

**Universidade Federal de Santa Catarina  
Programa de Pós-graduação em  
Engenharia de Produção**

**SISTEMAS *WORKFLOW* EM PROCESSOS  
EMPRESARIAIS, BASEADOS NO  
CONHECIMENTO, APLICANDO TÉCNICAS  
DA INTELIGÊNCIA ARTIFICIAL**

Dissertação de Mestrado

**LY FREITAS FILHO**

Florianópolis

**2002**

**LY FREITAS FILHO**

**SISTEMAS *WORKFLOW* EM PROCESSOS EMPRESARIAIS,  
BASEADOS NO CONHECIMENTO, APLICANDO TÉCNICAS  
DA INTELIGÊNCIA ARTIFICIAL**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção com Ênfase em Informática da Universidade Federal de Santa Catarina como requisito parcial para obtenção do título de Mestre em Engenharia de Produção.

Florianópolis

2002

Ly Freitas Filho

**SISTEMAS *WORKFLOW* EM PROCESSOS EMPRESARIAIS,  
BASEADOS NO CONHECIMENTO, APLICANDO TÉCNICAS  
DA INTELIGÊNCIA ARTIFICIAL**

Esta dissertação foi julgada e aprovada para a obtenção do título de  
**Mestre em Engenharia de Produção no Programa de Pós-Graduação  
em Engenharia de Produção com Ênfase em Informática** da  
Universidade Federal de Santa Catarina.

Florianópolis, 15 de Outubro de 2002

Prof. Edson Pacheco Paladini, Dr.  
Coordenador do Curso

**BANCA EXAMINADORA**

Prof. Rogério Cid Bastos, Dr.  
Orientador

Prof<sup>a</sup>. Ana Paula Soares Fernandes, Dr<sup>a</sup>.

Prof. Álvaro Guillermo Rojas Lezana, Dr.

## *Agradecimentos*

À Universidade Federal de Santa Catarina;  
Ao orientador Prof. Rogério Cid Bastos;  
Às professoras Édis Mafra Lapolli e Clarice Amaral;  
Aos professores do Curso de Pós graduação.

A Deus e a todos que, direta ou indiretamente,  
contribuíram para a realização desta dissertação,  
em especial, a Thereza que esteve sempre ao meu lado,  
a Lya que estará sempre,  
a Agência Nacional de Energia Elétrica e  
a Universidade Católica de Brasília.

## Resumo

FILHO, Ly Freitas. **Sistemas *Workflow* em processos empresariais, baseados no conhecimento, aplicando técnicas da inteligência Artificial.** Florianópolis, 2002. 161f. Dissertação (Mestrado em Engenharia de Produção) - Programa de Pós-graduação em Engenharia de Produção com Ênfase em Informática, UFSC, 2002.

Este trabalho é um estudo comparativo da aplicação de técnicas de inteligência artificial, em sistemas *Workflow*, ou fluxo de trabalho automatizado. Os fluxos de trabalho enfocados são processos empresariais que geram serviços e que apóiam os processos produtivos. A aplicação de alguma das técnicas de inteligência artificial neste tipo de sistema, é definida como sistema *Workflow* baseado no conhecimento.

**PALAVRAS CHAVES:** *Workflow*, inteligência artificial, reengenharia de processos, Agentes Inteligentes.

## Abstract

FILHO, Ly Freitas. **Sistemas *Workflow* em processos empresariais, baseados no conhecimento, aplicando técnicas da inteligência Artificial.** Florianópolis, 2002. 161f. Dissertação (Mestrado em Engenharia de Produção)- Programa de Pós-graduação em Engenharia de Produção com Ênfase em Informática, UFSC, 2002.

This work is a comparative study of the application of artificial intelligence techniques, in systems Workflow, or flow of automated work. The work flows focused are business processes that you/they generate services and that you/they support the productive processes. The application of some of the techniques of artificial intelligence in this system type, it is defined as system Workflow based on the knowledge.

**KEYWORDS:** Workflow, Artificial Intelligence, reengineering of processes, Intelligent Agents.

## Sumário

	Página
Lista de Figuras.....	viii
Lista de Reduções.....	x
<b>1. INTRODUÇÃO</b>	
1.1 - Motivação e justificativa .....	1
1.2 - Objetivos .....	4
1.3 - Descrição dos capítulos .....	4
<b>2. SISTEMAS WORKFLOW</b>	
2.1 - Reengenharia e ISO9000 .....	6
2.2 - Define processos .....	7
2.3- Fundamentos do sistema <i>Workflow</i> .....	11
2.4 - Define <i>Group ware</i> .....	13
2.5 - Define <i>Workgroup</i> .....	15
2.6 - Tipos ou níveis do sistema <i>Workflow</i> .....	16
2.7- Estrutura do sistema <i>Workflow</i> .....	21
2.8 - Elementos primários do sistema <i>Workflow</i> – os três Rs .....	24
2.9 - Natureza do sistema <i>Workflow</i> .....	31
2.10 - Documentos inteligentes .....	34
2.11 - INTERNET/INTRANET/EXTRANET.....	34
2.12 - Ambiente do sistema <i>Workflow</i> .....	35
2.13 - Implantação do sistema <i>Workflow</i> .....	41
2.14 - Padrões para o sistema <i>Workflow</i> .....	48
2.15 - Benefícios do uso do sistema <i>Workflow</i> .....	50
2.16 - Segmentação de mercado .....	50
2.17 - Banco de dados temporais.....	51
<b>3. INTELIGÊNCIA ARTIFICIAL.....</b>	<b>62</b>
3.1 - Define Inteligência Artificial .....	62
3.2 - Define Engenharia do Conhecimento .....	73
3.3 - Lógica Nebulosa – Fuzzy Logic.....	78
3.4 - Redes Neurais Artificiais .....	81
3.5- Agentes Inteligentes .....	92
<b>4. APLICAÇÃO .....</b>	<b>109</b>
4.1 - Aplicações correlatas .....	109
4.2 - Aplicação proposta- Ouvidoria Inteligente .....	134
<b>5. CONCLUSÕES E RECOMENDAÇÕES.....</b>	<b>137</b>
<b>6. BIBLIOGRAFIA .....</b>	<b>139</b>
<b>ANEXO – DOCUMENTAÇÃO DO SISTEMA DE GESTÃO DA OUVIDORIA.....</b>	<b>143</b>

## Lista de Figuras

	Página
Figura 2.1 – Processo e suas divisões.....	9
Figura 2.2 – Sistema de <i>Workflow</i> para escritório .....	13
Figura 2.3 – <i>Groupware</i> e suas tecnologias. ....	14
Figura 2.4 – Estrutura de <i>Workflow</i> .....	22
Figura 2.5 – Alguns tipos de rotas de <i>Workflow</i> .....	30
Figura 2.6 – Ciclo de desenvolvimento do <i>Workflow</i> .....	41
Figura 2.7 – Modelo de referência <i>Workflow</i> .....	49
Figura 2.8 – Gráfico de classificação de <i>software</i> de <i>Workflow</i> .....	51
Figura 2.9 - Banco de dados bitemporais.....	57
Figura 3.1 – Arquitetura típica de um sistema baseado em conhecimento. ..	71
Figura 3.2 – Rede semântica de conhecimentos sobre equipamentos eletrônicos.....	75
Figura 3.3 – Árvore correspondente ao problema de jarras de água.....	76
Figura 3.4 – Representação de conhecimento sobre animais, na forma de árvore.....	77
Figura 3.5 – Ilustração das formas de encadeamento para o caso de busca.....	77
Figura 3.6 – Função indicadora do conjunto A.....	78
Figura 3.7 – Função e pertinência para um conjunto nebuloso A. ....	79
Figura 3.8 – Estrutura de um controlador nebuloso.....	80
Figura 3.9 – Estrutura geral de uma unidade computacional. ....	81
Figura 3.10 – Algumas topologias para rede neuronal artificial. ....	85
Figura 3.11 – Apredizado supervisionado.....	87
Figura 3.12 – Representação de um agente genérico.....	92
Figura 3.13 – Parte do mapeamento ideral para um problema - Raiz Quadrada. ....	95
Figura 3.14 – O agente esqueleto. ....	97
Figura 3.15 – Diagrama esquemático de um agente de reflexo simples .....	97
Figura 3.16 – Diagrama esquemático de um agente de reflexo com estado interno .....	101
Figura 3.17 – Diagrama esquemático de um agente com metas explícitas	103
Figura 3.18 – Diagrama esquemático de um agente baseado em utilidade .....	104
Figura 3.19 – Programa simulador de um ambiente básico .....	107
Figura 3.20 – Programa simulador de um ambiente básico com medida de desempenho .....	108
Figura 4.1 – Visão de um sistema complexo legal.....	111
Figura 4.2 – Plano de uma condição de sucesso .....	112
Figura 4.3 – Representação formal de um processo de negócio.....	114
Figura 4.4 – Arquitetura do sistema EULE2 .....	116
Figura 4.5 – Arquitetura de uma memória organizacional.....	117

Figura 4.6 – Integração do sistema <i>Workflow</i> com o sistema EULE2.....	119
Figura 4.7 – Detalhamento da integração do sistema <i>Workflow</i> com o sistema EULE2.....	119
Figura 4.8 – Arquitetura de componentes de um empreendimento .....	122
Figura 4.9 – Relacionamento entre ontologias.....	125
Figura 4.10 – Visão do monitor de execução do KPM.....	128
Figura 4.11 – Arquitetura do SWIM .....	131
Figura 4.12 – Modelo conceitual do gerenciamento <i>Workflow</i> orientado a objeto .....	133
Figura 4.13 – Fluxo de Trabalho do Sistema <i>Workflow</i> para o SGO .....	136

## Lista de Reduções

### Siglas

<b>ANEEL</b>	Agência Nacional de Energia Elétrica
<b>AP</b>	Papéis do <i>Workflow</i>
<b>API</b>	<i>Application Program Interface</i>
<b>DMBS</b>	<i>Database Management System</i> ou SGBD
<b>DDE</b>	<i>Dynamic Data Exchange</i>
<b>DF</b>	Documentos e Formulários
<b>EDI</b>	<i>Eletronic Data Interchange</i>
<b>FTP</b>	<i>File Transfer Protocol</i>
<b>GUI</b>	<i>Graphical User Interface</i>
<b>IA</b>	Inteligência Artificial
<b>ISR</b>	Information Surveillance and Reconnaissance
<b>JIT</b>	<i>Just in Time</i>
<b>KBMS</b>	Sistema de Gerenciamento de Base de Conhecimento
<b>KQML</b>	Linguagem de Manipulação e Consulta de Conhecimento
<b>KPM</b>	Knowledge-Based Process Management
<b>MAPI</b>	<i>Messaging Application Program Interface</i>
<b>NET</b>	Internet, Intranet e Extranet
<b>ODA</b>	<i>Office Document Architecture</i>
<b>ODBC</b>	<i>Open DataBase Connectivity</i>
<b>ODMA</b>	<i>Open Document management API</i>
<b>OLE</b>	<i>Object Linking and Embedding</i>
<b>OO</b>	Orientado à Objeto
<b>OP</b>	Objetivo do Procedimento
<b>RNA</b>	Redes Neurais Artificiais
<b>RP</b>	Rota do Procedimento
<b>SGBD</b>	Sistema Gerenciador de Banco de Dados
<b>SGO</b>	Sistema de Gestão da Ouvidoria
<b>SI</b>	Sistema de Informação
<b>SNH</b>	Sistema Neuronal Híbrido
<b>SQL</b>	<i>Structure Query Language</i>
<b>SWAP</b>	<i>Simple Workflow Access Protocol</i>
<b>SWIM</b>	Smart <i>Workflow</i> for ISR Management
<b>TBA</b>	<i>Time-based Analysis</i>
<b>TI</b>	Tecnologia da Informação
<b>TQM</b>	<i>Total Quality Management</i>
<b>VM</b>	<i>Virtual Machine</i>
<b>WfMC</b>	<i>Workflow Management Coalition</i>
<b>WWW</b>	<i>World Wide Web</i>

# 1. INTRODUÇÃO

## 1.1 - Motivação e justificativa

Processos são a estrutura pela qual uma organização faz o necessário para produzir valor para os seus clientes. Segundo Davenport (1994), o estudo de processos surgiu, apenas, com os primeiros especialistas da área de manufatura.

A administração, por tempos, enfatizou os processos produtivos fabris em detrimento aos processos que geram serviços e aos que apóiam os próprios processos produtivos, também denominados **Processos Empresariais**. Segundo Porter (1999), estudos vieram a comprovar que, atualmente, esses processos são os principais diferenciais de competitividade entre as organizações.

Ao longo da década de 1980, a maioria das empresas concentrou seus esforços principais na correção e no aperfeiçoamento de seus processos de produção. Segundo Harrington (1993), somente na última década do milênio, a administração voltou-se para os processos empresariais, verificando que o processo de produção responde por apenas 10% do valor de um produto médio comum. A área de serviços, que emprega a maioria da população economicamente ativa, é constituída, apenas, de processos empresariais.

Segundo Sculley (2001), não se pode deixar de constatar que a informatização nos processos produtivos fabris levou a um aumento não

mensurável de produtividade, acima de todos os investimentos que tinha origem nos sistemas de informações.

Os princípios do fenômeno da produtividade fabril podem e devem ser aplicados nos processos empresariais, evidentemente, com as adaptações necessárias e desde que sejam repetitivos. Ainda, segundo Sculley (1998), a disciplina da automação fabril, como os métodos: *Total Quality Management - TQM*, *Eletronic Data Interchange - EDI*, *Just in Time - JIT* e *Time-based Analysis – TBA*, pode ser inerente, a todos os processos empresariais repetitivos conhecidos, a partir da implantação de sistemas de fluxo de trabalho automatizado, também conhecidos como sistemas *Workflow*.

Em todas as empresas ocorrem, diariamente, centenas de processos empresariais que, em sua maioria, são repetitivos. São, exatamente, nesses processos empresariais repetitivos que vai-se estudar a aplicação dos controles de modo semelhante aos usados nos processos de manufatura, com ênfase em suas principais características: fluxo de trabalho, resultados desejados, ou seja, sua eficácia. O alvo é a minimização dos recursos empregados, isto é, sua eficiência, tempo do ciclo e custo.

Para estudar sistemas *Workflow*, é essencial falar-se de processos, pois, em síntese, sistema *Workflow* nada mais é do que a automatização de processos.

Segundo Cruz (1998), um sistema *Workflow* baseado no conhecimento, desenvolvido com técnicas estatísticas, heurísticas, inteligência artificial e usando os mesmos princípios de reconhecimento de

padrões com que são construídas as redes neuronais, pode ser a solução para as freqüentes mudanças que um fluxo deva sofrer para acompanhar a dinamicidade do processo de negócio de qualquer empresa, estabelecendo, assim, uma nova geração de sistemas *Workflow*, quebrando mais um paradigma na gestão de processos.

O interesse em sistemas *Workflow* foi constatado, segundo White (1994), devido ao reconhecimento dos profissionais de que a demanda nos próximos anos serão pela automatização dos processos da atividade de negócio e não, somente, de tarefas individuais.

Nesse tipo de sistema, cada atividade é gerenciada de forma isolada, quanto aos tempos e métodos com os quais ela tem que ser realizada e, de forma coletiva, quanto ao processo como um todo.

Segundo Jackson e Twaddle (1997), o sistema *Workflow* tem como um de seus objetivos principais transformar os processos em ativos, agregando todos os sistemas e *softwares* já existentes e disponibilizando outros necessários para a perfeita implementação do fluxo de trabalho automatizado. Para tanto, será necessária a utilização das modernas tecnologias de redes de computadores, além das técnicas de inteligência artificial, interfaces gráficas para usuários - GUI, banco de dados temporais e as tecnologias NET – Internet, Intranet e Extranet, que mudaram a forma de fazer negócios, entre outras.

## **1.2 - Objetivos**

### **1.2.1 - Objetivo geral**

Comparação das técnicas de inteligência artificial que podem ser aplicadas em um sistema de fluxo de trabalho automatizado, *Workflow*, transformando-o no tipo baseado no conhecimento.

### **1.2.2 – Objetivos específicos**

- Estudo de modelos de processos empresariais repetitivos com as técnicas de processos produtivos fabris;
- Estudo da implementação de um modelo de processo empresarial aperfeiçoado em um *software* de fluxo de trabalho automatizado, *Workflow*;
- estudo da aplicação do conceito de processos pró-ativos;
- estudo da aplicação em processos pró-ativos das diversas técnicas existentes de Inteligência Artificial – IA, em especial o uso de Agentes Inteligentes; e
- estudo dos benefícios proporcionados na implementação em processos empresariais.

