

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

DANIEL DE MELLO VIERO

**Projeto de um Sistema de Gerenciamento
de Workflow Baseado em Padrões Abertos e
de sua Integração com um Ambiente de
Educação à Distância**

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

Prof. Dr. José Palazzo Moreira de Oliveira
Orientador

Porto Alegre, maio de 2005

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Viero, Daniel de Mello

Projeto de um Sistema de Gerenciamento de Workflow Baseado em Padrões Abertos e de sua Integração com um Ambiente de Educação à Distância / Daniel de Mello Viero. – Porto Alegre: PPGC da UFRGS, 2005.

97 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2005. Orientador: José Palazzo Moreira de Oliveira.

1. Gerenciamento de workflow. 2. Educação à distância. 3. Padrões abertos. 4. WfMC. 5. XPDL. 6. AdaptWeb. I. Oliveira, José Palazzo Moreira de. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Pró-Reitor de Coordenação Acadêmica: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Prof^a. Valquíria Linck Bassani

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

Coordenador do PPGC: Prof. Flávio Rech Wagner

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

AGRADECIMENTOS

Agradeço, em primeiro lugar, à Karla, que no momento em que eu ingressei no Mestrado era apenas uma amiga e, durante o período em que estive cursando, virou namorada, noiva e, hoje, é minha esposa. Desde o começo ela compreendeu meus momentos de ausência em função do estudo e me incentivou muito, com o próprio exemplo e muito carinho, a continuar até o fim. Para ela dirijo este agradecimento especial.

Agradeço muito também aos meus pais e meus irmãos, por todo o carinho, incentivo e, em especial, educação que recebo desde criança e que hoje me permite estar concluindo um curso de Mestrado em uma instituição de alto nível como a Universidade Federal do Rio Grande do Sul, onde também tive a oportunidade de fazer o curso de Graduação.

Minha gratidão também à iProcess Soluções em Tecnologia, empresa onde trabalho desde 2001, pelas oportunidades que me permitiram adquirir uma bagagem de experiência que me auxiliou bastante neste trabalho.

Gostaria de agradecer ainda ao meu orientador, prof. Palazzo, que foi compreensivo, permitiu que eu seguisse o meu ritmo de trabalho e sempre acreditou que eu iria terminar com sucesso esta empreitada.

Finalmente, agradeço ao Programa de Pós-Graduação em Computação da UFRGS, pela formação que me proporciona e pela compreensão ao conceder o prazo adicional que me foi essencial para concluir a Dissertação.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	6
LISTA DE FIGURAS	7
LISTA DE QUADROS	8
LISTA DE CÓDIGOS-FONTE	9
RESUMO	10
ABSTRACT	11
1 INTRODUÇÃO	12
1.1 Motivação e Objetivos	13
1.2 Requisitos	14
1.3 Organização do Trabalho	15
2 PADRÕES PARA A ÁREA DE WORKFLOW	16
2.1 Histórico	16
2.1.1 Workflow Management Coalition (WfMC)	16
2.1.2 BPM e Serviços Web	17
2.1.3 “Workflow Patterns”	18
2.1.4 Análise	19
2.2 Conceitos e Definições	20
2.3 Modelo de Referência da WfMC	21
3 MODELAGEM DO WORKFLOW DE UM CURSO	24
3.1 Projeto AdaptWeb	24
3.1.1 Ambiente de Software	25
3.1.2 Arquitetura	25
3.1.3 Exemplo: Curso de Computação Algébrica e Numérica	27
3.2 Modelagem de Workflow do Curso	28
3.2.1 Definição do Escopo do Processo	28
3.2.2 Definição do Fluxo	29
3.2.3 Uso de Outras Construções de Workflow	34
3.3 Descrição do Workflow do Curso em XPDL	34
3.3.1 Características Específicas da XPDL	34
3.3.2 Montagem do XPDL	36

4	SISTEMA DE GERENCIAMENTO DE WORKFLOW	44
4.1	Ambiente de Software	45
4.2	Arquitetura	46
4.3	Funcionamento dos Componentes do Sistema	47
4.3.1	Motor de Workflow	47
4.3.2	Gerenciamento de Definições de Processos	61
4.3.3	Manipulação de Listas de Trabalho	63
4.3.4	Gerenciamento de Aplicações	65
4.3.5	Gerenciamento de Entidades	67
4.3.6	Gerenciamento e Modelo de Segurança	68
4.4	Aspectos de Implementação do Sistema	70
4.4.1	Escopo da Implementação Inicial	70
4.4.2	Componentes de Software	73
4.4.3	Manipulação de Tipos de Dados	74
4.5	Estágio de Desenvolvimento Atual	74
5	INTEGRAÇÃO COM O AMBIENTE ADAPTWEB	76
5.1	Modo Dirigido de Aprendizado	76
5.1.1	Disparo do Processo	78
5.1.2	Lista de Tarefas	78
5.1.3	Acesso ao Material Instrucional	80
5.1.4	Histórico de Tarefas	81
5.1.5	Aspectos de Adaptabilidade no Modo Dirigido	82
5.2	Possibilidades de Extensão do Modelo de Autoria do AdaptWeb	82
5.3	Adequação do Ambiente Tecnológico	84
5.3.1	Cadastro de Usuários do Workflow	85
6	CONCLUSÃO	86
6.1	Resultados e Contribuições	86
6.2	Trabalhos Futuros	87
	REFERÊNCIAS	88
	APÊNDICE A WORKFLOW DO CURSO EM XPDL	91

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i> ("Interface de Programação de Aplicativo")
BPM	<i>Business Process Management</i> ("Gerenciamento de Processos de Negócios")
BPMS	<i>Business Process Management System</i> ("Sistema de Gerenciamento de Processos de Negócios")
DOM	<i>Document Object Model</i> , interface definida pela W3C
EAI	<i>Enterprise Application Integration</i> ("Integração de Aplicações Corporativas")
HTTP	<i>Hypertext Transfer Protocol</i> ("Protocolo de Transferência de Hipertexto")
PHP	<i>PHP, Hypertext Preprocessor</i> , linguagem de programação para aplicações Web
SQL	<i>Structured Query Language</i> , linguagem estruturada de consulta a bases de dados relacionais
WAPI	<i>Workflow Application Program Interface and Interchange Formats</i> ("Interface de Programação de Aplicativo de Workflow e Formatos de Intercâmbio"), definida pela WfMC
WfMC	<i>Workflow Management Coalition</i> , entidade voltada ao gerenciamento de workflow
WFMS	<i>Workflow Management System</i> ("Sistema de Gerenciamento de Workflow")
WP	<i>Workflow Pattern</i> ("Padrão de Workflow")
W3C	<i>World Wide Web Consortium</i> , entidade voltada a padrões para Web
XML	<i>Extensible Markup Language</i> ("Linguagem de Marcação Extensível"), definida pela W3C
XPDL	<i>XML Process Definition Language</i> ("Linguagem de Definição de Processos em XML"), definida pela WfMC

LISTA DE FIGURAS

Figura 2.1:	Conceitos da WfMC	21
Figura 2.2:	Modelo de Referência da WfMC	22
Figura 3.1:	Arquitetura do AdaptWeb	26
Figura 3.2:	Diagrama de Tarefas para os Cursos de Computação e Engenharia . .	30
Figura 3.3:	Diagrama de Tarefas para o Curso de Matemática	31
Figura 3.4:	Diagrama de Tarefas com Transição entre Ramos Paralelos	32
Figura 3.5:	Diagrama de Tarefas com Cruzamento de Transições	32
Figura 3.6:	Diagrama de Tarefas na Visão da XPDL	35
Figura 3.7:	Hierarquia do Processo Aberto na Ferramenta JaWE	43
Figura 4.1:	Arquitetura do Sistema de Gerenciamento de Workflow	46
Figura 4.2:	Diagrama de Estados de Instâncias de Processos	48
Figura 4.3:	Diagrama de Estados de Instâncias de Atividades	52
Figura 4.4:	Diagrama de Estados de Itens de Trabalho	56
Figura 4.5:	Atividade com Múltiplas Transições de Entrada	59
Figura 5.1:	Arquitetura do AdaptWeb com os Componentes de Workflow	77
Figura 5.2:	Protótipo da tela da Lista de Tarefas no AdaptWeb	79
Figura 5.3:	Protótipo da tela de detalhe da tarefa no AdaptWeb	80
Figura 5.4:	Protótipo da tela de histórico das tarefas no AdaptWeb	81

LISTA DE QUADROS

Quadro 3.1: Tópicos da disciplina de Comp. Algébrica e Numérica	26
Quadro 3.2: Estrutura da Disciplina de Computação Algébrica e Numérica	28
Quadro 3.3: Descrição das Tarefas do Workflow da Disciplina de Computação Algébrica e Numérica	37
Quadro 4.1: Comportamento do Motor de Workflow quanto às Atividades de Implementação	55
Quadro 4.2: Comportamento do Motor de Workflow quanto às Restrições de Transições de Entrada	59

LISTA DE CÓDIGOS-FONTE

3.1	Cabeçalho e Elemento Raiz do XPDL	36
3.2	Cabeçalho do Pacote XPDL	38
3.3	Elementos <i>ConformanceClass</i> e <i>Script</i> do XPDL	38
3.4	Declaração de Aplicações do XPDL	39
3.5	Início da Declaração do Processo de Workflow	40
3.6	Parâmetros de Entrada do Processo	40
3.7	Declaração da Atividade Referente ao Tópico 1 em XPDL	41
3.8	Declaração das Transições do Processo em XPDL	42
3.9	Declaração das Restrições de Transição do Tópico 2 em XPDL	43

RESUMO

A tecnologia de Workflow não é nova e muitos de seus benefícios já são largamente aproveitados em aplicações administrativas e de gestão de documentos, há bem mais de uma década. Atualmente, entretanto, tem-se buscado ampliar os horizontes da utilização do Workflow para outras aplicações. A área de educação à distância é uma das que tem grande potencial para aproveitar as características de coordenação de processos que o Workflow proporciona.

Este trabalho buscou fornecer ao AdaptWeb – um ambiente de ensino-aprendizagem adaptativo para a Web desenvolvido na plataforma PHP/MySQL – a possibilidade de agregar um novo modo de funcionamento, baseado na tecnologia de Workflow.

Para tanto, foi realizado o projeto de um Sistema de Gerenciamento de Workflow completo, apropriado para ser utilizado por aplicações no ambiente PHP/MySQL, seguindo as recomendações da WfMC, entidade que estabelece os padrões para a área de Workflow.

O trabalho também mostra como a execução de um curso à distância por um aluno no AdaptWeb pode ser mapeada para um processo de Workflow e como pode ser feita a descrição deste processo nos padrões da WfMC.

Por fim, é apresentada uma proposta de extensão ao ambiente AdaptWeb para suportar um novo modo de interação, o *modo dirigido*, baseado no uso do Workflow, bem como idéias para estender o ambiente de autoria e de execução do curso para aproveitar novas perspectivas que o Workflow apresenta.

Palavras-chave: Gerenciamento de workflow, educação à distância, padrões abertos, WfMC, XPDL, AdaptWeb.

Design of a Workflow Management System based on Open Standards and its Integration with a Distance Learning Environment

ABSTRACT

Workflow technology is not new and many administrative and document management applications have been taking benefits from it for more than one decade. Currently, however, usage of workflow is getting wider, since new kinds of applications start adopting the technology. Distance learning is one of those areas that have great potential for getting advantages from process coordination characteristics provided by workflow management systems.

This work intended to providing a new interaction mode, based on workflow technology, to the AdaptWeb system – an adaptive web-based learning environment developed on PHP/MySQL platform.

With this purpose, a Workflow Management System was designed to be used by applications running on PHP/MySQL environments, following WfMC's recommendations. WfMC is the main organization that establishes workflow standards and references.

This work also shows how the execution of a distance learning course by an student in AdaptWeb can be mapped into a workflow process and how the description of such a process can be built using WfMC standards.

Finally, some extensions to the AdaptWeb environment are proposed to support a new interaction mode based on workflow, as well as suggestions are discussed to extend the course authoring and execution environments to take benefits from the new perspective provided by the workflow approach.

Keywords: workflow management, distance learning, open standards, WfMC, XPDL, AdaptWeb.

1 INTRODUÇÃO

A utilização dos recursos da computação e da comunicação via rede, particularmente a Internet, no aprendizado individual e na promoção de cursos à distância já não é mais novidade. Há algum tempo, empresas, universidades, centros de treinamento e muitas outras organizações fazem uso da tecnologia para promover a disseminação de conhecimento, com maior ou menor desenvoltura e diversas abordagens.

Uma das diferenças trazidas pela “sala de aula virtual”, em relação à educação tradicional, é que não é mais necessário, em todas as circunstâncias, que se formem turmas, nem que haja a presença física simultânea de várias pessoas em um mesmo local, para que aconteça o curso e que se processe o aprendizado. É claro que, ainda assim, em função do papel importante que a interação e a comunicação exercem no processo de aquisição de conhecimento, persistem muitas vezes os conceitos de turmas, alunos e instrutores, interagindo através dos meios eletrônicos e, eventualmente, até mesmo promovendo encontros presenciais.

Essas características dão ao aprendiz de um sistema de educação à distância um grau bastante grande de autonomia quanto a seu ritmo de aprendizado. Cada participante do curso “virtual” pode ter a liberdade de comandar o andamento do seu trabalho e acessar o conteúdo do curso da maneira que melhor se adaptar ao seu estilo cognitivo, nível de conhecimento e velocidade de aprendizado, por exemplo.

Apesar dessa liberdade, ou até mesmo por causa dela, o conteúdo dos cursos deve ser organizado com cuidado para evitar que o aluno se sinta desorientado em seu estudo ou que avance para tópicos para os quais ainda não cumpriu todas as etapas preparatórias. O responsável pela autoria do curso tem normalmente a necessidade de pré-estabelecer algumas regras que o aluno precisa seguir para que o aprendizado seja efetivo, como a definição de pré-requisitos e determinação de tarefas de avaliação ou estudos complementares, de caráter obrigatório, opcional ou alternativo.

Um conjunto de regras, requisitos, tarefas e dependências como esse são aspectos típicos que caracterizam processos de *workflow*.

Segundo a *Workflow Management Coalition* – WfMC (1999), entidade que estabelece as definições e os padrões da área, *workflow* é “a automação de um processo de negócio, no todo ou em parte, durante o qual documentos, informações ou tarefas são passados de um participante para outro, de acordo com um conjunto de regras procedurais”. Casati et al. (1995) definem de forma semelhante, dizendo que *workflows* “são atividades envolvendo a execução coordenada de múltiplas tarefas desempenhadas por diferentes entidades para atingir um objetivo comum”.

Podemos enxergar, portanto, o conjunto de tópicos e atividades didáticas de um curso e as relações de seqüenciamento entre estas atividades como um processo, realizado individual e independentemente por cada um dos aprendizes, com interações eventuais entre

eles. Cada um desses processos individuais pode ser mapeado, acompanhado e gerenciado como um processo de *workflow*.

Pode-se considerar, por exemplo, que cada uma das tarefas delegadas ao aluno corresponde a uma atividade de seu processo de *workflow*. O conjunto das regras estabelecidas pelo professor indica as possibilidades de caminho que o aluno tem no fluxo de trabalho, e as suas escolhas determinam as transições que ocorrem entre uma atividade e outra do processo, habilitando a execução de novas atividades do processo por ele ou por outro agente, no momento apropriado.

A utilização de um sistema de gerenciamento de *workflow* para coordenar as tarefas do aluno em um curso à distância pode possibilitar um controle bastante eficaz da realização dessas atividades, simplificando o desenvolvimento das rotinas de controle do sistema e agregando maiores possibilidades de avaliação do perfil de comportamento de cada aluno, a partir da análise das informações armazenadas na meta-base de dados dos processos de *workflow*.

Pensando-se mais à frente, a idéia de se utilizar o conceito de *workflow* para gerenciar o processo de aprendizado de cada aluno pode trazer a possibilidade de que, mediante a coleta e análise de dados de navegação e avaliação, bem como da definição de *workflows* alternativos de aprendizado, cada aluno venha a trabalhar com um *workflow* diferente para o mesmo curso, adaptado ao seu estilo e comportamento.

1.1 Motivação e Objetivos

O AdaptWeb (Ambiente de Ensino-Aprendizagem Adaptativo na Web) (ADAPTWEB, 2004; FREITAS; MARÇAL; GASPARINI; AMARAL; PROENÇA JR.; BRUNETTO; PIMENTA; RIBEIRO; LIMA; OLIVEIRA, 2002) é um sistema hipermídia adaptativo de educação à distância baseado na *web*, que tem como finalidade adaptar o conteúdo, a apresentação e a navegação de acordo com o modelo do usuário (GASPARINI, 2003). O ambiente é composto por quatro módulos distintos:

- *Módulo de Autoria*: permite a estruturação e a organização dos tópicos a serem ministrados nos cursos e seu material de apoio;
- *Módulo de Geração de XML*: converte as informações do primeiro módulo em arquivos XML, para permitir a manipulação do conteúdo do curso pelos demais módulos;
- *Módulo de Adaptação do Conteúdo*: adequa o conteúdo elaborado pelo autor de acordo com o curso e o ambiente de trabalho do aluno;
- *Módulo de Interface Adaptativa*: adapta a navegação e a apresentação da interface *web*, mostrando versões diferentes de acordo com o curso, o modo de navegação escolhido e o conhecimento do usuário.

O Módulo de Interface Adaptativa do AdaptWeb, onde o aluno efetivamente interage com o conteúdo do seu curso e consulta o material de aprendizado, oferece originalmente ao aluno a escolha entre dois modos distintos de navegação:

- *Modo Tutorial*: modo em que a navegação entre os tópicos de estudo é restrita pelos pré-requisitos estabelecidos pelo autor durante a fase de autoria. Neste modo, o aluno só pode visualizar um tópico (correspondente a um novo conceito a ser

aprendido) uma vez que os seus requisitos já tenham sido visualizados. Ficam desabilitados na interface de navegação do aluno aqueles conceitos para os quais ele ainda não cumpriu as etapas predecessoras.

- *Modo Livre*: modo em que o aluno tem liberdade para acessar qualquer conceito previsto para o curso, sem considerar as restrições estabelecidas pelos pré-requisitos definidos pelo autor. Neste caso, o aluno visualiza toda a lista de tópicos do curso, podendo visitar livremente qualquer conceito a qualquer momento durante a navegação.

O modo de navegação livre, ao mesmo tempo em que dá mais liberdade ao aluno, pode fazer com que aqueles aprendizes mais dispersos ou menos acostumados com o ambiente percam o foco do seu aprendizado. O modo tutorial retira do aluno uma parte da responsabilidade de escolher seu trajeto de aprendizado, facilitando a interação e reduzindo a carga cognitiva do processo de aprendizado (SANTOS; PINTO; ROCHA, 1996).

Considerando as vantagens que a tecnologia de *workflow* pode agregar a sistemas de educação à distância, elencadas anteriormente, uma contribuição importante para que o ambiente AdaptWeb fosse ainda mais completo seria estender o modo tutorial, agregando a ele um mecanismo de *workflow* que permitisse fazer uso dessas vantagens, em especial o monitoramento do estilo dos alunos e a possibilidade de trabalhar com processos individualizados e dinâmicos.

Este trabalho foi concebido para colocar em prática essa idéia, ou seja, com o principal objetivo de agregar ao ambiente AdaptWeb um mecanismo de *workflow* que sirva de suporte para um novo modo de utilização baseado nessa tecnologia, propiciando o aproveitamento das vantagens decorrentes da sua adoção.

1.2 Requisitos

O projeto AdaptWeb tem como uma de suas premissas a utilização de software livre e de arquiteturas abertas na sua construção. O sistema foi desenvolvido com base em dois produtos que têm conquistado muitos adeptos na comunidade *web* por sua facilidade de uso, eficiência e por serem software livre: o PHP, linguagem de programação para páginas web dinâmicas, e o MySQL, sistema de gerenciamento de banco de dados relacional. Além disso, o armazenamento do conteúdo dos cursos é feito em XML, facilitando o seu compartilhamento entre os diversos módulos do sistema.

O mecanismo de *Workflow* a ser agregado ao AdaptWeb tem como requisitos, portanto, o enquadramento nessa filosofia aberta (software livre e padrões) e a capacidade de integrar-se como componente de software ao AdaptWeb, mais especificamente ao ambiente PHP com MySQL no qual ele foi desenvolvido.

Para que atendesse ao critério de software livre, o sistema de *workflow* deveria ser disponibilizado com o código-fonte aberto e passível de modificações. Para ser utilizado pelo AdaptWeb, esse sistema de *workflow* deveria ser também desenvolvido na linguagem PHP ou possuir APIs que permitissem que o programa em PHP o invocasse. Por fim, para seguir a regra de padrões abertos, o sistema de *workflow* deveria seguir as recomendações e padrões da WfMC, pelas razões detalhadas mais adiante, no capítulo 2.

Conforme estudo realizado anteriormente (VIERO, 2002), não foi encontrado até o início do trabalho, em 2003, nenhum sistema de gerência de *workflow* disponível de forma aberta na Internet na linguagem PHP. O suporte aos padrões da WfMC não foi encontrado

nem mesmo em sistemas desenvolvidos em outras linguagens, como Java, a não ser em estágio primário de desenvolvimento.

Por isso, tornou-se necessário o projeto e o desenvolvimento de um novo sistema de gerenciamento de *workflow* para ser incorporado ou utilizado pelo ambiente AdaptWeb, que atendesse aos requisitos expostos anteriormente.

A partir da necessidade de se desenvolver um novo sistema de gerenciamento de *workflow*, e visando o melhor aproveitamento do esforço oriundo deste trabalho, definiu-se como contribuição adicional que o mecanismo desenvolvido para o AdaptWeb deve ser de aplicabilidade genérica, podendo ser utilizado posteriormente em outras aplicações PHP com finalidades distintas, e deve também ser extensível, de modo a permitir a adição de novas funcionalidades além dessas necessárias para aplicações de educação à distância.

1.3 Organização do Trabalho

O capítulo 2 faz uma revisão dos principais padrões propostos na área de *workflow*, em especial das recomendações da WfMC, que foram utilizados como base para definição da arquitetura e das interfaces do sistema de *workflow* desenvolvido.

O capítulo 3 explica a estrutura dos cursos no ambiente AdaptWeb, a forma de elaborar um processo de *workflow* a partir do programa de um curso e como será feito o mapeamento desse processo para o padrão de especificação de processo da WfMC.

O capítulo 4 descreve em detalhes o sistema de gerenciamento de *workflow* projetado, especificando em maiores detalhes a arquitetura, o funcionamento dos componentes, os modelos de dados, bem como o escopo e estágio atual da implementação.

O capítulo 5 explica como foi projetada a integração do sistema de gerenciamento de *workflow* ao ambiente AdaptWeb, apresentando um novo modo de interação baseado nesta tecnologia.

Por fim, no capítulo 6, são apresentadas as conclusões do trabalho, suas lacunas e os trabalhos futuros.

2 PADRÕES PARA A ÁREA DE WORKFLOW

As tentativas de padronização para a área de *workflow* não são recentes. Muitas iniciativas, com paradigmas e abordagens muitas vezes divergentes, já foram apresentadas, mas até hoje nenhuma delas foi plenamente aceita, especialmente pelo mercado de fabricantes de software.

Este capítulo apresenta inicialmente, em linhas gerais, um histórico das tentativas de padronização nessa área, explicando por que os padrões da WfMC foram escolhidos para uso neste trabalho. Em seguida, são apresentados em maiores detalhes os conceitos e a arquitetura do Modelo de Referência da WfMC, que serviram de base para o projeto do WFMS resultante deste trabalho.

2.1 Histórico

A tecnologia de *workflow* não é uma das mais novas, quando comparada com aquelas geradas pela explosão de novos conceitos desde a popularização da Internet. O conceito existe há mais de vinte anos. A idéia surgiu como funcionalidade dos sistemas de Gerência Eletrônica de Documentos (GED), para controlar o ciclo de vida da documentação de empresas e instituições, nascendo como um componente integrado em softwares para esse tipo de aplicação.

Posteriormente, a idéia de se ter fluxo de trabalho sem a associação necessária com documentos começou a dar vida própria ao *workflow*, originando os Sistemas de Gerência de *Workflow* “independentes” (*standalone*). Os sistemas de gestão empresarial, como ERPs (*Enterprise Resource Planning*), também passaram a incorporar fluxos de trabalho como forma de distribuir tarefas não apenas entre os usuários do sistema, mas também entre seus diversos módulos e recursos de execução automática.

Com o lançamento constante de novos produtos de GED, *workflow* e ERPs, cada qual implementando a idéia do fluxo de trabalho sob sua própria perspectiva e buscando resolver o mesmo problema sob óticas distintas, criou-se uma verdadeira “Torre de Babel” com relação a terminologia, métodos e ferramentas de modelagem, arquitetura, formas de interação com o usuário e muitos outros aspectos. Em função disso, uma série de organizações acadêmicas e comerciais, especialmente as relacionadas à área de GED, passaram a se interessar por discutir o tema e buscar esse consenso sobre *workflow*.

2.1.1 Workflow Management Coalition (WfMC)

Em 1993, foi criada a *Workflow Management Coalition* (WfMC), um consórcio formado basicamente por fabricantes de software e instituições acadêmicas com o objetivo de “promover e desenvolver o uso de *workflow* através do estabelecimento de padrões para

terminologia, interoperabilidade e conectividade entre produtos de *workflow*” (Workflow Management Coalition, 2004). Entre os membros da WfMC, estão grandes empresas de tecnologia, como Sun, Oracle, FileNet, IBM e Fujitsu. Desde sua criação, a WfMC vem elaborando e publicando uma série de documentos, entre recomendações, modelos e padrões.

Entre as principais publicações da WfMC, destacam-se notadamente *Terminology and Glossary* (Workflow Management Coalition, 1999), que apresenta as terminologias e definições dos principais conceitos relacionados a *workflow*, e *The Workflow Reference Model* (HOLLINGSWORTH, 1995), que estabelece o Modelo de Referência da WfMC, descrevendo a arquitetura, os componentes e as interfaces que fazem parte de um ambiente de execução de *workflow*. Complementando o Modelo de Referência, foram criadas também as especificações das diversas interfaces de um WFMS que ele define, em especial o XPDL (*XML Process Definition Language*) (Workflow Management Coalition, 2002), um dialeto XML para representação da definição dos processos, e o WF-XML (Workflow Management Coalition, 2001), formato de intercâmbio de dados de execução de processos entre motores de *workflow* diversos, igualmente utilizando XML como base.

Os padrões da WfMC foram a única iniciativa relevante neste sentido por vários anos. Apesar desse pioneirismo e de grande parte dos membros da WfMC serem empresas de software para a área corporativa, seus padrões nunca foram plenamente incorporados aos principais Sistemas de Gerência de Workflow do mercado. Os novos produtos continuaram optando por soluções proprietárias ao invés de adotar ou estender as especificações existentes.

2.1.2 BPM e Serviços Web

Mesmo com a relutância em adotar os padrões da WfMC, a idéia da automatização de processos de negócios seguiu ganhando força durante a década de 90 e aos poucos foi se tornando mais difundida e aplicada, com o uso das mais diversas técnicas, ferramentas e abordagens. O investimento em sistemas de ERP contribuiu bastante para esse crescimento e para o fortalecimento da importância do conceito de processo de negócio dentro das organizações.

No momento em que as empresas conseguiram mapear e automatizar seus próprios processos de negócios internamente, juntamente com o fortalecimento dos relacionamentos B2B (*Business-to-Business*) através do uso maciço da Internet, iniciou-se um movimento para que essa automatização de processos se desse também em nível inter-organizacional, ou seja, permitindo que o fluxo de trabalho e dados se estendesse para fora da empresa, conectando processos de organizações diferentes. Embora a proposta da WfMC incluísse uma interface de intercâmbio de definições de processo e uma interface de troca de dados de execução de processos entre motores de workflow distintos, esses padrões foram ignorados e novas iniciativas passaram a surgir.

Essa nova onda deu origem, no final da década de 90, ao conceito de Gerenciamento de Processos de Negócios (BPM - *Business Process Management*), que agrega o conceito de *workflow* – através do controle de fluxo de atividades entre atores humanos e aplicações – à idéia da Integração de Aplicações Corporativas (EAI - *Enterprise Application Integration*) – vislumbrando as interações *business-to-business* como grandes processos de negócios em que cada atividade é realizada em ambientes distintos, coordenados por um BPMS (*BPM System*).

As novas propostas de padronização para BPM passaram a se movimentar em torno das tecnologias e padrões para Internet baseados em HTTP e XML. Mais notadamente,

a tecnologia de Serviços Web (*Web Services*) foi a escolhida pelos vendedores de software para a interconexão de sistemas distribuídos heterogêneos. Os Serviços Web constituíram-se então na base para uma avalanche de novas linguagens, modelos e arquiteturas, cada um passando a ser denominado um novo padrão pelo grupo que o elaborou.

Microsoft, IBM e outras empresas se uniram e propuseram a BPEL4WS (*Business Process Execution Language for Web Services*) (ANDREWS et al., 2003). Essa linguagem define uma notação para especificação do comportamento de processos de negócios baseados em Web Services. Representa a convergência das linguagens XLANG, da Microsoft, e WSFL (*Web Services Flow Language*), da IBM.

A Business Process Management Initiative (BPMI.org), iniciativa composta por empresas como a Intalio, propôs uma série de padrões: uma linguagem de especificação de processos, a BPML (*Business Process Modeling Language*); uma notação gráfica para a representação de processos, a BPMN (*Business Process Modeling Notation*); e uma linguagem de consulta e gerenciamento de processos, a BPQL (*Business Process Query Language*) (BPMI.ORG, 2004).

Uma outra proposta para coordenação de Web Services foi a WSCI (*Web Services Coreography Interface*), apresentada por Sun, BEA, SAP e Intalio (ARKIN et al., 2002). É uma linguagem de descrição de interfaces em XML que descreve o fluxo de mensagens entre Web Services que participam de interações com outros serviços. Esta proposta deu origem a um novo grupo de trabalho no *World Wide Web Consortium* (W3C).

OASIS e UN/CEFACT iniciaram um projeto denominado *Electronic Business XML* (ebXML) para desenvolver um *framework* de utilização consistente do XML para negócios eletrônicos. Um dos produtos desse projeto foi a BPSS (*Business Process Specification Schema*), um esquema XML para definição de processos de negócios (OASIS, 2004).

A quantidade de iniciativas superpostas é espantosa. Cada uma tem sua linguagem de descrição de processos e normalmente também um formato de intercâmbio de dados de execução de processos. Impressiona também o fato de que várias empresas participam simultaneamente de mais de um desses grupos, como é o caso da Sun, da BEA e da Intalio, entre outras, demonstrando que realmente o mercado ainda não sabe exatamente que direção tomar nesta área.

2.1.3 “Workflow Patterns”

O pesquisador holandês Wil van der Aalst se dedica há bastante tempo ao estudo do *workflow*. Preocupado em definir uma abordagem mais estruturada para avaliar as ferramentas e padrões de *workflow* existentes no mercado, ele propôs, em conjunto com outros pesquisadores, a definição dos *Workflow Patterns* (AALST; HOFSTEDE; KIEPUSZEWSKI; BARROS, 2003), uma coleção de construções de controle de fluxo que podem ser tipicamente encontradas em definições de processos, bem como uma documentação de como identificá-los ou implementá-los. O levantamento desses padrões foi feito com base na análise de várias ferramentas e linguagens de *workflow* existentes até então.

Exemplos dos *Workflow Patterns* vão desde estruturas de controle simples, como a “Seqüência” (*Sequence*) – em que uma atividade de um processo está habilitada após o término de outra atividade do mesmo processo – e a “Escolha Exclusiva” (*Exclusive Choice*) – um ponto do processo em que apenas um de diversos caminhos é escolhido, com base em uma decisão ou dados de controle do *workflow* –, até estruturas mais complexas, difíceis de serem encontradas na maioria dos produtos de *workflow*, como a “Es-

colha Tardia” (*Deferred Choice*) – um ponto do processo em que a definição do caminho a seguir não é definida de forma explícita, mas sim por alguma ação do ambiente, o mais tarde possível dentro do processo. Na página web do autor se podem encontrar as descrições de todos os *Workflow Patterns*, inclusive com animações em Flash de cada um deles (Workflow Patterns Homepage, 2005).

A proposta dos autores não é a de que os *Workflow Patterns* sejam padrões, mas sim parâmetros para avaliação da expressividade de linguagens de definição de processos e de capacidade de mecanismos de controle de workflow. Com base nesses parâmetros, Van der Aalst fez diversos estudos em que foram analisadas inúmeras ferramentas comerciais, propostas acadêmicas e padrões baseados em Serviços Web, incluindo os padrões da WfMC (AALST; HOFSTEDE; KIEPUSZEWSKI; BARROS, 2003; AALST; DUMAS; HOFSTEDE, 2003; STAAB; AALST et al., 2003). Segundo esses estudos, as linguagens e implementações analisadas têm poder de expressão insuficiente para representar grande parte dos *Workflow Patterns*.

Um dos grandes motivos para a falta de aceitação dos padrões já propostos, segundo Van Der Aalst, é a ausência de formalismos que determinem uma base sólida para o padrão, como aqueles que determinaram o sucesso da linguagem SQL e do modelo relacional para a área de bancos de dados, amplamente baseados em formalismos tanto para a modelagem quanto para a implementação dos sistemas. Faltaria, segundo ele, uma definição precisa da semântica de cada construção das linguagens de descrição de processos de negócios, tornando muito difícil a implementação de mecanismos de *workflow* consistentes entre si e que pudessem intercambiar processos de negócios sem gerar problemas.

