

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

CARLOS FRANCISCO GOMIDE

**Desenvolvimento de Workflow para Administração Pública em
Ambientes Corporativos de Engenharia de Software**

Trabalho de Conclusão apresentado como requisito
parcial para a obtenção do grau de Mestre em
Informática

Prof. Dr José Palazzo M. de Oliveira
Orientador

Porto Alegre, janeiro de 2004

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Gomide, Carlos Francisco

Desenvolvimento de Workflow para Administração Pública em Ambientes de Engenharia de Software / Carlos Francisco Gomide - Porto Alegre: Programa de Pós-Graduação em Computação, 2004.

109 fl.: il.

Trabalho de Conclusão (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre. BR-RS, 2004. Orientador: José Palazzo M. de Oliveira.

1. Workflow. 2. Automação de Processos. 3. Modelagem de Workflow. I. Oliveira, José Palazzo M. de. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof^a. Wrana Maria Panizzi

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

Pró-Reitor Adjunto de Pós-Graduação: Profa. Jocélia Grazia

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

Coordenador do PPGC: Prof. Carlos Alberto Heuser

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

A Deus, Pai de amor e bondade, que ilumina o caminho e sustenta o caminhante.

Àqueles que de uma forma ou de outra apoiaram, auxiliaram e incentivaram este trabalho e seu desafio.

Obrigado ao meu orientador pelo estímulo e paciência.

Em especial a minha esposa Estela e minha filha Mariana que vivenciaram minhas horas e dias de imersão em leituras, a tensão com os prazos e tudo o que estive envolvido neste período. Sem vocês e seu apoio e auxílio não seria realmente possível. Obrigado!

SUMÁRIO

SUMÁRIO.....	4
LISTA DE ABREVIATURAS.....	6
LISTA DE FIGURAS.....	8
LISTA DE TABELAS.....	9
RESUMO.....	10
ABSTRACT.....	11
1 INTRODUÇÃO.....	12
1.1 WORKFLOW.....	12
1.2 PROBLEMAS.....	14
1.2.1 Problemas Conceituais.....	14
1.2.2 Problemas Arquiteturais.....	16
1.2.3 Problemas com a Manipulação dos Dados e Informações.....	18
1.3 AUTOMAÇÃO DE PROCESSOS NA PREFEITURA DE PORTO ALEGRE.....	20
1.4 PROPOSTAS E OBJETIVOS.....	22
2 CONTEXTO TECNOLÓGICO.....	26
2.1 WORKFLOW.....	29
2.1.1 Histórico.....	29
2.1.2 Tipos de Workflow.....	30
2.1.3 Modelagem da Aplicação.....	31
2.1.4 Atores.....	34
2.1.5 Dados e Informações.....	37
2.1.6 Arquitetura.....	40
2.2 SISTEMAS DE INFORMAÇÃO.....	42
2.2.1 Histórico.....	43
2.2.2 Tipos de SI.....	45
2.2.3 Modelagem da Aplicação.....	47
2.2.4 Atores.....	51
2.2.5 Dados e Informações.....	53
2.2.6 Arquitetura.....	55
3 REQUISITOS EM UMA APLICAÇÃO DE WORKFLOW NA PMPA.....	61
3.1 REDESENHO ADMINISTRATIVO NA PREFEITURA DE PORTO ALEGRE.....	63
3.2 REQUISITOS NA AUTOMAÇÃO DO PROCESSO DA DECLARAÇÃO MUNICIPAL.....	66
3.2.1 Redesenho do Processo de Solicitação de Declaração Municipal.....	68
3.2.2 Modelagem do Workflow.....	70
3.2.3 Atores.....	73
3.2.4 Dados e Informações.....	74
3.2.5 Arquitetura.....	75
4 O WIDEFLOW NA AUTOMAÇÃO DE PROCESSOS DA PMPA.....	77
4.1 O MOTOR DE WORKFLOW WIDEFLOW.....	78
4.1.1 Controle do Fluxo.....	78
4.1.2 Assinalamento de Atores.....	83
4.1.3 Manipulação de Dados e Informações.....	84

4.1.4	Arquitetura	87
4.1.5	Definição do Processo	89
4.2	O PROCESSO DA DECLARAÇÃO MUNICIPAL AUTOMATIZADO.....	89
5	CONCLUSÕES	97
	REFERÊNCIAS.....	106

LISTA DE ABREVIATURAS

API	Application Program Interface
ASP	Active Server Pages®
BIRD	Banco Interamericano de Desenvolvimento
BPEL4WS	Business Process Especification Language for Web Services
BPR	Business Process Reengineering
CAM	Controle de Arrecadação Municipal
CDL	Cadastro de Logradouros
CEU	Cadastro de Expediente Único
CNPJ	Cadastro Nacional de Pessoas Jurídicas
COM+	Common Object Model®
CPF	Cadastro de Pessoas Físicas
CRC	Class-Responsability-Colaboration
CREA	Conselho Regional de Engenharia e Agronomia
CSCW	Computer Support Cooperative Work
CTM	Cadastro Técnico Municipal
DAM	Documento de Arrecadação Municipal
DFD	Diagrama de Fluxo de Dados
DLL	Dynamic Load Library
DM	Declaração Municipal
EIS	Executive Information System
ERP	Enterprise Resource Planning
EVU	Estudo de Viabilidade Urbanística
GED	Gestão Eletrônica de Documentos
GPA	Gerenciamento de Processos Administrativos
HIS	Microsoft Host Integration Server®
IHC	Interação Humano-Computador
IMB	Cadastro Imobiliário
IPTU	Imposto Predial e Territorial Urbano
J2EE	Java 2 Enterprise Edition
JAR	Java Archive
JCP	Java Community Process
JSP	Java Server Pages
MRP	Material Requirement Planning
OO	Orientação a Objetos
PDDUA	Plano Diretor de Desenvolvimento Urbano e Ambiental
PDF	Portable Document Format®

PED	Projetos de Edificações
PMPA	Prefeitura Municipal de Porto Alegre
PROCEMPA	Companhia de Processamento de Dados do Município de Porto Alegre
SAD	Sistemas de Apoio à Decisão
SDM	Solicitações de Declaração Municipal
SE	Sistemas Especialistas
SGE	Sistemas de Gestão Empresarial
SIG	Sistemas de Informações Gerenciais
SIT	Sistemas de Informações Transacionais
SMA	Secretaria Municipal da Administração
SMED	Secretaria Municipal de Educação
SMOV	Secretaria Municipal de Obras e Viação
SOAP	Simple Object Access Protocol
TI	Tecnologia da Informação
UML	Unified Modeling Language
UP	Unified Process
UTP	Unidade Territorial de Planejamento
WAPI	Workflow Application Programming Interface
WMC	Workflow Management Coalition
WMS	Workflow Management System
XML	Extensible Markup Language
XPDL	XML Process Definition Language

LISTA DE FIGURAS

FIGURA 1.1: CONJUNTO DE ATIVIDADES EM UM PROCESSO	13
FIGURA 1.2: WORKFLOW NA ARQUITETURA DE APLICAÇÕES	17
FIGURA 1.3: WORKFLOW CRUZANDO SETORES E SEUS SISTEMAS DE INFORMAÇÃO	18
FIGURA 1.4: DIAGRAMA DE ATIVIDADES DA UML.....	24
FIGURA 2.1: <i>THREAD</i> EM DIAGRAMA DE ATIVIDADES.....	33
FIGURA 2.2: INTERAÇÃO ENTRE ATORES E ATIVIDADES.....	34
FIGURA 2.3: ESTRUTURA DE MODELAGEM DE ATORES	35
FIGURA 2.4: INTERFACEAMENTO DE DADOS	38
FIGURA 2.5: INTERFACE UTILIZADA PARA TROCA DE DADOS E DISPARO DE EVENTOS.....	39
FIGURA 2.6: ARQUITETURA GENÉRICA DE UM MOTOR DE WORKFLOW	41
FIGURA 2.7: EVOLUÇÃO HISTÓRICA DOS SISTEMAS DE INFORMAÇÃO	43
FIGURA 2.8: SISTEMAS DE INFORMAÇÃO: SURGIMENTO E APLICAÇÃO.....	46
FIGURA 2.9: MODELO E OBJETO DA MODELAGEM NO MUNDO REAL	49
FIGURA 2.10: PROCESSO DIRIGIDO POR CASOS DE USO	50
FIGURA 2.11: FORMULÁRIO EM PAPEL E FORMULÁRIO ELETRÔNICO.....	52
FIGURA 2.12: CLASSES TÍPICAS DE SISTEMAS DE INFORMAÇÃO	53
FIGURA 2.13: ARQUITETURA DE UMA APLICAÇÃO N-CAMADAS	57
FIGURA 3.1: DECLARAÇÃO MUNICIPAL (DM).....	65
FIGURA 3.2: FLUXO DO PROCESSO DE SOLICITAÇÃO DE DM.....	72
FIGURA 3.3: PLANTA DE SITUAÇÃO.....	75
FIGURA 4.1: META-MODELO DE CLASSES DE DEFINIÇÃO DO WIDEFLOW	78
FIGURA 4.2: META-MODELO DE CLASSES DE EXECUÇÃO DO WIDEFLOW.....	79
FIGURA 4.3: WORKTASK TEMPORAL E COM INTERAÇÃO COM ATOR NO WIDEFLOW	79
FIGURA 4.4: PROPAGAÇÃO DE DOCUMENTOS ENTRE ATIVIDADES	80
FIGURA 4.5: LÓGICA DO PROCESSO E LÓGICA DA EXECUÇÃO	80
FIGURA 4.6: ESQUEMA DE EXECUÇÃO DE ATIVIDADES E CONDIÇÕES DE GUARDA.....	82
FIGURA 4.7: GRUPO DE ATORES	84
FIGURA 4.8: BLOQUEIO CONCORRENTE ENTRE OBJETOS.....	86
FIGURA 4.9: COMPONENTE DE SERVIÇOS DE WORKFLOW	87
FIGURA 4.10: DATA DO ESQUEMA NA DEFINIÇÃO.....	88
FIGURA 4.11: DIAGRAMA DE ATIVIDADES DO PROCESSO DA DM	90
FIGURA 4.12: DIAGRAMA DE ATIVIDADES DO PROCESSO DA DM	94
FIGURA 4.13 JUNÇÃO (JOIN) DE FLUXO	96

LISTA DE TABELAS

TABELA 2.1: CARACTERÍSTICAS DE WORKFLOW E DE SISTEMAS DE INFORMAÇÃO.....	58
TABELA 3.1: DELINEAMENTO DO PROCESSO CRÍTICO DE SOLICITAÇÃO DE DM	68

RESUMO

A demanda por aplicações de workflow tem crescido rapidamente. Por um lado há uma evolução em ambientes com plataformas de sistemas maduras e disseminadas, por outro, processos de redesenho administrativo têm contribuído no aumento desta demanda. Apesar disto, profissionais de TI ainda encontram alguns problemas, e por isto, têm se aproximado de workflow com receio.

Na administração pública de Porto Alegre, a otimização dos processos organizacionais de atendimento às demandas individuais dos cidadãos - que geram um registro em papel que constitui o que se denomina processo administrativo ou expediente, ou seja, um volume físico de papéis tramitando manualmente pelas estruturas que atuam no atendimento da demanda - redundou no emprego de ferramentas de automação para introduzir maior produtividade e eficiência na mudança dos métodos de trabalho. Mas alguns problemas se destacaram quando foi iniciado o desenvolvimento de aplicações de automação de processos.

Uma face deles se mostrou em problemas conceituais que vão desde a confusão entre workflow e outras áreas, como BPR ou GED, à falta de domínio sobre as abstrações fundamentais na elicitação de workflow. Outra face se mostrou nos WMS, verdadeiros ambientes proprietários, fechados e pesados, de difícil utilização e pouco flexíveis para os desenvolvedores utilizarem em seu ambiente. Finalmente, outro problema foi a integração de aplicações de workflow com sistemas de informação, buscando informações e eventos em aplicações legadas, algumas existentes por décadas.

Frente a isto, este trabalho aborda a produção de workflow sob a ótica dos desenvolvedores - analistas, projetistas e programadores - que necessitam criar aplicações corporativas em seus ambientes de Engenharia de Software. Neste cenário, a linha divisória entre a elaboração de uma aplicação de workflow e um sistema de informação se torna tênue, e compreender os aspectos envolvidos, dominá-los e utilizá-los neste ambiente é primordial para o sucesso e disseminação de aplicações de automação de processos.

Este trabalho propõe uma extensão do Diagrama de Atividades da UML para modelar e elicitar o controle de atividades, a criação de um motor de workflow para executar a máquina de estados do Diagrama de Atividades, a ser utilizado como componente na arquitetura das aplicações. A utilização do motor de workflow e do Diagrama de Atividades estendido são apresentados num case de automação de um processo da Secretaria do Planejamento Municipal da Prefeitura Municipal de Porto Alegre que atende a solicitações diretas dos cidadãos.

Palavras-chave: workflow, automação de processos, modelagem de workflow.

Development of Workflow for Public Administration in Corporative Environments of Engineering of Software

ABSTRACT

The demand for workflow applications has been growing quickly. On one hand, there has been an evolution in environments with mature and disseminated platforms of systems, on the other, processes of Business Process Reengineering have contributed to the increase of this demand. Despite this, the professionals of IT have approached the workflow with distrust.

In the public administration of Porto Alegre, the optimization of the organizational processes of assistance to the individual demands of the citizens - which generate registration on paper that constitutes what is called administrative proceeding, in other words, a physical volume of paper moving manually among the structures that act in the assistance of the demand - resulted in the job of automation tools to introduce larger productivity and efficiency in the change of work methods. But some problems stood out when the applications development for automation of processes started.

One of them was shown in conceptual problems that go from the confusion between workflow and other areas, like BPR or GED, to the domain lack on the fundamental abstractions in the workflow elicitation. Another one was shown in WMS, true environments proprietors, closed and heavy, difficult to use and not very flexible for the developers. Finally, another problem was the integration of workflow applications with information systems of information, where information and events are searched and observed, some of these systems have existed for decades.

In the face of this, this work approaches the workflow production under the optics of the developers - analysts, designers, architects and programmers - that need to create corporative applications in their environments of Software Engineering. In this scenery, the dividing line between the elaboration of a workflow application and a system of information becomes tenuous, and to understand the involved aspects, to dominate them and to use them in this environment it is primordial for the success and dissemination of applications of automation of processes.

This work proposes an extension of the UML Activity Diagram to model and elicit the control of activities, the creation of a workflow engine to execute the machine of states of the Activity Diagram to be used as component in the architecture of the applications. The use of the workflow engine and of the extended Diagram of Activities are presented in an automation case of a process of the Planning Bureau of the City Hall of Porto Alegre which assists to the citizens' direct requests.

Keywords: workflow, automation of processes, modeling of workflow.

1 INTRODUÇÃO

1.1 Workflow

O conceito de *workflow* é um conceito complexo e difícil de definir, estreitamente relacionado com o conceito de processo de negócio. (GONÇALVES, 2000) afirma no título de seu trabalho que “*as empresas são grandes coleções de processos*” pois “*todo trabalho importante feito numa empresa faz parte de algum processo*” Ibid, apud (GRAHAM; LEBARON, 1994), não existindo nelas serviço ou produto que não seja resultado de um processo ou um processo que não produza produto ou serviço.

Uma definição de processo de negócio genericamente aceita é a de um conjunto de atividades com um objetivo comum, que executadas em certa ordem, contribuem para o negócio como um todo, entregando um serviço ou produto a um cliente externo ou interno, como um departamento ou seção [...] (WORKFLOW MANAGEMENT COALITION, 1995). O processo em questão ocorre dentro de uma estrutura organizacional, definindo os papéis funcionais e os relacionamentos com os outros processos. A figura 1.1 mostra um conjunto de atividades em um processo.

Processos de avaliação de pedidos de empréstimo, matrículas de alunos, requisições de aprovação de plantas para construções por órgãos fiscalizadores municipais, são exemplos de processos nas organizações, os quais a tecnologia de *workflow* tem automatizado.

Diversos autores apresentaram definições para *workflow*, (NICOLAO, 1996) discute e expõe algumas das mais conhecidas, como as apresentadas por (GEORGAKOPOULOS, 1995) e (JOOSTEN, 1995). Além destas, ainda poderiam ser citadas as definições dadas em (CASATI et al., 1995) e (MANOLESCU, 2001).

Um conceito largamente aceito, e que será utilizado neste trabalho, é o publicado pelo Workflow Management Coalition (WMC), que dá a seguinte definição para *workflow*:

A automação de um processo de negócio, em todo ou em parte, na qual documentos, informações ou tarefas são passados de um participante à outro para execução, de acordo com um conjunto de regras preestabelecidas., [...], (WORKFLOW MANAGEMENT COALITION, 1995)

Workflow é a automação de um processo de negócio no contexto da tecnologia de informação. O processo pode estar automatizado no todo ou em parte, o que significa que podem existir atividades, informações e documentos tratados manualmente durante o processo, com participantes e regras situadas fora do escopo da automação.

Anterior à automação, é a etapa onde o processo de negócio é analisado, compreendido e definido conceitualmente. Nesta etapa são destacados os documentos, as informações e tarefas que fazem parte do processo. Também são identificados os participantes, a seqüência das atividades e sob quais regras o processo evolui. Em suma, a interação destes conceitos com as regras define a semântica do domínio.

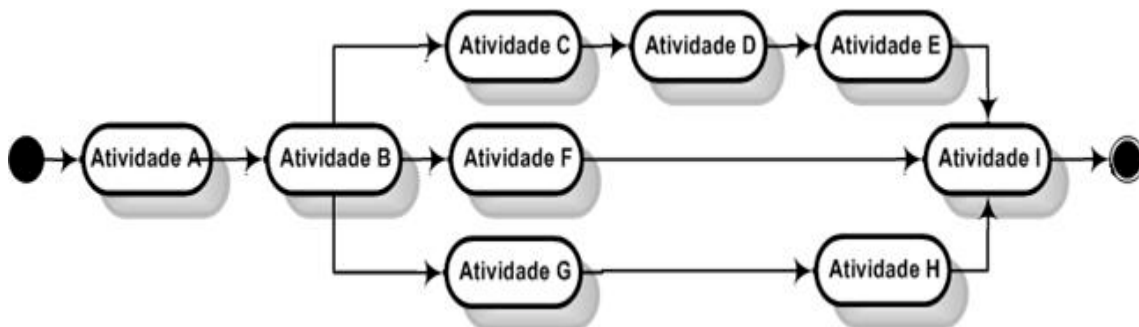


Figura 1.1: Conjunto de atividades em um processo

Em seguida, o processo deverá ser automatizado. O suporte para implementação de um sistema de workflow é chamado de *Workflow Management System* (sistema de gerência de workflow), ou WMS, que é “um sistema que define, administra e executa ‘workflows’ através da execução do software que dirige o workflow, de acordo com a representação computacional da lógica deste workflow” [...] (WORKFLOW MANAGEMENT COALITION, 1995), isto é, o software que permite a automação dos procedimentos de um processo de negócio, através do tratamento da seqüência de atividades e da chamada ao participante humano apropriado e/ou recurso computacional associado com as várias atividades a serem executadas.

Este é o componente de software com a responsabilidade de criar uma ocorrência de um processo previamente definido, quando necessário. Assim, se um processo de compra de material tiver sido analisado, definido e implementado num WMS, a cada solicitação de compra de material o sistema de gerência de workflow inicia uma ocorrência, uma instância, que será passada entre os participantes e acompanhada até o fim previamente definido.

Além de propiciar ganhos como diminuição dos tempos de um processo, melhoria da qualidade de execução, diminuição de erros e maior gerenciabilidade dos processos, workflow pode se integrar com outras aplicações, inclusive com a Internet, fazendo com que surja uma nova classe de aplicações que introduz vantagem competitiva real na empresa frente aos desafios de satisfação de seus mercados.

1.2 Problemas

Apesar de tudo isto, os profissionais de TI que necessitam implementar aplicações de workflow em ambientes de engenharia de software ainda possuem falsas noções e fazem muita confusão sobre o tema, e por isto, têm se aproximado de workflow com receio.

Na área de TI da administração pública de Porto Alegre, a tentativa de alcançar as metas para a otimização de processos organizacionais, empregando ferramentas de automação para introduzir maior produtividade e eficiência na mudança dos métodos de trabalho, otimizando os processos de trabalho e qualificando o sistema de gerenciamento, alguns destes problemas se destacaram no processo de desenvolvimento de aplicações, quando os profissionais de Engenharia de Software do município começaram a desenvolver os workflows.

Uma face do problema se materializa na grande parcela de usuários, analistas e outros profissionais, que não conseguem delimitar claramente a linha divisória conceitual entre redesenho de processos (BPR – Business Process Reengineering) e workflow, dificultando a análise de requisitos e modelagem de abstrações para realizar a automação.

Outra face se mostra nas diversas aplicações que a tecnologia de workflow possui, desde roteamento de documentos entre pessoas numa ou em mais organizações, até o controle de tarefas e recursos na produção industrial, fazendo com que fornecedores de WMS entreguem verdadeiros ambientes proprietários, fechados e pesados, tornando sua utilização difícil e pouco flexível.

Finalmente, outro problema é encontrado quando é necessário integrar aplicações de workflow com sistemas de informação, por meio de componentes de código que forneçam serviços específicos ou mesmo com acesso direto aos dados legados, em bancos de dados ou formatos diversos.

Estes são alguns dos problemas que têm impedido a rápida adoção de workflow e sua utilização em larga escala em ambientes de desenvolvimento de sistemas de informação como o existente na administração pública de Porto Alegre.

1.2.1 Problemas Conceituais

Os problemas conceituais iniciam quando a maioria dos analistas de sistemas ainda crê que os conceitos e abstrações necessários às aplicações de workflow são diametralmente opostos ou não comunicantes com os conceitos existentes em sistemas de informação, e que as técnicas necessárias para modelar a automação de um processo de negócio não possam sofrer influência de técnicas clássicas de modelagem utilizadas por décadas para a construção de sistemas.

A confusão começa quando se acredita que reengenharia ou redesenho de processos (GONÇALVES, 2000) e workflow sejam a mesma coisa. É verdade que muitos fornecedores de ferramentas de automação de workflow incluem narrativas de como seu produto modificou o relacionamento com os clientes, fornecedores e outros colaboradores. As vantagens alardeadas na melhoria dos negócios são fruto da reengenharia, mas workflow é só uma tecnologia de software que provê meios para automatizar um processo, agregando as vantagens próprias das soluções de TI (ULTIMUS, 2001).

Outra confusão é com o *workflow enabled* que muitos produtos, direcionados para atender necessidades diversas da automação de processos, dizem conter (ULTIMUS, 2001). São produtos com os quais os usuários de aplicações interagem, como planilhas, processadores de texto, etc, que buscam integração com sistemas de gerenciamento de workflow, geralmente através da implementação de interfaces inspiradas na WAPI 3 [...] (WORKFLOW MANAGEMENT COALITION, 1995), Workflow API Invoked Applications, do WMC. Assim, aplicações *workflow enabled* são aquelas que possuem em sua implementação, chamadas a WAPI, o que as possibilitaria interagir diretamente com os sistemas de workflow. Este tipo de aplicação não pretende nem possui as características necessárias para suportar processos automatizados.

Da mesma forma, a possibilidade de implementar workflows de documentos em GED, Gerenciamento Eletrônico de Documentos, por exemplo, tem contribuído para aumentar esta confusão. Para suportar automação é necessária uma série de funcionalidades [...] (WORKFLOW MANAGEMENT COALITION, 1995), mais que o simples gerenciamento de documentos.

Além disto, as abstrações a serem analisadas são essencialmente diferentes dos conceitos presentes na modelagem de sistemas de informação, mas muitas técnicas clássicas, típicas de modelagem de sistemas, podem ser utilizadas na modelagem de workflow. Os conceitos devem ser feitos “sob medida” para atender um ambiente de produção de sistemas de workflow e devem apontar os componentes físicos implementados, isto é, a arquitetura do motor de workflow deve estar em conformidade com a modelagem.

Na modelagem, o analista busca definir as regras de relacionamento entre tarefas e outras tarefas, entre tarefas e os dados necessários em cada uma delas e entre tarefas e os participantes que as executam, algo como a “integridade referencial” do workflow, de forma análoga ao que faz ao projetar um banco de dados relacional. O motor de workflow, o componente responsável por executar o workflow como numa máquina de estados, se responsabiliza por garantir que as regras e definições impostas pela modelagem sejam respeitadas e executadas.

AALST, HOFSTEDE e BARROS (2000b) afirma que as diferenças semânticas entre as diversas linguagens redundam em diferenças no grau de adequação e de expressividade para a modelagem de workflow e que

dominar os requisitos encontrados no processo de desenvolvimento de aplicações de workflow é tão ou mais importante que definir uma semântica que unifique as propostas de modelagem.

Na realidade, o conceito de workflow deve fluir naturalmente entre os membros da comunidade de desenvolvedores de sistemas de informação. A interação entre workflow e aplicações clássicas de sistemas deve evoluir até que a separação de conceitos seja o traçado de uma linha teórica que a prática ultrapasse diariamente, unindo os conceitos em soluções funcionais e bem integradas, que suportem automação de processos e integração com sistemas legados.

1.2.2 Problemas Arquiteturais

Os fornecedores de ferramentas de automação de processos praticamente têm determinado quais necessidades devem ser atendidas e como, entregando pacotes de ferramentas (MANOLESCU, 2001), sem o preparo necessário para atender às demandas e requisitos dos usuários ávidos por construir suas aplicações. Questões como performance, escalabilidade, integridade e principalmente, flexibilidade para uso em conjunto com aplicações corporativas, típicas preocupações arquiteturais para a construção de sistemas, têm recebido pouca atenção e comprometido o uso de workflow em larga escala (ALONSO et al., 1997).

As incompatibilidades ultrapassam as questões de sintaxe ou plataforma. A maioria das ferramentas são baseadas em paradigmas essencialmente diferentes, como Petri-Net ou State-Chart. Estas diferenças conceituais dificultam o mapeamento entre modelo e arquitetura, entre abstrações no modelo e a máquina de estados. Neste sentido, um dos maiores problemas que analistas e programadores têm com relação ao processo de modelagem, desenvolvimento e implementação de aplicações computadorizadas, diz respeito ao progressivo mapeamento entre os modelos e o produto de software construído. Um processo de desenvolvimento de software é fortemente influenciado pelas técnicas que adota e principalmente pela arquitetura para a qual todo o trabalho é direcionado (GOMIDE, 2002).

Por outro lado a maioria das ferramentas para automação possui capacidades sofisticadas para serem usadas como verdadeiras plataformas onde desenvolver soluções de workflow. Isto tem requerido um significativo investimento em treinamento no ambiente específico da ferramenta e no próprio esforço de criação das aplicações, que são construídas do “zero”, sem reuso de código e de soluções (MANOLESCU, 2001). São soluções do tipo “tudo ou nada”, onde, ou a ferramenta atende e abrange todo o processo e a solução, ou ela não pode ser utilizada, pois não possui a flexibilidade necessária.

Apesar das muitas ferramentas de automação de processos disponíveis no mercado, os desenvolvedores de sistemas descobrem o quão difícil é utilizar workflow em conjunto ou como parte das aplicações

que eles desenvolvem. (MANOLESCU, 2001) relata casos onde arquitetos e projetistas desejavam utilizar workflow como coordenador do fluxo de procedimentos, da lógica do processo, isolando-a do código específico do domínio de seus sistemas de informação. Eles tiveram que desenvolver soluções próprias para atender suas necessidades, acoplando-as fortemente a sua aplicação. Isto ocorre porque as implementações de workflow não são flexíveis e a forma como são disponibilizadas para sua integração com outros softwares não têm atendido às necessidades de analistas e projetistas que necessitam utilizar workflow como parte integrante da arquitetura de suas soluções de software.

Além disto, elas são soluções proprietárias e a migração de um sistema para outra plataforma ou ferramenta de outro fornecedor traria, no mínimo, enormes dificuldades e em muitos casos, seria impossível sem a re-escrita das aplicações.

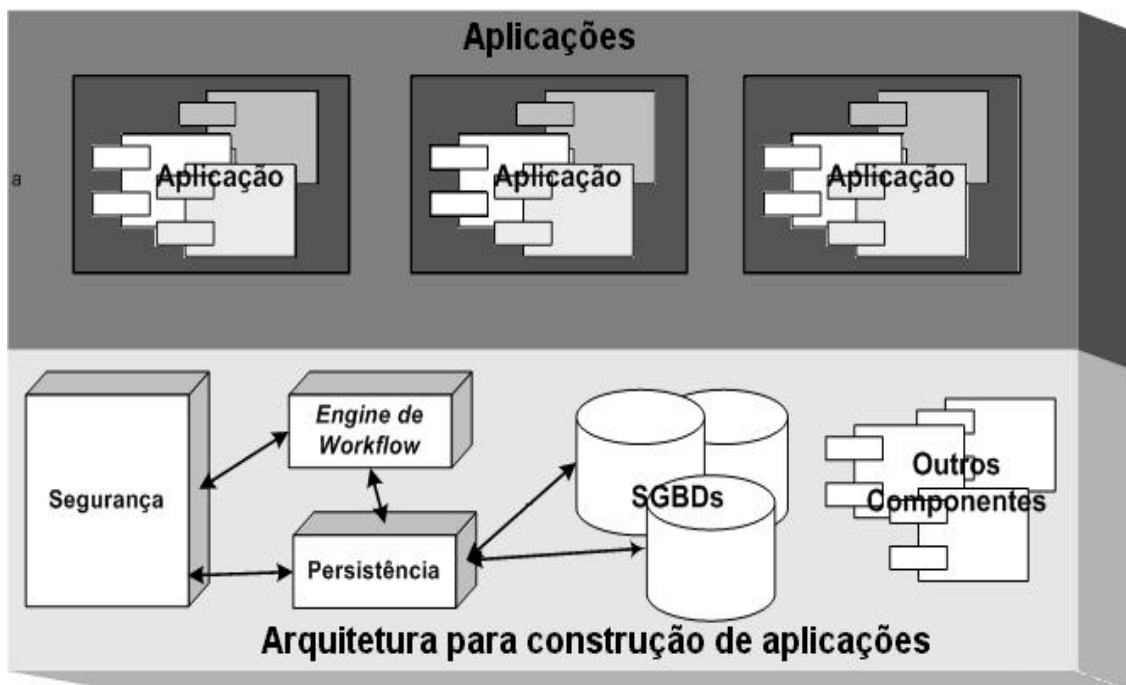


Figura 1.2: Workflow na arquitetura de aplicações

Neste cenário, workflow deveria ser parte da arquitetura necessária a um problema que envolva, por exemplo, um domínio somente com automação de processos ou um domínio híbrido, com requisitos de automação de processos e forte necessidade de integração com sistemas legados, como na figura 1.2. (MANOLESCU, 2001) afirma que a tecnologia de workflow tem se expandido e seu uso está migrando de ferramenta voltada ao usuário final para arquitetura - *middleware* - para suportar aplicações de automação de processos.

Percebe-se que, nos extremos da tecnologia, workflow pode ser o único fundamento para a aplicação resultante ou somente uma pequena parte da solução. De uma ou de outra forma, um analista ou projetista de sistemas busca workflow como base arquitetural para sua aplicação. Assim, ele necessita utilizar o motor de workflow de forma semelhante como tem

utilizado um sistema de gerenciamento de banco de dados: modelando a seqüência de tarefas, os dados necessários em cada uma delas de acordo com a lógica desejada e as possíveis integrações com bases legadas através de interfaces de dados.

Seja como for, encarar a tecnologia de workflow como parte da arquitetura de suporte da mesma forma que outros elementos arquiteturais utilizados em sistemas de informação, traz ao problema uma tonalidade mais apropriada para que os profissionais da comunidade de sistemas de informação utilizem workflow no dia a dia como parte de sua caixa de ferramentas. Isto significa que tanto as limitações quanto as possibilidades são conhecidas, mas as alternativas podem ser implementadas em padrões de análise, típicas soluções de análise de projeto.

1.2.3 Problemas com a Manipulação dos Dados e Informações

Apesar dos esforços dos organismos internacionais de padronização, as ferramentas para automação de processos têm sido implementadas de forma monolítica, não dirigidas aos desenvolvedores de sistemas, dificultando a interação com as diversas aplicações legadas existentes no ambiente de TI das organizações (JOOSTEN; BRINKKEMPER, 2002).

Existem cenários diversos na utilização da tecnologia de automação. Quando pensamos em workflow em processos de produção (NICOLAO, 1996), imaginamos o controle do fluxo de tarefas e da disponibilidade das informações necessárias a sua execução como reação à interação de cada participante, onde o mecanismo de execução se responsabiliza pelo roteamento das tarefas no fluxo, sua atribuição aos executores e à guarda dos dados do processo.



Figura 1.3: Workflow cruzando setores e seus sistemas de informação

Neste tipo de processo a aplicação resultante manipula dados do processo, informados pelos participantes, que preenchem formulários eletrônicos com os dados pertinentes ao caso em execução, como valores monetários, nomes de pessoas e seus dados de identificação pessoal ou de identificação funcional, endereços, informações sobre bens móveis ou imóveis, enfim, uma infinidade de dados diariamente utilizados em processos de produção na esfera da administração das organizações.

Os dados manipulados por estas aplicações geralmente são armazenados na forma original como informados pelo ator, não sofrendo transformações significativas, com validação sintática e semântica pouco complexas. Estas aplicações se assemelham aos tradicionais sistemas de informação, e mesmo muitos sistemas de informação têm desempenhado o papel de aplicações de workflow, controlando o estado do caso, de cada instância de processo, persistindo informações de controle em bancos de dados.

Muitos dados manipulados nas instâncias de workflow são típicos dados encontrados em bases legadas administradas por sistemas de informação. É comum um processo atravessar as fronteiras entre domínios de diversas aplicações, como mostra a figura 1.3. Assim, este tipo de workflow necessita estar fortemente integrado com dados de bases legadas, encontrados em muitas plataformas e sob muitos formatos, introduzindo uma complexidade mais que conceitual, uma dificuldade arquitetural neste esforço.