## **1.3 - Descrição dos capítulos**

Esta seção pretende fornecer ao leitor uma visão geral da estruturação adotada para a elaboração deste trabalho.

Além desta introdução, este trabalho está dividido em mais cinco partes, quais sejam:

- revisão bibliográfica sobre sistemas *Workflow*;
- revisão bibliográfica sobre Inteligência Artificial;
- aplicação e discussão; e
- conclusões e recomendações.

A primeira parte deste trabalho apresenta capítulos de revisão bibliográfica necessários para a melhor compreensão desta pesquisa, bem como da metodologia adotada como solução. Nesta primeira parte estão incluídos os capítulos 2 e 3.

No capítulo 2, são apresentados conceitos fundamentais sobre *Workflow*. Iniciando-se por um estudo de seu histórico, passando pela análise de seus componentes básicos, metodologias aplicadas, tendências, entre outros.

No capítulo 3, são apresentados conceitos fundamentais sobre Inteligência Artificial - IA e suas alternativas para atendimento à pesquisa.

A segunda parte do trabalho, formada pelos capítulos 4 e 5, apresenta a aplicação dos conceitos e técnicas pesquisadas e as conclusões e recomendações advindas da análise dos resultados obtidos.

## **2. SISTEMAS *WORKFLOW***

### **2.1 - Reengenharia e ISO9000**

Na década de 1980, Hammer (1994) definiu o termo reengenharia para designar uma nova abordagem de implantação de sistemas diferente da usada até então. Em síntese, nessa nova abordagem preconizava-se que, antes de tentar-se organizar um processo por meio do emprego de tecnologias da informação, perpetuando a desordem ou tentando engessá-la pelo uso de algum novo sistema ou dispositivo, deve-se, literalmente, destruir, ou melhor seria dizer, abandonar a forma como se vinha operando determinado processo e recriá-lo melhor, para só então implantar uma nova Tecnologia da Informação – TI.

É evidente que, no caso de criação de um novo processo, já seria inerente em sua construção, a racionalização e estruturação do mesmo.

As idéias de Hammer (1994) visavam a resolver os problemas decorrentes das tentativas de reorganização e racionalização de atividades pelo emprego da informática, além de integrar inúmeras abordagens organizacionais dentro de uma única metodologia.

Segundo Cruz (1998), antes da reengenharia, os processos só eram abordados como tal por um grupo muito restrito de especialistas da área de manufatura. A maioria dos profissionais, principalmente os da área de informática, falava de rotinas.

Ainda, conforme Cruz (1998), o surgimento de inúmeros fatores, tanto estruturais como conjunturais, em termos mundiais, fez com que a reengenharia fosse aplicada de forma errônea, intensificando os discursos de seus opositores.

A técnica de Hammer previa ainda que o processo que a reengenharia recriou precisava continuar sendo aperfeiçoado, a exemplo dos processos fabris, através do programa de melhoria contínua.

Após a reengenharia, outro grande fator de disseminação do conhecimento sobre processos foi a publicação da Norma ISO9000, que tem como pressuposto básico a necessidade da empresa organizar-se por meio do conhecimento e da documentação dos processos produtivos e empresariais, para que cada atividade seja realizada sempre da mesma forma, segundo as especificações que foram testadas e documentadas em seu manual da qualidade.

## **2.2 – Define processos**

“Processo é uma seqüência de atividades que consomem recursos e produzem um bem ou serviço.”, conforme Hronec (1994, página 54).

“Processo é um conjunto de atividades estruturadas e medidas destinadas a resultar num produto especificado para um determinado cliente ou mercado.”, conforme Davenport (1994, página 6).

Resumindo os vários conceitos estudados, pode-se melhor definir processo como o conjunto de atividades inter-relacionadas que tem por finalidade processar matéria-prima para produzir bens e serviços que serão disponibilizados para clientes, ou a maneira pela qual realiza-se alguma operação, segundo determinadas normas, métricas e técnicas.

Nas organizações hierárquico-funcionais, os processos não possuem visibilidade de forma a facilitar seu entendimento. Observa-se que, a partir da década de 1990, algumas organizações, inclusive brasileiras, passaram a estruturar-se através de processos, denominadas organizações estruturadas através de gestão por processos.

No tipo de organização hierárquico-funcional, os processos são também denominados processos multifuncionais, devido a serem compostos de atividades que pertencem a várias funções. É importante observar que mesmo que o processo não esteja claro, visível, organizado, simplificado, racionalizado e documentado, ele existe e, como tal, assume-se como verdadeiro.

Com o advento da Norma ISO9000, estruturar e documentar processos estão entre as principais necessidades de qualquer tipo de empresa. Essas necessidades são, atualmente, a base para o sucesso de qualquer empreendimento, inclusive, conforme propaga a norma citada, até mais do que o produto gerado, pois este, quando processado de forma desorganizada, pode acarretar prejuízos. Sendo o processo bem estruturado, a execução deste tenta minimizar os esforços despendidos na consecução de seus propósitos.

Segundo Kobelius (1997), os critérios para melhor avaliar um processo devem ser: velocidade, custo, exatidão, qualidade, satisfação do cliente e flexibilidade.

### 2.2.1 – Processos e suas divisões

Ainda, segundo Kobelius (1997), com a finalidade de melhor entender o processo, o mesmo deve ser dividido através de refinamentos sucessivos, conforme Figura 2.1..

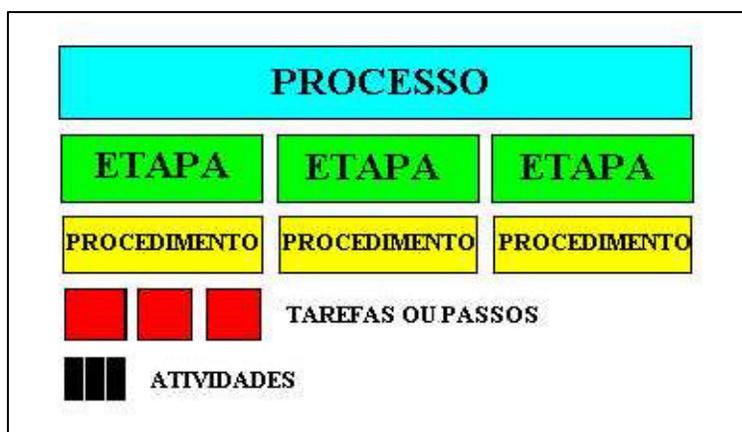


Figura 2.1 – Processo e suas divisões.

Assim, pode-se definir que:

- **Etapa ou Fase** é uma subdivisão ou parte de um processo, contendo as condições de início e término e o método a ser aplicado;
- **Procedimento** é o método e suas respectivas regras que indicam como, quando e com que recursos deve ser executada alguma etapa ou fase;

- **Tarefa** (passos) é a decomposição do procedimento; e
- **Atividade** é a atribuição trabalho específico, com a respectiva determinação de responsabilidade. Seu enfoque principal é a eficiência e sua execução é denominada de ação.

### 2.2.2 – Atividade

“Atividade é a unidade de trabalho executada por um único responsável, que tem condições determinadas de início e fim presumíveis”, conforme Kobelius (1997).

Ainda, segundo Kobelius (1997), a atividade pode ser classificada como:

- **Principal:** são as que têm participação direta na criação do bem ou serviço que é objeto do processo. Os agrupamentos de atividades para a criação de bem ou serviço são: Logística Interna, Operações, Logística Externa, Vendas e Marketing e Serviços; e
- **Suporte** (marginais ou secundárias): São as que não estão diretamente envolvidas com a produção do bem ou serviço que a empresa vende. As atividades de suporte existem para permitir que as atividades principais possam ser executadas, além de providenciarem todas as condições de operacionalidade necessárias. Compõem-se, geralmente, de quatro grupos: Infraestrutura, Recursos Humanos, Tecnologia, Compras e Qualidade.

### **2.3 - Fundamentos do sistema *Workflow***

Segundo pesquisa apresentada em Cruz (1998), a maioria dos processos são passivos, assim como os sistemas de informações – SI que tenham sido desenvolvidos para suportá-los. Ambos cumprem suas funções primordiais, somente se cada funcionário quiser; como quiser; e quando quiser, ou seja, existem em estado latente.

Se cada funcionário fizer a parte que lhe compete, no tempo e no espaço corretos, dentro dos limites do procedimento da atividade, o trabalho, provavelmente, se tornará eficaz e lucrativo.

Segundo Cruz (1998), um processo é passivo quando as atividades que o compõem não dispõem de tecnologia que as transformem de ações que são puxadas em ações que puxam o trabalho. As atividades passivas esperam para ser executadas, enquanto as ativas cobram de seu responsável a atuação. Atividades passivas esperam que o produto de sua operação seja enviado à próxima, ou próximas atividades; as ativas têm inteligência para o despacharem, tão logo esteja pronto. Atividades passivas não sabem para quem enviar o produto de sua execução; as ativas sabem. A mesma tecnologia que permite transformar as atividades de passivas em ativas pode executar uma série de outras atividades e possibilita que o processo seja gerenciado de modo muito mais eficaz.

Segundo Kobelius (1997), automatização é o componente que irá transformar processos, atividades e procedimentos em ativos e, dessa forma, levá-los a assumir uma postura de cobrar em vez de serem utilizados, de puxar em vez de serem empurrados, de fazer acontecer em vez de ficarem esperando acontecer.

Uma das ferramentas que permite a automatização de forma ativa é o Sistema *Workflow*, traduzindo: sistema de fluxo de trabalho automatizado. Entretanto, por uma questão de internacionalização da tecnologia na maioria dos trabalhos relativos a esta tecnologia, padronizou-se denominá-lo simplesmente de *Workflow*.

Podem-se verificar abaixo algumas das definições de *Workflow*

“*Workflow* é um conjunto de ferramentas que possibilita análise proativa, compressão e automação de atividades e tarefas baseadas em informação”, conforme Koulopoulos (1995).

“*Workflow* é a tecnologia que ajuda a automatizar políticas e procedimentos numa organização”, conforme Khoshafian (1995).

“*Workflow* é a tecnologia que possibilita automatizar processos, racionalizando-os e potencializando-os por meio de dois componentes implícitos: organização e tecnologia”, conforme Cruz (1998).

“*Workflow* é o fluxo de controle e informação num processo de negócio”, conforme Kobelius (1997).

“*Workflow* é a automação da rota de documentos e tarefas”, conforme Kobelius (1997).

O uso de sistemas *Workflow*, segundo Jackson e Twaddle (1997), será mais intenso nos escritórios, onde deverão ser integrados com os sistemas legados, com a internet, entre outros, conforme apresentado na Figura 2.2.

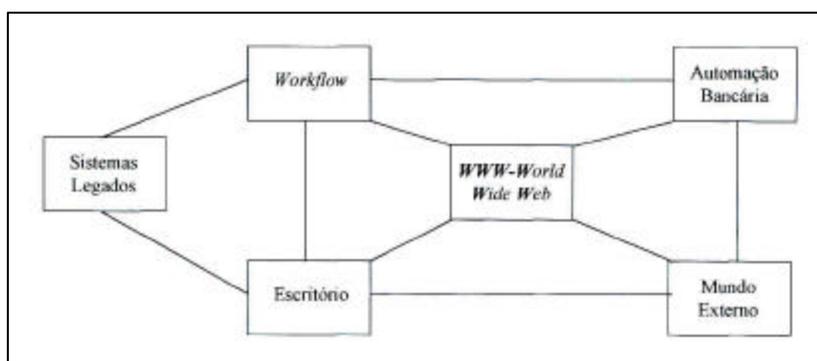


Figura 2.2 – Sistema de *Workflow* para escritório

Fonte: Jackson e Twaddle (1997, página 9)

## 2.4 - Define *Groupware*

*Groupware* é qualquer sistema computadorizado que permita que grupos de pessoas trabalhem de forma cooperativa, a fim de atingirem um objetivo comum.

*Groupware* pode ser caracterizado como um guarda-chuva para suportar o trabalho cooperativo, sob o qual estão inúmeras outras tecnologias que têm como idéia principal permitir às pessoas trabalharem uma para as outras, cada uma fazendo com que não só a sua própria atividade seja executada com sucesso, mas também, com que todas as atividades que compõem o processo o sejam.

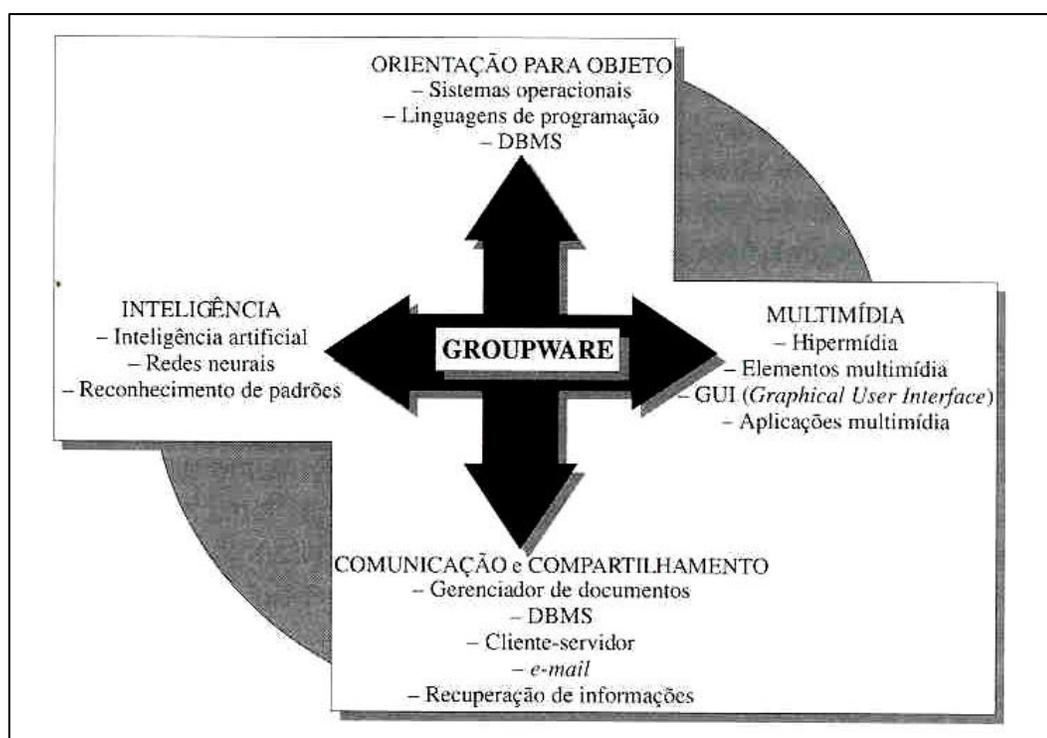


Figura 2.3 – *Groupware* e suas tecnologias.

Fonte: Cruz (1998, página 85)

A figura anterior mostra como, genericamente, os produtos *Groupware* estão agrupados e divididos:

- aplicações baseadas em documentos e formulários como e-mail, gerenciadores de imagens de documentos, SGBD, *Workflow* entre outros;
- aplicações baseadas em grandes volumes de dados e transações como SGBD, sistema de recuperação textual, redes neurais, entre outros; e
- aplicações baseadas em comunicação organizacional como calendário, agendas eletrônicas, videoconferência, entre outros.

## **2.5 - Define *Workgroup***

O foco principal está na informação que será processada. A informação em si mesma não tem capacidade de organizar e automatizar um fluxo de trabalho, por mais simples que o fluxo possa ser, ou seja, os processos continuam passivos.

No *Workflow*, a ênfase é dada ao processo. A importância do processo está no fato de ele ser o meio pelo qual a informação será processada. Com a construção das rotas da informação, finalmente, é possível transformar o processo de passivo em ativo.

Segundo Kobelius (1997), as atividades de organizar, documentar e simplificar processos têm que fazer parte de qualquer projeto de implantação de *Workflow*.

## 2.6 - Tipos ou níveis do sistema *Workflow*

Conforme Kobelius (1997) e Curz (1998), observam-se várias classificações de tipologia para *Workflow*. São os seguintes tipos ou níveis de *Workflow*

*Ad hoc*

Administrativo

Produção ou transação

Orientado a objeto

Baseado no conhecimento.

Comercialmente, observa-se que a maioria dos tipos implementados são misturas de alguns dos tipos relacionados.