A partir dessa análise, Van der Aalst construiu recentemente sua própria linguagem de workflow, a YAWL – *Yet Another Workflow Language*, seguida posteriormente da implementação de um protótipo (AALST; ALDRED; DUMAS; HOFSTEDE, 2004). Esta linguagem se baseia em Redes de Petri, estendidas para conseguirem representar todos os *Workflow Patterns* definidos, buscando assim aliar a base conceitual com o lado mais pragmático da definição de processos de *workflow*. O autor também ressalta que a YAWL não é a proposta de um novo padrão; foi construída, isto sim, como uma prova de conceitos para o trabalho de sua equipe.

2.1.4 Análise

Considerando-se a situação atual, é bastante difícil saber exatamente como os padrões para a área de *workflow* vão evoluir. Sabe-se, no entanto, que muitas propostas de padrão existem e que nenhuma delas é completa e plenamente aceita, sendo que algumas sequer estão bem definidas.

Os estudos mais confiáveis na área são do pesquisador Wil van der Aalst, já referenciados, e que indicam que as propostas de padrões existentes não têm a expressividade e clareza semântica necessárias para representar claramente os processos de *workflow*. Sua própria proposta, no entanto, não é um ponto de partida para novas iniciativas.

As novas propostas baseadas em Serviços Web foram nitidamente construídas sobre a estrutura de produtos comerciais já existentes e muitas vezes suas direções são motivadas mais por estratégias de *marketing* do que por aspectos realmente técnicos. A multiplicidade de iniciativas, todas ainda imaturas pelo pouco tempo de existência, não habilita nenhuma delas como um ponto de partida sólido para novos trabalhos.

Das opções disponíveis atualmente, aquela que se apresenta como a base mais sólida e com o reconhecimento de maior parte do mercado, especialmente do meio acadêmico,

é o conjunto de padrões da WfMC. Apesar da falha em suportar algumas construções identificadas pelos *Workflow Patterns*, o Modelo de Referência e as interfaces definidas pela WfMC representam o ponto de partida mais confiável e mais próximo do conceito de um padrão aberto para a construção do Sistema de Gerenciamento de Workflow proposto por este trabalho.

As seções a seguir detalham algumas definições estabelecidas pela WfMC e que foram importantes para o trabalho descrito nos capítulos seguintes.

2.2 Conceitos e Definições

Segundo a WfMC, workflow é “a automação de um processo de negócio, no todo ou em parte, durante o qual documentos, informações ou tarefas são passados de um participante para outro, de acordo com um conjunto de regras procedurais” (Workflow Management Coalition, 1999). Casati et al. definem de forma semelhante, dizendo que workflows “são atividades envolvendo a execução coordenada de múltiplas tarefas desempenhadas por diferentes entidades para atingir um objetivo comum” (CASATI et al., 1995). Georgakopoulos define de forma ainda mais simples, considerando que workflow é “uma coleção de tarefas organizadas para realizar determinado processo de negócio” (GEORGAKOPOULOS; HORNICK; SHETH, 1995).

Para a compreensão completa da arquitetura de um sistema de workflow, é necessário conhecer uma série de conceitos. A Figura 2.1, extraída da publicação *WfMC Terminology & Glossary* (Workflow Management Coalition, 1999), ilustra o relacionamento entre os conceitos principais que envolvem a tecnologia de workflow, que também são definidos no mesmo documento. Os principais deles são descritos a seguir.

O conceito mais importante é o de **processo de negócio**. Processo de negócio é o conjunto de um ou mais procedimentos ou atividades relacionados que coletivamente atingem um objetivo de negócios, dentro do contexto de uma estrutura organizacional que define papéis funcionais e relacionamentos. Um processo de negócios pode consistir de atividades automatizadas, passíveis de gerenciamento pelo workflow, e atividades manuais, que fogem ao escopo do sistema de workflow.

A representação de um processo de negócio é estabelecida em uma **definição de processo**, que consiste de uma rede de atividades e seus relacionamentos, critérios para indicar o início e o término do processo e informações sobre cada uma das atividades, como responsáveis, aplicativos e dados associados. Um processo de negócios pode incluir sub-processos, definidos separadamente, mas que também fazem parte da definição de processo.

As **atividades** são unidades de trabalho que representam um passo lógico dentro do processo. Elas podem ser manuais, quando é necessário associá-las a um participante do workflow para que sejam executadas, ou automáticas, quando o sistema de workflow é capaz de realizá-las automaticamente. Um **participante** de workflow é o responsável pela execução total ou parcial de uma determinada instância de atividade. Um participante pode ser uma pessoa, um papel ou um programa. Um **papel** corresponde a uma determinada função dentro do sistema.

O suporte computacional para a execução de um processo de workflow é dado por um **sistema de gerenciamento de workflow** (ou simplesmente WFMS, do inglês *Workflow Management System*). Um WFMS é “um sistema que define, cria e gerencia a execução de workflows através do uso de software, sendo capaz de interpretar definições de processo, interagir com os participantes do workflow e, quando necessário, invocar o uso de

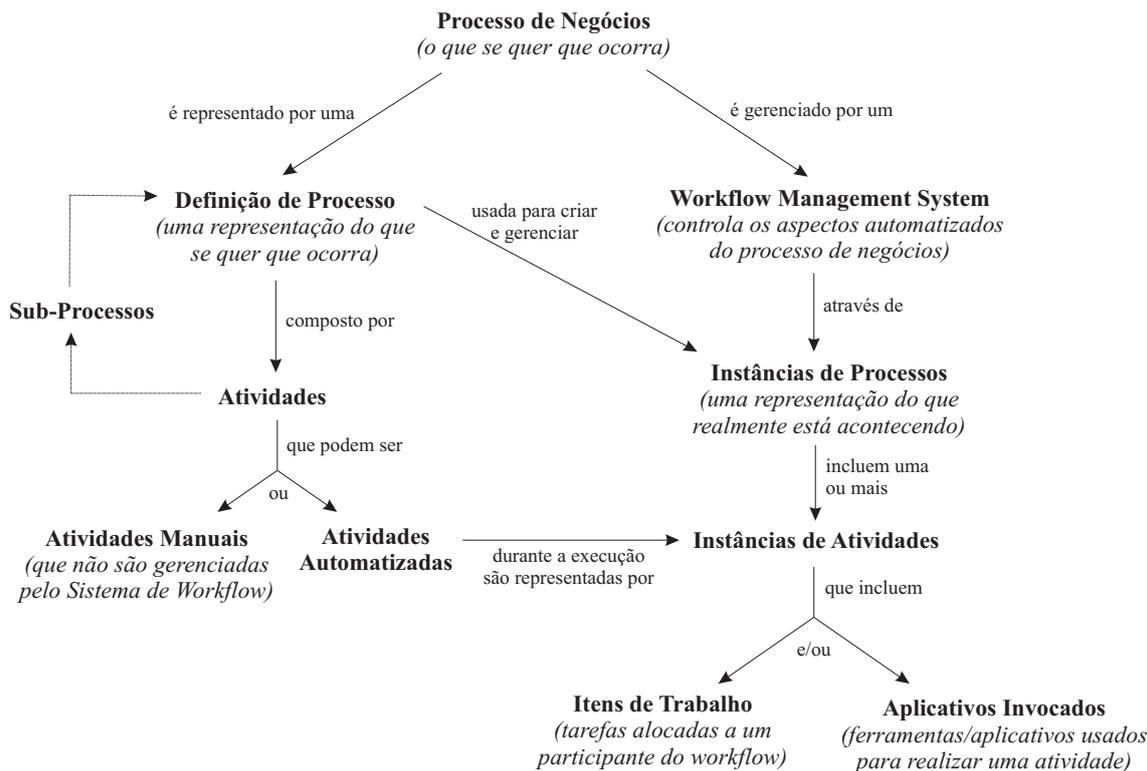


Figura 2.1: Conceitos da WfMC

ferramentas e aplicativos”.

Quando o WFMS inicia a execução de um processo, essa execução específica é denominada uma **instância de processo**. Cada instância possui seus estados internos próprios, é unicamente identificada e processada de forma independente de qualquer outra instância de processo. Para cada uma das atividades que formam a definição do processo, no devido momento, é criada uma **instância de atividade**, que é executada pelo "motor" de workflow, no caso de uma atividade automatizada, ou transformada em um item de trabalho designado a um participante do workflow, no caso de atividades manuais.

2.3 Modelo de Referência da WfMC

A WfMC publicou em 1995 o seu Modelo de Referência de Workflow (HOLLINGSWORTH, 1995). Este Modelo descreve os principais componentes de uma arquitetura de workflow e identifica as interfaces que necessitam ser padronizadas para a permitir a interoperação entre produtos em diversos níveis. A Figura 2.2 ilustra o Modelo de Referência de Workflow.

Esta arquitetura é composta por três tipos de elementos: componentes de software, que proporcionam suporte para várias funcionalidades do sistema de workflow; dados de controle e definição do sistema, que são utilizadas por um ou mais componentes de software; e aplicativos e suas bases de dados, que não são parte do sistema de workflow, mas que podem ser invocados por ele como parte do sistema global.

O primeiro componente de software é a **Ferramenta de Definição de Processos**. Ela é a ferramenta utilizada para criar a definição de processo. A definição de processo, como já visto, contém todas as informações que descrevem o processo de negócios a ser automatizado. A definição de processo pode referenciar um modelo organizacional, que

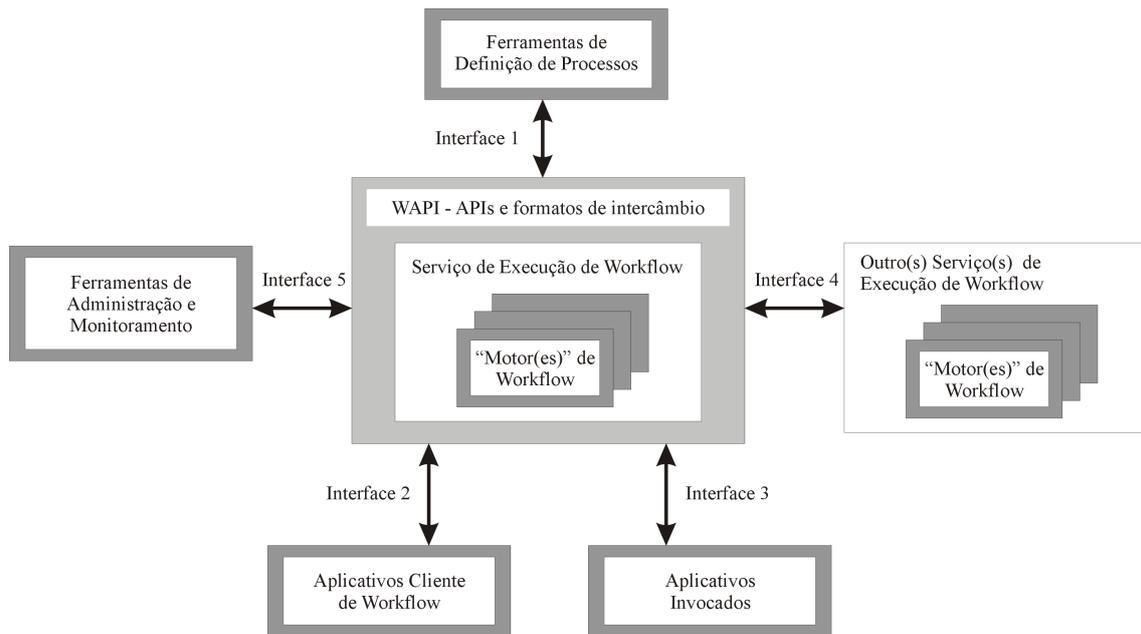


Figura 2.2: Modelo de Referência da WfMC

descreve a estrutura organizacional e os papéis dentro da organização. Em tempo de execução do workflow, essas informações serão utilizadas para determinar os participantes de workflow aos quais as tarefas serão associadas.

O principal componente de software da arquitetura é o **Serviço de Execução do Workflow** (denominação da WfMC para o WFMS). Ele é o responsável pela instanciamento dos processos e seqüenciamento das atividades, adicionando itens de trabalho às listas de trabalho dos participantes e invocando aplicativos quando necessário. Dentro deste serviço encontra-se o “motor” de workflow, módulo que efetivamente controla os processos e suas instâncias. Para executar suas funções, o “motor” de workflow altera e consulta dados de controle do workflow, que incluem informações sobre o estado interno de cada processo ou atividade em execução.

Cada participante do workflow encontrará as tarefas designadas a ele em sua lista de trabalho, com o qual ele interage através de um terceiro componente de software, o **Aplicativo Cliente de Workflow**. Este componente representa a interface com o usuário de um sistema de workflow. Através desse programa, o participante poderá informar que a tarefa foi concluída ou solicitar sua realocação para outro participante. Muitas vezes a funcionalidade desse componente vai além do simples gerenciamento da lista de trabalho, permitindo, por exemplo, disparar novos processos.

Para as tarefas de administração e controle do sistema, como geração de estatísticas, realocação de itens de trabalho, tratamento de erros e exceções e rastreamento de determinada instância de processo, existem funções exclusivas destinadas a um usuário supervisor.

O Modelo define ainda cinco interfaces entre os componentes, além de uma interface sobre o Serviço de Execução de Workflow, denominada WAPI (*Workflow API and Interchange Formats*).

A WAPI consiste de uma série de construções pelas quais os serviços de execução do workflow podem ser acessados. Com isso, os produtos de workflow podem implementar seu funcionamento interno da maneira que desejarem, desde que sejam oferecidos mapeamentos dessas interfaces para os métodos internos do produto.

A interface 1 define o formato de intercâmbio entre as ferramentas de definição de processos e o Serviço de Execução de Workflow, ou seja, estabelece um formato para importação e exportação de definições de processo. Para isso, foi definido um esquema XML que permite a representação de processos de negócio e seus componentes usando este padrão, denominado XPDL (XML Process Definition Language) (Workflow Management Coalition, 2002).

A interface 2 envolve os aplicativos cliente de workflow, que comunicam-se com o “motor” de workflow através do gerenciador de lista de trabalho. Esta interface padroniza as chamadas do gerenciador da lista de trabalho. A interface 3 trata da comunicação com aplicações externas, invocadas pelo Serviço de Execução de Workflow para realização de atividades automatizadas. A padronização envolve a chamada da aplicação pelo workflow, a comunicação durante a execução e a sinalização por parte da aplicação quanto ao término da atividade. As interfaces 2 e 3 são definidas em conjunto através de um padrão denominado WF-API – *Workflow Management API* (Workflow Management Coalition, 1997).

A interface 4 trata da comunicação entre diversos sistemas de gerência de workflow, quando envolvidos na administração de parcelas do mesmo processo. Com essa interface, torna-se possível a execução de um processo através de vários WFMS diferentes. A versão completa do Wf-XML, o padrão para esta interoperabilidade em XML, foi publicado pela WfMC em 2001 (Workflow Management Coalition, 2001).

A interface 5, finalmente, refere-se a ferramentas de administração e monitoramento dos processos de workflow. Através dessa interface, pode-se construir uma ferramenta de administração que funcione sobre qualquer Serviço de Execução de Workflow que implemente este padrão.

3 MODELAGEM DO WORKFLOW DE UM CURSO

O AdaptWeb é o ambiente de educação à distância ao qual este trabalho busca agregar as características da tecnologia de Workflow. Para tanto, é imprescindível que os processos de aprendizado executados neste ambiente possam ser representados como um processo de workflow.

Este capítulo inicia apresentando maiores detalhes sobre o ambiente AdaptWeb, como sua arquitetura, a organização dos cursos e forma de armazenamento do conteúdo dos tópicos de estudo. A seguir, apresenta o exemplo de um curso e discute como se pode modelar um processo de workflow que corresponda à estrutura deste curso. Por fim, é apresentado o processo de transformar o workflow do curso em uma descrição de processo de workflow na linguagem XPDL, que é o padrão proposto pela WfMC.

Não é objetivo deste trabalho definir uma metodologia ou mecanismo automático para traduzir definições de cursos do AdaptWeb para processos de workflow, mas sim integrar ao AdaptWeb o aparato de apoio para a coordenação de um curso por meio de workflow. O objetivo de fazer a modelagem de um curso do AdaptWeb é ilustrar o uso do workflow neste ambiente através de um exemplo prático e real, demonstrando a viabilidade e aplicabilidade da proposta.

3.1 Projeto AdaptWeb

O AdaptWeb (Ambiente de Ensino-Aprendizagem Adaptativo na Web) é um projeto desenvolvido por um consórcio de pesquisa envolvendo a Universidade Federal do Rio Grande do Sul (UFRGS) e a Universidade Estadual de Londrina (UEL), com suporte do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (ADAPTWEB, 2004; FREITAS; MARÇAL; GASPARINI; AMARAL; PROENÇA JR.; BRUNETTO; PIMENTA; RIBEIRO; LIMA; OLIVEIRA, 2002).

O foco do projeto é o desenvolvimento de um ambiente para a autoria e apresentação de cursos à distância na web, com condições de adaptabilidade. A idéia principal é a de permitir que o mesmo conteúdo de um curso seja disponibilizado de formas diferentes para aprendizes com perfis distintos, adaptando-o ao estilo de cada um. O AdaptWeb propõe adaptabilidade em dois níveis: apresentação adaptativa e suporte de navegação adaptativa, permitindo a adaptação no conteúdo a ser apresentado tanto na interface do ambiente da disciplina quanto na navegação, apresentando ou não informações ao usuário com base em seu modelo (GASPARINI, 2003).

Atualmente, em 2005, o ambiente encontra-se operacional e está disponibilizado na Internet no site `sourceforge.net` como um projeto “open source”.

3.1.1 Ambiente de Software

O ambiente AdaptWeb é baseado em um conjunto de ferramentas que suportam as fases de pré-autoria, adaptação de conteúdo e de navegação. As tecnologias utilizadas primordialmente na implementação do ambiente são a linguagem de programação PHP, o gerenciador de bancos de dados MySQL e a linguagem XML.

O PHP é a linguagem de programação para Web utilizada para implementar toda a lógica de apresentação e de negócio do sistema. O MySQL foi utilizado para armazenar a base de dados administrativa e contém informações do modelo do usuário, como o histórico navegacional e conhecimentos adquiridos. A tecnologia XML, graças à sua natureza estruturada e hierárquica, foi utilizada para armazenar todas as informações sobre a estrutura de tópicos das disciplinas.

A escolha dessas tecnologias se deve a dois fatores básicos: o primeiro deles é a simplicidade de aprendizado e utilização das tecnologias, especialmente no caso da linguagem PHP e do banco de dados MySQL; o segundo, e mais importante, é que todas essas tecnologias obedecem a padrões, possuem código-fonte aberto e são gratuitas. Esse último fator é essencial para que o ambiente possa ser utilizado em diversas localidades e, como já mencionado, foi adotado também como premissa para o desenvolvimento do motor de workflow.

3.1.2 Arquitetura

O ambiente AdaptWeb é composto por quatro módulos distintos:

- *Módulo de Autoria*: permite que o autor do curso realize a estruturação e organização dos tópicos a serem ministrados no curso, bem como a definição do seu material de apoio;
- *Módulo de Geração de XML*: converte as informações do primeiro módulo em arquivos XML, para permitir a manipulação do conteúdo do curso pelos demais módulos;
- *Módulo de Adaptação do Conteúdo*: adequa o conteúdo elaborado pelo autor de acordo com o curso e o ambiente de trabalho do aluno;
- *Módulo de Interface Adaptativa*: adapta a navegação e a apresentação da interface web, mostrando versões diferentes de acordo com o curso, o modo de navegação escolhido e o conhecimento do usuário.

A Figura 3.1 ilustra a organização desses componentes.

O conteúdo dos cursos no sistema AdaptWeb é montado no Módulo de Autoria. São cadastradas disciplinas no sistema, que consistem de uma estrutura hierárquica de tópicos, com relacionamentos de dependência entre si (pré-requisitos). Para cada um dos tópicos, o autor pode disponibilizar um conjunto de materiais instrucionais, que podem ser de quatro categorias: conceito, exemplo, exercício e material complementar.

O Quadro 3.1 relaciona uma parte da estrutura de tópicos da disciplina de Computação Algébrica e Numérica, que foi elaborada como estudo de caso para o AdaptWeb durante o trabalho de implementação do Módulo de Autoria (FREITAS, 2003). Este exemplo será utilizado no decorrer deste trabalho como base para elaboração de um processo de workflow de curso.

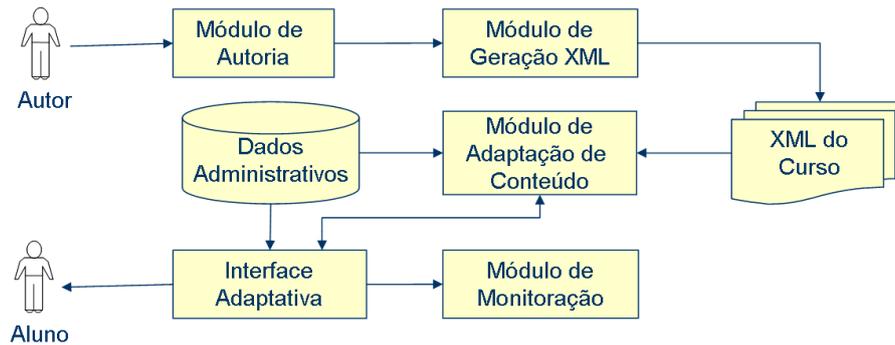


Figura 3.1: Arquitetura do AdaptWeb

- | |
|--|
| <ol style="list-style-type: none"> 1. Contexto e Objetivos da Computação Algébrica e Numérica 2. Sistemas Lineares de Equações Algébricas <ol style="list-style-type: none"> 2.1. Introdução/Motivação 2.2. Métodos Diretos <ol style="list-style-type: none"> 2.2.1. Método de Gauss <ol style="list-style-type: none"> 2.2.1.1. Algoritmo da Triangularização 2.2.1.2. Algoritmo da Retrossubstituição 2.2.2. Método de Gauss com pivotamento <ol style="list-style-type: none"> 2.2.2.1. Condicionamento de Matrizes 2.2.3. Método da Decomposição LU 2.2.4. Método de Cholesky 2.3. Métodos Iterativos <ol style="list-style-type: none"> 2.3.1. Método de Jacobi 2.3.2. Método de Gauss-Seidel |
|--|

Quadro 3.1: Tópicos da disciplina de Comp. Algébrica e Numérica

Tanto os tópicos de uma disciplina quanto os seus materiais instrucionais podem ser aplicáveis somente a determinados cursos. Essa informação pode ser cadastrada no ambiente de autoria. Dessa forma, a estrutura de uma mesma disciplina pode ser apresentada de forma diferente a alunos de cursos distintos, adaptando-se à necessidade de cada curso.

A saída da etapa de autoria é uma série de arquivos XML que descreve essa estrutura, criados pelo Módulo de Geração de XML do AdaptWeb. Um primeiro arquivo XML armazena a lista de tópicos da disciplina e as informações básicas sobre cada um, como seu número, nome, pré-requisitos, palavras-chave relacionadas e os cursos aos quais se aplica. Para cada um dos tópicos, por sua vez, é criado um novo arquivo XML, desta vez detalhando o material instrucional. Este XML conterá informações sobre os conceitos, exercícios, exemplos e material complementar, como o caminho dos arquivos e cursos aos quais se aplicam.

A diversidade dos estudantes é suportada pela criação de um modelo flexível do estudante (denominado “modelo do usuário”) onde, para cada aluno, são armazenadas informações sobre o curso, conhecimento, ambiente de trabalho, preferências e histórico navegacional.

Os documentos XML resultantes da etapa de autoria passam por um processo de filtragem antes de serem apresentados para o aluno. Os filtros acontecem dinamicamente durante a interação do aluno no ambiente e obedecem aos critérios de adaptação repre-

sentados no modelo desse aluno. Dessa forma, um mesmo programa de disciplina pode ser adaptado a diferentes cursos e a diferentes características de alunos.

O Módulo de Interface Adaptativa do AdaptWeb, onde o aluno efetivamente interage com o conteúdo do seu curso e consulta o material de aprendizado, oferece originalmente ao aluno a escolha entre dois modos distintos de navegação:

- *Modo Tutorial:* modo em que a navegação entre os tópicos de estudo é restrita pelos pré-requisitos estabelecidos pelo autor durante a fase de autoria. Neste modo, o aluno só pode visualizar um tópico (correspondente a um novo conceito a ser aprendido) uma vez que os seus requisitos já tenham sido visualizados. Ficam desabilitados na interface de navegação do aluno aqueles conceitos para os quais ele ainda não cumpriu as etapas predecessoras.
- *Modo Livre:* modo em que o aluno tem liberdade para acessar qualquer conceito previsto para o curso, sem considerar as restrições estabelecidas pelos pré-requisitos definidos pelo autor. Neste caso, o aluno visualiza toda a lista de tópicos do curso, podendo visitar livremente qualquer conceito a qualquer momento durante a navegação.

O modo de navegação livre, ao mesmo tempo em que dá mais liberdade ao aluno, pode fazer com que aqueles aprendizes mais dispersivos ou menos acostumados com o ambiente percam o foco do seu aprendizado. O modo tutorial retira do aluno uma parte da responsabilidade de escolher seu trajeto de aprendizado, facilitando a interação e reduzindo a carga cognitiva do processo de aprendizado (SANTOS; PINTO; ROCHA, 1996).

3.1.3 Exemplo: Curso de Computação Algébrica e Numérica

Para que se possa demonstrar a idéia de criar um workflow a partir do modelo de um curso do AdaptWeb, é apresentado a seguir um exemplo de curso criado no ambiente. Posteriormente, será feita a modelagem de um workflow que represente a execução deste curso por parte de um aluno.

Durante a elaboração da sistemática de autoria do ambiente AdaptWeb, foi criado e modelado no sistema um estudo de caso real para a apresentação da disciplina de Computação Algébrica e Numérica da Universidade Estadual de Londrina (FREITAS, 2003). O conteúdo da disciplina foi organizado em sete tópicos principais, cada qual subdividido em outros tópicos. Como o conteúdo do curso ficou bastante extenso, durante a fase de implementação do módulo de adaptação de conteúdo passou-se a trabalhar com um subconjunto dos tópicos da disciplina (GASPARINI, 2003).

Neste trabalho, também será adotado como curso de exemplo essa versão simplificada da disciplina de Computação Algébrica e Numérica. O Quadro 3.1, apresentado anteriormente, relaciona os tópicos da disciplina que formam essa versão.

A disciplina de Computação Algébrica e Numérica foi projetada para ser ministrada para três cursos diferentes: Engenharia, Computação e Matemática. Cada curso, no entanto, necessita de uma visão particular desse conjunto de tópicos, adequada aos seus objetivos. Para que a mesma estrutura de tópicos pudesse ser utilizada para todos os cursos, cada tópico também contém a informação dos cursos para os quais se aplica. Da mesma forma, os materiais instrucionais, como exemplos e exercícios, também possuem indicações dos cursos para os quais terão que ser apresentados.

Conceito	Pré-Req.	Comp	Eng	Mat
1. Contexto e Objetivos da Comp. Alg. e Numérica	-	X	X	X
2. Sistemas Lineares de Equações Algébricas	-	X	X	X
2.1 Introdução	2	X	X	X
2.2 Métodos Diretos	2	X	X	X
2.2.1 Método de Gauss	2.2	X	X	X
2.2.1.1 Algoritmo da Triangularização	2.2.1	X	X	X
2.2.1.2 Algoritmo da Retrosubstituição	2.2.1	X	X	X
2.2.2 Método de Gauss com Pivotamento	2.2 e 2.2.1	X	X	
2.2.2.1 Condicionamento de Matrizes	2.2.2	X	X	
2.2.3 Método da Decomposição LU	2.2	X	X	
2.2.4 Método de Cholesky	2.2 e 2.2.3	X	X	
2.3 Métodos Iterativos	2	X	X	X
2.3.1 Método de Jacobi	2.3	X	X	X
2.3.2 Método de Gauss-Seidel	2.3 e 2.3.1	X	X	

Quadro 3.2: Estrutura da Disciplina de Computação Algébrica e Numérica

O Quadro 3.2 relaciona os pré-requisitos de cada tópico da disciplina e a aplicabilidade do conceito para cada um dos cursos.

Pode-se perceber no quadro, por exemplo, que o tópico 2.2.4 (Método de Cholesky) tem como pré-requisitos os itens 2.2 (Métodos Diretos) e 2.2.3 (Método da Decomposição LU). Além disso, este tópico só fará parte da disciplina para os alunos dos cursos de Computação e de Engenharia. Aos alunos do curso de Matemática não será apresentado este tópico nesta disciplina.

Essas definições se refletem na interface do aluno no ambiente AdaptWeb. O aluno só irá visualizar um tópico na estrutura da disciplina se ele estiver matriculado em um dos cursos para os quais o tópico está disponível. Da mesma forma, se o estudante estiver acessando o ambiente no modo “tutorial”, ele só poderá visualizar o conteúdo do tópico 2.2.4 após ter estudado os conceitos do tópico 2.2 e do tópico 2.2.3.

3.2 Modelagem de Workflow do Curso

Esta seção descreve o trabalho de modelar um processo de workflow para o curso de Computação Algébrica e Numérica apresentado como exemplo. Essa modelagem inclui a identificação do contexto a ser abordado, a definição do escopo do processo e o levantamento de tarefas, transições entre elas, participantes, dados e aplicações envolvidas.

3.2.1 Definição do Escopo do Processo

Como foi visto no capítulo 2, o conceito de workflow está vinculado basicamente à ideia de um processo, ou conjunto de tarefas, que se deseja automatizar ou coordenar. O workflow é uma ferramenta que deve auxiliar ou direcionar o conjunto de participantes do processo a atingir seu objetivo final. Por isso, uma pré-condição para uma correta modelagem de workflow é definir claramente qual o processo que se quer executar.

No caso do ensino à distância, sem pensar especificamente no ambiente AdaptWeb, há algumas abordagens possíveis, considerando-se a existência de várias turmas, alunos, professores e disciplinas.

Do ponto de vista da instituição de ensino, o oferecimento de cada disciplina para uma turma de alunos pode ser considerado um processo de negócio. Neste caso, o objetivo é levar todo um conjunto de alunos, que estudam de forma independente, porém coordenada, a concluir todo o conteúdo da disciplina com aproveitamento satisfatório. A execução deste processo, no caso, envolve todos os alunos que estão matriculados na disciplina, executando suas tarefas em paralelo e com algumas possíveis interações coordenadas pelo workflow.

Outra abordagem possível é considerar que cada aluno, individualmente, executa um processo de cursar uma disciplina. Neste ponto de vista, a meta do processo é que o estudante consiga estudar cada um dos tópicos e atingir os parâmetros de conhecimento definidos para cada um, interagindo com professores e demais alunos quando necessário.

Ambas as abordagens são válidas e passíveis de implementação. A primeira delas é mais complexa, pois envolve um número maior de participantes, de tarefas e coordenação mais elaborada das atividades. A segunda tende a ser mais simples.

Outra possibilidade é uma modelagem conjunta entre as propostas anteriormente. Neste caso, a oferta de uma disciplina pela instituição de ensino pode ser considerada o processo principal. Este processo então seria responsável por coordenar a execução de um conjunto de sub-processos, ou processos filhos, executados concorrentemente, correspondentes à execução individual da disciplina por cada aluno.

A primeira abordagem, assim como a última, só são interessantes se houver o conceito de turmas para cada disciplina, pois consideram que um conjunto de alunos executa, paralelamente, um mesmo programa pré-definido de atividades.

No caso do ambiente AdaptWeb, não existe o conceito de turma. Uma vez que a disciplina tenha sido criada e cadastrada no ambiente de autoria, alunos podem individualmente se matricular e cursar a disciplina. A característica de adaptabilidade do ambiente, inclusive, torna o processo ainda mais individualizado, uma vez que cada aprendiz pode ver a disciplina de forma diferente dependendo do modelo do aluno.

O objetivo de modelar o workflow de um curso neste trabalho é obter o ponto de partida para definir a forma de integração do sistema de workflow desenvolvido com o ambiente AdaptWeb, bem como servir de estudo de caso para avaliar o mecanismo. Sendo assim, embora seja relativamente simples, é mais apropriado para este trabalho usar como exemplo o processo de execução individual de uma disciplina, pois se enquadra na forma de trabalho do ambiente e possibilitará uma avaliação inicial satisfatória do sistema de gerenciamento de workflow.

3.2.2 Definição do Fluxo

3.2.2.1 Participantes, Tarefas e Transições

No processo de aprendizagem de uma disciplina do AdaptWeb, o aluno tem basicamente que estudar uma série de tópicos, obedecendo a algumas regras de precedência definidas pelos pré-requisitos estabelecidos entre os tópicos.

Para aprender cada um dos tópicos, o estudante dispõe de um conjunto de materiais instrucionais elaborados pelo autor do curso. Um desses materiais é o “conceito” referente ao tópico, que sempre estará disponível. Opcionalmente, podem estar disponíveis também exemplos, exercícios e material complementar.

No modo tutorial do AdaptWeb, o que define se um aluno pode visualizar e estudar determinado tópico é o fato de ele ter estudado o conceito de todos os tópicos que são seus pré-requisitos. É necessário apenas que o estudante tenha tomado conhecimento do

conceito desses tópicos, mesmo que outros tipos de material instrucional, como exercícios ou exemplos, estejam disponíveis. Não é feita a exigência de que o aluno tenha consultado os outros tipos de material instrucional para que um tópico seja considerado estudado pelo aluno.

Para transformar este processo de aprendizado em um workflow, cada um dos tópicos que o aluno deve cursar será uma atividade (tarefa) e a relação de pré-requisitos será a base para o estabelecimento das transições entre as tarefas.