A estreita relação dos dados manipulados por aplicações baseadas em workflow como a citada acima e aplicações de sistemas de informação faz com que a integração entre estes dois tipos de aplicação seja necessária na quase totalidade das experiências de automação. A base de dados legados existente em organizações com plataformas de sistemas de informação maduras, implantadas ainda em décadas passadas não pode ser desprezada e os processos automatizados necessitam fazer uso delas. É inadmissível supor que os atores dos processos devam ter o re-trabalho de digitar informações existentes em sistemas de informação ou, no caminho inverso, digitar nos sistemas de informação os dados oriundos das instâncias de processo, provocando perda de agilidade, de qualidade e de produtividade, exatamente algumas das características que a automação de processos busca alcançar.

Outro cenário, citado por (MANOLESCU, 2001), é a utilização de workflow como forma de abstrair a lógica do fluxo de processos, encontrada nos sistemas de informação, do código que implementa as regras de tratamento das informações do domínio. Assim, alterações no fluxo dos procedimentos são feitas alterando-se o fluxo do workflow sem alterar o código da aplicação nem recompilá-lo e redistribuí-lo. Um sistema de informação trata dos serviços típicos do domínio, as transações do domínio, responsabilidades discretas que podem ser implementadas sem conexão com a lógica do fluxo da seqüência destas transações. Aqui o workflow se responsabiliza por fazer trafegar os dados entre os diversos componentes na seqüência necessária de acordo com a arquitetura do sistema de informação modelada pelo analista. O reuso de componentes e de serviços de sistemas de informação pode ser potencializado se suas arquiteturas se basearem em soluções deste tipo.

Mas o que mais comumente ocorre é a automação de processos que passam horizontalmente pelo domínio dos sistemas legados, onde além do roteamento de tarefas e informações armazenadas entre os participantes, encontramos intensa troca de dados com bases de sistemas legados,

quando dados legados são passados para o processo e disponibilizados nas tarefas aos participantes ou quando os dados são informados pelo participante e, além de armazenados como dados do processo, são passados para que aplicações legadas atualizem suas bases. Aqui o mecanismo de execução de processos também é responsável por ativar funções do sistema de informação e entregar-lhes as informações de que necessitam para, aplicando os filtros das regras específicas do domínio, atualizarem os dados.

1.3 Automação de Processos na Prefeitura de Porto Alegre

O Município de Porto Alegre constitui-se no principal pólo econômico do estado do Rio Grande do Sul e, como característica dos centros urbanos, concentra considerável parcela da população e das atividades econômicas, assumindo cada vez mais o perfil de centro de serviços [...] (PREFEITURA MUNICIPAL DE PORTO ALEGRE, 2003).

A partir da Constituição de 1988, vem ocorrendo um processo de transferência para os municípios de funções antes exercidas pela União e pelas unidades federativas. Com isto, as prefeituras passaram a assumir uma carga maior de atividades, tornando-se imperativa uma verificação apurada nos procedimentos desenvolvidos pela estrutura da administração pública sob o enfoque da racionalização do emprego de sua capacidade operacional no atendimento das demandas da comunidade[...] (PREFEITURA MUNICIPAL DE PORTO ALEGRE, 2003).

Adicionalmente, a aproximação entre a administração pública e a comunidade se dá em dois sentidos. Um é o movimento da estrutura da administração pública em direção à comunidade e como reação às ações desenvolvidas pela administração pública, o munícipe faz o movimento em direção à estrutura da Administração Pública Municipal, demandando serviços e soluções para suas reivindicações.

Buscando alternativas, o Município de Porto Alegre firmou convênio com o Banco Interamericano de Desenvolvimento, BIRD, para a promoção do Programa de Desenvolvimento Municipal. Este programa, além de contemplar melhorias na infra-estrutura urbana da cidade, objetiva fortalecer a administração pública municipal através da implementação de projetos que otimizem os processos de trabalho e qualifiquem o sistema de gerenciamento. Assim, foram executados serviços de identificação, priorização e redesenho de processos organizacionais [...] (PREFEITURA MUNICIPAL DE PORTO ALEGRE, 2003).

PIMENTA (2003) afirma que *“com relação ao provimento de serviços públicos, as tecnologias de informação têm importante contribuição. Desde os processos de gestão até os serviços mais próximos ao cidadão, a aplicação de tecnologias da informação pode proporcionar maior efetividade e transparência nas ações governamentais”*.

Para que todas aquelas metas pudessem ser alcançadas, muitos dos processos redesenhados devem ser automatizados com suporte de tecnologia de workflow. O objetivo é alcançar agilidade e presteza no atendimento ao cidadão ou mesmo nos tramites internos, qualificação dos serviços dependentes destes processos, racionalização na utilização dos recursos existentes e ampliação da quantidade de serviços entregues à população, nos processos automatizados ou mesmo em outros serviços, pela racionalização dos recursos causada pela automação destes processos, além da eliminação do grande volume de papel gerado. São processos típicos de produção (NICOLAO, 1996) que se caracterizam por possuírem passos repetitivos e conhecidos *à priori*, modelagem simples, tratarem de fluxo de trabalho entre pessoas e possuírem forte integração com legado (MOLZ; THOM; SCHEIDT, 1998), com forte utilização de dados de sistemas de informação em diversas plataformas.

Em aplicações desta classe, utilizar uma ferramenta pesada e monolítica (MANOLESCU, 2001) iria fazer com que a construção da solução fosse uma tarefa trabalhosa face ao problema real. A forte necessidade de integração com sistemas legados era fator crítico para a automação de processos de produção.

Durante os últimos 25 anos, a tecnologia da informação tem sido utilizada para melhoria direta do atendimento ao cidadão de Porto Alegre. Os sistemas informatizados da Administração Pública dão suporte às áreas de planejamento, materiais, serviços, orçamento, contabilidade e controle de contratos das secretarias municipais. Na maioria das vezes, esses sistemas não são visualizados ou usados diretamente pelos cidadãos, no entanto, seu uso resulta, significativamente, na melhoria do atendimento e nos serviços prestados ao público.

A PROCEMPA (Companhia de Processamento de Dados do Município de Porto Alegre) é a responsável pela execução da política de informática para a Prefeitura Municipal da cidade. Fundada em 1977, conta com um quadro de profissionais que conhecem profundamente o negócio do cliente e por profissionais com habilidades nas diversas áreas de tecnologia de informação e de comunicações.

Ao longo dos anos, a PROCEMPA elaborou e integrou uma grande infra-estrutura de sistemas de informação consolidada por mais de uma centena de aplicações, muitas delas corporativas, responsáveis pelo apoio aos serviços administrativos ou operacionais prestados pela administração do município. São ativos intangíveis representados em milhões de linhas de código e, talvez o principal no tocante à automação de processos, milhões de informações armazenadas e disponíveis para serem integradas aos processos administrativos a serem automatizados.

Além disto, a PROCEMPA conta com um processo de Engenharia de Software definido (ARAUJO; GOMIDE; PETRILLO, 2003) e é desejável que aplicações de workflow se integrem sem costuras ou degraus neste processo, isto é, que aplicações para automação de processos utilizem workflow como parte da arquitetura para a criação de soluções coesas e

integradas com dados legados armazenados e manipulados no domínio de sistemas de informação.

O ambiente de Engenharia de Software existente deve se apropriar das técnicas de elicitação, análise, projeto e implementação de processos automatizados, onde a separação entre sistemas de informação e workflow seja somente uma prática didática, pois na prática, os analistas de sistemas são os profissionais que tratam, por excelência, da elaboração de aplicações, e devem ter uma arquitetura que possibilite a criação de soluções integradas de forma transparente.

Um relacionamento forte, correto e sem emendas é necessário entre atividades de desenvolvimento dos processos e as atividades de desenvolvimento dos sistemas de informação, (JACOBSON; ERICSON; JACOBSON, 1994)

Frente aos problemas apresentados na automação de processos com tecnologia de workflow, um dos escopos deste trabalho contempla a criação de uma ferramenta, um motor de workflow capaz de executar processos de produção que automatizem o que denomina-se processo administrativo, ou seja, um volume de papéis nos quais se registram todos os pareceres, tramitações, providências, anexação de documentos e decisões de todas as partes da estrutura que atuam operacionalmente no atendimento ou necessitam tomar conhecimento do seu teor [...] (PREFEITURA MUNICIPAL DE PORTO ALEGRE, 2003).

Esta necessidade originou a criação do WideFlow, um motor de workflow que executa as ações baseado na máquina de estados implícita ao Diagrama de Atividades da UML [...] (OBJECT MANAGEMENT GROUP, 2003), com um meta-modelo que se beneficia de técnicas orientadas a objetos (LARMAN, 2000), (ARAUJO; GOMIDE; PETRILLO, 2003) e implementado em serviços de um componente distribuído elaborado com tecnologia COM+ (*Microsoft Common Object Model*) (KIRTLAND, 1999), (ARAUJO; GOMIDE; PETRILLO, 2003).

Esta solução pretende suportar a automação dos processos descritos acima, com flexibilidade e facilidade para utilizar ou alimentar bases de dados legados e permitir utilizar técnicas de modelagem da aplicação no processo de análise e projeto da solução, já largamente difundidas na comunidade de sistemas de informação e mais recentemente, na PROCENPA (ARAUJO; GOMIDE; PETRILLO, 2003).

1.4 Propostas e Objetivos

O uso de máquinas de estado para especificar o comportamento dos objetos com uma linguagem do tipo evento-estado-ação, seleções e iterações, fazem com que os modelos obtenham alto poder conceitual (BARROS, HOFSTEDE, 1996), (BARROS, HOFSTEDE, 1997). Todo o

processo de modelagem se apoia em uma arquitetura definida que suportará a solução implementada (JACOBSON; ERICSON; JACOBSON, 1994), assim, os modelos construídos para uma aplicação devem ser suportados por um motor de workflow que se comporte exatamente como as abstrações no modelo.

Para alcançar este objetivo contornando os problemas citados em 1.2 e no cenário descrito em 1.3, este trabalho pretende abordar:

- a) a automação de workflow através de um motor de workflow chamado WideFlow, constituído de componentes, numa arquitetura de workflow própria, ajustada e customizada às necessidades e possibilidades dos desenvolvedores para atender a um universo delimitado no cenário descrito anteriormente: workflow de processos repetitivos, como protocolos e requisições, tipicamente caracterizados como processos de produção construído e integrado na arquitetura e processo de desenvolvimento de aplicações descrito em (ARAUJO; GOMIDE; PETRILLO, 2003).
- b) estender o diagrama de atividades para modelagem de processos ((BASTOS; RUIZ, 2001), (BASTOS; RUIZ, 2002)) e para definir uma semântica capaz de fazer com que suportem workflow (ESHUI; WIERINGA, 2001b), aliado a propostas de interação com atores (CASATI et al., 1995) e padrões de projeto utilizados em orientação a objetos (GAMMA et al., 2000) para atender aos processos de produção na administração da Prefeitura Municipal de Porto Alegre.
- c) modelar o diagrama de atividades do processo por meio do Rational Rose Enterprise[®], utilizando os estereótipos discutidos para apreender a semântica desejada em um processo.
- d) apresentar a utilização do WideFlow e do processo de desenvolvimento de aplicações de workflow na automação de um processo da Secretaria do Planejamento Municipal da Prefeitura Municipal de Porto Alegre que atende a solicitações diretas dos cidadãos num típico serviço de e-government.

Um diagrama de atividades é um grafo, variante dos fluxogramas, composto basicamente por nós e vértices. Estes diagramas são utilizados para modelar uma função ou operação ou mesmo para descrever processos de negócio (fluxos de trabalho) (ARAUJO; GOMIDE; PETRILLO, 2003).

Um workflow representado por um diagrama de atividades como o da figura 1.4, inicia por um sinal redondo preenchido e termina por um círculo com um sinal redondo preenchido em seu centro. As atividades são representadas por elipses e uma barra representa um *fork*, mais que uma atividade sendo iniciada paralelamente, ou um *join*, sincronizando o término de mais que uma atividade. Um losango representa uma bifurcação na seqüência ou a fusão de seqüência anteriormente bifurcadas. As atividades são ligadas por setas, representando a transição entre elas. Uma transição pode possuir uma condição de guarda que significa que para que a transição ocorra é necessário satisfazer a condição de guarda, caso contrário a transição não ocorre [...] (OBJECT MANAGEMENT GROUP, 2003).

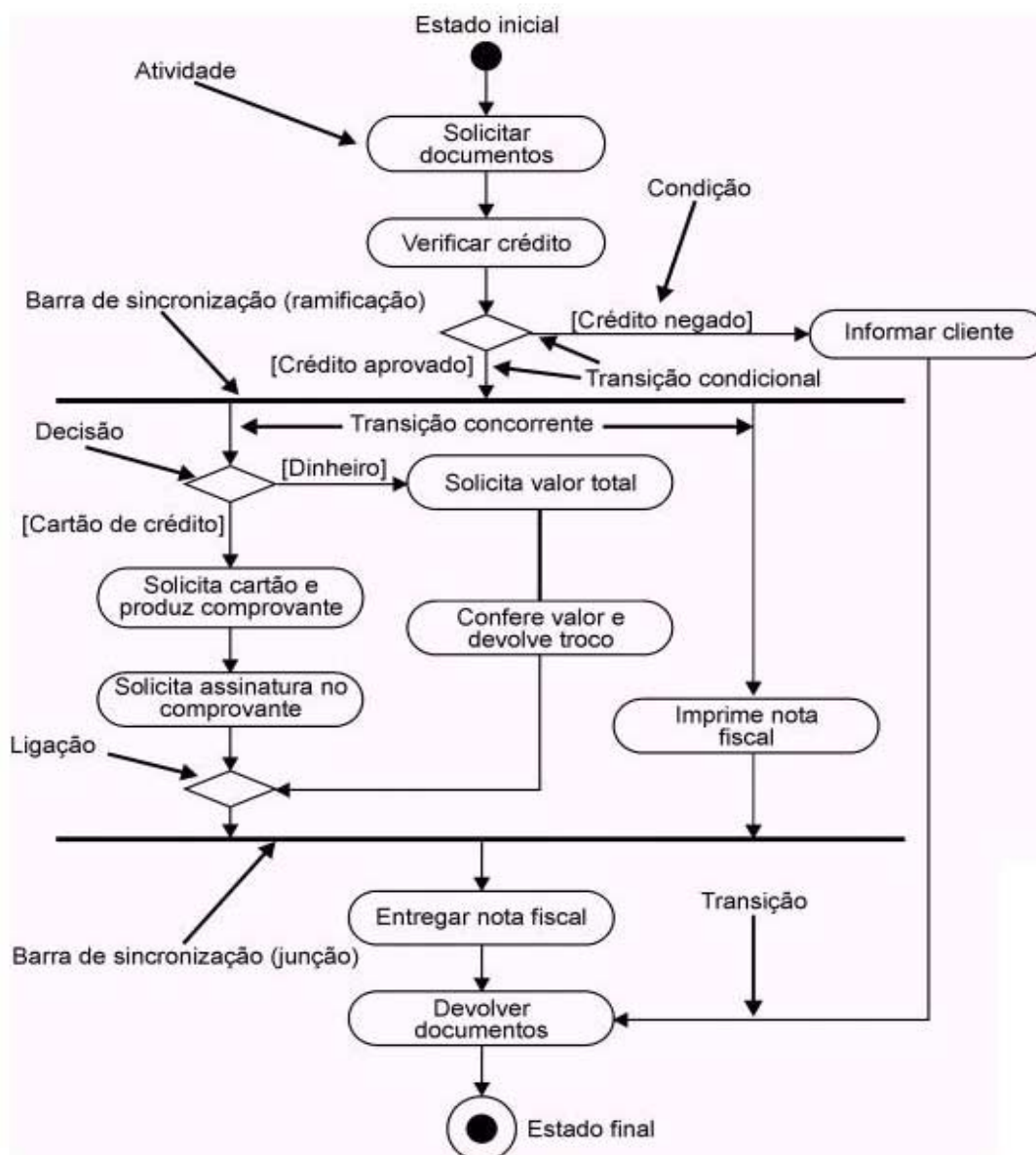


Figura 1.4: Diagrama de atividades da UML (ARAUJO; GOMIDE; PETRILLO, 2003)

Formalmente, um diagrama de atividades da UML descreve uma máquina de estados. Uma máquina de estado possui um comportamento que especifica a seqüência dos estados que um objeto toma durante sua vida em resposta aos eventos que disparam esta troca.

O diagrama de atividades têm sido utilizado com diversos propósitos mas o mais relevante na abordagem dada por este trabalho é o seu uso para a modelagem de processos de negócio. (ARAUJO; GOMIDE; PETRILLO, 2003) mostra o diagrama de atividades sendo utilizado no início do processo de compreensão dos conceitos e abstrações do domínio do problema, sendo a principal ferramenta modelagem e análise de processos para descrever o fluxo das atividades executadas no espaço do domínio.

Ele é um diagrama de fácil compreensão tanto por parte de usuários e clientes quanto pelos analistas, sendo talvez esta a principal razão para

sua difusão. Além disto, o diagrama de atividades da UML é um padrão de fato, fortemente acoplado aos processos de desenvolvimento orientados a objeto (ARAUJO; GOMIDE; PETRILLO, 2003), (LARMAN, 2000). Assim, suas funcionalidades são dominadas por analistas e projetistas e um WMS baseado nelas pode ser a base conceitual e a base arquitetural sobre a qual construir aplicações de workflow.

O capítulo 2 deste trabalho discutirá o contexto tecnológico das áreas de sistemas de informação e de workflow, o capítulo 3 apresentará o cenário de desenvolvimento de aplicações atual na PROCempa e os requisitos do processo a ser automatizado, o capítulo 4 apresentará a estrutura e funcionamento do WideFlow e a experiência com um case de automação, a seção 5 fecha este trabalho, discutindo as soluções alcançadas.

2 CONTEXTO TECNOLÓGICO

Workflow enfeixa conceitos de diversas outras áreas de conhecimento, aproveitando princípios de gerenciamento de documentos e imagens; sistemas gerenciadores de banco de dados; correio eletrônico; modelagem de empresas; BPR; práticas de negócio e engenharia (MATSUDA, 2000).

Workflow é hoje uma classe de software particular dentro do contexto de tecnologia de informação, utilizado para apoiar processos e fluxos de trabalho e também como novo paradigma para a criação de sistemas de informação.

MOLZ, THOM e SCHEIDT (1998) citam benefícios introduzidos por workflow do ponto de vista de Tecnologia de Informação como:

- a) Roteamento do trabalho – definição da seqüência, permitindo que o participante do workflow receba o item de trabalho automaticamente;
- b) chamada automática de aplicativos – a chamada de aplicativos com os quais o workflow deve interagir é transparente ao participante ou ainda, através do item de trabalho, o mecanismo de workflow faz esta chamada;
- c) distribuição dinâmica de trabalho – a escolha do participante pode ser feita dinamicamente, baseada em regras de balanceamento de carga;
- d) garantia de integridade do processo – o uso de um WMS garante a observância das regras e ordenação do processo;
- e) manipulação eletrônica de documentos – introduz vantagens como consistência, facilidade de armazenamento e recuperação, etc;
- f) eliminação do tempo de espera – com o uso de um WMS a coordenação entre as tarefas é administrada automaticamente.

A crescente integração entre workflow e sistemas de informação já existentes nas empresas, é outro fato que faz com que workflow possa ser implementado como fator diferenciador para alcançar os objetivos do negócio. Além disto, a facilidade com que workflow se integra com a Internet, faz com que surja uma nova classe de aplicações que representa, para muitos negócios, uma vantagem competitiva real: disponibilidade de

atendimento contínuo e ininterrupto, podendo instanciar processos a qualquer hora, na região de origem ou a partir de qualquer outra região geográfica, sem necessidade de intervenção humana (FISCHER, 2001).

Informação é a base de processos automatizados pela Tecnologia da Informação e a maioria das organizações usa as informações em todos os aspectos de seus negócios (PEROTTONI et al., 2001). A maioria destas organizações possui algum tipo de sistema para tratar e processar as informações necessárias nos diversos segmentos de suas atividades. Eles têm sido construídos ao longo das últimas décadas por meio de diversas técnicas e métodos. Sendo estruturados, planejados e organizados para armazenarem informações para o suporte à tomada de decisões, a extensão e a qualidade deste repositório de dados é imensa, e sua utilização com agilidade nos processos das organizações é uma prática diária e corriqueira.

Ao mesmo tempo em que necessitam armazenar e utilizar informações, as organizações têm de produzir respostas aos estímulos de seus usuários externos, clientes, parceiros e colaboradores, ou respostas a estímulos legais gerados pelo Estado ou ainda, respostas aos estímulos internos gerados na seqüência de ações para atender a outras requisições (LEITE, 2001). Cada um destes eventos dispara o que se convencionou chamar de processo na organização. Por meio deles é que são produzidas as respostas esperadas.

GONÇALVES (2000) define processo como “o conjunto de atividades que toma um input, adiciona valor a ele e fornece um output a um cliente do processo”. JACOBSON, ERICSON e JACOBSON (1994) afirmam que processos não são novidade nas organizações, pois todas elas sempre os tiveram. Eles nem sempre são formalmente visualizados nem recebem nomes ou definições, ao contrário dos elementos da estrutura organizacional como departamentos e funções, mas atravessam as fronteiras departamentais na execução das atividades para produzir o resultado esperado.

Automatizar processo pode imprimir uma série de vantagens do ponto de vista organizacional. FISCHER (2001) categoriza os benefícios em benefícios tangíveis e benefícios intangíveis.

Benefícios tangíveis:

- a) Redução do custo operacional – organizações que implementam workflow têm reportado que seu custo de execução de transações foi reduzido;
- b) Melhoria na produtividade – tarefas rotineiras e repetitivas podem ser automatizadas e disponibilizadas mais facilmente por meios eletrônicos, como *e-Commerce*;
- c) Incremento dos tempos de processamento – além de suportar execução de processos automatizados, o que por si só é mais rápido que execução

manual, workflow suporta processamento paralelo, que permite a execução de muitas atividades ao mesmo tempo, reduzindo o tempo total.

Benefícios Intangíveis:

- a) Melhoria dos produtos e serviços – os gerentes afirmam que podem entregar melhores serviços aos clientes, pois a incidência de erros diminuiu drasticamente e eles possuem melhores informações decisórias;
- b) Melhoria de condições para execução do trabalho – a automação das tarefas repetitivas libera os empregados para realizarem tarefas de nível mais alto;
- c) Melhores meios para manter atualizados os processos – as empresas podem otimizar continuamente os processos de negócio;
- d) Qualidade – a diminuição de erros com a automação é um fator clássico de melhoria de qualidade do produto e do processo;
- e) Melhoria na comunicação – a disponibilidade de informações sobre o andamento dos processos melhoram a comunicação entre pessoas e entre empresas;
- f) Suporte à decisão – as decisões são tomadas com melhor qualidade de informação;
- g) Melhoria na capacidade de planejamento – o planejamento torna-se mais efetivo ao dispor de informações de melhor qualidade sobre os processos e seu progresso;
- h) Melhoria da capacidade de mudança estratégica – rapidez para efetuar mudanças na estrutura organizacional, para responder às exigências do mercado e no aproveitamento de oportunidades;
- i) Melhoria na comunicação interorganizacional – workflow suporta atividades que se expandem entre muitas organizações, como em processos de *supply chain* entre organizações.

Assim, automatizar processos é uma atividade cada vez mais demandada pelas organizações e um desafio para os engenheiros de software, analistas, projetistas e programadores, responsáveis pela elaboração das aplicações. Eles terão de utilizar sua experiência com sistemas de informação - mesmo por que a integração com dados legados é imperativa - e apropriar-se dos conceitos de workflow.

2.1 Workflow

As tarefas de analisar, modelar e implementar aplicações de workflow introduzem alguns conceitos novos em relação às tarefas correspondentes em sistemas de informação, e entender e dominar estas diferenças é fundamental para instrumentalizar analistas de processos e de sistemas a atenderem os requisitos para automação de processos.

À abordagem de sistemas de informação, workflow adiciona o aspecto de coordenação de atividades ou controle de fluxo de atividades (BARTHELMESS, 1996). Esta coordenação é um dos pré-requisitos para alcançar os objetivos do processo de negócio. No processo de elicitação de aplicações de workflow o foco é na seqüência de atividades, sua ordenação, concorrência e paralelismo, no controle de sua ativação, execução e término, no assinalamento do recurso para a execução de cada atividade e no tratamento das informações necessárias ou produzidas em cada atividade (BARTHELMESS, 1996), (GORDJIN, AKKERMAN; VLIET, 2000).

Assim, workflow é a automação do processo em si, isto é, é a imagem automatizada do processo implementado com o objetivo de que se comporte de modo tão real quanto a arquitetura do motor de workflow permitir.

Um motor de workflow pode ser dividido em alguns componentes arquiteturalmente dispostos, responsáveis por alguma funcionalidade ou parte do trabalho.

2.1.1 Histórico

Nos anos 70, com a busca da eliminação do grande volume de papel, a motivação para as pesquisas em software de apoio ao trabalho em escritórios era grande. As primeiras experiências de implementação de workflow vieram dos trabalhos desta área, mas apesar de grande otimismo, as tentativas não foram muito bem recebidas.

Além das dificuldades tecnológicas, como a indisponibilidade geral de infra-estrutura de redes, a rigidez das especificações e implementações feitas foram a principal causa de sua rejeição. Para que as pessoas executassem suas atividades, elas tinham de infringir as regras e os procedimentos padronizados pelo workflow. A interferência foi tão forte e prejudicial que chegou a tornar as organizações ineficientes (NICOLAO, 1996).

Na década de 80, os sistemas de apoio ao trabalho em grupo se desenvolveram rapidamente. Muitos trabalhos acadêmicos na área foram feitos, e surgiram sistemas comerciais propondo suporte para CSCW (Computer Support Cooperative Work). Muito voltados para *groupware* nas organizações, a maioria auxiliava na coordenação das tarefas de grupos de trabalho heterogêneos, compostos por pessoas de diversas áreas. Um grupo de trabalho com estas características exigia mais comunicação, colaboração e coordenação entre seus membros (CAVALCANTE, 2000).

A tecnologia necessária – redes de computadores e Internet - estava tornando-se largamente disponível, mas a automação de processos, por causa da rejeição experimentada na década anterior, não acompanhou esta evolução. MATSUDA (2000) diz que durante a década de 90, além da melhoria das infra-estruturas de rede, a necessidade de redução de pessoal nas empresas e o trabalho orientado a processo deram novo impulso à automação de processos organizacionais.

Estes fatores certamente contribuíram para a aceitação de workflow, mas o crescente uso da tecnologia de informação nas empresas - principalmente com o uso de computadores pessoais na mesa dos executivos, a difusão das planilhas de cálculo, editores de texto, agendas, correio eletrônico, Internet e outros softwares, contribuíram para que workflow fosse visto pelas empresas como uma tecnologia capaz de proporcionar diferenciais de negócio.

Também a reestruturação de organizações apoiada em Reengenharia de Processos de Negócio (BPR - Business Process Reengineering), ou redesenho de processos, como também é conhecida, além de difundir e popularizar conceitos como processo e modelagem de empresas, passou a demandar a automação dos processos redesenhados.

2.1.2 Tipos de Workflow

As aplicações de workflow geralmente são categorizadas de acordo com o tipo de processo que elas automatizam. Assim, se um processo numa organização se classifica em determinada categoria, o workflow resultante se enquadrará na mesma categoria.

Não existem formas comuns para categorizar sistemas de workflow. Muitas taxionomias são utilizadas. NICOLAO (1996) faz uma análise de algumas delas e discute a clássica categorização de GEORGAKOPOULOS (1995): workflow *ad hoc*, administrativo e de produção.

Workflows do tipo *Ad Hoc* suportam definição rápida e execução de modelos de processos menos complexos. Este tipo de workflow executa processos de negócio onde não há um padrão de movimentação de informações entre pessoas e o trabalho é desenvolvido de forma mais colaborativa (NICOLAO, 1996).

Nestes casos o controle do fluxo geralmente é responsabilidade dos participantes, já que não é possível modelar todas as situações *a priori*. Eles são utilizados para apoiar atividades não estruturadas, que podem ocorrer uma única vez, como casos particulares num processo de negócio onde as exceções a modelar são tantas que nem todas podem ser previstas.

Workflow de processos administrativos são aqueles que automatizam atividades repetitivas cuja coordenação e fluxo é simples, com seqüência conhecida e fixa, pois poucas exceções podem ocorrer. (NICOLAO, 1996) afirma que este tipo de processo possui fraca necessidade de acesso a

sistemas de informação e suas informações possuem baixo grau de complexidade.

Já workflow de processos de produção envolvem igualmente atividades repetitivas e previsíveis, mas englobam um processo de informações complexas, com transformação destas informações, envolvendo acessos a múltiplos sistemas de informação (NICOLAO, 1996).

Neste tipo de processos as decisões podem ser tomadas com base em dados existentes fora do domínio específico do workflow, em sistemas legados. Também é possível que existam algumas atividades no fluxo que devam ser executadas de forma automática, sem intervenção de atores externos.

Workflows de produção são aplicações de larga escala, complexas e refletem a heterogeneidade do ambiente onde são executados (ALONSO et al., 1997), características comuns nos clássicos sistemas de informação corporativos.

2.1.3 Modelagem da Aplicação

A resposta à pergunta “*como representar um workflow de forma a gerar um WMS correspondente*” (JOOSTEN, 1995), onde o mapeamento entre os elementos dos modelos e os componentes do software construído seja feito de forma clara e precisa só é possível se a arquitetura utilizada para a solução de software tiver relação direta com os elementos do modelo (JACOBSON; ERICSON; JACOBSON, 1994). Desta forma, cada elemento do mundo real representado no modelo, sua associação com os outros elementos e a forma como o analista modela a interação entre eles deve ser a imagem do que realmente virá a ocorrer quando o motor de workflow executar uma instância do processo em questão.

Muitos destes modelos são feitos com grafos baseados em Redes de Petri. De forma genérica, diagramas baseados em Redes de Petri representam um sistema como uma coleção de lugares (*places*), transições, arcos e marcadores (*tokens*). Lugares contêm marcadores e funcionam como entrada para transições. Uma transição acontece quando um marcador se move para outro lugar, se existir um arco entre a transição e o lugar. Antes que a transição ocorra, um *token* precisa estar presente em cada lugar em que a transição é conectada por um arco (BARTHELMESS; WAINER, 1996b).

Embora a maioria dos formalismos de representação de processos seja baseada em Redes de Petri (BARTHELMESS; WAINER, 1996b), os diagramas de atividade possuem uma semântica mais adequada para esta tarefa. Uma Rede de Petri modela sistemas ativos, isto é, o próprio sistema modelado é o principal responsável pelas ativações e eventos, ao contrário, diagramas de atividades modelam sistemas reativos onde o ambiente que circunda o sistema pode causar as trocas de estados no sistema (ESHUI; WIERINGA, 2001a), numa relação de simbiose entre o workflow, sistemas de informação e os atores do processo.

Os diagramas de atividade da UML, Unified Modeling Language [...] (OBJECT MANAGEMENT GROUP, 2003) são gráficos que representam o fluxo de controle de uma atividade para outra, empregados para a modelagem de aspectos dinâmicos de sistemas. Os diagramas de atividade dão ênfase no fluxo de controle entre as atividades, ou seja, a execução não-atômica das atividades em andamento em uma máquina de estados ao longo do tempo (BOOCH; RUMBAUGH; JACOBSON, 2000).

As técnicas baseadas em atividades propiciam boa elicitación de requisitos para implementar automação de processos, boa comunicabilidade e poder de expressão (BARROS, HOFSTEDE, 1996). Um motor de workflow que implemente uma máquina de estados baseada em atividades pode ser utilizado de forma acoplada com gráficos de modelos de atividades para elicitar requisitos de automação de processos.

BASTOS; RUIZ (2002) estendem o diagrama de atividades com estereótipos para modelagem de workflow em sistemas de produção [...] (WORKFLOW MANAGEMENT COALITION, 1995) e aponta um meta-modelo (BASTOS; RUIZ, 2001) capaz de suportar a execução do processo modelado. Esta abordagem endereça para um motor de workflow acoplado com as abstrações modeladas, uma arquitetura em perfeita concordância com a promessa que a análise da automação faz: de criar uma aplicação que seja a imagem dos requisitos modelados.

ESHUI; WIERINGA (2001b) discutem a semântica de diagramas de atividade para suportar máquinas de estado que realizem workflows baseados nestes grafos. São propostos alguns mapeamentos e decisões que introduzem maior poder semântico e capacidade para executar workflows a partir de diagramas de atividades da UML.

Com o diagrama de atividades em sua forma original [...] (OBJECT MANAGEMENT GROUP, 2003), é possível capturar ações, trabalhos e atividades a serem executados e seus resultados, ou seja, as mudanças de estado dos objetos. O diagrama de atividades permite expressar as ações executadas, quando são executadas, que objetos elas modificam e onde elas acontecem ao modelar um processo que ocorre em termos de fluxo e coordenação de atividades, informações que trafegam e os atores que interagem.

Dois tipos de atividades básicas são necessárias para executar um workflow de produção como discutido na seção anterior: atividades assinaladas para um ator e atividades temporais, executadas periodicamente de acordo com a definição do processo. O primeiro tipo é reativo aos eventos disparados pelo ator, o segundo é reativo à passagem do tempo, e pode disparar verificações de eventos ocorridos no ambiente, como por exemplo a troca de estados nos dados em sistemas de informação. Desta forma o próprio processo é o responsável por perscrutar os eventos que lhe interessem.

