execução. No modo assíncrono, o processo pai chama o processo filho e continua a sua execução de forma independente da execução do processo filho.

O valor do atributo *Execution* é preenchido com a constante "SYNCHR" ou "ASYNCHR" que representam respectivamente uma execução síncrona ou assíncrona;

- D. **atributo *ActualParameters***: este atributo contém os parâmetro que o processo filho precisa receber para funcionar corretamente. Os parâmetros que o processo filho precisará receber estão declarados no atributo *FormalParameters* do elemento *Begin* do processo filho.

O valor do atributo *ActualParameters* deve seguir uma parte da sintaxe do XPDL. Os valores que podem ser colocados no atributo *ActualParameters* são os mesmos valores que podem ser colocados dentro do atributo *ActualParameters* do XPDL. Abaixo um exemplo de valor que pode ser colocado no atributo *ActualParameters*.

```
<ActualParameters>
    <ActualParameter>Ordem</ActualParameter>
    <ActualParameter>Quantidade</ActualParameter>
</ActualParameters>
```

## Referências

Um atributo semelhante ao atributo *href* foi proposto primeiramente pela especificação XLink da W3C. A especificação XLink define atributos que descrevem *links* entre elementos ou documentos.

Atributos semelhantes aos atributos *ExceptionAction* e *ExceptionReaction*, do AW foram primeiramente propostos no *process model*.

Atributos e elementos semelhantes aos atributos *ActualParameters* e *Execution* do AW foram primeiramente propostos na representação XPDL da WfMC.

### 6.1.9 Elemento *MultiTask*

O elemento ***MultiTask*** viabiliza a implementação de modularizações. Este elemento na representação AW tem características, comportamentos e funções semelhantes ao símbolo *multi tarefa* do *process model*. A existência de *multi tarefas* não é prevista na representação XPDL da WfMC.

Os atributos do elemento *MultiTask* são os seguintes:

- A. **atributo *href***: este atributo contém uma referência ao arquivo que contém o processo-filho do atual processo na hierarquia do *workflow*.  
O valor do atributo *href* é preenchido com uma URL válida que aponta para o arquivo que contém um processo-filho do corrente processo;
- B. **atributos *ExceptionAction* e *ExceptionReaction***: possuem as mesmas funções e comportamentos que os atributos *ExceptionAction* e *ExceptionReaction* do elemento *Task*;
- C. **atributo *Execution***: possuem as mesmas funções e comportamentos que o atributo *Execution* do elemento *SuperTask*;
- D. **atributo *ActualParameters***: possuem as mesmas funções e comportamentos que o atributo *ActualParameters* do elemento *SuperTask*.

- E. **atributo NJoin:** determina quantas instâncias do processo apontado pelo *MultiTask* serão disparadas. O valor do *NJoin* deve ser preenchido por um número inteiro.
- F. **atributo NFork:** determina quantas cópias do processo apontado pelo *MultiTask* precisam ser finalizadas para que o *workflow* considere como finalizada a multi tarefa. O valor do *NFork* deve ser preenchido por um número inteiro.

### Referências

Um atributo semelhante ao atributo *href* foi proposto primeiramente pela especificação XLink da W3C. A especificação XLink define atributos que descrevem *links* entre elementos ou documentos.

Atributos semelhantes aos atributos *Nfork*, *Njoin*, *ExceptionAction* e *ExceptionReaction*, do AW foram primeiramente propostos no *process model*.

Atributos e elementos semelhantes aos atributos *ActualParameters* e *Execution* do AW foram primeiramente propostos na representação XPDL da WfMC.

#### 6.1.10 Elemento ForkJoinTotal

O elemento **ForkJoinTotal** representa uma entidade de *roteamento* do modelo de *workflow*. Este elemento no modelo AW tem características, comportamentos e funções semelhantes ao símbolo "fork e join total" do *process model*. A existência de *forks* e *joins* totais também é prevista na representação XPDL da WfMC.

O elemento *ForkJoinTotal* possui somente os atributos comuns a todos elementos do *workflow*.

#### 6.1.11 Elemento JoinPartial

O elemento **JoinPartial** representa uma entidade de *roteamento* no processo de *workflow*. Este elemento na representação AW tem características, comportamentos e funções semelhantes ao símbolo *join parcial* do *process model*. A existência de *joins parciais* não é prevista na representação XPDL da WfMC. Porém o comportamento do *join parcial* do *process model* pode ser simulado numa representação XPDL através de um conjunto de elementos do XPDL. Este mapeamento foi proposto em (BARESI et al., 1999).

O elemento *JoinPartial* possui o seguinte atributo:

- A. **atributo NJoin:** quando *NJoin* conectores "chegam" ao *JoinPartial* o *JoinPartial* dispara o conector que o sucede. Quando um conector "chega" no elemento *JoinPartial* significa que as ações do *workflow* que antecedem o referido conector foram completadas. O *JoinPartial* só dispara 1 vez o conector que o sucede.

O valor do *NJoin* deve ser preenchido por um número inteiro maior ou igual a 1.

### Referências

Um Atributo semelhante ao atributo *Njoin* do AW foi primeiramente proposto no *process model*.

### 6.1.12 Elemento JoinInteractive

O elemento **JoinInteractive** representa uma entidade de *roteamento* no modelo de *workflow*. Este elemento na representação AW tem características, comportamentos e funções semelhantes ao símbolo *join interativo* do *process model*. A existência de *joins interativos* não é prevista na representação XPDL da WfMC. Porém o comportamento do *join interativo* do *process model* pode ser simulado numa representação XPDL através de um conjunto de elementos do XPDL. Este mapeamento foi proposto por Baresi et al.(1999).

O elemento JoinInteractive possui o seguinte atributo:

- A. **atributo NJoin:** o valor do *NJoin* deve ser preenchido por um número inteiro maior ou igual a 1.

Toda vez que um número de *NJoin* conectores “chegam” ao *JoinInteractive* o *JoinInteractive* dispara o conector que o sucede. Portanto, o *JoinInteractive* pode disparar o conector que o sucede várias vezes.

### Referências

Um atributo semelhante ao atributo *Njoin* da representação AW foi primeiramente proposto no *process model*.

### 6.1.13 Elemento ForkConditional

O elemento **ForkConditional** representa uma entidade de *roteamento* no processo de *workflow*. Este elemento na representação AW tem características, comportamentos e funções semelhantes ao símbolo *fork condicional* do *process model*. A existência de forks condicionais também é prevista na representação XPDL da WfMC.

O elemento *ForkConditional* possui somente os atributos comuns a todos elementos do *workflow*.

### 6.1.14 Elemento EForkConditional

O elemento **EForkConditional** representa uma entidade de *roteamento* no processo de *workflow*. Este elemento na representação AW tem características, comportamentos e funções semelhantes ao símbolo “fork condicional com exclusão mútua” da representação *process model*. A existência de forks condicionais com exclusão mútua também é prevista na representação XPDL da WfMC.

O elemento *EForkConditional* possui somente os atributos comuns a todos elementos do *workflow*.

## 6.2 Resumo da representação AW

A tabela 6.2 mostra na sua primeira coluna os elementos e atributos da representação AW. Os elementos estão em negrito.

A segunda coluna mostra a entidade do modelo AW correspondente.

A terceira coluna mostra um elemento/atributo das representações *process model* ou XPDL que possui a mesma semântica que o correspondente elemento da representação AW. A representação é indicada entre parênteses “( )”.

A quarta coluna mostra informações sobre a sintaxe do correspondente elemento ou atributo da representação AW.