Nas descrições de cursos do AdaptWeb não é definida nenhuma interação explícita entre estudante e instrutor, bem como com os demais alunos. Sendo assim, as únicas tarefas serão os tópicos a serem estudados e o aluno será o participante único de cada processo, ao qual serão encaminhadas todas essas tarefas.

Utilizando como exemplo o subconjunto da disciplina de Computação Algébrica e Numérica descrito anteriormente (Quadro 3.2), podemos identificar dois conjuntos de tópicos diferenciados a serem estudados, dependendo do curso ao qual o aluno pertence. Estudantes dos cursos de Computação e Engenharia devem cumprir todos os tópicos listados; já os do curso de Matemática não precisam cursar os tópicos 2.2.2, 2.2.2.1, 2.2.3, 2.2.4 e 2.3.2. Isso faz com que a disciplina exija do aluno fluxos de trabalho diferentes, dependendo do curso em que ele estiver matriculado.

As figuras 3.2 e 3.3 ilustram graficamente, através de uma notação informal, a sequência de tarefas da disciplina para os alunos dos cursos de Computação e Matemática, respectivamente.

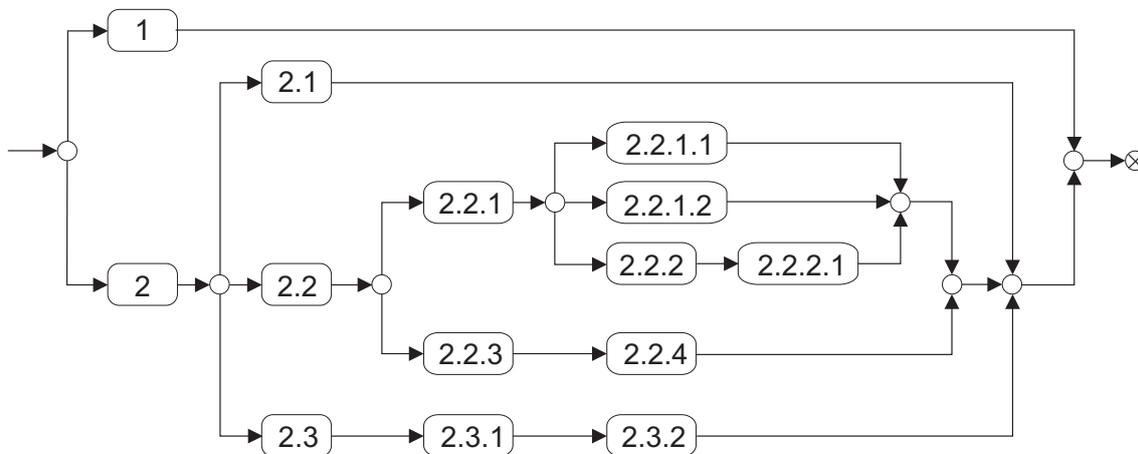


Figura 3.2: Diagrama de Tarefas para os Cursos de Computação e Engenharia

Cada caixa nos diagramas corresponde a uma tarefa a ser executada, ou seja, a um tópico que deve ser estudado pelo aluno. O número dentro da caixa indica qual é o tópico. As setas indicam a relação de precedência entre os tópicos.

Os círculos que aparecem no diagrama representam divisões ou sincronizações no fluxo. Os círculos dos quais *saem* mais de uma seta representam a divisão do fluxo em caminhos paralelos (conceito de *AND-split* ou *fork*). Por exemplo, após a conclusão da tarefa correspondente ao tópico 2, o fluxo divide-se em três ramos, possibilitando ao aluno estudar os tópicos 2.1, 2.2 e 2.3, simultaneamente. Os círculos nos quais *chegam* mais de uma seta representam a sincronização (ou junção) entre ramos paralelos (conceito de *AND-join*). Essa construção serve para fazer com que o fluxo aguarde a conclusão de *todas* as atividades antecessoras para que o fluxo prossiga. O círculo que aparece com um “X” no centro representa o final do workflow.

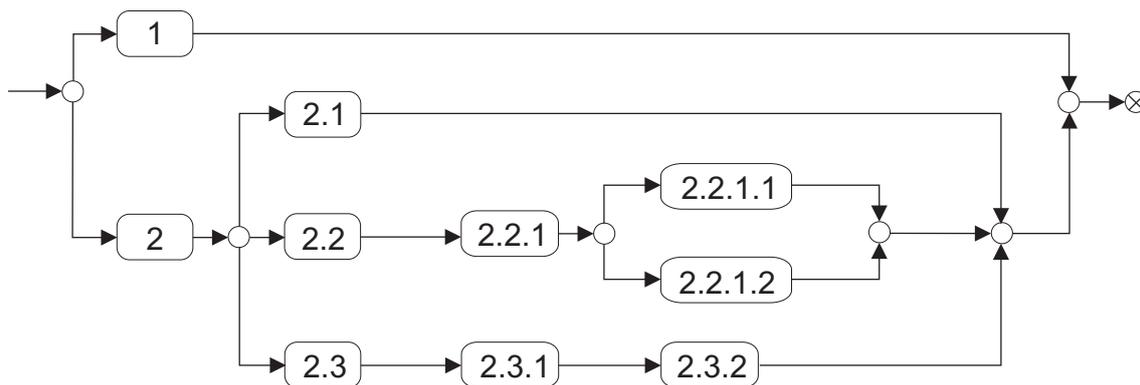


Figura 3.3: Diagrama de Tarefas para o Curso de Matemática

Modelando o workflow da disciplina dessa forma, cada tarefa endereçada ao aluno irá aparecer em sua lista de trabalho como um item. Ou seja, cada tópico que o aluno tiver que estudar será uma tarefa e a sua lista de trabalho será uma “lista de tópicos a estudar”. Se em determinado momento a rede de pré-requisitos permitir ao estudante a possibilidade de estudar mais de um tópico simultaneamente, haverá em sua lista de trabalho uma tarefa em aberto para cada tópico habilitado, podendo o aluno optar por estudar um tópico de cada vez ou alternar entre eles.

Com a utilização do workflow, não bastará mais ao aluno simplesmente visitar a página de conceito de um tópico para que o sistema considere que aquele item já foi estudado e, portanto, libere a visualização dos tópicos que sucedem àquele na hierarquia da disciplina, como acontece atualmente no “Modo Tutorial” da interface do aluno do AdaptWeb. O aprendiz, com o workflow, passará a ter que informar explicitamente ao sistema quando ele considera que o estudo do tópico já foi concluído. Somente dessa maneira ele receberá em sua lista de trabalho os próximos tópicos da disciplina a serem estudados.

A montagem da rede de transições entre as atividades do workflow obedece à ordem de precedência dos tópicos, dada pelos seus pré-requisitos. Para cada pré-requisito, foi criada no diagrama uma transição correspondente indo de um tópico para seu sucessor. Entretanto, é necessária uma certa atenção para evitar complexidade maior que a necessária. Pode-se observar no Quadro 3.2 que, por exemplo, o tópico 2.2.2 possui dois pré-requisitos: os tópicos 2.2 e 2.2.1. No entanto, o item 2.2 já é pré-requisito do próprio item 2.2.1. Ou seja, o requisito 2.2 não é necessário para o tópico 2.2.2, uma vez que esta condição já está estabelecida por transitividade. Na montagem do diagrama, essa transição do item 2.2 para o item 2.2.2 foi suprimida, simplificando o diagrama e evitando uma sincronização a mais, sem comprometer as regras do fluxo.

Como pode ser visto nos diagramas de tarefas, o conjunto de pré-requisitos desta disciplina permite que o fluxo seja desenhado com um aspecto “bem-comportado”, ou seja, para cada divisão (*split*) que ocorre no fluxo, existe uma sincronização (*join*) posterior entre os mesmos ramos do fluxo.

No entanto, o modelo de autoria do AdaptWeb possibilita que sejam criados cursos cujos pré-requisitos não permitam ao fluxo manter esse aspecto “bem-comportado”, sem que isso signifique algum problema ou erro. Se, por exemplo, o item 2.2.3 também fosse pré-requisito do item 2.3.1, o diagrama de tarefas teria uma junção entre dois ramos paralelos que não foram gerados pelo mesmo *split*. A área em destaque na Figura 3.4 ilustra essa situação.

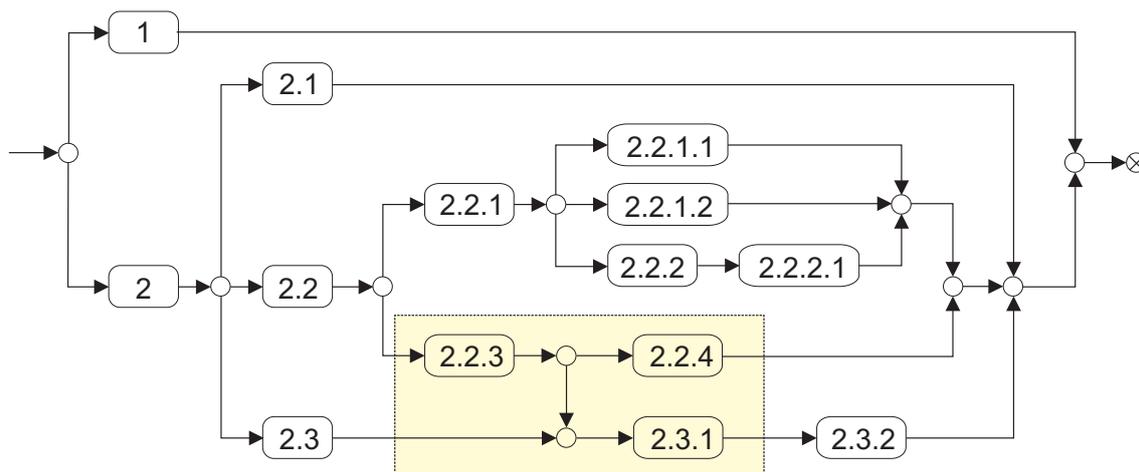


Figura 3.4: Diagrama de Tarefas com Transição entre Ramos Paralelos

Outra situação que pode acontecer, de acordo com a estrutura de tópicos e pré-requisitos criada pelo autor, é uma exigência de pré-requisitos cruzados, fazendo com que o diagrama tenha situações em que as setas se sobreponham obrigatoriamente. A Figura 3.5 a seguir mostra como ficaria o diagrama da disciplina para o curso de Matemática se os tópicos 2.2.1 e 2.3.1 tivessem, ambos, como pré-requisitos os tópicos 2.2 e 2.3.

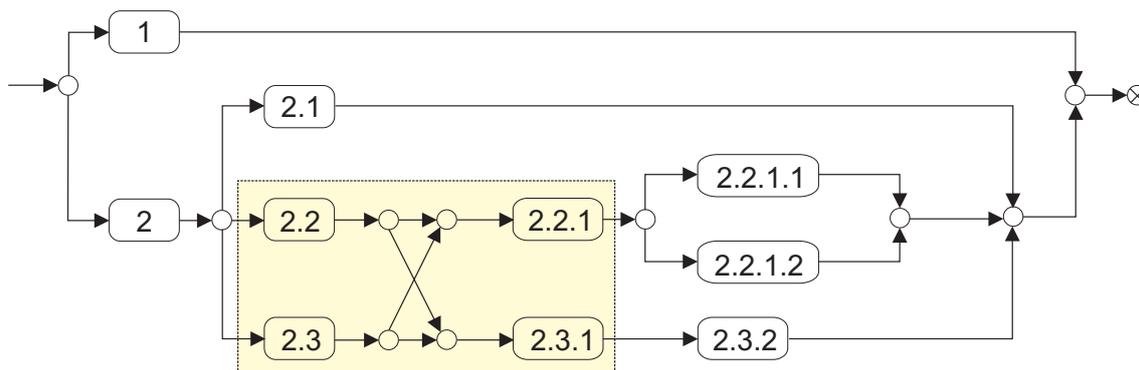


Figura 3.5: Diagrama de Tarefas com Cruzamento de Transições

Essas situações um pouco mais peculiares não são verificadas no caso escolhido como exemplo, mas o sistema de gerenciamento de workflow desenvolvido deve estar preparado para tratá-los, pois qualquer curso desenvolvido poderá apresentar situações que levem a esse tipo de diagrama.

Uma questão mais delicada de se tratar em workflow é o surgimento de ciclos nos diagramas dos processos. O modelo de autoria atual do AdaptWeb não possibilita gerar cursos que dêem origem a diagramas com ciclos, já que não faz sentido que um tópico seja pré-requisito de outro que o preceda. Esse tipo de construção pode vir a ser necessária no futuro, com a possibilidade de agregar novas atividades nos processos de aprendizado, como avaliações, mas no modelo atual essas construções ainda não aparecem.

3.2.2.2 Aplicações Externas

A lista de trabalho de um usuário de workflow deve ser vista como uma referência das tarefas a serem realizadas e ponto de acesso para os dados e aplicações que servem de apoio para a execução das atividades. Por isso, o material instrucional referente a cada

tópico não estará disponível ao aluno diretamente dentro da lista de trabalho. Pode existir, entretanto, uma aplicação externa associada a cada tarefa.

Neste caso, para visualização do material instrucional, essa aplicação externa será na verdade alguma tela do próprio AdaptWeb, gerada pelo módulo de interface adaptativa, na qual é listado o material instrucional do tópico de acordo com os registros do modelo do aluno e registrando as estatísticas de navegação normalmente. A aplicação pode também possibilitar que o aluno navegue pelos tópicos que ele já estudou, para referência.

Portanto, deve ser associada a cada tarefa no workflow uma aplicação externa, que deverá estar devidamente registrada junto ao sistema de gerenciamento de workflow, que dará acesso ao material instrucional do tópico correspondente. Com isso, ao visualizar sua tarefa na lista de trabalho, o aluno poderá acionar a aplicação, abrindo a tela do AdaptWeb que irá trazer a ele todo o conteúdo que ele necessita para cumprir o objetivo da sua atividade, que é estudar o assunto.

3.2.2.3 *Dados do Processo*

Além dos itens já identificados acima, um processo de workflow pode possuir dados de uso interno (ou “dados relevantes do processo”, na terminologia da WfMC), para comunicação com aplicações ou para usar na escolha de transições condicionais.

Como o processo de aprendizado não possui, na forma atual, nenhuma atividade condicional, é pequena a necessidade de variáveis internas do processo. No caso de serem agregadas decisões ou atividades alternativas ao fluxo futuramente, essa necessidade poderá surgir de forma mais forte.

O único dado relevante do processo que se mostra necessário desde já é a identificação do aluno, uma vez que as tarefas serão designadas de forma genérica para um participante “aluno”. Este dado será um parâmetro de entrada do processo, de forma a identificar desde o disparo do fluxo para qual lista de trabalho serão dirigidas as atividades geradas.

3.2.2.4 *Prazos*

Na modelagem de processos de workflow em geral, uma preocupação comum é com a obediência de prazos para a execução das tarefas. Quando a atividade exige, normalmente é estabelecida uma data ou um tempo após o qual, se a tarefa não foi concluída, uma ação é tomada. Esse prazo (“deadline”, na nomenclatura da WfMC) pode ser definido de forma independente para cada atividade e uma transição pode ser seguida caso o prazo máximo seja alcançado. Em um workflow administrativo, por exemplo, se uma tarefa não for concluída em tempo hábil ela pode ser encaminhada para um substituto ou uma notificação enviada ao superior.

No caso do workflow do curso de Computação Algébrica e Numérica, o AdaptWeb não estabelece limites de tempo para que o aluno cumpra os requisitos para cada tópico. Sendo assim, não há necessidade de definir os prazos nem ações a serem tomadas. No futuro, no entanto, esse recurso pode ser aproveitado como forma de identificar que o aluno está necessitando de ajuda ou parou de estudar, por exemplo.

3.2.2.5 *Exceções*

Outra preocupação na modelagem de processos de workflow é com o tratamento de exceções, ou seja, de situações inesperadas durante a execução do processo. Essa questão é especialmente importante quando o processo envolve muitas atividades automáticas, como execução de rotinas, atualizações ou consultas em bancos de dados e invocação de

aplicações externas.

Mais uma vez, essa necessidade não aparece no processo do curso do AdaptWeb, já que se trata de um fluxo com atividades executadas exclusivamente por participantes humanos. Boa parte das questões técnicas que podem surgir normalmente serão tratadas pelo sistema de gerenciamento de workflow, que irá possivelmente encaminhar um aviso de erro para o administrador do sistema. Por exemplo, se é criado um workflow e é informado como parâmetro o *username* incorreto do aluno, ou se o registro desse aluno é excluído da base de alunos sem o cancelamento do workflow, o administrador do sistema deverá intervir pessoalmente, seja cancelando a instância de processo ou alterando algum dado para que o workflow possa prosseguir. Nesses casos, o tratamento não necessita ser modelado no processo em si, já que se trata de questão de infra-estrutura do sistema de workflow.

3.2.3 Uso de Outras Construções de Workflow

Da forma como é permitido atualmente pelo modelo do AdaptWeb, o processo de aprendizado de um aluno em uma disciplina é bastante simples, comparado com processos de workflow típicos, como revisão de documentos, aprovação de compras, atendimento a clientes, entre outros. O workflow da disciplina de Computação Algébrica e Numérica, especificamente, possui apenas um participante (o aluno), as tarefas são todas manuais e bastante parecidas e não existem transições condicionais, ciclos, nem sincronizações elaboradas.

De fato, este é um modelo de processo quase trivial, que não faz uso de todo o potencial de um sistema de gerenciamento de workflow e suas ferramentas de acompanhamento. O uso do workflow possibilita uma ampla gama de construções além dessas que estão sendo utilizadas neste processo.

A escolha deste curso e seu uso como um exemplo para o sistema de workflow permitem avaliar a idéia de integrar o sistema de workflow ao AdaptWeb, como é o objetivo deste trabalho, mas muitas outras possibilidades podem ser abertas.

Ao final deste trabalho, no capítulo 5, são apresentadas possibilidades de extensão ao ambiente AdaptWeb, tanto na parte de autoria quanto da execução do curso, para tirar proveito das potencialidades da tecnologia de workflow no processo de ensino à distância.

3.3 Descrição do Workflow do Curso em XPDL

Nesta seção será detalhada, passo a passo, a montagem do arquivo de descrição do processo de workflow da disciplina de Computação Algébrica e Numérica na linguagem XPDL. Essa é a linguagem de intercâmbio de definições de processo estabelecida pela WfMC e que o sistema de workflow projetado neste trabalho conseguirá interpretar.

Não é objetivo do presente trabalho elaborar um algoritmo ou mecanismo para gerar automaticamente um workflow a partir da representação XML do curso no AdaptWeb. Esse arquivo XPDL, bem como outros utilizados durante os testes, foram gerados manualmente. No entanto, a descrição a seguir pode servir como base para um futuro trabalho visando a criação de um gerador automático de definições de workflow para os cursos do AdaptWeb.

3.3.1 Características Específicas da XPDL

Algumas características específicas da XPDL devem ser levadas em consideração ao se começar a descrever o processo nesta linguagem. Essas características podem levar

a visualizar o diagrama de tarefas elaborado anteriormente de forma mais simplificada, ainda que não se modifique sua semântica.

A primeira característica é que não existem, em XPDL, os conceitos de atividade inicial e de atividade final do fluxo. Para suprir a ausência de uma atividade inicial declarada, a semântica sugerida pela WfMC é de que todas as atividades que não tiverem nenhuma transição de entrada sejam as atividades iniciais do processo e, portanto, sejam iniciadas imediatamente após o disparo do fluxo. Já o término de um processo ou sub-processo deve ser identificado automaticamente pelo motor de workflow. O fluxo deve ser considerado concluído quando não possuir mais nenhuma atividade ativa nem passível de ativação.

Outra característica importante da XPDL é que as construções de *split* e *join* não precisam necessariamente ser atividades separadas de roteamento. As atividades normais, como as que correspondem aos tópicos da disciplina, podem possuir mais de uma transição de entrada ou saída, associando-se ao conjunto de transições as regras que definem o tipo de divisão do fluxo (*AND-split* e *XOR-split*) ou de junção (*AND-Join* ou *XOR-join*). Só é necessário utilizar atividades separadas para roteamento por clareza ou quando realmente não há necessidade de executar nenhuma tarefa entre um *join* e um novo *split*, por exemplo.

Os diagramas elaborados no tópico anterior foram feitos sem pensar exclusivamente nas construções existentes na linguagem XPDL. O fluxo foi descrito de forma a identificar simplesmente a seqüência, dependência e sincronização das atividades.

Em função das características do XPDL, o diagrama de tarefas da disciplina de Computação Algébrica e Numérica pode ser visto de forma um pouco mais simplificada, como ilustrado na Figura 3.6.

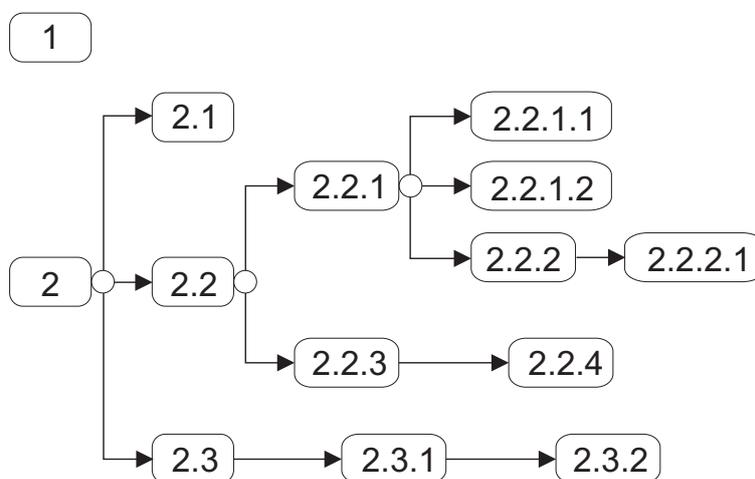


Figura 3.6: Diagrama de Tarefas na Visão da XPDL

Uma das simplificações do diagrama foi a remoção da atividade de início. No diagrama apresentado anteriormente, o fluxo possui uma atividade inicial: um *split* que divide a execução entre os tópicos 1 e 2. Na modelagem XPDL, além de não ser necessário identificar qual a atividade inicial, o próprio *split* deixa de ser necessário, já que a semântica do início de processo indica que as duas atividades que não têm transições de entrada serão instanciadas em paralelo.

Foi também retirada do diagrama a atividade final. Antecedendo ao símbolo de final de fluxo, o diagrama anterior também possui uma série de atividades de sincronização (*joins*), que servem apenas para conectar os ramos paralelos antes do final do fluxo. Todas

essas atividades deixam de ser necessárias dado que o fluxo será considerado encerrado quando não houver mais nenhuma tarefa pendente no processo.

A outra mudança foi que os círculos que representam as divisões do fluxo também deixaram de ser atividades separadas e passaram a estar agregadas às atividades. Isso é possível porque cada *split* possui apenas uma transição de entrada, ou seja, as divisões ocorrem logo após a conclusão de uma única atividade. Em determinados casos isso pode não acontecer. Por exemplo, se duas atividades tiverem os mesmos dois pré-requisitos, será necessário um *join* antes do *split*.

A semântica do processo definido para a disciplina não se altera por essa mudança de “ponto de vista”, mas pelo diagrama fica clara a forma como o fluxo será descrito em XPDL.

Uma questão que é importante ressaltar, no entanto, é que o fluxo poderia ser descrito em XPDL à risca conforme o diagrama especificado anteriormente, ou seja, sem remover as atividades de roteamento e definindo-as separadamente. A simplificação foi feita por não haver influência no comportamento do processo e para tornar mais clara e sucinta a descrição em XPDL. A única simplificação obrigatória é que não há construção explícita para identificar as tarefas inicial e final.

3.3.2 Montagem do XPDL

A base para a montagem da descrição do processo em XPDL é a modelagem feita no item anterior, em que se definiram o diagrama com as tarefas e transições, os participantes, aplicações e dados relevantes do processo, entre outros aspectos. O Quadro 3.3 resume as informações necessárias sobre cada atividade do workflow da disciplina para os cursos de Computação e Engenharia.

A descrição XPDL é um arquivo em XML, sendo portanto composto por uma série de elementos (*tags*), cada qual podendo conter sub-elementos, atributos e/ou conteúdo textual.

Todo arquivo XPDL deve conter a descrição completa de um *pacote* (“package”). Um pacote é uma entidade que pode conter a declaração de vários processos de workflow. Podem existir também no pacote definições de tipos de dados, participantes, aplicações e dados que sejam compartilhados entre todos os processos constantes do pacote.

Por ser um arquivo XML, a definição de processo deve iniciar com a inclusão de um cabeçalho que indica isso e com a definição do elemento raiz. O elemento raiz, no caso do XPDL, é o *Package*, e deve conter a identificação do pacote e o nome do pacote. Normalmente, em arquivos XML, o elemento raiz também contém a declaração dos *namespaces* utilizados no arquivo. A listagem 3.1 mostra este primeiro trecho do arquivo.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<Package xmlns="http://www.wfmc.org/2002/XPDL1.0"
  xmlns:xpdl="http://www.wfmc.org/2002/XPDL1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.wfmc.org/2002/XPDL1.0
    http://wfmc.org/standards/docs/TC-1025_schema_10_xpdl.xsd"
  Id="COMP_ALG_NUM"
  Name="Disciplina de Computação Algébrica e Numérica">
  ...
</Package>
```

Listagem 3.1: Cabeçalho e Elemento Raiz do XPDL

Tópico	Descrição	Particip.	Aplicações	Transições de saída
1	Estudar tópico “1. Contexto e Objetivos da Computação Algébrica e Numérica”	aluno	Consulta ao mat. instrucional	– nenhuma –
2	Estudar tópico “2. Sistemas Lineares de Equações Algébricas”	aluno	Consulta ao mat. instrucional	2.1, 2.2 e 2.3
2.1	Estudar tópico “2.1 Introdução”	aluno	Consulta ao mat. instrucional	– nenhuma –
2.2	Estudar tópico “2.2. Métodos Diretos”	aluno	Consulta ao mat. instrucional	2.2.1 e 2.2.3
2.2.1	Estudar tópico “2.2.1. Método de Gauss”	aluno	Consulta ao mat. instrucional	2.2.1.1, 2.2.1.2 e 2.2.2
2.2.1.1	Estudar tópico “2.2.1.1. Algoritmo da Triangularização”	aluno	Consulta ao mat. instrucional	– nenhuma –
2.2.1.2	Estudar tópic. “2.2.1.2. Algoritmo da Retrossubstituição”	aluno	Consulta ao mat. instrucional	– nenhuma –
2.2.2	Estudar tópico “2.2.2. Método de Gauss com Pivotamento”	aluno	Consulta ao mat. instrucional	2.2.2.1
2.2.2.1	Estudar tópic. “2.2.2.1. Condicionamento de Matrizes”	aluno	Consulta ao mat. instrucional	– nenhuma –
2.2.3	Estudar tópic. “2.2.3. Método da Decomposição LU”	aluno	Consulta ao mat. instrucional	2.2.4
2.2.4	Estudar tópico “2.2.4. Método de Cholesky”	aluno	Consulta ao mat. instrucional	– nenhuma –
2.3	Estudar tópico “2.3. Métodos Iterativos”	aluno	Consulta ao mat. instrucional	2.3.1
2.3.1	Estudar tópico “2.3.1. Método de Jacobi”	aluno	Consulta ao mat. instrucional	2.3.2
2.3.2	Estudar tópico “2.3.2. Método de Gauss-Seidel”	aluno	Consulta ao mat. instrucional	– nenhuma –

Quadro 3.3: Descrição das Tarefas do Workflow da Disciplina de Computação Algébrica e Numérica

Dentro do elemento *Package*, serão declarados todos os demais elementos do arquivo XPD. O elemento *PackageHeader* é obrigatório e possui informações sobre o pacote. O elemento *RedefinableHeader* elenca informações opcionais sobre o pacote, que podem ser redefinidas dentro de cada processo de workflow constante do pacote. Essas, assim como outras declarações referentes ao pacote, são válidas para todos os processos de workflow declarados dentro dele, a não ser que sejam redefinidas localmente. A listagem 3.2 mostra esses dois elementos.

O elemento *ConformanceClass* indica as restrições estruturais às quais os processos do pacote estão submetidas. Esse elemento pode assumir um dos seguintes valores:

- “FULL-BLOCKED”: indica que o fluxo possui apenas uma atividade de início, que cada *split* possui exatamente um *join* correspondente e que não há ciclos no fluxo.

```

<PackageHeader>
  <XPDLVersion>1.0</XPDLVersion>
  <Vendor>Instituto de Informática, UFRGS</Vendor>
  <Created>12/11/2003 4:26:00 PM</Created>
  <Description>Disciplina de Computação Algébrica e Numérica</Description>
</PackageHeader>

<RedefinableHeader PublicationStatus="UNDER_TEST">
  <Author>Daniel de Mello Viero</Author>
  <Version>0.1</Version>
</RedefinableHeader>

```

Listagem 3.2: Cabeçalho do Pacote XPDL

Se não houvesse sido feita a simplificação do diagrama do fluxo, o processo da disciplina de exemplo poderia se enquadrar neste item. No entanto, como citado, qualquer outro curso do AdaptWeb poderia ter transições entre ramos distintos, o que faria com que as restrições deste item não fossem obedecidas.

- “LOOP-BLOCKED”: representa que o fluxo não possui ciclos, mas que também não existe necessariamente correspondência entre *splits* e *joins*. Essa é a situação em que se enquadra o exemplo da disciplina de Computação Algébrica e Numérica e outros fluxos que sejam modelados a partir de cursos do AdaptWeb.
- “NON-BLOCKED”: significa que não há restrições na estrutura do workflow. Essas indicações servem para que o sistema de workflow identifique se possui suporte para controlar corretamente processos com a estrutura descrita. Esta última opção é a padrão, caso o elemento não seja declarado.

Algumas construções no arquivo XPDL permitem que sejam definidas expressões em vez de valores constantes. Essas expressões devem ser escritas em uma linguagem de scripts que seja suportada pelo motor de workflow onde o processo será executado. Na cláusula *Script* do arquivo XPDL, deve ser indicada a linguagem de scripts usada para escrever as expressões nessa definição. Como será visto no capítulo 4, o sistema de workflow desenvolvido dá suporte a expressões escritas na própria linguagem PHP. O valor do elemento para esta linguagem é “text/php”.

```

<ConformanceClass GraphConformance="LOOP-BLOCKED"/>
<Script Type="text/php"/>

```

Listagem 3.3: Elementos *ConformanceClass* e *Script* do XPDL

As aplicações a serem invocadas ou disponibilizadas para o usuário dentro de um processo de workflow devem ser declaradas em um elemento denominado *Applications*, no nível do pacote ou diretamente no nível do processo de workflow. Essa declaração consiste basicamente de uma referência, cabendo ao sistema de gerência de workflow (através de um módulo de gerenciamento de objetos ou aplicações) a real interação com a aplicação ou ferramenta.

No processo de exemplo que está sendo descrito, uma única aplicação será necessária: a tela de consulta de material instrucional do AdaptWeb. Existem duas formas de declarar a aplicação. A primeira delas é através de uma referência externa, como a URL de um *web service*, por exemplo. A outra delas é utilizando um identificador conhecido do sistema de gerenciamento do workflow e indicando quais são os parâmetros que devem

ser passados para a aplicação no momento de sua invocação. Essa segunda abordagem é a mais apropriada para definir a aplicação de consulta ao material instrucional, já que para cada tarefa será necessário passar parâmetros diferentes para a aplicação. A Listagem 3.4 mostra como foi feita a declaração da aplicação.

A aplicação foi declarada com o identificador “APL_MATERIAL_INSTRUCIONAL” e com três parâmetros de entrada: o curso, o tópico e o aluno. Com essas três informações, o sistema pode invocar a tela do AdaptWeb quando solicitado trazendo o material instrucional apropriado para o aluno e para a tarefa em questão. Os parâmetros são definidos no sub-elemento *FormalParameters* da aplicação. Os valores reais a serem passados como parâmetros em cada momento são especificados mais adiante, juntamente com a declaração de cada uma das atividades.

```
<Applications>
  <Application Id="APL_MATERIAL_INSTRUCIONAL"
    Name="Material Instrucional">
    <Description>
      Aplicação que permite visualizar o material instrucional referente ao tópico.
    </Description>
    <FormalParameters>
      <FormalParameter Id="curso" Index="1" Mode="IN">
        <DataType>
          <BasicType Type="STRING"/>
        </DataType>
      </FormalParameter>
      <FormalParameter Id="topico" Index="2" Mode="IN">
        <DataType>
          <BasicType Type="STRING"/>
        </DataType>
      </FormalParameter>
      <FormalParameter Id="aluno" Index="3" Mode="IN">
        <DataType>
          <BasicType Type="STRING"/>
        </DataType>
      </FormalParameter>
    </FormalParameters>
  </Application>
</Applications>
```

Listagem 3.4: Declaração de Aplicações do XPDL

Dentro de um pacote descrito no arquivo XPDL, podem ser declarados vários processos de workflow. Os processos são declarados dentro de um elemento *WorkflowProcesses*, que pode conter vários *WorkflowProcess*. Os processos devem ter um identificador e um nome. Cada processo pode ser independente ou apenas um sub-processo a ser utilizado dentro de algum outro workflow. Essa informação é dada no XPDL pelo atributo *AccessLevel*, que tem o valor “PRIVATE” para os sub-processos e “PUBLIC” para os processos que podem ser disparados diretamente.

A Listagem 3.5 mostra a declaração do processo da disciplina de Computação Algebrica e Numérica para os cursos de Computação e Engenharia. Se fosse desejado, poderia ser descrito no mesmo documento XPDL mais um processo de workflow, para o curso de Matemática.

Conforme definido durante a modelagem, o nome (ou *username*) do aluno que está frequentando o curso deve ser passado como parâmetro para o workflow no momento do seu disparo. Dessa forma, o modelo do processo pode fazer referências genéricas ao aluno, tendo a referência real definida no momento do início do fluxo.

Para declarar em XPDL que o processo de workflow tem um parâmetro, deve ser especificado um elemento *FormalParameter* dentro da especificação do processo, definindo

```

<WorkflowProcesses>
  <WorkflowProcess
    Id="PROC_COMP_ENG"
    Name="Computação Algébrica e Numérica para os Cursos de Computação e Engenharia"
    AccessLevel="PUBLIC">

    <ProcessHeader />

    ...
  </WorkflowProcess>
</WorkflowProcesses>

```

Listagem 3.5: Início da Declaração do Processo de Workflow

o identificador do parâmetro, o modo (entrada e/ou saída) e o tipo de dados. Existe um tipo de dado particular definido na especificação do XPDL que é “PERFORMER”. Este tipo representa que o dado contido no parâmetro deve ser interpretado como um participante do workflow, que pode ser uma pessoa, um papel, uma unidade organizacional ou outra entidade, a ser resolvida pelo sistema de gerenciamento de workflow em tempo de execução do processo. Esse tipo de dado aplica-se também a outras declarações no XPDL.

A Listagem 3.6 mostra que o parâmetro declarado para o processo é do tipo “PERFORMER”, é um parâmetro de entrada e chama-se “PARAM_ALUNO”. Esse parâmetro será referenciado posteriormente na declaração das atividades do workflow, para definição do seu executor.