Decisões indicam diferentes possibilidades de transição no fluxo, dependendo da condição de guarda da transição [...] (OBJECT

MANAGEMENT GROUP, 2003). Fluxos concorrentes mostram as múltiplas transições, a partir da saída de uma atividade, que serão avaliadas pela máquina de estados e onde somente as condições retornadas verdadeiras permitem que as transições sejam realizadas.

Os conectores, elementos de imposição e controle de sincronia, representam a junção de linhas de atividades executadas paralelamente ou concorrentemente, mostrando o ponto no fluxo onde a seqüência de atividades se torna única. Eles “*Expressam a semântica do sincronismo baseada no término de atividades anteriores*” (BARTHELMESS; WAINER, 1996b).

Os disjuntores representam possíveis linhas de atividades executadas paralelamente, mostrando o ponto onde o fluxo se divide em várias seqüências de atividades concomitantes.

Nos diagramas de atividades, *forks* (*fork total* (CASATI et al., 1995)) e *joins* são implementados em barras de sincronização, *synch states* [...] (OBJECT MANAGEMENT GROUP, 2003). Um Fork é uma ramificação na transição entre uma atividade de origem e duas ou mais atividades de destino. Esta ramificação indica fluxos paralelos.

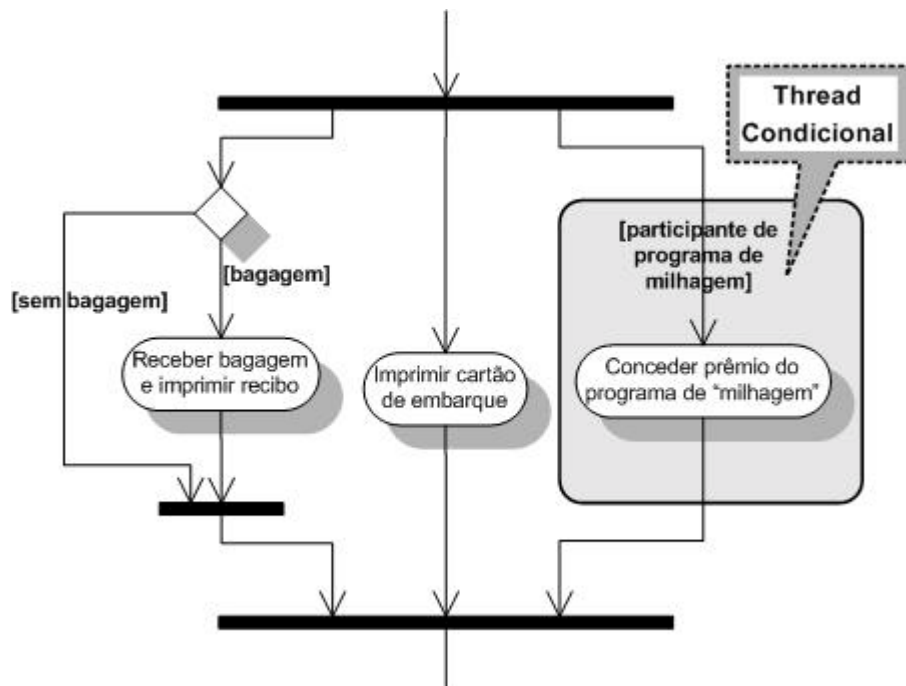


Figura 2.1: *Thread* em diagrama de atividades (ERICSON, 2002)

Outro tipo de ramificação é representada por uma *decision*. Ela implementa ramificações concorrentes, um *fork* condicional (CASATI et al., 1995) ou implícito (VAN DER AALST, W.M.P. 1994), que realiza todas as transições cuja condição de guarda seja verdadeira ou aquelas nas quais não existe condição de guarda.

Uma ramificação pode ainda implementar uma *thread* (ERICSON, 2002) – figura 2.1, uma transição em uma das ramificações paralelas, executada somente se a condição de guarda da transição for verdadeira.

Diferentemente das decisões, uma *thread* é uma ramificação paralela executada em casos específicos.

Também o início e o fim de um workflow devem ser marcados de forma precisa, pois demonstram quando a instância é iniciada e quando a instância será encerrada e passará a fazer parte do histórico de instâncias executadas (CASATI et al., 1995).

Os elementos básicos para representar as abstrações e para a execução de workflows estão presentes nos diagramas de atividade. Alguns mapeamentos e premissas podem incrementar o poder destes diagramas, permitindo que eles se tornem na principal ferramenta para todo o processo de desenvolvimento de aplicações de workflow, desde a fase de elicitação até a fase de implementação, quando as necessidades de manipulação pelos atores e os requisitos de integração com legado são também codificados.

2.1.4 Atores

A estratégia de seleção e assinalamento do recurso humano ou computacional para a realização da atividade tem sido um dos pontos-chave da modelagem e um dos requisitos mais difíceis de modelar.

Em modelos reativos, uma atividade entra em estado de espera tipicamente quando sua execução depende de um evento externo ou da ocorrência de um sinal [...] (OBJECT MANAGEMENT GROUP, 2003). Ao ser assinalada para um ator ou participante, o controle sobre o estado da atividade passa do motor de workflow, para o qual a atividade entra em estado de espera, para o ator, fazendo com que o fluxo permaneça aguardando que o ator dispare o evento que marca a finalização da atividade para que a transição seguinte possa ocorrer.

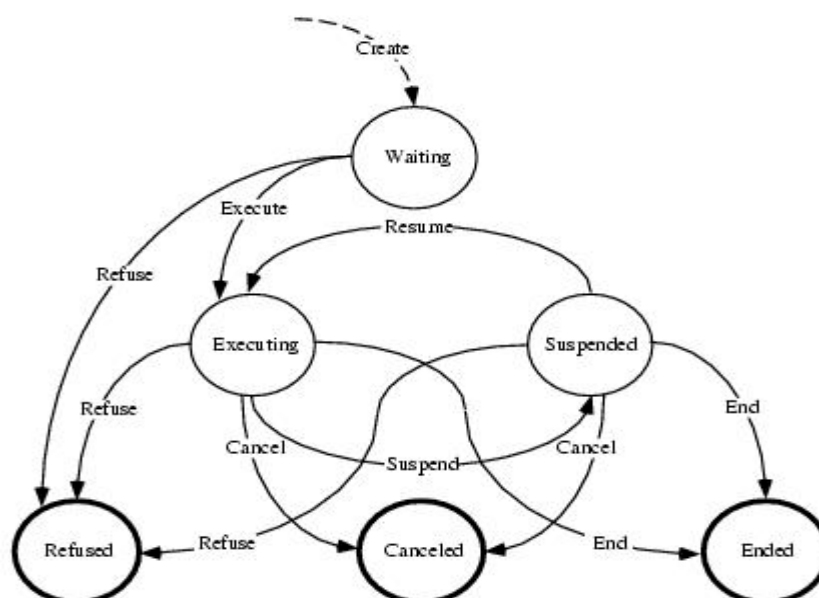


Figura 2.2: Interação entre atores e atividades (CASATI et al., 1995)

CASATI et al. (1995) discute a arquitetura de um WMS onde a interação entre um ator e uma atividade depende dos eventos disparados pelo ator, descrevendo atômica e as transições, mostrada na figura 2.2. Os atores alteram o estado das atividades assinaladas para si. Uma atividade assinalada para um ator fica aguardando a ação do ator, isto é, aguarda que o ator inicie a execução da atividade, quando ocorre a transição para o estado “*executando*”. A partir de então, a atividade pode ser finalizada, cancelada ou recusada pelo ator. Se ocorrer a transição da atividade para um destes estados, o controle volta para o WMS para que seja decidida qual a próxima ação. O ator pode ainda suspender a atividade para retornar posteriormente a sua execução. Se existir um mecanismo de *time-out*, o WMS pode retomar a atividade para seu controle, retirando-a do controle do ator.

O WMS deve ter capacidade de selecionar o ator de acordo com a política definida para a execução de determinada atividade, baseado num algoritmo de avaliação destas regras. Isto pode ser feito automaticamente pelo mecanismo de workflow, utilizando uma estratégia *push* ou assinalamento passivo, ou uma estratégia *pull* ou assinalamento ativo. Esta estratégia de assinalamento descrita em CASATI et al. (1995) também é mostrada com poucas alterações pelo WMC [...] (WORKFLOW MANAGEMENT COALITION, 1998).

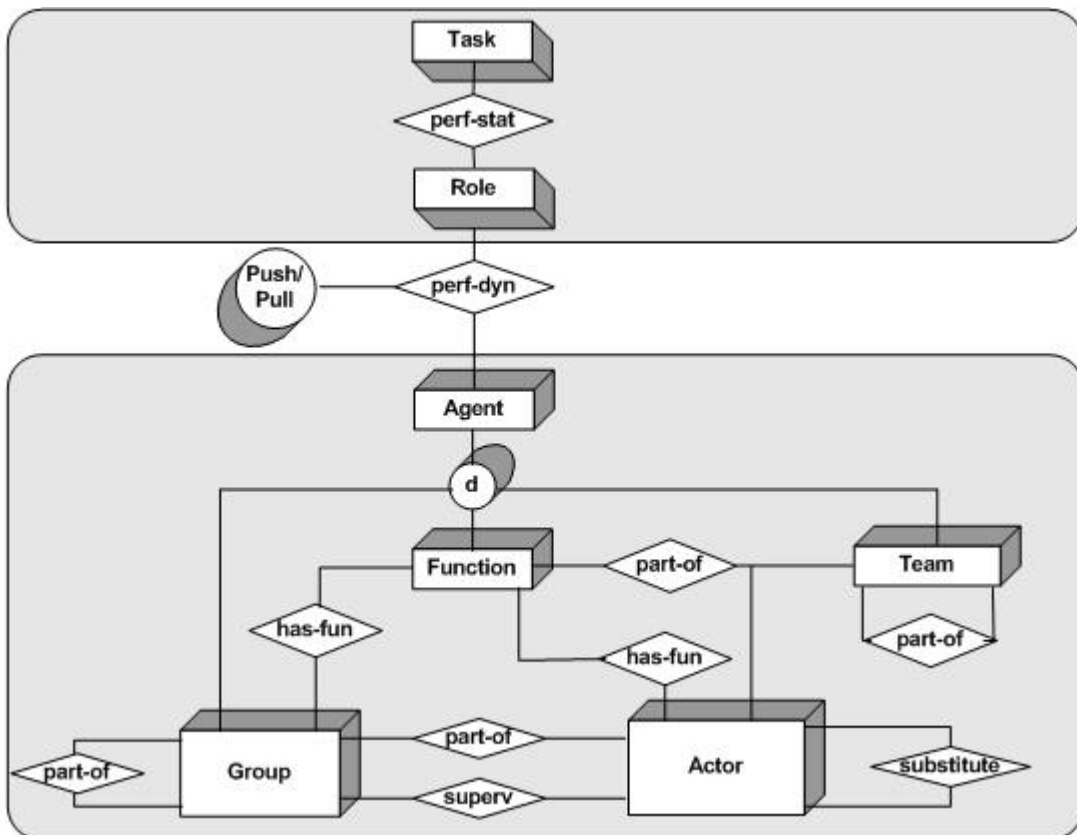


Figura 2.3: Estrutura de modelagem de atores (GUTIERREZ, 1997)

Na primeira estratégia o WMS entrega ou envia a atividade para o ator selecionado, responsabilizando-o pela execução. Na segunda, o

workflow entrega ou envia a atividade para um conjunto de atores com habilidades para executá-la, depositando-a numa lista compartilhada por atores que desempenham um mesmo papel ou que participem do mesmo grupo. O ator que primeiro selecionar a atividade para si, será assinalado pelo WMS como o responsável pela execução, que retira a atividade da lista compartilhada e a coloca na lista individual do ator (CASATI et al., 1995).

GUTIERREZ (1997) mostra uma modelagem organizacional, figura 2.3, que busca apreender a estrutura da organização em termos de quatro elementos básicos para subsidiar o assinalamento de atividades para atores:

- a) Ator: um ente individual que pode ser de natureza humana ou não.
- b) Grupo: a especificação de uma classe de atores baseada em características organizacionais comuns.
- c) Função: a especificação de uma classe de grupos, atores ou times que têm características comuns fundamentada no que eles podem fazer e o que lhes é permitido fazer.
- d) Time: uma lista de funções baseada (parte de) na estrutura organizacional. Um time pode conter duplicatas, por exemplo, um time de projeto consiste em um diretor geral, um gerente técnico, e dois investigadores.

As atividades do workflow seriam executadas por uma única entidade de processo. Um “agente” selecionado dinamicamente de acordo com os dois elementos básicos do workflow:

- a) Papel: uma descrição das habilidades exigidas para executar uma atividade específica.
- b) Agente: uma entidade de processo que pode executar (direta ou indiretamente) tarefas numa execução de workflow.

Esta estrutura suportaria a seleção automática de executores (BARTHELMESS; P., 1996), onde o próprio sistema se encarrega de selecionar os candidatos e assinalar a atividade para o ator correto. O problema com abordagens deste tipo é que a tarefa de modelar atores como membros de estruturas e conjuntos na organização se torna tão grande quanto a tarefa de modelar o próprio fluxo.

Os processos atravessam fronteiras departamentais e em muitos casos sua execução é atribuída aos atores, independente das posições ou funções que ocupam na estrutura hierárquica, dependendo muito mais de sua posição frente aos processos dos quais participa que de seu cargo. Modelar competências dos participantes do workflow pode implicar em modelar todos os atores da organização que um dia poderão vir a participar do processo.

Na administração pública é clássico o fato dos funcionários públicos se enquadrarem em cargos genéricos na estrutura, mas serem designados

para tarefas diversas. Assim, as responsabilidades não são limitadas pelo cargo, mas pelas atribuições específicas. Também a mobilidade de pessoas através das atribuições é grande e gerenciá-las é uma função complexa exercida pelos órgãos de recursos humanos.

Do ponto de vista da execução do workflow interessa o conjunto de permissões de cada ator, ou seja, o conjunto de permissões que cada funcionário possui frente a um processo. Este conjunto de permissões é singular e não corresponde exatamente a uma posição na hierarquia funcional e não é decidido em função disto. Ele deve permitir que alguns atores executem a atividade e que outros atores possam acessar a atividade sem permissão de executá-la, controlando o acesso dos atores.

Este modelo é similar aos conceitos de segurança de acesso às aplicações utilizados em muitos sistemas de informação atualmente. Ele é baseado em grupos que possuem um conjunto definido de permissões ao qual são adicionados os atores com permissão para aquele conjunto de operações. Os grupos podem ainda estar relacionados com o domínio de rede ao qual pertencem os atores.

Em aplicações de workflow, os grupos permitiriam executar alguma atividade e os atores inseridos no grupo herdariam esta permissão. As estratégias de assinalamento *PUSH* ou *PULL*, definidas para a atividade, utilizam o grupo e o domínio para descobrir contra quais atores disparar a atividade. A gestão é sempre feita para aquele workflow específico pelo proprietário do processo.

Em situações mais complexas, em que BARTHELMESS; P., (1996) propõe o uso de “regras”, como no exemplo de devolução de valores onde devoluções superiores a determinado valor deveriam ser aprovadas por um gerente e não por um subalterno, pode-se utilizar uma solução de análise com a criação de duas atividades concorrentes cuja execução depende do valor da devolução. Assim, cada atividade possui um grupo diferente e no grupo com permissão para a atividade de aprovar devoluções vultosas estaria presente somente o gerente.

Este tipo de solução pode resolver a maioria dos problemas reais. A simplificação da seleção de atores de modelagem organizacional para atribuição de permissões de acesso, além de utilizar um conceito largamente conhecido da comunidade de sistemas de informação permite que o foco principal do trabalho do analista de sistemas e do projetista seja fixado na implementação do workflow sem o risco de desviar o foco para problemas organizacionais, geralmente tratados no âmbito de redesenho de processos.

2.1.5 Dados e Informações

Ao interagir numa aplicação de workflow os atores inserem dados relevantes à tramitação do processo, para apoiar as atividades e documentar o trabalho, geralmente utilizando formulários eletrônicos, recurso largamente utilizado por muitos anos em sistemas de informação. Alguns dados também podem ser oriundos das bases legadas existentes.

JOOSTEN; B. (1996) reporta que um dos maiores problemas tecnológicos é a integração de dados legados ao workflow, já que ainda não existe solução padrão e cada ferramenta utiliza uma solução proprietária. Algumas diretrizes para analisar as necessidades são apresentadas em forma de questões:

- a) Quais atividades do workflow já são suportadas por aplicações?
- b) Quais as opções de integração existentes para cada caso?
- c) Quais são desejáveis?
- d) Qual o esforço necessário para cada caso?
- e) Quais alternativas possíveis (reescrever aplicações, etc)?
- f) Quais tendências podem ser antecipadas?

A estas podem ser adicionadas mais algumas, buscando entender como o workflow se relaciona com os dados que deverá manipular:

- a) Quais informações são necessárias ao processo?
- b) Em que momento elas são necessárias?
- c) Qual a fonte de cada dado em cada atividade?
- d) Quais dados são necessários em cada decisão no fluxo?
- e) Existe possibilidade de acessar os dados necessários existentes em bases legadas e encapsulá-los na atividade?

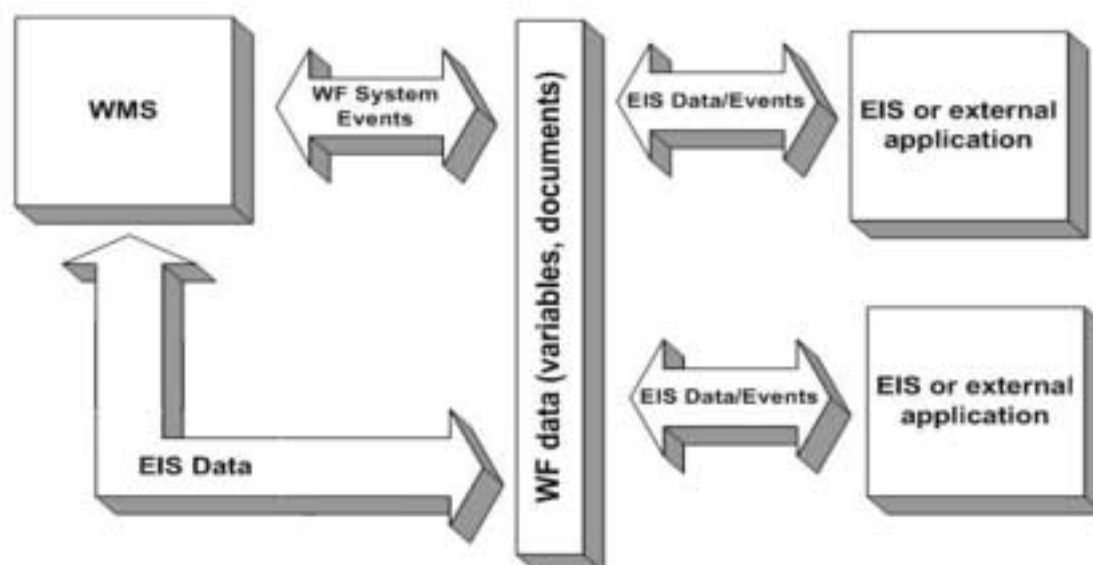


Figura 2.4: Interfaceamento de dados (BARESI et al., 1999)

BARESI et al. (1999) afirma que a principal preocupação na análise de informações legadas é a interação entre o legado e o workflow. Outra preocupação é a elicitación de quais informações são necessárias e em que

formato. É sugerida a modelagem da interação com o uso de cartões CRC (Class-Responsability-Collaboration). O mapeamento dos tipos de dados e conversões é feito durante esta análise. Um diagrama de interação é construído para demonstrar como os objetos colaboram com o workflow. Ele apresenta graficamente os pontos onde o workflow e a aplicação legada interagem e trocam informações.

Identificar as informações que fazem parte do processo e que são necessárias a sua execução, seja para que os atores possam tomar conhecimento do caso em execução, para armazenamento histórico junto ao processo “arquivado” ou para a tomada de decisão do motor de workflow, é uma tarefa de elicitação de requisitos do workflow. Estes são os dados do caso.

Muitos pesquisadores e o WMC classificam os dados do domínio de workflow em categorias de acordo com sua origem e uso. O WMC [...] (WORKFLOW MANAGEMENT COALITION, 1995) classifica os dados em:

- a) dados de controle: dados internos gerenciados pelo WMS que dizem respeito ao status das instâncias de processo e de atividades;
- b) dados relevantes: dados usados pelo WMS para determinar as transições de uma instância de processo;
- c) dados de aplicação: são os dados específicos do caso, as informações que os processos tratam como valores, matrículas que são manipuladas pelos atores do processo.

Percebe-se a divisão entre dados específicos do processo e dados necessários para decidir sobre as transições. ESHUI e WIERINGA (2001) elaboram uma semântica para os diagramas de atividade onde dois tipos de dados, os que o WMC chama de dados relevantes e de dados da aplicação, são dados do caso, uma vez que as condições de guarda, que determinam a possibilidade de transições entre as atividade, também determinando a direção do fluxo, são avaliações sobre dados do caso, manipulados por atividades anteriores – dados que os atores introduziram em atividades anteriores ou trazidos para o caso a partir de bases legadas.

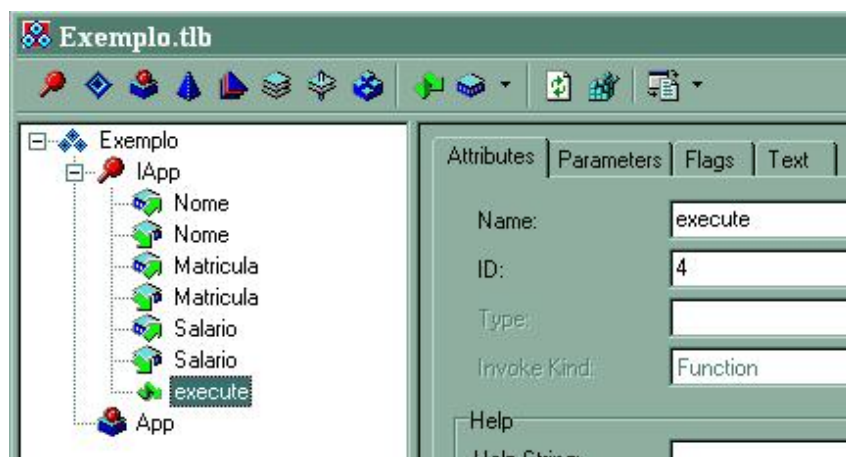


Figura 2.5: Interface utilizada para troca de dados e disparo de eventos

Esta abordagem simplifica o tratamento dos dados que os casos devem manipular uma vez que o workflow se apropria de todos os dados, armazenando-os e os ligando-os ao caso específico, encapsulando tudo o que se refere ao caso. Os dados legados podem ser trazidos para o domínio do workflow por meio de mecanismos de interfaceamento, como na figura 2.4, onde o workflow não sabe o que ocorre atrás da interface mas consegue tanto trazer quanto enviar dados para aplicações legadas.

Esta solução pode se basear no padrão *Command* (GAMMA et al., 2000) onde uma requisição é feita por um cliente, no caso uma instância de atividade onde tenha sido prevista a necessidade de apropriação de dados legados que instancia e popula um objeto e dispara o método padrão de execução sem conhecimento de suas propriedades. O objeto pode implementar, na execução, o algoritmo necessário para acessar os dados, tratá-los e devolvê-los à atividade. A atividade recolhe os dados na interface do objeto, figura 2.5, e armazena-os de acordo com a definição dos dados do processo.

Modelar os dados em abstrações lógicas que os agreguem introduz mais expressividade e possibilita uma série de operações sobre este conjunto de dados, mas existe a dificuldade relacionada ao seu armazenamento. É o problema clássico em manter a conformidade do conteúdo em relação à estrutura durante a evolução histórica do documento.

Um processo tende a evoluir, os documentos utilizados se modificam ao longo do tempo fazendo com que ao mesmo tempo, cada processo utilize documentos com estrutura variada: como apreender esta estrutura para manipular estas informações? As recentes soluções baseadas em XML [...] (WORLD WIDE WEB CONSORTIUM, 2000) podem auxiliar na solução deste problema pois permitem que estrutura e conteúdo sejam encapsulados em um só objeto, o que significa que ao acessar o objeto, a estrutura e o conteúdo distribuído nesta estrutura são conhecidos.

2.1.6 Arquitetura

A arquitetura usual nos WMS atuais busca cobrir o maior número de requisitos e atender ao maior número de cenários possíveis (MANOLESCU, 2001). Elas são ferramentas voltadas ao usuário final, fornecendo facilidades para que usuários não técnicos os utilizem e para isto, são implementados em pacotes de funcionalidades fechados cujas API's nem sempre são flexíveis (MANOLESCU, 2001).

O alto acoplamento das características de um WMS nas funções das ferramentas, figura 2.6, tornam sua utilização em ambientes de desenvolvimento de sistemas uma árdua tarefa para analistas e programadores.

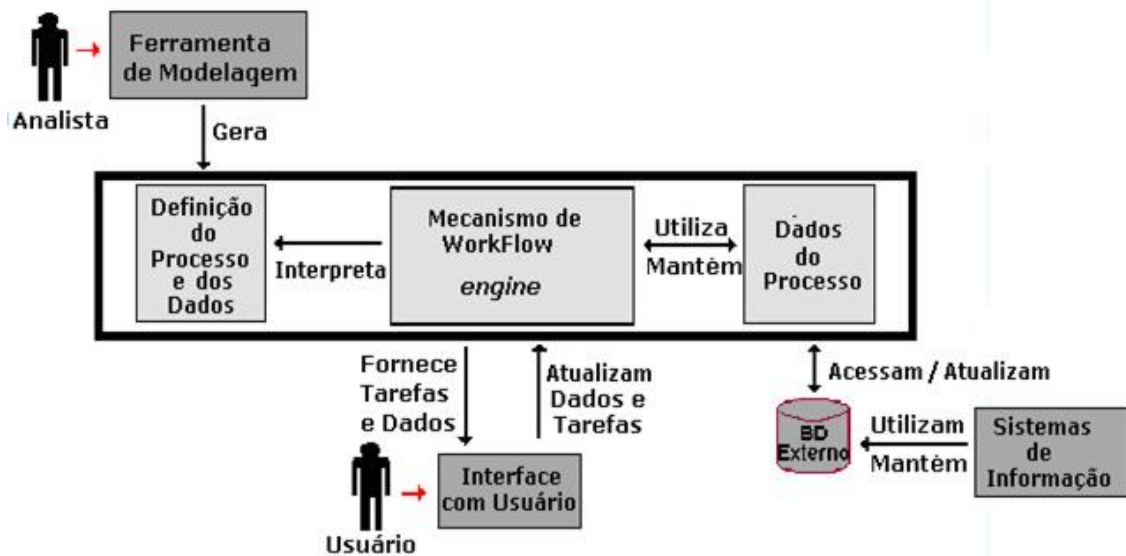


Figura 2.6: Arquitetura genérica de um motor de workflow

Por exemplo, muitos WMS suportam a chamada ou se integram à ferramentas que os atores utilizam no processo para executar suas atividades. Eles possuem capacidade para manter arquivos em diversos formatos e controlam o acesso e o versionamento, fazendo muitas vezes o papel de gerenciadores de documentos (GED). Este controle é feito em uma arquitetura cliente-servidor onde o ator está conectado ao servidor do WMS e trabalha no ambiente do WMS. Outro exemplo é a necessidade de customização de interface com os atores, que na maioria dos sistemas é fixa (BARTHELMESS; WAINER, 1996b), exigindo grande esforço para escrever camadas de apresentação utilizando APIs não estruturadas e proprietárias.

MANOLESCU (2001) afirma que a padronização do WMC reflete a arquitetura destes WMS, reproduzindo seus problemas e impedindo a flexibilidade e escalabilidade de sistemas de gerenciamento de workflow. O particionamento funcional dos componentes dado pela padronização do WMC não tem sido suficiente para assegurar que os sistemas de workflow não reproduzam os problemas citados.

Por outro lado as pesquisas em torno de meta-modelos orientados a objeto (BASTOS; RUIZ, 2001), aproveitando o paradigma de classe como o molde para o objeto, para implementar o conceito de *template* da definição do processo e a criação da instância do processo (MANOLESCU, 2001), se beneficiando da capacidade de encapsulamento de dados (ESHUI; WIERINGA, 2001b) e comportamento das classes e da possibilidade de criar hierarquias de classes, têm dado um novo rumo na criação de sistemas de workflow. Esta estratégia também facilita a evolução do esquema do processo pois faz com que a execução do caso seja interpretada passo a passo em oposição à criação de toda a estrutura do *template* ainda no início do caso.

O paradigma orientado a objetos é o fundamento principal que tem orientado o desenvolvimento de sistemas atualmente. Ele tem levado a

construção de sistemas mais flexíveis e customizáveis, incentivando reuso de código e componentes. Um motor de workflow customizado para um determinado cenário pode ser parte do *framework* para suporte a construção de aplicações, apoiando tanto a automação de processos quanto a integração de sistemas. Além disto, tecnologias como XML [...] (WORLD WIDE WEB CONSORTIUM, 2000) podem ser utilizadas como suporte à evolução dos esquemas de definição de processos e de definição de dados.

Esta arquitetura se aproveita do *middleware* existente para sistemas de informação, do modelo de transação de bancos de dados para armazenar o estado da instância de processo e da infra-estrutura de correio eletrônico a qual os sistemas de informação têm utilizado em casos específicos, entre outros.

2.2 Sistemas de Informação

Os sistemas de informações implementam o processo de apreensão, guarda e transformação de dados utilizadas no trabalho operacional, administrativo e gerencial das organizações.

Os sistemas de informações são capazes de contribuir com:

- a) Redução dos custos das operações;
- b) Melhoria no acesso às informações;
- c) Melhoria na produtividade;
- d) Melhoria nos serviços realizados e oferecidos;
- e) Melhoria na tomada de decisões;

À medida que o tamanho e a complexidade da organização e do ambiente onde ela atua, aumentam, torna-se mais complexa a manipulação de informações produzidas no dia a dia. Os sistemas de informações introduzem agilidade e capacidade de processar um grande volume de dados e produzir informações importantes (PEROTTONI et al., 2001).

Esta Informações devem subsidiar a identificação dos problemas e das necessidades organizacionais nos vários níveis da empresa (estratégico, tático e operacional), bem como fornecer elementos para avaliar o impacto das diversas decisões a serem tomadas.

Os sistemas de informação são sistemas reativos às ações dos seus usuários. Estes introduzem dados em interfaces que simulam formulários eletrônicos, agregações lógicas de informação, que passam pelo crivo das regras do domínio impostas pelo sistema, que ao aceitar as informações, persiste-as geralmente em bancos de dados.

Num sentido restrito, sistemas de informações são responsáveis pela captação, validação e guarda de dados de um domínio produzidos por organizações na execução de suas atividades.

2.2.1 Histórico

A evolução dos sistemas de informação está ligada diretamente à evolução da computação: das plataformas de hardware, rede e das técnicas da Engenharia de Software.

PRESSMAN (1992) descreve a evolução do software em eras apresentadas na figura 2.7. A primeira era ocorre nas décadas de 50 e 60. O software era construído de forma artesanal, isto é, o próprio usuário era o responsável pela elaboração do software de que necessitava. Eram sistemas baseados no processamento de dados em lotes (*batch*), com distribuição limitada, que atendiam aos departamentos de grandes empresas.

A segunda era ocorreu de meados da década de 1960 até o final da década de 1970. A multiprogramação e os sistemas multiusuários trouxeram novidades na interação homem-máquina. Sistemas de tempo real coletavam, analisavam e transformavam dados de múltiplas fontes, controlando processos e produzindo saídas em milissegundos. O software produzido era mais amplamente distribuído, as “software houses” produziam software para mainframe e minicomputadores e projetos de desenvolvimento internos nas empresas produziram dezenas de milhares de linhas de código (PRESSMAN, 1992). Aplicações corporativas já não eram novidades e cresceu o número de organizações que passou a informatizar atividades operacionais.

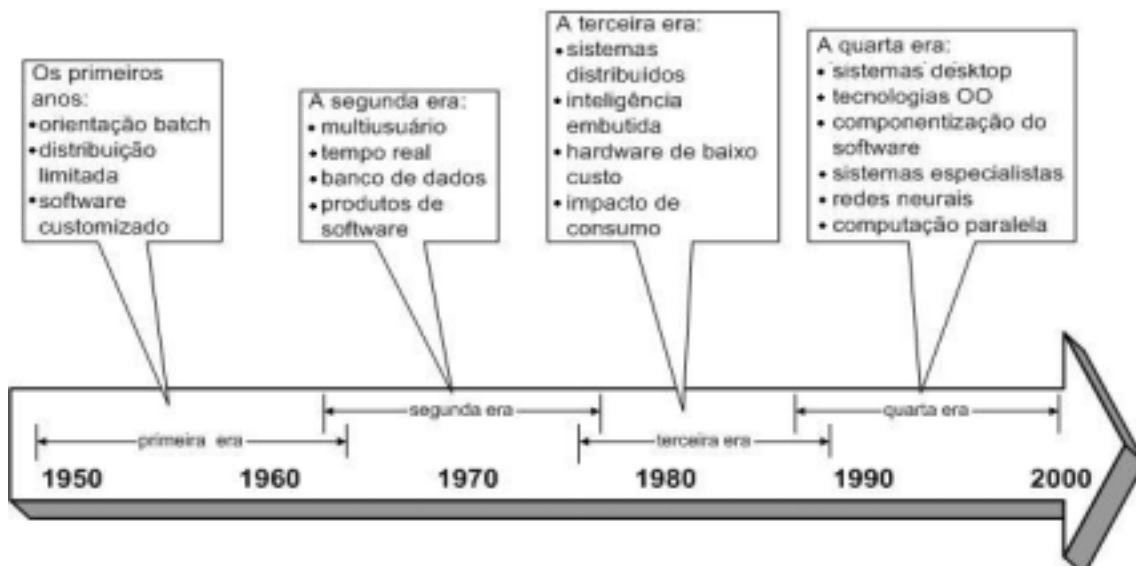


Figura 2.7: Evolução histórica dos sistemas de informação (PRESSMAN, 1992)

A terceira era começou em meados de 1970 com o surgimento de sistemas distribuídos aumentando a complexidade dos sistemas. As redes globais e locais, a comunicação digital e a crescente demanda por acesso “instantâneo” aos dados mudaram o cenário existente (PRESSMAN, 1992).