Tabela 6.2: Elementos e atributos da representação AW

| Atributo ou elemento da representação AW | Entidade do modelo AW               | Elemento ou atributo do <i>process model</i> ou XPDL semanticamente equivalente.    | Sintaxe                        |
|--|-------------------------------------|---|--------------------------------|
| Begin                                    | Início                              | <b>Begin</b> ( <i>process model</i> )   | Adapt. do <i>process model</i> |
| * XLink                                  | Processo pai                        | (proposto nesta pesquisa)   | Sintaxe do XLink               |
| TypeDeclarations                         | Tipos de dados                      | TypeDeclarations (XPDL)   | Adapt. do <i>process model</i> |
| Participants                             | Participantes                       | Participants (XPDL)   | Adaptação do XPDL              |
| Applications                             | Aplicativos                         | Applications (XPDL)   | Adaptação do XPDL              |
| DataFields                               | Dados relevantes                    | DataFields (XPDL)   | Adaptação do XPDL              |
| FormalParameters                         | Parâmetros do processo              | FormalParameters (XPDL)   | Adaptação do XPDL              |
| AccessLevel                              | Nível de acesso                     | AccessLevel (XPDL)  | Adaptação do XPDL              |
| PackageHeader                            | Dados do pacote                     | ConformanceClass, ExternalPackages, PackageHeader, RedefinableHeader e Script(XPDL) | Adaptação do XPDL              |
| ProcessHeader                            | Dados do processo                   | ProcessHeader, RedefinableHeader e AccessLevel (XPDL)                               | Adaptação do XPDL              |
| <b>End</b>                               | Fim                                 | <b>End</b> ( <i>process model</i> )   | Adapt. do <i>process model</i> |
| <b>JoinPartial</b>                       | Join parcial                        | <b>JoinPartial</b> ( <i>process model</i> )   | Adapt. do <i>process model</i> |
| Njoin                                    | Njoin                               | <b>Njoin</b> ( <i>process model</i> )   | Adapt. do <i>process model</i> |
| <b>ForkJoinTotal</b>                     | Fork join total                     | <b>ForkJoinTotal</b> ( <i>process model</i> )                                       | Adapt. do <i>process model</i> |
| <b>JoinInterative</b>                    | Join interativo                     | <b>JoinInterative</b> ( <i>process model</i> )                                      | Adapt. do <i>process model</i> |
| Njoin                                    | Njoin                               | <b>Njoin</b> ( <i>process model</i> )   | Adapt. do <i>process model</i> |
| <b>ForkConditional</b>                   | Fork conditional                    | <b>ForkConditional</b> ( <i>process model</i> )                                     | Adapt. do <i>process model</i> |
| <b>EforkConditional</b>                  | Fork condicional com exclusão mútua | <b>EforkConditional</b> ( <i>process model</i> )                                    | Adapt. do <i>process model</i> |
| <b>Connector</b>                         | Conector                            | <b>Conector</b> ( <i>process model</i> )  | Adapt. do <i>process model</i> |
| FromID                                   | De                                  | From (XPDL)   | Adaptação do XPDL              |
| ToID                                     | Para                                | To (XPDL)   | Adaptação do XPDL              |
| Conditions                               | Condição                            | Conditions (XPDL)   | Adaptação do XPDL              |
| <b>MultiTask</b>                         | Multi atividade                     | <b>MultiTask</b> ( <i>process model</i> )   | Adapt. do <i>process model</i> |
| * Xlink                                  | Id do processo invocado             | Id (XPDL)   | Sintaxe do XLink               |
| ExceptionAction                          | Exceção                             | <b>Exceção</b> ( <i>process model</i> )   | Adaptação do XPDL              |
| ExceptionReaction                        | Reação                              | <b>Reação</b> ( <i>process model</i> )  | Adaptação do XPDL              |
| Execution                                | Modo de execução                    | Execution (XPDL)  | Adaptação do XPDL              |
| ActualParameters                         | Parâmetros atuais                   | ActualParameters (XPDL)   | Adaptação do XPDL              |
| Njoin                                    | Njoin                               | <b>Njoin</b> ( <i>process model</i> )   | Adapt. do <i>process model</i> |
| Nfork                                    | Nfork                               | <b>Nfork</b> ( <i>process model</i> )   | Adapt. do <i>process model</i> |
| <b>SuperTask</b>                         | Super atividade                     | <b>SuperTask</b> ( <i>process model</i> )   | Adapt. do <i>process model</i> |
| * Xlink                                  | Id do processo invocado             | Id (XPDL)   | Sintaxe do XLink               |
| ExceptionAction                          | Exceção                             | <b>Exceção</b> ( <i>process model</i> )   | Adaptação do XPDL              |
| ExceptionReaction                        | Reação                              | <b>Reação</b> ( <i>process model</i> )  | Adaptação do XPDL              |
| Execution                                | Modo de execução                    | Execution (XPDL)  | Adaptação do XPDL              |
| ActualParameters                         | Parâmetros atuais                   | ActualParameters (XPDL)   | Adaptação do XPDL              |
| <b>Task</b>                              | Atividade atômica                   | <b>Task</b> ( <i>process model</i> )  | Adaptação do XPDL              |
| Participant                              | Participante                        | Participant (XPDL)  | Adaptação do XPDL              |

|   |                     |   |                         |
|---|---------------------|---|-------------------------|
| Action  | Ação                | Tool (XPDL)   | Adaptação do XPDL       |
| ExceptionAction   | Exceção             | <b>Exceção (<i>process model</i>)</b>   | Adaptação do XPDL       |
| ExceptionReaction   | Reação              | <b>Reação (<i>process model</i>)</b>  | Adaptação do XPDL       |
| ActualParameters  | Parâmetros atuais   | ActualParameters (XPDL)   | Adaptação do XPDL       |
| TaskHeader  | Dados da tarefa     | Deadline, BlockActivity, Finish Mode, Icon, Limit, Priority, SimulationInformation e StartMode (XPDL) | Adaptação do XPDL       |
| <b>Atributos/Entidades encontrados em vários elementos.</b> |                     |   |                         |
| Id  | Identificador único | Id (XPDL)   | Adaptação do XPDL       |
| Name  | Nome                | Name (XPDL)   | Adaptação do XPDL       |
| Description   | Descrição           | Description (XPDL)  | Adaptação do XPDL       |
| ExtendedAttributes  | Atributo estendido  | ExtendedAttributes (XPDL)   | Adaptação do XPDL       |
| Error   | Erros               | <b>(Proposta nesta pesquisa)</b>  | proposta nesta pesquisa |
| <b>* SVG</b>  | Dados gráficos      | <b>(Proposta nesta pesquisa)</b>  | Sintaxe do SVG          |

Para compreender a **semântica** de cada elemento e atributo da representação AW basta ver a semântica da correspondente entidade no modelo AW. A correspondência entre a representação e o modelo AW é dada nas colunas 1 e 2 da tabela 6.2.

Para auxiliar a compreensão semântica da representação AW também pode ser utilizada a coluna 3 da tabela 6.2. A coluna 3 mostra um elemento/atributo das representações *process model* ou XPDL que tem um significado semântico semelhante ao correspondente elemento AW da coluna 1.

Comparando-se as colunas 1 e 3 da tabela 6.2 pode-se notar que quase todos elementos e atributos da representação AW tem um significado semântico semelhante ao significado semântico de elementos das representações *process model* e XPDL. Os atributos error, \*XLink e \*SVG por terem sido definidos nesta pesquisa não possuem um atributo correspondente nas representações *process model* e XPDL.

A **sintaxe** de cada elemento/atributo da representação AW é descrita na coluna 4 da tabela 6.2. Abaixo uma explicação mais detalhada de cada descrição contida na coluna 4 da tabela 6.2:

- **adaptação do *process model*:** o *process model* descreve um *workflow* através de textos e gráficos. Na representação AW, a sintaxe destes elementos teve que ser adaptada para o padrão XML;
- **adaptação do XPDL:** o XPDL contém uma sintaxe relativamente complexa, formada por várias estruturas de elementos aninhadas. Como o AW é uma representação que tem por objetivo ser facilmente manipulada por editores gráficos, algumas simplificações foram feitas. Assim, na representação AW, a sintaxe destes elementos teve que ser adaptada para um formato mais simples;
- **sintaxe do XLink:** são atributos da representação XLink que são adicionados na representação AW através do padrão namespace;
- **sintaxe do SVG:** são atributos da representação SVG que são adicionados na representação AW através do padrão namespace;
- **proposta nesta pesquisa:** a sintaxe destes atributos foram propostas nesta pesquisa.

## 6.3 Análise da representação

A seguir serão analisadas e justificadas algumas das decisões que determinaram o atual formato da representação AW.

### 6.3.1 O *software* Amaya e a tecnologia XML

O AW foi construído como uma extensão do *software* Amaya. Portanto, foi necessário tornar a representação AW compatível com as representações processadas pelo Amaya. Para se manter compatível com o *software* Amaya, basta a representação AW ser uma representação hierárquica compatível com o padrão XML. O fato do XML ser uma tecnologia com muitos recursos, amplamente difundida e das representações SVG, XLink e XPDL também serem linguagens XML reforçou ainda mais a escolha de uma representação do modelo de *workflow* compatível com o padrão XML.

### 6.3.2 As representações *process model* e XPDL

A representação gráfica do *process model* é uma representação relativamente simples contendo poucos elementos. Isto facilita a compreensão e a manipulação do modelo por parte do usuário. Entretanto o *process model* não contém toda a informação necessária para se executar um *workflow*.

Já a representação XPDL pode representar todos os dados necessários para se executar um processo segundo os padrões da WfMC. Todavia a representação XPDL é relativamente complexa, contendo muitos elementos e atributos. Esta característica dificulta a compreensão do modelo e a sua manipulação.

Tendo em vista as características destes dois modelos, o AW se propôs a fornecer a facilidade de compreensão e manipulação do *process model* com a capacidade de representação do XPDL.

Para atingir esta meta foi definido que os elementos da representação AW seriam os mesmos elementos definidos no *process model*. Com isto o AW passa a ter poucos elementos e todos com uma representação gráfica que pode ser facilmente compreendida e manipulada pelo usuário. Contudo ainda falta ao modelo uma maior representatividade.

Uma maior representatividade é dada ao modelo adicionando-se atributos aos elementos do AW. Os atributos adicionados irão armazenar toda informação necessária para se definir completamente um *workflow* compatível com os padrões da WfMC.

Criar um novo atributo no AW para cada elemento do XPDL tornaria o modelo AW complexo. Haveria muitos atributos para cada elemento. Por isso os elementos e atributos do XPDL foram agrupados visando tornar o modelo mais compreensível. Cada grupo de elementos foi adicionado ao modelo AW como um único atributo. Assim, tem-se no modelo AW atributos que devem ser preenchidos por valores que seguem a sintaxe de uma ou várias sub-árvores da linguagem XPDL.