```

<FormalParameters>
  <FormalParameter Id="PARAM_ALUNO" Index="1" Mode="IN">
    <DataType>
      <BasicType Type="PERFORMER"/>
    </DataType>
  </FormalParameter>
</FormalParameters>

```

Listagem 3.6: Parâmetros de Entrada do Processo

Cada atividade elencada no Quadro 3.3 deve possuir uma declaração dentro do processo de workflow, em um elemento *Activity*. Devem ser definidas nesta seção não apenas as atividades manuais, mas também, se houver, atividades executadas pelo sistema, como invocação automática de aplicações.

Em XPDL, as atividades podem ser basicamente de três tipos:

- **Implementação:** são as atividades mais comuns, executadas pelo sistema ou por um participante especificado. A implementação pode ser subdividida em mais três tipos:
 - *Nenhuma:* o procedimento é manual e deve ser atribuído a um participante.
 - *Ferramenta:* o procedimento é executado através da invocação de uma aplicação declarada no processo, podendo ser disparado automática ou manualmente.
 - *Sub-fluxo:* corresponde à execução de um sub-processo de workflow.
- **Roteamento:** atividade que existe somente para conter a lógica de roteamento através das transições e regras de *split* e *join*;

- *Bloco de Atividades*: executa um conjunto auto-contido de atividades e transições declarada como um conjunto de atividades no processo.

Toda atividade deve ter um identificador, um nome e, opcionalmente, uma descrição. Dependendo do seu tipo, podem conter também a definição de aplicações, modo de início e término (automático ou manual), responsável (*performer*), prioridade, prazo, entre outros.

Na definição de vários desses itens, podem ser usadas expressões em lugar de valores fixos, como já mencionado anteriormente. Essas expressões, no sistema de workflow incorporado ao AdaptWeb, são escritas na linguagem PHP. Como será explicado melhor no Capítulo 4, para referenciar um dado relevante do workflow ou um parâmetro, deve ser utilizado o símbolo \$, seguido do nome da variável. No caso de referenciar o parâmetro *PARAM_ALUNO*, por exemplo, a expressão será “\$PARAM_ALUNO”.

A Listagem 3.7 mostra como ficou a declaração da tarefa correspondente ao tópico 1 da disciplina de Computação Algébrica e Numérica. Nela, nota-se a declaração do elemento *Performer*, que identifica o executor da atividade, usando a expressão que referencia o parâmetro de entrada descrito anteriormente. Os elementos *StartMode* e *FinishMode* estão declarados como “Manual”, indicando que o executor (no caso, o aluno) será o responsável por indicar o início e término da atividade.

```
<Activities>
  <Activity Id="ATIV_TOP_1"
    Name="Contexto e Objetivos da Computação Algébrica e Numérica">
    <Description>
      Estudar o tópico "1. Contexto e Objetivos da Computação Algébrica e
      Numérica"
    </Description>
    <Implementation>
      <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
        <ActualParameters>
          <ActualParameter>CSN</ActualParameter>
          <ActualParameter>1</ActualParameter>
          <ActualParameter>$PARAM_ALUNO</ActualParameter>
        </ActualParameters>
      </Tool>
    </Implementation>
    <Performer>$PARAM_ALUNO</Performer>
    <StartMode>
      <Manual />
    </StartMode>
    <FinishMode>
      <Manual />
    </FinishMode>
  </Activity>
  ...
</Activities>
```

Listagem 3.7: Declaração da Atividade Referente ao Tópico 1 em XPD

O tipo da atividade é identificado pela presença do elemento *Implementation*. Dentro dele, é definido através do elemento *Tool* que a atividade fará uso da aplicação declarada anteriormente, que é a tela do AdaptWeb para acesso ao material instrucional do tópico. Neste momento é feita a declaração dos valores dos parâmetros a serem passados para a aplicação: o código do curso, o número do tópico e a identificação do aluno. Como o modo de início da aplicação está definido como manual, cabe ao executor da tarefa o disparo dessa aplicação, correspondendo assim ao comportamento desejado para o processo

de aprendizado do aluno, que é fazer com que ele acesse o material instrucional quando necessário.

Uma declaração semelhante a essa deverá ser feita no arquivo XML para cada tarefa do processo, conforme o levantamento feito na modelagem. Não há necessidade de relacionar neste momento o código XML de todas as atividades, uma vez que o Apêndice A contém o código completo da descrição de processo em XPDL.

As transições entre as atividades são especificadas no elemento *Transitions*, ainda dentro da declaração do processo de workflow. Cada possível transição corresponde a um subelemento *Transition*, que contém um identificador (opcional), e uma referência para a atividade de origem e outra para a de destino.

O processo da disciplina de Computação Algébrica e Numérica para os cursos de Computação e Engenharia tem as transições descritas na Listagem 3.8, considerando que as demais atividades obedecem ao mesmo esquema de nomenclatura da tarefa referente ao tópico 1, descrita anteriormente.

```
<Transitions>
  <Transition Id="TRANS_2_PARA_21"      From="ATIV_TOP_2"      To="ATIV_TOP_2_1"/>
  <Transition Id="TRANS_2_PARA_22"      From="ATIV_TOP_2"      To="ATIV_TOP_2_2"/>
  <Transition Id="TRANS_2_PARA_23"      From="ATIV_TOP_2"      To="ATIV_TOP_2_3"/>
  <Transition Id="TRANS_22_PARA_221"    From="ATIV_TOP_2_2"    To="ATIV_TOP_2_2_1"/>
  <Transition Id="TRANS_22_PARA_223"    From="ATIV_TOP_2_2"    To="ATIV_TOP_2_2_3"/>
  <Transition Id="TRANS_221_PARA_2211"  From="ATIV_TOP_2_2_1"  To="ATIV_TOP_2_2_1_1"/>
  <Transition Id="TRANS_221_PARA_2212"  From="ATIV_TOP_2_2_1"  To="ATIV_TOP_2_2_1_2"/>
  <Transition Id="TRANS_221_PARA_222"   From="ATIV_TOP_2_2_1"  To="ATIV_TOP_2_2_2"/>
  <Transition Id="TRANS_222_PARA_2221"  From="ATIV_TOP_2_2_2"  To="ATIV_TOP_2_2_2_1"/>
  <Transition Id="TRANS_223_PARA_224"   From="ATIV_TOP_2_2_3"  To="ATIV_TOP_2_2_4"/>
  <Transition Id="TRANS_23_PARA_231"    From="ATIV_TOP_2_3"    To="ATIV_TOP_2_3_1"/>
  <Transition Id="TRANS_231_PARA_232"   From="ATIV_TOP_2_3_1"  To="ATIV_TOP_2_3_2"/>
</Transitions>
```

Listagem 3.8: Declaração das Transições do Processo em XPDL

Se houvesse transições condicionais nesse fluxo, as expressões de condição seriam especificadas neste ponto, junto da transição correspondente. No entanto, não é neste elemento que é especificado o comportamento do fluxo no caso de *joins* e *splits*. Essas definições são feitas através de elementos *TransitionRestriction* dentro da própria definição da atividade.

Por exemplo, ao final da execução da atividade correspondente ao tópico 2 da disciplina, existe uma divisão do fluxo, fazendo com que se abram três novos ramos paralelos, para os tópicos 2.1, 2.2 e 2.3. É necessário declarar ao sistema de workflow que este é um *AND-split*, ou seja, que todas as três transições de saída devem ser seguidas. A Listagem 3.9 mostra como é feita a declaração dessa condição em XPDL para a tarefa correspondente ao tópico 2 do curso. As definições das outras tarefas em que essa situação acontece também contêm o elemento que descreve as restrições de transições.

Com essas construções, o documento XPDL contém a descrição do processo de workflow correspondente à modelagem que foi feita da disciplina de exemplo. O Apêndice A contém o código completo da descrição de processo em XPDL.

Como se pode perceber, nenhuma construção específica do sistema de gerenciamento de workflow ou extensão à linguagem XPDL foi utilizada nesta descrição de processo. Teoricamente, a descrição pode ser lida e compreendida por qualquer ferramenta de definição de processos ou motor de workflow que utilize o padrão XPDL. Por exemplo, a Figura 3.7 ilustra a hierarquia do processo obtida após abrir o arquivo XPDL na ferramenta de código aberto JaWE – Java Workflow Editor (ObjectWeb, 2005).