O computador pessoal catalisou o crescimento de empresas de software que passou a vender milhões de cópias. A produção de sistemas continuou crescendo e eles passaram a atender demandas setorializadas com a popularização de linguagens de fácil aprendizagem e de compiladores e ambientes para desenvolvimento em plataforma baixa. Também os softwares integrados passaram a ser distribuídos fornecendo pacotes de software corporativo para todas as áreas das empresas.

Atualmente estamos na quarta era com as tecnologias orientadas a objeto se popularizando, saindo das universidades em direção das empresas. Paralelamente surgem *frameworks* que encapsulam muitas funcionalidades e serviços comuns aos sistemas de informação, facilitando a criação de aplicações de tal forma que se vislumbra a possibilidade de atender a demanda reprimida por sistemas. Também a componentização do software possibilita reuso de peças em ambiente de execução, fazendo com que os sistemas de informação possam ser montados a partir de componentes existentes, economizando esforço, tempo e incrementando qualidade no produto.

Paralelamente os métodos para elaboração de sistemas de informação têm evoluído, levando o esforço para desenvolvimento de software da categoria de artesanato para processo de produção.

Em MANCUSO (1996) são encontradas definições para 3 grandes metodologias para apoiar o desenvolvimento de sistemas. Segundo ela, a análise estruturada foi desenvolvida na segunda metade dos anos 70. A metodologia estrutura o processo para uma abordagem *top-down*, segmentando o sistemas em módulos e foi criada para se integrar às técnicas de programação estruturada, com um enfoque unificado na solução de problemas.

A análise estruturada é a união de várias técnicas praticadas desde a década de 60 e novas técnicas surgidas na década de 70. O modelo principal é o diagrama de fluxo de dados (DFD) que exprime graficamente o movimento do dados através do domínio de um sistema. Com algumas variações podia incluir ferramentas como diagramas de estrutura de dados, tabelas de decisão, linguagem estruturada e dicionário de dados (MANCUSO, 1996).

A engenharia da informação de James Martin propunha um conjunto integrado de técnicas formais para a criação e derivação de modelos de empresa, modelos de dados e modelos de processos. Ela buscava fazer a integração de 4 faces distintas sustentadas pelos dados, pelas atividades, pela tecnologia e pelas pessoas. Estas 4 faces eram analisadas de forma *top-down* em etapas sucessivas abrangendo o planejamento estratégico, a análise das áreas de negócio e o projeto, a construção e a manutenção do sistema (MANCUSO, 1996) num ciclo de vida em cascata clássico.

Finalmente é apresentada a análise orientada a objetos e o projeto orientado a objetos, métodos para analisar a estrutura e o comportamento

de um sistema através da identificação dos objetos que fazem parte de um domínio (MANCUSO, 1996). Aqui a análise de dados e das possibilidades de trocas de estado destes dados permanecem encapsuladas pelos objetos. Uma outra inovação é a introdução do ciclo de desenvolvimento incremental e iterativo, onde partes do software são produzidas ciclicamente com cada iteração entregando mais funcionalidades em relação à anterior, com a maioria das metodologias sendo baseadas no Unified Process (UP) (BOOCH; RUMBAUGH; JACOBSON, 2000). Ultimamente outros processos orientados a objeto como Agile Process (COCKBURN, 2002) e Extreme Programming (BECK, 2000) têm surgido com novos conceitos e técnicas para elaboração de software.

2.2.2 Tipos de SI

Como na figura 2.8, que mostra os tipos de sistemas de informações computadorizados associados ao período em que começaram a ser desenvolvidos, cada sistema possui uma ênfase específica, podendo seus usuários serem diferenciados nos níveis hierárquicos da empresa, contendo assim características próprias.

O primeiro tipo de sistema desenvolvido foi o Sistema de Informação Transacional (SIT), também conhecido como Sistema de Processamento de Transações. Estes sistemas, ainda utilizados na maioria das organizações, monitoram, coletam, armazenam, processam e distribuem os dados das diversas transações realizadas dentro da empresa como compra de materiais, controle de estoque, faturamento, folha de pagamento e outros. Eles servem como base para os demais sistemas existentes dentro da empresa (PEROTTONI et al., 2001).

Após a organização das atividades operacionais, surgiu a necessidade de gerar informações consolidadas que auxiliassem os gerentes no processo decisório da empresa. Foram desenvolvidos os Sistemas de Informações Gerenciais (SIG). Um SIG extrai as informações de base de dados compartilhadas que estão de acordo com o que o SIG necessita para suas operações. Após a coleta dos dados e a transformação dos mesmos em informação, ele provê o gerente com informações passadas e presentes sobre as operações internas (PEROTTONI et al., 2001).

Com a evolução da tecnologia e dos sistemas de informações, surgiu a necessidade e a possibilidade de automatizar as atividades realizadas nos escritórios, surgindo assim, os Sistemas de Automação de Escritórios (SAE) que disponibilizam diversas funções, tais como: processadores de textos, agendas eletrônicas, editores de imagens e a possibilidade de gerenciamento de diversos tipos de projetos, entre outros (PEROTTONI et al., 2001).

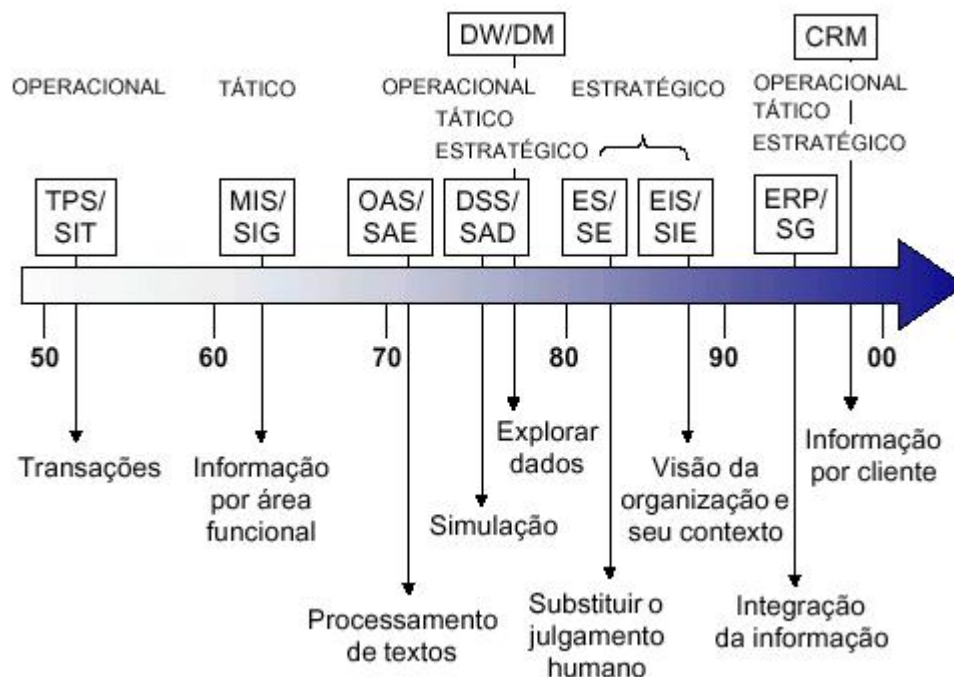


Figura 2.8: Sistemas de informação: surgimento e aplicação (PEROTTONI et al., 2001)

Após esta fase, o foco então passou para a busca e comparação de diversas alternativas para o mesmo problema, auxiliando o tomador de decisões. Surgiram os Sistemas de Apoio à Decisão (SAD), um tipo de sistema de informação computadorizado que fornece, normalmente, suporte às decisões semi-estruturadas e não-estruturadas tendo como principais características o uso de modelos e de dados de diferentes fontes, preocupação com o estilo do decisor e possibilidade de simulação (PEROTTONI et al., 2001).

Ainda nos anos 70, surgiu um novo conceito, o *data warehouse*, ou seja, uma base de dados que integrasse informações provenientes de outros bancos de dados e que pudesse ser explorada em busca de informações não facilmente percebidas (PEROTTONI et al., 2001).

Um *data warehouse* é um grande banco de dados consolidado e mantido separadamente das bases de dados dos sistemas da organização, podendo ser utilizada para relatórios e análises gerenciais em mais de uma dimensão.

A mineração de dados geralmente precede a criação destes grandes bancos. O *data mining* é utilizado para isto. O *data mining* é utilizado para transformar os dados coletados em dados de análise por meio do *data warehouse*, identificando oportunidades comerciais desconhecidas.

No início dos anos 80 surgem os Sistemas Especialistas (SE), tendo como objetivo a incorporação do conhecimento humano sobre atividades especializadas. (PEROTTONI et al., 2001)

Atividades especializadas requerem conhecimento específico em determinada área e para que essas atividades possam ser realizadas por pessoas que não detêm tais conhecimentos, começaram a ser desenvolvidos os Sistemas Especialistas com a finalidade de substituir o homem na solução de problemas mais específicos utilizando o conceito de *Inteligência Artificial*. Sendo assim, todo o conhecimento do especialista deve ser transferido para a máquina, possibilitando uma maior agilidade na conclusão das operações (PEROTTONI et al., 2001).

Os Sistemas de Informações para Executivos (EIS) possibilitam uma maior aproximação do executivo com os SI e a obtenção das informações em menor tempo, melhorando assim, as decisões estratégicas e possibilitando autonomia de pesquisa ao decisor. Eles surgiram nos anos 80, tendo como público alvo o nível estratégico das organizações, ou seja, os altos executivos das mesmas. O objetivo principal dos EIS é a filtragem dos dados mais relevantes para os executivos, reduzindo o tempo de obtenção e gerando informações de real interesse, as quais permitam o acompanhamento e controle da informações em vários níveis de detalhe (PEROTTONI et al., 2001).

Para disponibilizar para todos as informações dos diversos sistemas surgiram os Sistemas de Gestão Empresarial (SGE), tendo como objetivo principal prover todos os níveis da organização com informações sobre todos os setores, possibilitando um maior controle sobre suas operações (PEROTTONI et al., 2001).

Nos anos 60 surgiram os primeiros sistemas de manufatura digital que controlavam apenas o estoque de produtos da organização, proporcionando uma visão muito limitada do processo de produção. Depois de alguns anos o foco mudou para o planejamento de requisição de materiais (MRP - *Material Requirement Planning*), permitindo o controle do fluxo de componentes e matérias-primas, permitindo realizar o planejamento antecipadamente. A evolução destes sistemas redundou, por volta dos anos 90, nos sistemas denominados de Sistemas de Gestão Empresarial (ERP - *Enterprise Resource Planning*). Esses sistemas são constituídos por módulos integrados, permitindo a administração de diversas operações, tais como financeira, contábil, logística e recursos humanos, possibilitando um maior controle das operações e dos custos devido à forte integração das áreas citadas e dos dados dos sistemas (PEROTTONI et al., 2001).

2.2.3 Modelagem da Aplicação

Em ambientes de Engenharia de Software, o desenvolvimento de sistemas se dá de forma gradual e planejada. Uma analogia à construção de um prédio é muitas vezes traçada para que se possa compreender o processo e a evolução do trabalho no desenvolvimento de aplicações – figura 2.9. Em GOMIDE (2002) é apresentada uma comparação entre o processo de trabalho na construção civil e o desenvolvimento de sistemas, descrita a seguir.

Um arquiteto produz plantas arquitetônicas e entrega-as para um engenheiro civil. Este, produz outras plantas a partir das especificações do arquiteto. O engenheiro eletricitista, fará seus modelos a partir dos anteriores. Este processo de produção se estende até o mestre-de-obras, que comanda os operários na execução da obra, interpretando algumas destas plantas.

No processo sumariamente descrito acima, pode-se afirmar que os operários executam a construção de acordo com as especificações. As especificações foram criadas no transcorrer de um processo que iniciou com o tratamento do problema em um nível conceitual bastante alto, isto é, especificando “o que fazer”, e chegou ao nível de implementação, isto é, demonstrando “como fazer”, para só então executar a solução.

Igualmente, o processo de desenvolvimento de software conta com a participação de diversos especialistas. Existem profissionais que buscam compreender o objeto do produto de software, isto é, as necessidades que serão atendidas, modeladas sob ponto de vista dos clientes, denotando “o que” o produto deve fazer; também há profissionais responsáveis pela modelagem sob o ponto de vista do produto de software a ser desenvolvido, denotando “como” o produto deve fazer para atingir os objetivos e também os profissionais responsáveis pela transformação dos modelos em código executável de acordo com as especificações produzidas anteriormente, além de outros profissionais e especialistas. Assim, muitas pessoas estão envolvidas num processo de desenvolvimento de software, (GOMIDE, 2002)

No entender de JACOBSON; ERICSON e JACOBSON (1994), “A diferença entre assimilar os princípios e conceitos e estar habilitado a praticá-los é muito grande”. Por isto há necessidade de muito cuidado em desenvolver softwares grandes, complexos e com o gerenciamento de grandes equipes. A grande maioria dos pesquisadores propõe a utilização de uma sistemática de produção, para transformar o desenvolvimento em um processo de compreensão do problema e de construção de uma solução – um processo de Engenharia de Software.

Como em outras disciplinas de engenharia, a Engenharia de Software tem como objetivo a melhoria da qualidade do seu produto, com propostas de modelos de desenvolvimento, métodos e técnicas para aplicação nas diversas etapas de desenvolvimento do software.

A maioria dos ambientes de Engenharia de Software atuais utilizam técnicas orientadas a objeto no projeto de sistemas. Nelas, toda a modelagem é voltada para a arquitetura do software a ser construído (BOOCH; RUMBAUGH; JACOBSON, 2000). Esta característica faz com que todo o trabalho tenha um objetivo: alocar requisitos em peças de software que os realizem.

Quando isto não é possível, o resultado dependerá muito mais da experiência da equipe de desenvolvimento que da aplicação de técnicas, e a alocação será feita de forma empírica (BARESI et al., 1999), sem formalismo na derivação das abstrações para o software (MANCUSO, 1996). A descrição dos procedimentos, refinadas sucessivamente em especificações para a construção de componentes de software, fazem com que ao final do processo, seja muito difícil relacionar o componente desenvolvido a uma das abstrações modeladas. Daí a importância da estreita relação entre o modelo e arquitetura, entre modelo e componentes de software. Isto reflete diretamente na utilização de workflow neste mesmo contexto de Engenharia de Software.



Figura 2.9: Modelo e objeto da modelagem no mundo real

As atividades nas quais os modelos são executados fazem parte de um mesmo processo que objetiva apreender as necessidades de quem pretende automatizar algumas de suas tarefas até que um programador possa construir algoritmos em uma linguagem definida para gerar os componentes de software projetados.

Este processo é também chamado de processo de desenvolvimento ou metodologia, esquematizado na figura 2.10. Ele reúne um conjunto de técnicas e procedimentos que visa atender o processo de trabalho em um setor, área, divisão ou empresa que desenvolva sistemas, se propondo a organizar os projetos em um ciclo de vida, dispondo as etapas de forma que permitam ir avançando, ordenadamente, com cuidado e detalhe, evitando desvios em relação às idéias originais, com início e fim planejados, com produtos tangíveis e verificáveis, propondo a aceitação das partes envolvidas com as conclusões de cada atividade finalizada e escriturando todo o processo (ARAUJO; GOMIDE; PETRILLO, 2003).

A maioria dos processos corporativos atuais são baseados no Unified Process (BOOCH; RUMBAUGH; JACOBSON, 2000). Ele recomenda que o processo de desenvolvimento tenha as seguintes características:

- Seja iterativo e incremental;
- Seja guiado por casos de uso;
- Seja centrado na arquitetura do software;

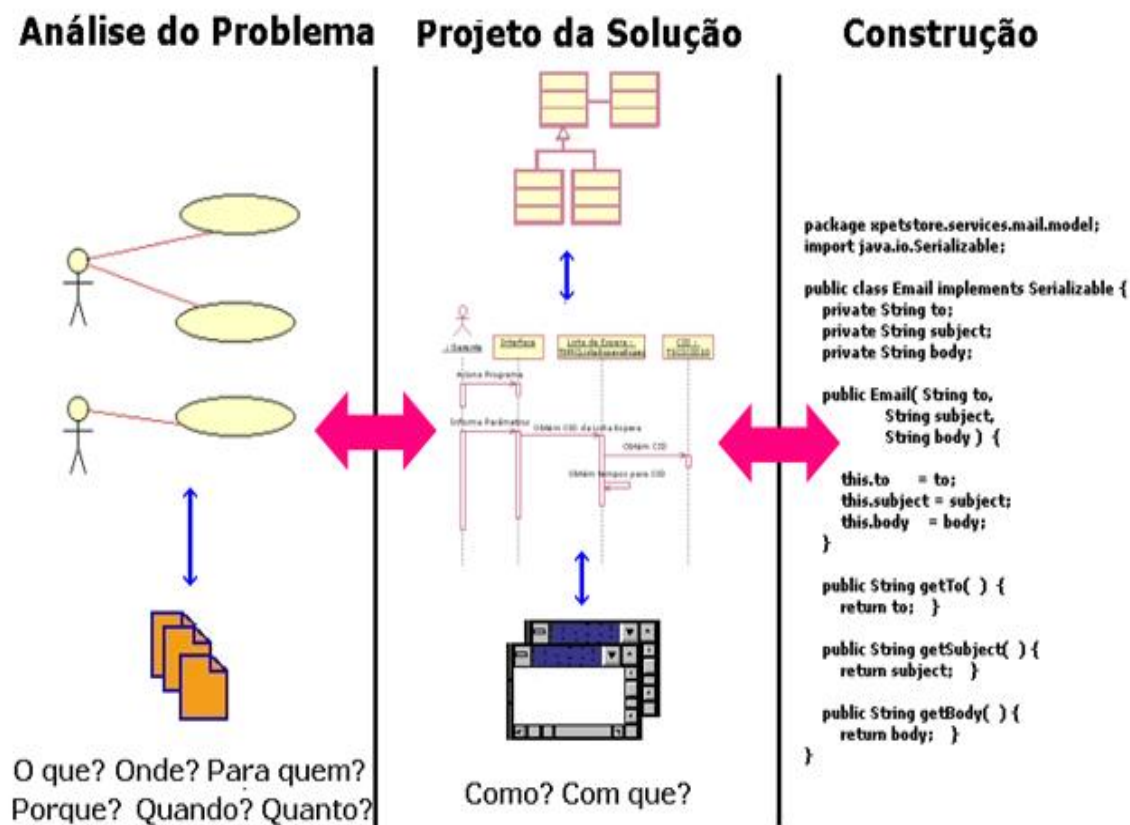


Figura 2.10: Processo dirigido por casos de uso

Um ciclo de vida iterativo busca o aperfeiçoamento sucessivo de um sistema por meio de múltiplos ciclos de análise, projeto e construção (LARMAN, 2000). Cada ciclo trata de um conjunto relativamente pequeno de requisitos e o sistema cresce à medida que cada ciclo é completado, daí este processo ser caracterizado como iterativo e incremental (LARMAN, 2000).

O processo de desenvolvimento é organizado em torno dos casos de uso pois estes refletem os principais requisitos funcionais de um sistema, que influenciam as estimativas, as atividades para implementá-los e os outros modelos (LARMAN, 2000).

A arquitetura de um sistema pode ser entendida como sua estrutura lógica e física, um conjunto de classes e responsabilidades (LARMAN,

2000), nela são definidos todos os componentes distribuídos nas diversas camadas que formam a aplicação (ARAUJO; GOMIDE; PETRILLO, 2003).

As práticas da utilização destes processos têm introduzido qualidade, confiabilidade, repetitividade e algumas outras vantagens tanto no processo de trabalho para construir software quanto no próprio produto resultante do processo. Aprender com elas e utilizá-las no desenvolvimento de outros tipos de aplicação num ambiente de Engenharia de Software facilita o trabalho dos analistas de sistemas responsáveis por elaborar aplicações de workflow.

2.2.4 Atores

Os usuários ou atores dos sistemas de informação são aqueles que interagem com os sistemas através de interfaces de apresentação, utilizando a metáfora de formulários, mais conhecidas como formulários eletrônicos como na figura 2.11, onde são mostrados os dados e informações do sistema e onde é possível aos atores dispararem eventos contra o sistema.

No desenvolvimento de sistemas, duas têm sido as preocupações com os atores que irão interagir com o sistema e a primeira diz respeito à ergonomia que as interfaces de apresentação terão.

A ergonomia do software é estudada pela área de Interação Humano-Computador (IHC) e tem por objetivo principal fornecer aos pesquisadores e desenvolvedores de sistemas explicações e previsões para fenômenos de interação usuário-sistema e resultados práticos para o *design* da interface de usuário, dos modelos de interfaces e do processo de interação usuário-sistema: para desenvolver modelos abstratos do processo de interação compatíveis com as capacidades e limitações físicas e cognitivas dos usuários (LEITE, 2001).

Modelos sobre perfis dos usuários ajudam a compreender as diferenças individuais e a definir interfaces com usabilidade adequada para grupos de indivíduos com características semelhantes. Atributos como experiência de trabalho, usuários assíduos ou eventuais, experiência com computadores, entre outros, descrevem o macro-perfil do usuário e são fáceis de observar com técnicas simples, tais como grupos de interesse, investigação de contexto, observação de campo e questionários (LEITE, 2001).

A segunda preocupação diz respeito ao nível de acesso que o ator terá perante o sistema, restringindo o acesso a determinados eventos de acordo com o papel que o ator estiver representando.

Um ator representa um conjunto de papéis que os usuários dos casos de uso desempenham quando interagem com os casos de uso. Eles representam um papel que um ser humano, ou um dispositivo, desempenha com o sistema (BOOCH; RUMBAUGH; JACOBSON, 2000), como “caixa”, “cliente”, “operador” ou mais genericamente “usuário”. Em sistemas de

informação, os atores são os disparadores dos eventos que iniciam uma transação e também são os responsáveis por confirmá-las.

The figure shows two versions of a form for a 'Declaração Municipal' (Municipal Declaration). The top version is a paper form with rounded rectangular fields. It includes fields for 'Logradouro - Rua e Número', 'Loteamento / Condomínio', 'Nome Requerente', and 'Endereço para correspondência'. There are also fields for 'Quadra' and 'Lote', and 'Area Privativa'. A section titled 'Declaração Municipal' contains six checkboxes: 'Edificação habit familiar', 'Edificações Diversas', 'Edificação provisória', 'Parcelamento do solo', 'Aquisição de imóvel', and 'Diversos fins'. Below this is a 'DATA:' field with a grid for date entry and a line for 'Assinatura'.

The bottom version is an electronic form titled 'DM / Solicitação'. It has a similar layout but with rectangular input boxes. The checkboxes are arranged in two columns. At the bottom, there are four buttons: 'Salvar', 'Salvar e Sair', 'Excluir', and 'Voltar'.

Figura 2.11: formulário em papel e formulário eletrônico

Ao restringir as responsabilidades dos atores frente aos sistemas de acordo com suas responsabilidades no mundo real, alguns atores não poderão ter acesso às interfaces que disparam determinados eventos ou que apresentam determinadas informações. Um gerente, por exemplo, pode ter acesso a um objeto que dispare um evento de autorização de transações de financiamento e um caixa, acesso a um objeto que efetive transações de débito de cheques e crédito de depósitos em conta-corrente. O importante é que a segurança seja controlada de tal forma que as responsabilidades estejam mapeadas nas permissões.

Esta modelagem tem sido feita por meio de casos de uso que descrevem a seqüência de eventos que um ator que utiliza o sistema necessita disparar para completar sua tarefa (LARMAN, 2000). Destas descrições se depreende o conjunto de permissões que os atores possuirão. O sistema de informação controla o acesso por meio de *login* e senha, podendo autenticar o ator e verificar quais suas permissões de acesso (ARAUJO; GOMIDE; PETRILLO, 2003).

Como um caso de uso modela uma operação discreta que um ator necessita realizar, controlar o acesso à interface projetada para suportar a operação é controlar quem pode realizar aquela classe de operações. Esta restrição se faz somente em relação ao acesso à operação, já que sistemas

de informação não controlam um caso em execução como as aplicações de workflow. Sistemas de informação apresentam todas as informações com estado válido para aquela tarefa, permitindo ao ator interagir com informações relacionadas ao caso desejado ou com informações de outros casos.

As atividades em um workflow podem ser equiparada às funções de sistemas de informação que suportam as tarefas dos atores. Controlar o acesso à atividade ou à função pode ser feito de forma similar, com os sistemas de workflow adicionalmente controlando os dados do caso apresentados ao ator.

A segurança de acesso e controle de permissões geralmente faz parte do *framework* disponível para construção de sistemas. A maioria das soluções opta por uma estratégia de papéis ou grupos de atores. Um grupo possui um conjunto de permissões e um ator ao ser associado a este grupo herda estas permissões. Este modelo também é utilizado no ambiente Windows NT conjugado com o gerenciador de objetos COM+ (KIRTLAND, 1999) pode servir como fundamento para uma política de autenticação e controle de permissões.

2.2.5 Dados e Informações

Todos os dias, organizações mantêm e gerenciam uma quantidade enorme de informações persistidas por sistemas de informação em suas bases de dados (PEROTTONI et al., 2001). Estas informações têm sido coletadas durante décadas e formam, juntamente com o conjunto de sistemas desenvolvidos, um ativo intangível de considerável valor e são armazenadas em algumas variedades de formatos e de tecnologias de armazenamento.

Estas informações são elicitadas no trabalho de projetar e desenvolver os sistemas que implementam algoritmos ou outras técnicas, se responsabilizando pelo filtro - a validação sintática e semântica dos dados. Na análise orientada a objetos as abstrações, figura 2.12, que modelam as informações em um sistema de informação são as classes, que agregam um conjunto de dados representados por seus atributos, caracterizando a abstração que ela representa.

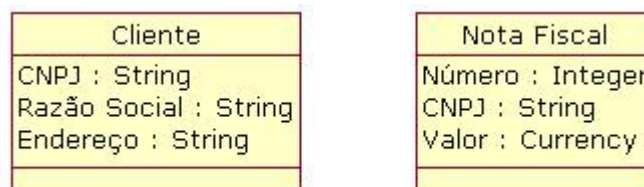


Figura 2.12: Classes típicas de sistemas de informação

Nas primeiras arquiteturas, os sistemas de informação eram construídos de forma monolítica e centralizada, sem desacoplamento entre a camada de apresentação, o processamento e o acesso aos dados. Era mais fácil administrar, controlar acesso, sintonizar a performance e prestar

suporte técnico em um ambiente centralizado. No entanto, sua flexibilidade era pequena, devido à utilização de tecnologias proprietárias; o custo era muito elevado para solucionar problemas de pequeno e médio porte, de forma que somente as grandes empresas eram capazes de pagar; a qualidade do serviço ficava abaixo das expectativas dos usuários, em virtude das limitações da tecnologia de interface dos aplicativos e, também, da falta de liberdade do usuário para manipular seus dados (SILVEIRA; BARROS; PEREIRA, 1999).

A evolução desta arquitetura levou à construção de sistemas onde o desacoplamento ou era feito ao nível de programa ou em nível de dados (SILVEIRA; BARROS; PEREIRA, 2000). No primeiro caso a aplicação possui um módulo que trata da interface e outro módulo que trata do processamento e dos dados, como em aplicações que utilizavam terminais de tele-processamento para interação com os atores do sistemas. No segundo caso é possível encontrar um módulo que trata dos dados e outro módulo que trata da interface e da lógica juntos, como em arquiteturas cliente-servidor.

Uma quarta arquitetura particionou os sistemas em camadas de responsabilidade separadas. Por meio dela a lógica de apresentação fica separada da lógica do processamento que por sua vez está separada da camada de persistência. A quantidade de camadas é na verdade uma função direta do framework utilizado para o desenvolvimentos de sistemas. O termo 3 camadas tem sido utilizado para definir as 3 camadas lógicas existentes.

Com esta separação as aplicações se tornam mais robustas e escaláveis, podendo atingir milhares de usuários, com múltiplos servidores heterogêneos interagindo com frequência, ampla distribuição geográfica, baseada em componentes e mantêm a lógica do negócio em servidores independentes situados nas camadas intermediárias (SILVEIRA; BARROS; PEREIRA, 1999).

Atualmente a principal forma de armazenamento das informações em sistemas de informação é a utilização de bancos de dados relacionais. Eles contêm informações sobre os dados e suas estruturas, incluindo os relacionamentos com outros elementos de dados, informações sobre chaves e índices de busca.

Atualmente muitos estudos têm sido efetuados para viabilizar o armazenamento de dados em formato XML (Extensible Markup Language). A possibilidade de manipular dados sem necessitar conhecer seu esquema estrutural incrementa a capacidade de integração entre aplicações e aumenta o grau de desacoplamento dos sistemas.

Infelizmente a performance da manipulação de dados armazenados em formato XML ainda não permite seu uso em larga escala. Mas em situações específicas, por exemplo, quando a estrutura de dados que a aplicação manipulará é variável e desconhecida, XML possui os requisitos necessários para permitir o tratamento das informações.

Dado que a lógica de negócio dos sistemas de informação é a principal responsável pela validação e transformação das informações, uma prática saudável na Engenharia de Software é buscar implementar integração de sistemas no nível da camada de negócios. Componentes desta camada seriam os responsáveis por fornecer interfaces de troca para recebimento ou fornecimento de informações e dados às aplicações clientes. Desta forma estariam resguardados os requisitos impostos às informações do domínio pelas regras do negócio.

Muitas vezes não é possível integrar sistemas ao nível da camada de processamento. Nestes casos os bancos de dados são a última instância de integração para estes sistemas - última instância porque não há outra forma de fazê-lo e também como referência à última camada lógica.

Isto pode causar problemas pois as regras impostas no domínio devem ser observadas pelo cliente, que não passa sua solicitação para um componente que valide os dados e devolva sua solicitação, mas ele mesmo acessa diretamente os dados no banco e deve se responsabilizar por aquelas regras. Aqui um domínio acaba tendo de conhecer as regras de um outro domínio, disseminando código específico de um sistema por vários outros sistemas.

Na integração de diferentes sistemas, muitas vezes construídos em tecnologias não compatíveis, a complexidade não reside somente na diferença das abstrações persistidas mas pela dificuldade intrínseca das tecnologias utilizadas em cada sistema. Muitas vezes ambientes diferentes têm de se comunicar para que a integração seja possível.

Algumas técnicas atuais como a utilização de adaptadores de recursos padrão como o JCA, Java Communication Adapter, da tecnologia J2EE ou o HIS, Host Integration Server, tecnologia COM+ da Microsoft, têm auxiliado na integração de dados entre plataformas diferentes.

Outra forma de integrar aplicações é através de Web Services ou SOAP. Web Services são interfaces publicadas que podem ser invocadas sobre protocolo HTTP. Eles provêm interfaces bem definidas que descrevem os serviços e são projetados para permitir uma integração sem acoplamento entre o cliente e servidor. Quer dizer, implementações de servidor não exigem para os clientes usem uma plataforma específica ou linguagem de programação. Além de definir interfaces independentes de linguagem, eles são projetados para permitir mecanismos de comunicações múltiplos.

2.2.6 Arquitetura

Uma arquitetura de software é uma visualização conceitual da estrutura de um aplicativo. Nela são definidos todos os componentes de hardware e de software que formam a aplicação (ARAUJO; GOMIDE; PETRILLO, 2003).

Os sistemas de informação atuais estão migrando rapidamente de arquiteturas de duas camadas, cliente-servidor, para arquiteturas de

n-camadas como a da figura 2.13. Elas introduzem uma série de vantagens que influem decididamente na escolha da arquitetura para sistemas. SILVEIRA, BARROS e PEREIRA (1999) enumeram algumas destas vantagens:

- a) A administração da distribuição da aplicação é mais fácil pois os componentes de código residem em servidores e não tem de ser distribuídos para os clientes. Para estes é necessário somente as interfaces de apresentação.
- b) Evita-se a exposição do esquema do banco de dados ao cliente e a autorização no servidor pode ocorrer a nível de serviço ou método;
- c) Os dados são encapsulados pois os clientes invocam serviços ou métodos;
- d) Aplicações n-camadas têm escalabilidade excelente devido à possibilidade de distribuir a carga de trabalho através de múltiplos servidores (Load Balance);
- e) Na arquitetura n-camadas, freqüentemente se reutilizam serviços e objetos;
- f) A integração com aplicações legadas se faz através de *gateways* encapsulados por serviços ou objetos;
- g) Com aplicações n-camadas é possível distribuir *thin clients* pela rede, os quais apenas invocam serviços remotos disponíveis nos servidores. (*É importante diferenciar thin cliente de poor client. Os primeiros são clientes leves, fáceis e cujas exigências de tráfego de dados na rede são baixas, os segundos são constituídos de interfaces pobres em funcionalidade e ergonomia*);
- h) A arquitetura n-camadas é capaz de usar múltiplos bancos de dados na mesma transação de negócio;
- i) A flexibilidade nas aplicações n-camadas é excelente, pois cada camada lógica pode ser colocada em um servidor distinto, se necessário;
- j) As aplicações n-camadas têm a capacidade de reiniciar componentes da camada intermediária em outros servidores.

Em contrapartida a complexidade de desenvolvimento aumenta, sendo necessárias boas ferramentas e ambiente de desenvolvimento integrados num *framework* produtivo que permitem desenvolver todos os lados da aplicação, o cliente, o servidor e a persistência.