### 2.6.1 – Nível 1: *Ad hoc*

Segundo Aurélio Buarque de Holanda, “*ad hoc* é uma expressão latina que significa ‘para isto’ ou ‘para este caso’.”

O *Workflow ad hoc* é aquele criado para ser usado dinamicamente por grupos de trabalho, cujos participantes necessitem executar procedimentos para cada tipo de documento processado de trabalho. Por este motivo, é impossível estabelecer um fluxo de trabalho padronizado, pois este fluxo raramente se repete.

É o mais elementar tipo de *Workflow*, o que não significa que sua implantação seja elementar, mas observa-se que esse tipo não possibilita a transformação de processos passivos em processos ativos, mas, apenas, tirá-los do estado de desorganização comum, a fluxos de trabalho baseados apenas em papéis físicos.

Um exemplo de *Workflow ad hoc* é o uso de correio eletrônico ou *e-mail*, cuja tela principal do módulo de correio é um *browser* tipo Netscape ou Exchange.

Entretanto, existem sistemas bastante sofisticados voltados, apenas, à implantação de *Workflow* tipo *ad-hoc*. Esses sistemas permitem tratar e armazenar vários tipos de documentos como: imagem, som, texto, hipertextos e realidade virtual.

### **2.6.2 – Nível 2: Administrativo**

O tipo ou nível administrativo tem características de sistemas de correio eletrônico, mas, com algumas especificidades, o que o torna ideal para o tratamento de documentos e formulários que servem de suporte para rotinas que, embora repetitivas e aparentemente sem complexidade, precisam ser realizadas corretamente. Exemplos desse tipo de *Workflows* são rotinas de aprovação de despesas, controle de gastos com viagens, aprovação de ordens de compras e todo um conjunto de necessidades que genericamente chamamos de rotinas administrativas.

Muitos *softwares* de *e-mail* que estão no mercado atualmente já incorporam funcionalidades mais avançadas de tratamento de rotinas administrativas. Dentro desse conjunto estão:

- Ferramentas para criação de formulários simplificados;
- Ferramentas para rotear formulários;
- Interação com os formulários; e
- Ferramentas para notificações de avisos e prazos.

### **2.6.3 – Nível 3: Produção ou transação**

O *Workflow* do tipo produção ou transação, geralmente, envolve grandes quantidades de dados, muitas regras de negócio e recursos financeiros em grande escala. Essa mistura de elementos críticos dá um caráter especial a esse tipo de *Workflows* faz com que seu desenvolvimento e implantação tenha que ser cuidadosamente planejado.

Os dados tratados por um sistema de *Workflow* de produção ou transação têm duas origens: uma no próprio fluxo, por exemplo, uma solicitação de atendimento técnico; e outra nos bancos de dados que suportam as aplicações, por exemplo, os dados do cliente que solicitou um atendimento.

Tais sistemas são muito sensíveis, por se referirem ao próprio negócio da empresa e, geralmente, envolvem muitos de seus departamentos.

Na maioria das vezes, há a necessidade de uma ferramenta de auditoria no *Workflow* que possa tornar seguro o processo, especialmente

por possuir algum relacionamento com dados financeiros. Cada atividade, cada procedimento, cada documento, enfim, deve ser auditado para impedir fraudes e possibilitar que a qualquer momento os responsáveis possam verificar todo o andamento do processo. Exemplo de fluxo: aprovação de empréstimo.

#### 2.6.4 – Nível 4: Orientado a Objeto

São as versões mais sofisticadas dos sistemas de *Workflow*, pois incorporam a tecnologia OO – Orientado a Objeto, que surgiu na década de 1980, como uma evolução da tecnologia estruturada.

O *Workflow* orientado a objeto abrange os dados ou atributos do objeto e as instruções sobre qual método de processamento sobre esses dados, ou seja, o comportamento a ser apresentado.

Esse tipo de *Workflow* tem características específicas, muitas, advindas da Análise Orientada a Objetos, como as que citamos a seguir:

- **Classe:** é a união de campos de dados e outros campos que contêm funções para operar os campos de dados, ou seja, um objeto é uma variável de uma classe, segundo Mizrahi (1994);
- **Encapsulamento:** o objeto oculta a implementação, expondo, apenas a interface, segundo Votre (1998);
- **Polimorfismo:** permite que uma linha vertical de derivações sintonize as funções de forma que a função chamada seja a

mais adequada ao objeto em foco na chamada, baseado no recurso de função virtual, segundo Votre (1998);

- **Herança:** possibilidade de criar novos procedimentos baseados em regras e propriedades herdadas de procedimentos já existentes, além de poder incorporar características específicas; e
- **Integridade Referencial:** Característica que garante que qualquer informação adquirida ou passada por herança será igual à informação que tiver dado origem às informações descendentes.

Com o *Workflow* OO, permite-se a convivência de várias versões de fluxos de trabalho e regras diferentes para um mesmo objeto.

O *Workflow* de produção e o orientado a objeto são semelhantes quanto ao tratamento de volumes de dados, a diferença é, somente, o uso da tecnologia OO pelo último.

### **2.6.5 – Nível 5: Baseado no conhecimento**

Esse tipo de *Workflow* deve aprender com seus próprios erros e acertos, ou seja, possuir tecnologia para ir além da execução pura e simples das regras preestabelecidas e incorporar exceções a seus procedimentos.

A Inteligência Artificial - IA é uma das tecnologias que permitirá a sistemas *Workflow* aprenderem a absorver conhecimento, tornando-os adaptativos.

Um sistema *Workflow* desenvolvido com técnicas estatísticas, heurísticas, com alguns princípios de reconhecimento com que são construídas as redes neuronais, e, em especial, a aplicação de agentes inteligentes, pode ser a solução para as freqüentes mudanças que um fluxo deva sofrer para acompanhar a dinâmica do processo de negócio de qualquer empresa.

Essa tecnologia ainda não está disponível comercialmente, embora se saiba que vários fabricantes de *software* trabalham na criação de protótipos que podem vir a ser uma nova geração de sistemas.

## **2.7- Estrutura do sistema *Workflow***

Segundo Kobelius (1997), todo sistema *Workflow* deve ser estruturado sobre uma arquitetura de cinco elementos principais. Cada um desses elementos tem diferentes funções que se complementam para formar o fluxo de trabalho automatizado.

Os cinco grupos de um sistema de *Workflows* são:

Processo

Instâncias ou casos

Pastas

Papéis, regras e caminhos

Documentos

A Figura 2.4 apresenta graficamente como esses grupos interagem.

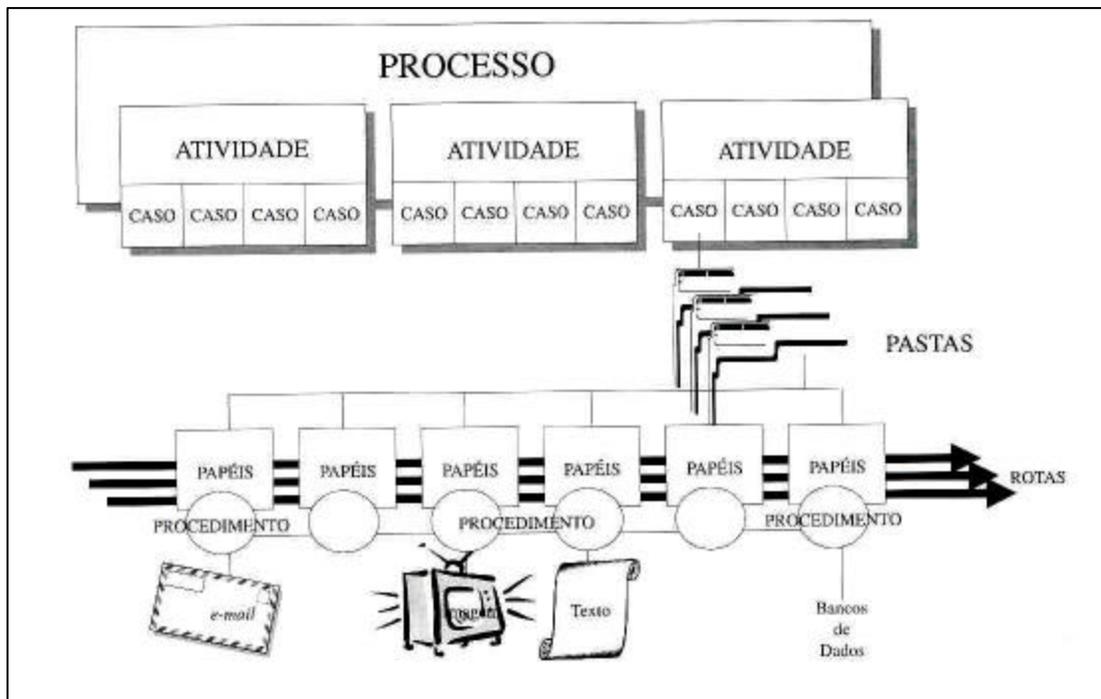


Figura 2.4 – Estrutura de *Workflow*.

Fonte: Cruz (1998, página 94)

### **2.7.1 – Instâncias ou casos**

Conforme pode ser observado na Figura 2.4, processo está dividido em atividades, devido a não obrigatoriedade da existência de subprocessos. A instância ou caso é uma ocorrência individual dentro de atividade presente no processo produtivo. Cada vez que o sistema *Workflow* é acionado para processar um novo documento, cria-se, neste momento, um novo caso ou um novo objeto, se for objetos.

A quantidade de instâncias ou casos que podem ocorrer numa atividade está diretamente ligada ao tempo de ciclo da atividade, por exemplo, a execução de uma ordem de produção.

### **2.7.2 – Pastas**

Para organizar *e-mails*, dados, informações, textos, voz e imagem foi criado o nível que identifica pasta, que é a reunião lógica de documentos. Isso permite o acesso fácil e rápida localização.

### **2.7.3 – Documento**

Documento é o nível mais elementar de uma estrutura *Workflow*

Documentos são coleções de dados, informações e regras de processamento colocadas numa pasta onde serão usados por uma instância ou caso dentro de um processo. As informações integrantes de um

documento podem ser sobre o formato, o processamento e a apresentação dos respectivos dados.

## **2.8 - Elementos primários do sistema *Workflow* – os três Rs**

Segundo Kobelius (1997) e Curz (1998), os três Rs – Roles, Rules and Routes, traduzindo: Papéis; Regras; e Caminhos, são considerados os elementos primários do Sistema *Workflow*. Um sistema de *Workflow* precisa ser montado sobre as definições de quem faz o quê, de que forma e quando, e quais os caminhos que levam e trazem os pacotes de dados e informações que dão vida a uma instância ou caso.

Essas definições têm como paradigma o planejamento intitulado 5QRC, ou seja, as respostas as seguintes perguntas por quê?, o quê?, quem?, quando?, qual lugar?, recursos? e como?.

### **2.8.1 – Papéis ou *roles***

O detalhe mais importante no *Workflow* é “Quem faz o quê?”, como o papel de uma personagem em peça de teatro.

Papel ou *role* é o conjunto de características e habilidades necessárias para executar determinada tarefa ou tarefas pertencentes a uma atividade.

O papel ou *role* sempre é o mesmo, as responsabilidades são sempre as mesmas, como um personagem existente naquela peça de teatro, o que muda são os autores, mas os papéis não.

Assim, como no caso dos atores, cada participante do *Workflow* deve ser treinado cuidadosamente. Esse treinamento deve dar atenção especial ao aprendizado do papel que cada um vai desempenhar, as responsabilidades inerentes ao papel, bem como aos resultados esperados e que serão medidos.

Existem dois tipos de papel ou *role*: usuário e função.

O **papel usuário** é pessoal e intransferível e serve para identificar todos os usuários que tenham acesso a um sistema *Workflow*. Quando o papel usuário deixar de existir por demissão, promoção ou outra ocorrência qualquer, deve ser desativado.

O **papel função** é o que a atividade deve desempenhar dentro da cadeia de valores. É obrigatória a definição de cada um dos papéis função existentes no fluxo de trabalho de forma particularizada. É necessário, também, definir um papel genérico e associá-lo a todos os usuários do sistema.

Para o *Workflow*, o papel função desempenhado numa atividade tem uma série de atributos que devem ser definidos como forma de se construir as camadas necessárias ao fluxo de trabalho do processo. É primordial que o sistema *Workflow* deva permitir que um papel função possa ser alterado de papel usuário, sem alterar seus atributos de papel função.

Seus atributos mínimos são: nome do papel ou *role*, posição organizacional, nível administrativo, papel supervisor e direitos de acesso.

- **Posição Organizacional:** deve descrever a localização do papel ou *role* dentro da empresa, que pode ser um grupo de trabalho, um departamento, um projeto ou até outra empresa;
- **Nível Administrativo:** indica o nível em que o papel ou *role* está definido na organização. Isto permite criar um organograma dinâmico da estrutura do fluxo de trabalho;
- **Papel Supervisor:** é a definição de qual papel supervisiona o papel estudado; e
- **Os Direitos de Acesso:** são um conjunto de atributos que definem com que tipo de permissão o papel pode acessar o *Workflow*.

Papéis ou *roles* são sempre associados a outros elementos, mais, especificamente, aos caminhos e às regras.

### 2.8.2 – Regras ou *Rules*

As regras são elementos que definem de que forma os dados que trafegam no fluxo de trabalho devem ser processados, roteados e controlados pelo sistema *Workflow*.

As regras definem, basicamente, quais informações devem transitar pelo fluxo de trabalho e sob quais condições. Cada documento enviado contém informações que serão usadas por quem as receber. Associadas ao documento, podem existir regras que especificam com clareza e exatidão a

operação ou processamento do documento, quais as atividades que devem recebê-lo, quais as rotas a seguir e a quais cuidados especiais, se houver, o documento deve ser submetido.

A garantia de que a implantação do sistema terá sucesso encontra-se no conhecimento e organização prévios das regras que sustentam o fluxo de informações no sistema *Workflow*. Quanto mais detalhada for a descrição das regras que suportam cada documento, mais seguro será seu processamento, diminuindo a chance para que um papel ou *role* qualquer comprometa a agregação de valores no produto ou serviço.

Existe um tipo de dado num sistema de *Workflow* que, por suas características, se diferencia dos demais dados existentes num fluxo de trabalho. Esse dado leva o nome genérico de *dado relevante*. Um *dado relevante*, para *Workflow*, é toda e qualquer informação usada no fluxo de trabalho para avaliar condições e determinar qual a rota a ser seguida, qual processamento deve ser executado, e como um documento deve ser manuseado. Os componentes de uma regra, como início, tempo, execução e segurança são *dados relevantes* para qualquer sistema de *Workflow*.

### **2.8.3 – Caminhos ou *Routes***

O terceiro R, rotas ou caminhos, para *Workflow*, é o controle de movimentação exercido sobre os documentos. Os comandos do elemento rota controlam como os documentos se movem de um ponto a outro, dentro do fluxo de trabalho. Este elemento é tão importante quanto os outros dois, papéis e regras, pois, entre outras funcionalidades, garante que o documento

chegue ao destino sempre que uma instância ou caso seja iniciado no *Workflow*.

Rota é o caminho lógico que, definido sob regras específicas, tem a função de transferir a informação dentro do processo, ligando as atividades associadas ao fluxo de trabalho.

Existem quatro tipos de Rotas ou Caminhos e cada um desses tipos tem um propósito que não deve ser confundido como simples sofisticação da ação de transferência:

- serial;
- paralelo;
- condicional; e
- conclusão.

O caminho serve para levar alguém, ou alguma coisa, de um lugar para outro. Às vezes, de um ponto a outro existem vários caminhos e nos decidimos por um deles sob determinadas condições. Para *Workflow*, essas condições, também, se aplicam e isso faz com que uma informação percorra um determinado caminho sob o controle de determinadas condições.

### **2.8.3.1 – Caminho serial**

É o Caminho linear e direto entre os passos, não permitindo variações. Nesse tipo de caminho, cada atividade tem, apenas, uma atividade anterior

e, apenas, uma atividade posterior. Além disso, cada atividade deve ser completada antes que o trabalho seja enviado a atividade seguinte.

### **2.8.3.2 – Caminho paralelo**

Se dá quando um grupo de passos pode ocorrer ao mesmo tempo e tem o mesmo passo anterior e o mesmo passo seguinte. Esse caminho propicia a agilização do processo, pois permite caminhos independentes entre si, que podem ocorrer ao mesmo tempo e, logo após, serem reunidos num ponto, denominado Ponto de Encontro. Esse Ponto de Encontro deve, também, ter a função de armazenamento até que todos os passos paralelos sejam concretizados e, somente, após isso, ser enviado ao próximo passo.