```

<Activity Id="ATIV_TOP_2"
  Name="Sistemas Lineares de Equações Algébricas">
  <Description>
    Estudar o tópic "2. Sistemas Lineares de Equações Algébricas"
  </Description>
  ...
  <TransitionRestrictions>
    <TransitionRestriction>
      <Split Type="AND">
        <TransitionRefs>
          <TransitionRef Id="TRANS_2_PARA_21"/>
          <TransitionRef Id="TRANS_2_PARA_22"/>
          <TransitionRef Id="TRANS_2_PARA_23"/>
        </TransitionRefs>
      </Split>
    </TransitionRestriction>
  </TransitionRestrictions>
</Activity>

```

Listagem 3.9: Declaração das Restrições de Transição do Tópico 2 em XPDL

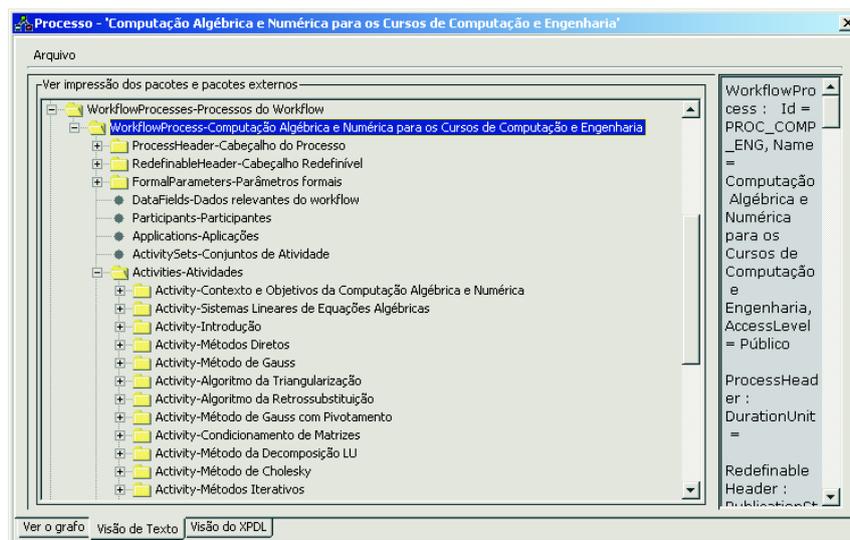


Figura 3.7: Hierarquia do Processo Aberto na Ferramenta JaWE

4 SISTEMA DE GERENCIAMENTO DE WORKFLOW

Um dos objetivos originais deste trabalho é o de agregar ao ambiente AdaptWeb um sistema de workflow, de forma a usar os recursos que essa tecnologia proporciona em prol de um melhor acompanhamento e monitoramento das etapas de aprendizado dos estudantes de cursos à distância.

Naturalmente, o primeiro passo a se dar neste sentido seria buscar um sistema de gerência de workflow já pronto, que pudesse ser agregado ao AdaptWeb, sem a necessidade de haver o retrabalho de desenvolver uma nova solução. Este sistema de workflow, no entanto, deveria se adequar aos princípios e ao ambiente do projeto AdaptWeb. Mais especificamente, deveria ser possível utilizar esse sistema dentro de uma aplicação *web* desenvolvida na linguagem PHP e sobre uma base de dados em MySQL. Além disso, o WFMS deveria ser software livre e utilizar padrões abertos. As recomendações da WfMC foram os padrões considerados mais adequados para este caso, como descrito no Capítulo 2.

Um estudo realizado anteriormente (VIERO, 2002) não encontrou nenhum sistema de gerência de *workflow* disponível de forma aberta na Internet na linguagem PHP até então. O suporte aos padrões mais sólidos na área, as recomendações da WfMC, só foi encontrado em sistemas desenvolvidos em outras linguagens, como Java, e em estágio primário de desenvolvimento.

Por isso, tornou-se necessário o desenvolvimento de um novo WFMS para ser incorporado ou utilizado pelo ambiente AdaptWeb, tendo como linha-mestra esse conjunto de pré-requisitos gerado pelas necessidades do projeto.

A partir da necessidade de desenvolver um novo sistema de gerência de workflow, ampliou-se o horizonte do trabalho, de forma que esse novo WFMS não atendesse apenas ao ambiente AdaptWeb, mas pudesse ser utilizado também por qualquer aplicação *web* compatível com o ambiente de software PHP+MySQL.

As necessidades dos processos a serem modelados nesse ambiente passaram então a ser referência para definir o escopo inicial de implementação do sistema, já que a complexidade de um sistema de gerência de workflow completo não permitiu que se conseguisse desenvolvê-lo por completo até o fim deste trabalho. No entanto, o projeto do sistema foi feito com a visão de um sistema completo, ainda que a implementação inicial seja parcial.

A seqüência deste capítulo descreve o projeto deste novo Sistema de Gerência de Workflow, revisando o ambiente de software e elencando desde as definições de arquitetura até os detalhes de cada componente.

4.1 Ambiente de Software

Uma das premissas do projeto AdaptWeb é a utilização de software livre e de padrões e arquiteturas abertas na sua construção. A interface do sistema é toda em ambiente *web*, o que permite sua utilização através de navegadores nas mais diversas plataformas de software.

O AdaptWeb foi desenvolvido com base em dois produtos que têm conquistado muitos adeptos na comunidade *web* por sua facilidade de uso, estabilidade, eficiência, gratuidade e código aberto: o PHP, linguagem de programação para páginas web dinâmicas, e o MySQL, sistema de gerenciamento de banco de dados relacional. Além disso, o padrão XML foi utilizado para o armazenamento da estrutura dos cursos, permitindo a portabilidade entre os diversos módulos do sistema.

O Sistema de Gerenciamento de Workflow incorporado ao sistema deve se adequar a este ambiente e a esta filosofia.

Para poder ser incorporado ao AdaptWeb, a opção natural é de que o novo Sistema de Gerenciamento de Workflow seja também desenvolvido na linguagem PHP. Na versão mais recente do AdaptWeb até o momento (0.9), a versão do PHP utilizada é a 4.3. O PHP é uma linguagem eminentemente voltada para o desenvolvimento de aplicações *web*, mas é expressiva e poderosa o suficiente para permitir o desenvolvimento de módulos de aplicações que não sejam necessariamente voltados para *web*, como é o caso de um WFMS.

A hipótese de utilizar alguma outra linguagem de programação para o desenvolvimento do WFMS foi descartada, seja pela dificuldade de chamar seus métodos através do PHP – como aconteceria com linguagens como C++ –, seja por agregarem requisitos não desejáveis ao sistema – como seria o caso de Java, em que seria obrigatória a instalação e configuração de uma Máquina Virtual –, considerando-se ainda a complexidade que seria trazida ao ambiente pela construção de um componente em outra linguagem.

Quando ao banco de dados, considerando a necessidade do WFMS armazenar dados de execução de processos e informações de configuração, não existe razão para não utilizar a base de dados relacional MySQL já existente.

Para atender ao critério de software livre, o novo WFMS será disponibilizado à comunidade com o código-fonte aberto e gratuito, sob a licença GNU GPL (*General Public License*), a mesma utilizada pelo AdaptWeb, que dá direito à utilização e modificação do código do software, desde que as modificações também sejam disponibilizadas à comunidade pela mesma licença.

Disponibilizar abertamente o código do sistema permitirá não apenas que muitas outras pessoas utilizem o sistema, mas também que várias delas ofereçam otimizações, correções de problemas, melhorias, novas funcionalidades e toda sorte de contribuições.

O fácil acesso que a Internet permite atualmente a qualquer conteúdo disponível na *web* faz com que, ao se desenvolver um software e disponibilizá-lo livremente, o público não esteja restrito apenas à região ou país onde ele foi desenvolvido. Como um WFMS é um software de uso amplo, aplicável a soluções não dependentes de aspectos regionais, é possível e até mesmo desejável que ele seja utilizado e aprimorado também fora do Brasil. Para permitir que isso seja possível, como será percebido mais adiante neste capítulo, os nomes de variáveis, de métodos das APIs e de objetos do modelo de dados serão em inglês.

4.2 Arquitetura

Como já foi citado, o projeto do WFMS leva em consideração o sistema na íntegra, pensando em todos os seus aspectos, sem se restringir às necessidades imediatas de integração com o ambiente AdaptWeb. Desta forma, quando implementado em sua totalidade, o sistema poderá ser utilizado como um gerenciador de workflow autônomo e incorporado em outras aplicações com exigências mais complexas.

O projeto tem em mente a arquitetura proposta no Modelo de Referência da WfMC. A grande maioria dos conceitos apresentados se baseia nas proposições desse documento, bem como dos demais padrões elaborados pela WfMC, em especial o padrão de intercâmbio de definições de processo (XPDL) e as APIs de interação de clientes e aplicações (WAPI).

A gama de funcionalidades que o Sistema de Gerenciamento de Workflow tem que implementar é bastante grande. Por isso, o sistema foi organizado em uma série de módulos, cada um responsável por implementar algumas funções específicas, oferecendo-as aos demais módulos e a sistemas externos através de APIs.

O diagrama apresentado na Figura 4.1 ilustra de maneira geral a arquitetura do WFMS, apresentando os componentes planejados e como será a interação entre eles.

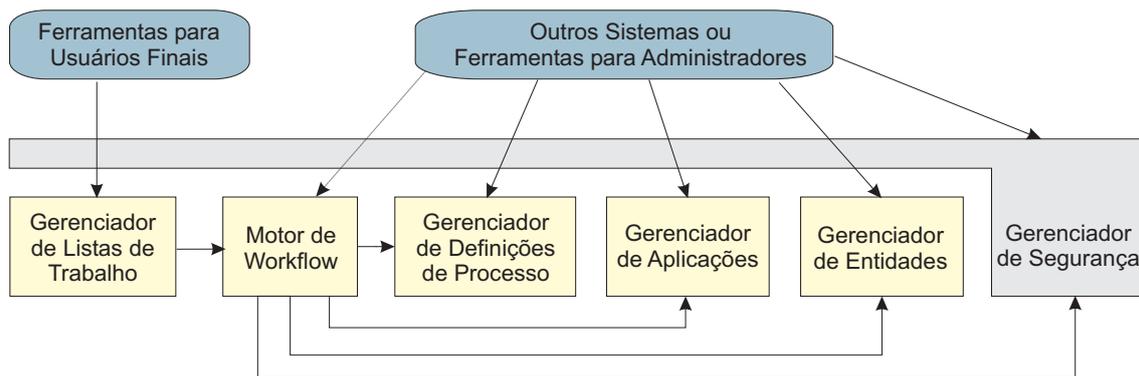


Figura 4.1: Arquitetura do Sistema de Gerenciamento de Workflow

Na figura, podem-se observar dois tipos principais de componentes: na parte superior, as ferramentas de interface com os usuários, e, na parte inferior, os módulos de gerenciamento. Estes últimos são os que efetivamente compõem o WFMS e implementam as funcionalidades necessárias. A função de cada um desses módulos é a seguinte:

- *Motor de Workflow*: implementa toda a lógica de fluxo do processo, incluindo disparo do workflow, distribuição e execução de atividades, avaliação de transições, controle de sincronizações e ciclos, gerenciamento de dados relevantes do processo, tratamento de exceções, entre outras funções.
- *Gerenciador de Lista de Trabalho*: disponibiliza rotinas para consultar a lista de trabalho dos usuários e interagir com elas, encerrando e reencaminhando itens de trabalho ou invocando aplicações, por exemplo. É a base para a construção de aplicações de Lista de Trabalho.
- *Gerenciador de Definições de Processos*: permite que as definições de processo sejam carregadas e armazenadas no sistema, bem como fornece ao Motor de Workflow as informações sobre essas definições de processos. Também é responsável pelo controle de versões das definições.

- *Gerenciador de Aplicações*: implementa o suporte para a execução de diversos tipos de aplicações a partir do workflow. Possibilita o cadastramento das aplicações e configuração da invocação, incluindo forma de chamada e passagem de parâmetros.
- *Gerenciador de Entidades*: oferece acesso aos dados de todos os possíveis participantes de processos ou usuários do sistema, sejam pessoas, papéis, unidades organizacionais, entre outros. Pode possuir um repositório próprio de informações ou ser apenas uma camada de acesso para um outro serviço, como um servidor de diretórios que usa o protocolo LDAP (*Lightweight Directory Access Protocol*), por exemplo.
- *Gerenciador de Segurança*: permite a configuração e o gerenciamento dos aspectos de segurança do sistema, como autenticação, gerenciamento de sessões, registro e verificação de permissões e *logs*. Trabalha cooperativamente com o Gerenciador de Entidades, podendo compartilhar o mesmo repositório de informações, inclusive se for externo.

4.3 Funcionamento dos Componentes do Sistema

Nos próximos tópicos é descrita em detalhes a função de cada componente, a forma de implementação prevista e a sua interação com os demais componentes do sistema.

4.3.1 Motor de Workflow

O Motor de Workflow é o responsável por implementar toda a dinâmica da execução de processos e atividades, que é propriamente o objetivo da tecnologia de workflow. Este é, portanto, o coração ou o cérebro do sistema, é o componente primordial.

Algumas das tarefas do Motor de Workflow são as seguintes:

- interpretar as definições de processos;
- gerenciar a instanciação, execução, suspensão e finalização de processos e atividades;
- criar e distribuir itens de trabalho;
- armazenar e gerenciar dados relevantes de workflow;
- armazenar e fornecer acesso aos dados de execução dos processos;
- definir o momento da execução de aplicações e encaminhar a solicitação ao Gerenciador de Aplicações, gerenciando a passagem de parâmetros entre o processo e a aplicação;
- implementar a lógica de seqüenciamento de atividades, incluindo avaliação de condições, tratamento de *joins*, *splits*, gerenciamento de ciclos e tratamento de exceções;
- controlar o prazo de execução das atividades e executar as ações correspondentes;

A seguir será abordada uma série de tópicos que, no seu conjunto, constituem o funcionamento do Motor de Workflow e explicam como serão implementadas essas tarefas.

4.3.1.1 Ciclo de vida das instâncias de processos

Uma definição de processo é como se fosse o código-fonte de um programa. Por si só, o código-fonte não faz nada, mas expressa todo o comportamento que o programa terá quando for executado. Quando o programa for executado, aí sim, aquilo que está definido é efetivamente realizado. O mesmo programa pode ser executado múltiplas vezes, cada uma recebendo parâmetros e estímulos diferentes do usuário e, portanto, se comportando de forma particular. Analogamente, uma instância de processo de workflow (ou *caso*, como também pode ser denominada) pode ser considerada a execução de uma definição de processo, em que o comportamento definido ganha vida e varia de acordo com as interações que faz com o ambiente e os participantes do workflow.

Desde o momento em que é criada, uma instância de processo de workflow pode passar por diversos estados. A especificação da WfMC define um diagrama de transições de estados para as instâncias de processo, mas não é rígida com relação ao grau de refinamento que a implementação deve suportar. A Figura 4.2 demonstra o diagrama de estados utilizado nesta implementação, que possui leves diferenças, mas é compatível com a recomendação.

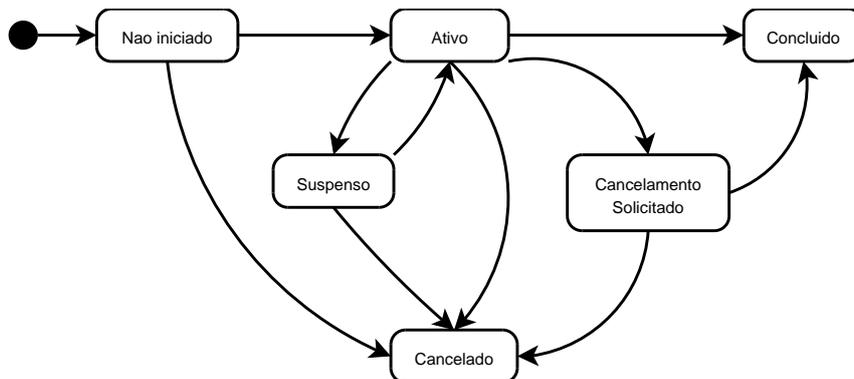


Figura 4.2: Diagrama de Estados de Instâncias de Processos

Resumidamente, o significado de cada um dos estados é o seguinte:

- *Não Iniciado*: a instância de processo foi criada, mas ainda não foi iniciada. Neste estado, podem ser definidos os valores dos parâmetros e dos dados relevantes do processo, antes que ele seja iniciado;
- *Ativo*: o workflow foi iniciado e está em execução;
- *Concluído*: o processo foi concluído normalmente, através da execução completa do fluxo de atividades;
- *Suspenso*: o processo foi temporariamente suspenso. Neste caso, todas as atividades e itens de trabalho também são suspensos e nenhuma nova ação pode ser tomada até que o processo seja reativado;
- *Cancelamento Solicitado*: foi solicitado o cancelamento harmonioso do processo, que está aguardando que as atividades ativas sejam concluídas para ser encerrado definitivamente;
- *Cancelado*: o processo foi encerrado, forçadamente ou não, sem ter sido feita a execução completa do fluxo definido.

Quando o processo é criado, ele fica no estado “Não Iniciado”. Após a criação do processo e antes de iniciá-lo, podem ser definidos pelo criador da instância os valores dos parâmetros de entrada e/ou saída do processo ou modificados os seus dados relevantes. Após essas ações preliminares, o processo pode então ser iniciado, quando passará para o estado “Ativo”. Neste momento, as atividades iniciais do processo serão instanciadas e o fluxo começará a ser executado pelo Motor de Workflow.

Enquanto estiver no estado “Ativo”, o workflow pode passar para o estado “Suspenso”, por solicitação do dono do processo ou de um administrador, através da API. Ao ser suspenso, todas as atividades e itens de trabalho em aberto no momento, referentes a este processo, passam também ao estado suspenso e nenhuma ação poderá ser tomada até que o processo seja reativado.

O encerramento de uma instância de processo pode se dar de duas maneiras. O término normal ocorre quando uma atividade é concluída e o Motor de Workflow detecta que não há mais nenhuma transição a ser seguida, nem nenhuma outra atividade em aberto para esta instância de processo. Neste caso, o fluxo passa para o estado de “Concluído” e o processo é encerrado normalmente.

A outra possibilidade é um encerramento prematuro, denominado aqui de cancelamento do processo. O cancelamento poderá ser solicitado pelo dono do processo ou por algum administrador, porque a sua execução deixou de ser necessária, ou determinado pelo próprio Motor de Workflow, devido a alguma situação excepcional, como uma alteração incorreta da definição de processo, por exemplo. A solicitação de cancelamento da instância de processo será recebida pelo Motor de Workflow via API, o que irá disparar um cancelamento em cadeia de todas as atividades e itens de trabalho referentes ao processo que ainda não tenham sido concluídos. Este cancelamento poderá ser forçado, encerrando abruptamente qualquer atividade ou item de trabalho que esteja ativo no momento, ou harmonioso, fazendo com que o processo se encerre assim que todas as atividades e itens de trabalho atuais sejam concluídos. Após este tipo de encerramento, a instância de processo ficará no estado de “Cancelado”. Podem ser canceladas instâncias de processo que estejam nos estados “Não Iniciado”, “Ativo” ou “Suspenso”.

Se, ao solicitar o encerramento harmonioso do processo, houver alguma atividade ativa, o processo irá passar para o estado “Cancelamento Solicitado” até que essas atividades sejam concluídas. Se a conclusão dessas atividades resultar em um encerramento normal do processo, o estado final será “Concluído” e não “Cancelado”.

O processo será considerado “Em aberto” se estiver nos estados “Não Iniciado”, “Ativo”, “Suspenso” ou “Cancelamento Solicitado” e será considerado “Encerrado” se o estado for “Concluído” ou “Cancelado”. Uma vez que o processo tenha atingido um estado em que seja considerado encerrado, ele não poderá mais voltar para um dos estados em aberto.

Somente podem ser criados e iniciados diretamente pela API aqueles processos cujo nível de acesso (atributo *AccessLevel* do elemento *WorkflowProcess*, na XPDL) seja público. Se for privado, só podem ser criadas instâncias daquela definição de processo como parte de um outro processo, através do uso de atividades do tipo *subfluxo*. O ciclo de vida do sub-processo é o mesmo de um processo independente. Todos os dados relevantes e parâmetros terão escopo local ao sub-processo, não podendo acessar diretamente essas informações do processo pai. A troca de informações entre um sub-processo e seu processo pai deverá acontecer exclusivamente através de parâmetros de entrada e saída. A única diferença é que, ao final da execução de um sub-processo, o sistema irá verificar a partir de que atividade ele foi invocado e, se a chamada tiver sido síncrona, concluí-la.

4.3.1.2 *Parâmetros de entrada e saída do processo*

A definição de um processo pode prever que o mesmo terá alguns parâmetros, de entrada e/ou saída, para propiciar a troca de informações com a aplicação que está disparando o processo. A definição dos valores de entrada e obtenção dos valores de saída será feita através de chamadas da API específicas para este fim.

Os valores de entrada devem ser definidos logo após a criação do processo, mas antes que ele seja iniciado. Este é o motivo pelo qual a criação e o início do processo são eventos separados. Os valores dos parâmetros não podem mais ser definidos depois que a instância de processo passar para o estado “Ativo”.

Os valores dos parâmetros de saída, ou de entrada/saída, serão atualizados pelo sistema no decorrer da execução do fluxo. Os seus valores podem ser obtidos a qualquer momento, através da API, mas só terão seu valor definitivo quando o processo já estiver concluído. Se o processo for cancelado, seja de forma forçada ou não, os valores dos parâmetros de saída não necessariamente serão consistentes.

Os tipos de dados aceitos para os parâmetros de processos são os mesmos válidos para os dados relevantes do processo, descritos a seguir.

4.3.1.3 *Dados relevantes do processo*

Na definição de um processo de workflow, pode ser definida uma série de “variáveis”, que conterão dados a serem utilizados na avaliação de expressões de transições condicionais, nomes de participantes, entre outras. Essas variáveis, também chamadas de dados relevantes do processo ou de atributos do processo, podem ser consultadas e modificadas pelo próprio sistema ou através da API do Motor de Workflow.

A linguagem de definição de processos da WfMC estabelece uma série de tipos de dados predefinidos e permite que sejam declarados novos tipos de dados junto com a definição do processo.

O Motor de Workflow dará suporte a todos os tipos básicos previstos na XPDL, que são os seguintes: *String*, Inteiro, Real (*float*), Data/hora, Binário (*boolean*) e *Performer*. Este último, que não está normalmente presente nas linguagens de programação, serve para armazenar referência a participantes do processo.

Quanto aos tipos estruturados, a XPDL considerou obsoletos a partir da versão 1.0 os tipos Registro, União, Enumeração, Lista e *Array*. Esses tipos não serão suportados pelo Motor de Workflow. O único tipo estruturado predefinido restante foi o *Schema Type*, que representa um conteúdo XML que deve obedecer à declaração de um esquema XML. O Motor de Workflow irá dar suporte a este tipo de dado, armazenando o conteúdo XML em um local separado no modelo de dados, juntamente com o esquema declarado. No momento de atribuir o valor a este atributo, será verificado se o XML está em conformidade com o esquema definido. A atribuição do valor só será permitida se houver esta conformidade. Qualquer outra manipulação do valor poderá ser feita somente através de expressões ou de invocação de aplicações.

O valor de um dado do processo será sempre armazenado no sistema como um par <tipo de dado, valor do atributo>, onde o valor do atributo será uma representação *string* do valor para o tipo de dado correspondente. No momento de utilizar o valor para realizar operações, o sistema irá convertê-lo da forma apropriada.

Quaisquer outros tipos de dados que não possam ser considerados como um desses já definidos serão tratados pelo sistema como seqüências de caracteres (*strings*). Por exemplo, os atributos cujo tipo de dados declarado seja uma Referência Externa não terão

nenhuma espécie de validação ou controle, sendo armazenados e retornados exatamente como passados ao sistema.

4.3.1.4 Avaliação de expressões

Na definição do processo, existem vários pontos onde podem ser utilizadas expressões em vez de dados fixos para definir o valor de determinadas construções. O uso mais comum de expressões é na definição do responsável das atividades e no estabelecimento de condições para as transições condicionais.

A sintaxe em que serão escritas essas expressões é totalmente dependente do Motor de Workflow, pois é ele quem irá fazer a avaliação dessas expressões em tempo de execução. Para evitar que a definição de processo fique extremamente vinculada a um único Sistema de Gerência de Workflow, o ideal é que se use alguma sintaxe padrão. Em muitos casos, é usada a linguagem JavaScript, mas não existe atualmente uma linguagem de scripts que possa ser considerada um padrão.

Como o WFMS desenvolvido neste trabalho estará rodando sobre um interpretador PHP, que propicia uma flexibilidade muito grande para a execução dinâmica de scripts, definiu-se que este sistema irá utilizar a própria linguagem PHP como sintaxe para as expressões das definições de processo.

As definições de processo em XPDL possuem um elemento *Script*, onde deve ser definida a linguagem em que as expressões forem escritas. Nas definições utilizadas no sistema que está sendo descrito, esse elemento deve conter o valor “text/php”. Se o elemento não for declarado, PHP será considerada a linguagem padrão. Se for declarada uma outra linguagem, o sistema não poderá criar processos utilizando esta definição.

Para permitir a utilização de dados relevantes do processo e de parâmetros do processo nas expressões, o valor de cada um deles será copiado para uma variável no momento da avaliação da expressão. O nome da variável será o mesmo nome do dado ou do parâmetro no processo, acessando-os como se eles fossem variáveis declaradas no contexto do script.

Essas variáveis que representam os dados e parâmetros do processo conterão uma cópia dos valores desses itens no momento da avaliação da expressão. Não é possível, portanto, tentar alterar o valor dos dados ou parâmetros do processo usando a operação de atribuição na expressão. O conteúdo dos atributos e parâmetros só pode ser alterado através das APIs ou internamente, pelo próprio Motor de Workflow.

O funcionamento desse tratamento de dados relevantes e parâmetros do processo é discutido em mais detalhes na seção 4.4.

Existirão algumas variáveis internas do Motor de Workflow que também poderão ser utilizadas nas expressões, especialmente para transições condicionais ou passagem de parâmetros para aplicações. Essas variáveis são:

- `wf_process_instance_id`: identificador da instância de processo onde a expressão está sendo avaliada, do tipo Inteiro. Estará disponível durante toda a execução do processo, para qualquer expressão;
- `wf_activity_instance_id`: identificador da instância de atividade que está sendo processada, do tipo Inteiro. Poderá ser utilizada na passagem de parâmetros para aplicações, especialmente em atividades de execução automática;
- `wf_work_item_id`: identificador do item de trabalho, do tipo Inteiro. Estará disponível na invocação de aplicações pelo usuário, através da interface de lista de trabalho;

- `wf_current_user_id`: identificador do usuário autenticado atualmente no sistema, do tipo `String`. Pode ser útil para passar a alguma aplicação invocada, para registrar o usuário que a está executando. Estará disponível sempre, para qualquer método executado.

4.3.1.5 Instanciação e ciclo de vida de atividades

Um dos principais componentes de uma definição de processo é a descrição das atividades que serão executadas, bem como do estabelecimento das transições entre elas. As atividades podem representar a invocação automática de aplicações, a execução de um sub-processo, uma tarefa manual ou ainda apenas um ponto de roteamento no fluxo. A forma como a execução de uma instância de atividade será comandada pelo Motor de Workflow varia bastante de acordo com o tipo de atividade, mas algumas características são comuns.

A Figura 4.3 representa o diagrama de transições de estados para as instâncias de atividades. Da mesma forma que o diagrama de estados para as instâncias de processo, o conjunto de estados possíveis não é exatamente igual ao das recomendações da WfMC, mas é compatível com a proposta.

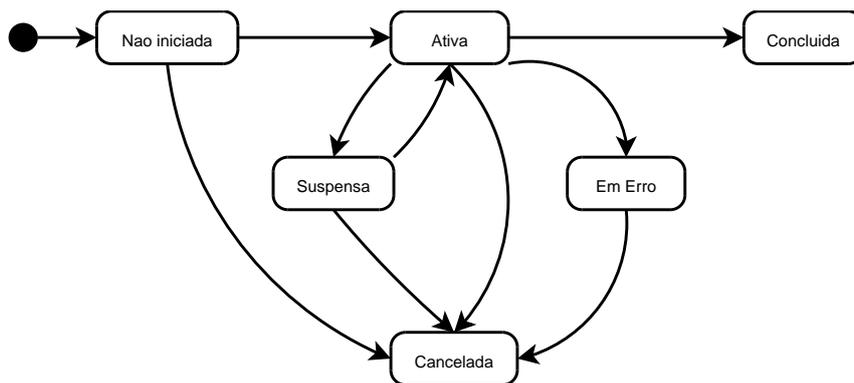


Figura 4.3: Diagrama de Estados de Instâncias de Atividades

Os estados definidos na figura são os seguintes:

- *Não Iniciada*: a instância de atividade foi criada, mas ainda não foi iniciada sua execução;
- *Ativa*: a execução da atividade está em andamento;
- *Concluída*: a atividade terminou sua execução normalmente;
- *Suspensa*: foi paralizada a execução da atividade, provavelmente devido à suspensão do processo do qual ela faz parte;
- *Cancelada*: a atividade foi terminada prematuramente, seja por algum erro na sua execução, pelo cancelamento do processo pai ou por ação de um administrador;
- *Em Erro*: a tentativa de execução da atividade resultou em erro que não foi previsto no fluxo e o sistema está aguardando a intervenção de um administrador.

Toda instância de atividade é criada internamente pelo Motor de Workflow, quando este determina que alguma transição para ela foi habilitada no fluxo, ou quando é uma atividade inicial de algum processo. Não há a possibilidade de se criar uma nova instância de atividade pela API do Motor de Workflow, já que é o fluxo quem determina quando executar cada atividade.

O padrão de intercâmbio de definições de processos da WfMC descreve cinco tipos de atividades: manual (*Implementation/No*), implementada por aplicação (*Implementation/Tool*), sub-processo (*Implementation/Subflow*), roteamento (*Route*) e bloco de atividades (*ActivityBlock*). A dinâmica da transição de estados e o comportamento do sistema variam bastante para cada um desses tipos.

As atividades de **roteamento** são as mais simples. Elas não precisam fazer nada, já que servem somente para “hospedar” regras de transição. A idéia desse tipo de atividade é permitir combinações de transições em seqüência, para permitir condições que não possam ser representadas em uma única atividade. Este tipo de atividade não possui nem participante nem aplicação associada. Após criada, uma instância de tarefa de roteamento passa automática e imediatamente pelos estados “Ativa” e “Concluída”, em seqüência, sendo então avaliadas as transições de saída da atividade.

A linguagem XPDL permite especificar mapas auto-contidos de atividades e transições, denominados de *ActivitySets*, ou conjuntos de atividades. Se há trechos de um processo de workflow que se repetem algumas vezes durante o fluxo, cada trecho pode ser declarado como um conjunto de atividades e reutilizado quando necessário.

Os pontos do fluxo onde a execução de um desses conjuntos deve acontecer são definidos através de uma instância de atividade do tipo **Bloco de Atividades**. Logo após a criação de uma instância de atividade deste tipo, ela passa automaticamente para o estado “Ativo” e é iniciada a execução do conjunto de atividades referenciado, instanciando-se as atividades que ele contém. Ao término da execução do trecho definido no *ActivitySet*, a atividade que lhe deu origem passa para o estado “Concluída” e o fluxo prossegue normalmente. Neste tipo de atividade também não faz sentido a associação de *performer* ou de aplicações.

As instâncias de atividade que fizerem parte de um *ActivitySet* terão um registro no modelo de dados identificando qual a atividade do tipo Bloco de Atividades correspondente, para possibilitar que, ao final da execução do conjunto, se possa retomar a execução do fluxo principal.

Apesar de serem parecidas com a execução de blocos de atividades, as atividades do tipo **sub-processo** possuem diferenças importantes. Uma atividade do tipo sub-processo dispara a execução de um outro processo de workflow declarado no mesmo pacote, ou em um pacote externo referenciado na mesma definição. Este processo secundário é executado em um contexto isolado do processo que o disparou, não acessando os mesmos dados nem parâmetros. O sub-processo terá seus próprios parâmetros de entrada e/ou saída, definidos no momento do disparo, sendo a única forma de intercâmbio direto de dados com o processo principal.

Do ponto de vista do motor de workflow, um sub-processo é tratado como se fosse um processo independente, sem diferenças em termos de avaliação das transições ou execução das atividades. A única diferença é que, no caso dos sub-processos, será armazenado o identificador da instância de atividade que o disparou, possibilitando que o fluxo principal seja retomado quando o processo secundário terminar de executar, se for o caso.

A execução do sub-processo pode ser dar de forma síncrona ou assíncrona com relação ao processo principal. Se a execução for síncrona, a atividade do processo principal irá

ficar no estado “Ativa” até que o sub-processo seja encerrado, quando passará para o estado “Concluída” e será dado prosseguimento ao fluxo principal. No caso da execução assíncrona, após o início do sub-processo, a atividade que o disparou passará para o estado “Concluída” e o fluxo principal terá prosseguimento, em paralelo com o fluxo secundário.

Uma atividade do tipo sub-processo naturalmente não possui participantes nem aplicações associados, já que se trata sempre de uma tarefa desempenhada pelo próprio Motor de Workflow.

Uma atividade **sem implementação** (que geralmente corresponde a uma atividade **manual**) é declarada na definição em XPDL com os elementos *Implementation / No*. Nesse caso, a definição da atividade traz no atributo *Performer* uma expressão que indica a quem será delegada a tarefa de executar a atividade, podendo ser até mesmo mais de uma pessoa. Para cada um, será criado um item de trabalho e o executor não terá nenhum apoio adicional do sistema de workflow para a execução da tarefa, a não ser a sua descrição.

A regra para escolha dos executores da tarefa e instanciação dos itens de trabalho é descrita mais adiante nesta seção. O modo de início e término da atividade podem ser definidos como “Automático” ou “Manual”. O modo “Manual” geralmente é usado, uma vez que não faz sentido uma atividade automática à qual não está associada nenhuma aplicação ou comportamento automático.

As atividades **implementadas por aplicação** possuem ao menos uma ferramenta (do tipo “aplicação” ou “procedimento”) associada. A sua execução pode ser automática ou manual, dependendo dos modos de início e término especificados na definição da atividade. No caso da execução manual – em que pelo menos um dos modos deve ser “Manual” – um participante deve estar associado para executar a atividade. As regras de escolha dos participantes e criação de instâncias de itens de trabalho são as mesmas utilizadas para as atividades sem implementação.

Se a atividade possui tanto o modo de início quanto o de término “Automático”, o sistema encaminhará para o Gerenciador de Aplicações a solicitação para invocar uma a uma, na ordem da definição de processo, as ferramentas associadas à atividade, logo que a execução da atividade iniciar. A execução das aplicações será síncrona. O Motor de Workflow irá aguardar o término da execução das aplicações e, assim que a última acabar, a atividade será concluída, dando prosseguimento ao fluxo.

Quando acontece algum erro durante a execução automática de alguma das aplicações, é lançada uma exceção denominada *ApplicationException*. Se uma das transições de saída da atividade for para essa exceção, ou existe uma transição padrão para exceções (*DEFAULTEXCEPTION*), a transição é seguida. Caso contrário, a atividade é colocada no estado “Em Erro”. O administrador do WFMS ou o responsável pelo processo pode escolher a ação mais apropriada e executá-la via interface de administração ou API do sistema. Ele pode cancelar a atividade, parализando o andamento do fluxo naquele ponto, ou mandar reexecutar a atividade. Neste segundo caso, a atividade que estava em erro passa para o estado “Cancelada” e uma nova instância da mesma atividade é criada e iniciada.

Quando uma atividade que possui aplicações associadas tem o modo de início ou de término “Manual”, será criado um item de trabalho (ou mais, dependendo de como for resolvida a indicação de executores da atividade) e o usuário a quem for associada é que deverá disparar manualmente a execução da aplicação, através de funcionalidades disponibilizadas na sua interface de lista de trabalho. Neste caso, o modo de término será também sempre considerado “Manual”. Este tipo de atividade será utilizado sempre que houver alguma ferramenta dando apoio à execução da atividade para o usuário, como é o

caso do estudo dos tópicos da disciplina, no exemplo do curso do AdaptWeb.

Quando uma instância de atividade der origem a mais de um item de trabalho, a atividade só será considerada concluída quando todos os itens de trabalho também forem encerrados.

O Quadro 4.1 resume o comportamento do Motor de Workflow com relação às atividades manuais e de implementação por ferramenta.

<i>Tipo de Implementação</i>	<i>Início/Término</i>	<i>Executor</i>	<i>Itens de Trabalho</i>	<i>Significado</i>
Nenhuma	Ambos automáticos	Não considerado	Não	Nenhum efeito no sistema
Nenhuma	Um dos dois Manual	Normal	Sim	Tarefa manual
Ferramenta	Ambos automáticos	Não considerado	Não	Execução automática
Ferramenta	Um dos dois Manual	Normal	Sim	Tarefa manual com apoio de aplicações

Quadro 4.1: Comportamento do Motor de Workflow quanto às Atividades de Implementação

A cada atividade, de qualquer um dos tipos descritos, também pode estar vinculada uma prioridade, dada por um número inteiro maior ou igual a zero. Segundo a descrição na XPDL, quando maior este número, maior a prioridade da atividade. No caso do WFMS projetado neste trabalho, o valor da prioridade será utilizado como critério principal de ordenação sempre que houver a opção de escolher mais de uma atividade. Os itens de trabalho criados para determinada atividade também herdarão esta prioridade.

4.3.1.6 Definição de executores das atividades

A cada atividade do sistema que possua o modo de início ou de término manual, deve ser designado um participante que será o responsável por executar a tarefa. As entidades que podem ser indicadas como executoras de uma atividade, como descrito pela WfMC, podem ser pessoas, programas, papéis (*roles*), unidades organizacionais, entre outras possibilidades.

Se a atividade for designada para uma entidade atômica (como uma pessoa ou um programa), o sistema irá criar um item de trabalho designado a este participante, que portanto terá este item na sua lista de trabalho até que assinale que o mesmo foi concluído.

Quando o executor da atividade for uma entidade que agrega outras entidades (como papéis ou unidades organizacionais), o modelo da WfMC deixa a critério da implementação do WFMS a definição do comportamento do sistema. A abordagem escolhida para o sistema projetado neste trabalho é a seguinte: será criado um novo item de trabalho para cada um dos participantes atômicos que fazem parte da entidade indicada para executar a tarefa. No caso de uma unidade organizacional, por exemplo, o sistema irá criar um item de trabalho igual para cada membro da unidade. Neste caso, a atividade só será considerada concluída quando todos os itens de trabalho estiverem também concluídos.

No entanto, muitas vezes o que se deseja é escolher um participante, ou um número determinado de participantes, a partir de um conjunto. Por exemplo, pode ser desejável que uma tarefa de programação seja enviada a algum membro do papel “Programador”, independente de qual seja. Neste caso, não faz sentido atribuir um item de trabalho para cada um dos programadores, nem estabelecer um programador predefinido. O desejado é que o WFMS escolha em tempo de execução a pessoa mais apropriada para executar a

ação.

Para permitir ao autor do processo determinar com mais precisão as regras de seleção do executor da atividade, o sistema disponibilizará rotinas que poderão ser utilizadas nas expressões que definem o executor da atividade. Essas rotinas implementarão alguns métodos de filtro e seleção de entidades, como sorteio, balanceamento de carga e rodízio. A implementação desses métodos será feita no módulo de Gerenciamento de Listas de Trabalho, detalhado mais adiante neste capítulo.

Com a possibilidade de utilizar esses métodos de escolha, o autor do workflow tem bastante flexibilidade para estabelecer os critérios para os executores das tarefas.

Se a expressão que determina o executor de uma atividade resultar em um conjunto vazio de entidades, ou em uma entidade agregadora que não contenha membros, o WFMS não poderá criar nenhum item de trabalho. Já que uma atividade manual não tem como prosseguir sem que algum usuário a execute, o sistema irá levantar uma exceção denominada *NoPerformerException*. Neste caso, o comportamento é o mesmo de quando ocorre um erro na execução automática de uma aplicação: se a atividade possui uma transição de saída específica para esta exceção, ou se existe uma transição do tipo *DEFAULTEXCEPTION*, a transição é seguida. Caso contrário, a atividade passa para o estado “Em Erro”. O administrador do WFMS ou o responsável pelo processo pode cancelar a atividade, paralyzando o andamento do fluxo naquele ponto, ou mandar reexecutar a atividade, através das APIs ou de alguma interface de administração do sistema. No caso de reexecutar a atividade, a atividade que estava em erro vai para o estado “Cancelada” e uma nova instância da mesma atividade é criada e iniciada.

4.3.1.7 Ciclo de vida de itens de trabalho

Um item de trabalho é uma tarefa designada a algum participante do sistema. Este participante deverá executar a ação indicada pelo item de trabalho e, uma vez concluída, indicar ao sistema que a tarefa foi encerrada. O conjunto de itens de trabalho que uma pessoa tem em aberto é considerado sua “lista de trabalho” ou “lista de tarefas”.

O item de trabalho é criado quando uma atividade possui o modo de início ou de término “Manual”. Como descrito na seção anterior, existe um controle para definir quantos itens de trabalho serão criados para uma atividade e para quem serão designados.

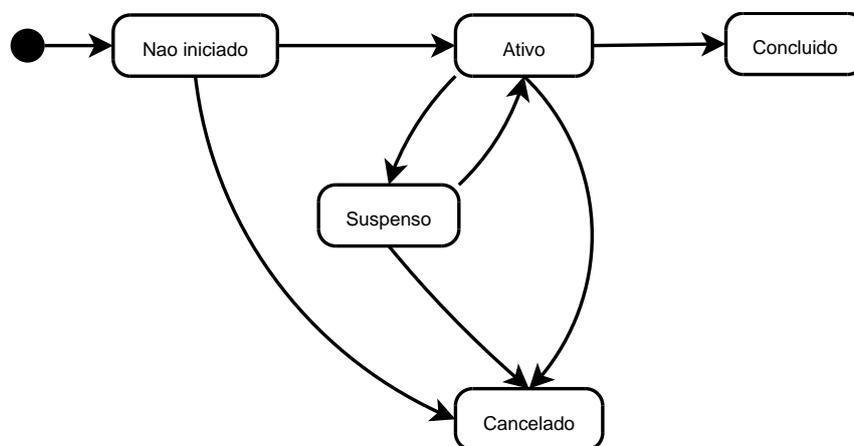


Figura 4.4: Diagrama de Estados de Itens de Trabalho

O ciclo de vida dos itens de trabalho é ilustrado no diagrama de estados da Figura 4.4. Os estados são semelhantes aos das instâncias de processos e de atividades:

- *Não Iniciado*: o usuário ou programa ao qual está designado ainda não começou a executar a tarefa;
- *Ativo*: a execução da atividade está em curso;
- *Concluído*: a tarefa foi finalizada;
- *Suspensa*: a execução do item de trabalho está paralizada, devido à suspensão da atividade ou processo ao qual está associado;
- *Cancelado*: o item de trabalho foi encerrado prematuramente, sem ter sua execução concluída, devido ao cancelamento da atividade à qual está associada ou ao fato de ter sido reencaminhada para outra pessoa.

Se o modo de início do item de trabalho for “Manual”, assim que é criado ele fica imediatamente no estado “Não iniciado”. O participante ao qual o item de trabalho foi designado (“dono” do item de trabalho) deverá explicitamente indicar ao sistema quando começou a trabalhar no item de trabalho, através da interface de Lista de Trabalho que, por sua vez, irá utilizar a API do Módulo de Lista de Trabalho. Quando receber essa indicação, o Motor de Workflow irá passar o item de trabalho para o estado “Ativo”.

Se, no entanto, o modo de início for “Automático”, o item de trabalho já inicia no estado “Ativo”. Neste caso, mesmo havendo aplicações associadas à atividade, elas não são executadas, já que um item de trabalho sempre corresponde à execução manual de uma atividade.

Uma vez iniciado o item de trabalho, o usuário ou programa irá executar a devida tarefa. Ao concluir o trabalho, ele deve encerrar o item de trabalho através da interface de usuário ou da API de Lista de Trabalho. O item de trabalho passará então para o estado “Concluído”.

Enquanto estiver “Ativo”, o item de trabalho poderá ir para o estado “Suspensa”, se a atividade à qual estiver vinculado for suspensa. Neste caso, nenhuma ação poderá ser tomada, como concluir o item de trabalho ou invocar a execução de aplicações associadas a ele. Quando a execução da atividade for restaurada, o item de trabalho voltará ao estado “Ativo”.

Um item de trabalho ativo ou não iniciado pode ser reencaminhado por seu responsável para outro participante. Este item de trabalho, na verdade, passa para o estado “Cancelado” e um novo item de trabalho é criado para o participante selecionado. A única restrição que o WFMS fará quanto ao novo responsável pelo item de trabalho é que o mesmo faça parte de alguma das entidades listadas na declaração de participantes da definição do processo.

O estado “Cancelado” pode ser atingido pelo cancelamento da atividade à qual o item de trabalho corresponde, ou, como descrito acima, quando o item de trabalho é reencaminhado para outro participante.

Os itens de trabalho terão associados a si uma prioridade, herdada da atividade da qual foram originadas. A prioridade do item de trabalho só pode ser alterada através da modificação da prioridade da atividade correspondente. Se uma atividade tiver gerado mais de um item de trabalho, todos eles terão a prioridade alterada conjuntamente.

4.3.1.8 *Processamento das regras de transição*

Uma das tarefas mais complexas e importantes que um Sistema de Gerenciamento de Workflow deve realizar é avaliar e processar as regras de transição do fluxo.

Quando uma atividade é finalizada normalmente, a identificação dos passos seguintes do fluxo é feita em duas etapas:

- (1) avaliar as pós-condições da atividade finalizada (localizar as transições de saída e avaliar suas restrições e condições); e
- (2) avaliar as pré-condições das atividades destino (verificar as restrições e a situação de cada uma das atividades destino das transições habilitadas na primeira etapa).

Na avaliação das pós-condições da atividade, é feita a análise das restrições de transição de saída (*TransitionRestrictions* do tipo *Split*) e obtido o resultado das expressões condicionais de cada transição, determinando quais delas devem ser consideradas e quais serão ignoradas.

A etapa seguinte é executada se houver alguma transição selecionada no passo anterior. Para cada uma das atividades de destino das transições que foram selecionadas no primeiro passo, é feita a avaliação das restrições de transição de entrada (*TransitionRestrictions* do tipo *Join*). Ao final da avaliação, cada uma das transições pode ficar em estado de espera (caso de um “AND-join” incompleto), pode ser ignorada (caso de um “OR-join” já atendido), ou pode dar origem a uma nova instância da atividade, dependendo das transições que já tiverem acontecido no fluxo para aquela atividade.

Essas duas etapas serão descritas em mais detalhes a seguir.

Avaliação das transições de saída

Quando uma atividade de qualquer tipo é encerrada no fluxo, o primeiro passo é a avaliar as pós-condições da atividade, ou seja, as transições de saída. O elemento *TransitionRestrictions* na definição da atividade indica quais transições estão reunidas através de uma das regras de divisão do fluxo: *XOR-split* ou *AND-split*. Se na definição de processo em XPDL houver uma restrição do tipo “AND” relacionando um determinado conjunto de transições de saída, isso significa que todas as transições cuja expressão condicional seja verdadeira devem ser seguidas. Uma restrição do tipo “XOR” significa que somente a primeira delas cuja condição for verdadeira deve ser habilitada.

Se houver mais de uma cláusula *TransitionRestrictions* do tipo *Split* para a mesma atividade, cada uma delas é processada de forma independente. As transições que não fazem parte de nenhuma restrição também são avaliadas separadamente e consideradas se a avaliação de sua expressão condicional for positiva. As transições selecionadas a partir de cada uma das restrições, bem como essas “avulsas”, são eleitas para a segunda etapa da avaliação.