Com isto tem-se um modelo mais compreensível que o XPDL e mais representativo que modelo WIDE. Contudo a manipulação e edição do modelo continuam complexas, pois a dificuldade de se preencher um atributo que segue a sintaxe de uma sub-árvore do XPDL é a mesma de se definir esta sub-árvore no XPDL. Esta dificuldade pode ser superada através de interfaces que disponibilizam aos usuários formulários amigáveis com recursos que irão facilitar o preenchimento dos atributos.

Portanto, houve uma transferência de complexidade. A complexidade foi tirada da estrutura de elementos e atributos do modelo e colocada no preenchimento dos valores dos atributos.

As principais vantagens de se manter a complexidade no preenchimento dos valores dos atributos são:

- ao invés de ter um grande e complexo problema que é, criar, manter, implementar, compreender, analisar e utilizar uma complexa estrutura de elementos e atributos. Pode-se ter vários pequenos problemas que são: a criação de interfaces amigáveis que facilitem o preenchimento de cada atributo. Utilizou-se, portanto a tática do "dividir para conquistar", uma vez que cada interface é um problema relativamente isolado que pode ser pensado a parte;
- a manipulação das estruturas de atributos e elementos é mais utilizada pelos usuários que a manipulação dos valores dos atributos. A grande maioria das manipulações de um processo mexe ou utiliza a estrutura de atributos e elementos do AW. Já o preenchimento de atributos é uma ação que ocorre poucas vezes. Poucos processos necessitam do preenchimento de todos atributos. Os atributos definidos em um pacote não precisam ser preenchidos nos demais processos do pacote. Utilizou-se, portanto, a tática de se tornar mais simples os recursos mais utilizados. Em contra partida, os recursos menos utilizados tornaram-se mais complexos;
- a estrutura de atributos e elementos influencia ou é utilizada por quase todas as funcionalidades da ferramenta. Já o preenchimento dos valores de um atributo é um aspecto mais isolado que influencia ou é utilizada por poucas funcionalidades da ferramenta AW. Novamente utilizou-se a tática de se tornar mais simples os recursos mais utilizados.

A principal desvantagem de se transferir a complexidade é ter que tratar de um modo os elementos de *workflow* que estão na estrutura do modelo e de outro modo, os elementos que estão dentro dos atributos.

Da forma como foi desenvolvido o AW deve-se tratar de uma maneira as informações dos elementos e atributos e de outra maneira as informações das sub-árvores XPD L que estão contidas nos valores dos atributos. A segunda forma de tratamento não seria necessária se a complexidade das estruturas dos valores fossem colocadas diretamente na estrutura de elementos e atributos do AW.

### 6.3.3 A representação SVG

Para viabilizar a implementação de uma visão gráfica de um processo de *workflow* é necessário que atributos gráficos sejam associados aos elementos de *Workflow*.

Para estar de acordo com os objetivos do AW os recursos de representação gráfica devem ser: (i) compatíveis com o padrão XML e (ii) padronizados, amplamente estudados, difundidos e utilizados.

O SVG atende a todos estes requisitos, pois é uma linguagem XML padrão da W3C para descrever gráficos bidimensionais. O SVG é compatível com o Amaya (e conseqüentemente com o AW), não só por ser uma linguagem XML, mas também pelo fato do Amaya possuir um visualizador SVG. Tal visualizador, através de poucas adaptações, foi utilizado por esta pesquisa para mostrar uma visão gráfica das representações AW.

### 6.3.4 XLink

Para permitir a navegação e a definição de uma hierarquia de processos são necessários recursos que estabeleçam ligações entre processos que estão definidos em diferentes documentos.

Para estarem de acordo com os objetivos do AW os recursos que estabelecerão as ligações entre os processos devem ser compatíveis com o padrão XML, padronizados, amplamente estudados e amplamente difundidos.

O XLink atende a todos estes requisitos, pois é uma linguagem XML padrão da W3C para descrever ligações (*links*) entre elementos de diferentes documentos. O XLink é compatível com o Amaya (e conseqüentemente com o AW). Não só por ser uma linguagem XML, mas também pelo fato do Amaya ser um editor de hiperdocumentos capaz de localizar e mostrar documentos apontados por um *link*.

### 6.3.5 Namespace

Para se armazenar informações gráficas e *links* em uma definição de processo existem duas opções: (i) criar novos atributos na representação, ou (ii) utilizar a tecnologia Namespace para inserir atributos de linguagens externas na representação do modelo de *workflow*.

A primeira opção é a solução mais simples de ser implementada, pois não precisa que a estrutura que utiliza o modelo tenha suporte a tecnologia Namespace. Entretanto esta solução torna o modelo mais complexo, pois ele precisará definir a semântica e a sintaxe de atributos que armazenarão informações sobre *workflows*, apontamentos e gráficos. Outra desvantagem é que a evolução das três funcionalidades citadas dependerá da evolução de um único modelo. As três funcionalidades são dependentes entre si, pois pertencem a um mesmo modelo.

Já utilizando Namespace pode-se obter a ortogonalidade das três funções. As 3 funcionalidades estão definidas em 3 modelos/representações diferentes, que podem evoluir independentemente. No caso do AW, há uma representação de *workflow* que utiliza através do Namespace os atributos das linguagens SVG e XLink. Não importa o quão complexas sejam estas linguagens. A complexidade das demais linguagens não influencia a complexidade do AW. O AW utiliza somente os atributos externos que lhe podem ser úteis. Os demais atributos e elementos do SVG e XLink podem ser utilizados em futuras evoluções do AW sem que no momento influenciem na complexidade do modelo AW.

### 6.3.6 Funcionalidades do AW

Alguns atributos que viabilizaram a implementação de funcionalidades do AW foram adicionados aos elementos da representação AW. Estes atributos foram propostos por esta pesquisa e estão diretamente relacionados com alguma funcionalidade. É o caso do atributo *error* que viabilizou a implementação de funcionalidades de verificação de erros.

## 7 ANÁLISE E COMPARAÇÃO DOS SISTEMAS EXISTENTES

Os objetivos deste capítulo são: (i) demonstrar a aplicação do método de descrição de modelos conceituais proposta no capítulo 4; (ii) mostrar um panorama das funcionalidades, conceitos e interações das principais ferramentas de definição existentes; (iii) apresentar um método de avaliação de modelos conceituais de ferramentas de definição e (iv) fazer uma análise comparativa entre o modelo conceitual do AW e os modelos conceituais de sistemas existentes.

Para tanto, será feita uma análise comparativa dos modelos conceituais das principais ferramentas de definição disponíveis. Após, será definido um método de avaliação de modelos conceituais. Então, serão concisamente apresentados e avaliados os modelos conceituais de alguns sistemas existentes. Estes sistemas foram escolhidos por possuírem as principais características deste tipo de ferramenta, por serem bem conceituados e pela sua disponibilidade para testes. Finalmente, será feita uma análise comparativa entre o modelo conceitual do AW e o modelo conceitual dos sistemas existentes.

### 7.1 Análise comparativa de modelos conceituais de ferramentas de definição

Este capítulo apresenta uma análise comparativa dos modelos conceituais de ferramentas de definição disponíveis nos meios acadêmicos e comerciais. As conclusões deste capítulo foram obtidas através de: (i) pesquisas sobre trabalhos e artigos relacionados com o assunto (apresentados nesta dissertação); (ii) estudos sobre o funcionamento de ferramentas como o *scarabaeus* (CASTROIANNI, 1995), *WebSphere Business Integration Modeler* da IBM (IBM, 2001), *Oracle workflow builder* da Oracle (ORACLE, 2002) e (iii) análises e testes detalhados nas ferramentas de definição *Foro model designer*, *ILOG workflow modeler*, *Ultimus designer*. A seguir a análise comparativa.

Os modelos adotados pelas ferramentas de definição geralmente seguem o modelo definido pela WfMC possuindo poucas entidades a mais ou a menos. O conceito de definição de processo é adotado por todas ferramentas. O conceito de repositório é um diferencial de algumas ferramentas, pois nem todas principais ferramentas de definição adotam este conceito. Já, as representações do modelo de *workflow* adotadas por cada ferramenta variam bastante. Praticamente há um conjunto diferente de representações para cada ferramenta de definição.

As funcionalidades indispensáveis em aplicativos de qualidade e que são comuns à maioria das ferramentas de definição são: pelo menos uma visão gráfica, várias funcionalidades de edição, controles da definição, regras dos controles, modularização e reaproveitamento de processos.

Já as seguintes funcionalidades não estão presentes em todas as ferramentas de definição de boa qualidade. Porém podem ser consideradas como importantes diferenciais destas ferramentas. São elas: várias visões sincronizadas, vários formatos de exportação e importação, gerenciamento do repositório, verificação de erros, simulação, recursos para cooperação, adaptabilidade e interoperabilidade.