### **2.8.3.3 – Caminho condicional**

Ocorre quando múltiplas rotas podem ser usadas e a escolha é feita por meio de determinada condição. Esse tipo de rota é determinado dinamicamente por eventos que ocorrem no processo, à medida que as atividades são executadas.

### **2.8.3.4 – Caminho de conclusão**

É um caminho que tem apenas um motivo específico. É o caminho que um objeto de informação ou um processo toma para concluir uma operação. Existem dois tipos de rotas de conclusão:

- um tipo endereça a ação a  $n$  atividades; e
- o outro endereça a ação a, apenas, uma atividade.

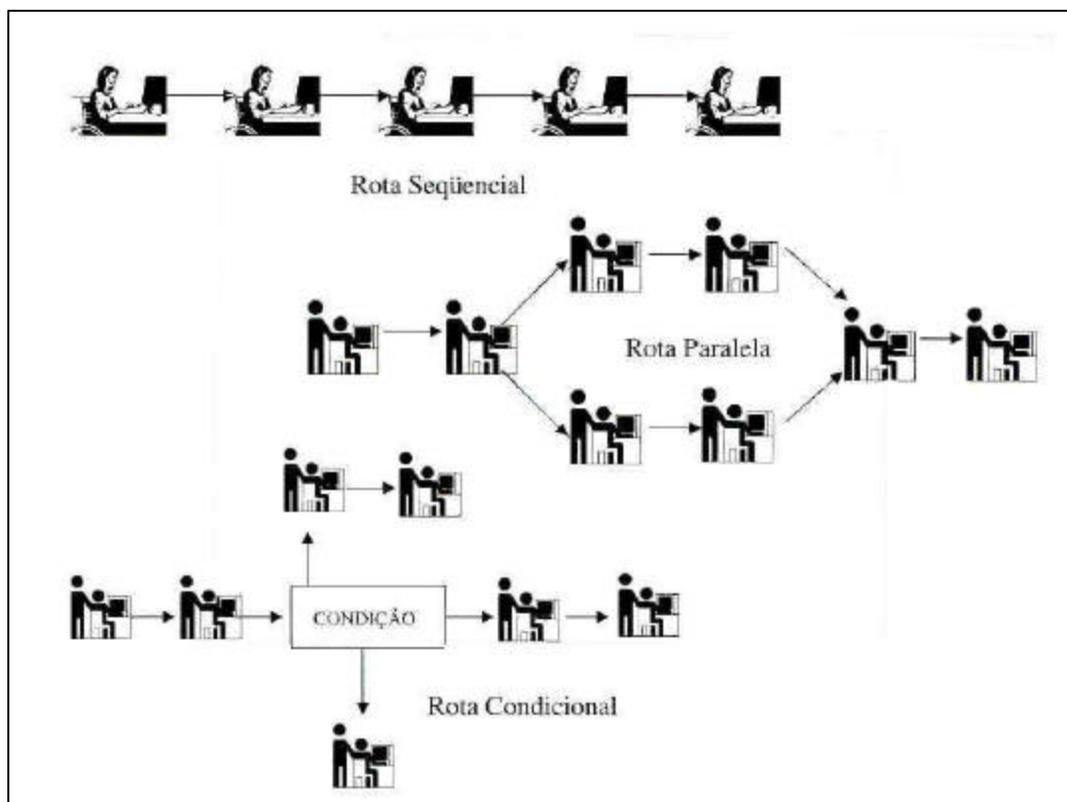


Figura 2.5 – Alguns tipos de rotas de *Workflow*

É claro que num ambiente automatizado não existirá somente um tipo de caminho definido, nem os caminhos serão todos de um só tipo. Num sistema *Workflow* é comum existirem múltiplos tipos de caminhos.

A possibilidade de gerenciar inúmeros documentos e caminhos ao mesmo tempo faz dos atuais sistemas de *Workflow* o que há de mais avançado em automatização de processos. Entretanto, qualquer que seja o sistema *Workflow* adotado na empresa, é preciso que alguns cuidados sejam tomados quando se pensa em automatizar fluxos de trabalho.

Em termos de rotas ou caminhos, a principal preocupação é que o sistema que se pretende implantar na empresa permita que a informação flua

através de variadas formas de rotas ou caminhos, pois, disso dependem as inúmeras possibilidades de processamento das informações existentes.

## **2.9 - Natureza do sistema *Workflow***

Segundo Cruz (1998), são componentes intrínsecos da natureza de um sistema *Workflow* a duração, o evento e o objeto.

Uma das abordagens com que pode-se tratar *Workflow* é como esse tipo de sistema pode tratar os eventos tendo como base o tempo necessário para executá-los.

Isso significa que pode-se fazer com que eventos sejam tratados num tempo otimizado com base na definição de regras específicas para processamento, intrínsecas a cada evento, ou seja, cada evento, ao processar um objeto, deve fazê-lo na forma otimizada quanto ao tempo de execução, dados os parâmetros previamente programados para processar cada objeto.

### **2.9.1 – Duração**

A duração ou tempo de processamento de um objeto possui, sempre, dois componentes:

- **Duração Necessária:** é o intervalo no qual um passo deve ser executado. Esse tempo deve ser estimado em função das condições de executabilidade do passo e do tempo total necessário para que o processo possa completar um ciclo. A executabilidade de um passo está diretamente ligada às

condições que a suportam, tais como: tecnologia, instalações e conhecimento por parte dos recursos humanos encarregado de executá-los; e

- **Duração Viável:** É o tempo no qual, efetivamente, o passo é executado. Esse tempo torna-se conhecido por meio de registro histórico da média de tempos realizados em determinado período.

A duração é a principal componente da natureza do ambiente *Workflow*, pois, os outros dois, eventos e objetos, têm suas existências reguladas por ela.

### 2.9.2 – Evento

O evento é o menor componente de um sistema *Workflow*. É, também, denominado de ocorrência de algum fato.

Um evento pode ser de um dos cinco tipos abaixo:

- **Inicialização:** É o evento que dá início a instância ou caso. Esse evento pode ser simples, como quando um usuário se conecta ao sistema, ou complexo, quando dispara inúmeros outros eventos ao mesmo tempo;
- **Notificação:** É um evento que existe em decorrência de outro evento de *Workflow*. É, geralmente, uma mensagem eletrônica que indica a ocorrência de fim de processo, por exemplo; embora possa disparar outro evento;

- **Interação ou negociação:** É a execução repetitiva de um objeto por várias tarefas, seguindo as mesmas regras todo o tempo. Um documento que necessite ser assinado três vezes é um bom exemplo;
- **Dependência:** É um evento de espera. Ocorre, por exemplo, Quando um objeto aguarda por outro objeto que é pré-requisito de seu processamento; e
- **Finalização:** É o último evento de um objeto dentro de um sistema *Workflow*

### 2.9.3 – Objeto

Um objeto em um sistema de *Workflow* é a reunião, em um só componente, de dados e funções, como regras e procedimentos. As regras e procedimentos servem para que a informação da instância ou caso possa ser processada de acordo com o que foi estabelecido em sua criação.

Esse tipo de abordagem serve, entre outras coisas, para fazer com que todos os atores, responsáveis por um papel ou *role*, evitem os possíveis desvios provocados pela confusão de entendimento de sua função.

Um sistema *Workflow* atual deve dispor de ferramentas que permitam criar, gerenciar, modificar objetos, além de permitir relacioná-lo com outros objetos, transportando entre eles dados e funções.

## 2.10 - Documentos inteligentes

Ainda segundo Cruz (1998), a caracterização de documento em inteligente ocorre quando os mesmos permitem a incorporação de vários tipos de objetos.. Essa utilização é a mais simples de um sistema de *Workflow* e serve, principalmente, para aplicações de automação de escritório.

Segundo Kobelius (1997), a incorporação de objetos em um documento deve fazer uso de padrões como o *Office Document Architecture* – ODA e de recursos como *Dynamic Data Exchange* - DDE, *Object Linking and Embedding* - OLE e de outros que permitam em um documento; anexar outro documento, imagens, sons, dados e informações.

A automação de escritório pode ser considerada como uma das primeiras tentativas de automatização de fluxo de trabalho, pois baseia-se na integração do trabalho das pessoas, trazendo como consequência um aumento de produtividade.

## 2.11 - INTERNET/INTRANET/EXTRANET

Os sistemas atuais de *Workflow* não podem deixar de interagir com a tecnologia Internet, Intranet e Extranet, conhecida como NET. Assim, os sistemas de *Workflow* devem usar como meio para operacionalizar o fluxo de trabalho, o correio eletrônico ou *e-mail*, pois, através desse, se interage com usuários ou clientes no mundo.

Através da rede mundial pode-se aplicar a última palavra em trabalho cooperativo, que atende pelo nome de *equipe virtual*. Esse tipo de equipe

permite organizar o trabalho sem a necessidade de que todos os seus integrantes estejam presentes no mesmo ambiente físico, ou seja, na sede de uma empresa.

A automação dos processos de negócios na *World Wide Web* – WWW traz inúmeros benefícios, como por exemplo: a redução do ciclo de tempo do processo; aumento da produtividade; melhoria do relacionamento com o cliente, empregado, parceiros; redução nos custos; e a habilidade de operação global com redução nos custos de infra-estrutura.

Observa-se que sistemas de *Workflow*, que não permitam sua integração com as tecnologias NET, não atendem a realidade vigente da Tecnologia da Informação – TI.

## **2.12 - Ambiente do sistema *Workflow***

Segundo Kobelius ( 1997), um sistema *Workflow* não pode ser implementado isoladamente, necessitando de tecnologias que participam como infra-estrutura de suporte.

O conjunto dessas tecnologias é denominado Ambiente *Workflow* ou *Workflow Environment*, que é dividido em três componentes: o *hardware*, o *software* e o Sistema de *Workflow*.

Caso se apresente restrições ou uso de tecnologias ultrapassadas no conjunto *hardware-software-Workflow*, proporcionalmente menores serão as funcionalidades disponíveis, podendo refletir no grau e na profundidade dos ganhos de produtividade do fluxo de trabalho automatizado.

### **2.12.1 – *Hardware e Software***

O componente *hardware* ou plataforma de *hardware*, que suportará o sistema *Workflow*, divide-se em duas partes: computador e rede.

A base para a existência de um ambiente *Workflow* é a rede, sobre a qual esse ambiente será implantado. A arquitetura de rede mais usual, a cliente-servidor, permite a integração entre estações cliente e diversos servidores, entre eles o de Internet, o de *e-mail* e o de Banco de dados, podendo, inclusive, dependendo do sistemas de *Workflow*, ser necessário um servidor específico para este.

O componente *software* deve, também, acompanhar o avanço da tecnologia, seja no tocante ao sistema operacional, bem como em relação aos Sistemas Gerenciadores de Banco de Dados – SGBD.

Diante da possibilidade de tratamento de documentos inteligentes, poderá ser necessário a integração do sistema *Workflow* com *softwares* específicos para processamento de objetos anexados ao documento.

### **2.12.2 – Sistema de *Workflow***

Um sistema de *Workflow* deve atender a algumas especificações básicas, conforme relacionado abaixo:

#### **2.12.2.1 – Parte cliente**

A parte cliente de um sistema *Workflow* deve ser composta, ao menos, pelos seguintes módulos:

- **Módulo com ferramentas para desenho do *Workflow*:** Módulo que permite a criação dos modelos que serão usados no fluxo de trabalho. Esse módulo deve ser gráfico;
- **Módulo com ferramentas para ativação do *Workflow*:** Esse módulo tem a responsabilidade de ativar os modelos utilizados no fluxo de trabalho, fazendo com que suas funcionalidades sejam executadas ao longo do fluxo de trabalho; e
- **Módulo com ferramentas para roteamento e verificação do *Workflow*:** Esse é o módulo responsável por duas atividades fundamentais para a automatização de qualquer fluxo de trabalho. O gerenciamento das rotas e a verificação do andamento dos trabalhos permitem um gerenciamento proativo de todas as ocorrências, casos ou instâncias, tratadas dentro do *Workflow*.

#### 2.12.2.2 – Parte servidor

A parte servidor de um sistema *Workflow* deve ser composta, ao menos, pelos seguintes módulos:

- **Módulo de gerenciamento dos serviços *Workflow*:** Todas as funcionalidades de um sistema *Workflow* são gerenciadas por esse módulo. Ativação e desativação de modelos, roteamento, verificação, manutenção de grupo e usuários, de regras, de rotas são as atividades executadas por este módulo;

- **Módulo de gerenciamento dos serviços de banco de dados:** Esse módulo é responsável pela utilização concorrente do ambiente transacional; e
- **Módulo de transporte de mensagens e comunicações:** Toda comunicação e o transporte de mensagens é gerenciada por esse módulo. Não somente as comunicações internas, mas, também, entre essas pessoas e o mundo exterior, por meio das facilidades de telecomunicações, em especial pela tecnologia NET.

### 2.12.2.3 – Parte visível

Baseado em um sistema Workflow de nível 3, ou seja, Produção/transação, por ser o mais oferecido no mercado segundo Cruz (1998), a parte visível deve ser, ao menos composta, pelos seguintes módulos:

- **Caixa de Ferramentas:** É um conjunto de ferramentas de desenvolvimento de aplicações que permite automatizar fluxos de trabalho, construir formulários eletrônicos e prover extensão ao código de uma linguagem Visual. A mais importante é a que permite modelar os procedimentos;
- **Modelagem de Procedimentos:** Sendo procedimento um conjunto inter-relacionado de tarefas, esta ferramenta deve permitir modelar graficamente fluxos de trabalho, definindo, inclusive, as etapas, ações, tarefas e execuções de eventos

automáticos. Deve permitir, também, o desenho e a construção de formulários eletrônicos, definindo consulta SQL aos sistemas de informação;

- **Administração:** É, geralmente, um produto que permite realizar análise dos dados históricos dos procedimentos e estabelecer medições de rendimento da organização, mediante a monitorização do processo em tempo real. Torna possível a geração de estatísticas e gráficos. É importante e imprescindível para o apoio à gestão do processo;
- **Linguagem de programação SQL-GUI:** É a capacidade que um sistema de *Workflow* tem de interagir com qualquer aplicação existente na empresa sem estar amarrado a ela. Isso possibilita que a aplicação possa ser modificada sem a obrigação da modificação no sistema de *Workflow* ou vice-versa. Esse módulo é uma ferramenta para facilitar a construção interativa de consultas a bases de dados para poder tratá-los como elementos dentro dos formulários do *Workflow*. Essa linguagem transformou os principais comandos da linguagem SQL em elementos gráficos, contendo todos os componentes característicos das aplicações estilo Windows;
- **Agenda:** Módulo que alguns sistemas de *Workflow* têm para permitir aos usuários agendar compromissos e atividades de forma integrada. Esse módulo usa os princípios do trabalho

cooperativo para permitir o compartilhamento dos recursos existentes na empresa;

- **Módulo Internet/e-mail:** Os sistemas mais sofisticados permitem tanto integrar os processos de negócio de uma empresa com o mundo e seus clientes, como realizar o roteamento integral entre as atividades que estejam fora e dentro da Internet com segurança e eficiência; e
- **Módulo Usuário:** Deve ser um ambiente operacional baseado em *Workflow*, com uma poderosa capacidade de integração, de fácil manejo, pensado e orientado para usuário final. Sua interface gráfica permite a qualquer usuário desenvolver seu trabalho de forma amigável e segura, sem precisar ser um especialista, já que cada componente da aplicação corresponde, claramente, a sua função dentro do processo produtivo. Esse módulo deve prover elementos que permitam uma administração integral da informação por meio de pastas, pelas quais o usuário pode visualizar e realizar distintas ações sobre as mensagens e objetos que lhe são enviados, através de ícones, que facilitam as execuções das etapas.

## 2.13 - Implantação do sistema *Workflow*

Segundo Kobelius (1997), o *Workflow* baseia-se em dois alicerces: a organização do fluxo do trabalho e os *softwares* que suportarão todo o processo. O sistema *Workflow*, ao ser implantado, possibilita a reorganização do fluxo do trabalho, racionalizando na busca da eficácia e eficiência do processo.