A cada transição pode estar associada uma expressão condicional, que será avaliada e, se o resultado for verdadeiro, fará com que a transição seja considerada “habilitada”. Caso o resultado da expressão seja o valor falso, a transição será ignorada e não será seguida pelo Motor de Workflow.

A avaliação das expressões de cada transição é efetuada da forma descrita no item 4.3.1.4. Se o resultado da expressão não for um valor do tipo *boolean*, será convertido e avaliado conforme as regras específicas da implementação, descritas mais adiante. Se houver qualquer erro durante a avaliação da expressão, o valor retornado será considerado falso e a transição não será seguida.

Caso esta atividade que terminou não tenha nenhuma transição de saída, ou caso nenhuma das transições avaliadas na primeira etapa tenha sido selecionada, este ramo do fluxo é dado como encerrado. Se não houver nenhuma outra atividade em aberto no mesmo fluxo neste instante, a instância de processo como um todo estará concluída.

Quando houver transições de saída que devem ser seguidas, antes de começar a avali-

ação das pré-condições das atividades de destino, o WFMS ainda irá verificar se foi solicitado o cancelamento harmonioso da instância de processo (ou seja, se a instância de processo está no estado “Cancelamento Solicitado”). Se este for o caso, nenhuma transição é seguida. Se, além disso, não restarem outras atividades em aberto neste mesmo fluxo, o processo é cancelado.

Instanciação das atividades de destino

A parte mais delicada envolve a segunda etapa do processo de avaliação das transições. Neste momento, para cada transição de saída selecionada, é verificada a atividade destino.

Cada atividade pode conter uma restrição de transições do tipo *Join*, que pode conter os valores “AND” ou “XOR”. No primeiro caso, a restrição indica que, para que a atividade possa ser instanciada, todas as transições de entrada devem ser habilitadas primeiro. No segundo caso, basta que uma das transições seja habilitada para que a instância de atividade seja criada – as transições seguintes devem ser ignoradas.

Existe ainda uma terceira opção, pois não é obrigatório que exista uma restrição de transição de entrada na definição da atividade. A interpretação no caso da ausência do elemento não está definida na recomendação da WfMC. No WFMS projetado neste trabalho, quando não houver o elemento *TransitionRestrictions* do tipo *Join* na atividade de destino da transição, cada transição de entrada habilitada irá sempre criar uma nova instância da atividade.

Considere, por exemplo, um fluxo em que uma atividade (A) possui três transições de entrada habilitadas (T1, T2 e T3), conforme ilustra a Figura 4.5.

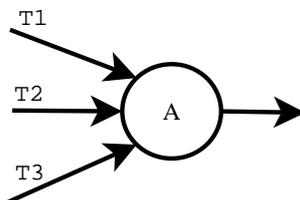


Figura 4.5: Atividade com Múltiplas Transições de Entrada

O comportamento do Motor de Workflow se modifica significativamente de acordo com a regra de restrições de transições de entrada definida para a atividade A. O Quadro 4.2 define a ação em cada um dos casos.

<i>Restrição</i>	<i>Comportamento</i>
Nenhuma	Cria três novas instâncias da atividade A, uma ao habilitar cada transição.
AND-Join	Cria apenas uma nova instância de A, quando todas as transições tiverem sido habilitadas.
XOR-Join	Cria apenas uma nova instância de A, quando a primeira das transições tiver sido habilitada, independentemente de qual seja.

Quadro 4.2: Comportamento do Motor de Workflow quanto às Restrições de Transições de Entrada

Como o controle das restrições de entrada necessita considerar informações de transições já ocorridas, o modelo de dados de execução do Motor de Workflow conterá uma tabela para armazenar informações das transições que já aconteceram ou que ainda estão pendentes. Esta tabela conterá os identificadores das definições das atividades de origem

e destino, o estado da transição e os identificadores da instância que efetivou a transição e que foi criada pela transição.

Os estados possíveis para a transição são: “Atendida”, “Ignorar”, “Ignorada”, “Aguardando” e “Não Atingível”. O significado de cada um é esclarecido a seguir.

Nos casos em que a atividade destino não possui restrições de transição de entrada, uma nova instância da atividade é criada e é registrada a transição na tabela de histórico de transições com estado “Atendida”. Este é o caso mais simples e normalmente vai ocorrer em atividades que são alcançadas somente por uma transição de entrada. Esta característica pode ser utilizada também, por exemplo, nos ciclos do fluxo, em que uma transição de entrada cria a instância pela primeira vez, e uma segunda transição fecha um ciclo, necessitando criar uma nova instância da mesma tarefa, sem relação com a transição anterior.

Quando a atividade destino possui restrição de transição de entrada do tipo XOR, a primeira transição de entrada habilitada irá criar uma nova instância da atividade e será criado um registro na tabela de histórico, com estado “Atendida”. No entanto, o Motor de Workflow precisa saber que as próximas transições habilitadas com destino a esta mesma atividade devem ser ignoradas. Para isso, será criada também na tabela de histórico de transições uma entrada para cada transição de entrada, com situação “Ignorar”. Desta forma, quando qualquer uma delas for alcançada pelo fluxo, o sistema irá identificar que a transição não precisa ser seguida, mudando o estado para “Ignorada” na tabela.

Em caso de existência de ciclos no fluxo, no entanto, uma atividade com XOR-join que esteja dentro do ciclo necessita ser instanciada uma vez para cada nova “rodada” de execução do mesmo trecho de processo. Neste caso, é necessário que haja um controle para que o WFMS consiga identificar que a transição de entrada é de uma nova execução do ciclo e que não deve ignorá-la. Para contornar este problema, a avaliação da restrição de transições de entrada do tipo XOR irá verificar se cada transição de entrada tem como ancestral no histórico do processo a própria instância da atividade destino instanciada anteriormente. Se for o caso, caracteriza-se uma nova rodada do ciclo e a transição pode ser considerada. Para efetuar este controle é armazenada na tabela de histórico das transições o identificador da instância de atividade que foi instanciada a partir de cada transição.

Nos casos em que a atividade destino possui restrição de transição de entrada do tipo AND, no momento em que a sua primeira transição de entrada for habilitada, deve ser criado um registro na tabela de histórico para ela com estado “Atendida” e um para cada uma das demais transições de entrada, com o estado “Aguardando”. Para cada transição que atinge esta atividade destino, o estado será alterado para “Atendida”. Quando todas as transições estiverem com esse estado, a instância da atividade destino pode ser criada.

Como o fluxo pode conter transições condicionais, vários trechos do processo podem não ser acionados. Há o risco, portanto, no caso das atividades com AND-join, de alguma de suas transições de entrada não seja atingível, embora outras sejam. Neste caso o Motor de Workflow poderia ficar aguardando indefinidamente pela habilitação de uma transição de entrada que nunca iria ocorrer e, portanto, a execução da atividade destino nunca aconteceria.

Em alguns casos, este é comportamento desejado, já que se exige que todas as atividades predecessoras tenham sido executadas. Em outros casos, no entanto, se deseja que apenas as transições provenientes de ramos do fluxo que tenham sido acionados participem da avaliação do “AND-join”.

Para permitir ambos os comportamentos, uma configuração do Motor de Workflow irá indicar se o AND-join deverá ficar em estado de “impassé” quando uma das transições de

entrada não for atingível ou se esta transição deve ser desconsiderada. Neste último caso, para cada transição com o estado “Aguardando” que aponte para a atividade destino, será feita a verificação se ainda é possível atingir a transição. Se não for, o estado da transição será alterado para “Não Atingível” e esta não será considerada na avaliação da restrição de transição de entrada da atividade destino.

No caso do AND-join, somente será criada uma nova instância da atividade quando não restarem mais transições com estado “Aguardando” na tabela de histórico das transições.

4.3.2 Gerenciamento de Definições de Processos

O Módulo de Gerenciamento de Definições de Processos é o componente do WFMS responsável por controlar o ciclo de vida das descrições de processos e armazenar suas descrições XPDL. A API deste módulo possibilita o cadastramento de novas definições de processo, a suspensão ou exclusão das definições já cadastradas ou ainda fornece informações sobre as definições armazenadas. Também é responsável pelo controle de versões das definições.

Não é tarefa do WFMS desenvolvido suportar a edição de definições de processo. Elas poderão ser criadas e editadas em qualquer ferramenta que possibilite a exportação no formato XPDL, ou até mesmo editadas diretamente em algum editor de texto ou de XML. O WFMS possui interfaces para que as definições de processo sejam carregadas para o repositório do WFMS, o que permite que possam ser disparadas novas instâncias de processo obedecendo a essa definição. Da mesma forma, podem ser atualizadas ou removidas definições de processo deste repositório.

4.3.2.1 Armazenamento e recuperação das definições

O módulo de Gerenciamento de Definições de Processo utiliza o próprio banco de dados MySQL para armazenar as definições de processo e alguns metadados de controle, como a data e o usuário que fizeram o cadastramento e a indicação de que processo está habilitado ou não.

A definição do processo em si é armazenada no próprio formato XPDL. O arquivo XPDL, na verdade, consiste da definição de um pacote que pode, por sua vez, conter mais de uma definição de processo. Sendo assim, cada entrada no repositório de definições de processo corresponde a um pacote.

Como o armazenamento é feito nativamente no formato XPDL, o sistema não tem necessidade de implementar rotinas de exportação, pois o formato original é preservado. Qualquer editor que gere a definição de processo em XPDL poderá ser utilizado para gerar a definição, uma vez que não são utilizadas extensões ao formato proposto pela WfMC.

Quando é solicitado o registro de um pacote XPDL no sistema, é feita uma verificação de sintaxe, para evitar que um XML mal-formatado ou um arquivo que não obedeça a especificação XPDL seja carregado no repositório. Essa verificação de sintaxe é feita através do uso da API DOM (*Document Object Model*).

Uma tabela auxiliar é utilizada pelo Módulo de Gerenciamento de Definições de Processo para indexar as definições processos contidas em cada pacote. O identificador, o nome, a versão e outras informações sobre o documento são extraídas do próprio XPDL e armazenadas juntamente com este índice, para facilitar a consulta e recuperação.

As operações disponibilizadas pelo módulo de Gerenciamento de Definições de Processo são a carga de um pacote XPDL, a exclusão de um pacote já cadastrado, a suspensão de um pacote (situação em que deixa de ser possível iniciar novos fluxos a partir daquela

definição) e a consulta aos metadados e às informações contidas nas definições de pacote, processos ou atividades em si. É permitida também a atualização de definições de processo já cadastradas no repositório, mas estarão neste caso sujeitas às restrições apresentadas no tópico a seguir, em função das características de versionamento das definições e evolução dinâmica das definições de processo.

O Motor de Workflow utiliza a API deste módulo para recuperar informações sobre a descrição do processo quando necessário, em especial durante a instanciação de atividades e processamento das regras de transição. Outros programas, como a interface de administração proposta para o módulo, também podem utilizar esta API.

Quando é solicitada a informação a respeito de uma definição de processo ou atividade, a descrição do pacote correspondente em XPDL é lida do banco de dados e é feito o processamento do conteúdo XML através do uso de DOM (*Document Object Model*). Desta forma, toda a estrutura do arquivo XPDL é carregada para memória e, posteriormente, convertida nos objetos das classes ou tipos apropriados. É feita a localização da informação solicitada, que é devolvida para o Motor de Workflow ou qualquer outro programa que tenha invocado a API, sem que o mesmo tenha necessidade de conhecer como ela está guardada. Um mecanismo de cache é utilizado para evitar a necessidade de interpretar o mesmo documento XPDL múltiplas vezes durante a mesma requisição.

4.3.2.2 *Controle de versões*

O documento XPDL possui um elemento que indica o número da versão do pacote. Cada processo individual dentro da especificação do pacote também pode conter um número de versão, que será herdado do pacote se não for informado. Além disso, cada definição de processo dentro do pacote XPDL possui um elemento que indica a data de início e término da sua validade.

Quando a definição de um pacote é registrada no repositório, é feita a indexação do conteúdo do pacote na tabela auxiliar já mencionada. Além do identificador e do nome de cada definição de processo constante no pacote, é relacionado também o seu número de versão e suas datas de início e término da validade.

Múltiplas versões de uma mesma definição de processo podem estar cadastradas simultaneamente no repositório. Sempre que uma instância de processo é iniciada, ela fica vinculada àquela versão da definição de processo que foi utilizada para seu disparo, não sendo possível alternar posteriormente esta associação.

Para cadastrar no repositório uma versão nova de um processo, deve ser feita a carga de um arquivo XPDL referente ao mesmo pacote já existente, mas com identificadores de versão das definições de processo diferentes daqueles cadastrados anteriormente. A nova versão é cadastrada sem sobrescrever a pré-existente. Desta forma, aquelas instâncias de processo que haviam sido disparadas a partir da versão anterior da definição de processo permanecem vinculadas a ela e podem ter prosseguimento normalmente.

Um aspecto importante a salientar, finalmente, é que a versão mais atual do processo nem sempre será aquela que foi carregada por último, uma vez que pode carregar processos que passarão a ter validade futuramente. Qualquer chamada à API do módulo de Gerenciamento de Definições de Processo deverá informar não apenas o identificador do pacote ou processo que se deseja consultar, mas também a sua versão. Sempre que a informação de versão não estiver presente, o Gerenciador de Definições de Processo irá considerar a definição de processo que estiver válida no momento e que houver sido cadastrada mais recentemente.

4.3.2.3 *Evolução dinâmica de definições*

O Gerenciador de Definições de Processo do WFMS possibilita também atualizar uma versão específica de uma definição de pacote ou de processo já presente no repositório. Para tanto, basta que seja feita a carga de um novo arquivo XPDL que contenha identificadores de versão iguais a alguma definição já cadastrada. Com a atualização, os processos vinculados à versão substituída passarão a estar vinculados à nova definição.

Esta alternativa deve ser utilizada com muito cuidado. Apesar de ela permitir uma flexibilidade muito grande, tornando possível ajustar e atualizar a definição de um processo para instâncias de processos já existentes (concluídas ou ativas), o Motor de Workflow não conseguirá coordenar mais aqueles processos cujas instâncias de atividades em andamento não sejam compatíveis com a nova versão da definição de processo.

Se a nova definição de processo, por exemplo, não contiver mais a definição de alguma atividade que tenha instâncias em andamento, ou tiver excluído a especificação de uma transição pela qual uma instância de atividade com restrição do tipo AND-join esteja aguardando, não será mais possível dar prosseguimento normal ao fluxo.

O módulo de Gerenciamento de Definições de Processo do WFMS não prevê neste momento a execução de validações para verificar se uma definição de processo atualizada é compatível com as instâncias de processo em andamento. Também não está prevista a execução automática de adaptações nas instâncias de processo para se adaptarem a mudanças na definição de processo. Essas questões, que são importantes para ampliar o potencial de uso do workflow para sistemas adaptáveis, poderão ser melhor exploradas futuramente em novos trabalhos.

No entanto, o simples fato de permitir a atualização de definições de processo já abre um grande leque de possibilidades, permitindo que, por exemplo, os passos futuros dos processos de aprendizado do AdaptWeb sejam revisados a partir dos dados coletados durante a execução dos cursos.

4.3.3 **Manipulação de Listas de Trabalho**

O Módulo “*Gerenciador de Listas de Trabalho*” disponibiliza rotinas para consultar a lista de trabalho dos participantes de workflow e interagir com elas, encerrando e reencominhando itens de trabalho ou invocando aplicações, por exemplo. É a base para a construção de aplicações de workflow voltadas para os usuários finais, como a interface de Lista de Trabalho.

A lista de trabalho é o elemento central da interação dos participantes do workflow, pois é nela que serão apresentadas todas as atividades pendentes, onde o usuário poderá obter as informações relevantes para executar sua tarefa e onde ele irá indicar ao sistema que a tarefa está concluída.

O Motor de Workflow é o responsável por criar os itens de trabalho e gerenciar seu ciclo de vida, enquanto o módulo de Gerenciamento de Listas de Trabalho é o responsável por fornecer as consultas sobre a base de dados de itens de trabalho, receber os comandos dos usuários e, se necessário, repassá-los aos demais componentes, como o próprio Motor de Workflow e o Gerenciador de Aplicações.

4.3.3.1 *Interação do usuário com a lista de trabalho*

Segundo a definição da WfMC, a lista de trabalho é uma consulta sobre o conjunto de itens de trabalho segundo algum critério qualquer (Workflow Management Coalition, 1997).

A idéia mais comum sobre a lista de trabalho é que ela consiste no conjunto de atividades pendentes destinadas ao usuário atual do sistema. De fato, a grande maioria das consultas feitas sobre um conjunto de itens de trabalho por participantes do workflow será deste tipo. No entanto, se um gerente quiser fazer uma busca de todos os itens de trabalho já concluídos por funcionários do seu departamento, ele também estará montando uma lista de trabalho.

Sendo assim, o módulo de Gerenciamento de Listas de Trabalho deverá permitir não apenas a obtenção do conjunto de itens de trabalho pendentes, mas também a possibilidade de consulta ao históricos de tarefas e filtros por diversos critérios, como datas de início e encerramento, prioridade, processo do qual fazem parte, entre outros.

Naturalmente, cada usuário que fizer a consulta poderá visualizar apenas aquelas tarefas que fazem parte do universo de itens de trabalho para os quais tiver permissão de acessar. Este controle é realizado pelo módulo de Gerenciamento de Segurança, como descrito mais adiante.

Além das consultas, o módulo de Gerenciamento de Listas de Trabalho deve permitir as interações do usuário com as suas tarefas. A principal delas é a indicação de que uma tarefa foi concluída, que na verdade é a mudança de estado do item de trabalho de “Ativo” para “Concluído”. A obtenção e atualização de valores dos dados relevantes do workflow também é possível através da API deste módulo.

Por fim, o usuário que estiver realizando suas tarefas pode solicitar a execução de aplicações associadas à tarefa através da sua interface de Lista de Trabalho. Nesses casos, o componente de Lista de Trabalho repassa a solicitação ao módulo de Gerenciamento de Aplicações, que irá efetuar a execução.

4.3.3.2 Regras de seleção de participante

Na especificação do XPDL, não existe nenhuma construção que indique se alguma atividade designada para um papel ou organização deve gerar apenas um item de trabalho, que poderia ser realizado por qualquer um de seus membros, ou se deve gerar vários itens de trabalho, para os diversos componentes do referido grupo. Essa decisão é deixada para a implementação do WFMS, o que é um dos motivos pelos quais se considera que a especificação deixa lacunas importantes.

Como já descrito na seção 4.3.1.6, o comportamento definido para o WFMS descrito neste trabalho é o de que será criado um item de trabalho para cada membro de uma entidade composta, se esta for indicada como executora da atividade.

Para dar mais flexibilidade ao autor do processo de workflow, o módulo de Gerenciamento de Listas de Trabalho implementará uma série de funções que permitirão selecionar participantes atômicos dentro de uma entidade a partir de algumas regras de filtragem pré-definidas. Essas rotinas podem ser invocadas na expressão que define o responsável por um item de trabalho.

As regras de filtragem a serem disponibilizadas são as seguintes:

- *sorteio*: escolhe aleatoriamente um participante (ou um número especificado de participantes) dentre os que fazem parte de uma entidade indicada;
- *balanceamento de carga por atividade*: seleciona os membros de uma entidade que possuam em suas listas de trabalho a menor quantidade de itens pendentes correspondentes à definição de atividade em questão;

- *balanceamento de carga total*: seleciona aqueles participantes da entidade que têm o menor número total de itens ativos em sua lista de trabalho;
- *rodízio*: promove um revezamento entre os membros da entidade, escolhendo aquele que está a mais tempo sem receber uma tarefa, como na ordem de uma estrutura de fila circular.

Por exemplo, se existe um papel denominado “Analista de Sistemas” no repositório de entidades e existe uma atividade de avaliação que precisa ser executada por pelo menos dois analistas, o elaborador do processo pode desejar que os itens de trabalho sejam criados para os dois analistas que têm menos tarefas em sua lista de trabalho. Neste caso, indicaria a entidade, a quantidade de participantes que deseja selecionar (dois) e o método de seleção (balanceamento de carga total) na expressão de executor da atividade na definição do processo.

Para efetuar a escolha dos participantes, o Gerenciador de Listas de Trabalho irá recuperar a estrutura hierárquica das entidades, através da API do Gerenciador de Entidades, e irá consultar a base de dados de execução de processos para gerar as estatísticas necessárias para os cálculos.

Rotinas que selecionam usuários com base na hierarquia organizacional (por cargos, relações de chefia ou funções exercidas) serão implementadas pelo módulo de Gerenciamento de Entidades.

4.3.3.3 Reencaminhamento automático

É comum, em qualquer organização, que as pessoas se ausentem temporariamente de suas atribuições em algumas situações, como viagens, férias ou qualquer tipo de indisponibilidade. Nesses casos, quando as atividades que essas pessoas exercem não podem ser postergadas até o seu retorno, normalmente é designado um substituto, que irá desempenhar as funções durante o tempo necessário. As incumbências continuam sendo endereçadas à pessoa ausente, mas outra as executa em seu lugar.

Quando as atividades são coordenadas por sistemas de workflow, este tipo de situação precisa ser previsto, para que a lista de trabalho de uma pessoa ausente ou indisponível não fique acumulando tarefas que precisam ser realizadas antes do seu retorno.

O módulo de Gerenciamento de Listas de Trabalho possuirá rotinas para que um participante de workflow possa indicar um substituto temporário, ao qual suas atividades serão automaticamente reencaminhadas quando forem criadas. Ao ser cadastrada a regra de encaminhamento, deverão ser informados obrigatoriamente a data de início e o substituto, e opcionalmente a data de término e a definição de processo e/ou atividade para a qual a substituição é válida. Se não for indicado um processo ou atividade, todos os itens de trabalho serão reencaminhados durante o período especificado.

4.3.4 Gerenciamento de Aplicações

O Módulo “*Gerenciador de Aplicações*” implementa o suporte para a execução de alguns tipos de aplicações a partir do workflow. Possibilita o cadastramento das aplicações e configuração da invocação, incluindo forma de chamada e passagem de parâmetros.

O WFMS projetado dará suporte à execução de três tipos de aplicação: aplicativos web, executados via browser com interação do usuário final; chamadas nativas, executados na própria linguagem e ambiente do Gerenciador de Aplicações; e serviços web (*web services*), baseados no conjunto de padrões XML, SOAP, HTTP e UDDI.

Independentemente do tipo da aplicação a ser chamada, ela deverá estar cadastrada no repositório do Gerenciador de Aplicações para que sua execução seja possível. Este cadastro, além do identificador da aplicação, que será utilizado para referenciá-las nas definições de processo, armazena o seu endereço (URL, descritor, etc.) e suas interfaces, ou seja, da sua lista de parâmetros, tipos de dados e direção (entrada, saída ou entrada/saída).

A invocação de aplicações pode ser feita em dois momentos: pelo usuário final, a partir da interface de lista de tarefas, quando existem aplicações associadas a um item de trabalho, ou pelo próprio Motor de Workflow, durante a execução de atividades automáticas.

As aplicações invocadas pelo usuário a partir da lista de trabalho devem permitir a ele visualizar o resultado da execução e, no caso de aplicativos Web, acessar e interagir com a mesma. Por isso, o módulo de Gerenciamento de Aplicações terá um componente visual, que poderá ser acessado via browser através de uma URL e irá executar o aplicativo e exibir seu resultado, no caso de chamadas nativas ou serviços web, ou irá abrir a URL no caso de aplicativos web.

4.3.4.1 Aplicativos Web

Os aplicativos web consistem de uma URL a ser acessada pelo protocolo HTTP, passando pelo método GET os parâmetros de entrada. No caso de chamadas feitas pelo usuário através de sua lista de trabalho, não é possível haver parâmetros de saída, uma vez que a requisição deve gerar um documento (normalmente HTML) que será exibido para o usuário no navegador. Nas chamadas feitas internamente pelo motor de workflow, a URL é invocada e seu resultado pode ser colocado em um atributo de saída. Por isso, nesses casos faz sentido haver apenas um atributo de saída.

Se não for possível acessar a URL cadastrada por indisponibilidade de rede ou do servidor remoto, a execução será considerada mal-sucedida. O mesmo acontecerá se o resultado da requisição for um código de erro HTTP, como página não encontrada (erro 404).

4.3.4.2 Chamadas nativas

As chamadas nativas consistem em rotinas disponíveis diretamente para o Gerenciador de Aplicações. No caso deste trabalho, em que a implementação é no ambiente PHP, as chamadas nativas consistirão da chamada de rotinas em PHP. Se, no futuro, este WFMS for implementado em Java, por exemplo, a execução de chamadas nativas consistirá na invocação de métodos de classes Java.

Neste caso, no cadastro da aplicação no módulo de Gerenciamento de Aplicações deverá ser cadastrado o caminho do código (o arquivo PHP, ou o nome da classe Java, dependendo da implementação), o nome do método a ser invocado e os parâmetros de entrada e saída da rotina.

Se a execução resultar em uma exceção, ela é considerada mal-sucedida. Se, no entanto, a execução termina normalmente, o sistema considera que houve sucesso e transfere os valores dos parâmetros de saída da chamada para os campos de dados da atividade. Maiores detalhes sobre a implementação deste mecanismo no ambiente PHP são dados na seção 4.4.

4.3.4.3 Serviços Web

O padrão de Serviços Web (World Wide Web Consortium, 2004) é atualmente consenso com relação à invocação de procedimentos remotos em ambientes distribuídos e heterogêneos. Ele utiliza os mesmos princípios dos antigos mecanismos de Chamada Remota de Procedimentos (RPC, do inglês *Remote Procedure Call*), mas baseado em padrões modernos como XML e o protocolo HTTP.

Por isso, para a invocação de rotinas externas ao ambiente do WFMS, o Gerenciador de Aplicações permite a definição de aplicações do tipo Serviço Web.

A comunicação entre um cliente e um provedor de serviço web é feita via HTTP, tráfegando pacotes com mensagens XML que obedecem ao padrão SOAP (do inglês *Simple Object Access Protocol*, Protocolo Simples de Acesso a Objetos). Todo serviço web deve possuir um descritor, escrito também em XML, obedecendo à linguagem WSDL (do inglês *Web Services Description Language*, Linguagem de Descrição de Serviços Web). Este descritor contém a interface do serviço e deve ser de conhecimento do cliente que vai acessá-lo.

Para cadastrar um Serviço Web junto ao módulo de Gerenciamento de Aplicações, portanto, é necessário informar a URL do seu descritor WSDL, além da sua lista de parâmetros. O sistema irá montar a requisição no formato SOAP e enviar para a URL do serviço. A resposta deverá consistir de um conjunto de parâmetros de saída, também encapsulados em uma mensagem SOAP. Estes resultados da chamada serão mapeados para os parâmetros de saída configurados para a aplicação.

4.3.5 Gerenciamento de Entidades

O Módulo de Gerenciamento de Entidades é responsável por fornecer ao Motor de Workflow e demais componentes as informações a respeito dos possíveis participantes do Workflow e da estrutura organizacional à qual estão vinculados. As entidades podem ser atômicas, como pessoas, programas ou recursos em geral, ou podem ser agregadoras, como grupos, papéis, unidades organizacionais, entre outros.

Este componente é necessário uma vez que o Sistema de Gerenciamento de Workflow necessita dessas informações para a definição da atribuição de tarefas, segurança e administração do sistema. Além disso, tarefas a serem executadas durante o fluxo podem necessitar de informações a respeito do modelo organizacional que tipicamente ficam armazenadas em um Sistema de Gerenciamento de Diretório.

Exatamente por isso, a denominação deste módulo poderia ser também Módulo de Gerenciamento de Diretório, considerando-se a terminologia utilizada por padrões como o do protocolo LDAP (*Lightweight Directory Access Protocol*), embora possua um âmbito mais restrito.

Os padrões da WfMC não estabelecem interfaces referentes a este tipo de componente, nem determinam que funcionalidades o Motor de Workflow deverá prover para que informações da estrutura organizacional possam ser especificadas nos processos.

Considerando-se simplesmente o Motor de Workflow e seu funcionamento interno, o módulo de Gerenciamento de Entidades precisa apenas informar quais os participantes habilitados a participar de processos, o tipo de cada participante e, no caso de entidades que possuem membros, como papéis e unidades organizacionais, a composição de cada uma delas.

Em sistemas de workflow típicos, no entanto, o andamento do fluxo de trabalho é definido muitas vezes em função da hierarquia organizacional, estabelecendo tarefas de

acordo com cargos ou perfis, definindo alçadas de aprovação, reencaminhando tarefas para substitutos e, muitas vezes, enviando notificações de atraso para chefias. Por isso, a estrutura organizacional precisa estar disponível para que a aplicação de workflow possa utilizar suas informações e tomar as decisões de roteamento das tarefas.

O modelo de dados inicial para este componente preverá uma implementação genérica e flexível de uma estrutura organizacional, mas relativamente simples. Ela é suficiente para necessidades básicas de um sistema de workflow como o projetado neste trabalho.

Toda empresa ou instituição possui a sua própria estrutura organizacional, com papéis, hierarquia e características criadas especificamente para suas necessidades. Algumas vezes, esta estrutura já está cadastrada e gerenciada em algum sistema específico, como uma outra base de dados ou um sistema baseado no protocolo LDAP.

Não faz sentido exigir que a organização seja obrigada a replicar toda essa estrutura para uma base proprietária do Sistema de Gerenciamento de Workflow, gerando duplicidade de recursos de armazenamento e administração.

Por isso, embora inicialmente o módulo preveja a utilização de um modelo de dados em que essas informações são armazenadas no banco de dados do WFMS, a estrutura do Módulo de Gerenciamento de Entidades foi elaborada de modo que permita a sua reimplementação voltada para outros repositórios. Pode-se substituir, por exemplo, a implementação deste módulo que acessa banco de dados por outra que acesse um servidor LDAP, desde que continuem sendo obedecidas as APIs que os demais componentes do WFMS utilizam.

4.3.6 Gerenciamento e Modelo de Segurança

Embora bastante relacionado com o gerenciamento de entidades, o Módulo de Gerenciamento de Segurança é um componente à parte, que permite a configuração e gerenciamento dos aspectos de segurança do sistema, como autenticação, gerenciamento de sessões, registro e verificação de permissões e *logs*.

Como a tecnologia de workflow permite aplicações nas mais diversas áreas, algumas delas podem exigir requisitos de segurança bem mais rígidos que outras. Em processos como os exemplificados neste trabalho, referentes a educação à distância, alguns prejuízos poderiam advir de uso mal intencionado do sistema, como uma pessoa realizar tarefa por outra, mas as conseqüências normalmente seriam contornáveis. Existem aplicações, no entanto, em que uma falha de segurança no sistema de workflow pode implicar na movimentação indevida de grandes somas de dinheiro ou na divulgação de informações confidenciais para pessoas não autorizadas, por exemplo.

Por isso, um Sistema de Gerenciamento de Workflow tem a necessidade de cuidar dos aspectos de segurança, bem como oferecer mecanismos para rastrear eventuais operações não idôneas.

O objetivo deste módulo é, portanto, fornecer informações para os demais módulos sobre a autoridade que cada usuário possui para executar cada ação no sistema, bem como fornecer a possibilidade de registrar cada operação, efetivada ou não.

As funcionalidades do Módulo de Segurança do WFMS podem ser divididas basicamente em autenticação, autorização e registros de auditoria. Os itens a seguir descrevem cada um deles de forma superficial, uma vez que não é o objetivo do trabalho aprofundar-se muito em detalhes sobre segurança.

4.3.6.1 Autenticação

Autenticação consiste em garantir que o usuário que está acessando o sistema é realmente quem ele alega ser. A especificação da WfMC determina que cada método da API do WFMS deverá receber como parâmetro uma estrutura de dados que representa uma sessão estabelecida com o Motor de Workflow. A criação desta sessão deve se dar por uma rotina de autenticação prevista na API, na qual as credenciais do usuário (nome de login e senha, ou outros mecanismos, como certificado digital) são validados. Uma vez estabelecida a sessão, esse identificador deve ser utilizado em cada interação com a API para que o sistema verifique se nenhuma operação indevida está sendo executada.

Para implementar este comportamento, o módulo de Gerenciamento de Segurança irá possuir métodos para validação de credenciais e criação e armazenamento de dados de uma sessão. Na implementação proposta inicialmente, será uma verificação simples de senha, que será um atributo do usuário armazenado no mesmo repositório utilizado pelo Gerenciador de Entidades.

Sempre que as APIs do Motor de Workflow ou de algum dos demais componentes for invocada, irá receber como parâmetro uma estrutura de dados que conterá um identificador de sessão, que será validado a cada chamada e permitirá ao sistema saber quem está executando o procedimento.

Implementações mais complexas deste módulo no futuro podem prever a autenticação através de mecanismos como certificados digitais, integração com login de rede ou utilização de protocolos de autenticação diversos.

4.3.6.2 Autorização

O processo de autorização consiste em verificar se o usuário autenticado possui legitimidade para solicitar a execução de determinada operação dentro do sistema.

Algumas operações são destinadas apenas a perfis específicos de usuários. A administração de instâncias de processo, como abortar e suspender a execução, assim como as operações dos módulos de Gerenciamento de Definições de Processos e de Gerenciamento de Aplicações, por exemplo, cabem apenas a um usuário com perfil de administrador do sistema.

Muitas vezes, entretanto, além de simplesmente verificar qual é a operação, é necessário verificar qual o objeto no qual essa operação está tentando ser executada. No caso do Motor de Workflow, o “objeto” em questão pode ser uma instância de processo, uma instância de atividade, um item de trabalho ou mesmo um atributo de um desses objetos. Nesses casos, a execução da operação só deve ser autorizada para os usuários que exercem uma função com relação ao objeto. A conclusão de um item de trabalho, por exemplo, somente deve ser permitida a algum usuário que o tenha em sua lista de trabalho.

O usuário ou entidade que disparou a criação de uma instância de processo é considerado o seu proprietário e terá o privilégio de realizar operações de controle sobre ele, como iniciar, cancelar, suspender e reativar. Se na definição do processo estiver especificada alguma outra entidade responsável pelo mesmo, estas entidades também terão os mesmos direitos.

A execução de aplicações por parte de usuários comuns só é permitida quando associada a itens ainda pendentes da sua lista de trabalho.

Usuários que possuam um perfil de Administrador do Workflow normalmente terão permissão para executar qualquer operação, interferindo nos processos livremente. Mesmo nestes casos, a operação sempre será registrada e é possível identificar que usuário ex-

ecutou a operação.

Na implementação de cada método das APIs do Sistema de Gerenciamento de Workflow proposto, o primeiro passo será solicitar ao módulo de Segurança que consulte o cadastro de permissões, verificando se a operação é permitida.

4.3.6.3 *Registros de auditoria*

Além de verificar permissões e autorizar ou não a execução de ações, o Sistema de Gerenciamento de Workflow precisa fornecer aos administradores do sistema e dos processos informações para rastrear operações executadas, mesmo que tenham sido feitas por pessoas autorizadas.

Na definição da interface 2 do Modelo de Referência (Workflow Management Coalition, 1997), a WfMC indica em cada operação da API quais os registros de auditoria devem ser gerados. Cada um dos módulos do sistema que implementam as funções da API solicitam ao Módulo de Segurança o registro de auditoria correspondente, sempre que for necessário.

Não é a intenção, neste trabalho, projetar as funcionalidades da interface 5 do Modelo de Referência da WfMC (Workflow Management Coalition, 1998), que diz respeito a ferramentas de administração e auditoria. No entanto, o armazenamento de dados para análise das ocorrências relevantes do sistema já é possível.

4.4 Aspectos de Implementação do Sistema

O objetivo desta seção é apresentar as decisões que foram tomadas durante o planejamento e fase inicial da implementação do WFMS proposto na seção anterior. Estas decisões são específicas para concretizar o desenvolvimento do Sistema de Gerenciamento de Workflow na plataforma PHP/MySQL compatível com o ambiente AdaptWeb, levando em consideração as premissas estabelecidas pelo projeto e as necessidades imediatas do ambiente.

Como um grande esforço deste trabalho se concentrou no projeto do WFMS e da sua integração com o ambiente AdaptWeb, até a sua conclusão não foi possível levar a implementação a um estágio avançado, mesmo considerando o escopo mais reduzido que foi definido. No entanto, é descrita a seguir uma série de decisões de arquitetura de software que foram tomadas para balisar o desenvolvimento inicial.

4.4.1 Escopo da Implementação Inicial

Como já comentado, um Sistema de Gerenciamento de Workflow completo, como projetado neste trabalho, é uma aplicação bastante grande e complexa. Como o objetivo principal deste trabalho é propor a integração do WFMS ao ambiente AdaptWeb, analisaram-se as necessidades que o modelo atual de cursos do AdaptWeb exige de um sistema de workflow, para determinar um escopo inicial de desenvolvimento que possibilite uma implementação piloto dessa integração.

A base para essa análise é a modelagem do workflow de um curso, descrita no capítulo 3, definindo então um subconjunto das funcionalidades descritas na seção 4.3 a ser implementado. A seguir, para cada componente do WFMS projetado, é apresentada a análise das funcionalidades necessárias.

4.4.1.1 Motor de Workflow

O fluxo dos cursos do AdaptWeb é bastante simples, perto da complexidade que pode atingir um processo que envolva muitos participantes e regras de transição complexas. Sendo assim, nem toda essa complexidade precisa estar disponível inicialmente.

Em primeiro lugar, dos tipos de atividades definidos pelo padrão da WfMC, somente as atividades manuais aparecem nos processos dos cursos do AdaptWeb. Essas atividades manuais são apoiadas por aplicações web. Eventualmente, alguma atividade automática de execução de rotinas PHP também pode aparecer, como é proposto no capítulo 5, assim como atividades de roteamento. Sendo assim, os tipos de atividade que precisam estar disponíveis na versão inicial do WFMS são manual, implementação por aplicação (automática) ou de roteamento. Subprocessos e blocos de atividades não são necessárias em uma primeira implementação. A criação, execução e conclusão de itens de trabalho também precisa ser suportada, para dar origem às listas de trabalho dos alunos.

Com relação aos dados relevantes e parâmetros de processos e aplicações, os tipos de dados “participante” e “string” são utilizados no workflow do curso de exemplo do AdaptWeb. Portanto, é necessário suportar o armazenamento de dados destes tipos inicialmente. Apenas a passagem de parâmetros de entrada é necessária no workflow modelado, não sendo necessária a passagem de parâmetros de saída ou entrada/saída na versão inicial do Motor de Workflow.

Quanto às transições entre as atividades e suas regras de sincronização, o fluxo definido para o aprendizado de um aluno exige apenas algumas regras. Não existem transições condicionais, nem junções do tipo “OR” no fluxo. No entanto, para dar flexibilidade e poder à modelagem dos processos no AdaptWeb, já na elaboração dos primeiros cursos, é importante que essas funcionalidades estejam disponíveis. Com isso, o Motor de Workflow deve prover transições condicionais e incondicionais, bem como restrições de transições dos tipos *AND-split*, *AND-join*, *OR-split* e *OR-join*. Por conseqüência, é também necessária a avaliação de expressões correspondentes às condições.

Em resumo, as funcionalidades que o Motor de Workflow deve implementar já em sua primeira versão são as seguintes:

- criação e controle do ciclo de vida de instâncias de processo;
- criação e controle do ciclo de vida de atividades dos tipos manual, automática e roteamento;
- criação e controle do ciclo de vida de itens de trabalho;
- armazenamento e recuperação de dados relevantes dos tipos String e Participante;
- passagem de parâmetros de entrada para o processo e aplicações;
- avaliação de expressões condicionais;
- transições condicionais, incondicionais e de exceções;
- divisões e sincronizações no fluxo dos tipos *AND-split*, *AND-join*, *OR-split* e *OR-join*.

4.4.1.2 Gerenciador de Definições de Processo

A principal função do Gerenciador de Definições de Processo é buscar no banco de dados a descrição dos processos em XPDL, efetuar a interpretação dos arquivos e devolver ao componente que fez a solicitação (normalmente o Motor de Workflow) uma estrutura de dados que represente a definição.

Esta tarefa é primordial e necessita estar implementada antes de qualquer outra no sistema, pois sem ela não é possível conhecer o fluxo a ser coordenado. Desta forma, todo o processamento (*parse*) de um arquivo XPDL deve estar disponível, consistindo da interpretação dos elementos XML que descrevem o pacote, os processos, as atividades, aplicações, participantes, dados relevantes e todos os demais componentes descritos no padrão da WfMC.

Além disso, o Gerenciador de Definições de Processo necessita ao menos de uma rotina para a inclusão de novas definições de processo. O controle do versionamento dos processos e a atualização ou remoção de definições de processo não é uma funcionalidade fundamental na primeira versão do sistema.

A estrutura das descrições de processos em XPDL permite que sejam referenciadas definições de processos externas, ou seja, em outros pacotes fora do que está especificado no documento XPDL que está sendo processado. Esta também é uma característica que não necessita suporte imediato do sistema, já que não deverá ser utilizada na definição dos processos de workflow dos cursos do AdaptWeb.

4.4.1.3 Gerenciador de Listas de Trabalho

Conforme descrito mais adiante, no capítulo 5, a proposta de integração do Sistema de Gerenciamento de Workflow com o AdaptWeb prevê a implementação de tela em que o aluno poderá consultar a sua lista de trabalho, consistindo de tarefas pendentes ou do histórico de tarefas já executadas.

Ao acessar uma das tarefas da sua lista, o participante do workflow poderá ainda visualizar as informações sobre ela, listar as aplicações associadas e executá-las. Desta forma, o aluno poderá invocar a tela do AdaptWeb que apresenta o material instrucional dos cursos.