Numa arquitetura n-camadas, a camada de regra de negócio é responsável pela aglutinação de todas as regras do domínio, formando o cerne funcional da aplicação. É nesta camada que se encontram os conceitos especificados na fase de análise (ARAUJO; GOMIDE; PETRILLO, 2003). Outras camadas da aplicação e até outras aplicações podem utilizar os serviços dos componentes da camada de negócio, entretanto ela não está associada a nenhum cliente específico.

Os componentes nesta camada também não tem nenhum conhecimento de como ou onde os dados sobre os quais ela atua estão armazenados, em vez disto ela utiliza os serviços de acesso a dados, que realizam o real trabalho de recuperar e armazenar dados em um banco de dados (ARAUJO; GOMIDE; PETRILLO, 2003).

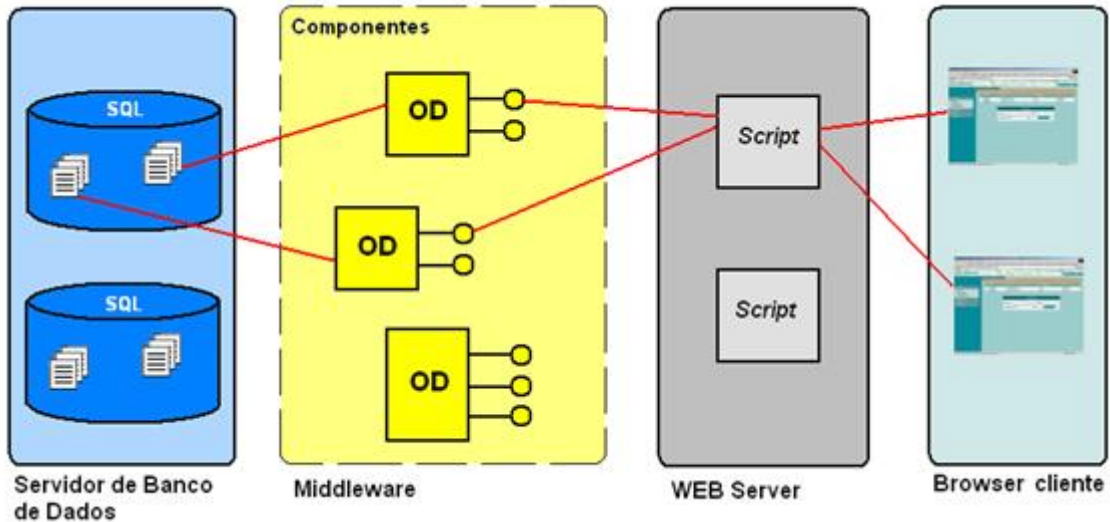


Figura 2.13: Arquitetura de uma aplicação n-camadas

A camada intermediária na maioria das aplicações n-camadas não é implementada como um programa monolítico, mas sim como uma coleção de componentes. Cada componente automatiza uma função de negócio relativamente pequena. Frequentemente os clientes combinam diversos componentes dentro de uma única transação de negócio. Um componente pode chamar outros componentes para auxiliá-lo a responder uma requisição. Além disso, alguns componentes podem agir como *gateways* para encapsular aplicações legadas rodando em *mainframe* (MAR99).

Atualmente os processos de desenvolvimento de sistemas propõem o uso de modelos capazes de modelar um software de modo que o modelo seja uma imagem dos componentes reais que serão colocados nos servidores (JACOBSON; ERICSON; JACOBSON, 1994), a visão lógica e a visão física são ambas necessárias, como na construção de um prédio, os modelos lógicos feitos nas plantas e maquetes se transformam em coisas físicas e reais (BOOCH; RUMBAUGH; JACOBSON, 2000). Onde existia o desenho arquitetônico de uma parede, a especificação de uma parede numa planta e o protótipo da parede na maquete, existirá uma parede real feita de acordo com estes modelos (BOOCH; RUMBAUGH; JACOBSON, 2000).

Com sistemas de informação, onde existem classes modeladas que abstraem os conceitos do domínio existirá um arquivo contendo o código daquela classe, onde há a especificação de atributos das classes existirá variáveis privadas para suportá-los, onde há a especificação de um conjunto de classes num componente existirá um arquivo físico, uma DLL ou um JAR, por exemplo, realizando aquele componente. Desta forma a arquitetura dirige os modelos, os projetistas e analistas sabem exatamente o

que devem projetar e aos programadores só resta a tarefa de codificar, pois todas as decisões sobre o software e sua construção já foram tomadas.

Construir software baseado nestas técnicas e premissas, facilita a manutenção do software, a evolução do software no decorrer do tempo (PRESSMAN, 1992), e a tarefa de integrar seus componentes em usos muitas vezes não previstos *a priori*, seja nas aplicações de workflow ou outros sistemas.

A seguir é apresentado um resumo dos diversos conceitos, como discutidos neste trabalho, comparando workflow e sistemas de informação como meio de apresentar as abordagens utilizadas e indicando formas de aproximação e possibilidades de melhoria no processo de elicitação e desenvolvimento e também na construção de WMS e motores de workflow:

Tabela 2.1: Características de Workflow e de Sistemas de Informação

Workflow	Sistemas de Informação
Tipos	
São caracterizados de acordo com o cenário da vida real que automatizam.	Caracterizados de acordo com a finalidade ou uso para a qual são criados.
Modelagem	
Apreensão do aspecto dinâmico dos conceitos do sistema	Apreensão da estrutura dos conceitos que formam o domínio do sistema.
Elicitação de um conjunto de ações que concatenadas formam uma atividade num fluxo de trabalho cuja descrição é delimitada pelo evento de início e o momento em que é encerrada.	Elicitação das transações, o conjunto de ações que concatenadas efetivam no sistema uma operação, um fato ocorrido no mundo real.
Gráficos de Redes de Petri ou gráficos baseados em atividade são utilizados para expressar a modelagem. Recentemente os diagramas de atividade da UML têm sido objeto de pesquisas para serem utilizados largamente como fundamento de execução e modelagem de workflow. Basicamente um só diagrama é utilizado.	Depois de muitos anos de técnicas baseadas em dados, como as técnicas relacionais, a orientação a objetos introduz maior poder de expressão, principalmente no tocante a modelagem das transformações dos dados pelas operações do sistema, modelando os procedimentos com a utilização dos casos de uso. Vários diagramas são utilizados para apresentar as diversas visões do sistema.
Processo de desenvolvimento dirigido por processos de negócio onde a modelagem do fluxo de atividades pretende representar a semântica a ser realizada pela aplicação.	Processo de desenvolvimento dirigido por casos de uso voltado para transformar a modelagem da semântica dos conceitos em componentes de código por meio de técnicas de derivação.
Proposta:	
Para aumentar o poder da modelagem de workflow, os blocos de construção devem permitir especificar os requisitos necessários em domínios de processos de produção. Outros diagramas poderiam auxiliar a apresentar as diversas visões da aplicação.	

Workflow	Sistemas de Informação
Atores	
<p>As atividades, unidades para a interação ator X aplicação, são assinaladas (“entregues”) aos atores, chamando-os a interagir. A finalização da atividade ocorre por um evento disparado pelo ator. O assinalamento de uma atividade é ativo.</p>	<p>Os atores têm de acessar as funcionalidades do sistema, unidades para a interação ator X transações, para registrarem operações ou consultar informações agregadas. A efetivação das operações ocorre por um evento disparado pelo ator. A funcionalidade é passiva quanto ao acesso por um ator.</p>
<p>Algumas propostas pretendem modelar tanto as habilidades quanto as responsabilidades para escolher um ator e assinalar uma atividade, decidindo o acesso. Obter acesso a uma atividade corresponde a obtenção de acesso aos dados manipulados na atividade somente para aquele caso.</p>	<p>A grande maioria das propostas restringe o acesso às funcionalidades da aplicação baseada em papéis e grupos. Obter acesso a uma funcionalidade corresponde a obtenção de acesso a todos os dados da mesma classe daqueles manipulados pela funcionalidade.</p>
<p>Proposta: Para aumentar a adequabilidade pode-se abordar o acesso dos atores às atividades em um processo como têm sido tratados os usuários no acesso aos sistemas de informação.</p>	
Dados	
<p>Dizem respeito ao caso em execução e são armazenados com ele. Utilizar XML para que os dados sejam armazenados em agregações lógicas.</p>	<p>Dizem respeito às abstrações dos conceitos encontrados no domínio do sistema e são armazenados em estruturas tabulares de acordo com seu tipo.</p>
<p>Os dados são informações dos processos digitadas pelos atores ou provenientes de sistemas de informação ou ainda, arquivos de imagem, de texto ou outros, anexados pelos atores. Os dados são pouco modificados, mantendo as características originais por todo o ciclo do processo.</p>	<p>Os dados registram fatos reais digitados pelos atores no sistema e são transformados pelas regras do negócio que o sistema apoia ao longo do ciclo de operações ocorridas.</p>
<p>A integração com informações em sistemas legados tem permitido a invasão de domínios, onde uma aplicação acessa diretamente bases de dados de uma aplicação externa, acoplado a necessidade do conhecimento da estrutura dos dados da aplicação externa.</p>	<p>A integração com dados externos ocorre ao nível de aplicação preservando os domínios.</p>
<p>Dizem respeito ao caso em execução e são armazenados com ele. Utilizar XML para que os dados sejam armazenados em agregações lógicas.</p>	<p>Dizem respeito às abstrações dos conceitos encontrados no domínio do sistema e são armazenados em estruturas tabulares de acordo com seu tipo.</p>

Workflow	Sistemas de Informação
Dados	
<p>Os dados são informações dos processos digitadas pelos atores ou provenientes de sistemas de informação ou ainda, arquivos de imagem, de texto ou outros, anexados pelos atores.</p> <p>Os dados são pouco modificados, mantendo as características originais por todo o ciclo do processo.</p>	<p>Os dados registram fatos reais digitados pelos atores no sistema e são transformados pelas regras do negócio que o sistema apoia ao longo do ciclo de operações ocorridas.</p>
<p>A integração com informações em sistemas legados tem permitido a invasão de domínios, onde uma aplicação acessa diretamente bases de dados de uma aplicação externa, acoplando a necessidade do conhecimento da estrutura dos dados da aplicação externa.</p>	<p>A integração com dados externos ocorre ao nível de aplicação preservando os domínios.</p>
<p>Proposta: Cada atividade de um processo deve encapsular o conjunto de dados do qual necessita. A integração com dados externos ao processo tem de ocorrer por mecanismos de interfaces preservando domínios. A abstração dos dados deve refletir um conceito real, como documentos modelados no fluxo do processo.</p>	
Arquitetura	
<p>Geralmente têm sido utilizadas ferramentas fechadas que fornecem ambiente próprio para desenvolver a aplicação numa arquitetura monolítica.</p>	<p>Geralmente é formada por um framework de classes e serviços baseados em padrões que permitem que o foco esteja somente nas responsabilidades específicas do sistema sendo construído.</p>
<p>Diversos paradigmas têm sido utilizados como base para a construção dos WMS's e muitas vezes é impossível representar uma aplicação num modo que reflita o que ocorrerá quando implementada.</p>	<p>O modelo lógico reflete a arquitetura física e o paradigma OO perpassa o processo todo, em todos os níveis de abstração, até as peças físicas de código.</p>
<p>O software é executado em ambientes fechados e proprietários e sua definição reflete as capacidades específicas do motor de workflow utilizado.</p>	<p>O software é uma coleção de componentes ou peças que colaboram e interagem para responder às requisições dos atores buscando tornar cada pequena parte o mais desacoplada possível.</p>
<p>Proposta: Mais importante que os modelos representacionais são os blocos de construção implementáveis baseados em padrões que mapeiam os requisitos de aplicações de workflow Para aumentar a adequação para uso em ambientes de engenharia de software, deve ser baseada em <i>frameworks</i> de componentes e de serviços no ambiente de desenvolvimento. Podem ser utilizadas APIs como a estudada pela especificação JSR 207, do JCP, para J2EE.</p>	

O propósito da comparação é analisar as diferenças para sublinhar como ultrapassá-las pois é necessário que sejam apontadas as similaridades com sistemas de informação para que aplicações de workflow utilizem esta *expertise* acumulada e sejam igualmente elaboradas num processo claro de engenharia.

3 REQUISITOS EM UMA APLICAÇÃO DE WORKFLOW NA PMPA

Nos últimos anos a necessidade em atender a demanda por aplicações de workflow tem crescido rapidamente. Por um lado esta é uma evolução natural em ambientes com plataformas de sistemas maduras e bastante disseminadas, por outro lado, processos de redesenho administrativo realizados nas organizações também têm contribuído para que esta demanda aumente.

Na administração de Porto Alegre, as demandas individuais dos cidadãos geram, dadas as características operacionais da estrutura, o registro em papel que passa a constituir o que denomina-se processo administrativo ou expediente, ou seja, um volume de papéis nos quais se registram todos os pareceres, tramitações, providências, anexação de documentos e decisões de todas as partes da estrutura que atuam operacionalmente no atendimento ou necessitam tomar conhecimento de seu teor [...] (PREFEITURA MUNICIPAL DE PORTO ALEGRE, 1999).

O processo administrativo tem seu volume físico tramitado manualmente por todas estas partes da estrutura municipal, gerando as atividades de malotes, estafetas, transporte, protocolos internos setoriais, leituras e releituras, pedidos de informação, memorandos e setores de arquivos, com excessiva ênfase no registro da atuação da estrutura [...] (PREFEITURA MUNICIPAL DE PORTO ALEGRE, 1999).

Desde a década de 80 a administração pública pode contar com um sistema de apoio responsável por informar o estado de uma solicitação, um processo tramitando em qualquer órgão da administração. O sistema GPA, Gerenciamento de Processos Administrativos, foi elaborado com o objetivo de controlar a situação e localização de um processo.

Uma nova instância de processo aberta fazia com que uma pasta em papel fosse criada. Os documentos necessários ao processo trafegariam dentro desta pasta. A informação sobre a abertura, a situação inicial e a localização do processo seria inserida no GPA, que geraria um número identificador para o processo, fixado na capa da pasta. A cada tramitação ("transição") do processo, o GPA seria atualizado com a nova situação e localização. Em suma, o GPA informa o estado e a situação do processo, se cada ator deste processo zelar pela sua atualização, mas não é capaz de garantir a tramitação do processo, de gerir suas informações, de enviar o

processo ao setor ou funcionário correto, enfim, de executar requisitos típicos de aplicações de workflow.

Algumas funcionalidades do GPA são:

- a) Acompanhar a tramitação dos processos administrativos dentro da estrutura organizacional;
- b) Cadastramento dos processos com emissão de etiquetas e guias de andamento;
- c) Andamento dos processos efetuados por qualquer órgão da estrutura, desde que autorizado;
- d) Troca de situação do processo;
- e) Anexação e desanexação de processos (agregação);
- f) Diferentes formas de pesquisas aos processos por requerente, número e endereço; do trâmite do processo;
- g) Estatísticas sobre o tempo médio de permanência dos processos por órgão e assunto.

Com base nas informações do GPA, constatou-se uma tendência de crescimento na quantidade de expedientes dentro deste fluxo operacional que é predominante na administração pública em todos os níveis [...] (PREFEITURA MUNICIPAL DE PORTO ALEGRE, 1999). Esta tendência aliada ao grau crescente de dificuldade para obter melhorias dentro do método operacional tradicional, apontou para a necessidade de mudança do modelo organizacional e suas peculiaridades. Este modelo organizacional está apoiado em sistemas de informação que são produto de suas práticas operacionais, não apoiando melhorias significativas na forma de atendimento.

Neste contexto foi apontada a busca de otimização de processos organizacionais, procurando empregar ferramentas que introduzam maior produtividade, para alcançar maior eficiência pela mudança dos métodos de trabalho e pelo emprego de tecnologia adequada.

O Programa de Desenvolvimento Municipal, financiado pelo BIRD, pretende exatamente introduzir melhorias na administração pública através de projetos para otimizar os processos de trabalho e qualificar o sistema de gerenciamento. Como parte deste programa, o Projeto de Modernização Administrativa projetou o Módulo Operacional [...] (PREFEITURA MUNICIPAL DE PORTO ALEGRE, 1999) que objetiva, através do redesenho de processos, otimizar o atendimento das demandas externas da população, internas da estrutura administrativa e gerenciais dos gestores do município.

3.1 Redesenho Administrativo na Prefeitura de Porto Alegre

Juntamente com técnicos da Prefeitura Municipal de Porto Alegre, os técnicos da Procempa vem trabalhando em projetos de redesenho de processos administrativos nos últimos 3 anos, como parte do esforço da administração pública da cidade para modernizar e racionalizar seus serviços e rotinas.

Através de concorrência pública internacional [...] (PREFEITURA MUNICIPAL DE PORTO ALEGRE, 1999), foi contratada uma consultoria para agilizar e qualificar o atendimento às demandas, tanto externa, dos munícipes, quanto internas, da estrutura da administração, através da aplicação de uma metodologia de redesenho de processos. Como produto deste trabalho estavam previstos:

- a) Elaboração de um plano de gerência com um planejamento global contemplando todas as etapas do trabalho;
- b) Identificação e priorização dos processos para redesenho nas áreas de administração de recursos humanos, parcelamento do solo, atendimento nos postos de saúde, manutenção da cidade, concessão de alvarás, trânsito, habitação e ensino;
- c) Capacitação e transferência de tecnologia na metodologia para redesenho de processos organizacionais através de treinamentos, seminários, etc;
- d) Redesenho dos cinco processos selecionados, com a utilização da metodologia, em conjunto com os técnicos designados pela administração pública e os consultores e técnicos contratados contemplando 5 fases:
 - d.1) Levantamento da situação atual com o mapeamento gráfico da situação em que o processo se encontra;
 - d.2) Análise crítica da situação atual com identificação de inconsistências, tempos e passos desnecessários, retrabalhos, sobreposições, etc;
 - d.3) Proposição de uma nova situação evidenciando as alternativas contemplando a seleção de uma delas para aplicação do novo fluxo com uma respectiva estrutura de gerenciamento para aferição de eficácia e eficiência;
 - d.4) Validação da situação proposta por meio de reuniões, seminários, debates e discussões;
 - d.5) Implementação dos processos redesenhados e dos controles previstos..

A metodologia utilizada no processo de redesenho (BRACHE, A.P.; RUMMLER;G.A., 1995) propunha a organização de uma comissão executiva, formada pela alta gerência, e uma comissão coordenadora, composta por gerentes e supervisores das funções envolvidas com a participação do patrocinador ou dono do processo, e equipes de aperfeiçoamento do processo com o propósito de realizar as etapas de trabalho em cada processo, com a presença de um líder de equipe e de um

facilitador, responsável pelo treinamento, orientação do uso da metodologia e documentação do trabalho da equipe.

Após a definição do projeto e da capacitação dos envolvidos, o trabalho se encaminhou para a análise do ambiente a nível de organização, com a análise do contexto organizacional, com confirmação das questões e dos processos críticos, e da projeção do novo sistema ao mesmo nível, onde são determinadas as necessidades e as mudanças organizacionais.

O principal produto desta etapa é o mapa de relacionamentos que mostra os relacionamentos básicos de entrada-saída no modelo cliente-fornecedor de uma organização (BRACHE, A.P.; RUMMLER;G.A., 1995). Eles foram utilizados para definir o escopo do projeto de aperfeiçoamento, para identificar as funções principais, identificar desconexões, analisar o impacto das mudanças, analisar e discutir a solução dos problemas encontrados, adequar o relacionamento interno e entender como a organização trabalha para gerar seus serviços ou produtos.

Na segunda etapa foi feita a análise do nível de processo, modelando os processos atuais e identificando suas desconexões e modelando os futuros processos, desenvolvendo controles específicos e denotando as mudanças desejadas. O principal produto desta etapa são os mapas de processo, que mostram a seqüência de eventos que convertem uma entrada específica numa saída desejada. O mapa de processo é utilizado para tornar visíveis as fases de um processo, demonstrando seu fluxo e anotando dados como tempos e desempenho (BRACHE, A.P.; RUMMLER;G.A., 1995).

A última etapa fez a análise e projeção dos processos ao nível de cargo, onde foram elicitados os cargos e suas responsabilidades críticas foram definidas as necessidades do cargo frente às atividades do processo redesenhado e as mudanças necessárias ao nível de recursos humanos. O principal documento nesta etapa é a matriz de papéis/organizacionais/responsabilidades que correlaciona cada atividade e subatividade do processo com as responsabilidades de cada papel (BRACHE, A.P.; RUMMLER;G.A., 1995).

O trabalho realizado redundou no redesenho de 5 macro-processos, apontados como os primeiros a sofrerem mudanças no escopo da modernização da administração pública de Porto Alegre, são eles:

- a) Atendimento geral de demandas rápidas e serviços de limpeza de equipamentos de drenagem, galerias e arroyos;
- b) Processo de inclusões, exclusões e alterações na folha de pagamento;
- c) Licenciamento de atividades localizadas e não-localizadas;
- d) Atendimento ambulatorial básico e especializado;
- e) Desenvolvimento Urbanístico: estudo de viabilidade urbanística (EVU) e parcelamento do solo.

Alguns projetos estão em adiantado grau de mudança, e destes, dois estão em processo de implementação de aplicação de workflow para suporte de processos, no escopo do seu macro-processo, com suporte do WideFlow.

O processo de solicitação de gratificação e vantagens apoiará a SMA, Secretaria Municipal de Administração, a fazer tramitar as solicitações, provenientes dos diversos órgãos da administração, de concessão dos benefícios monetários legais a serem agregados à remuneração do servidores municipais. Cada solicitação tem origem no órgão onde o servidor está lotado, percorrendo o caminho entre o órgão demandante e a SMA, onde é avaliado, concedido na forma legal prevista, publicado e lançado no sistema de folha de pagamento.

EXPEDIENTE UNICO - 002 999999 00 5 DATA INSTITUICAO - 10/11/1986

10/11/1986 LOCALIZACAO MAIS ANTIGA CADASTRADA
8278301 AV CARLOS GOMES , 999898

10/11/1986 CONST.IRREGULARIDADE RECAD.1986

19/10/1987 DECLARACAO MUNICIPAL EMITIDA MANUALMENTE
DEFERIDA
TIPO AREA ATINGIDA SIST.CIRCUL.
ORIGEM PARCELAMENT ORIGEM N/ PESQUISADA

19/01/1988 APRESENTA ESTUDO VIAB. EDIF. C/PERMUTA INDICE

07/06/1988 ESTUDO DE VIABILIDADE PARA ALTURA EDIFICACOES DEFERIDO

19/07/1989 ESTUDO DE VIABILIDADE PARA SUBS DE PILOTIS P/1 PAU.TIPO DEFERIDO

16/04/1990 PROJ.EDIFICACAO DE CONSTRUCAO NOVA DEFERIDO
AREA TERRENO - 712.60 M2 AREA OCUPADA - 476.70 M2
ALTURA(S)
BLOCO 1: 7 PAU- SEM TERRACO- SEM PILOTIS- DOIS SUBSOLOS
ATIV. - S E R V I C O S

	TOTAL A CONSTR.	EXISTENTE A DEMOLIR		
AREA COMPUTADA	1813,60	1813,60	00,00	00,00
AREA NAO COMPUTADA	940,00	940,00	00,00	00,00
AREA EXCLUIDA	1238,82	1238,82	00,00	00,00
AREA TOTAL PROJETO	3992,42	3992,42	00,00	00,00
ECONOMIAS	43	43	0	0
ECONOMIAS ZELADOR	1	1	0	0
DORMITORIOS ZELADOR	1	1	0	0
ESTACIONAM.COBERTOS	38	38	0	0
ESTACIONAM.SEMI-COBERTOS	9999	9999	9999	9999

TIPO CONSTRUCAO: PRED.ALVENARIA 7 PAU

19/07/1989 ESTUDO DE VIABILIDADE PARA SUBS DE COBERT P/1 PAU.TIPO DEFERIDO

22/06/1993 REQ. VISTORIA TOTAL EDIF.

Figura 3.1: Declaração Municipal (DM)

O projeto piloto é o processo de gratificação e vantagens para os servidores lotados na SMED, Secretaria Municipal da Educação, constituído na grande maioria por quase 4.000 professores da rede de ensino público municipal dispersos em mais de 87 escolas, 124 creches conveniadas e 117 turmas de alfabetização de adultos, segundo dados de 2002.

A mobilidade dos professores é intensa e cada lotação em uma escola é um ato administrativo do poder público que pode resultar em alteração no conjunto de benefícios. Um professor ao ser lotado em uma

escola considerada de difícil acesso, por exemplo, passa a ter direito a perceber uma gratificação.

O outro processo, para solicitação de DM, Declaração Municipal – figura 3.1, documento que informa as condições urbanísticas de ocupação do solo, pretende atender a população em geral, em especial aqueles que necessitam consultar o município para modificar a realidade urbanística da cidade por meio do uso do solo, seja pela construção ou modificação de edificações individuais ou condominiais, ou pelo parcelamento da divisão territorial privada.

Este processo se insere no macro-processo de Desenvolvimento Urbanístico: estudo de viabilidade urbanística (EVU) e parcelamento do solo, regulado por diversas legislações específicas como o PDDUA, Plano Diretor de Desenvolvimento Urbano Ambiental, Lei Complementar 434, e o Decreto 12.715/2000 do Município de Porto Alegre. Estes processos prevêm a solicitação da DM pelo cidadão através da Internet com o correspondente envio da documentação necessária, que deverá tramitar pelos diversos órgãos e secretarias municipais responsáveis pela avaliação das condições de uso do solo em sua área de competência específica - como por exemplo, esgotos e meio ambiente - para produzir o documento final contendo as prescrições, recomendações e proibições relativas à área territorial em análise.

As próximas seções discutem os requisitos do processo de solicitação de DM, o redesenho, a análise e modelagem do fluxo, a integração com dados e informações legadas e a estratégia de assinalamento de atores para as atividades do processo.

3.2 Requisitos na Automação do Processo da Declaração Municipal

O primeiro projeto de automação foi realizado num dos processos que compõem o macro-processo de Desenvolvimento Urbanístico. Ele propõe automatizar a aprovação e licenciamento de edificações, processo no qual os responsáveis técnicos, engenheiros e arquitetos, encaminham o conjunto de plantas e documentos, submetendo o projeto de edificação à aprovação.

Uma das características deste processo é o forte requisito de gestão documental e de ações *ad hoc* no trâmite do processo. Muitas são as situações possíveis, a legislação é complexa, com uma lógica intrínseca não estruturada que dificulta a normalização do processo não permitindo sua modelagem *a priori* – seção 2.1.2.

Uma ferramenta para aplicações de GED com algumas funcionalidades de workflow foi utilizada no suporte à automação deste processo. Apesar de atender muito bem aos requisitos do processo de aprovação e licenciamento de edificações, implementando-o, sua utilização

mostrou o quão improdutivo, do ponto de vista do desenvolvimento de aplicações, pode ser uma ferramenta de workflow – seção 2.1.

A elaboração de interfaces de apresentação para a interação dos atores foi necessária e sua customização estava restrita a uma linguagem, suas APIs tinham de ser acessadas por meio dela – seção 2.1.6. As exigências de infra-estrutura de hardware para os atores do processo e o custo de licenças de uso individuais onerou a implantação.

Como ferramenta GED ela não pretendia atender de forma precisa e flexível todos os cenários possíveis de automação. A proliferação de aplicações de workflow tinha de ser feita de forma controlada e planejada, mas como parte de um processo de desenvolvimento produtivo – seção 1.2. Além de tudo, o resultado do esforço ao utilizar a ferramenta existente era um produto escrito num ambiente totalmente proprietário – seção 1.2, o que contraria os objetivos da empresa em contar com soluções não proprietárias ou soluções que pudessem ser elaboradas com recursos do ambiente para desenvolvimento de software existente.

O processo de solicitação de DM, no âmbito da SPM, Secretaria do Planejamento Municipal, também um macro-processo do processo de Desenvolvimento Urbanístico, tomou um caminho diferente daquele utilizado na automação do processo de aprovação e licenciamento de edificações. Seus requisitos estratégicos e funcionais podiam ser atendidos por uma solução própria, criada exatamente para situações com estas mesmas características, discutidas anteriormente na seção 2.1:

- a) eventos de disparo e término de atividades reativos aos atores;
- b) passos repetitivos que podem ser modelados *a priori*;
- c) grande necessidade de dados legados de várias origens como base para as atividades e decisões do fluxo;
- d) forte necessidade de integração com diversos sistemas de informação no nível de programa - seção 2.2.5, disparando eventos para que estes realizem atividades em seus domínios;
- e) complexidade de interação da interface com os atores, demandando em customizar a apresentação das atividades – seção 2.1.4;
- f) aplicação corporativa de larga escala que deverá suportar solicitações que tramitarão por quase uma dezena de órgãos geograficamente dispersos pela cidade.

Estas características se ajustam às responsabilidades e funcionalidades do motor de workflow do WideFlow. A seguir os requisitos do processo de solicitação de DM, originados no trabalho de redesenho deste processo, serão brevemente discutidos.

Eles fazem parte do projeto SDM, Solicitação de Declaração Municipal, uma aplicação que irá utilizar o motor de workflow WideFlow, entregando-lhe a responsabilidade de controlar o estado e os dados das requisições, e da integração com sistemas e bases legadas no que diz

respeito aos dados do processo, e irá possuir persistência de dados própria para algumas informações como a identificação do requerente, ator fora do domínio da rede de computadores, não possuindo autenticação na mesma.

3.2.1 Redesenho do Processo de Solicitação de Declaração Municipal

A DM é solicitada no protocolo da SPM. Ela faz parte da primeira etapa a ser cumprida para o requerimento e licenciamento, junto à Prefeitura de Porto Alegre, de projetos de edificações ou atividades ou para projetos de parcelamento do solo de acordo com o Decreto Municipal 12.715/2000.

Ela informa o regime urbanístico, o traçado viário do Plano Diretor em vigor, a localização de equipamentos urbanos e comunitários, as restrições administrativas e áreas não edificáveis, os dados relativos ao alinhamento atual e projetado e todos os condicionantes legais a serem observados para execução do projeto no lote em questão. A definição de DM pode ser encontrada no Capítulo II, artigos de 7 a 9 do Decreto Municipal 12.715/2000.

O processo atual faz a tramitação manual de um expediente, um processo administrativo, contendo a documentação e uma planta de situação, através de diversos órgão e setores onde são feitas as consultas cartográficas e aos sistemas de informação, quando cada setor verifica e apõe condicionantes ou impedimentos, se for o caso. Este é o cenário a otimizar.

Do conjunto de inovações propostas é relevante destacar que a solicitação da DM será feita através de auto-atendimento na Internet, com a solução sendo encaminhada simultaneamente aos órgão envolvidos, permitindo a descentralização da solicitação. O delineamento do processo crítico é o seguinte:

Tabela 3.1: Delineamento do Processo Crítico de Solicitação de DM

Onde o processo começa?	Na SPM com o requerimento feito pelo cidadão
Onde o processo termina?	Com a emissão de uma DM ou de uma segunda via.
Quais as saídas do processo?	A DM.
Quem são os clientes do processo?	A população interessada ou representada por profissional habilitado.
Qual a questão estratégica?	Reduzir tempo, assegurar transparência, simplificar processo e otimizar o uso dos recursos envolvidos.
Quais são as principais deficiências do processo atual?	<ul style="list-style-type: none"> • Não cumprimento do prazo de 15 dias da emissão, estabelecido no artigo 8º do Decreto 12715/2000; • Atrasos e desvios na execução do fluxo do processo; • Substituição freqüente de atores do processo; • Documentação incompleta; • Demora na análise inicial da documentação;
O que precisa ser aperfeiçoado no processo para afetar a questão estratégica?	<ul style="list-style-type: none"> • Assegurar a execução correta do fluxo; • Incrementar a eficácia do processo (solicitações x DM emitidas);

	<ul style="list-style-type: none"> • Validar os documentos e dados de preenchimento da solicitação.
Quais os critérios para o novo processo?	<ul style="list-style-type: none"> • Privilegiar o atendimento ao requerente; • Estimular meios digitais na documentação apresentada; • Uso intensivo de workflow; • Uso intensivo de formulários eletrônicos.
Qual o nível de desempenho desejado?	DM emitida em: <ul style="list-style-type: none"> • 60% imediatamente; • 25% em 5 dias; • 10% em 7 dias; • 05% em 13 dias.

Com este projeto pretende-se resolver as seguintes rupturas:

- a) a relação de documentos e o requerimento padrão só está disponibilizada no protocolo setorial da SMOV, Secretaria Municipal de Obras e Viação, e no núcleo de expediente da SPM, Secretaria de Planejamento Municipal;
- b) o horário bancário diverge do horário de atendimento das secretarias;
- c) as consultas da análise são feitas deslocando o expediente único em mãos;
- d) mão de obra inadequada no ato de recebimento dos documentos no protocolo;
- e) quando da emissão da DM um técnico se desloca com o expediente em mãos gerando atrasos;
- f) o visto da documentação é fornecido num local diferente do protocolo.

Analisando as alternativas de viabilidade para automação deste processo a PROCEMPA e a SPM decidiram em conjunto por planejar o projeto de automação em dois patamares. O primeiro patamar automatiza o fluxo e os dados do processo com a planta de situação digitalizada e o integra com os diversos sistemas de informação que detêm dados que constarão dos processos ao tramitarem. O segundo patamar prevê a interação com sistemas georreferenciados, ainda em desenvolvimento e implantação, proporcionando maior automação das atividades que atualmente os atores desempenham perante o processo.