### 2.13.1 – Ciclo de desenvolvimento do *Workflow*

O Ciclo de desenvolvimento do *Workflow* tem cinco etapas:

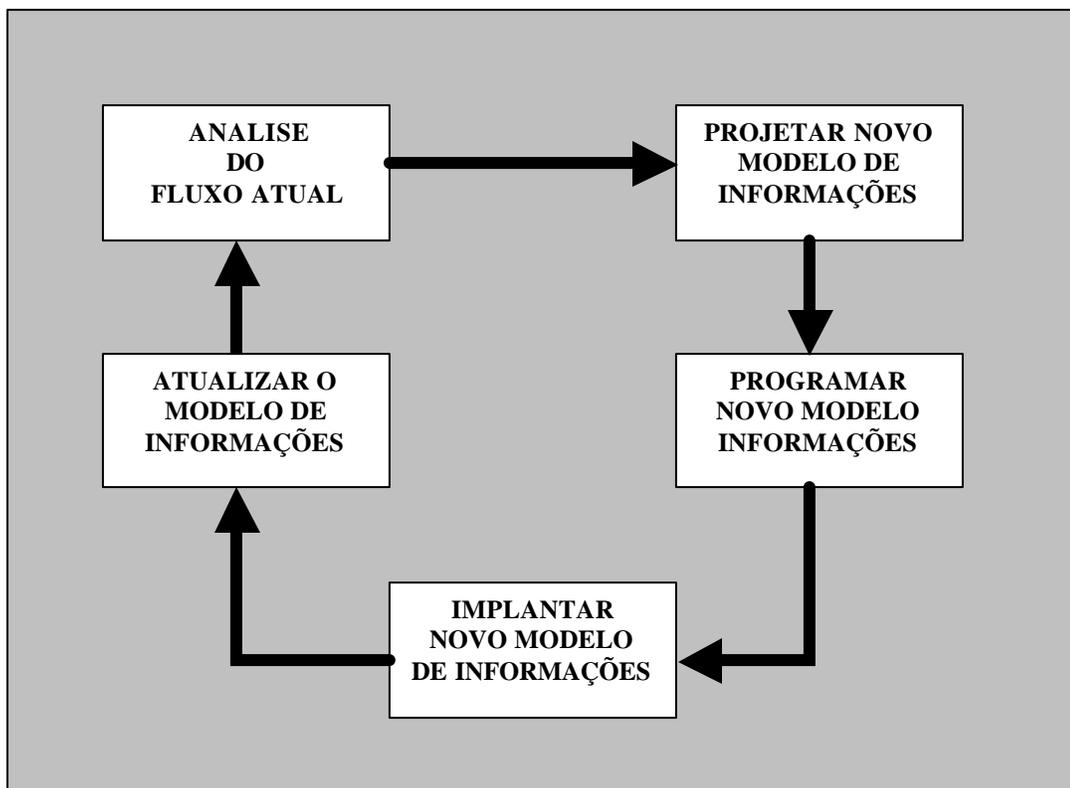


Figura 2.6 – Ciclo de desenvolvimento do *Workflow*

### **2.13.1.1 – 1ª Etapa: Análise do fluxo de trabalho atual**

A partir do modelo atual, pode-se projetar um novo modelo de informação. São três os grandes momentos desta análise:

- levantamento de como o processo é executado hoje;
- a determinação das melhorias que precisam ser implementadas; e
- a elaboração do modelo conceitual do novo processo.

### **2.13.1.2 – 2ª Etapa: Projetar o modelo de informações do fluxo de trabalho que se quer automatizar**

Após a reorganização que o fluxo possa ter sofrido, pode-se seguir levantando os seguintes itens:

- escolha do modelo de informação mais adequado; e
- determinação da composição deste modelo de informação.

Não se pode esquecer que o modelo de informação é o conjunto de objetos que faz parte de um fluxo de trabalho e que tem por função dar-lhe existência por meio da automatização dos procedimentos.

O modelo da informação deverá descrever a estrutura e os atributos dos vários tipos de objetos, tais como formulários, documentos, pastas e tudo que se relacione com eles, envolvidos no procedimento que a empresa quer automatizar.

As etapas que devem ser cuidadosamente levantadas:

- **Início:** Sob quais condições o processo se inicia;

- **Programação:** Descrição dos tempos máximo e mínimo permitidos para cada atividade;
- **Pré-condição:** Estabelecimento da forma em que determinado ator ou *role* pode iniciar a atividade correta;
- **Execução:** Descrição dos tipos de ferramentas, aplicações, metodologias e técnicas usadas para processar um item de trabalho;
- **Notificação:** descrição das condições sob as quais as pessoas serão notificadas sobre um evento do processo;
- **Pós-condição:** Estabelecimento do momento em que determinada pessoa pode completar certa atividade e o que deve ocorrer quando ela for terminada;
- **Segurança:** Descrição de quem está autorizado a participar do processo, que funções será autorizado a executar e que informações poderão ser manuseadas;
- **Auditoria:** Descrição de quais eventos serão auditados e em que nível de detalhamento; e
- **Término:** descrição das condições em que o processo deverá terminar ou ser interrompido.

Para projetar o novo modelo de informação, o sistema de *Workflow* necessitará da definição dos seguintes elementos:

- **Objetivo do Procedimento (OP):** O objetivo geral é a gestão do fluxo de trabalho por meio de seus documentos, a fim de torná-lo mais ágil, seguro, eficiente e eficaz;
- **Papéis do Workflow (AP):** Definição de quem vai participar do ambiente *Workflow*, isto é, quais serão os atores que executarão as tarefas necessárias para que cada etapa do procedimento seja cumprida. Os papéis podem ser grupais ou individuais. É por meio dessa definição que o sistema *Workflow* poderá distribuir corretamente as tarefas que devem ser executadas;
- **Rota do Procedimento (RP):** Definição das rotas que os documentos, formulários, instruções, a informação, enfim, devem percorrer para que o procedimento tenha vida. Como já foi verificado, existem vários tipos de rotas, seqüencial, paralela e condicional. Observa-se, que devem ser definidos, neste momento, o controle de filas de entrada e de saída de documentos;
- **Documentos e Formulários (DF):** No ambiente *Workflow*, documentos e formulários são veículos para os dados do caso, ou instância, que devem ser processados para que o trabalho seja realizado. Esses elementos podem conter vários tipos de dados, texto, imagem, planilhas e som;
- **Mensagens:** Que serão utilizadas nos mais diversos níveis do *Workflow*, e

- **Pastas:** As pastas necessárias aos mais diversos momentos do fluxo de trabalho. De uso pessoal, de classificação, associadas a um papel, e informativas.

### 2.13.1.3 – 3ª Etapa: Programar o modelo de informação

Tal programação deverá definir o detalhamento de cada um dos elementos contidos nele.

Essa etapa pode variar com o *software* escolhido, entretanto, algumas características são encontradas em todos os produtos disponíveis no mercado. Um procedimento, a ser automatizado, é constituído dos seguintes níveis e elementos:

- **Propriedades globais do procedimento:** nome, descrição, calendário, prazos, pastas de controle, definição de ícones, tempo de duração, entre outros dados;
- **Definição das etapas que compõem os níveis do procedimento:** nome, descrição, formulário, parâmetros de controle, calendário, duração, entre outros dados;
- **Ações que devem ser executadas em cada uma das etapas:** Representação através de ícone com suas respectivas propriedades. Eventos que, associados às etapas, fazem fluir os documentos através do fluxo de trabalho;
- **Programação das propriedades do evento.** Cada etapa poderá conter um ou mais eventos que devem ter as seguintes características: nome, pasta, destino e parâmetros;

- **Definição das autorizações:** Definição de níveis de autorizações, tanto para modificar procedimentos, papéis como para criar mensagens;
- **Pastas que conterão os casos, ou instâncias:** Definição dos repositórios dos casos, ou instâncias do fluxo de trabalho. Têm as mesmas características das pastas comuns, contendo documentos, imagens, fotos, textos e sons; e
- **Papéis, tanto usuários, quanto funcionais:** Criação dos papéis físicos, ou pessoais, atribuídos a pessoas e os papéis lógicos, ou funcionais, ou seja, as funções a serem desempenhadas no procedimento. Após a criação, deve-se definir o direito de acesso às pastas e aos procedimentos que podem-se acessar.

#### 2.13.1.4 – 4ª Etapa: Implantar o *Workflow*

Apresentam-se, aqui, algumas etapas que não devem ser menosprezadas na implantação de um sistema *Workflow*.

- treinamento dos analistas responsáveis pela implantação do sistema, com a finalidade de dar segurança ao usuário final na forma de suporte; e
- treinamento do usuário final.

A implantação pode ocorrer das seguintes formas:

- em paralelo com o fluxo anterior;
- com descontinuação parcial do fluxo anterior;

- com descontinuação total do fluxo anterior; e
- com sobreposição do fluxo anterior pelo novo.

### **2.13.1.5 – 5ª Etapa: Atualizar o modelo de informação implantado**

Uma etapa de revisão é importante, à medida que se tiver a intenção de melhorar continuamente o sistema implantado.

A revisão do modelo implantado é baseado no uso de ferramentas que permitem verificar o desempenho de cada etapa do fluxo de trabalho. Essas ferramentas, geralmente, coletam dados em tempo real e abrangem três níveis: o total, o de um grupo de etapas e o de uma atividade específica. Essas informações são disponibilizadas através de gráficos e dados estatísticos.

Decorrente da análise dos dados de desempenho, pode-se planejar e programar as atualizações necessárias para uma melhoria contínua do modelo implantado. Para tal, pode-se seguir o seguinte modelo:

- maximizar na origem a captura dos dados relevantes;
- maximizar o número de atividades, executando-as em paralelo;
- minimizar o número de atividades ou participantes no processo;
- minimizar o número de formulários no processos;
- minimizar o tempo em que o trabalho fica na fila esperando para começar ou terminar;
- minimizar o tamanho das filas internas;
- minimizar o tempo necessário para iniciar aplicações externas; e

- minimizar o tempo necessário para transferir trabalho entre atividades.

## 2.14 - Padrões para o sistema *Workflow*

Em 1995, os principais fabricantes criaram a *Workflow Management Coalition* – *WfMC*, para ser o órgão coordenador dos trabalhos de padronização do ambiente *Workflow* e a interoperabilidade entre as ferramentas.

Todo *software* de *Workflow* comprometido com a padronização deve conter um módulo interno de conectividade a vários tipos de bancos de dados e convivência com uma multiplataforma de *hardware* e *software*. Este módulo é denominado API - *Application Program Interface*.

Essa interface possibilitará a interoperabilidade de um fluxo de trabalho automatizado, através de todo e qualquer tipo de rede, banco de dados e outros equipamentos com os quais seja necessário interagir.

Outra preocupação que o sistema *Workflow* deva possuir é conter a abertura para tecnologias, que as mais diversas aplicações necessitarão, como *software* para *e-mail*, gerenciadores de textos, planilhas que não, necessariamente, precisam ser do mesmo fabricante do *software* de *Workflow*.

A API deve possibilitar a integração de elementos como *Object Linking and Embedding* - OLE, *Messaging Application Program Interface* - MAPI,

*Open Document Management API - ODMA, Structure Query Language - SQL e Open Database Conectivity - ODBC.*

Assim, a WfMC criou um modelo considerado como referência para o ambiente *Workflow*. Baseado no princípio que se deve interagir com um número ilimitado de tecnologias, criou-se o modelo baseado em seis pontos principais:

- ferramentas para definição do processo;
- serviço *Workflow*;
- interface com o usuário;
- links com outras aplicações e serviços;
- ferramentas que transferem trabalho para outro serviço *Workflow*;
- ferramentas de administração e monitoramento.

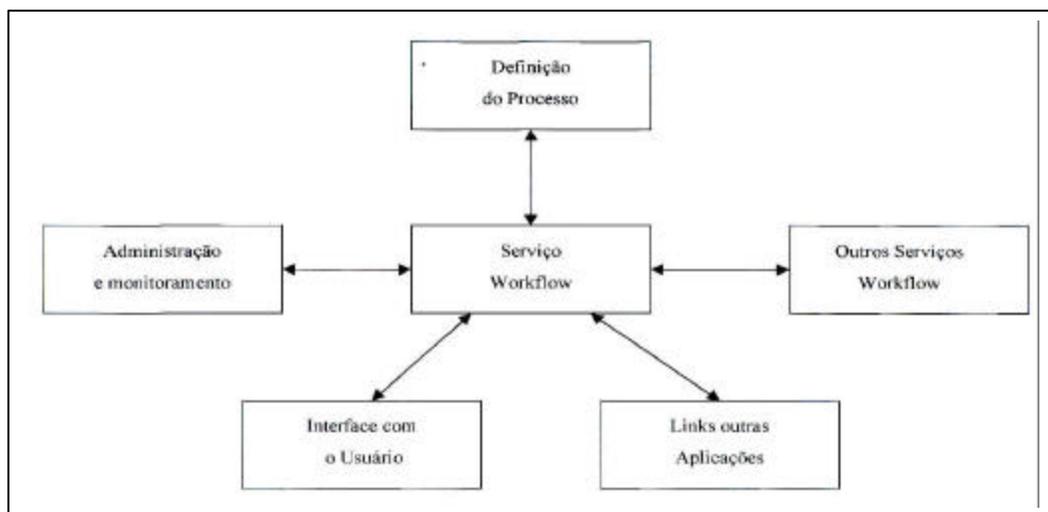


Figura 2.7 – Modelo de referência *Workflow*

Fonte: Kobelius (1997, página 48)

Em 2002, já estão aceitos os padrões relativos as interface: 1 – definição de processos de negócio, 2 – aplicação de API, 4 – Interoperabilidade e 5 – ferramentas de monitoração e administração. Ainda está em discussão a interface 3 – interface de chamada de aplicação.

Está para ser lançado um protocolo para o uso de *Workflow* nas operações da Internet e das Intranets, o SWAP - *Simple Workflow Access Protocol*.

## **2.15 - Benefícios do uso do sistema *Workflow***

A *WfMC*, apresenta como os principais benefícios da utilização do Sistema *Workflow*.

- integridade e integração dos processos;
- aumento da produtividade;
- aumento da comunicação interna;
- disseminação do ciclo do processo; e
- métrica de processos.

## **2.16 - Segmentação de mercado**

Todos os *softwares* para sistemas *Workflow* podem ser classificados dentro de um gráfico onde pode ser caracterizado o seu tipo: ad-hoc, Produção, Orientado a Objeto – OO e baseado no Conhecimento; e seu

ambiente de desenvolvimento, ou seja, centrado no Processo ou no Documento ou no *e-mail*. Na Figura 2.8 é apresentado esse gráfico, classificando os *softwares* de alguns dos fornecedores de *Workflow*.

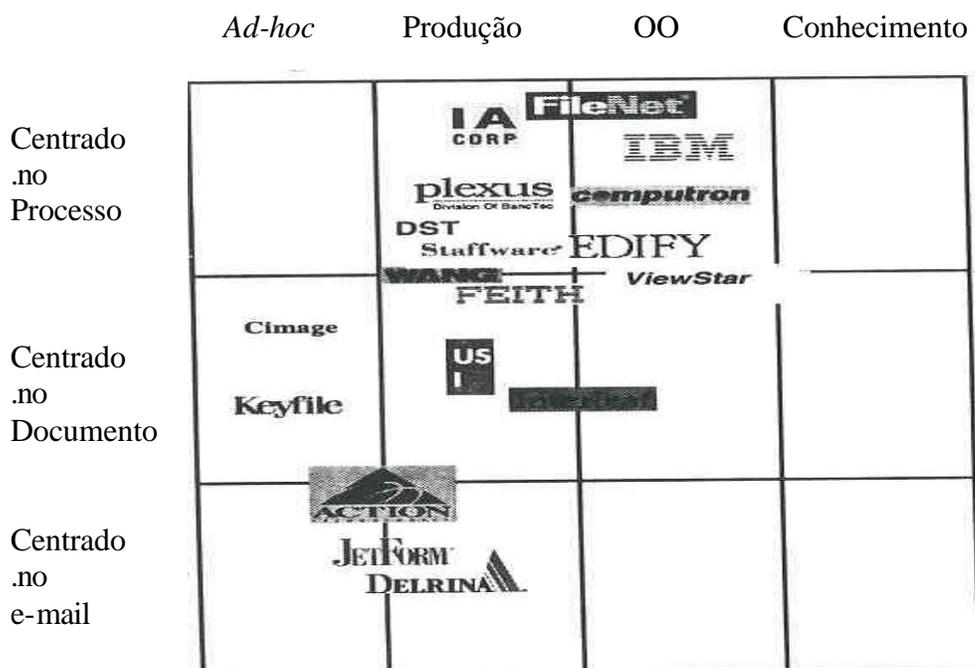


Figura 2.8 – Gráfico de classificação de *software* de *Workflow*

Fonte:Scultey (2001, página 64)

## 2.17 - Banco de dados temporais

Segundo Votre (1998), os bancos de dados relacionados a um sistema *Workflow*, em especial, os internos ao *software* aplicativo, que têm como finalidade a administração e monitoramento do fluxo, devem possuir tratamento de tempo. Esse tipo de banco de dados, que trata o tempo, é denominado banco de dados temporais.