Os dispositivos de hardware são geralmente o teclado, o mouse, o monitor e a impressora. São geralmente utilizados todos os estilos de interação apresentados nesta dissertação (menus, formulários, hiperlink e manipulação direta). Finalmente, cada ferramenta geralmente adota um padrão de interação.

Pode-se concluir desta análise que há um grupo de requisitos básicos que são comuns aos modelos conceituais das principais ferramentas disponíveis. A falta de um destes requisitos compromete a qualidade do modelo conceitual analisado. Já outros requisitos (requisitos de distinção) variam de ferramenta para ferramenta e podem ser considerados os diferenciais de cada sistema. O contexto onde a ferramenta será aplicada é que determinará se a falta ou a presença de um destes requisitos pode ou não comprometer a qualidade do modelo conceitual analisado. A tabela 7.1 mostra os requisitos citados.

Tabela 7.1: Tabela de requisitos de modelos conceituais de ferramentas de definição

| Partes do modelo conceitual | Requisitos básicos   | Requisitos de distinção   |
|-----------------------------|--|---|
| Conceitos                   | Adotar um modelo de <i>workflow</i> semelhante ao modelo da WfMC. Adotar 1 representação gráfica. Adotar o conceito definição de processo  | Itens adicionados ou retirados do modelo da WfMC.<br>Adotar várias representações.<br>Adotar o conceito repositório.  |
| Funcionalidades             | 1 visão gráfica, funcionalidades de edição, controles da definição, regras dos controles, reaproveitamento de processos e modularização  | Várias visões sincronizadas, verificação de erros; simulação; recursos para cooperação; adaptabilidade; interoperabilidade; vários formatos de exportação; vários formatos de importação; gerenciamento do repositório. |
| Modelo de interação         | Dispositivos de hardware: mouse e teclado (entrada), impressora e monitor (saída).<br>Estilos de interação: menus, formulários, hiperlink e manipulação direta.<br>Adoção de um padrão de interação. |   |

Com base na análise comparativa apresentada, esta dissertação propõe um método de avaliação de modelos conceituais de ferramentas de definição composto dos seguintes passos:

- montar a tabela de descrição do modelo conceitual da ferramenta avaliada;
- comparar a tabela de descrição do modelo conceitual com a tabela 7.1;
- verificar se a ferramenta avaliada possui os requisitos básicos para este tipo de ferramenta. Tais requisitos são essenciais para qualquer contexto de aplicação;

- verificar os requisitos de distinção da ferramenta avaliada;
- comparar os requisitos de distinção da ferramenta com as necessidades específicas do contexto onde a mesma será aplicada. Numa boa configuração, todas as necessidades específicas do contexto deverão ser atendidas pelos diferenciais da ferramenta.

Este método será utilizado no capítulo 7.2 para avaliar os modelos conceituais das ferramentas *Foro model designer*, *Ilog workflow modeler* e *Ultimus designer*. No capítulo 7.3 será utilizado para avaliar o modelo conceitual do AW.

## 7.2 Sistemas existentes

### 7.2.1 Foro model designer

O *Foro model designer* (FORO, 2001) é uma ferramenta de definição desenvolvida pela SchlumbergerSema (SCHLUMBERGER, 2004) que faz parte do sistema de *workflow* Foro. Além do *Foro model designer*, o sistema Foro inclui ferramentas de monitoração, ferramentas de administração e uma máquina de *workflow*. O sistema de *workflow* Foro utiliza as pesquisas e conceitos desenvolvidos no projeto WIDE (BARESI et al., 1999) dentre as quais destacam-se as representações *Process Model* e *Information Model*.

Os conceitos adotados pelo *Foro model designer* são semelhantes aos conceitos adotados pelos principais sistemas de definição de processo disponíveis. É adotado no *Foro model designer* um modelo de *workflow* semelhante ao modelo da WfMC com a adição das entidades transações, recursos, exceção e informações gráficas. Assim como a maioria das ferramentas de definição, o *Foro* adotou o conceito de definição de processo e suas próprias representações, sendo 1 delas gráfica. A adoção do conceito repositório é um de seus diferenciais.

O modelo de interação adotado pelo *Foro* também é semelhante aos modelos de interação adotados pelos principais sistemas de definição de processo disponíveis. O *Foro* utiliza como dispositivos de hardware: o teclado, o mouse, o monitor e a impressora. São utilizados no *Foro* os estilos de interação: menus, formulários, *hiperlink* e manipulação direta. Foi adotado no *Foro* o padrão de interação Windows.

O *Foro* possui as seguintes funcionalidades básicas e comuns à maioria das ferramentas de definição: pelo menos uma visão gráfica, várias funcionalidades de edição, controles da definição, regras dos controles, modularização e reaproveitamento de processos.

Com relação as suas funcionalidades, os principais diferenciais do *Foro* são: a verificação automática de erros de sintaxe, interoperabilidade do modelo e a interoperabilidade da representação, exportação, importação e gerenciamento do repositório.

Podem ser considerados pontos fracos do *Foro* os poucos recursos de cooperação, adaptabilidade, interoperabilidade tecnológica e a disponibilização de apenas uma visão.

O próximo passo de uma avaliação da ferramenta *Foro* seria a comparação destas características com as necessidades do contexto onde a ferramenta seria utilizada.

A tabela 7.2 mostra a tabela de descrição do modelo conceitual da ferramenta *Foro model designer*.

Tabela 7.2: Tabela de descrição do modelo conceitual do *Foro model designer*.

| <b>Itens do modelo</b>             | <b>Modelo conceitual do Foro</b>   |
|------------------------------------|--|
| <b>Conceitos</b>                   |  |
| Modelo de <i>workflow</i>          | Atividade, roteamento, participante, aplicativos, dados relevantes, aspectos temporais, prioridade, versão, processo, pacote, repositório, exceções, informações gráficas, transações e recursos |
| Representação do modelo            | 1 Gráfica (Representação gráfica do Process Model), 2 representações textuais complementares ( <i>Process Model</i> e <i>Information Model</i> )   |
| Definição de processo              | Interpretada (Foro); Exportada (WPDL), Importada (WPDL)  |
| Repositório                        | O repositório é definido em cada seção, fazem parte do repositório os processos abertos em cada seção.   |
| <b>Funcionalidades</b>             |  |
| Visões                             | 1 visão (Representação gráfica do <i>Process Model</i> )   |
| Funcionalidades de edição          | Copiar, recortar, colar, desfazer ações, refazer ações, impressão, zoom, barra de rolagem, barra de status   |
| Controles da definição de processo | Manipular os elementos; manipular os atributos;  |
| Regras dos Controles               | Verificar e reagir a erros; prevenção de erros; estabelecer relacionamentos; manter a coerência  |
| Reedição e reaproveitamento        | Abrir, fechar, salvar  |
| Exportação                         | (WPDL)   |
| Importação                         | (WPDL)   |
| Gerenciamento do repositório       | Manipulação de documentos, visualização dos processos, acesso a um processo  |
| Modularização                      | Navegar pela hierarquia, editar hierarquia   |
| Verificação de erros               | Verifica sintaxe, verifica semântica, relatório de erros   |
| Simulação                          | Modelagem da simulação e interoperabilidade com a WfMC   |
| Recursos para cooperação           |  |
| Adaptabilidade                     | Configuração das funcionalidades   |
| Interoperabilidade                 | Modelo: Compatível com o modelo da WfMC<br>Representação: Mapeamento para o XPDL sem perdas e sem o uso do ExtendedAttribute<br>Tecnológica: Web Services  |
| <b>Interação</b>                   |  |
| Dispositivos de hardware           | Entrada: Teclado, mouse.<br>Saída: Monitor, Impressora   |
| Estilos de Interação utilizados    | Menus, formulários, hiperlink, manipulação direta  |
| Padrão de interação                | Windows  |

A figura 7.1 mostra uma tela do *Foro model designer*.