A partir dessas necessidades, o módulo de Gerenciamento de Listas de Trabalho deve permitir pelo menos a consulta sobre a base de itens de trabalho, para montar as listas de tarefas. Deve implementar também a consulta aos detalhes de cada tarefa e às suas listas de aplicações associadas. O módulo precisa ainda fornecer as rotinas para invocar aplicações associadas aos itens de trabalho.

4.4.1.4 Gerenciador de Aplicações

Como citado anteriormente, o Motor de Workflow irá permitir, já na implementação inicial, a execução de atividades automáticas e de itens de trabalho com aplicações associadas. De acordo com o projeto do WFMS, são três os tipos de aplicações que poderão ser invocadas: aplicativos web, rotinas nativas e serviços web.

Na versão inicial do módulo de Gerenciamento de Aplicações, deverá ser suportada a chamada de aplicativos web, para que os alunos que participam dos processos dos cursos do AdaptWeb possam abrir a aplicação de consulta ao material instrucional do tópico que estão estudando.

Além desse tipo de aplicação, para as atividades automáticas que o Motor de Workflow permitirá executar, o Gerenciador de Aplicações permitirá a chamada de rotinas nativas

em PHP.

A implementação da chamada de serviços web, por ser mais complexa, poderá ser disponibilizada em uma versão futura deste módulo.

4.4.1.5 Gerenciador de Entidades

O workflow do curso do AdaptWeb prevê, segundo a modelagem já apresentada, a existência de uma série de participantes que são os alunos matriculados em cada disciplina. Além disso, todos os alunos deverão ao menos fazer parte de um papel no modelo organizacional, denominado “Aluno”.

Em função dessas definições, o módulo Gerenciador de Entidades deverá possibilitar inicialmente o cadastramento e a consulta a entidades do tipo “Pessoa” e o cadastramento de papéis, bem como sua associação a essas pessoas. Deverá ainda possibilitar o cadastramento e consulta de atributos dessas entidades, para armazenar informações como o nome completo e a senha de cada usuário.

4.4.1.6 Gerenciador de Segurança

Ainda que o planejamento futuro seja de implantar os primeiros processos de workflow no AdaptWeb em caráter experimental, é necessária uma mínima preocupação com a segurança do sistema. Por isso, a senha de cada usuário será cadastrada no Gerenciador de Entidades, e o módulo de Segurança deverá prover uma rotina de estabelecimento de sessão que efetue a validação desta senha, bem como uma rotina de verificação da validade dos identificadores de sessão utilizados nas chamadas das APIs.

Além da autenticação, as rotinas de autorização também necessitam ser implementadas, possibilitando ao menos a verificação do perfil de Administrador de um usuário, para permitir ou não que o mesmo execute operações de criação e cancelamento de processos ou acessos aos módulos de Gerenciamento de Entidades, Aplicações e Definições de Processos.

As APIs implementadas de todos os demais módulos deverão efetuar chamadas ao módulo de segurança para verificar a validade do identificador de sessão recebido e a autorização básica do usuário para executar a operação, pelo menos nos casos em que um perfil de administrador é exigido para a tarefa.

As funcionalidades de registros de auditoria (*logs*) do módulo de Segurança não são essenciais na versão inicial do WFMS, devendo ser implementadas nas versões seguintes.

4.4.2 Componentes de Software

O PHP é uma linguagem interpretada, em que o código fonte está em arquivos que normalmente possuem a extensão “.php”. O código pode estar distribuído em vários arquivos. Para permitir que a chamada feita a um arquivo-fonte PHP possa utilizar código presente em outros arquivos, a linguagem possui as diretivas `include`, `include_once`, `require` e `require_once`, que declaram o uso de código vindo desses outros arquivos. A diferença entre essas diretivas indica se a presença do arquivo é imprescindível e ainda se o código presente nesses arquivos deve ser interpretado mais de uma vez na mesma execução.

Na implementação do Sistema de Gerenciamento de Workflow previsto neste trabalho, cada módulo terá sua API declarada em um arquivo “.php” diferente, mas único para cada um. A implementação interna de cada módulo poderá ser dividida em diversos arquivos, podendo inclusive existir arquivos compartilhados pela implementação de mais de um

módulo. No entanto, para que as aplicações que farão uso das APIs possam identificar claramente quais os arquivos que necessitam incluir, elas estarão em arquivos claramente diferenciados para cada módulo.

A utilização das APIs deverá se dar através do uso da diretiva `require_once` ou `include_once`, que indicam que o código-fonte referenciado deve ser interpretado apenas uma vez. A presença dessas diretivas fará com que as APIs do respectivo módulo do WFMS estejam disponíveis para o uso na aplicação PHP cliente.

4.4.3 Manipulação de Tipos de Dados

O valor de um dado do processo será sempre armazenado no sistema como um par <tipo de dado, valor do atributo>, onde o valor do atributo será uma representação *string* do valor para o tipo de dado correspondente.

No momento de utilizar o valor para realizar operações ou passar como parâmetro a rotinas automáticas, o sistema irá convertê-lo da forma apropriada para uma variável do tipo correspondente na linguagem PHP.

Todos os dados relevantes do workflow e parâmetros do processo ou atividade, quando necessários, serão copiados temporariamente em variáveis com os mesmos nomes, já com o tipo de dados convertido adequadamente, e válidas no escopo da rotina que avalia expressões do workflow. Desta forma, na avaliação das expressões, esses dados ou parâmetros estarão disponíveis no formato `$nome_da_variavel`, uma vez que as variáveis no PHP recebem como prefixo o caracter `$`.

Por exemplo, se um processo possui dois dados relevantes chamados “LIMITE” e “CUSTO_TOTAL”, pode-se verificar se o primeiro é maior ou igual ao segundo através da seguinte expressão: `$LIMITE >= $CUSTO_TOTAL`.

A avaliação de expressões em PHP irá ocorrer através do mecanismo propiciado pelo comando `eval`. A expressão completa será colocada em uma variável e submetida ao referido comando, que executa dinamicamente um trecho de código PHP.

No caso da avaliação de expressões para transições condicionais, a expressão sempre necessita de um resultado binário (*boolean*), ou seja, verdadeiro ou falso. Nesses casos, independente do tipo de dado resultante da avaliação da expressão, o resultado será convertido para o tipo *boolean* da linguagem PHP e, portanto, será considerado verdadeiro ou falso conforme a regra de conversão de tipos desta linguagem (PHP HOMEPAGE, 2005).

4.5 Estágio de Desenvolvimento Atual

Conforme já mencionado, o trabalho de projeto e planejamento da integração do Sistema de Gerenciamento de Workflow descrito neste trabalho foi bastante extenso. Isso não permitiu que a implementação do sistema pudesse ser realizada até um estágio avançado.

Durante o trabalho, foi iniciada uma implementação a partir de um projeto inicial, mais simples do que o apresentado neste capítulo. A partir de algumas dificuldades encontradas, o padrão da WfMC foi explorado em mais detalhes e o projeto do WFMS foi refinado, chegando-se ao formato final já descrito. A implementação inicial foi então revista.

A implementação inicial já desenvolvida consta de:

- uma parte do módulo de Gerenciamento de Definições de Processo, em que um arquivo XPDL qualquer pode ser lido de um arquivo e interpretado, resultando

em um objeto em memória que representa toda a definição do pacote, dos seus processos e suas atividades;

- uma parte do módulo de Motor de Workflow, que permite a instanciação de um processo, ainda sem parâmetros, a instanciação de atividades (ainda sem implementação) e o seu fluxo através de transições condicionais e incondicionais;
- avaliação de restrições de transições de saída dos tipos AND-split e XOR-split, combinadas com a avaliação de expressões condicionais das transições;
- um mecanismo de avaliação de expressões, que permite o uso de dados relevantes do processo e execução de funções PHP. Este mecanismo pode ser utilizado na avaliação das expressões das transições condicionais e na execução de aplicações do tipo nativas PHP.

Esta implementação é bastante preliminar e ainda carece de testes e complementação, conforme a descrição do escopo apresentada anteriormente. A conclusão desta implementação está elencada no capítulo de conclusão deste trabalho como uma das oportunidades de continuidade deste trabalho no futuro.

5 INTEGRAÇÃO COM O AMBIENTE ADAPTWEB

Uma vez definidos o projeto e a forma de implementação do Sistema de Gerenciamento de Workflow proposto, este capítulo apresenta propostas de como integrar esta ferramenta ao ambiente de ensino-aprendizagem AdaptWeb e como utilizar a tecnologia de workflow para aprimorar o seu funcionamento e até mesmo estender as funcionalidades atuais do ambiente.

5.1 Modo Dirigido de Aprendizado

No capítulo 3, foi descrito que o AdaptWeb possui atualmente dois modos distintos de navegação por parte do aluno: o modo *livre* e o modo *tutorial*. O primeiro permite que o usuário acesse livremente qualquer tópico da disciplina em que estiver matriculado; o segundo libera acesso a cada tópico apenas depois que o aluno houver consultado o material instrucional de todos os seus pré-requisitos.

Em ambos os casos, o módulo de “Interface Adaptativa” do AdaptWeb é o responsável por montar na tela os elementos de navegação do aluno pelo material instrucional dos tópicos que ele está habilitado a acessar. Para executar esta tarefa, são levados em consideração o “modelo do aluno” (informações sobre características pessoais e histórico navegacional) e opções ligadas ao curso (como os pré-requisitos da disciplina e o modo de navegação escolhido).

Durante a elaboração do AdaptWeb, não foi considerada a construção de um modo de interação baseado em workflow. A tecnologia de workflow centra-se basicamente na idéia de seqüenciamento e alocação de atividades (ou tarefas). Essa idéia se presta muito bem para o ensino à distância, especialmente no caso do AdaptWeb, uma vez que é muito clara a identificação de tarefas durante o curso de uma disciplina, bem como de seus responsáveis e as suas regras de dependência.

Foram também descritos no capítulo 3 o processo de modelagem do workflow de um curso do AdaptWeb e a sua especificação na XPDL, o formato de intercâmbio de definições de processo da WfMC. O workflow especificado nessa linguagem representa o processo de aprendizado individual de um aluno em uma disciplina, considerando cada um dos tópicos como uma atividade e utilizando a relação de pré-requisitos para definir a rede de transições entre as tarefas.

Do ponto de vista de um usuário de workflow, ou seja, de alguém que seja participante de um ou mais processos de workflow, o ponto central de utilização do sistema é uma “lista de tarefas”. Lá o usuário irá encontrar, em ordem de prioridade, de prazo ou como quiser, uma relação de tudo o que ele tem que fazer. Esta é a idéia de um ponto único de acesso, que serve como uma lista de pendências e facilita o controle das atividades, diminuindo os atrasos e evitando que a pessoa esqueça ou abandone alguma de

suas incumbências por falta de organização.

Na visão do administrador de um processo, interessado em acompanhar o andamento das tarefas de cada instância do fluxo que ele gerencia, o workflow permite que ele esteja sempre ciente do ponto em que cada processo está. A partir das informações armazenadas na meta-base de dados do workflow, o administrador pode extrair de estatísticas ou relatórios sobre tudo o que acontece nos processos, possibilitando identificar gargalos e definir ações para aprimorar o fluxo.

O modo tutorial, de certa forma, tem algumas características de workflow, uma vez que a tendência é que a ordem de acesso aos itens obedeça à rede de relacionamentos estabelecida pelos pré-requisitos. No entanto, o sistema não oferece ao aluno a visão de uma lista de tarefas unificada. Se o aluno estiver matriculado em várias disciplinas, ele precisa necessariamente acessar cada uma delas e ver quais tópicos estão pendentes. O aluno também não vê facilmente a informação de qual dos tópicos está habilitado há mais tempo, especialmente se forem de disciplinas diferentes.

Além disso, no modo tutorial, há um problema para controlar o efetivo cumprimento das tarefas, já que basta que o aluno acesse uma vez o conteúdo instrucional de um tópico para que tenha acesso aos tópicos seguintes, mesmo sem efetivamente ter concluído o estudo. O aluno não afirma explicitamente que aprendeu o conceito relacionado ao tópico.

Como forma de utilizar o potencial da tecnologia de workflow em prol do processo de ensino-aprendizagem no ambiente AdaptWeb, é proposta a seguir a criação de um novo modo de utilização do sistema pelo aluno, denominado de *modo dirigido*.

O *modo dirigido* baseia-se na execução de um processo de workflow correspondente à matrícula de um aluno em uma disciplina, conforme descrito anteriormente, e propõe o uso de uma lista de tarefas como ponto central de interação do aluno com o ambiente. Desta forma, o estudante é guiado de forma mais explícita e direcionada, através da atribuição de atividades referentes a cada tópico a ser estudado.

A figura 5.1 demonstra os componentes agregados à arquitetura do AdaptWeb com a utilização do sistema de gerenciamento de workflow para criação do modo dirigido de aprendizado.

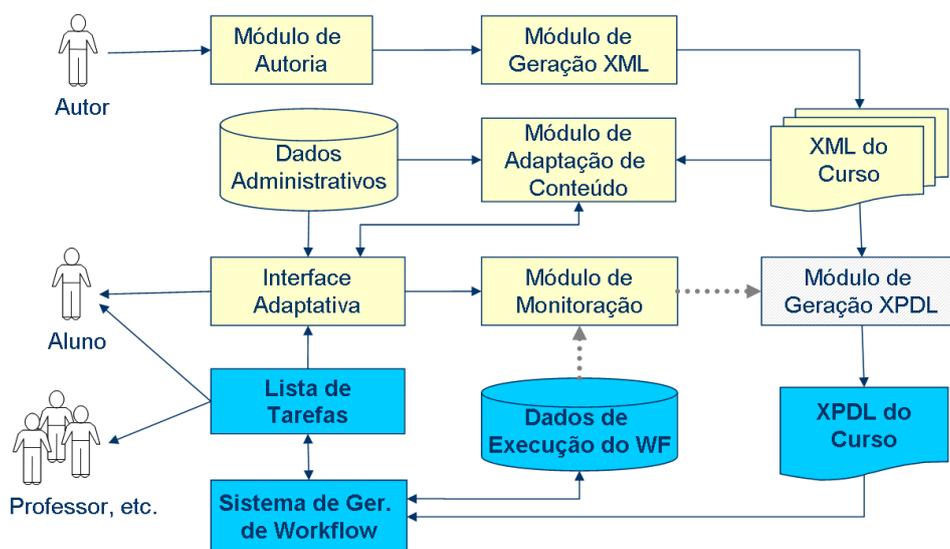


Figura 5.1: Arquitetura do AdaptWeb com os Componentes de Workflow

5.1.1 Disparo do Processo

Todo processo de workflow deve ser iniciado, seja explicitamente por algum usuário, usando uma interface de disparo de processos, seja por uma aplicação, através de chamadas à API do motor de workflow, o que é mais comum. O processo de workflow do AdaptWeb corresponde à participação de um aluno em uma disciplina. Desta forma, o disparo do processo deve acontecer quando a matrícula é efetuada, que é o momento em que o aluno pode começar a executar suas tarefas.

A rotina de disparo do workflow deve escolher qual é a definição de processo que será utilizada pela nova instância a ser iniciada. Cabe ao AdaptWeb estabelecer a associação entre as disciplinas e cursos do sistema e as definições de processo armazenadas no repositório do workflow.

Se cada disciplina possui uma única definição de processo associada, a escolha é trivial. Basta disparar o processo correspondente àquela disciplina. Nada impede, no entanto, que cada disciplina tenha várias definições de processo alternativas. É o caso, por exemplo, de existirem fluxos diferentes da disciplina de Computação Algébrica e Numérica para os cursos de Computação, Engenharia ou Matemática. Neste caso, a escolha do fluxo apropriado depende do curso ao qual o aluno está vinculado.

Pode também ser o caso de existirem definições de processos da mesma disciplina para estilos de alunos diferentes. Podem ser construídos, por exemplo, um fluxo predominantemente seqüencial e outro, alternativo, com várias tarefas em paralelo.

No disparo, a nova instância de processo de workflow deve receber os parâmetros que a definição de processo estabelece. No caso da definição do curso de Computação Algébrica e Numérica elaborado anteriormente, o parâmetro a ser passado é a identificação (*username*) do aluno no sistema.

5.1.2 Lista de Tarefas

A lista de tarefas será o componente principal de interação do aluno com o sistema no modo dirigido. Nela serão apresentadas ao aluno todas as atividades em aberto designadas para ele, ou seja, todos os tópicos que ele tem que estudar. Nesta interface, o aluno poderá obter informações mais detalhadas sobre a tarefa que ele deverá realizar e também terá acesso ao material instrucional referente ao tópico que ele vai estudar.

A tela de lista de tarefas deverá ser construída dentro do ambiente AdaptWeb, utilizando-se das APIs do módulo de Lista de Trabalho do sistema de gerenciamento de workflow.

A figura 5.2 ilustra um protótipo da interface da lista de tarefas, elaborado a partir da versão de demonstração do AdaptWeb, disponível no site do projeto (ADAPTWEB, 2004).

Neste exemplo, são apresentadas três tarefas para o aluno. As duas primeiras são os tópicos 1 e 2 do curso de Computação Algébrica e Numérica. Como foi visto na elaboração da definição do processo de exemplo, essas são as duas primeiras tarefas do fluxo, que podem ser executadas em paralelo. A outra tarefa que aparece na lista é de uma outra disciplina, ilustrando o caso em que o aluno está matriculado em mais de uma delas.

As informações apresentadas para cada tarefa são obtidas unicamente a partir dos dados do workflow. A linha inicial da coluna “Tarefa”, em negrito, é o nome da atividade, como foi definida no XPDL do processo. A linha abaixo é a descrição da atividade, definida no mesmo local. Por fim, o link para o material instrucional é montado em função da associação do respectivo aplicativo à atividade na definição de processo. A coluna de “disciplina” indica na verdade qual o tipo de processo de workflow que foi

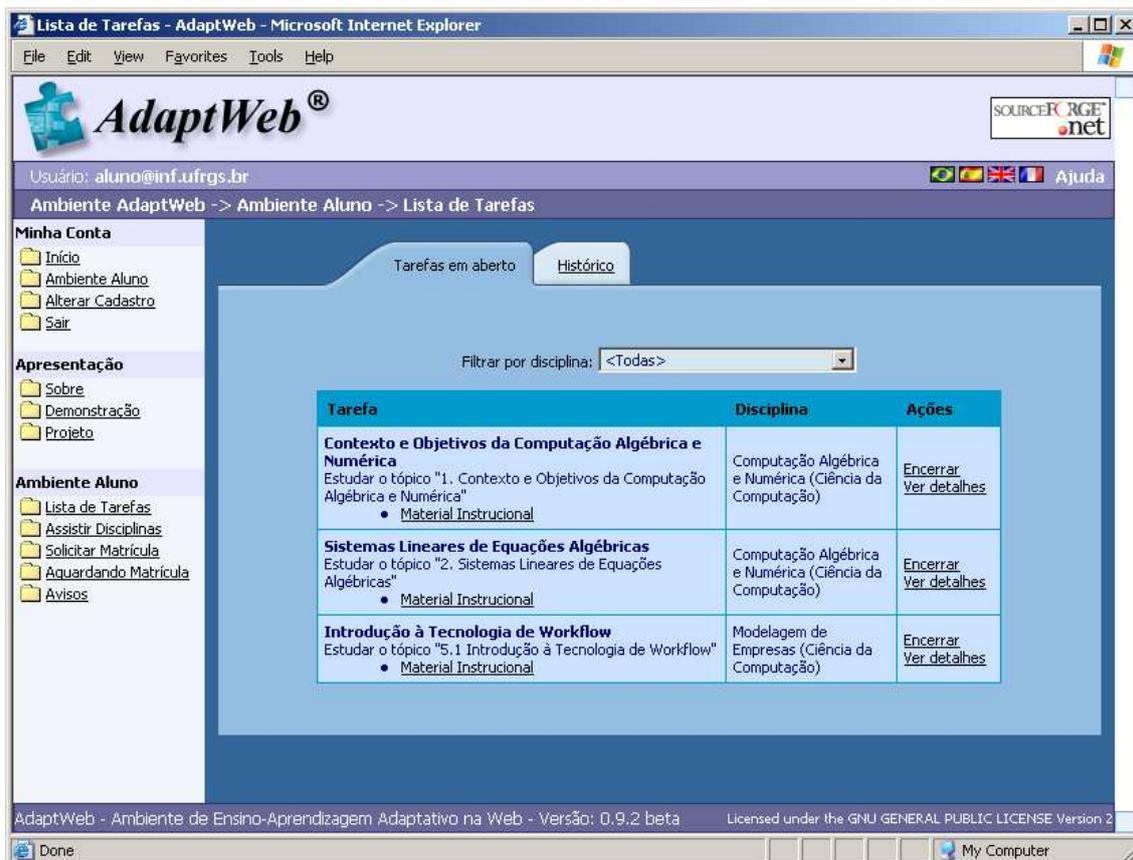


Figura 5.2: Protótipo da tela da Lista de Tarefas no AdaptWeb

disparado.

Como forma de auxiliar o aluno a organizar seu trabalho, a lista de tarefas pode oferecer diferentes formas de ordenação, como por data de chegada ou por disciplina, ou filtros para que o aluno visualize apenas um subconjunto das suas pendências.

Na coluna “Ações”, são apresentadas as possibilidades do aluno com relação a essa tarefa. A principal delas é o link “Encerrar”, onde poderá indicar para o sistema que foi concluída a execução da tarefa. Neste caso, será invocada a API do sistema de workflow que encerra o item de trabalho, disparando a avaliação das transições. Logo a seguir, deverão surgir na lista de trabalho as atividades correspondentes aos próximos tópicos habilitados para o usuário.

O link “Detalhes”, também presente na coluna de ações do protótipo da lista de tarefas, seria uma possibilidade de fornecer ao aprendiz maiores informações a respeito da execução daquela tarefa. Neste caso, essas informações também são exclusivamente relacionadas ao fluxo de trabalho e não às informações do AdaptWeb. A figura 5.3 apresenta um exemplo do que poderia conter essa tela.

Além das informações já apresentadas na tela anterior, são apresentadas informações adicionais sobre a execução e a definição da atividade. A data de início indica o momento a partir do qual o tópico passou a estar habilitado para que o aluno o estudasse. Outras informações, como prazo para conclusão da tarefa, e prioridade podem ser apresentados.

Essa tela de detalhes pode ser apresentada não apenas para as tarefas em aberto, que aparecem na lista de tarefas, mas também para tarefas já encerradas, canceladas ou suspensas, que podem ser consultadas através de outros aplicativos. Por isso, é interessante que a tela de detalhes exiba também a situação da atividade e sua data de conclusão, por

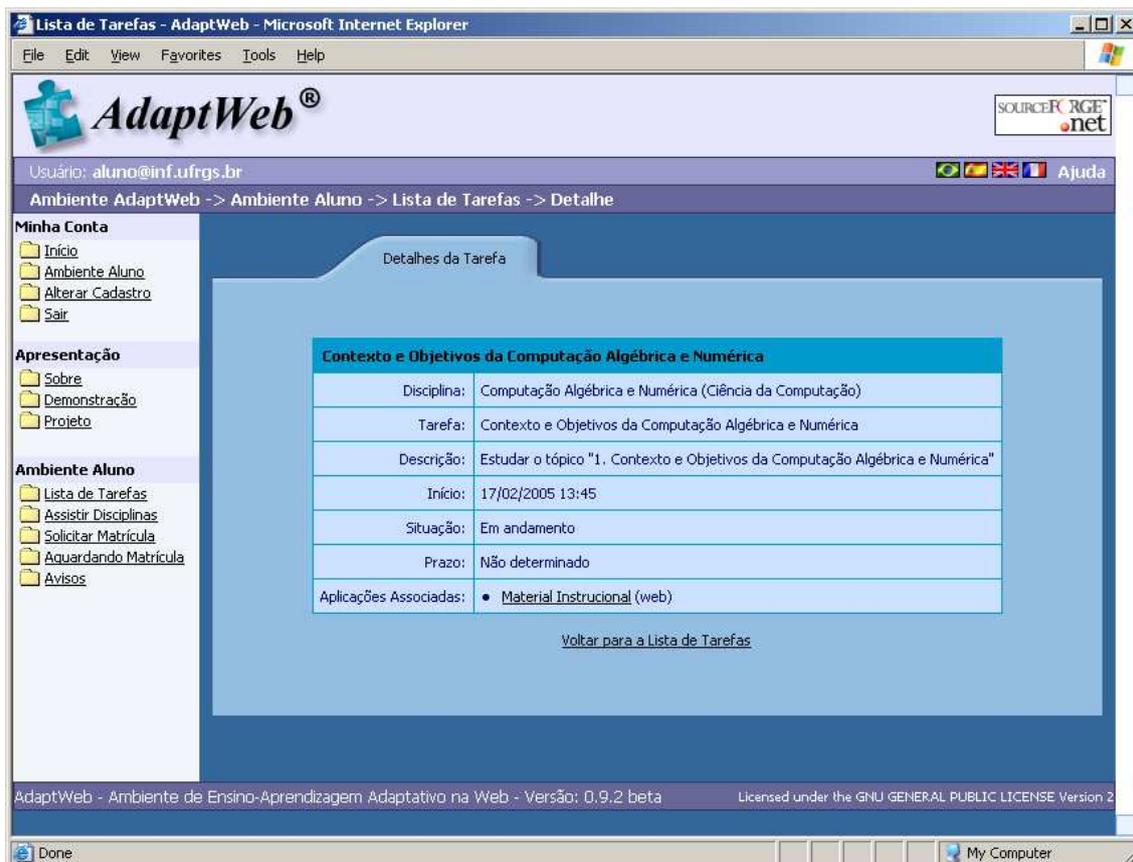


Figura 5.3: Protótipo da tela de detalhe da tarefa no AdaptWeb

exemplo.

5.1.3 Acesso ao Material Instrucional

Como pode ser percebido, a lista de tarefas em si não apresenta nenhuma informação específica sobre o conteúdo do tópico a ser estudado. Ela representa somente a orientação para que o aluno saiba quais estão disponíveis e decida o que vai fazer.

O link para a aplicação “Material Instrucional” apresentado em cada tarefa da lista deve levar o aluno para a mesma tela que ele teria se acessasse a disciplina por um dos modos já existentes (tutorial ou livre). Lá, ele poderá consultar o conceito, os exemplos, os exercícios e o material complementar referente ao tópico. Poderá também revisar os outros tópicos que ele já estudou, navegando pelo índice de tópicos da disciplina.

A grande diferença com relação aos demais modos de navegação do AdaptWeb é que, no modo dirigido, o fato de navegar pelos tópicos não irá liberar nenhum outro tópico para o aluno. Cada tópico só estará habilitado no momento que o aluno o acessar através da lista de tarefas do modo dirigido.

A URL utilizada para invocar a tela de material instrucional é montada pelo módulo de Aplicações do sistema de workflow. A URL é estabelecida a partir do cadastro da aplicação nesse módulo, combinado com os parâmetros passados pela atividade para a aplicação, conforme estabelecido na definição do processo. A tela de lista de tarefas do AdaptWeb irá obter essa URL através das APIs do módulo de lista de tarefas do workflow.

5.1.4 Histórico de Tarefas

Uma ferramenta complementar que pode ser muito útil, e que o uso da tecnologia de workflow possibilita, é a consulta do histórico da execução das atividades. Isso pode ser interessante tanto para o aluno, que pode visualizar mais claramente quando fez cada uma de suas atividades, como para o professor ou administrador do processo, que pode acompanhar o andamento de cada aluno ou disciplina.

A figura 5.4 mostra um protótipo da tela de histórico de tarefas no ambiente do aluno. Uma tela semelhante pode ser elaborada para o ambiente do professor, adicionando alguns campos a mais além dos apresentados neste exemplo, como um filtro por nome do aluno.

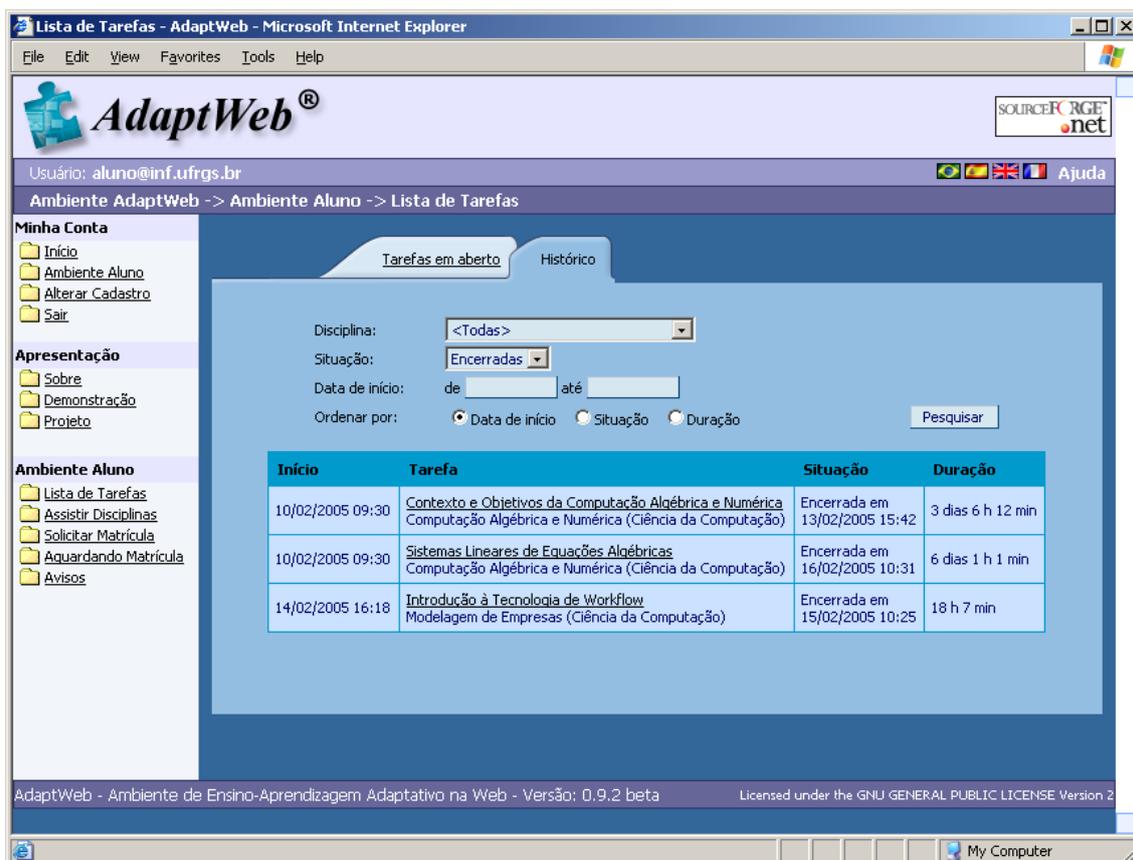


Figura 5.4: Protótipo da tela de histórico das tarefas no AdaptWeb

Neste protótipo, a tela consiste de uma série de campos de filtro para a pesquisa (disciplina, situação, data de início), que irão determinar o conjunto de dados a ser acessados na meta-base de dados do workflow, e de uma tabela que exibe os resultados. Cada linha do resultado corresponde a uma tarefa. São exibidas a data de início, o nome e a disciplina da tarefa, sua situação e a duração total da atividade, considerando desde o momento em que ela foi habilitada para o aluno até o seu encerramento.

A extração de estatísticas sobre a meta-base de dados do workflow também é possível e seria extremamente útil tanto para a consulta do professor e do autor do curso quanto para o próprio enriquecimento do modelo do aluno no AdaptWeb. Dados como o tempo médio de execução de cada tarefa, seu cruzamento com o grau de dificuldade de cada atividade, etc. seriam muito úteis. O sistema de gerenciamento de workflow projetado neste trabalho não contém APIs para a obtenção desses dados estatísticos, mas como os dados de execução do processo estão armazenados em banco de dados relacional, é

perfeitamente possível desenvolver ferramentas para a análise desses dados.

5.1.5 Aspectos de Adaptabilidade no Modo Dirigido

A questão da adaptabilidade ao modelo do usuário é sem dúvida um dos principais objetivos do ambiente de ensino proposto pelo AdaptWeb. Os módulos de “Adaptação do Conteúdo” e de “Interface Adaptativa” foram concebidos justamente para que o conteúdo e a interface apresentados aos aprendizes do sistema sejam adequados ao seu estilo cognitivo, definido a partir de informações explícitas ou da análise de seu histórico comportamental no sistema.

A criação do “modo dirigido” transfere algumas das incumbências desses módulos para o próprio motor de workflow, que é quem gerencia quais são as tarefas que serão apresentadas a cada momento para cada aluno. O fato de que o motor de workflow não tem acesso aos dados do modelo do aluno faz com que, em princípio, se perca um pouco do aspecto de adaptabilidade do ambiente, já que o processo de aprendizado tende a seguir sempre o mesmo andamento, independente do aluno.

No entanto, a utilização do workflow na verdade não diminui as possibilidades de adaptabilidade da ferramenta. Pelo contrário, agrega-se ainda mais flexibilidade para adequar a execução do processo de aprendizado às características do aluno.

Uma das principais formas de adaptabilidade é o fato de que uma mesma disciplina pode ter várias definições de processo, cada uma adequada a determinado estilo de aluno. Por exemplo, devem existir fluxos diferentes de uma disciplina quando ela é ministrada para cursos distintos. Podem também existir processos da mesma disciplina para estilos de alunos diferentes, como o exemplo já citado em que uma versão do processo pode ser predominantemente seqüencial e outra determine a execução de várias tarefas em paralelo. Dispondo dessas alternativas, o processo mais apropriado para o aprendiz poderá ser iniciado quando ele matricular-se na nova disciplina.

No decorrer do curso, pode-se identificar alguma nova característica no perfil do aluno que demande uma alteração no fluxo do seu aprendizado. Normalmente, uma instância de processo de workflow obedece a uma definição de processo constante, que não se altera desde seu disparo até sua conclusão. Nos casos em que se quer que a definição de processo mude durante o andamento dessa instância de processo, diz-se que é necessária a evolução do esquema do workflow, consistindo de um workflow dinâmico. Embora não tenha sido feito um projeto específico neste sentido, um dos trabalhos futuros propostos adiante é que seja construído um módulo de adaptação de processos, que permita a troca da definição do processo durante seu andamento, fazendo-se as respectivas validações e adequações nos dados de execução do processo. Essa característica dará ao sistema uma possibilidade virtualmente ilimitada de adaptação no processo de aprendizado do aluno.

5.2 Possibilidades de Extensão do Modelo de Autoria do AdaptWeb

Do modo como foi descrito até o momento, o workflow seria utilizado para coordenar o processo de aprendizado considerando apenas as atividades e o comportamento que o ambiente AdaptWeb permite atualmente.

O processo de aprendizado de um aluno no AdaptWeb, em uma disciplina, é bastante simples do ponto de vista da tecnologia de workflow. Basicamente, ele possui apenas um participante (o aluno) e as tarefas têm uma ordem bem definida, sem a possibilidade de que o aluno possa tomar decisões que influenciem nas transições do processo.

Essas restrições se dão em função do modelo de autoria adotado pelo AdaptWeb, que

corretamente se preocupa com os aspectos de adaptabilidade às características e preferências do aluno, mas que restringe o curso a um conjunto relativamente simples de ações. Ainda que, para cada tópico, um conjunto de materiais instrucionais esteja disponível, eles não são tratados como atividades do aluno, mas sim como insumos para qualificar o seu aprendizado, dentro do contexto de uma tarefa única, que é estudar o tópico.

A utilização do workflow para modelar o processo de aprendizado abre a possibilidade de se agregar uma série de novos tipos de tarefas e variações no fluxo do AdaptWeb.

A abordagem de lista de tarefas permite que sejam atribuídas ao aprendiz outras atividades, mesmo que não relacionadas diretamente com o estudo de tópicos das disciplinas. O programa da disciplina pode prever, por exemplo, uma atividade específica para que o aluno apresente algum relatório ou artigo. Ou então podem ser colocadas como tarefas no processo reuniões com o professor ou com o grupo. Essas tarefas iriam aparecer como as demais, na lista de tarefas do aluno.

O workflow abre também a possibilidade para a utilização de atividades automáticas, decisões e criação de tarefas para outros participantes no fluxo. Pode-se imaginar, por exemplo, que quando um aluno comunica ao sistema que encerrou a execução de determinada tarefa, uma atividade automática do sistema verifica qual foi seu aproveitamento nos exercícios referentes àquele conteúdo. Se houve um aproveitamento baixo, por exemplo, o aluno pode ser direcionado a fazer uma tarefa adicional de reforço ou o professor pode ser notificado a fazer um acompanhamento individualizado com este aluno. Neste caso, essas três novas características estariam sendo agregadas ao processo: um novo participante – o professor –, uma atividade automática e a criação de transições condicionais no fluxo. O sistema de workflow projetado permite todas essas construções.

Duas outras características vinculadas ao conceito de atividades no workflow também podem ser aproveitadas pelo AdaptWeb. Uma delas é o estabelecimento de prioridades para as tarefas. Elas poderiam ser definidas *a priori* pelo autor do curso, estabelecendo que determinadas atividades são mais importantes ou relevantes, dando uma indicação para o aluno sobre qual ele deve concentrar mais esforço quando tiver vários tópicos para estudar. A prioridade das tarefas poderia ser refletida na lista de tarefas do aluno através de indicadores por cor, podendo ser utilizada como critério de ordenação ou de filtro da lista de tarefas. O próprio aluno também poderia aumentar ou diminuir a prioridade das tarefas, conforme seu interesse para a organização do seu aprendizado.

A outra característica a ser utilizada seria a definição de prazos para a conclusão das tarefas. O autor do curso poderia determinar um tempo limite que considera adequado para o estudo de cada tópico, ou de algum tópico que mereça atenção especial. A lista de tarefas do aluno poderia permitir, então, a ordenação das tarefas por prazo, de forma que o mesmo evite atrasos. O fato de uma tarefa não ser concluída no prazo poderia gerar uma notificação para o professor, ou o desvio do fluxo para uma outra atividade de revisão, se o aprendizado necessita de uma continuidade que não foi obedecida.

No ensino à distância, é natural que se queira dar ao aluno a maior liberdade possível para que ele estude conforme o seu ritmo. A utilização de prazos não precisa necessariamente ser utilizada para restringir o tempo de execução das atividades. Pode ser estabelecido um tempo após o qual o sistema emite um alerta ou um aviso ao professor, notificando da demora na conclusão da tarefa, sem que isso interfira na lista de tarefas do aluno.

Mais um possível aproveitamento das características do sistema de gerenciamento de workflow no AdaptWeb seria o cadastramento de outras aplicações para darem apoio ao aluno no aprendizado de determinados tópicos. Segundo a modelagem feita neste tra-

balho, há sempre uma única aplicação vinculada a cada atividade, que é o acesso à página do material instrucional correspondente. O autor do curso poderia ter a possibilidade de associar outros programas à tarefa de workflow, permitindo que o aluno os executasse diretamente da lista de trabalho. Nem sempre isso será uma vantagem, uma vez que o próprio ambiente que apresenta o material instrucional já pode disponibilizar esse tipo de ferramenta. Mas é mais uma facilidade que está disponível para o ambiente.

Naturalmente, todas as novas características que o workflow pode proporcionar devem estar devidamente modeladas na descrição do processo em XPDL, que será utilizada pelo motor de workflow para coordenar o andamento do fluxo de trabalho e a montagem da lista de tarefas. A geração desta descrição de processo, que foi demonstrada de forma manual no capítulo 3, poderá ser automatizada em trabalhos futuros, através da criação de ferramentas que extraíam as informações necessárias a partir dos arquivos que descrevem a disciplina, oriundos do Módulo de Autoria do AdaptWeb. Por isso, essas características adicionais descritas devem encontrar suporte adequado no modelo de autoria do AdaptWeb, para que possam ser especificadas pelo autor do curso e armazenadas nos arquivos que descrevem as disciplinas, permitindo que a geração das descrições de processo de workflow possa ser feita automaticamente através de uma ferramenta.

Uma das áreas de pesquisa bastante estudadas com relação ao workflow é a evolução dinâmica de processos. Esta é uma outra característica que, com o devido suporte por parte do sistema de gerência de workflow, poderá trazer benefícios interessantes ao ambiente AdaptWeb. Atualmente, os aspectos de adaptabilidade do sistema são definidos através do Módulo de Interface Adaptativa, que filtra os elementos do curso que poderão ser apresentados ao aluno. Com o uso do workflow, isso passa a ser controlado pelo próprio desenho do processo, no caso de apresentar ao aluno quais os tópicos que devem ser estudados. No entanto, cada aluno pode possuir uma definição de processo própria, que pode evoluir constantemente de acordo com as entradas do módulo de Monitoramento do comportamento do aluno. Por exemplo, se o sistema detecta que o aluno possui determinada característica cognitiva, ele pode modificar o processo de workflow da disciplina durante seu andamento para aquele aluno, refletindo a percepção do seu comportamento ainda durante o curso.

Esta característica de evolução dinâmica de processos, bem como de avaliação do comportamento do aluno, não faz parte deste trabalho. Entretanto, isso dá mais uma demonstração do potencial de contribuição que o uso da tecnologia de workflow pode trazer ao ambiente AdaptWeb.

5.3 Adequação do Ambiente Tecnológico

Considerando-se que uma das motivações principais para o projeto deste novo Sistema de Gerenciamento de Workflow foi a sua utilização pelo AdaptWeb, a sua infra-estrutura de software foi escolhida especialmente de forma que fosse compatível com esse ambiente de ensino à distância. Como já descrito anteriormente, as principais definições foram: ambiente web, linguagem PHP e banco de dados MySQL.

A integração dos sistemas, no entanto, não implica em uma “fusão” completa dos códigos e dados. Não é necessário, nem conveniente, que os recursos de ambas as aplicações se misturem, uma vez que a aplicação de ensino à distância e seus dados são coisas independentes do sistema de workflow e os dados de definição e execução dos processos.

Com relação ao servidor de banco de dados, o ideal é que as tabelas do AdaptWeb e do Workflow estejam claramente separadas, para evitar descuidos na administração ou

problemas de segurança. O modelo de dados do sistema de workflow pode ser gerenciado no mesmo servidor MySQL, mas residindo em um repositório (banco de dados) separado. Nas versões mais recentes, o MySQL introduz o conceito de esquema (*schema*). Neste caso, a instalação das tabelas do workflow poderia ser feita no mesmo banco de dados, em esquemas diferentes.

Quanto ao código em PHP, os arquivos que compõem o WFMS (ou seja, a implementação dos módulos descritos no item 4.3) devem ficar em uma pasta separada no servidor web, não se confundindo com os arquivos do AdaptWeb. As novas telas propostas para o AdaptWeb, como a lista de tarefas e o histórico dos processos, serão partes do próprio AdaptWeb e poderão ser organizadas conforme o desejado pela sua equipe de desenvolvimento. Elas farão chamadas ao Sistema de Gerenciamento de Workflow através de suas APIs, fazendo “includes” do código PHP daqueles componentes do WFMS que forem necessários.

5.3.1 Cadastro de Usuários do Workflow

Como visto no capítulo 4, o WFMS apresentado possui um módulo Gerenciador de Entidades, onde devem ser cadastrados todos os possíveis participantes de processos de workflow, bem como a relação de papéis e/ou estrutura hierárquica das organizações envolvidas.

Para que o workflow possa ser utilizado no AdaptWeb, todos os alunos que forem utilizar o modo dirigido de aprendizado devem ser registrados no Gerenciador de Entidades. Um papel denominado "Aluno"(ou outro nome equivalente que for utilizado no arquivo XPDL dos processos) deverá ser criado e os aprendizes deverão estar associados a este papel.

6 CONCLUSÃO

Este capítulo apresenta os resultados obtidos com o presente trabalho, sua relação com os objetivos propostos e as lacunas e oportunidades de continuidade que foram deixadas para trabalhos futuros.

6.1 Resultados e Contribuições

A tecnologia de Workflow não é nova e muitos de seus benefícios já são largamente aproveitados em aplicações administrativas e de gestão de documentos há bem mais de uma década. Atualmente, entretanto, tem-se buscado ampliar os horizontes da utilização do Workflow para outras aplicações. A área de educação à distância é uma das que tem grande potencial para aproveitar as características de coordenação de processos que o Workflow proporciona.

Este trabalho buscou fornecer ao AdaptWeb – ambiente de ensino-aprendizagem adaptativo para a Web – a possibilidade de agregar um novo modo de funcionamento aos já existentes, baseado na tecnologia de Workflow.

Para tanto, foi realizado o projeto de um Sistema de Gerenciamento de Workflow completo, apropriado para ser utilizado por aplicações no ambiente PHP/MySQL, seguindo as recomendações da WfMC, entidade que estabelece os padrões para a área de Workflow. Foi dado início também a uma implementação deste Sistema, visando trazer em breve à realidade este projeto de integração.

O Sistema de Gerenciamento de Workflow projetado conseguiu atingir o objetivo de ser flexível e genérico, podendo ser utilizado por qualquer aplicação PHP com banco de dados MySQL, mesmo sem nenhuma relação com educação à distância. A XPDL – linguagem de intercâmbio de definições de processos de Workflow estabelecida pela WfMC – foi utilizada como linguagem nativa do motor de workflow, sem necessitar de nenhuma extensão adicional, de modo que qualquer descrição de processo elaborada em um editor compatível com o padrão pode ser utilizada. Ainda não foi encontrado até o momento, em 2005, projeto com características equivalentes para o ambiente PHP/MySQL com implementação disponível na Internet.

O trabalho também mostrou como a execução de um curso à distância por um aluno no AdaptWeb pode ser mapeada para um processo de Workflow e como pode ser feita a descrição deste processo na XPDL. Com isto, atingiu-se também o objetivo de demonstrar que a XPDL pode ser utilizada na modelagem de um processo de Workflow referente a um curso à distância, permitindo que o mesmo seja executado pelo Sistema de Gerenciamento de Workflow projetado.

Por fim, foi elaborada e apresentada uma proposta de extensão ao ambiente AdaptWeb para suportar um novo modo de interação, o *modo dirigido*, baseado no uso do Workflow,

em que o elemento central de interação do aluno é a sua lista de tarefas e o rumo do aprendizado é determinado pela descrição do fluxo da disciplina. Esta proposta consistiu de protótipos de novas telas para o sistema, da descrição da nova arquitetura e de inúmeras idéias para estender o ambiente de autoria e de execução do curso para aproveitar novas perspectivas que o Workflow apresenta.

6.2 Trabalhos Futuros

A grande maioria das contribuições deste trabalho consiste de projetos ou de idéias. Sendo assim, a partir dessas contribuições abre-se um grande leque de oportunidades para dar continuidade ao trabalho, seja implementando suas propostas, estendendo suas idéias ou preenchendo as lacunas.

As principais oportunidades de continuidade deste trabalho são as seguintes:

- **Implementação do Sistema de Gerenciamento de Workflow:** concluir o desenvolvimento do WFMS completo projetado neste trabalho, cuja implementação foi apenas iniciada.
- **Módulo de geração automática do Workflow no AdaptWeb:** desenvolver um módulo que consiga fazer a geração automática de uma descrição de processo em XPDL para qualquer curso do AdaptWeb, a partir das definições armazenadas nos arquivos XML do curso resultantes da fase de autoria.
- **Ampliar o modelo de autoria do AdaptWeb:** estudos para a ampliação do modelo de autoria do AdaptWeb, considerando as novas perspectivas que o uso do workflow pode proporcionar. Entre as novas perspectivas estariam a possibilidade de uso de prioridades e prazos para a execução das tarefas pelos alunos, definição de tarefas para o aluno relacionadas com exercícios ou atividades complementares e inclusão de tarefas para o professor, como avaliações ou correção de exercícios;
- **Implementação do “modo dirigido” no AdaptWeb:** implementação da interface proposta para o “modo dirigido” do AdaptWeb, bem como sua interação com o Sistema de Gerenciamento de Workflow;
- **Verificação de consistência para alteração dinâmica dos processos:** projeto e implementação de um módulo para controlar a alteração dinâmica de workflows de processos de aprendizado dos alunos, possibilitando a validação e, eventualmente, a readequação de processos em andamento;
- **Monitoramento dos metadados do Workflow:** utilização da meta-base de dados do Workflow para extrair informações sobre o comportamento dos alunos e sobre a própria dinâmica das disciplinas do AdaptWeb, usando essas informações para realimentar o modelo do aluno e até mesmo gerar modificações nas definições de processos de Workflow em andamento.

REFERÊNCIAS

- AALST, W. M. P. van der; ALDRED, L.; DUMAS, M.; HOFSTEDE, A. H. M. ter. Design and Implementation of the YAWL System. In: CAISE, 2004. **Proceedings...** Springer, 2004. p.142–159. (Lecture Notes in Computer Science, v.3084).
- AALST, W. M. P. van der; DUMAS, M.; HOFSTEDE, A. H. M. ter. Web Service Composition Languages: old wine in new bottles? In: EUROMICRO, 2003. **Proceedings...** IEEE Computer Society, 2003. p.298–307.
- AALST, W. M. P. van der; HOFSTEDE, A. H. M. ter; KIEPUSZEWSKI, B.; BARROS, A. P. Workflow Patterns. **Distributed and Parallel Databases**, [S.l.], v.14, n.1, p.5–51, 2003.
- ADAPTWEB. **Página do Projeto AdaptWeb**. Disponível em: <<http://www.inf.ufrgs.br/adapt/adaptweb.htm>>. Acesso em: abril 2004.
- ANDREWS, T. et al. **Business Process Execution Language for Web Services, version 1.1**. [S.l.]: Microsoft, IBM, BEA, Siebel, SAP, 2003. Disponível em: <<http://www.ibm.com/developerworks/library/ws-bpel/>>. Acesso em: agosto 2004.
- ARKIN, A. et al. **Web Service Coreography Interface (WSCI) 1.0**. [S.l.]: BEA, Intalio, SAP, Sun, 2002. Disponível em: <<http://www.w3.org/TR/wsci/>>. Acesso em: agosto 2004.
- BPMI.ORG. **BPML.org Specifications (homepage)**. Disponível em: <<http://www.bpmi.org/specifications.esp>>. Acesso em: agosto 2004.
- CASATI, F. et al. Conceptual Modelling of WorkFlows. In: OOER, 1995. **Proceedings...** Springer, 1995. p.341–354. (Lecture Notes in Computer Science, v.1021).
- FREITAS, V. de. **Autoria Adaptativa de Hipermídia Educacional**. Porto Alegre: UFRGS, 2003. 101p. Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/11486/000615801.pdf?sequence=1>>. Acesso em: 17 abr. 2009.
- FREITAS, V.; MARÇAL, V. P.; GASPARINI, I.; AMARAL, M. A.; PROENÇA JR., M. L.; BRUNETTO, M. A. C.; PIMENTA, M. S.; RIBEIRO, C. H. F. P.; LIMA, J. V.; OLIVEIRA, J. P. M. AdaptWeb: an adaptive web-based courseware. In: ICTE - INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGIES IN EDUCATION, 2002, Badajoz, Espanha. **Proceedings...** [S.l.: s.n.], 2002.

GASPARINI, I. **Interface Adaptativa no Ambiente AdaptWeb: navegação e apresentação adaptativa baseada no modelo do usuário.** Porto Alegre: UFRGS, 2003. Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2003. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/12076/000622639.pdf?sequence=1>>. Acesso em: 17 abr. 2009.

GEORGAKOPOULOS, D.; HORNICK, M. F.; SHETH, A. P. An Overview of Workflow Management: from process modeling to workflow automation infrastructure. **Distributed and Parallel Databases**, [S.l.], v.3, n.2, p.119–153, 1995.

HOLLINGSWORTH, D. **The Workflow Reference Model, issue 1.1.** Hampshire, United Kingdom: Workflow Management Coalition, 1995.

OASIS. **ebXML SPECS (homepage).** Disponível em: <<http://www.ebxml.org/specs/index.htm>>. Acesso em: agosto 2004.

ObjectWeb. **Enhydra JaWE Homepage.** Disponível em: <<http://jawe.objectweb.org>>. Acesso em: maio 2005.

PHP Homepage. Disponível em: <<http://www.php.net>>. Acesso em: maio 2005.

SANTOS, N.; PINTO, S. C. S.; ROCHA, A. R. C. **Navegação em Documentos Hiperídia: estado da arte.** Rio de Janeiro: COPPE/Sistemas/UFRJ, 1996. Disponível em: <<http://www.ime.uerj.br/professores/neidenew/Reltec.htm>>. Acesso em: abril 2004. (Relatório Técnico ES-373/96).

STAAB, S.; AALST, W. M. P. van der et al. Web Services: been there, done that? **IEEE Intelligent Systems**, [S.l.], v.18, n.1, p.72–85, 2003.

VIERO, D. M. **Sistemas de Workflow em Software Livre.** 2002. Trabalho Individual — Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002. (T.I. 1058).

Workflow Management Coalition. **Workflow Client Application (Interface 2) Application Programming Interface (WAPI) Specification, version 2.0e (Beta).** Hampshire, United Kingdom: Workflow Management Coalition, 1997. n.WFMC-TC-1009.

Workflow Management Coalition. **Audit Data Specification, version 1.1.** Hampshire, United Kingdom: Workflow Management Coalition, 1998. n.WFMC-TC-1015.

Workflow Management Coalition. **Terminology & Glossary, issue 3.0.** Hampshire, United Kingdom: Workflow Management Coalition, 1999.

Workflow Management Coalition. **Workflow Standard - Interoperability Wf-XML Binding, version 1.1.** Hampshire, United Kingdom: Workflow Management Coalition, 2001. n.WFMC-TC-1023.

Workflow Management Coalition. **Workflow Process Definition Interface – XML Process Definition Language, version 1.0.** Hampshire, United Kingdom: Workflow Management Coalition, 2002. n.WFMC-TC-1025.

Workflow Management Coalition. **WfMC Homepage**. Disponível em: <<http://www.wfmc.org>>. Acesso em: junho 2004.

Workflow Patterns Homepage. Disponível em: <<http://www.workflowpatterns.com>>. Acesso em: maio 2005.

World Wide Web Consortium. **Web Services Architecture (W3C Working Group Note 11)**. Disponível em: <<http://www.w3.org/TR/ws-arch/>>. Acesso em: maio 2005.

APÊNDICE A WORKFLOW DO CURSO EM XPDL