Segundo Costa e Edelweiss (1998), Navathe e Tanaka (1994) e Elmasri (1991), nos casos de relacionamento do tempo ao banco de dados, todas as atualizações, inclusão, alteração e exclusão devem ser executadas na ordem temporal de sua ocorrência no mundo real. Portanto, não será possível fazer atualizações referentes ao passado.

O tempo não deve ser incorporado diretamente aos registros, mas através da definição de fatos específicos. A cada inclusão, alteração ou exclusão de uma informação no banco de dados deverá ser incluído um fato, associando os dados dessa informação ao dia e à hora daquele evento.

A informação propriamente dita, nesse caso, será incluída, somente, no momento de sua primeira definição, ficando presente a partir desse momento. O fato da informação estar, em algum momento posterior, definida, ou não, será determinado pela informação correspondente aos momentos de inclusão, alteração ou exclusão.

Para que seja possível a manipulação das informações contidas nos bancos de dados, será necessária a definição de atributos que indiquem a sua existência, ou não, no instante considerado. Para que seja possível o registro da evolução dos dados com o passar do tempo, é necessário associá-los aos dados armazenados, identificando quando a informação foi definida ou o tempo de sua validade.

A noção de tempo, como datas, períodos, duração de validade, surge em diferentes níveis: a) na modelagem de dados; b) na linguagem de recuperação e manipulação de dados; e c) na implementação do SGBD.

### 2.17.1 – Conceitos de representação temporal

As definições completas dos conceitos, aqui, apresentados podem ser encontradas em Jensen (1994).

**Dimensão temporal** – Os modelos de dados tradicionais apresentam duas dimensões: a primeira, as instâncias dos dados em linhas de uma tabela, e a segunda, os atributos de cada instância através das colunas de determinada tabela. Cada atributo de uma instância apresenta um só valor, caso seja feita alteração deste valor, o anterior é perdido.

Os modelos temporais acrescentam mais uma dimensão aos modelos tradicionais, a dimensão temporal. Essa dimensão associa alguma informação temporal a cada valor. Segundo Edelweiss (1998), caso o valor de um atributo seja alterado, o valor anterior não é removido do banco de dados; o novo valor é acrescentado, associado a alguma informação que define seu tempo inicial de validade. Desse modo, é possível acessar toda a história dos atributos, sendo possível analisar sua evolução temporal.

**Ordem no tempo** – A dimensão temporal é composta por uma seqüência de pontos consecutivos no tempo, que recebe o nome de eixo temporal. A definição de uma ordem a ser seguida no tempo é fundamental. Quando na ordenação se trabalha com uma só história passada e admite-se múltiplas histórias futuras, determina-se a combinação “passado linear, futuro ramificado”. Essa combinação consegue representar a realidade atual de uma forma bastante fiel.

**Tempo absoluto** – Consiste de uma informação temporal que define um tempo específico associado a um fato. Exemplo: José morreu em 10/10/1999.

**Tempo relativo** – Quando sua validade é relacionada à validade de outro fato. Exemplo: a gasolina aumentou ontem.

**Variação temporal** - Duas formas de variação temporal são aceitas o tempo contínuo e o tempo discreto. Os modelos de dados que suportam a noção discreta de variação temporal são baseados em uma linha de tempo composta de uma seqüência de intervalos temporais consecutivos de idêntica duração, chamadas de *chronons*.

A variação temporal discreta pode ser das seguintes formas: a) variação ponto a ponto – onde o valor definido vale somente no ponto temporal onde foi definido; b) variação por escada – onde o valor fica constante desde o ponto em que foi definido até o instante em que outro valor seja definido; e c) variação temporal definida por uma função – existe uma função que define os valores e que permite a interpretação para obter os valores nos pontos não definidos.

É importante observar que a variação por escada corresponde, geralmente, à definição de valores, em conseqüência da ocorrência de eventos.

**Granularidade temporal** – A granularidade temporal de um sistema consiste na duração de um *chronon*. Entretanto, dependendo da aplicação considerada, às vezes, é necessário considerar simultaneamente diferentes granularidades (minutos, dias, anos) para permitir uma melhor representação

da realidade. Embora o *chronon* do sistema seja único, é possível manipular essas diferentes granularidades.

### **2.17.2 – Classificação de bancos de dados temporais**

Um banco de dados temporal é aquele que apresenta alguma forma implícita de representação de informações temporais. Pode ser utilizado o tempo de transação e/ou o de validade para representar essa informação temporal. Esses bancos de dados podem ser classificados em quatro diferentes tipos:

- banco de dados instantâneos;
- banco de dados de tempo de transação;
- banco de dados de tempo de validade; e
- banco de dados bitemporais.

#### **2.17.2.1 – Banco de dados instantâneos**

Correspondem aos bancos de dados convencionais, onde são armazenados os valores presentes. A cada modificação no atributo, o valor, anteriormente armazenado é descartado e, somente, o último valor fica disponível;

#### **2.17.2.2 – Banco de dados de tempo de transação**

Associação de cada valor definido com o tempo de transação, sob a forma de rótulo temporal. Esse tempo é fornecido automaticamente pelo SGBD, sendo esta operação transparente ao usuário.

### **2.17.2.3 – Banco de dados de tempo de validade**

Associa a cada informação, somente, o tempo de sua validade no mundo real. Esse pode representar o início de sua validade, somente, naquele ponto no tempo, ou seu intervalo de validade. O tempo de validade deve ser sempre fornecido pelo usuário.

Caso ocorra que o dia em que é procedida a atualização do campo não coincida com o dia em que começa sua validade, a data de início da validade pode ser armazenada como um atributo explícito; e

### **2.17.2.4 – Banco de dados bitemporais**

Forma mais completa de armazenar informações temporais, nos quais os tempos de transação e de validade são associados a cada informação. Assim, pode-se saber não, somente, o valor atual de um atributo, como o valor que era válido em qualquer data passada e, ainda, aqueles que se acreditava como válidos, mas que em datas posteriores foram modificados.

Essa situação é apresentada na Figura 2.9, onde é apresentado um exemplo hipotético de toda a história de atualização do salário de um funcionário chamado João. Observa-se o registro de que momento foi definido os valores ou tempo de transação e do registro de sua validade ou tempo de validade.

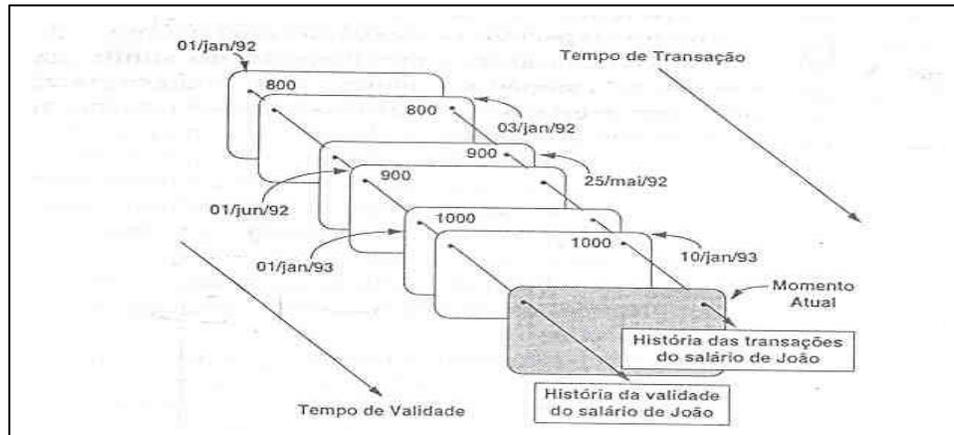


Figura 2.9 – Banco de dados bitemporal

Fonte: Edelweiss (1998, página 236)

### 2.17.3 – Consultas a bancos de dados temporais

Uma linguagem de consulta temporal é importante, quando se utiliza um banco de dados temporal. Esse tipo de linguagem deve possibilitar a recuperação de todo tipo de informação, em especial, as informações temporais, de modo que se tire o real proveito do acréscimo da dimensão temporal.

Dentre alguns problemas constatados quanto ao processamento de consultas temporais, pode-se citar:

- o grande volume de dados armazenado em um banco de dados temporal implica a determinação de novos métodos de indexação;

- métodos tradicionais de indexação só podem ser utilizados para valores com algum tipo de ordenação completa, com estruturas de acesso para intervalos; e
- manipulação de informações incompletas, a partir da qual devem ser inferidas informações, devido a suas incertezas ou indeterminações.

As consultas temporais dependem do tipo de banco de dados definido. Os bancos de dados instantâneos não apresentam suporte para informações temporais, não permitindo, assim, consultas temporais.

Já, nos bancos de dados de tempo de transação podem ser feitas consultas a valores atuais das informações armazenadas e a valores definidos em tempos passados. Nos bancos de dados de tempo de validade podem ser recuperadas as informações válidas em momentos presentes e passados, além de valores armazenados sob forma de previsão para o futuro, de acordo com a atual percepção da história das informações.

Os bancos de dados bitemporais permitem que sejam elaboradas consultas a respeito de valores atuais, passados e futuros, considerando o tempo de transação e o de validade. Qualquer estado do banco de dados pode ser consultado, sendo seu conjunto de estados (passados, atual e futuros) caracterizado como a sua história.

Uma consulta apresenta dois componentes ortogonais: um componente de seleção e outro de saída ou projeção. Sob este aspecto,

apresentamos, a seguir, as diferentes formas de consultas, quando utilizados bancos de dados temporais.

### **2.17.3.1 – Componentes de seleção**

O componente de seleção, geralmente, é representado através de uma condição lógica. Quando essa condição envolve informações temporais, é utilizada a chamada lógica temporal. Essa lógica pode utilizar-se de vários operadores, tais como: os booleanos (antes, depois e durante) e operadores, que retornam valores temporais (depois, agora, início, duração).

Segundo Edelweiss (1998), conforme o componente de seleção, as consultas são classificadas em:

- consultas de seleção sobre dados – quando as condições de seleção são estabelecidas somente sobre valores de dados. Exemplo: selecionar os nomes de funcionários que apresentam uma determinada data de nascimento;
- consultas de seleção temporal – são as consultas nas quais, somente, informações temporais associadas aos dados são analisadas pela condição de seleção. Exemplo: selecionar todos os empregados de uma empresa no ano de 1999; e
- consultas de seleção mista – as condições de seleção atuam não, somente, nos valores de dados, mas, também, nas informações temporais associadas a eles.

### **2.17.3.2 – Componentes de saída ou projeção**

Nas consultas podem ser solicitados valores de dados e/ou valores relativos às informações temporais associadas aos dados.

Ainda segundo Edelweiss (1998), conforme o componente de projeção, as consultas são classificadas em:

- consultas de saídas de dados – nas quais as informações selecionadas correspondem exclusivamente a valores de dados;
- consultas de saídas temporal – recuperam informações abstraídas das informações temporais associadas aos dados. Deste modo, podem ser recuperados pontos no tempo, em intervalos temporais e nas durações temporais; e
- consultas de saídas mistas – recuperam, simultaneamente, valores de dados e valores temporais associados a estes dados.

Como não tem sentido uma consulta temporal sem apresentar algum outro dado envolvido, entre as possíveis combinações, entre os componentes de seleção e de saída de uma consulta, somente, a combinação de seleção temporal com saída temporal não pode ser utilizada.

### **2.17.4 – Modelagem de sistema *Workflow***

Segundo Edelweiss (1998), um dos maiores problemas detectados na gerência de um sistema de *Workflow* é o controle dos problemas decorrentes da coordenação das atividades. Mesmo nos processos administrativos mais

comuns, não é possível controlar todas as atividades envolvidas. A necessidade de desenvolver sistemas que gerenciem o fluxo de trabalho com eficiência levou ao desenvolvimento de técnicas de modelagem específicas para estas aplicações.

Conclui-se, então, que num modelo de sistema *Workflow* devem estar representados os processos, os passos dos quais os processos são compostos, os agentes responsáveis por cada passo, as restrições temporais à execução de cada um dos passos.

São, exatamente, as restrições temporais que definem a seqüência válida dos passos a serem executados e o seu sincronismo entre os passos. Assim, a representação de informações temporais em modelo de sistemas *Workflow* é de fundamental importância para possibilitar a seqüência e o sincronismo citado.

## 3. INTELIGÊNCIA ARTIFICIAL

### 3.1 - Defina Inteligência Artificial

Definir Inteligência Artificial – IA é uma tarefa difícil, assim, buscou-se algumas definições fornecidas por autores de renome na literatura especializada:

“IA é o estudo das idéias que permitem facilitar os computadores a fazerem coisas que tornam as pessoas inteligentes.”, conforme Winston (1977).

“IA é o estudo de como fazer os computadores realizarem tarefas que, no momento, são feitas melhor por pessoas.”, conforme Rich (1983).

“IA é o estudo das faculdades mentais através do uso de modelos computacionais.”, conforme Charniak e McDermott (1985).

“IA é o campo de conhecimento onde se estudam sistemas capazes de reproduzir algumas atividades mentais humanas.”, conforme Nilsson (1986).

“O objetivo da IA é o desenvolvimento de paradigmas ou algoritmos que requeiram máquinas para realizar tarefas cognitivas, para quais os humanos são atualmente melhores.”, conforme Haykin (2001).

A reprodução de algumas atividades humanas pode ser obtida através de uma coleção de técnicas sustentadas por computadores, como por exemplo: representação do conhecimento, as capacidades de inferência, a resolução de problemas, o engajamento num dialogo e a compreensão da linguagem natural, bem como o reconhecimento e síntese da fala, a visão computadorizada e a robótica.

Conforme Yoneyama e Nascimento (2000), a Inteligência Artificial busca prover máquinas com a capacidade de realizar algumas atividades mentais do ser humano. Em geral, são máquinas com algum recurso computacional, de variadas arquiteturas, que permitem a implementação de rotinas não, necessariamente, algorítmicas.

As atividades psíquicas mais envolvidas na aplicação de Inteligência Artificial, ainda segundo Yoneyama e Nascimento (2000), são:

- a) **Sensação:** fenômeno elementar resultante de estímulos mecânicos, físicos, químicos ou elétricos sobre o organismo;
- b) **Percepção:** é a tomada de conhecimento e um objeto exterior considerado real;
- c) **Juízo:** é a capacidade de exprimir os vínculos e as relações entre os fatos e os objetos da natureza. Em termos da engenharia, corresponde à capacidade de qualificar os fenômenos e os objetos do sistema, segundo as leis físicas conhecidas;
- d) **Raciocínio:** é a concatenação disciplinada dos juízos, de modo a gerar novos juízos, ou seja, a capacidade de dedução;

- e) **Memória:** é a capacidade de armazenamento de informações para utilização posterior;
- f) **Orientação:** é a capacidade do indivíduo de ter consciência de sua situação temporal e espacial em relação ao meio; e
- g) **Linguagem:** é o mecanismo que permite a expressão simbólica. Tipicamente, verbais ou gráficas.

### 3.1.1 - Define aprendizado

O aprendizado é uma área fundamental em Inteligência Artificial. Para Kodratoff (1986), aprendizado é a aquisição de conceitos e de conhecimentos estruturados. Essa aquisição envolve as seguintes atividades psíquicas: o juízo, o raciocínio e a memória.

Segundo Chorafas (1988), o aprendizado pode ocorrer com ou sem a presença de um tutor – professor ou supervisor, sendo classificado, respectivamente, como supervisionado ou não-supervisionado. Dependendo da intensidade do envolvimento do tutor, o aprendizado pode ser por descoberta, por exemplos ou por programação.