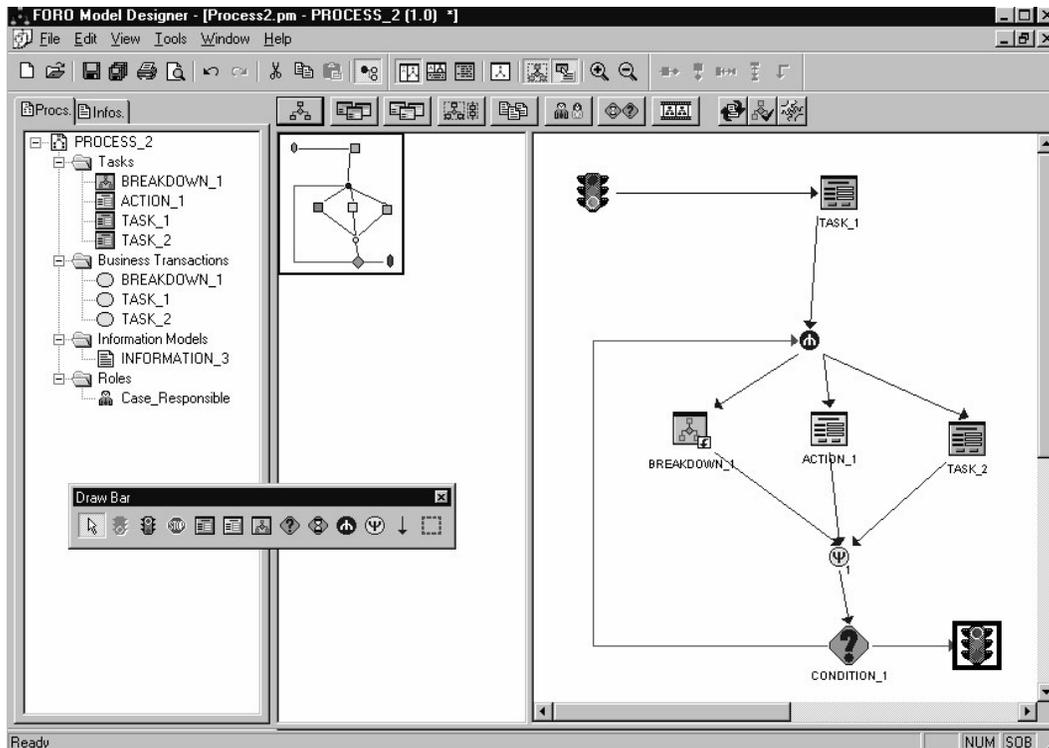


Figura 7.1: Tela do *Foro model designer*

### 7.2.2 ILOG *workflow modeler*

O *ILOG workflow modeler* (ILOG, 2001) é uma ferramenta de definição desenvolvida pela ILOG (ILOG, 2004). Esta ferramenta de definição é um aplicativo construído a partir de componentes Java que fazem parte de um conjunto de soluções desenvolvidas pela ILOG chamado de *ILOG Components for Business Process Management Solutions*. Os componentes Java da ILOG são disponibilizados junto com um ambiente de desenvolvimento de ferramentas que permite a criação e adaptação de ferramentas. O *ILOG workflow modeler* é uma ferramenta desenvolvida a partir destes componentes que pode ser utilizada da forma como é disponibilizada ou adaptada a partir do seu ambiente de desenvolvimento.

Os conceitos adotados pelo *ILOG workflow modeler* são semelhantes aos conceitos adotados pelos principais sistemas de definição de processo disponíveis. É adotado no *ILOG workflow modeler* um modelo de *workflow* semelhante ao modelo da WfMC com a adição das entidades transações e informações gráficas. Assim como a maioria das ferramentas de definição, o *ILOG* adotou o conceito de definição de processo e suas próprias representações, sendo 1 delas gráfica. A adoção do conceito repositório é um de seus diferenciais.

O modelo de interação adotado pelo *ILOG* também é semelhante aos modelos de interação adotados pelos principais sistemas de definição de processo disponíveis. O *ILOG* utiliza como dispositivos de hardware: o teclado, o mouse, o monitor e a impressora. São utilizados no *ILOG* os estilos de interação: menus, formulários, *hyperlink* e manipulação direta. Foi adotado no *ILOG* o padrão de interação *Swing*.

O *ILOG* possui as seguintes funcionalidades básicas e comuns à maioria das ferramentas de definição: pelo menos uma visão gráfica, várias funcionalidades de

edição, controles da definição, regras dos controles, modularização e reaproveitamento de processos.

Com relação as suas funcionalidades, os principais diferenciais do *ILOG* são: adaptabilidade, interoperabilidade do modelo, interoperabilidade da representação, interoperabilidade tecnológica, exportação, importação, disponibilização de várias visões sincronizadas e gerenciamento do repositório.

Podem ser considerados pontos fracos do *ILOG* os poucos recursos de cooperação e verificação de erros.

O próximo passo de uma avaliação da ferramenta *ILOG* seria a comparação destas características com as necessidades do contexto onde a ferramenta seria utilizada.

A tabela 7.3 mostra a tabela de descrição do modelo conceitual do *ILOG workflow modeler* (foi avaliada a versão disponibilizada pela ILOG, sem adaptações).

Tabela 7.3: Tabela de descrição do modelo conceitual do *ILOG workflow modeler*

| <b>Itens do modelo</b>             | <b>Modelo conceitual do ILOG</b>   |
|------------------------------------|--|
| <b>Conceitos</b>                   |  |
| Modelo de <i>workflow</i>          | Atividade, roteamento, participante, aplicativos, dados relevantes, aspectos temporais, extensão, processo, pacote, repositório, informações gráficas e transações |
| Representação do modelo            | 3 Gráficas (Diagram, Time-line, Table), 1 XML (XML Ilog format, inspirada na XPDL)   |
| Definição de processo              | Interpretada (XML Ilog format); Exportada (XPDL), Importada (XPDL)   |
| Repositório                        | Um repositório pode ser definido logicamente e as informações podem ser armazenadas em um documento que pode ser recuperado posteriormente.                        |
| <b>Funcionalidades</b>             |  |
| Visões                             | 3 Visões gráficas sincronizadas (Diagram, Time-line, Table)  |
| Funcionalidades de edição          | Copiar, recortar, colar, desfazer ações, refazer ações, impressão, zoom, barra de rolagem, barra de status, algoritmos de re-arranjo, filtros e adaptação do zoom. |
| Controles da definição de processo | Manipular os elementos; manipular os atributos;  |
| Regras dos Controles               | Verificar e reagir a erros; prevenção de erros; estabelecer relacionamentos; manter a coerência  |
| Reedição e reaproveitamento        | Abrir, fechar, salvar, recarregar  |
| Exportação                         | (WPDL, XPDL, SVG)  |
| Importação                         | (WPDL, XPDL)   |
| Gerenciamento do repositório       | Manipulação de documentos, visualização dos processos, acesso a um processo  |
| Modularização                      | Navegar pela hierarquia, editar hierarquia   |
| Verificação de erros               |  |
| Simulação                          | modelagem da simulação, animação gráfica, armazenamento dos dados, geração de relatórios   |
| Recursos para cooperação           | compartilhamento (http)  |
| Adaptabilidade                     | Adaptação dos símbolos, configuração das funcionalidades, adaptação da interface   |
| Interoperabilidade                 | Modelo: Compatível com o modelo da WfMC<br>Representação: Mapeamento para o XPDL sem perdas e sem o uso do ExtendedAttribute                                       |

|                                 |  |
|---------------------------------|--|
|                                 | Tecnológica: Web Services, XML, http, ambiente de programação Java |
| <b>Interação</b>                |  |
| Dispositivos de hardware        | Entrada: Teclado, mouse.<br>Saída: Monitor, Impressora             |
| Estilos de Interação utilizados | Menus, formulários, hiperlink, manipulação direta                  |
| Padrão de interação             | Swing (Sun Microsystems)   |

A figura 7.2 mostra uma tela do *ILOG workflow modeler*.

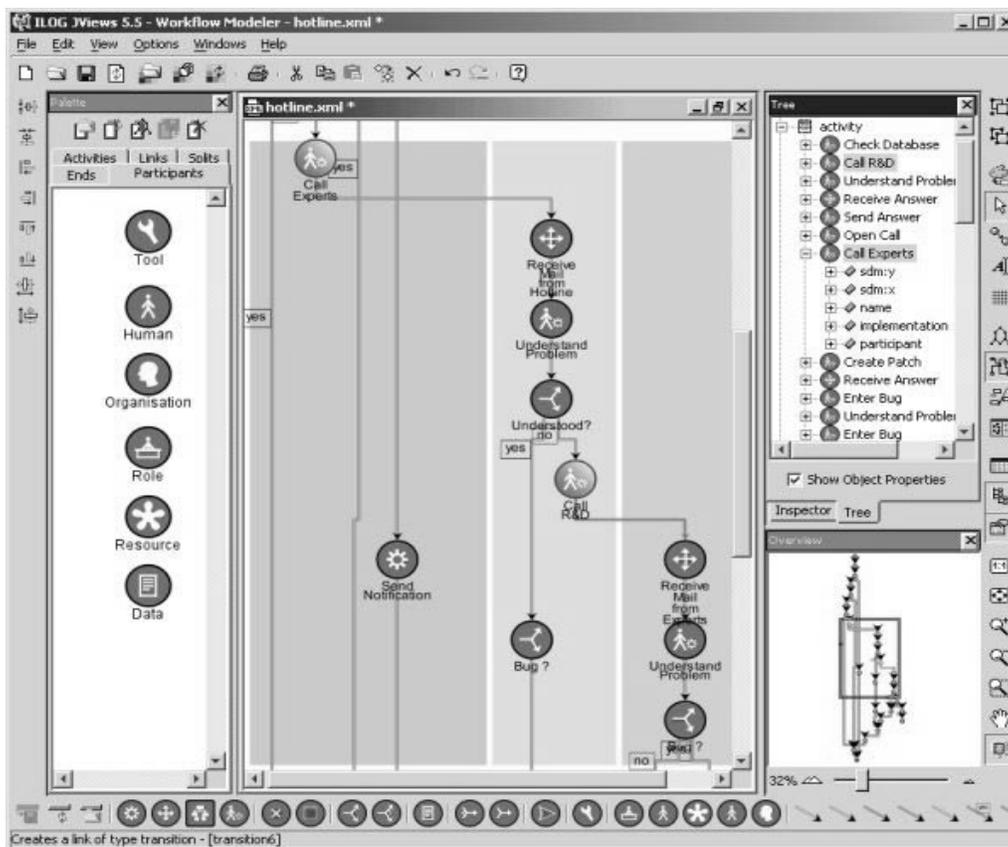


Figura 7.2: Tela do *ILOG workflow modeler*

### 7.2.3 Ultimus designer

O *Ultimus designer* (ULTIMUS, 2003) é uma ferramenta de definição desenvolvida pela Ultimus (ULTIMUS, 2004) que faz parte do sistema de *workflow Ultimus Workflow Suite*. Além do *Ultimus designer*, o sistema *Ultimus Workflow Suite* inclui ferramentas de monitoração, ferramentas de administração, ferramentas de integração de aplicativos e uma máquina de *workflow*.