```

<?xml version="1.0" encoding="iso-8859-1"?>
<Package xmlns="http://www.wfmc.org/2002/XPDL1.0"
  xmlns:xpdl="http://www.wfmc.org/2002/XPDL1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.wfmc.org/2002/XPDL1.0
    http://wfmc.org/standards/docs/TC-1025_schema_10_xpdl.xsd"
  Id="0"
  Name="PKG_COMP_ALG_NUM">

  <PackageHeader>
    <XPDLVersion>1.0</XPDLVersion>
    <Vendor>Instituto de Informática, UFRGS</Vendor>
    <Created>2/4/2004 7:26:00 PM</Created>
    <Description>Disciplina de Computação Algébrica e Numérica</Description>
  </PackageHeader>

  <RedefinableHeader PublicationStatus="UNDER_TEST">
    <Author>Daniel de Mello Viero</Author>
    <Version>0.1</Version>
  </RedefinableHeader>

  <ConformanceClass GraphConformance="LOOP-BLOCKED"/>

  <Script Type="text/php"/>

  <Applications>
    <Application Id="APL_MATERIAL_INSTRUCIONAL"
      Name="Material Instrucional">
      <Description>
        Aplicação que permite visualizar o material instrucional referente ao tópico.
      </Description>
      <FormalParameters>
        <FormalParameter Id="curso" Index="1" Mode="IN">
          <DataType>
            <BasicType Type="STRING"/>
          </DataType>
        </FormalParameter>
        <FormalParameter Id="topico" Index="2" Mode="IN">
          <DataType>
            <BasicType Type="STRING"/>
          </DataType>
        </FormalParameter>
        <FormalParameter Id="aluno" Index="3" Mode="IN">
          <DataType>
            <BasicType Type="STRING"/>
          </DataType>
        </FormalParameter>
      </FormalParameters>
    </Application>
  </Applications>

  <WorkflowProcesses>
    <WorkflowProcess
      Id="PROC_COMP_ENG"
      Name="Computação Algébrica e Numérica para os Cursos de Computação e Engenharia"
      AccessLevel="PUBLIC">

```

```

<ProcessHeader />

<FormalParameters>
  <FormalParameter Id="PARAM_ALUNO" Index="1" Mode="IN">
    <DataType>
      <BasicType Type="PERFORMER"/>
    </DataType>
  </FormalParameter>
</FormalParameters>

<Activities>
  <Activity Id="ATIV_TOP_1"
    Name="Contexto e Objetivos da Computação Algébrica e Numérica">
    <Description>
      Estudar o tópico "1. Contexto e Objetivos da Computação Algébrica e Numérica"
    </Description>
    <Implementation>
      <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
        <ActualParameters>
          <ActualParameter>CSN</ActualParameter>
          <ActualParameter>1</ActualParameter>
          <ActualParameter>$PARAM_ALUNO</ActualParameter>
        </ActualParameters>
      </Tool>
    </Implementation>
    <Performer>$PARAM_ALUNO</Performer>
    <StartMode>
      <Manual />
    </StartMode>
    <FinishMode>
      <Manual />
    </FinishMode>
  </Activity>

  <Activity Id="ATIV_TOP_2"
    Name="Sistemas Lineares de Equações Algébricas">
    <Description>
      Estudar o tópico "2. Sistemas Lineares de Equações Algébricas"
    </Description>
    <Implementation>
      <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
        <ActualParameters>
          <ActualParameter>CSN</ActualParameter>
          <ActualParameter>2</ActualParameter>
          <ActualParameter>$PARAM_ALUNO</ActualParameter>
        </ActualParameters>
      </Tool>
    </Implementation>
    <Performer>$PARAM_ALUNO</Performer>
    <StartMode>
      <Manual />
    </StartMode>
    <FinishMode>
      <Manual />
    </FinishMode>
    <TransitionRestrictions>
      <TransitionRestriction>
        <Split Type="AND">
          <TransitionRefs>
            <TransitionRef Id="TRANS_2_PARA_21"/>
            <TransitionRef Id="TRANS_2_PARA_22"/>
            <TransitionRef Id="TRANS_2_PARA_23"/>
          </TransitionRefs>
        </Split>
      </TransitionRestriction>
    </TransitionRestrictions>
  </Activity>

  <Activity Id="ATIV_TOP_2_1"
    Name="Introdução">
    <Description>
      Estudar o tópico "2.1. Introdução"
    </Description>

```

```

<Implementation>
  <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
    <ActualParameters>
      <ActualParameter>CSN</ActualParameter>
      <ActualParameter>2.1</ActualParameter>
      <ActualParameter>$PARAM_ALUNO</ActualParameter>
    </ActualParameters>
  </Tool>
</Implementation>
<Performer>$PARAM_ALUNO</Performer>
<StartMode>
  <Manual />
</StartMode>
<FinishMode>
  <Manual />
</FinishMode>
</Activity>

<Activity Id="ATIV_TOP_2_2"
  Name="Métodos Diretos">
  <Description>
    Estudar o tópico "2.2. Métodos Diretos"
  </Description>
  <Implementation>
    <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
      <ActualParameters>
        <ActualParameter>CSN</ActualParameter>
        <ActualParameter>2.2</ActualParameter>
        <ActualParameter>$PARAM_ALUNO</ActualParameter>
      </ActualParameters>
    </Tool>
  </Implementation>
  <Performer>$PARAM_ALUNO</Performer>
  <StartMode>
    <Manual />
  </StartMode>
  <FinishMode>
    <Manual />
  </FinishMode>
  <TransitionRestrictions>
    <TransitionRestriction>
      <Split Type="AND">
        <TransitionRefs>
          <TransitionRef Id="TRANS_22_PARA_221"/>
          <TransitionRef Id="TRANS_22_PARA_223"/>
        </TransitionRefs>
      </Split>
    </TransitionRestriction>
  </TransitionRestrictions>
</Activity>

<Activity Id="ATIV_TOP_2_2_1"
  Name="Método de Gauss">
  <Description>
    Estudar o tópico "2.2.1. Método de Gauss"
  </Description>
  <Implementation>
    <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
      <ActualParameters>
        <ActualParameter>CSN</ActualParameter>
        <ActualParameter>2.2.1</ActualParameter>
        <ActualParameter>$PARAM_ALUNO</ActualParameter>
      </ActualParameters>
    </Tool>
  </Implementation>
  <Performer>$PARAM_ALUNO</Performer>
  <StartMode>
    <Manual />
  </StartMode>
  <FinishMode>
    <Manual />
  </FinishMode>
  <TransitionRestrictions>

```

```

        <TransitionRestriction>
            <Split Type="AND">
                <TransitionRefs>
                    <TransitionRef Id="TRANS_221_PARA_2211"/>
                    <TransitionRef Id="TRANS_221_PARA_2212"/>
                    <TransitionRef Id="TRANS_221_PARA_222"/>
                </TransitionRefs>
            </Split>
        </TransitionRestriction>
    </TransitionRestrictions>
</Activity>

<Activity Id="ATIV_TOP_2_2_1_1"
    Name="Algoritmo da Triangularização">
    <Description>
        Estudar o tópico "2.2.1.1. Algoritmo da Triangularização"
    </Description>
    <Implementation>
        <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
            <ActualParameters>
                <ActualParameter>CSN</ActualParameter>
                <ActualParameter>2.2.1.1</ActualParameter>
                <ActualParameter>$PARAM_ALUNO</ActualParameter>
            </ActualParameters>
        </Tool>
    </Implementation>
    <Performer>$PARAM_ALUNO</Performer>
    <StartMode>
        <Manual />
    </StartMode>
    <FinishMode>
        <Manual />
    </FinishMode>
</Activity>

<Activity Id="ATIV_TOP_2_2_1_2"
    Name="Algoritmo da Retrossubstituição">
    <Description>
        Estudar o tópico "2.2.1.2. Algoritmo da Retrossubstituição"
    </Description>
    <Implementation>
        <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
            <ActualParameters>
                <ActualParameter>CSN</ActualParameter>
                <ActualParameter>2.2.1.2</ActualParameter>
                <ActualParameter>$PARAM_ALUNO</ActualParameter>
            </ActualParameters>
        </Tool>
    </Implementation>
    <Performer>$PARAM_ALUNO</Performer>
    <StartMode>
        <Manual />
    </StartMode>
    <FinishMode>
        <Manual />
    </FinishMode>
</Activity>

<Activity Id="ATIV_TOP_2_2_2"
    Name="Método de Gauss com Pivotamento">
    <Description>
        Estudar o tópico "2.2.2. Método de Gauss com Pivotamento"
    </Description>
    <Implementation>
        <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
            <ActualParameters>
                <ActualParameter>CSN</ActualParameter>
                <ActualParameter>2.2.2</ActualParameter>
                <ActualParameter>$PARAM_ALUNO</ActualParameter>
            </ActualParameters>
        </Tool>
    </Implementation>
    <Performer>$PARAM_ALUNO</Performer>

```

```

    <StartMode>
      <Manual />
    </StartMode>
  </FinishMode>
  <Manual />
</FinishMode>
</Activity>

<Activity Id="ATIV_TOP_2_2_2_1"
  Name="Condicionamento de Matrizes">
  <Description>
    Estudar o t3pico "2.2.2.1. Condicionamento de Matrizes"
  </Description>
  <Implementation>
    <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
      <ActualParameters>
        <ActualParameter>CSN</ActualParameter>
        <ActualParameter>2.2.2.1</ActualParameter>
        <ActualParameter>$PARAM_ALUNO</ActualParameter>
      </ActualParameters>
    </Tool>
  </Implementation>
  <Performer>$PARAM_ALUNO</Performer>
  <StartMode>
    <Manual />
  </StartMode>
  <FinishMode>
    <Manual />
  </FinishMode>
</Activity>

<Activity Id="ATIV_TOP_2_2_3"
  Name="M3todo da Decomposi3o LU">
  <Description>
    Estudar o t3pico "2.2.3. M3todo da Decomposi3o LU"
  </Description>
  <Implementation>
    <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
      <ActualParameters>
        <ActualParameter>CSN</ActualParameter>
        <ActualParameter>2.2.3</ActualParameter>
        <ActualParameter>$PARAM_ALUNO</ActualParameter>
      </ActualParameters>
    </Tool>
  </Implementation>
  <Performer>$PARAM_ALUNO</Performer>
  <StartMode>
    <Manual />
  </StartMode>
  <FinishMode>
    <Manual />
  </FinishMode>
</Activity>

<Activity Id="ATIV_TOP_2_2_4"
  Name="M3todo de Cholesky">
  <Description>
    Estudar o t3pico "2.2.4. M3todo de Cholesky"
  </Description>
  <Implementation>
    <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
      <ActualParameters>
        <ActualParameter>CSN</ActualParameter>
        <ActualParameter>2.2.4</ActualParameter>
        <ActualParameter>$PARAM_ALUNO</ActualParameter>
      </ActualParameters>
    </Tool>
  </Implementation>
  <Performer>$PARAM_ALUNO</Performer>
  <StartMode>
    <Manual />
  </StartMode>
  <FinishMode>

```

```

        <Manual />
    </FinishMode>
</Activity>

<Activity Id="ATIV_TOP_2_3"
    Name="Métodos Iterativos">
    <Description>
        Estudar o tópico "2.3. Métodos Iterativos"
    </Description>
    <Implementation>
        <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
            <ActualParameters>
                <ActualParameter>CSN</ActualParameter>
                <ActualParameter>2.3</ActualParameter>
                <ActualParameter>$PARAM_ALUNO</ActualParameter>
            </ActualParameters>
        </Tool>
    </Implementation>
    <Performer>$PARAM_ALUNO</Performer>
    <StartMode>
        <Manual />
    </StartMode>
    <FinishMode>
        <Manual />
    </FinishMode>
</Activity>

<Activity Id="ATIV_TOP_2_3_1"
    Name="Método de Jacobi">
    <Description>
        Estudar o tópico "2.3.1. Método de Jacobi"
    </Description>
    <Implementation>
        <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
            <ActualParameters>
                <ActualParameter>CSN</ActualParameter>
                <ActualParameter>2.3.1</ActualParameter>
                <ActualParameter>$PARAM_ALUNO</ActualParameter>
            </ActualParameters>
        </Tool>
    </Implementation>
    <Performer>$PARAM_ALUNO</Performer>
    <StartMode>
        <Manual />
    </StartMode>
    <FinishMode>
        <Manual />
    </FinishMode>
</Activity>

<Activity Id="ATIV_TOP_2_3_2"
    Name="Método de Gauss-Seidel">
    <Description>
        Estudar o tópico "2.3.2. Método de Gauss-Seidel"
    </Description>
    <Implementation>
        <Tool Id="APL_MATERIAL_INSTRUCIONAL" Type="APPLICATION">
            <ActualParameters>
                <ActualParameter>CSN</ActualParameter>
                <ActualParameter>2.3.2</ActualParameter>
                <ActualParameter>$PARAM_ALUNO</ActualParameter>
            </ActualParameters>
        </Tool>
    </Implementation>
    <Performer>$PARAM_ALUNO</Performer>
    <StartMode>
        <Manual />
    </StartMode>
    <FinishMode>
        <Manual />
    </FinishMode>
</Activity>
</Activities>

```

```
<Transitions>
  <Transition Id="TRANS_2_PARA_21"      From="ATIV_TOP_2"      To="ATIV_TOP_2_1"/>
  <Transition Id="TRANS_2_PARA_22"      From="ATIV_TOP_2"      To="ATIV_TOP_2_2"/>
  <Transition Id="TRANS_2_PARA_23"      From="ATIV_TOP_2"      To="ATIV_TOP_2_3"/>
  <Transition Id="TRANS_22_PARA_221"    From="ATIV_TOP_2_2"    To="ATIV_TOP_2_2_1"/>
  <Transition Id="TRANS_22_PARA_223"    From="ATIV_TOP_2_2"    To="ATIV_TOP_2_2_3"/>
  <Transition Id="TRANS_221_PARA_2211"  From="ATIV_TOP_2_2_1"  To="ATIV_TOP_2_2_1_1"/>
  <Transition Id="TRANS_221_PARA_2212"  From="ATIV_TOP_2_2_1"  To="ATIV_TOP_2_2_1_2"/>
  <Transition Id="TRANS_221_PARA_222"   From="ATIV_TOP_2_2_1"  To="ATIV_TOP_2_2_2"/>
  <Transition Id="TRANS_222_PARA_2221"  From="ATIV_TOP_2_2_2"  To="ATIV_TOP_2_2_2_1"/>
  <Transition Id="TRANS_223_PARA_224"   From="ATIV_TOP_2_2_3"  To="ATIV_TOP_2_2_4"/>
  <Transition Id="TRANS_23_PARA_231"    From="ATIV_TOP_2_3"    To="ATIV_TOP_2_3_1"/>
  <Transition Id="TRANS_231_PARA_232"   From="ATIV_TOP_2_3_1"  To="ATIV_TOP_2_3_2"/>
</Transitions>

</WorkflowProcess>
</WorkflowProcesses>
</Package>
```