Os mecanismos empregados, segundo Yoneyama e Nascimento (2000), no processo de aprendizado podem ser:

- a) **Numérico ou conceptual:** O aprendizado onde os valores de certos parâmetros são ajustados no processo de armazenamento - exemplo: redes neuronais. Será classificado como conceptual se usar uma linguagem simbólica;

- b) **Punição e recompensa:** Onde tentativa e erros são disciplinadas por um supervisor, que fornece ao aprendiz um sinal de realimentação na forma de punição ou recompensa;
- c) **Empírico ou racional:** Empírico, quando ocorre com base em experimentação ou amostragem; ou racional, quando é direcionado por um mecanismo de inferência; e
- d) **Dedutivo/Indutivo/Inventivo:** Dedutivo, quando pode ser alcançado, a partir de mecanismos de inferência sobre o conjunto de premissas; indutivo, quando há a necessidade de generalizações dos conceitos apreendidos, a partir de exemplos e do conjunto de premissas; e inventivo, se há a necessidade de se adquirir novas premissas, sem ser as generalizações definidas na indutiva.

Ainda, segundo Yoneyama e Nascimento (2000), são inúmeras as ferramentas utilizadas em IA. Neste trabalho, optou-se por limitá-las nas ferramentas mais empregadas, são elas:

- i) **Sistemas de Produção:** onde se busca caracterizar a heurística e os conhecimentos de especialistas humanos, através da aplicação de conceitos de lógica, sendo heurística as regras e métodos que conduzem a resolução de problemas;
- ii) **Lógica Nebulosa:** onde se busca considerar as incertezas inerentes à representação humana dos fenômenos da

natureza, refletida nas expressões verbais empregadas corriqueiramente;

- iii) **Redes Neurais Artificiais:** onde se busca mimetizar o sistema nervoso através de elementos processadores simples denominados neurônios, também denominados de neurocomputadores, de conexionismo ou sistemas de processamento paralelo e distribuído; e
- iv) **Agentes Inteligentes:** onde se busca perceber o ambiente através de sensores, respondendo ao ambiente através de atuadores ou *effectors*.

### 3.1.2 - Define maquina de inferência

A palavra máquina é de origem grega, mas possuía um significado diferente do que lhe é dado hoje. Na antiguidade, uma máquina era qualquer instrumento ou artifício. A palavra não se relacionava nem à tecnologia, nem ao uso intencional do objeto, seja na produção, transformação, transporte, cálculo ou comunicação.

Segundo Chorafas (1988), a inferência é a subestrutura menos conhecida do domínio maior que inclui a tomada de decisões, bem como o pensamento científico. A inferência é tanto um ato quanto um processo: como utilizar estratégias de raciocínio aproximadas para chegar-se a uma estimativa válida de uma situação, enquanto possuímos dados incertos e regras imperfeitas.

É a existência e a realidade que desejamos compreender. Para fazê-los, devemos estabelecer conexões entre os fatos da experiência. Devemos agir de forma a podermos prever ocorrências posteriores àquelas já experimentadas. Alguns acham que a realização plena desta tarefa é o único fim do esforço científico.

A tecnologia nasce de conceitos que evoluem regularmente, dos vínculos que estabelece-se entre os fatos, experiência e habilidades e dos instrumentos de medida que desenvolvidos, com o objetivo de obter maior exatidão e precisão na resolução de problemas concretos.

Administradores, profissionais, cientistas e tecnólogos visam criar ou estabelecer o sistema e pensamento mais simples possível, que aglutinará os fatos observados. O sistema mais simples é aquele que contém o mínimo possível de postulados ou axiomas mutuamente independentes, sendo axioma definido como premissa evidente que se admite como verdadeira sem exigência de demonstração.

O objetivo especial e perene é a unificação lógica. Einstein observou, que o conteúdo de axiomas lógicos, mutuamente independentes representam a sobra do que não é compreendido. A unificação lógica foi, pela primeira vez, aplicada como um princípio no campo da física e, portanto, no mundo físico.

O que se conhece da unificação lógica são algumas equações lógicas básicas, entre elas:

Regras + Metodologia = Inferência

Contexto + Informação = Conhecimento

Inferência + Conhecimento = Sistema Inteligente

Algoritmo + Dados = Programa Computadorizado

### 3.1.3 - Define instrumentação, controle e automação

Segundo, Yoneyama e Nascimento (2000), um sistema é constituído de diversos componentes que interagem entre si, em conformidade com as leis da natureza. Ao restante do universo que não integra o sistema, é denominado de meio ambiente. Obviamente, que o meio ambiente interage com o sistema, alterando as suas características. As grandezas envolvidas nas interações do sistema com o meio ambiente nem sempre podem ser ajustadas, convenientemente, e são classificadas de **entradas** e **saídas**. As entradas, que não podem ser ajustadas, são denominadas perturbações ou ruídos, mas, as que podem são denominadas variáveis de controle ou manipuladas.

Controlar é, justamente, atuar sobre essas grandezas, de modo que o sistema possua um comportamento adequado. Quando o controle é realizado com pouca ou nenhuma intervenção humana, diz-se que está automatizado.

O problema de controle é o de obter uma estratégia de atuação sobre um sistema, de modo que este se comporte de forma conveniente. Devido,

na maioria das vezes, deseja-se indicar quantitativamente, as especificações de desempenho, torna-se necessária a utilização de modelos matemáticos para este fim. Observa-se, que modelos são as representações úteis do sistema em estudo e, neste contexto, um modelo matemático é uma representação de como as diversas grandezas interagem entre si, geralmente, expressa na forma de equações.

### **3.1.3.1 - Controle por computador**

Computadores digitais têm sido empregados em tarefas de controle automatizado, em função de apresentarem diversas vantagens em relação a sistemas baseados em alavancas e cames, ou circuitos elétricos, pneumáticos ou hidráulicos.

Computadores digitais utilizados em controle automático recebem dados dos sensores, em determinados instantes de tempo e, após processá-los, produzem saídas que são aplicadas nos transdutores. As entradas são, usualmente, analógicas e são transformadas em forma digital através de conversores análogo-digitais. As saídas dos computadores são números que devem ser transformados em grandezas analógicas através de conversores digital-analógicos. Portanto, no controle existem pontos onde os dados estão em formato numérico.

Para exemplificar, muitos dos sistemas dinâmicos encontrados na prática são descritos por equações diferenciais. Os movimentos de um satélite artificial poderiam ser descritos com base nas Leis de Newton. As

variações de tensões e correntes em um circuito poderiam ser descritas pelas Leis de Kirchoff.

### **3.1.3.2 - Controladores baseados em conhecimento**

Muitas vezes, mesmo sob a ausência de um modelo matemático preciso ou de algoritmos bem definidos, o operador humano é capaz de agir sobre uma determinada planta, utilizando a experiência acumulada ao longo dos tempos. Entre as tarefas que podem ser realizadas por sistemas baseados em conhecimento estão: interpretação dos dados, predição, diagnose, síntese, planejamento, monitoração, correção das falhas, treinamento e controle ativo.

A arquitetura mais simples de um sistema baseado em conhecimento envolve um banco de conhecimentos, onde as regras estão armazenadas; um banco de dados, onde as informações sobre as condições da planta a ser controlada e as medidas estão armazenadas. É uma máquina de inferência que deverá deduzir as ações a serem tomadas, em função das informações dos bancos de dados e de conhecimentos.

A figura, a seguir, apresenta um diagrama de blocos de um sistema baseado em conhecimento, com interface para interação com o meio ambiente.

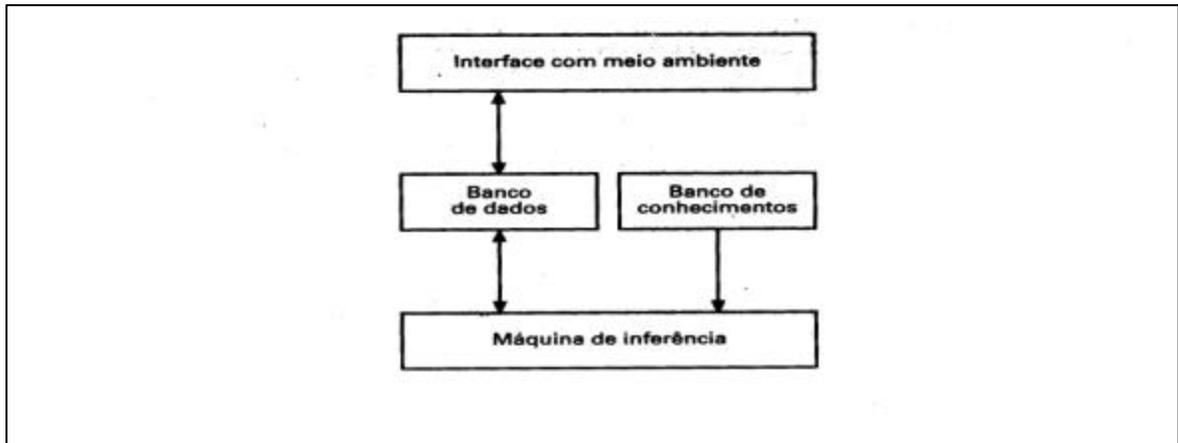


Figura 3.1 – Arquitetura típica de um sistema baseado em conhecimento

Fonte: Yoneyama e Nascimento (2000, página 58)

A máquina de inferência, apresentada na figura 3.1, executaria os seguintes passos:

- 1) Busca de regras no banco de conhecimentos que tenham as condições satisfeitas, em termos de conteúdo do banco de dados;
- 2) Se houver uma ou mais regras com as condições satisfeitas, então selecionar uma ou, senão, retornar ao passo 1; e
- 3) Executar a ação descrita na regra selecionada e retornar ao passo 1.

Conforme Mylopoulos (1986) e Randall (1982), pode-se passar a falar em *Knowledge-Base Management Systems* – KBMS ou Sistema de

Gerenciamento de Base de Conhecimento, que não pode ser comparado com o Sistema de Gerenciamento de Banco de Dados – SGBD, em especial, quanto a funções oferecidas voltadas para usuário final como o SQL, no caso de banco de dados relacionais, sendo que no KBMS as funções estão focadas no desenvolvedor da base de conhecimento.

Segundo Mylopoulos (1986), os princípios do KBMS são:

- a) **Projeção:** Captura do relacionamento semântico entre o objeto físico e sua projeção em vários planos, implementados através de mecanismos de abstração;
- b) **Exceções:** O conhecimento com respeito a esta dimensão é baseado em regras de estrutura caracterizadas como exceção das regras;
- c) **Reflexão:** Organização e controle do conhecimento; e
- d) **Relatividade:** A base de conhecimento assume que podem existir visões objetivas do domínio, que podem ser tratadas antes que o conhecimento seja armazenado na base de conhecimentos.

### **3.2 - Define Engenharia do Conhecimento**

Segundo Durkin (1993), o objeto da Engenharia do Conhecimento é o desenvolvimento, produção e distribuição de inteligência através de sistemas elaborados pelo homem. A Engenharia do Conhecimento é o lado da ciência aplicada da Inteligência Artificial.

Ainda segundo Durkin (1993), o maior desafio, nesse esforço de ciência aplicada, é a representação do conhecimento: da aquisição do conhecimento ao desenvolvimento de regras e a facilidade fornecida pelo uso interativo dessas regras, também, conhecida como utilização do conhecimento.

Uma das maiores dificuldades é a constatação de que grande parte das tarefas não podem ser automatizadas, utilizando-se técnicas de computação convencionais, exigindo, assim, a habilidade de manipulação simbólica. Assim, a Engenharia do Conhecimento, além de instrumentalizar a manipulação simbólica, permite a construção de caminhos de inferência para decisão, explicação e justificativa.

O estudo da organização através de abordagens que conduzem à manipulação lógica resulta, na Engenharia do Conhecimento, uma estrutura baseada em regras, que constitui em conjunto com o registro de fatos, o banco de conhecimento da organização.

### 3.2.1 - Representação de conhecimento

Segundo Yoneyama e Nascimento (2000), o conhecimento, sendo um conjunto de informações que permite articular os conceitos, os juízos e o raciocínio, usualmente, disponíveis em um particular domínio de atuação. Empregando-se um sistema de símbolos, esse conhecimento pode ser representado de modo a permitir atividades como inferência ou memorização. A representação pode ser feita de múltiplas formas, como através de textos, fórmulas matemáticas, figuras, maquetes, filmes, discos, protótipos e muitas outras alternativas.

#### 3.2.1.1 - Sistemas de Produção ou Lógica de Predicados

Opção muito comum de representar conhecimentos heurísticos, tipicamente, na forma “Se <condições> Então <Conclusões>”. Muitos sistemas especialistas utilizam sistemas de produção aliadas a uma máquina de inferência para realizar a sua tarefa. Como exemplo de sistema de produção, a representação de conhecimentos sobre alguns animais:

i.	Se (produz Leite e tem pelo)	Então (mamífero)
ii.	Se (mamífero e come carne)	Então (carnívoro)
iii.	Se (mamífero e possui presas e possui garras)	Então (carnívoro)
iv.	Se (mamífero e possui casco)	Então (ungulado)
v.	Se (carnívoro e pardo e pintado)	Então (onça)
vi.	Se (carnívoro e pardo e listrado)	Então (tigre)
vii.	Se (ungulado e pardo e pintado)	Então (girafa)
viii.	Se (ungulado e branco e listrado)	Então (zebra)

### 3.2.1.2 - Redes semânticas

A rede semântica é uma coleção de nós conectados por arcos. Os nós representam objetos, conceitos ou eventos e os arcos representam as relações. Usualmente, os nós e os arcos são etiquetados para indicar o que representam:

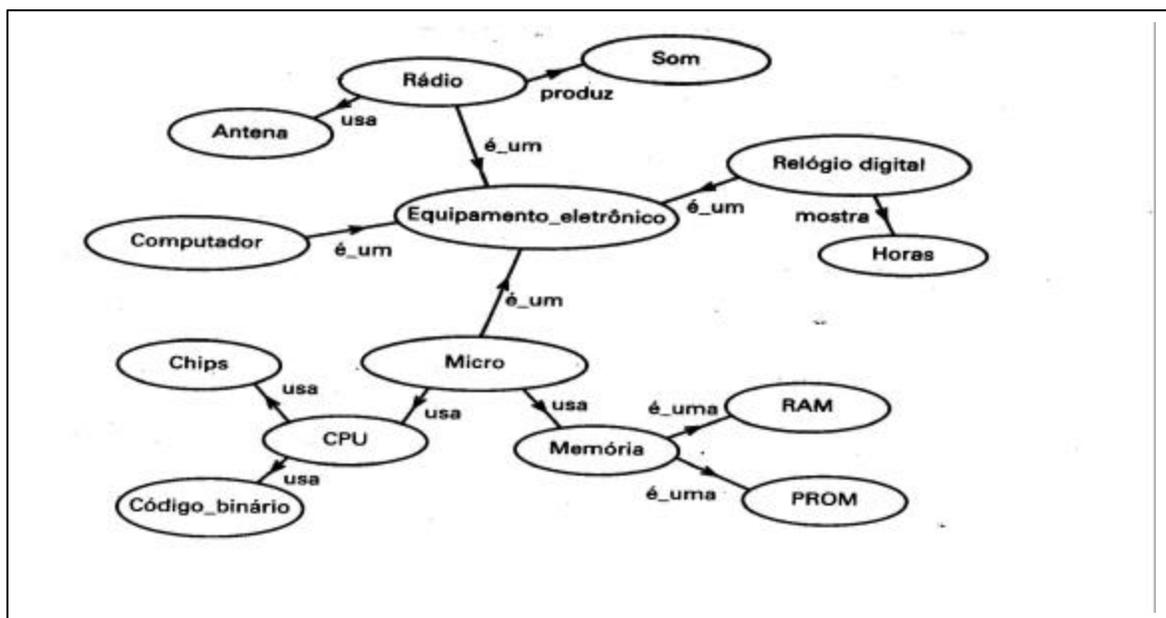


Figura 3.2 – Rede semântica de conhecimentos sobre equipamentos eletrônicos.

Fonte: Yoneyama e Nascimento (2000, página 51)

### 3.2.1.3 - Métodos de busca

Muitos problemas de IA necessitam de métodos de casamentos (*matching*) e busca (*search*) durante o processo de solução.