Os conceitos adotados pelo *Ultimus designer* são semelhantes aos conceitos adotados pelos principais sistemas de definição de processo disponíveis. É adotado no *Ultimus designer* um modelo de *workflow* semelhante ao modelo da WfMC com a adição das entidades eventos e informações gráficas. Assim como a maioria das ferramentas de definição, o *Ultimus* adotou o conceito de definição de processo e suas próprias representações, sendo 1 delas gráfica.

O modelo de interação adotado pelo *Ultimus* também é semelhante aos modelos de interação adotados pelos principais sistemas de definição de processo disponíveis. O *Ultimus* utiliza como dispositivos de hardware: o teclado, o mouse, o monitor e a impressora. São utilizados no *Ultimus* os estilos de interação: menus, formulários, *hiperlink* e manipulação direta. Foi adotado no *Ultimus* o padrão de interação *Windows*.

O *Ultimus* possui as seguintes funcionalidades básicas e comuns à maioria das ferramentas de definição: pelo menos uma visão gráfica, várias funcionalidades de edição, controles da definição, regras dos controles, modularização e reaproveitamento de processos.

Com relação as suas funcionalidades, os principais diferenciais do *Ultimus* são: os recursos de simulação e a interoperabilidade tecnológica.

Podem ser considerados pontos fracos do *Ultimus* os poucos recursos de cooperação, a interoperabilidade do modelo, a interoperabilidade da representação, a verificação de erros, a importação, e o gerenciamento do repositório. O próximo passo de uma avaliação da ferramenta *Ultimus* seria a comparação destas características com as necessidades do contexto onde a ferramenta seria utilizada.

A tabela 7.4 mostra a tabela de descrição do modelo conceitual da ferramenta *Ultimus designers*.

Tabela 7.4: Tabela de descrição do modelo conceitual do *Ultimus designers*

| <b>Itens do modelo</b>          | <b>Modelo conceitual do Ultimus</b>   |
|---------------------------------|---|
| <b>Conceitos</b>                |   |
| Modelo de <i>workflow</i>       | Atividade, roteamento, participante, aplicativos, dados relevantes, aspectos temporais, eventos, exceções, processo, pacote, informações gráficas   |
| Representação do modelo         | 1 Gráfica (Process Map), 1 representação de baixo nível (WFL), 1 Texto (Relatório adaptável no formato Microsoft Word)  |
| Definição de processo           | Interpretada (WFL); Exportada (Relatorio Word), Importada (XPDL)  |
| Repositório                     |   |
| <b>Funcionalidades</b>          |   |
| Visos                           | 1 Visão gráfica (Process Map)   |
| Funcionalidades de edição       | copiar, recortar, colar, desfazer ações, refazer ações, impressão, barra de rolagem, barra de status, fundo quadriculado  |
| Controles da definição          | manipular os elementos; manipular os atributos;   |
| Regras dos Controles            | Verificar e reagir a erros; prevenção de erros; estabelecer relacionamentos; manter a coerência   |
| Reedição e reaproveitamento     | abrir, fechar, salvar   |
| Exportação                      | (Word)  |
| Importação                      |   |
| Gerenciamento do repositório    |   |
| Modularização                   | Navegar pela hierarquia, editar hierarquia  |
| Verificação de erros            |   |
| Simulação                       | modelagem da simulação, animação gráfica, armazenamento dos dados, geração de relatórios  |
| Recursos para cooperação        |   |
| Adaptabilidade                  | Adaptação dos símbolos, configuração das funcionalidades  |
| Interoperabilidade              | Modelo: Parcialmente compatível com o modelo da WfMC (faltam alguns conceitos e sobram outros)<br>Representação: Baixa interoperabilidade pois as representações gráficas e de baixo nível possuem elementos e atributos que estão sobrando ou faltando em relação ao XPDL. Além disto é oneroso mapear uma representação gráfica ou de baixo nível para uma representação XML ou mesmo textual.<br>Tecnológica: Web services, XML, Microsoft word e excel, ODBC, smtp, dll |
| <b>Interação</b>                |   |
| Dispositivos de hardware        | Entrada: Teclado, mouse.<br>Saída: Monitor, impressora  |
| Estilos de Interação utilizados | Menus, formulários, hiperlink, manipulação direta   |
| Padrão de interação             | Windows   |

A figura 7.3 mostra uma tela do *Ultimus designers*.

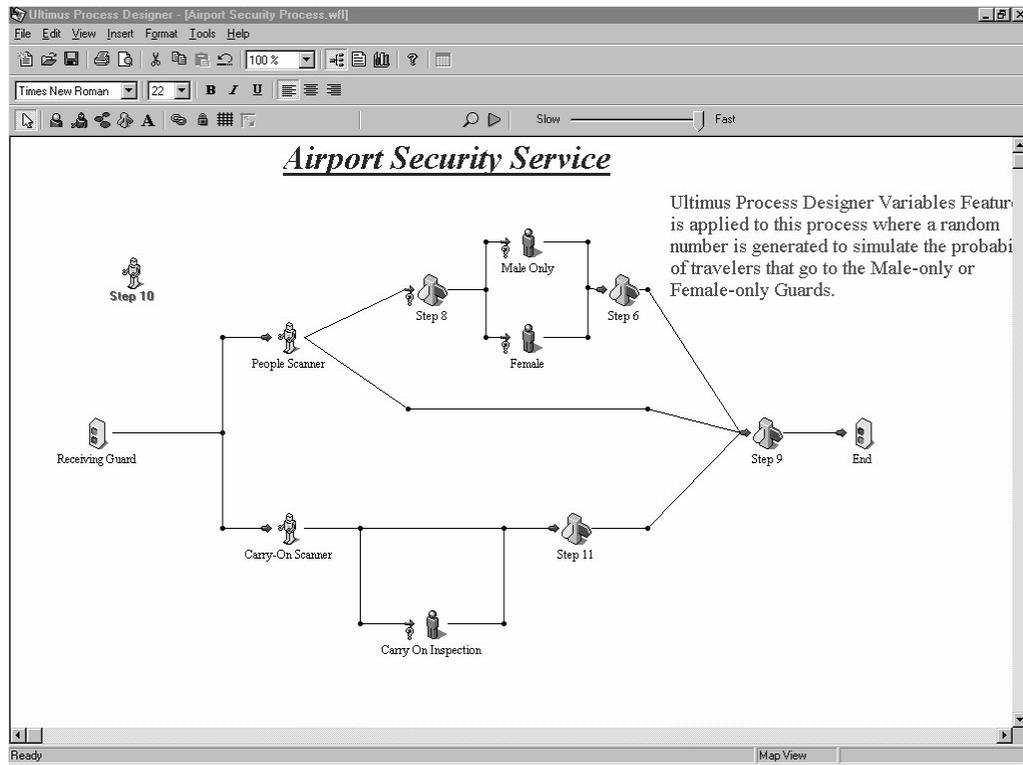


Figura 7.3: Tela do *Ultimus designers*

### 7.3 Análise comparativa do modelo conceitual do Amaya Workflow

Os conceitos adotados pelo AW são semelhantes aos conceitos adotados pelos principais sistemas de definição de processo disponíveis. É adotado no AW um modelo de *workflow* semelhante ao modelo da WfMC com a adição das entidades exceção e informações gráficas. Assim como os demais sistemas, o AW adotou o conceito de definição de processo e suas próprias representações sendo 1 delas gráfica. A adoção do conceito repositório é um de seus diferenciais.

O modelo de interação adotado pelo AW também é semelhante aos modelos de interação adotados pelos principais sistemas de definição de processo disponíveis. O AW utiliza como dispositivos de hardware: o teclado, o mouse, o monitor e a impressora. São utilizados no AW os estilos de interação: menus, formulários, *hiperlink* e manipulação direta. Foi adotado no AW o padrão de interação OpenMotif.

O AW possui as seguintes funcionalidades básicas e comuns à maioria das ferramentas de definição: pelo menos uma visão gráfica, várias funcionalidades de edição, controles da definição, regras dos controles, modularização e reaproveitamento de processos.