Assim a aplicação, no primeiro patamar, deverá contemplar os seguintes requisitos:

- a) Permitir solicitação de DM pela Internet para imóveis com até 1000m²;
- b) Identificar os requerentes, cadastrando-os pela Internet;
- c) Possibilitar o envio de arquivos como a planta de situação e o título de propriedade digitalizados, via Internet;
- d) Emitir e controlar o pagamento do DAM, Documento de Arrecadação Municipal, para andamento do processo ou excluir a solicitação após 15 dias sem pagamento;
- e) Interação com sistema de informação responsável por gerar a DM;

- f) Enviar a DM gerada ao requerente via e-mail, em formato PDF;
- g) Enviar títulos de propriedades de imóveis para a SMF, Secretaria Municipal da Fazenda;
- h) Disponibilizar a DM e os documentos do processo para consultas internas;
- i) Indeferimentos por falta de complemento dos dados, por parte do requerente, após 90 dias de espera
- j) Acessar dados dos sistemas, com possibilidade de modificar alguns destes dados:
 - j.1) PED, Aprovação de Projetos de Edificação – manipula o cadastro de requerentes;
 - j.2) GPA, Gerência de Processos Administrativos – mostra a localização de um processo ou expediente para toda a estrutura municipal;
 - j.3) CEU, Cadastro de Expedientes Únicos – cadastro atualizado de Expedientes Únicos de edificação e parcelamento do solo em diversas etapas de seu processo;
 - j.4) CTM, Cadastro Técnico Municipal - manutenção das informações urbanas, vinculado a referências cartográficas e informações sobre UTP ou Área Especial, Quarteirão, Faces, Imóveis, Declarações Municipais, etc;
 - j.5) IMB, Cadastro Imobiliário - cadastro de economias do imobiliário e o lançamento de IPTU, Taxa de lixo e outros tributos imobiliários;
 - j.6) CDL, Cadastro de Logradouros - centraliza as informações sobre denominação e localização de logradouros, histórico de cada logradouro, legislação, etc;
 - j.7) CAM, Controle de Arrecadação Municipal – lançar, controlar e efetivar os tributos e todas as receitas do Município.

3.2.2 Modelagem do Workflow

A modelagem do fluxo por meio do Diagrama de Atividades é apresentada na figura 3.2. É apresentado um modelo de alto nível, isto é, que não possui toda a semântica que o motor de workflow necessita, mas com a visão do usuário, dos participantes do workflow. Por isto o rigorismo semântico é menor e os detalhes são anotados no estilo da especificação de casos de uso (ARAUJO; GOMIDE; PETRILLO, 2003).

Para iniciar o processo o requerente acessa a aplicação no seu navegador e informa a inscrição do imóvel no IPTU, o que identifica uma economia.

A aplicação deve verificar todos os endereços que compõem o bloco de endereços ao qual o imóvel pertence. Após ela deve selecionar o Expediente Único que engloba o conjunto de endereços selecionados anteriormente.

Se existir uma DM válida, com data menor que 1 ano, ela deve ser apresentada para que o requerente possa optar pela reemissão de uma 2ª via, caso em que não será necessário enviar os arquivos como descrito abaixo.

O requerente deve fazer o *upload* dos arquivos contendo a planta de situação e o título de propriedade, digitalizados para que sejam armazenados e tramitem com o processo.

Após ele emite o documento para pagamento da taxa relativa à expedição da DM, o DAM – Documento de Arrecadação Municipal, e a solicitação está completa.

A segunda atividade prevê que o requerente efetue o pagamento do DAM e após o aviso do crédito bancário o processo tem seguimento.

Se o requerente tiver optado por receber uma 2ª via de DM, como descrito na atividade de Solicitação da DM, ela é enviada e o processo termina.

Caso contrário, a solicitação é encaminhada para a atividade de Conferência da Solicitação onde o conferente verifica todos os dados e pode alterar o conjunto de endereços e/ou o Expediente Único da solicitação. Se o conferente perceber a falta de informações ou de arquivos ele envia um e-mail ao requerente que deve encaminhar as informações faltantes enquanto a solicitação permanece aguardando a complementação de informações.

Quando o conferente receber a resposta do requerente com a complementação ele as anexa à solicitação e encaminha para a atividade de Análise da Documentação.

É nesta atividade que os arquivos com planta de situação e título de propriedade são analisados em conjunto com as outras informações. Desta atividade pode resultar a impropriedade da solicitação, quando o analista pode pedir por e-mail que o requerente complemente ou altere as informações técnicas, fazendo com que a solicitação permaneça aguardando a resposta do requerente.

Quando as informações são complementadas o processo segue para a atividade de Análise de Projetos e paralelamente para a atividade de Análise de Infra-Estrutura.

Além de analisar sob perspectiva própria os detalhes da solicitação, os participantes nestas duas atividades, técnicos especializados, decidem os próximos passos da solicitação.

Ela pode ser enviada para que outros técnicos em setores, órgãos ou secretarias diversas, analisem e aponham informações, condicionantes ou vetos às possibilidades de uso da área geográfica em exame.

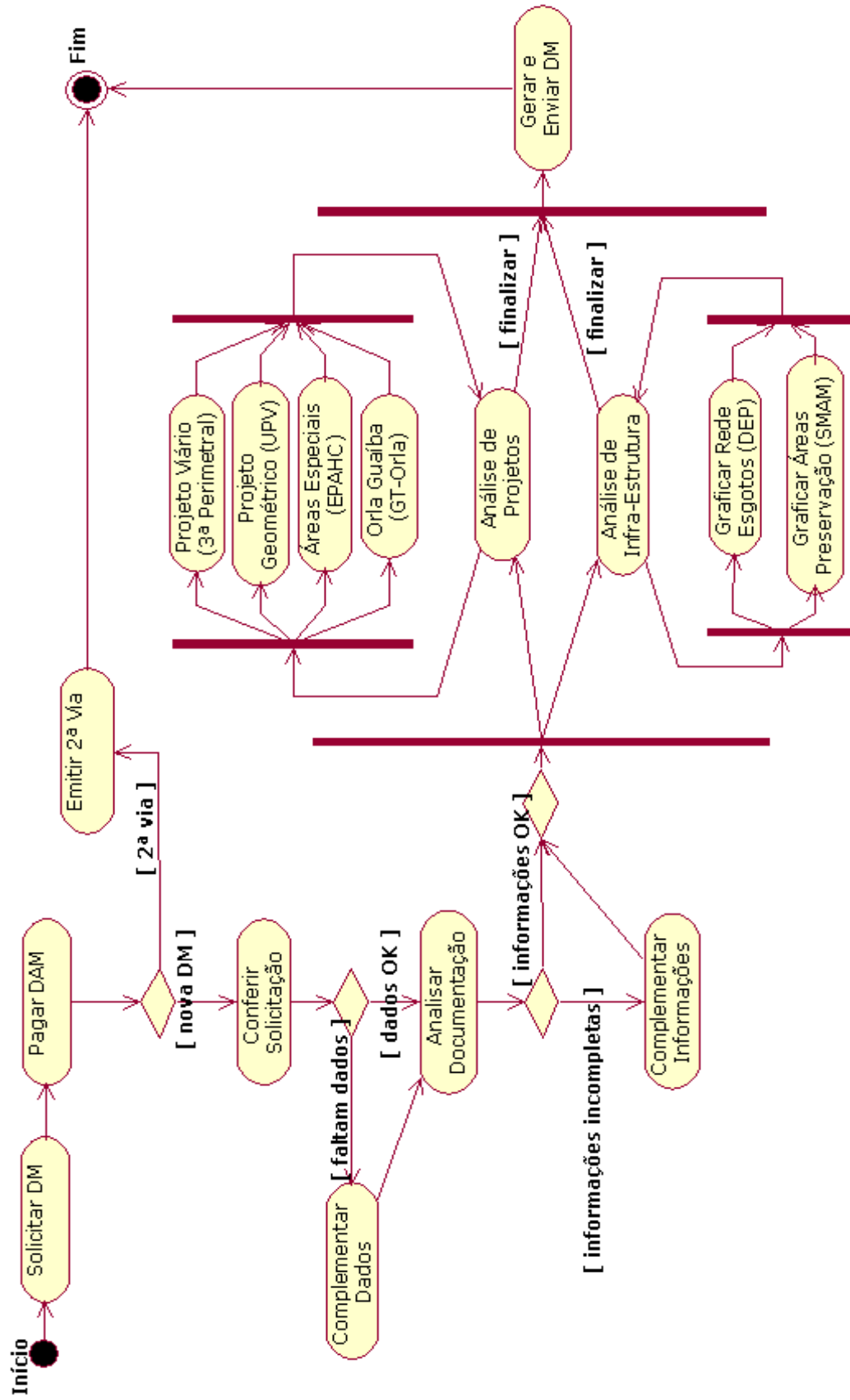


Figura 3.2: Fluxo do processo de solicitação de DM

Após esta circulação o processo deve retornar às atividades de origem – Análise de Projetos e Análise de Infra-Estrutura - quando os analistas podem decidir pelo reenvio para algum daqueles setores ou órgãos para nova análise ou podem decidir pela finalização de seu trabalho.

Para que o processo realmente possa ser finalizado, tanto o técnico responsável pela atividade de Análise de Projetos quanto o técnico responsável pela atividade de Análise de Infra-Estrutura têm de considerar o processo finalizado.

Se o requerente não pagar o DAM em 72 horas a solicitação é encerrada sem avisos. Igualmente se ele não responder aos pedidos de complementação de informações em 15 dias, no caso do pedido feito pelo conferente, ou não responder em 90 dias, no caso do pedido ser feito pelo analista da documentação.

Quando isto ocorrer a DM deve ser gerada e enviada ao requerente e também ser armazenada para consultas ou emissões posteriores.

3.2.3 Atores

A abordagem dada aos participantes ou atores também merece destaque. Na maioria das propostas eles são agregados de acordo com estruturas organizacionais e funcionais, o que nem sempre facilita a gestão do acesso às atividades e processos. Uma abordagem diferente se faz necessária, pois a administração pública utiliza descrições de cargos genéricos, isto é, dois funcionários com o mesmo cargo podem desempenhar funções muito diversas, mesmo que no mesmo órgão ou secretaria. Assim, o acesso às atividades e processos feito através de grupo de papéis, com permissão de acesso definida, como feito na maioria dos sistemas de informação, permite uma boa flexibilidade na gerencia de acesso não acoplada ao trabalho de remodelagem da estrutura organizacional, permitindo que esta responsabilidade seja repassada ao dono do processo.

A grande mobilidade encontrada entre os atores deste processo, apontadas no redesenho, insere o requisito flexibilidade na inclusão e exclusão das permissões de atores no acesso às atividades do processo.

Ao aproximar as abordagens encontradas na atribuição de permissões em sistemas de informação à atribuição de permissões para realizar atividades em workflow, passamos a utilizar o conceito de grupos de atores. Um grupo é um conjunto de atores que possui permissão para execução de uma atividade.

A análise do assinalamento das atividades pode determinar algumas estratégias. Uma atividade pode ter sido modelada para ser assinalada ao iniciador do processo. Neste caso o ator que criou a instância do processo possui acesso a atividade por definição, não sendo necessário mais nenhuma informação ao motor de workflow.

Outra possibilidade é a atividade **X** ter sido modelada para ser assinalada para o executor da atividade **Y**. Também neste caso o modelador definiu que o ator executor da atividade **X** possui permissão para executar a atividade **Y** sem necessidade de outras averiguações quanto às estratégias de assinalamento.

No caso de atividades que podem ser assinaladas para qualquer membro de uma equipe de trabalho, ou seja, membros de uma equipe de trabalho que possuem competência para realizar uma atividade, são utilizados grupos delimitando a equipe.

Os atores do processo são os funcionários públicos dos diversos órgãos e setores envolvidos. Cada atividade modelada terá um grupo correspondente. No grupo serão cadastrados os funcionários que possuem a responsabilidade de executar a atividade correspondente.

Quanto ao requerente da solicitação, ele é um ator externo, um *Business Actor* (JACOBSON; ERICSON; JACOBSON, 1994). Ele se cadastra pela Internet e recebe uma senha, ligada ao seu CREA/CPF. O mecanismo de workflow irá controlar processos e seu fluxo, cabendo à aplicação SDM controlar requerentes, pois estes são atores especiais do processo, os demandantes de processos, não executando atividades para que o processo se realize.

3.2.4 Dados e Informações

O requerente será cadastrado numa base de dados própria do SDM. Seus dados de identificação são dados cadastrais típicos de sistemas de informação como CNPJ ou CPF, CREA, nome, e-mail e telefone.

O requerente informa as inscrições de IPTU, Imposto sobre Propriedade Territorial Urbana, das quais deseja obter a DM.

Outros dados e informações da solicitação neste processo são provenientes de diversos sistemas de informação, como informações sobre os logradouros, sobre outras inscrições de imóveis que venham a fazer parte de Expedientes Únicos preexistentes, dados para emissão do DAM, Documento de Arrecadação Municipal, atualização do Expediente Único, atualização do sistema GPA, Gerenciamento de Processos Administrativos, que permanece como uma referência para toda administração pública, em suma, a interação com todos os sistemas apontados em 3.2.1.

Atualizar alguns destes sistemas também é necessário, fazendo com que as informações geradas ou transformadas pela tramitação do processo surta efeitos nos sistemas de informação, que continuam suportando e apoiando outras atividades em seus domínios.

Ao final do processo, a geração das informações da nova DM continua a cargo do sistema de informação, cabendo ao workflow disparar um evento para que este sistema crie a nova DM. Recebendo a DM como retorno o

workflow se responsabiliza por criar e enviar o arquivo com *layout* e formato definidos.

PREFEITURA MUNICIPAL DE PORTO ALEGRE

PLANTA DE SITUAÇÃO

✳

REQUERENTE _____

ESTA PLANTA NÃO TEM VALOR PARA FINS DE REGISTRO

A PRESENTE PLANTA DE SITUAÇÃO É PARTE INTEGRANTE
DA "DM" EMITIDA EM ____/____/____.

Figura 3.3: Planta de Situação

A planta de situação, figura 3.3, é um arquivo gráfico anexado que tramita entre os vários órgãos. Ela será manipulada em diversas atividades por órgãos e setores diferentes. Em algumas situações é possível que exista concorrência sobre o arquivo, com um setor executando sua atividade concorrentemente a outro setor em outra atividade. É necessário o controle desta concorrência pois ao final todas as informações devem ter sido apostas no mesmo arquivo físico.

3.2.5 Arquitetura

Um ponto onde a flexibilidade é necessária é na customização da *interface*, isto é, a customização da apresentação das tarefas e documentos eletrônicos ao participante. Com tradição de atender transações complexas, os sistemas de informação, principalmente em arquitetura cliente-servidor, mas mesmo os com arquitetura *n*-camadas com *interface stateless*, possuem *interfaces* de apresentação ricas em informação intergradadas. Atender este requisito, resolvendo as questões inerentes à modelagem de documentos (como suporte aos dados do processo) é muito difícil. Assim, a liberdade e facilidade da customização de interfaces é um requisito importante para a aceitação e difusão de processos automatizados.

O processo de solicitação de DM possui interação complexa com seus atores. A falta de uma base de dados georreferenciados e as particularidades da DM não permitem que a aplicação automatize muitas das ações dos atores na execução da atividade. Um exemplo disto é a necessidade de verificação se a solicitação da DM não coincide com um expediente único existente. Um expediente único, artigo 8º do Decreto 12.715/2000, reúne a história de um território, a evolução e as alterações num determinado lote de terra. Fazer esta verificação depende de intervenção humana pois o imóvel desejado pode constar de mais de um expediente único. Um expediente único reúne um conjunto de imóveis, um exemplo simples é um sub-bloco de edifício construído dentro de um condomínio horizontal, mas a evolução histórico-geográfica de um lote pode ser muito mais complexa que este exemplo.

Com as estruturas de dados atuais não há como um sistema decidir qual expediente utilizar para a DM em questão. A solução é sugerir a série de possibilidades para que um funcionário com habilidades para fazer esta análise tome a decisão.

De forma análoga, a planta de situação receberá novas informações gráficas a cada análise nos diversos órgãos responsáveis por uma parte do controle urbanístico do projeto de edificação pelos quais o processo irá tramitar. Neste patamar de automação não pretendemos atender às necessidades de manipulação gráfica de cada funcionário em cada um destes órgãos, mas entregar a planta e garantir o formato padrão definido para este arquivo na tramitação do processo.

Destas considerações depreendemos que além de forte integração com dados e sistemas legados existe a necessidade de criar interfaces mais ergonômicas que possam realmente auxiliar os atores em suas atividades como discutido anteriormente na seção 2.2.

4 O WIDEFLOW NA AUTOMAÇÃO DE PROCESSOS DA PMPA

O WideFlow é um motor de workflow criado para ser utilizado no cenário descrito nas seções 1.3 e 3 para suportar um tipo bem específico de automação: processos de produção (NICOLAO, 1996) tipicamente encontrados em administrações públicas, onde uma pasta contendo documentos trafega entre várias pessoas num fluxo previamente conhecido e com poucas variações, sendo alvo de verificações de conformidade, autorizações e despachos, para ao final, produzirem um documento com o resultado do processo.

Sua arquitetura está altamente acoplada ao ambiente de desenvolvimento de sistemas existente (ARAUJO; GOMIDE; PETRILLO, 2003), tipificado na seção 2.2, no qual é preponderante atender aos seguintes requisitos:

- a) tratamento dos atores do processo pode ser aproximado ao tratamento dado aos atores dos sistemas de informação: detentores de permissões para acessar determinadas funcionalidades – menus e interfaces de sistemas de informação ou atividades de processos em execução.
- b) integração com dados de sistemas de informação existentes naquele cenário, com uma clara divisão dos domínios da aplicação de workflow e dos sistemas, fazendo com que os dados legados sejam introduzidos nos documentos do workflow e não manipulados diretamente pelo motor de workflow nas bases de dados dos sistemas.
- c) atividades definidas no processo busquem em aplicações externas (no ambiente) a ocorrência de eventos que disparam trocas de estado do caso.

Toda a arquitetura lógica e física do motor de workflow se beneficia da existência de um *framework* comum (ARAUJO; GOMIDE; PETRILLO, 2003), de padrões e práticas utilizadas na elaboração de sistemas de informação ao longo de anos – seções 2.1 e 2.2.

O WideFlow se baseia no Diagrama de Atividades da UML [...] (OBJECT MANAGEMENT GROUP, 2003), uma forma especializada de diagrama de estados, do pressuposto da reatividade de um WMS (ESHUI; WIERINGA, 2001) e do encapsulamento dos dados do caso. O motor de workflow realiza uma máquina de estados capaz de executar as atividades,

as transições simples ou concorrentes ou ainda, paralelas e os objetos manipulados. Nele as atividades representam a execução de uma ou mais ações, um conjunto de tarefas discretas, com um evento implícito na sua terminação para ativar uma transição de saída (BOOCH; RUMBAUGH; JACOBSON, 2000).

Estas particularidades mostram-se muito apropriadas para sua utilização na automação de processos de produção como discutido na seção 1.4. Algumas extensões e pressupostos são adotados para aumentar o poder de expressão e o formalismo dos diagramas de atividade para definição e suporte de um motor de workflow.

4.1 O motor de workflow WideFlow

4.1.1 Controle do Fluxo

O WideFlow possui um meta-modelo, figuras 4.1 e 4.2, onde uma atividade é especializada em atividades de interação com um ator ou com execução temporal, uma **WorkTask**, atividades que tratam das condições de guarda com ou sem ramificação do fluxo, um **ForkDefinition**, e atividades responsáveis pela união de ramificações, um **JoinDefinition**.

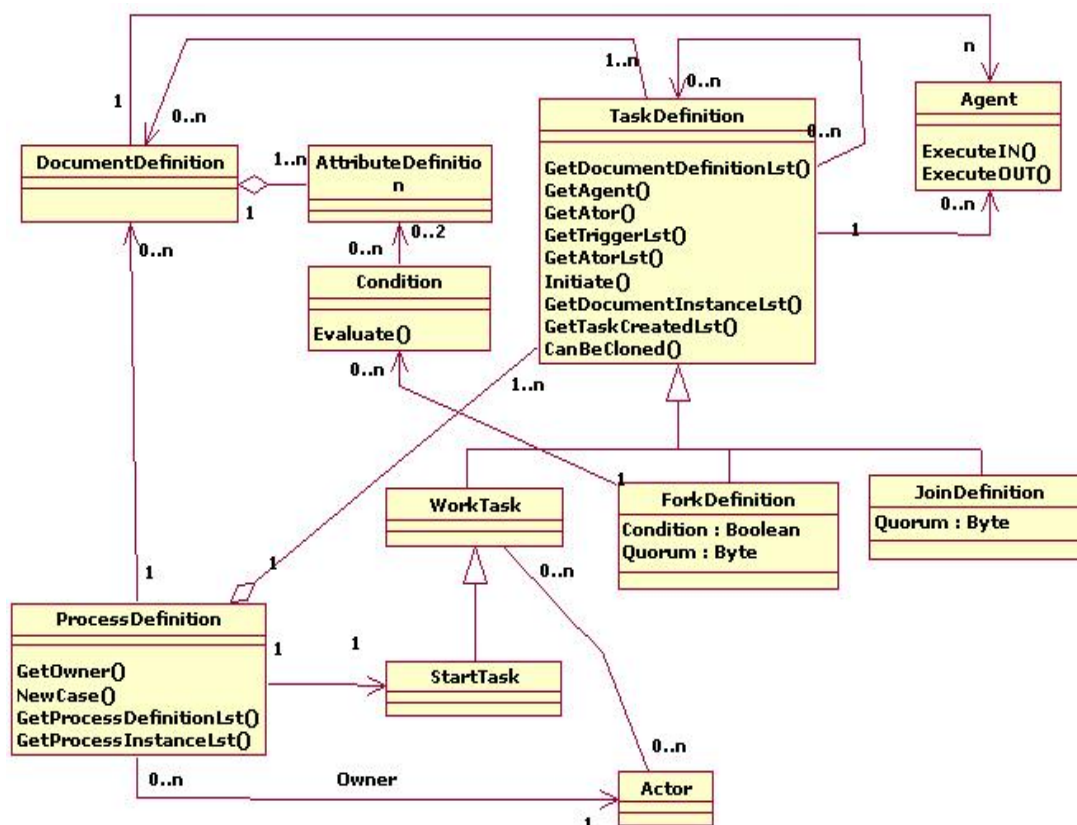


Figura 4.1: Meta-modelo de classes de definição do WideFlow

Seus métodos de inicialização e execução implementam responsabilidades distintas. Uma **WorkTask**, ao ser assinalada para um

ator, permanecerá aguardando pela interação que altere seu estado para “terminada”. Um **Fork** avalia sua condição específica que, se verdadeira, faz com que o próprio **Fork** altere seu estado para “terminada”. Um **Join** avalia se o número de atividades antecessoras terminadas corresponde ao *quorum* esperado, quando também altera seu estado para “terminada”. Uma *WorkTask* sem assinalamento para um ator, assim como uma *Temporal WorkTask*, é executada automaticamente pelo motor de workflow, que é o responsável pela alteração do estado da atividade, figura 4.3.

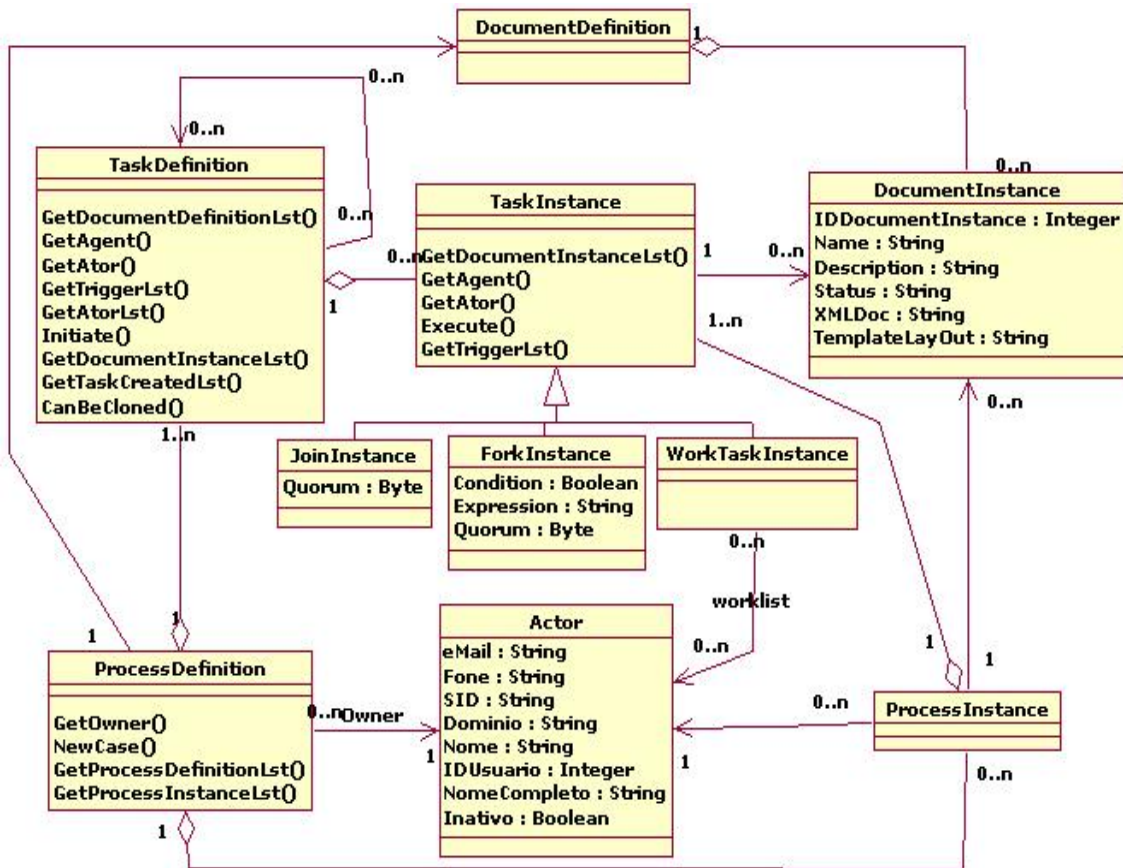


Figura 4.2: Meta-modelo de classes de execução do WideFlow

As atividades encapsulam os dados do processo em forma de documentos, abstração lógica para um conjunto de atributos de dados. Os dados são propagados entre atividades de acordo com a definição como apresenta a figura 4.4.

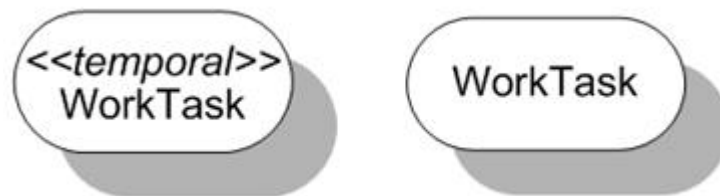


Figura 4.3: WorkTask temporal e com interação com ator no WideFlow

O WideFlow incorpora o conceito *template* → caso, discutido na seção 2.1.6, implementado do ponto de vista da orientação a objetos. Nele

uma classe é utilizada como o *template* para a criação de um objeto de outra classe, uma classe instância.

O meta-modelo estrutural do Wideflow mostra a existência de um conjunto de abstrações que se preocupam em conter a definição do processo, as tarefas e seu seqüenciamento, as informações manipuladas por cada uma e os participantes que podem interagir com elas. Na outra extremidade vemos as instâncias dos processos, os objetos que concretizam um processo particular de acordo com sua definição (BASTOS; RUIZ, 2001). Assim, os objetos das classes de definição são os responsáveis por inicializar as instâncias de objetos das classes de execução como na figura 4.5.

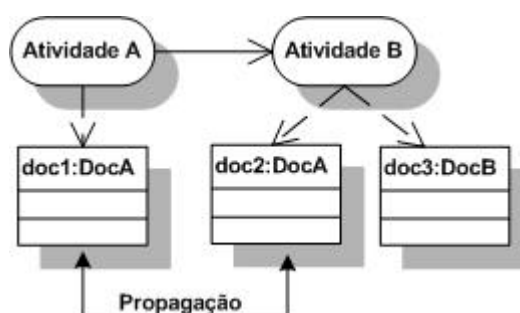


Figura 4.4: Propagação de documentos entre atividades

Existem dois eventos principais no ciclo de vida de uma instância de atividade: o evento de inicialização da atividade, que ocorre quando a atividade anterior termina e o evento de execução da atividade, quando o objetivo da tarefa está realizado e o motor de workflow passa seu estado para terminada.

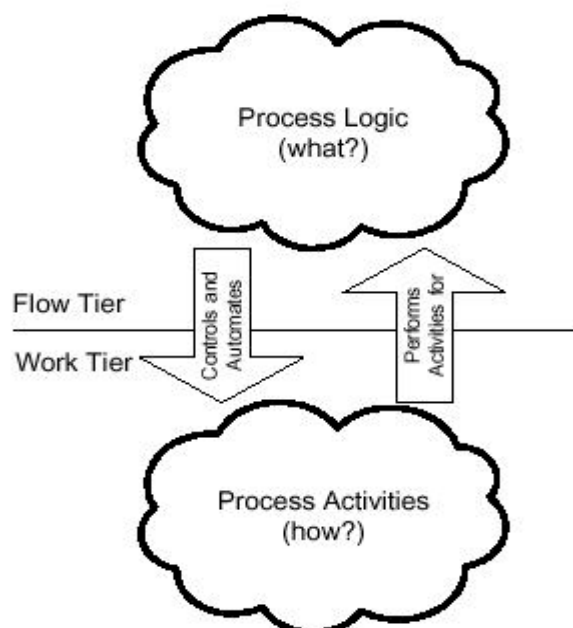


Figura 4.5: Lógica do Processo e Lógica da Execução (MANOLESCU, 2001)

Ao iniciar uma nova instância de processo, o WideFlow não cria todos os objetos definidos para o processo, como atividades, documentos, etc. Ao contrário, o WideFlow interpreta a definição e inicializa a atividade somente com os objetos necessários de acordo a definição. Assim, ao término da atividade de origem o WideFlow cria uma instância de objeto para a atividade de destino, assinala atores, ou não, para a instancia de acordo com a definição, cria também as instâncias de objeto com os documentos que a atividade irá manipular, propagando os dados nos documentos utilizados em atividades anteriores de tal modo que cada instância de atividade tenha seu próprio *snapshot*, preservando os dados da atividade de origem. Cada atividade no caso possui uma instância própria que ao ser finalizada passa para o estado “terminada”.

Um processo definido pode ter vários casos em execução e cada caso terá várias instâncias de atividade, algumas vezes, mais de uma instância para o mesmo tipo de atividade modelada pois o término de uma atividade resulta na sua passagem para um estado não mais relevante para a execução e uma posterior transição para esta mesma atividade, se previsto no fluxo, faz com que seja criada e executada uma nova instância de atividade e não o retorno da instância anterior a um estado válido para execução. Assim a própria seqüência de execução documenta todos os passos do processo.

Isto porque o fluxo é interpretado a cada passo, com a criação apenas dos objetos definidos para o passo, em oposição à instanciação de todos os objetos previstos para o processo inteiro já na sua criação, adotando a estratégia discutida na seção 2.1.6 a partir de pressupostos de (MANOLESCU, 2001).

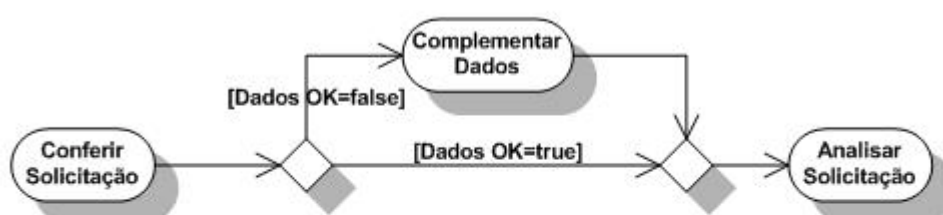
Quando no motor de workflow WideFlow uma atividade assume o estado “terminada”, o controle passa para a atividade seguinte, se não existir uma condição de guarda [...] (OBJECT MANAGEMENT GROUP, 2003) a ser satisfeita. Nos diagramas de atividade as transições são executadas imediatamente após a saída da atividade de origem, iniciando a atividade de destino sem passagem por estágios intermediários. A seqüência é contínua, de atividade em atividade.

Para controlar o fluxo do processo, expressões semânticas como condições de guarda podem ser impostas sobre transições entre a origem e o destino das atividades ou sobre cada transição existente entre uma divisão do fluxo e um destino, prevenindo concorrência ou paralelismo.

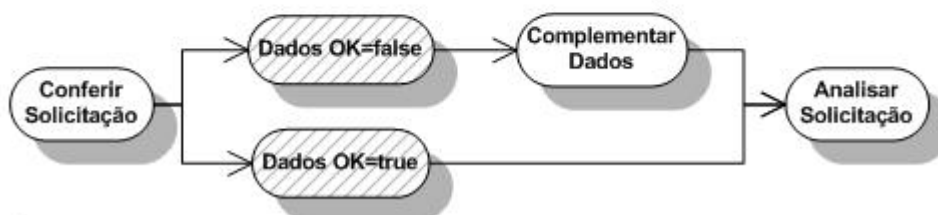
Do ponto de vista de execução, o WideFlow não diferencia os propósitos das condições de guarda. Cada uma é mapeada para uma atividade interna da máquina de estados e a satisfação da condição avaliada faz com que a atividade interna esteja concluída, assumindo o estado “terminada”, o que significa executar sua transição de saída como mostra a figura 4.6. O aninhamento de decisões deve garantir que exista sempre uma condição verdadeira prevenindo que o fluxo entre em *deadlock*.

O motor de workflow WideFlow avalia tantas condições de guarda quantas existirem nas decisões. Cada uma delas é mapeada para um **Fork** cujas transições não dependem de eventos externos, executado de forma automática sem interação com um ator, e cuja execução faz com que a condição correspondente seja avaliada. Se o resultado da avaliação retornar a condição esperada, o *fork* assume o estado “terminado”, o que faz com que o motor de workflow execute a transição de saída correspondente. A avaliação é sempre feita entre operandos cujos valores estão armazenados nos atributos dos documentos contra outros atributos ou constantes.