Os principais diferenciais do AW são: os recursos para cooperação, a adaptabilidade, a interoperabilidade tecnológica e a verificação de erros. Também se destacam na ferramenta as funcionalidades: várias visões sincronizadas, interoperabilidade do modelo, interoperabilidade da representação, exportação, importação e gerenciamento do repositório. Um ponto fraco do modelo conceitual do AW em relação aos demais é o

fato do mesmo conter poucos recursos de simulação. A tabela de descrição do modelo conceitual do AW está no capítulo 5.6.

Um contexto que se adequaria bem às características do AW seria um ambiente distribuído, onde *workflow* designers com diferentes características e níveis de conhecimento (da ferramenta e do modelo de *workflow*) precisariam definir cooperativamente um processo de *workflow* dentro dos padrões da WfMC. Uma limitação importante deste contexto é que a necessidade de simulação dos processos modelados deve ser baixa. Maiores informações sobre o contexto do AW são mostradas no capítulo 5.2.3.

Cabe salientar que nem todos itens do modelo conceitual do AW foram implementados no protótipo AW e que neste capítulo foi comparado o modelo conceitual do AW com os modelos conceituais de sistemas existentes. Os resultados e os parâmetros desta comparação seriam diferentes de uma comparação do protótipo AW com os próprios sistemas existentes. Por ser um protótipo, o AW seria inferior aos sistemas comerciais existentes em requisitos como robustez, interface, existência de erros, itens não implementados, etc. Porém, o foco desta dissertação é o modelo conceitual e uma análise comparativa entre sistemas está fora do escopo desta pesquisa.

## 8 CONCLUSÃO

Visando uma padronização da tecnologia de workflow, a WfMC recomenda que todos sistemas desta área possuam uma ferramenta de definição de processos de workflow ou simplesmente ferramentas de definição.

Entretanto, estas ferramentas divergem significativamente de produto para produto com respeito aos conceitos de workflow que podem ser representados. Além do mais, com base nestes conceitos podem ser disponibilizadas diversas funcionalidades e para cada tipo de funcionalidade existem várias alternativas de interações com os usuários. Toda esta diversidade de possibilidades cria um cenário que dificulta a padronização e avaliação deste tipo de ferramenta. Podem ser encontrados desde produtos que se quer seguem os paradigmas mínimos da modelagem de processos até ferramentas altamente especializadas e adaptadas aos mais diferentes tipos de workflows.

Visando facilitar o trabalho de desenvolvimento e análise destes softwares, a presente dissertação teve como foco um estudo sobre o modelo conceitual de ferramentas de definição.

A partir deste estudo foram fornecidos um protótipo, dois métodos e várias informações que poderão auxiliar desenvolvedores e usuários de ferramentas de definição a compreender, descrever e avaliar os conceitos, funcionalidades e interações deste tipo de ferramenta.

Mais detalhadamente foram contribuições desta dissertação os seguintes itens:

- o levantamento de informações sobre modelos conceituais de ferramentas de definição. Este levantamento foi mostrado no capítulo 4;
- a proposta de um método de descrição de modelos conceituais de ferramentas de definição. O método de descrição foi apresentado no capítulo 4 e está concisamente descrito na tabela 4.1. O método também foi utilizado nos capítulos 5 e 7 para descrever os modelos conceituais do AW e de três ferramentas existentes;
- o modelo conceitual AW. Este modelo conceitual de ferramenta de definição foi descrito no capítulo 5 e avaliado no capítulo 7. O modelo conceitual AW contém os requisitos básicos necessários para este tipo de ferramenta. Seus principais diferenciais são os recursos para cooperação, a adaptabilidade, a interoperabilidade tecnológica e a verificação de erros;
- a representação AW. Esta representação de definições de processo foi descrita no capítulo 6, utilizada no modelo conceitual AW e implementada no protótipo AW;

- o protótipo AW. O protótipo é um *software* livre que implementa a maioria das funcionalidades, conceitos e interações do modelo conceitual AW. Este protótipo está disponível para uso em (TELECKEN, 2004);
- a apresentação de um panorama geral das funcionalidades, conceitos e interações das principais ferramentas de definição existentes. Tal panorama é mostrado no capítulo 7;
- a proposta de um método de avaliação de modelos conceituais de ferramentas de definição. Este método é proposto no capítulo 7. Ainda no capítulo 7 o método de avaliação é utilizado para avaliar o modelo conceitual AW e os modelos conceituais das ferramentas *Foro model designer*, *Ilog workflow modeler* e *Ultimus designer*.

## 8.1 Trabalhos futuros

Os resultados desta pesquisa abriram perspectivas para a realização dos seguintes trabalhos futuros:

- a proposição de métodos de avaliação e descrição de modelos conceituais específicos para certos contextos como, por exemplo, o ensino a distância, o comércio eletrônico e aplicações ligadas à engenharia. Tais propostas seriam especializações das propostas feitas nesta dissertação;
- a proposição de modelos conceituais de ferramentas também especializados para determinados contextos;
- a adição de funcionalidades, interações e conceitos no protótipo AW. Inicialmente serão adicionados os itens previstos em seu modelo conceitual e ainda não implementados. Também se pretende adicionar novos módulos que irão explorar os principais diferenciais do protótipo AW que são os recursos para cooperação, a adaptabilidade, a interoperabilidade tecnológica e a verificação de erros;
- a utilização do protótipo AW e do seu ambiente de desenvolvimento como uma plataforma de testes para pesquisas sobre workflow, estruturação de documentos e desenvolvimento de software livre;
- a aplicação do protótipo AW em ambientes reais;
- a formalização da representação AW do modelo AW e da compatibilidade de ambos com o modelo e a representação da WfMC;

E, por fim, numa área tão dinâmica, recomenda-se o acompanhamento das tendências de padronização de *workflows*. Tal acompanhamento ajudará a manter os métodos, sistemas e modelos propostos nesta dissertação sempre atualizados.

## 8.2 Publicações

Também foram resultados desta dissertação as seguintes publicações:

PINHEIRO, M. K.; TELECKEN, T.L.; VALDENI, J.; ZEVE, C.D.; EDELWEIS, N. A Cooperative Environment for E-Learning Authoring. **Document Numérique**, França, v.5, n. 3-4, p. 89-114, 2002.

TELECKEN, T.L.; VALDENI, J.; ZEVE, C.D.; MACIEL, C.; BORGES, T. Modeling of Courses through Workflow using the standard SVG/XML. In: WORLD CONFERENCE ON EDUCATIONAL MULTIMEDIA, HIPERMEDIA & TELECOMUNICATIONS, Denver, **Proceedings...** Norfolk: AACE,2002. p24-29.

TELECKENb, T.L.; VALDENI, J.;EDELWEISS, N.; ZEVE, C.D; MACIEL, C.; BORGES, T. ; PINHEIRO, M. **Implementação de um editor de Workflow baseado no padrão SVG.** CPGCC-UFRGS, RS, Brasil, 2002. Porto Alegre: PPGC da UFRGS, 2002. (Relatório Técnico).

TELECKEN, Tiago Lopes; ZEVE, Carlos Dal'col; LIMA, José Valdeni de; PINHEIRO, Manuele Kirsch; EDELWEISS, Nina; BORGES, Tharso; MACIEL, Cassiano. O uso de open source e standards para o ensino a distância dentro do projeto CEMT. Cadernos de informática. v. 2, n. 1, p. 103. Porto Alegre: Gráfica da UFRGS,2002.

## REFERÊNCIAS

AALST, W.M.P. ; HOFSTEDE, A.H.M.; KIEPUSZEWSKI, B.; BARROS, A.P. Advanced Workflow Patterns. In: INTERNATIONAL CONFERENCE ON COOPERATIVE INFORMATION SYSTEMS, CoopIS, 7.,2000. **Proceedings...** Berlin: Springer-Verlag, 2000. p.18-29. (Lecture Notes in Computer Science, v.1901).

AALST, W.M.P. Don't go with the flow: Web services composition standards exposed. **IEEE Intelligent Systems**, New York, v.18,n.1,p.72-76, 2003.

BALZER, R.M.; GOLDMAN, N.; WILE, D. Informality in program specifications. **IEEE Transactions on Software Engineering**, New York, v.SE-4,p.94-103,1978.

BARESI, L. et al. **WIDE Workflow Development Methodology**. Technical Report, 1999. Disponível em: <<http://dis.sema.es/projects/WIDE/Documents/3027-6.pdf>>. Acesso em: 2004.

BPMI. **Business Process Management Initiative Home Page**. Disponível em: <<http://www.bpmi.org>>. Acesso em: 2004.

BRAY, T. et al. **eXtensible Markup Language (XML) 1.0**. 2nd ed.2000. Disponível em: <<http://www.w3.org/TR/REC-xml>>. Acesso em: 2004.

BRAY, T.; HOLLANDER, D.; LAYMAN, A. **Namespace in XML, W3C Recommendations**. 1999. Disponível em: <<http://www.w3.org/TR/REC-xml-names>>. Acesso em: 2004.

CASATI, F. et al. **WIDE Workflow Model and Architecture**. Netherlands: Centre for Telematics and Information Technology (CTIT), University of Twente, 1996. (Technical Report, 96-19).