Decisão num Diagrama de Atividades UML



Desdobramento da decisão no engine



▣ atividades internas da máquina de estados (*transições com condição de guarda*)

Figura 4.6: Esquema de Execução de Atividades e Condições de Guarda

No WideFlow as sincronizações de fluxos paralelos ou concorrentes são mapeadas para os *joins*, também atividades internas que avaliam uma condição específica, isto é, o número ou *quorum* de atividades anteriores que deve ser observado para que o fluxo siga adiante. Em *joins* de fluxos paralelos a condição é sempre igual ao número de transições iniciadas na barra de sincronia (*fork*) [...] (OBJECT MANAGEMENT GROUP, 2003). Nos *joins* de fluxos concorrentes ou condicionais é avaliado o *quorum* da definição para verificar quantas atividades precedentes terminaram pois a transição de saída só ocorre quando ele for satisfeito.

A necessidade de verificar se alguma atividade específica foi instanciada antes de dar seqüência ao fluxo após o *join*, pode ser imposta numa condição de guarda na transição de saída do *join*, que avaliaria dados inseridos em atividades anteriores para este fim. Assim, a cada término de atividade um *join* seria instanciado, a avaliação do *quorum* sendo verdadeira causaria a execução da transição de saída que teria de avaliar sua condição de guarda e somente se esta for verdadeira o *join* será dado como terminado.

Resumindo, uma atividade interna do tipo *fork* ou *join* executa uma avaliação sobre uma condição, de acordo com seu tipo, que se verdadeira causa o término da atividade, que é o único evento necessário para disparar a atividade seguinte.

4.1.2 Assinalamento de Atores

No WideFlow o principal critério para decidir a granularidade de uma atividade é dado pelo assinalamento da atividade a um ator (CASATI et al., 1995) a partir da análise da interação. Todas as trocas de estado daquela atividade nesta interação são ativadas pelo ator participante do fluxo que repassa o controle ao WideFlow quando termina, cancela ou recusa a atividade.

No meta-modelo do WideFlow a associação de um **Actor** com uma instância de **TaskInstance** representa o assinalamento da atividade para um dos participantes que é feito durante a inicialização de uma **WorkTask**.

Os participantes do processo são tratados como atores que recebem acesso para um tipo de atividade do processo. Estão organizados em grupos os quais o motor de workflow percorre para atribuir atividades, observando a estratégia de assinalamento definida. Atividades com estratégias PUSH ou PULL (CASATI et al., 1995) têm seu assinalamento baseado neste mecanismo. Os outros dois tipos de assinalamento são diretos.

- Ao iniciador do processo – a atividade é colocada na lista de trabalho da instância de ator que iniciou o caso;
- Ao executor da tarefa **X** – a atividade é colocada na lista de trabalho da instância de ator que executou a atividade **X** no mesmo caso.

Desta forma é feita a simplificação da modelagem de atores que passa a interessar a cada tipo de processo. Os usuários de workflow, de acordo com o cenário descrito na seção 3, apesar de possuírem atribuições variadas, na maioria das vezes ocupam o mesmo cargo. Do ponto de vista da execução do processo, interessa o conjunto de permissões de cada ator. Este conjunto de permissões é singular e não coincidente com a hierarquia funcional. Os grupos, como o da figura 4.7, são modelados no diagrama de atividades utilizando as raias de responsabilidades, *swinlanes* [...] (OBJECT MANAGEMENT GROUP, 2003). Cada raia representa um grupo de pessoas que possuem acesso à atividade e aos objetos associados de acordo com a definição.

A preexistência de uma aplicação de segurança no ambiente descrito na seção 1.3 e 3, facilitou a adoção desta estratégia. Para o sistema de segurança, um sistema de informação possui programas, um conceito lógico sobre a partição das funcionalidades do sistema, e grupos. Aos grupos são atribuídas funcionalidades do sistema, como por exemplo incluir venda ou cadastrar clientes. Um ator se torna usuário do sistema ao ser incluído num grupo, herdando as permissões daquele grupo.

Para que o motor de workflow pudesse atribuir uma atividade para um ator foi utilizada esta aplicação de segurança. Um processo definido possui o mesmo status que um sistema de informação frente à aplicação de segurança. Cada atividade do processo é como um programa e um grupo será criado com permissão de acesso para aquela atividade. Inserir atores no grupo significa atribuir permissão de execução da(s) atividade(s) para o ator, cabendo ao motor de workflow observar a estratégia de assinalamento definida para cada atividade.

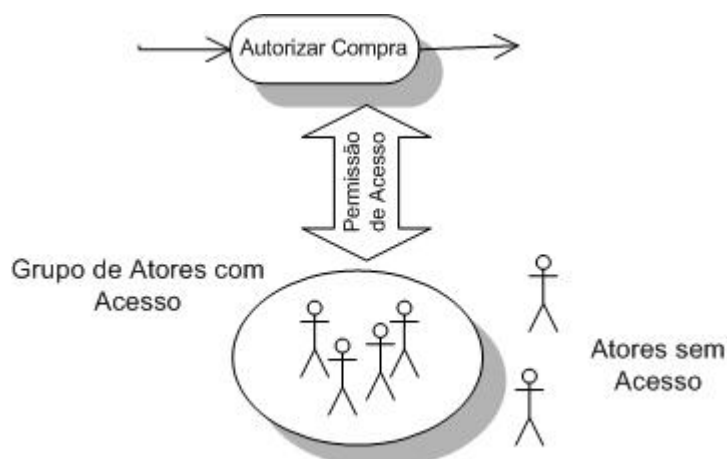


Figura 4.7: Grupo de Atores

Cada funcionário público possui um *login* de rede. É por meio do login que ele é identificado, tornando-se um **usuário ativo** no contexto do sistema operacional, e para as aplicações é um ator que possui uma sessão de trabalho autenticada. Através da autenticação de rede o ator acessa sua lista de trabalho e interage com as atividades assinaladas para ele.

Esta abordagem une os requisitos de segurança das aplicações de workflow à política de segurança desejada. A gestão de permissões é responsabilidade dos próprios usuários na figura de um administrador de aplicações que pode incluir ou retirar pessoas dos grupos. No caso de uma aplicação de workflow, um processo automatizado, este administrador é o dono do processo, sendo permitido que ele designe outras pessoas de sua confiança para este trabalho.

Assim são alcançados requisitos de segurança por meio de autenticação integrada e controle de permissões, de flexibilidade com o desacoplamento do gerenciamento dos executores de processo da necessidade de modelagem da estrutura organizacional ao mesmo tempo que o assinalamento de atores é feito na medida exata da necessidade apresentada e discutida nas seções anteriores.

4.1.3 Manipulação de Dados e Informações

Quando interage com o workflow para executar suas atividades os atores preenchem de conteúdo os dados do processo. O Workflow armazena e apresenta estes dados em cada atividade posterior. A armazenagem é feita em estruturas próprias, mesmo que os dados sejam

provenientes de sistemas de informação, isolando os dados do processo do ambiente externo ao workflow, como discutido na seção 2.1.5.

Um conjunto de dados é formado por dados que se relacionam entre si formando o que usualmente conhecemos como documento, uma agregação lógica com uma finalidade específica no mundo real. Na análise do processo são elicitadas as informações inerentes e necessárias ao processo, fazendo surgirem os documentos do processo.

No WideFlow os documentos são persistidos em *strings* XML [...] (WORLD WIDE WEB CONSORTIUM, 2000) num atributo das instâncias de **DocumentInstance** (meta-modelo da figura 24). Ao manipular a estrutura do objeto XML, o WideFlow obtém o conteúdo para envio ao cliente do motor de workflow. Armazenar documentos em XML alinha o encapsulamento dos dados do workflow à abordagem de interpretar o processo passo a passo a partir da sua definição, apresentada na seção anterior. Cada instância de processo guarda em si a estrutura de seus documentos, tornando-a independente da definição após sua criação, o que também facilita a evolução dos dados nos esquemas de processos.

Um documento pode possuir mais de uma ocorrência em uma mesma atividade. Por exemplo, num processo de solicitação de férias o formulário para fazer a solicitação prevê uma matrícula funcional, ou seja, a solicitação de férias para um empregado. Se as férias são solicitadas pelo supervisor no setor de origem, para o setor de recursos humanos, pode ser que o supervisor queira solicitar férias para um lote de empregados. Neste caso deve ser possível inserir no processo mais de uma ocorrência para o mesmo documento. Isto é representado no modelo com o estereótipo <<occurs>> imposto no *ObjectFlow* [...] (OBJECT MANAGEMENT GROUP, 2003) traçado entre o documento e a atividade onde a criação de ocorrências é possível, de forma análoga ao estereótipo mostrado na figura 4.8.

Quando um documento é acessado por múltiplas atividades concorrentemente, o isolamento da informação deve ser previsto *a priori* e modelado. A menor unidade de isolamento é o documento e sobre ele ocorrerá o bloqueio de transação. A necessidade de *lock* de um documento entre duas atividades é demonstrada no fluxo pelo uso de um estereótipo de um *Object Flow* que liga os dois documentos como na figura 30.

A integração com sistemas legados é feita a partir de *interfaces* de troca definidas de acordo com a lógica e necessidade do processo. Uma *interface* possui propriedades, as quais o WideFlow percorre num mecanismo similar à reflexão, comparando o nome dado à propriedade com os nomes dos atributos dos documentos, e um método padrão de execução. Ao encontrar coincidência entre atributos e propriedades, o WideFlow atribui o valor do documento à *interface*. Depois desta atribuição dos valores à *interface* o WideFlow chama o método padrão publicado. Após a execução o WideFlow faz a operação inversa, varrendo a *interface* e atribuindo os valores contidos nas suas propriedades aos atributos dos documentos quando seus nomes forem correspondentes. Este comportamento é

baseado no padrão *Command* (GAMMA et al., 2000), onde uma requisição é feita sem que o requisitante saiba exatamente o que ou onde ela executa.

Esta *interface* é modelada numa classe chamada **Agent**, um *plug-in* no contexto do WideFlow pois pode ser “plugada” nos eventos de inicialização ou de término de uma atividade. Um **Agent** também pode ser associado a um documento específico utilizado numa determinada atividade, sendo executado somente quando o estado do documento é alterado. No modelo isto é representado pelo nome do agente sobre a linha que liga a atividade ao documento, o *ObjectFlow*. O resultado da execução de um **Agent**, **TRUE** ou **FALSE**, também determina se a atividade pode alterar seu estado para “terminada”.

Com isto é possível trocar dinamicamente os dados, transformando e permitindo interação com legado. Desde que o resultado da execução do **Agent** também determina o andamento do fluxo, é possível validar os dados, impondo regras específicas ao código do **Agent** implementado. Pode-se também imprimir dinamismo entre documentos, quando dados de um documento podem ser atribuídos ou mesclados num outro documento com *lay-out* determinado.

Os agentes executados nos eventos de início ou finalização das atividades são modelados por meio de *actions* [...] (OBJECT MANAGEMENT GROUP, 2003), como na atividade **A** da figura 4.8. A *action* é implementada por uma *Type Library* COM+ (KIRTLAND, 1999) fisicamente existente no ambiente COM+. O WideFlow pode obter uma referência para esta *interface* e manipulá-la como descrito.

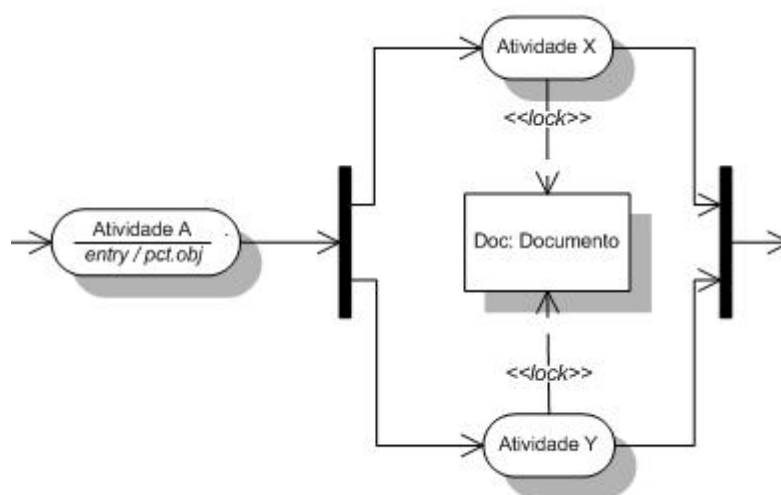


Figura 4.8: Bloqueio concorrente entre objetos

Arquivos anexados ao processo são armazenados em uma estrutura de diretórios e a localização da *url* é persistida em uma instância de **DocumentInstance**. Toda a manipulação ocorre por intermédio da especialização dos métodos desta classe.

4.1.4 Arquitetura

A arquitetura do WideFlow faz uso do mesmo *framework* para desenvolvimento de aplicações com componentes distribuídos COM+ (ARAUJO; GOMIDE; PETRILLO, 2003) utilizado para a elaboração de aplicações corporativas em uso no ambiente discutido anteriormente. Nele está o modelo de persistência que preenche a lacuna existente entre objetos que devem ser persistidos e o mecanismo de armazenamento no banco de dados por meio de uma hierarquia de classes básicas extensíveis contendo as principais necessidades de armazenamento e recuperação dos objetos do meta-modelo. Ele permite a manipulação de coleções de objetos que opcionalmente podem ser transformadas em um *recordset* para uso pela camada de apresentação.

Em uma arquitetura de múltiplas camadas existe uma camada responsável por todas as regras do domínio, formando o cerne funcional da aplicação (ARAUJO; GOMIDE; PETRILLO, 2003). No WideFlow os componentes que realizam as abstrações do meta-modelo encapsulam as informações e o comportamento inerente a cada uma. Utilizar estes componentes exigiria que o cliente tivesse conhecimento de sua estrutura, sabendo exatamente para qual objeto apontar, qual método chamar e principalmente, como navegar através dos relacionamentos. Para facilitar o uso do motor de workflow foi criado um componente que fornece uma API de serviços mostrada na figura 4.9, inspirada nas WAPI 2 e 3 [...] (WORKFLOW MANAGEMENT COALITION, 1995), para que os clientes o utilizem e realizem as ações que desejam. O serviço se encarrega da verificação das regras do processo e do estado de todas as informações e atividades do processo.

WideFlowExec
UsuarioAtivo : IGRDUsuario
GetProcessDefinitionList() : TFWProcessDefinitionLst InitProcess(IDProcessDefinition : Long, Descricao : String) : Boolean GetTaskInstanceList(IDProcess : Integer) : TaskDefinitionLst GetTaskInstanceDocLst(IDtaskInstance : Integer) : TFWDocumentInstanceLst GetDocumentInstance(IDDocumentInstance : Integer) : ADORecordSet UpdateDocumentInstance(IDDocumentInstance : Integer, Doc : ADORecordSet) : Boolean ExecuteTaskInstance(IDTaskInstance : Integer) : Boolean GetTaskInstanceExpiredList() : TFWTaskInstanceLst GetTaskInstanceInDeadLockList() : TFWTaskInstanceLst GetActorList(IDTaskInstance : Integer) : ActorLst GetProcessParticipationList(IDUsuarioAtivo : Integer) : TFWProcessInstanceLst toDelegate(IDTaskInstance : Integer, IDActor : Integer) : Boolean GetTaskInstance(IDTaskInstance : Integer) : TaskInstance

Figura 4.9: Componente de serviços de workflow

Ergonomia de software tem sido uma das preocupações dos analistas com relação aos sistemas de informação ao longo de muitos anos.

Aqui também a preocupação com a ergonomia se faz presente, ainda mais que com a automação de atividades pretendemos auxiliar a própria

execução do trabalho dos participantes. Mesmo com a flexibilidade para qualquer analista possa criar interfaces próprias para a camada de apresentação (IHM) de sua aplicação de workflow e utilizar os serviços do componente de execução, o WideFlow possui um conjunto de interfaces que utiliza os serviços do componente para expor dados de processos aos usuários, atores de processos, recebendo as respostas destes ao interagirem com as atividades e os documentos.

Esta camada é feita em ASP e JavaScript, com o uso de diversos *templates* que encapsulam funcionalidades típicas de interface (ARAUJO; GOMIDE; PETRILLO, 2003). Nesta arquitetura o ASP é responsável por instanciar o componente COM+ recebendo tipos primitivos, *recordsets* ou coleções de objetos, e utilizar as informações de acordo com a lógica de apresentação.

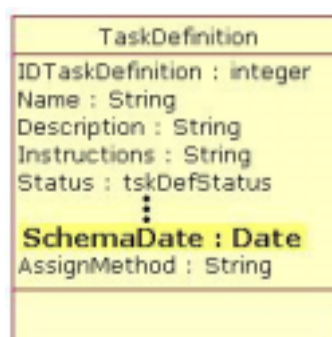


Figura 4.10: Data do Esquema na Definição

O Wideflow possibilita a evolução dos esquemas de definição dos processos, permitindo que alterações no fluxo das atividades se reflitam somente nas instâncias futuras de processos, pois o motor de workflow faz a verificação da validade do esquema nos atributos dos objetos de definição – figura 4.10, selecionando o esquema existente na data da criação da instância do processo. Novos processos somente são criados com base no esquema ativo, na maioria dos casos o mais recente, mas o analista do processo pode ativar um esquema anterior, fazendo com que os novos processos passem a obedecer àquele esquema.

Não só o esquema das definições pode evoluir sem que nenhuma intervenção seja feita no motor de workflow ou nas aplicações, mas também os documentos modelados também podem ter sua estrutura alterada com a mesma transparência. Implementados em XML, as novas definições não influenciam a tramitação de instâncias anteriores. Já que a apresentação dos dados do processo é baseada na sua estrutura e que XML encapsula a estrutura e dados, a evolução dos documentos modelados é independente dos processos em tramitação.

4.1.5 Definição do Processo

A modelagem e definição do processo, WAPI 1 [...] (WORKFLOW MANAGEMENT COALITION, 1995), são feitas no diagrama de atividades utilizando o Rational Rose Enterprise®.

O modelo comporta os elementos e estereótipos apresentados e discutidos nas seções anteriores. Com ele é possível expressar os requisitos em automação de processos administrativos como os encontrados no cenário discutido nas seções 1.3 e 3.2.

O diagrama de atividades da UML desenhado com auxílio do Rose já é utilizado naquele cenário por desenvolvedores no processo de modelagem de sistemas de informação (ARAUJO; GOMIDE; PETRILLO, 2003). Sua utilização não é novidade para aqueles profissionais que fazem modelagem de processos e procedimentos na elicitação de requisitos de sistemas de informação. A novidade são os estereótipos para modelar o comportamento e semântica encontrados no motor de workflow do Wideflow.

A definição do processo é obtida deste diagrama e persistida para que o WideFlow faça a interpretação dos passos em cada nova instância de processo. A disponibilização da definição para o WideFlow é feita com a execução de Rational Rose Script Language® escritos no escopo deste trabalho para percorrer o gráfico e traduzir suas construções para as abstrações do meta-modelo do WideFlow. Futuramente novas funções como validar processos verificando caminhos (flow), condições de guarda, *deadlocks* e otimizações podem ser inseridas.

4.2 O Processo da Declaração Municipal Automatizado

A automação do processo de solicitação de DM, apresentado e discutido na seção 3, permite ganhos projetados no atendimento às demandas da população à administração pública do município de Porto Alegre.

- a) Ganhos em escala permitem atender uma parcela maior de cidadãos em tempo compatível e ainda liberando recursos para atendimento em outras áreas como saúde, saneamento, urbanismo, etc.
- b) Ganhos em qualidade de atendimento, oferecendo serviços diretamente à população pela Internet, facilitando o acesso de muitos cidadãos.

O motor de workflow criado e apresentado no escopo deste trabalho é utilizado na PROCEMPA para automação de processos que passam pelo domínio de diversos sistemas de informação, coletando ou enviando dados para estes, numa simbiose natural e desejada.

Foi criada uma aplicação, o SDM (Solicitação de DM) para manipular o processo, interagindo com o motor de workflow e implementando IHM customizadas que apoiam os requerentes a fazerem solicitações. Em suma, o SDM implementa uma lógica própria e fina de interface e delega para o

motor de workflow as regras do processo, utilizando-o por meio de uma fachada.

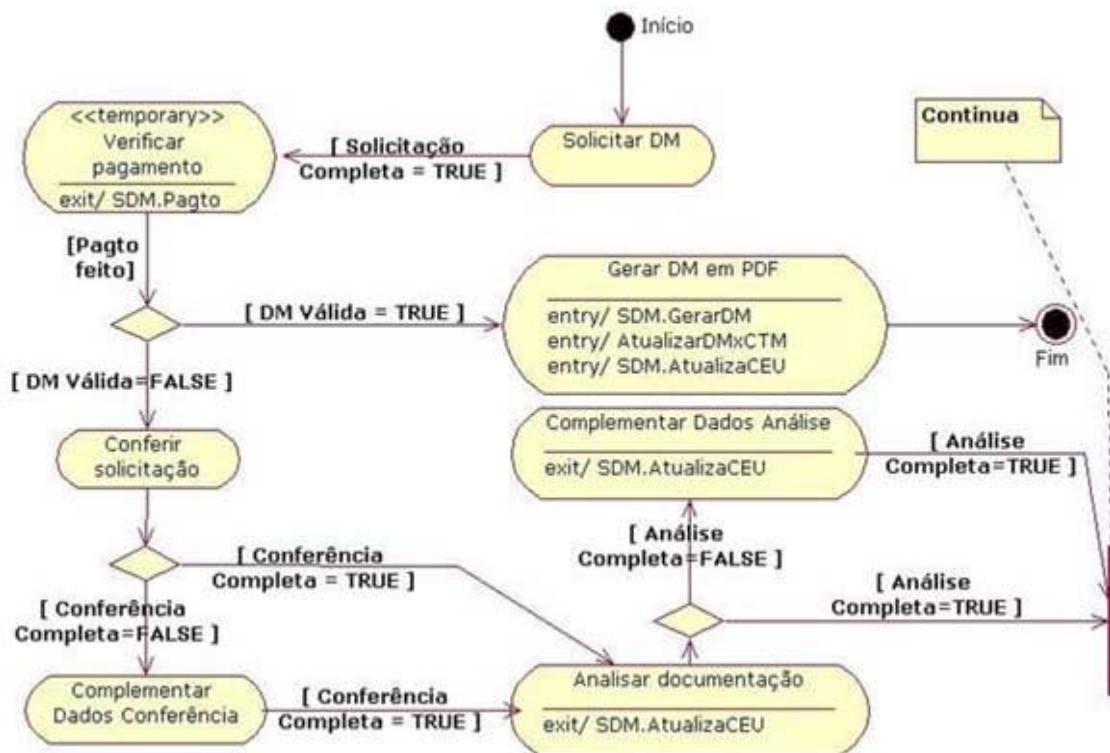


Figura 4.11: Diagrama de Atividades do Processo da DM

Neste cenário existem centenas de processos deste tipo e sua automação passa a fazer parte do trabalho diário de analistas de sistemas e arquitetos (ARAUJO; GOMIDE; PETRILLO, 2003) num esforço de produção de soluções de TI para suportar a mudança organizacional e a melhoria dos processos na administração pública com a utilização do enorme ativo intangível representado pelas informações disponíveis nos bancos de dados.

Além de ganhos do ponto de vista administrativo, esta solução traz uma enorme economia ao orçamento público em licenças, treinamento, equipamentos e infra-estrutura, que de outra forma seriam necessários para utilizar ferramentas de workflow comerciais.

A automação do processo solicitação de DM, com semântica e decisões detalhadas, é apresentada nesta seção. As figuras 4.11 e 4.12 apresentam o gráfico do processo.

A primeira atividade, solicitação da DM, é ativada pelo sistema de informação SDM. Ele irá controlar o acesso dos requerentes, apoiar a entrega dos documentos e informações necessárias à solicitação, disparando eventos contra o motor de workflow que fazem executar **Agents** responsáveis por reunir dados de expedientes únicos, logradouros, geração

de documento para pagamento de taxa, etc, em vários sistemas de informação.

Esta interação com o requerente é feita em interfaces customizadas pelo SDM que utilizará o WideFlow como parte de sua arquitetura, solicitando ao motor de workflow a criação de uma nova instância de processo, entregando os dados e arquivos previstos na primeira atividade e disparando o evento de execução da atividade Solicitar DM. A especificação de caso de uso abaixo mostra esta interação.

Evento de Início:

Este caso de uso inicia quando o Requerente seleciona a opção "Solicitação de DM / Incluir".

Pré-Condição:

Requerente autenticado.

Fluxo Básico:

- a) O sistema solicita as inscrições de IPTU dos endereços para os quais deseja uma DM e a área total titulada destes imóveis.
- b) O Requerente informa uma ou mais inscrições e a área total titulada e submete a solicitação.
- c) O sistema valida os dados informados e dispara no WideFlow os eventos que executam os agentes responsáveis por preencher os dados nos documentos: os endereços para cada inscrição no banco do Cadastro Imobiliário, a descrição dos logradouros no banco do Cadastro de Logradouros e o número de Expediente Único se já existir um para o conjunto de endereços. Se faltar alguma inscrição, que também pertença ao imóvel ou bloco de imóveis ela deverá ser obtida. Apresentar para confirmação dos endereços.
- d) O Requerente confirma os endereços.
- e) Após, exibe a tela com a opção "Enviar Documentos" habilitada e a opção "Emitir DAM" desabilitada, com a seguinte mensagem: "Solicitação enviada. Aguardando o envio da Planta de Situação e do Título de Propriedade. O código desta solicitação é 99".
- f) O Requerente seleciona a opção de *upload*, selecionando os arquivos e iniciando a transferência.
- g) O sistema recebe os arquivos dispara no Wideflow os eventos que executam os agentes responsáveis pela atualização dos respectivos documentos.
- h) O sistema DESABILITA a opção "Enviar Arquivos" e habilita a opção "Emitir DAM", com a seguinte mensagem: "Documentos enviados. Aguardando a emissão e o pagamento do DAM. O código desta solicitação é 99."
- i) O Requerente seleciona a opção de emissão de DAM.
- j) O sistema dispara no Wideflow o evento que executa o agente que fornece os dados para o DAM (ver use case "Emitir DAM") e emite-o. O sistema marca como *true* o atributo "Solicitação Completa", dispara o evento de atualização do documento correspondente no WideFlow e dispara a execução da atividade Solicitar DM para realizar a transição para a próxima

atividade do processo. Após, exibe a tela com somente a opção "Voltar" habilitada

Pós-Condição:

Uma nova instância de processo para solicitação de uma DM, com os documentos previstos, inicia a tramitação.

Fluxo Alternativo:

No passo "c", se o sistema verificar que existe uma DM válida, isto é, com data de emissão menor que um ano, deverá se comportar como segue:

- a) O sistema mostra esta DM e oferece a opção de emissão de uma segunda via.
- b) O Requerente opta por emitir a segunda via;
- c) O sistema marca a informação "DM Válida" modelada no objeto "Solicitação de DM" como *true* e executa o método de atualização de documentos do WideFlow para este documento.
- d) O fluxo segue para o passo i

Se o sistema SDM, ao interagir com o requerente, a qualquer momento considerar que a documentação para solicitação do processo está completa, disparará o evento de execução da atividade.

A atividade Verificar Pagamento, uma atividade temporal, é instanciada e sua execução é feita pelo próprio motor de workflow diariamente as 06:00 pois o processamento diário dos avisos de crédito bancário ocorre antes deste horário. O motor de workflow altera o estado da tarefa para terminada, o que dispara o **Agent**, previsto na *action* **SDM.Pagto** como discutidos na seção 4.1.3. Este agente executa um algoritmo que acessa o sistema de Controle de Arrecadação Municipal, verificando se o pagamento do documento emitido já foi apropriado. Em caso positivo o **Agent** retorna *true*, em caso contrário, *false*.

A execução do **Agent** com sucesso (*true*) faz com que o motor de workflow mantenha o estado da atividade como terminado, de outro modo, o motor de workflow faz com que a atividade retorne para o estado anterior, pronta para executar novamente quando for fechado mais um período.

Na seqüência é feita uma avaliação: se o requerente optou por emitir uma segunda via, ainda na primeira atividade. Em caso afirmativo o fluxo segue para a atividade onde a DM será gerada e transformada em PDF e enviada ao requerente e o processo tem seu fim.

Caso contrário - ou não existia uma DM válida ou o requerente tenha optado por emitir uma nova DM - o fluxo se encaminha para a atividade de Conferência da Solicitação. Neste passo um conferente fará diversas análises entre a solicitação e os dados, podendo decidir por utilizar um outro código para o Expediente Único que considere mais apropriado para o processo ou ainda, inserir novos endereços para aquele Expediente Único.

Com isto o SDM dispara eventos contra o Wideflow que fazem com que ele atualize os dados sobre os logradouros e endereços da solicitação.

O conferente pode decidir pela complementação de documentação marcando a decisão num atributo do processo e disparando a execução da atividade. O motor de workflow passa o estado da instância da atividade para “terminada” e inicia a atividade Complementar Dados Conferência. Um e-mail é enviado ao requerente pelo agente **SDM.E-mail**, solicitando os dados desejados com o texto redigido pelo conferente. O requerente envia os dados para o conferente que os anexa ao processo substituindo os anteriores. Estas duas atividades serão executadas pelo mesmo ator e sua divisão ocorre por decisão de análise, para marcar dois estados distintos no processo. Se o requerente não responder em um período de 15 dias o motor de workflow irá encerrar o processo. Ao receber a complementação dos dados, o conferente encerra a atividade de complementação e o fluxo segue para a atividade seguinte.

Uma instância de atividade Analisar Documentação é iniciada e um ator assinalado. A interação ator *versus* atividade é análoga à atividade anterior de conferência, com análise e envio de e-mail ao requerente para possível complementação de documentação. Quando a análise estiver completa, ou na atividade Analisar Documentação ou na atividade Complementar Dados Análise, será executado o agente **SDM.AtualizaCEU**, que atualiza o sistema de Cadastro de Expedientes Únicos com as informações deste processo.

É importante frisar que a implementação de cada **Agent** é responsabilidade conjunta dos analistas da aplicação de workflow e do sistema legado, pois são trechos de código que devem obedecer às regras de cada domínio.

O SDM também criou interfaces próprias para apoiar as atividades de conferência análise e complementação de dados. Quando o ator responsável tanto pela Análise da Documentação quanto pela Complementação Dados da Análise estiver satisfeito irá marcar o processo como completo e acionar o evento de execução da atividade e a transição será efetuada pois a condição será verdadeira.

Após esta primeira análise o fluxo segue paralelamente para a atividade Análise de Projetos Geométrico, Viário, do Regime Urbanístico, e de Equipamentos Comunitários e para a atividade Analisar Infra-Estrutura.

A atividade Analisar Projetos Geométricos, Viários, Regime Urbanístico e Equipamentos Comunitários e a atividade Analisar a Infra-Estrutura são executadas também com apoio de interfaces próprias do SDM introduzindo a lógica própria que os atores utilizam para realizarem seu trabalho. A planta de situação é utilizada de forma concorrente nessas atividades.

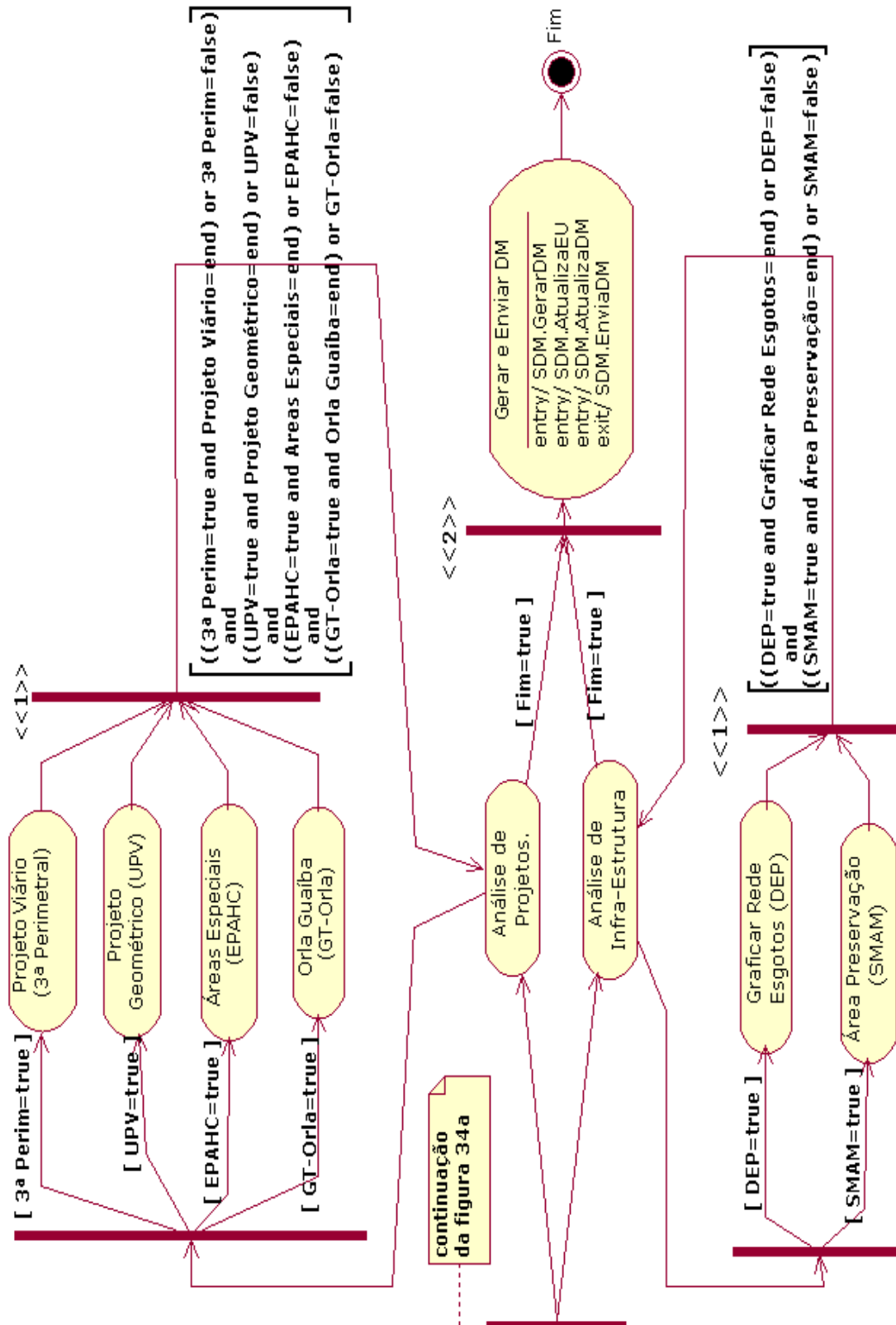


Figura 4.12: Diagrama de Atividades do Processo da DM

Na primeira, por decisão do ator com base no contexto, o processo pode ser enviado para detalhamento, verificação ou inclusão de

informações em outros setores ou órgãos. Ao fazer o encaminhamento a atividade é encerrada. Um detalhe em relação às diversas atividades em paralelo é a concorrência entre os documentos enquanto as atividades são realizadas.

As atividades Detalhar Projeto Geométrico, Verificar Áreas de Preservação, Detalhar Projeto Viário, Informar Área da Orla e Definir Projeto Geométrico, ao modificarem dados nos documentos do processo, devem modificar a mesma instância de documento, preservando a integridade das informações que, ao retornarem para a atividade Analisar Projetos Geométrico, Viário, Regime Urbanístico e Equipamentos Comunitários, estarão completos, permitindo ao ator dar seguimento à solicitação.

Interação semelhante sofrerá a atividade Analisar Infra-Estrutura, onde também os documentos serão compartilhados concorrentemente. Se existirem documentos comuns nas atividades de análise de projetos e análise de infra-estrutura e suas seqüências, eles deverão observar o requisito de concorrência.

O participante assinala os atributos correspondentes aos setores, departamentos ou secretarias por onde o processo deverá tramitar. Ao executar a atividade, o Wideflow avalia as condições impostas pelo ator e realiza o fluxo desejado, instanciando as atividades cuja transição de origem tenham as correspondentes condições de guarda resultado como verdadeiras.

Após a execução de cada atividade, o *join* é instanciado, pois o *quorum* definido é igual a 1. A condição de guarda na transição de saída do *join* faz o Wideflow avaliar se todas as atividades que foram instanciadas já encerraram, quando a transição ocorre e o fluxo faz com que o motor de workflow crie uma nova instância da atividade Analisar Projetos Geométricos, Viários, Regime Urbanístico e Equipamentos Comunitários ou Análise Infra-Estrutura, conforme o seqüenciamento.

Este conjunto de atividades de análise e verificação é feito com auxílio das informações recolhidas dos sistemas de informação e também pela análise não automatizada de plantas, mapas e outras informações gráficas que cada órgão mantém. A automação destas atividades necessita da conclusão de projeto de informações georreferenciadas em execução pela administração municipal. As atividades são realizadas com o auxílio das IHM do próprio Wideflow que incorporam o padrão lista→ detalhe, muito utilizado em sistemas de informação que têm sua camada de apresentação baseada em interfaces Web.

Quando considerarem finalizados seus respectivos trabalhos, os atores responsáveis pelas atividades Analisar Projetos Geométricos, Viários, Regime Urbanístico e Equipamentos Comunitários ou Análise Infra-Estrutura marcarão o atributo “Finalizado” em suas respectivas atividades e dispararão o evento de execução da atividade. Desta feita o único caminho verdadeiro para cada atividade é a transição que leva ao *join* final que possui *quorum* igual a 2, significando que o WideFlow somente executará a transição de saída do *join* quando as duas transições de origem ocorrerem.

A atividade Gerar DM executa três agentes em sua inicialização como na figura 4.13. Dois deles atualizam sistemas de informação, o outro dispara o evento para execução do programa que gera a DM no domínio do sistema de informação, recebe-a e transforma-a em PDF, armazenando o arquivo resultante na árvore de diretórios do motor de workflow e o endereço no objeto “DM em PDF”.



Figura 4.13 Junção (join) de fluxo

Nesta última atividade o workflow aciona um sistema de informação que historicamente tem sido utilizado para gerar a DM. Ele irá criar uma nova DM em seu domínio. O workflow recebe esta DM como retorno e se encarrega de gerar um documento em formato PDF e enviá-lo ao requerente.

Esta atividade não é assinalada para nenhum ator, caso em que o motor de workflow mesmo se responsabiliza pelo evento que dispara sua execução imediatamente após sua inicialização. Na finalização desta atividade um agente para envio da DM ao requerente será executado. Esta ação é feita por um agente, com base no cadastro do requerente do próprio SDM, pois o requerente não é um ator do processo, mas um *Business Actor* (JACOBSON; ERICSON; JACOBSON, 1994) como discutido na seção 3.2.3, que o motor de workflow desconhece.

5 CONCLUSÕES

Nos últimos anos crescem as pesquisas sobre o uso da UML para modelagem e execução de workflow que têm frutificado em alguns motores de workflow e ferramentas baseadas no diagrama de atividades.

Estas pesquisas têm produzido um significativo acréscimo conceitual ao diagrama de atividades, introduzindo formalismo para que ele seja a base para os profissionais que necessitam criar aplicações baseadas em fluxo de atividades.

Em BASTOS e RUIZ (2001) e BASTOS e RUIZ (2002) temos um meta-modelo orientado a objetos para a criação de uma máquina de estados capaz de executar workflows modelados em diagramas de atividade estereotipados. ESHUI e WIERINGA (2001b) introduzem semântica nos diagramas de atividade, formalizando decisões e conceitos para aumentar o seu poder de expressão.

CASATI et al. (1995) instituem a participação dos atores como disparadores de eventos na interação com atividades num fluxo de trabalho, lançando o gérmen para motores de workflow reativos aos atores citados em ESHUI e WIERINGA (2001b).

GUTIERREZ (1997) mostra uma interface para troca de dados entre os domínios das aplicações de workflow e de sistemas de informação, lançando as bases para a criação de *hooks* ou agentes que possam imprimir dinamismo aos dados manipulados pelos processos ou estender o comportamento específico do caso.

MANOLESCU (2001) debate a adequabilidade arquitetural da maioria das ferramentas de workflow existentes no mercado e as premissas nas quais são baseadas. Em contrapartida prega a aplicação de técnicas orientadas a objetos para a criação de suporte à workflow mais adequado à realidade dos ambientes de engenharia de software atuais.

As características próprias do diagrama de atividades e sua simplicidade aliadas a um alto poder de expressão para processos de produção (NICOLAO, 1996) e ao princípio de que o evento para o início de uma atividade seja o término da atividade anterior - que leva à generalização de todas as atividades executadas como no meta-modelo do WideFlow, simplificando o funcionamento do motor de workflow - contribuem para sua adoção em pesquisas e soluções de mercado.

Além disto, atender requisitos como capacidade de evolução dos esquemas do fluxo e dos dados, simplificação da modelagem de atores em grupos e papéis e existência de agentes para imprimir dinamismo aos dados e para realizarem a integração com sistemas legados, eqüivale a caminhar rumo à um grau satisfatório de abrangência e de adequabilidade para soluções integradas em ambientes de engenharia de software, fechando o ciclo de vida das aplicações de workflow.

Aproximar workflow de sistemas de informação, fazendo com que as duas áreas não só convivam, mas se integrem no dia a dia dos profissionais, colocando as práticas e conceitos de cada um lado a lado, permite encontrar similaridades e fazer convergir os conceitos, flexibilizando e determinando o sucesso de aplicações de workflow como uma nova classe de soluções para os desenvolvedores de sistemas de informação.

Num trabalho que aborda as construções fundamentais elicítadas na modelagem de workflow, AALST, HOFSTEDE e BARROS (2000a) e AALST, HOFSTEDE e BARROS (2000b) são analisados vários padrões que apresentam os principais blocos de construção para as linguagens e, conseqüentemente, para o motor de workflow que a suportará.

Os padrões são construções que mapeiam requisitos básicos e avançados encontrados na análise e modelagem de aplicações de workflow. A partir do cotejamento das soluções discutidas com as funcionalidades de uma máquina de estados implementada em um motor de workflow, é possível avaliar sua adequabilidade e capacidade de expressão, pois os padrões desviam o foco para os requisitos dos processos independentemente de qualquer linguagem.

A seguir é apresentada uma discussão do Wideflow com os padrões de van der Aalst e a capacidade do diagrama de atividades proposto neste trabalho em expressá-los.

Sequence:

O Wideflow implementa este padrão sem problemas.

A expressividade é direta.

Parallel Split:

O Wideflow implementa paralelismo sem problemas. *Threads* podem ser implementadas sem perda de funcionalidade.

A expressão da construção é explícita no modelo.

Synchronization:

A convergência de paralelismo é feita em um join e a sincronia é feita por meio do *quorum* imposto na definição.

A expressão da construção é explícita no *quorum* definido.

Exclusive Choice:

O Wideflow implementa condições de guarda que, colocadas após uma decisão, farão com que a seqüência siga por um ou por outro fluxo.

A semântica é definida pela condição de guarda.

Simple Merge:

Já que somente um fluxo terá seqüenciamento até o *join*, o *quorum* padrão (um) é suficiente para garantir que a transição de saída do *join* seja executada. De acordo com a definição não existe risco de múltipla instanciação.

A expressão da construção é explícita no *quorum* definido e modelado no estereótipo do *join*.

Multi-Choice:

O Wideflow avalia todas as condições de guarda nas transições de saída de uma decisão, isto é, executa todas as atividades internas. Se não existir um caminho válido a seguir a WorkTask de origem, anterior à decisão, não é terminada, permanecendo sob a responsabilidade do ator.

A expressão da construção está explícita na condição de guarda.

Synchronizing Merge:

Como a quantidade de transições verdadeiras só ocorre em tempo de execução, a aplicação deve controlar os disparos para sincronizar o término de todas as atividades disparadas. No Wideflow isto é possível colocando uma condição de guarda na transição originada no *join*, que avalie se o número de atividades que terminaram é igual ao número de atividades que foram iniciadas. A atribuição do número de atividades disparadas deverá ser feito no algoritmo de um agente (**Agent**).

A expressão da construção está explícita na condição de guarda

Multi-Merge:

Como o Wideflow interpreta a definição, instanciando uma a uma as atividades conforme a definição do caso, o *join* seria instanciado a cada transição de entrada que ocorresse se seu *quorum* for igual a um.

O problema é a sincronização das possíveis versões dos dados nos documentos tratados pelas atividades posteriores. Se no contexto do caso modelado puderem existir várias instâncias para um mesmo documento o Wideflow os manipularia sem problemas, permitindo aos participantes interagir com cada uma das “versões” de dados.

A expressão da construção seria explícita no modelo.

Discriminator:

A implementação deste padrão é levada a efeito com o uso de uma condição de guarda na transição de saída do *join*.

Ele teria um *quorum* igual a um e a cada ativação a avaliação da condição de guarda na transição de saída seria levada a efeito verificando se nenhuma outra instância daquele *join* existe no mesmo caso, quando a transição poderia ocorrer. Se ao contrário, uma outra instância existir, a transição não ocorre.

Desta forma somente uma instância de *join* realizaria sua transição de saída.

A expressão da construção é explícita na condição de guarda.

N-out-of-M Join:

Para que ocorra somente uma instanciação do *join*, o controle é feito no *quorum* definido no estereótipo de *join*.

Para a escolha de ativações específicas, a solução utilizada anteriormente para o padrão *Discriminator* pode ser facilmente portada para este caso.

A expressão da construção igualmente é explícita na condição de guarda.

(Também aqui, se mais de uma transição de origem puder ocorrer encontraremos o problema de sincronização de dados discutido no padrão *Multi-Merge*.)

Arbitrary Cicles:

A ocorrência de ciclos arbitrários implica na criação de mais de uma instância para a mesma atividade e dependendo do caso, no término implícito de um processo.

O Wideflow faz esta manipulação sem problemas.

A expressão da construção é explícita no modelo.

Implicit Termination:

Apesar do Diagrama de Atividades da UML exigir que somente uma terminação (*final state*) exista e que toda a atividade que não possua transição de saída, caracterizando o fim do fluxo, tenha uma transição para o *final state*, o Wideflow só termina o caso se não existir nenhuma *WorkTask* aguardando execução.

Para o Wideflow a ligação com o *final state* é representacional mas a construção é explícita no diagrama.

Padrões envolvendo múltiplas instâncias :

São os padrões *Multiple Instances with a Priori Design Time Knowledge*; *Multiple Instances with a Priori Runtime Knowledge*; *Multiple Instances with no a Priori Runtime Knowledge* e *Multiple Instances Requiring Synchronization*.

Apesar de possuir capacidade para manipular mais de uma instância da mesma atividade ao mesmo tempo, não existe uma construção no diagrama de atividades para a ativação iterativa de atividades.

Uma solução seria o uso de agentes que disparem atividades anteriores no fluxo, o que simularia um *until*. Apesar de factível a construção permaneceria implícita no modelo.

Paralelamente, o Wideflow permite a criação de coleções de documentos numa atividade, quando várias ocorrências de um documento podem ser criadas. Posteriormente elas podem ser tratadas naturalmente por um participante do workflow ou em lote com o auxílio de um ou mais agentes (**Agent**).

Por carecer de meios para ativar iterativamente uma atividade (exceto com o uso de construções complexas e não modeláveis), o único padrão envolvendo múltiplas instâncias que o Wideflow suporta explicitamente é o **Multiple Instances with a Priori Design Time Knowledge**.

Temporal Relations:

Neste padrão existe o relacionamento entre duas atividades em diferentes fluxos paralelos. Num dos fluxos uma atividade **X** deve aguardar que uma atividade **Y** no outro fluxo inicie antes de realizar a sua transição de saída.

Com o auxílio de uma atividade temporal, Temporal WorkTask, disparando um agente que perscrute a atividade **Y**, quando poderá assumir o estado “terminada” e realizar sua transição de saída.

O intento da construção é explícito na condição de guarda aposta na transição de saída da atividade temporal.

Deferred XOR:

Como na construção discutida para atender ao padrão anterior, *Temporal Relations*, combinando atividades temporais e agentes, o cancelamento de uma atividade pode ser feito pelo mesmo **Agent** que verifique o evento externo, caso seja satisfeito o algoritmo definido, interagindo com a *interface* COM+ do componente de execução do Wideflow.

Aqui é aconselhado utilizar um agente ligado a uma WorkTask sem assinalamento para que a expressividade do modelo seja explícita pela WorkTask.

Interleaved Parallel Routing:

Como no padrão *Deferred XOR*, quando um **Agent** interage com o *status* de um caso, é possível condicionar a transição de uma atividade ao estado de outra atividade no mesmo caso.

A expressividade é dada pela condição de guarda na transição de saída.

Milestone:

O Wideflow faz o controle o prazo de expiração de uma atividade, o que equívale à expiração da atividade por *time out* e disparo de aviso ao *owner*.

A criação de um *milestone* é feita com auxílio de uma atividade temporal. Esta construção também permite ativação de múltiplas instâncias se a atividade temporal, após ter sido executada satisfatoriamente for reativada por um agente em uma atividade posterior como discutido anteriormente junto com os padrões que tratam de múltipla instanciação.

A expiração de uma atividade não é expressa em modelo, é uma funcionalidade do motor de workflow. Já o *milestone* imposto numa atividade temporal torna-se um elemento explícito na no modelo.

Cancel Activity:

O WideFlow permite o cancelamento de uma atividade do caso por intermédio de um agente que perscrute o *status* do caso e interaja com o componente de execução, cancelando a atividade desejada.

Para tornar mais expressiva a construção e aproveitando que o WideFlow permite a definição de WorkTask sem assinalamento para atores, pode-se modelar uma atividade especificamente para acionar o agente com o algoritmo para cancelamento de outras atividades, preservando a expressividade da construção no modelo.

Cancel Case:

Este padrão é realizado com construções idênticas aquelas do padrão *Cancel Activity*.

A comparação com os padrões de workflow pode lançar bons esclarecimentos e boas pistas para utilização de workflow em larga escala, ao mesmo tempo em que é possível visualizar problemas típicos envolvendo construções avançadas, requisitos que para serem atendidos muitas vezes exigem soluções de contorno, prejudicando tanto a adequabilidade da aplicação quanto a expressividade dos modelos.

Apesar disto, da mesma forma que as técnicas de programação preservam a estrutura e legibilidade dos algoritmos codificados, a legibilidade dos modelos de fluxo de atividades deve ser preservada com boas técnicas de modelagem. Os detalhes de implementação das aplicações de workflow devem permanecer em segundo plano, como com os algoritmos dos agentes (**Agent**), mas a intenção do modelador tem de estar explícita nos elementos constituintes do diagrama, quando da criação de atividades, dos *joins* e *forks*, das expressões nas condições de guarda e dos objetos do fluxo dos dados. Assim, mesmo com possíveis lacunas na expressividade, os modelos apresentarão a semântica exata a ser realizada pela aplicação.

Outros padrões apresentados por van der Aalst são os relacionados com a sincronização entre casos ou até entre tipos de processos distintos (AALST, HOFSTEDE; BARROS, 2000a). Mecanismos de manipulação de mensagens assíncronas são discutidos para realizar estes padrões.

No trabalho que resultou na criação do WideFlow não foram abordados estes requisitos mas estas funcionalidades poderiam vir a ser suportadas futuramente pelo motor de workflow.

Reforçando a preocupação em trazer, de forma efetiva, workflow para os atuais ambientes de desenvolvimento de sistemas, muitas iniciativas têm surgido ultimamente, a maioria delas em esforços de comunidades de software livre, preocupadas em colocar os conceitos para análise, modelagem e desenvolvimento de workflow numa perspectiva mais próxima dos analistas e projetistas.

Eles têm publicado suas iniciativas sob o guarda chuva do SourceForge, uma organização que provê um site centralizado para que desenvolvedores cooperem no desenvolvimento de software *Open Source*, ou do consórcio ObjectWeb, com propósitos análogos. Estas soluções têm sido implementadas principalmente em J2EE, aproveitando o fato que, atualmente, esta tem sido a principal plataforma para o desenvolvimento e execução de aplicações distribuídas – um verdadeiro padrão de mercado.

Elas oferecem motores de workflow no *application server*, capazes de executar os processos definidos e publicados. Permitem extensão de funcionalidades e do comportamento do processo por intermédio de *hooks*, semelhantes aos agentes do WideFlow, e extensão de *interfaces* de cada componente.

Estas iniciativas assumem real importância quando experimentam propostas oriundas de pesquisas verificando sua adequação e viabilidade e contribuindo para a maturidade da área, realimentando o processo.

Neste campo, o JCP, *Java Community Process*, entidade responsável pela especificação dos padrões da tecnologia J2EE, iniciou, através de um *expert group*, a trabalhar na especificação de uma API java - o pacote `javax.process.*` - com a qual a plataforma J2EE deverá passar a possuir metadados, *interfaces* e um *runtime model* para definição e execução de processos no *container* do *application server*.

A especificação ainda está em fase inicial de elaboração e quando disponível colocará juntos, na mesma “paleta”, meios para construir peças de código de workflow e sistemas de informação no mesmo componente, numa mesma aplicação, de forma coesa e consistente – workflow passa a ser parte da arquitetura de aplicações mistas, onde os conceitos são utilizados de forma integrada.

Resumindo, como resultado do trabalho da criação do Wideflow e da modelagem de workflow com o diagrama de atividades e seus estereótipos no ambiente de engenharia de software da Procempa, pode-se sublinhar o seguinte:

- a) foi criado um motor de workflow executando a semântica de uma máquina de estados do Diagrama de Atividades da UML;
- b) foram delimitadas construções semânticas para o Diagrama de Atividades da UML capazes de suportar requisitos fundamentais em aplicações de workflow;
- c) o motor de workflow foi implementado num meta-modelo com o paradigma da orientação a objetos;
- d) a modelagem organizacional foi reduzida utilizando conceito de grupo de acesso, podendo ser estendida para suportar papéis no grupo;
- e) foram dominados os principais conceitos necessários para a criação de aplicações de workflow num processo de desenvolvimento corporativo;
- f) foram construídas aplicações de workflow integrando entre automação de processos e sistemas de informação sem invasão de domínios;
- g) foi feita a modelagem da aplicação tomando o motor de workflow como arquitetura para a aplicação, num componente distribuído com *interface* definida;
- h) foram implementadas aplicações de workflow de produção (NICOLAO, 1996) modeladas com uma extensão do diagrama de atividades e implementadas com o uso de um motor de workflow próprio;
- i) foi feita a modelagem de dados do workflow em abstrações de documentos aumentando o poder de expressão das operações possíveis em torno dos dados.

Com isto, acreditamos que os objetivos apontados na seção 1.5 foram alcançados, contribuindo para que workflow se popularize de forma efetiva e atenda de forma eficaz aos processos de automação discutidos em 1.3, projetos de desenvolvimento de aplicações de workflow a serem utilizados na administração pública de Porto Alegre.

Construir aplicações de workflow num processo de Engenharia de Software adequado, utilizando o motor de workflow como arquitetura para a aplicação, o que também permite a criação de aplicações mistas: sistemas de informação/workflow, num ciclo de vida de desenvolvimento integrado para atender às demandas por automação de processos de produção da administração do município de Porto Alegre é o resultado específico deste trabalho.

Apesar disto, é lícito estender as propostas aqui apresentadas para a resolução de problemas idênticos, mesmo que em cenários diversos, notadamente encontrados em grandes organizações com circulação de processos com informações complexas residentes em domínios diversos, desde sistemas de informação transacionais tradicionais até sistemas de informação georreferenciadas, quando os requisitos de flexibilidade arquitetural, simplificação do modelo organizacional, modelagem gráfica do fluxo e dos dados com diagramas de atividade e necessidade de produção

de aplicações de workflow por times de desenvolvedores de sistemas estejam presentes.

Mesmo tendo alcançado sucesso, apontamos aqui pistas para futuras pesquisas, testes ou implementações que poderiam aumentar ainda mais a adequação e a semântica do Diagrama de Atividades da UML, bem como definir uma abordagem para desenvolvimento de workflow num processo voltado para a arquitetura, utilizando outros diagramas da UML num ciclo de vida de desenvolvimento. Duas propostas são voltadas para aumentar o formalismo do Diagrama de Atividades da UML para suportar requisitos avançados de aplicações de workflow e outra proposta para utilizar outros diagramas da UML no processo de desenvolvimento de aplicações de workflow.

- a) definir semântica para que o diagrama de atividades realize construções de workflow de padrões baseados em XML, como BPEL4WS, XPDL e outros;
- b) desenvolver estudos para que o diagrama de atividades suporte outros padrões apontados por AALST, HOFSTEDE e BARROS (2000a), inclusive os relacionados com a sincronização entre casos ou entre tipos de processos distintos;
- c) propor técnicas para utilização de diagramas de interação da UML para a modelagem da colaboração entre os diversos objetos presentes numa aplicação de workflow para suporte ao desenvolvimento de workflow.
- d) elaborar Rational Rose Scripts para validar processos verificando caminhos (flow), condições de guarda, *deadlocks*, otimizações, etc.
- e) acompanhar a execução de processos automatizados e extrair medidas de satisfação, melhoria, produtividade, caminhos críticos no fluxo do processo, etc, de modo a mudar a expectativa dos patrocinadores, participantes, clientes, usuários e desenvolvedores quanto aos resultados de aplicação de TI na resolução de problemas de trabalho.
- f) comparação de custos com execução de processos anterior e posteriormente à automação e cotejamento com índices financeiro-econômicos para apoiar seleção e priorização de projetos de automação de processos.

REFERÊNCIAS

- ALONSO, G. et al. Functionality and Limitations of Current Workflow Management Systems. **IEEE Expert**, Los Alamitos, v. 12, n. 5, 1997.
- ARAUJO, A. DE; GOMIDE, C. F.; PETRILLO, F. **Metodologia e Projeto de Software Orientados a Objetos**. São Paulo: Ed. Érica, 2003.
- BARESI, L. et al. Workflow Development Methodology. In: GREFEN, P.; PERNICI, B.; POZZI, B.; SANCHEZ, G. (Ed.) **Database Support for Workflow Management: the WIDE Project**. Boston: Kluwer Academic, 1999. p. 47-94.
- BARROS, A.P.; HOFSTEDE, A.H.M. TER. Realizing the Full Potential of Workflow Modelling: a practical perspective. **International Journal of Information Technology**, [S.l.], v.3, n.2, p.59-86, Dec. 1997.
- BARROS, A.P.; HOFSTEDE, A.H.M. TER; PROPER, H.A. Essential Principles for Workflow Modelling Effectiveness. In: PACIFIC ASIA CONFERENCE ON INFORMATION SYSTEMS, PACIS, 3., 1997. **Proceedings...** Brisbane: [s.n.], 1997. p. 134-147.
- BARTHELMESS, P. **Sistemas de Workflow: Análise da área e proposta de modelo**. 1996. Dissertação (Mestrado) - IC-UNICAMP, Campinas.
- BARTHELMESS, P. Workflow Modeling. In: CYTED-RITOS INTERNATIONAL WORKSHOP ON GROUPWARE, CRIWG, 1., 1995, Lisboa. **Proceedings of CRIWG' 95**. Lisboa: Junta Nacional de Investigação Científica e Tecnológica, 1995.
- BASTOS, R.M.; RUIZ, D.D.A. Towards an Approach to Model Business Processes using Workflow Modeling Techniques in Production Systems. In: HAWAII INTERNATIONAL CONFERENCE SYSTEMS SCIENCES, 34. 2001. **Proceedings...** Maui: [s.n.], 2001.
- BASTOS, R.M.; RUIZ, D.D.A. Extending UML Activity Diagram for Workflow Modeling in Production Systems. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEMS SCIENCES, 35., 2002. **Proceedings...** Island of Hawaii: [s.n.], 2002.

- BECK, K. **Extreme Programming Explained**. [S.l.]: Addison- Wesley, 2000
- BOOCH, J.; RUMBAUGH, J.; JACOBSON, I. **UML : guia do usuário**. Rio de Janeiro: Campus, 1999.
- BRACHE, A. P.; RUMMLER, G. A. **Improving Performance** : how to manage the white space in the organization chart. San Francisco: Jossey-Bass, 1995.
- CASATI, F. et al. Conceptual Modeling of Workflow. In: OO-ER CONFERENCE, 1995. **Proceedings...** Gold Coast: [s.n.], 1995.
- CAVALCANTE, E. C. de M. **Modelagem e Implementação de Aplicações de Automação de Fluxo de Trabalho**: Proposta de Abordagem e Estudo de Caso. 2000. Dissertação (Mestrado) – Instituto de Informática, PUC, Campinas.
- COCKBURN, A. **Agile Software Developmen**. [S.l.]: Addison-Wesley, 2002.
- DUMAS, M.; HOFSTEDE, A.H.M. TER. UML Activity Diagrams as a Workflow Specification Language. In: INTERNATIONAL CONFERENCE ON THE UNIFIED MODELING LANGUAGE (UML). 2001. **Proceedings...** Toronto: Springer Verlag, 2001.
- ERICSON, M. **Activity Diagrams**: what they are and how to use them. [S.l.]: Rational Edge, 2002. Disponível em: <http://www.therationaledge.com/content/jan_02/t_activityDiagrams_me.html> . Acesso em: 10.jan.2003.
- ESHUIS, R.; WIERINGA, R. A Comparison of Petri Net and Activity Diagram Variants. In: INTERNATIONAL COLLOQUIUM ON PETRI NET TECHNOLOGIES FOR MODELLING COMMUNICATION BASED SYSTEMS, 2, 2001. **Proceedings...** Berlin: [s.n], 2001.
- ESHUIS, R.; WIERINGA, R. A real-time execution semantics for UML activity diagrams. In: FUNDAMENTAL ASPECTS OF SOFTWARE ENGINEERING FASE, 2001. **Proceedings...** Genova: Springer, 2001. p. 76-90 (Lecture Notes in Computer Science, v. 2029)
- ESHUIS, RIK; WIERINGA, ROEL. Verification Support for Workflow Design with UML Activity Graphs. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 24., 2002. **Proceedings...** New York: ACM, 2002. P. 166-176.

- FISCHER, L. **The Workflow Handbook**. Bruxelas: WfMC, 2001. 420p.
- GAMMA, E. et al. **Padrões de Projeto**. Porto Alegre: Bookman, 2000.
- GEORGAKOPOULOS, D. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. **ACM Distributed and Parallel Databases**, New York, n.3, p.119-153, Mar. 1996.
- GOMIDE, C. F. **Modelagem de Workflow**. 2002. Trabalho Individual (Mestrado em Sistemas de Informação) - Instituto de Informática, UFRGS, Porto Alegre.
- GONÇALVES, J. E. L. As Empresas são Grandes Coleções de Processos. **EAESP/FGV RAE – Revista de Administração de Empresas**, São Paulo, v.40, n.1, p. 6-19, jan./mar. 2000.
- GORDJIN, J.; AKKERMANS, H.; VLIET, H. VAN. Business Modelling is not Process Modelling. In: ER Workshop, 2000. **Proceedings...** Berlin: Springer-Verlag, 2000. p. 40-51.
- JACOBSON, I.; ERICSON, M.; JACOBSON, A. **The Object Advantage – Business Process Reengineering with Object Technology**. New York: ACM Press, 1994.
- JOOSTEN, S.; BRINKKEMPER, S. Fundamental Concepts for Workflow Automation in Practice. In: INTERNATIONAL CONFERENCE INFORMATION SYSTEMS DEVELOPMENT (ISD +96), 1996. **Proceedings...** Gdansk: [s.n.], 1996.
- KIRTLAND, M. **Projetando Soluções Baseadas em Componentes**. Rio de Janeiro: Campus, 1999.
- LEITE, J. C. **Anotações de Aula**. Disponível em: <<http://www.dimap.ufrn.br/~jair/>>. Acesso em jun. 2001.
- MANCUSO, F. **Modelagem de Empresas**. 1996. Trabalho Individual (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- MANOLESCU, D. A. **Micro-Workflow**: a workflow architecture supporting compositional object-oriented software development. 2001. Tese (Doutorado). – University of Illinois, Champaign.

MATSUDA, K. M. **Análise de Problemas para a Implantação de Workflow**. 2000. Dissertação (Mestrado) - IC-UNICAMP, Campinas.

MOLZ, K. W.; THOM, L. H.; SCHEIDT, N. **Estudo sobre modelagem e aplicação de sistemas de workflow**. Santa Cruz, RS: UNISC, 1999.

NICOLAO, M. **Um Estudo sobre Conceituação de Workflow**. 1996. Trabalho Individual (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS , Porto Alegre

OMG. **Unified Modeling Language Specification**, Versão 1.5. [S.l.], 2003.

PEROTTONI, R. et al. Sistemas de Informações: um estudo comparativo das características tradicionais às atuais. **Revista Eletrônica de Administração**, Porto Alegre, v.21, jul. 2001.

PIMENTA, M. A. Telecomunicações: alternativas locais para universalização do acesso. **Revista de Informática Pública**, Belo Horizonte, v. 05, p. 11-24, jan./jun. 2003.

PORTO ALEGRE. Prefeitura Municipal. Secretaria de Modernização Administrativa. **Carta Convocatória de Concorrência Pública Internacional**. 1999.

QUIXA SOLUTIONS. The 10 Myths about Workflow Automation (White Paper). Disponível em: <www.quixa.com/library/docs/QLSIB-ULT-104.pdf>. Acesso em: dez. 2002.

SILVEIRA, M. DE L. DA; BARROS, L. C. DA S.; PEREIRA, R. B. Acesso a sistemas legados através da Internet. **Revista de Informática Pública**, Belo Horizonte, v.02, p. 147-162, maio. 2000.

SILVEIRA, M. DE L. DA; BARROS, L. C. DA S.; PEREIRA, R. B. Rumo às tecnologias de objetos e de componentes distribuídos. **Bate Byte**, maio. 1999. Disponível em: <<http://www.pr.gov.br/batebyte/>>. Acesso em jun. 2002.

VAN DER AALST, W.M.P. et al. Advanced Workflow Patterns. In: INTERNATIONAL CONFERENCE ON COOPERATIVE INFORMATION SYSTEMS COOPIS 7., 2000. **Proceedings...** Berlin: Springer-Verlag, 2000. p. 18-29. (Lecture Notes in Computer Science, v. 1901).

VAN DER AALST, W.M.P. et al. **Workflow Patterns**. Eindhoven, Netherlands: Eindhoven University of Technology, 2000. (Beta Working Paper Series, WP 47).

VAN DER AALST, W.M.P. Petri-Net-Based Workflow Management Software. In: WORKSHOP ON WORKFLOW AND PROCESS AUTOMATION IN INFORMATION SYSTEMS, 1996. **Proceedings...**[S.l.:s.n.], 1996. p 114-118.

VAN DER AALST, W.M.P.;VAN HEE, K.M.;HOUBEN, G.J. Modelling and analysing work flow using a Petri-net based approach. In: WORKSHOP ON COMPUTER-SUPPORTED COOPERATIVE WORK,PETRI NETS AND RELATED FORMALISMS, 2., 1994. **Proceedings...** [Sl.:s.n.], 1994. p.31 .50.

WORKFLOW MANAGEMENT COALITION. **Interface 1: Process Definition Interchange**. Disponível em: <<http://www.wfmc.org>>. Acesso em: jun. 2002.

WORKFLOW MANAGEMENT COALITION. **The Workflow Reference Model**, Disponível em: <<http://www.wfmc.org>>. Acesso em: jun. 2002.

WORLD WIDE WEB CONSORTIUM. **Extensible Markup Language (XML) 1.0 (Second Edition)**. W3C Recommendation. oct. 2000. Disponível em: <<http://www.w3c.org/XML/>>. Acesso em: jun. 2003.