CASATI, F. et al. Conceptual modeling of workflows. In: INTERNATIONAL CONFERENCE ON CONCEPTUAL MODELING, ER,1995, Gold Cost, Australia. **Conceptual Modeling: proceedings**. Berlin: Springer Verlag, 1995.

CASTROIANNI, C. **Development of a Workflow Design Tool for the Scarabaeus Project**. 1995. Dissertação (Mestrado em informática) - University of Twente, Netherlands.

DEROSE, S.; MALER, E.; ORCHARD, D. **XML Linking Language (XLink) Version 1.0. W3C Recommendation**, 2001. Disponível em: <<http://www.w3.org/TR/xlink/>>. Acesso em: 2004.

DODDS, D. et al. **Professional XML Meta Data**. Birmingham, AL: Wrox press, 2001.

EDER, J.; LIEBHART, W. The Workflow Activity Model WAMO. In: INTERNATIONAL CONFERENCE ON COOPERATIVE INFORMATION SYSTEMS, CoopIS, 3., 1995. **Proceedings...** Viena, Áustria: [s.n.], 1995.

FORO. **Models Designer Manual**. 2001. Disponível em: <<http://www.foro-wf.com/docs/english/ModelDesigner2.1.3.pdf>>. Acesso em: 2004.

GEORGAKOPOULOS, D.; HORNICK, M.; SHETH, A. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. **Distributed and Parallel Databases**, [S.l.], v.3, p.115-153, 1995.

HEUSER, C. A. **Modelagem Conceitual de Sistemas**. 2. ed. Campinas: UNICAMP, 1991. 150p.

IBM. **IBM MQ series Workflow Programming Guide Version 3.3**. Armonk, USA, 2001.

ILOG. **ILOG Home Page** . Disponível em: <<http://www.ilog.com/>>. Acesso em: 2004.

ILOG. **Ilog components for business process management solutions**. Technical Report. 2001. Disponível em: <[http://www.ilog.com/products/jviews/workflow/workflow\\_wp.pdf](http://www.ilog.com/products/jviews/workflow/workflow_wp.pdf)>. Acesso em: 2004.

JOOSTEN, S. Trigger Modeling for Workflow Analysis. In: WORKFLOW MANAGEMENT, CHALLENGES, PARADIGMS AND PRODUCTS, CON, 1994. **Proceedings ...** Viena: R. Oldebourg Muchen, 1994.

JOOSTEN, S.M.M.; BRINKKEMPER, S. Fundamental concepts for workflow automation in practice. In: INTERNATIONAL CONFERENCE INFORMATION

SYSTEMS DEVELOPMENT: METHODS AND TOOLS, THEORY AND PRACTICE, 5., 1996. **Proceedings...** [S.l.]: S. Wrycza&J.Zupancic, 1996.

KENT, W. **Data and Reality**. New York: North Holland, 1978.

KOIVUNEN, M. **Annotea Project**. 2001. Disponível em: <<http://www.w3.org/2001/Annotea/>>. Acesso em: 2004.

LEITE, J. C.; SOUZA, C. S. Uma linguagem de especificação para a engenharia semiótica de interfaces de usuário. In: WORKSHOP SOBRE FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS, IHC,2., 1999, Campinas. **Rompendo barreiras entre pessoas e computadores: anais**. Campinas: Instituto de computação da UNICAMP, 1999. 1 CD-ROM.

LEYMANN, F.; ROLLER, D. **A quick overview of BPEL4WS**. [S.l.]: IBM DeveloperWorks, 2002.

LEYMAN, F. **Web Services Flow Language (WSFL)**. [S.l.]: IBM, 2001.

LOH, S. **Uma linguagem comum entre usuários e analistas para a definição de requisitos de sistemas de informação**. 1991. Dissertação ( Mestrado em Ciência da Computação ) – Instituto de Informática, UFRGS, Porto Alegre.

MCCREADY, S. There is more than one Kind of Workflow Software, **Computerworld**, [S.l.], Nov. 2, 1992.

NICOLAO, M. **Modelagem de Workflow utilizando um Modelo de Dados Temporal Orientado a Objetos com Papéis**. 1998. Dissertação. ( Mestrado em Ciência da Computação ) – Instituto de Informática, UFRGS, Porto Alegre.

OPEN GROUP. **Open Motif Portal**. Disponível em: <<http://www.opengroup.org/openmotif/>> . Acesso em: 2004.

ORACLE. **Oracle Workflow Guide Release 2.6.2**. 2002. Disponível em: <[http://sqlzoo.napier.ac.uk/big/B10501\\_01/workflow.920/a95265/toc.htm](http://sqlzoo.napier.ac.uk/big/B10501_01/workflow.920/a95265/toc.htm)>. Acesso em: 2004.

OTSUKA, J. L. Proposta de um Sistema de Aprendizagem Colaborativa baseado no WWW. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 8., São José dos Campus, SP. **Anais ...** São José dos Campus: SBC, 1997.

PINHEIRO, M. K.; TELECKEN, T.L.; VALDENI, J.; ZEVE, C.D.; EDELWEIS, N. A Cooperative Environment for E-Learning Authoring. **Document Numérique**, França, v.5, n. 3-4, p. 89-114, 2002.

SCHLENOFF, C. et al. **The Process Specification Language (PSL)**: Overview and version 1.0 specification. Gaithersburg, MD: National Institute of Standards and Technology, 2000 (NISTIR6459).

SCHLUMBERGER. **Schlumberger Home Page**. Disponível em: <<http://www.schlumbergersema.com/>>. Acesso em: 2004.

SHETH, A. P.; et al. Report from the NSF Workshop on Workflow and Process Automation in Information Systems. **SIGMOD Record**, New York, v.25, n.4, p.55-67,1996.

SIZILIO, G. R. M. A. **Técnicas de Modelagem de Workflow Aplicadas à Autoria e Execução de Cursos de Ensino à Distância**. 2000. 114 p. Dissertação ( Mestrado em Ciência da Computação ) – Instituto de Informática, UFRGS, Porto Alegre.

SOUZA, C. S. et al. Interação Humano-Computador: Perspectivas Cognitivas e Semióticas. In: JORNADAS DE ATUALIZAÇÃO EM INFORMÁTICA. **Anais...** Rio de Janeiro: [s.n.], 1999. v. 2, p. 425-476

TELECKEN, T.L.; VALDENI, J.; ZEVE, C.D.; MACIEL, C.; BORGES, T. Modeling of Courses through Workflow using the standard SVG/XML. In: WORLD CONFERENCE ON EDUCATIONAL MULTIMEDIA, HIPERMEDIA & TELECOMMUNICATIONS, 2002, Denver. **Proceedings...** Norfolk: AACE,2002. p24-29.

TELECKENb, T.L.; VALDENI, J.;EDELWEISS, N.; ZEVE, C.D; MACIEL, C.; BORGES, T. ; PINHEIRO, M. **Implementação de um editor de Workflow baseado no padrão SVG**. Porto Alegre: PPGC da UFRGS, 2002. (Relatório Técnico).

THATTE, S. **XLANG. Web Services for Business Process Design**. [S.l.]: Microsoft Corporation, 2001. (Technical report).

ULTIMUS. **Ultimus Home Page**. Disponível em: <<http://www.ultimus.com/index>>. Acesso em: 2004.

**ULTIMUS. Understanding Business Process Modeling & Analysis:**The First Step to Achieving Breakthroughs in Productivity. 2003. Disponível em: <<http://www.ultimus.com/ultwhite/BPMA White Paper.pdf>>. Acesso em: 2004.

**W3C. World Wide Web Consortium Home Page.** Disponível em: <<http://www.w3.org/>>. Acesso em: 2004.

**W3Cb. Amaya Home Page.** Disponível em: <<http://www.w3.org/Amaya/>>. Acesso em: 2004.

**W3C. Web Service Choreography Interface 1.0.** 2002. Disponível em: <<http://www.w3.org/TR/wsci/>> Acesso em: 2004.

**W3C. Scalable Vector Graphics (SVG) 1.0 Specification.** W3C recommendation. 2001. Disponível em: <<http://www.w3.org/TR/SVG/>>. Acesso em: 2004.

**W3C. Resource Description Framework (RDF).** W3C Recommendation, 1999. Disponível em: <<http://www.w3.org/TR/REC-rdf-syntax>>. Acesso em: 2004.

**WFMC. Workflow Management Coalition Home Page.** Disponível em: <<http://www.wfmc.org/>>. Acesso em: 2004.

**WFMC. Workflow Management Coalition Interface 1:** process definition interchange process model. [S.l.], 1999.( Technical Report WFMC-TC-1016).

**WFMC. Workflow Management Coalition Terminology & Glossary.**[S.l.], 1996 (WFMC-TC-1011, Document Status- Issue 2.0).

**WIDE. The WIDE Workflow Modeling Language.** Technical Report. 2002. Disponível em: <<http://dis.sema.es/projects/WIDE/Documents/4080-2.pdf>>. Acesso em: 2004.

**ZSCHORNACK, F. Evolução de Esquemas de Workflow Representados em XML.** 2003. 91f. Dissertação ( Mestrado em Ciência da Computação ) – Instituto de Informática, UFRGS, Porto Alegre.