Considera-se o problema de obter, exatamente, 1 (um) litro de água em uma das jarras, a partir de uma jarra com 5 (cinco) litros de capacidade, inicialmente cheia, e uma de 2 (dois) litros, inicialmente vazia. Representando as duas jarras por um par ordenado  $(x,y)$ , onde  $x$  é o número de litros de água na primeira jarra e  $y$  na segunda, os passos para a solução do problema podem ser apresentados na seguinte forma gráfica:

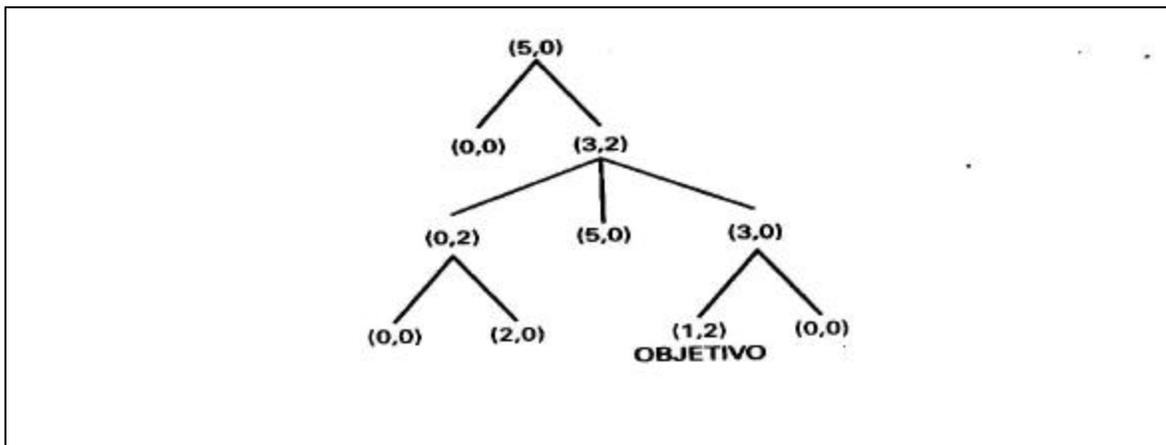


Figura 3.3 – Árvore correspondente ao problema de jarras de água.

Fonte: Yoneyama e Nascimento (2000, página 52)

Ainda, segundo Yoneyama e Nascimento (2000), o problema das jarras pode, portanto, ser resolvido mediante o uso de um algoritmo de busca em árvores. Considerando a representação de conhecimentos na forma da Lógica de Predicados, pode-se verificar que algoritmos de busca em árvores são, também, aplicáveis.

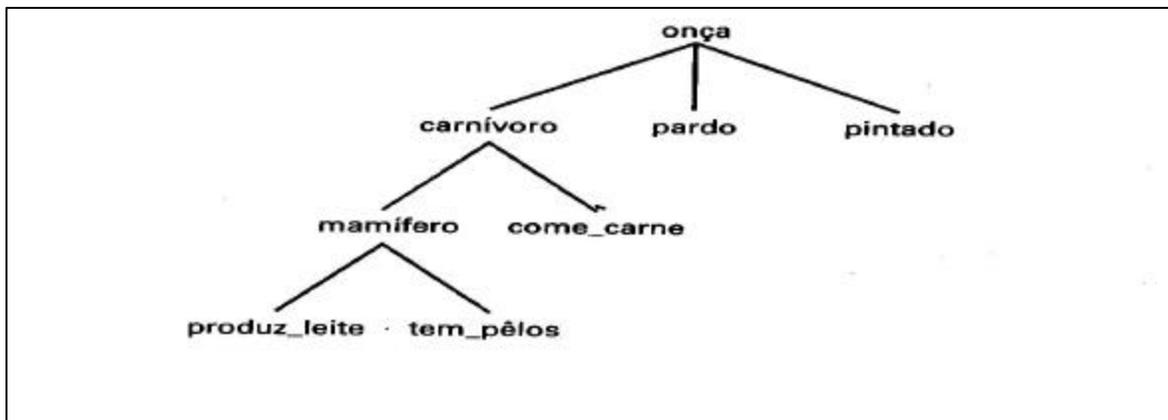


Figura 3.4 – Representação de conhecimento sobre animais, na forma de árvore.

Fonte: Yoneyama e Nascimento (2000, página 53)

Os algoritmos de buscas podem ser do tipo encadeamento progressivo - busca do objetivo pelas premissas e retroativo - busca das premissas pelo objetivo.

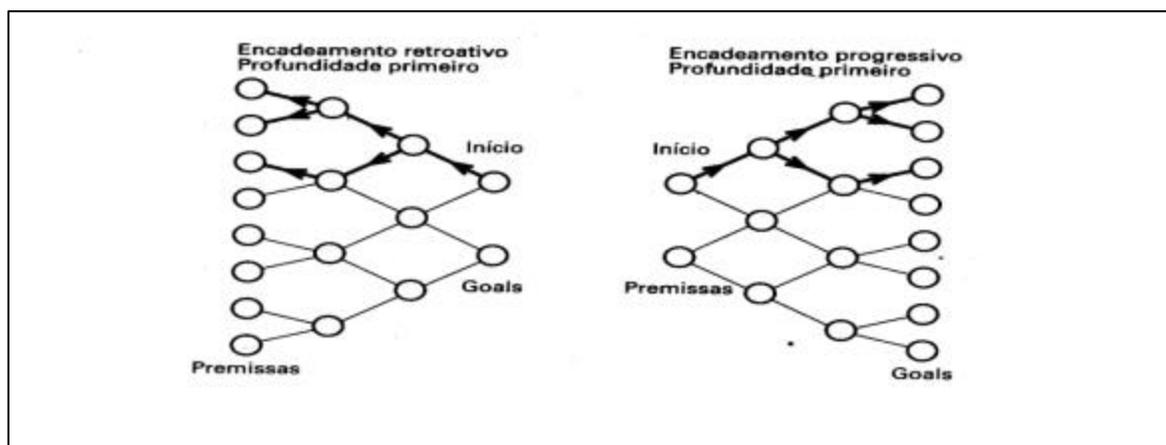


Figura 3.5 – Ilustração das formas de encadeamento para o caso de busca.

Fonte: Yoneyama e Nascimento (2000, página 55)

### 3.3 - Lógica Nebulosa - *Fuzzy Logic*

A Lógica Nebulosa permite o tratamento de expressões que envolvem grandezas descritas de forma não exata. Sendo um conjunto uma coleção de objetos, um objeto só poderia possuir duas possibilidades quanto a sua relação com o conjunto, ou seja, ele é elemento ou não do conjunto. Na teoria de Lógica Nebulosa, um objeto poderia possuir variados graus de pertinência.

A formalização do conceito de conjuntos nebulosos pode ser obtida estendendo-se a teoria clássica de conjuntos. Assim, na teoria clássica, um conjunto pode ser caracterizado pela sua função indicadora, ou seja, dado um conjunto  $A$  no universo  $X$ , define-se:

$$I_A(x) = 1 \text{ se } x \in A \text{ ou } I_A(x) = 0 \text{ se } x \notin A$$

Se  $X$  for o conjunto  $\mathbb{R}$  e  $A$  um intervalo fechado, a função indicadora de  $A$  assume o aspecto ilustrado na figura a seguir:

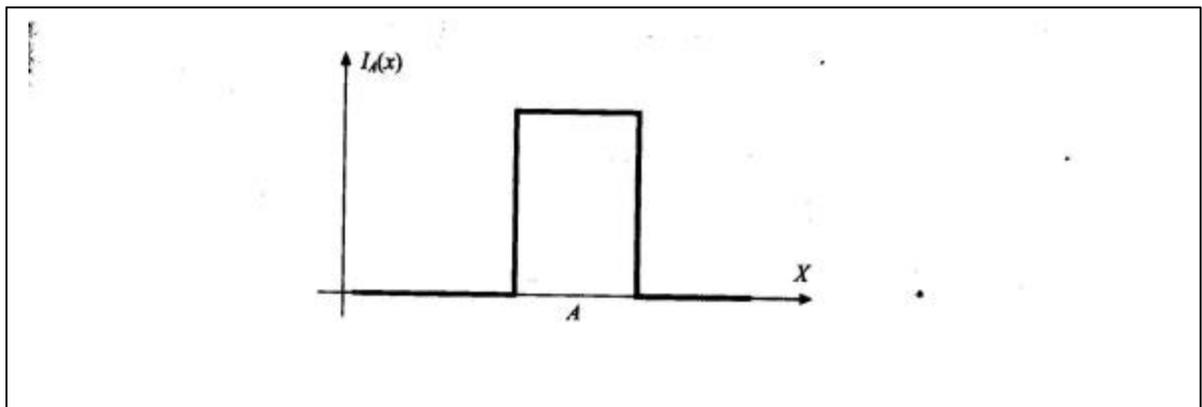


Figura 3.6 – Função indicadora do conjunto  $A$ .

Fonte: Yoneyama e Nascimento (2000, página 69)

De maneira análoga, segundo Yoneyama e Nascimento (2000), os conjuntos nebulosos são definidos por:

$$A = \{x \in X \mid \mu_A(x) = 1\}$$

Onde  $\mu_A(x):X \rightarrow [0,1]$  é a função de pertinência que expressa o quanto um dado elemento  $x$  pertence a  $A$ .

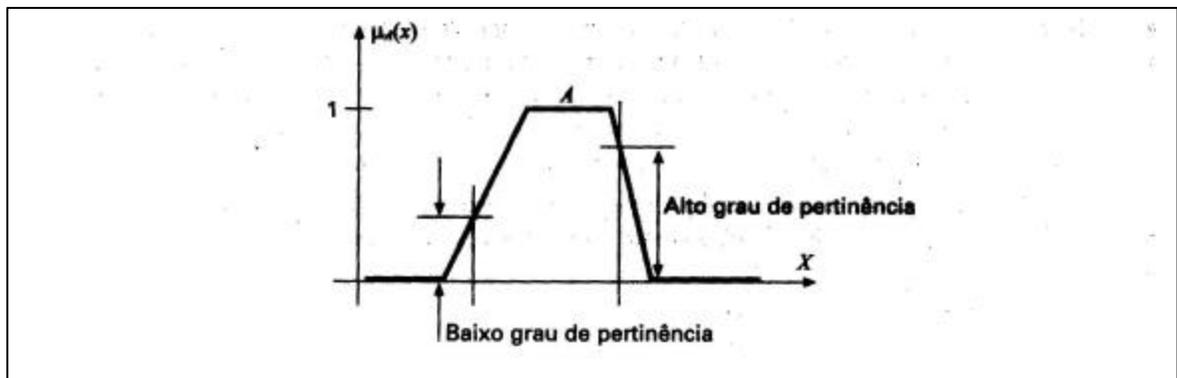


Figura 3.7 – Função de pertinência para um conjunto nebuloso  $A$ .

Fonte: Yoneyama e Nascimento (2000, página 70)

A teoria dos conjuntos nebulosos busca traduzir em termos formais a informação imprecisa que ocorre de maneira natural na representação dos fenômenos da natureza, como descrito por humanos utilizando a linguagem corriqueira.

### 3.3.1 - Controladores nebulosos

Segundo Sandri e Correia (1998), os controladores nebulosos são robustos e de grande adaptabilidade, incorporando conhecimento que outros

sistemas nem sempre conseguem acomodar. Também são versáteis, principalmente, quando, o modelo físico é complexo e de difícil representação matemática.

Na figura 3.8, é apresentada a estrutura básica de um modelo de controlador nebuloso que identifica os módulos que o compõe.

Observa-se a presença da interface de “fuzificação” e da interface de “desfuzificação”. A “fuzificação” faz a identificação dos valores das variáveis de entrada e as normaliza em um universo de discurso padronizado, transformando-os em conjuntos nebulosos. A “desfuzificação” é utilizada para obter uma única ação de controle precisa, a partir do conjunto nebuloso.

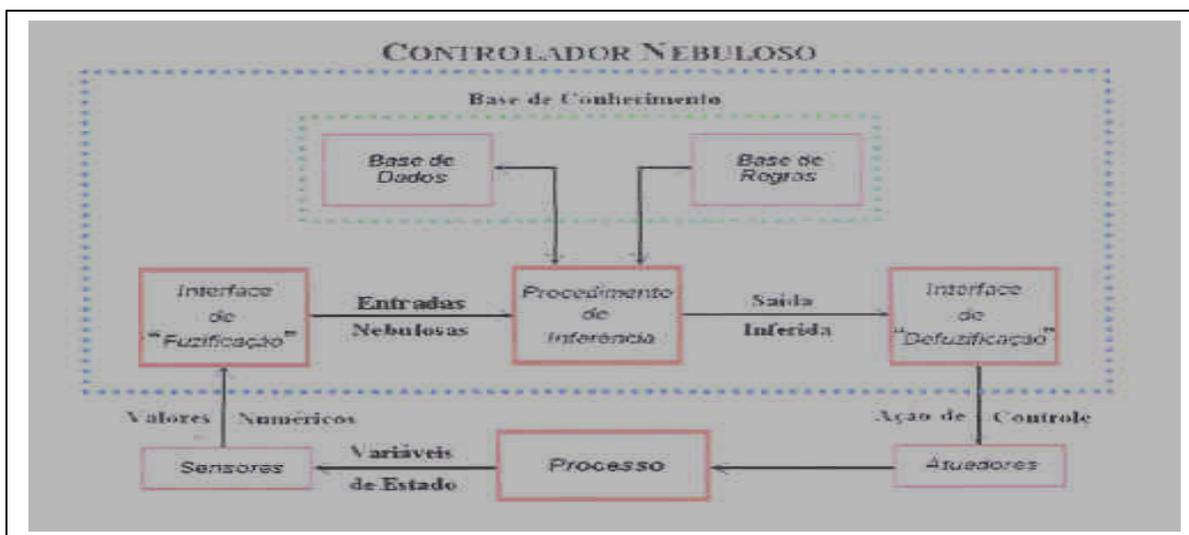


Figura 3.8 – Estrutura de um controlador nebuloso.

Fonte: Sandri e Correa (1998)

### 3.4 - Redes Neurais Artificiais – RNA

#### 3.4.1 - Define Redes Neurais Artificiais

O enfoque algoritmo na aplicação da inteligência artificial, seja nos casos de natureza numérica ou simbólica, pode ser muito útil para a classe de problemas onde é possível encontrar a solução, aplicando-se uma seqüência precisa de operações matemáticas ou regras. Entretanto, tal enfoque pode apresentar as seguintes limitações:

- Processamento predominantemente seqüencial: as operações são encadeadas, mesmo não sendo inter-relacionadas, provocando o não paralelismo de operações;
- Representação local: o conceito ou regra bem localizado não permitindo qualquer falha, por menor que seja, sendo motivo preponderante na interrupção da computação seqüencial;
- Dificuldade de aprendizado: dificuldade de incorporar os dados adquiridos via interação com o ambiente no modelo computacional;

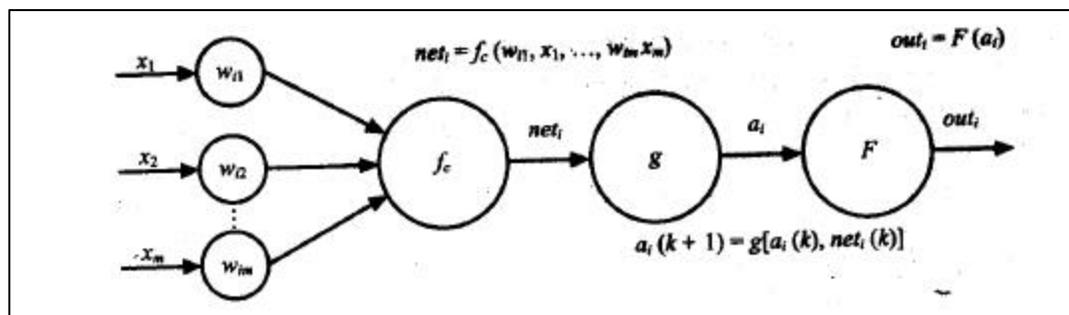


Figura 3.9 – Estrutura geral de uma unidade computacional.

Fonte: Yoneyama e Nascimento (2000, página 116)

O estudo de Redes Neurais Artificiais – RNA fornece um enfoque alternativo a ser aplicado em problemas onde os enfoques numéricos e simbólicos não são julgados muito adequados. As RNA são, apenas, inspiradas no nosso conhecimento atual sobre os sistemas nervosos biológicos da natureza, e não buscam ser reis em todos os detalhes, isto é, o modelo de sistemas nervosos biológicos não é o ponto principal de interesse.

Algumas definições formais sobre Rede Neuronal Artificial:

“Uma Rede Neural Artificial é uma estrutura que processa informação de forma paralela e distribuída e que consiste de unidades computacionais (as quais podem possuir uma memória local e podem executar operações locais) interconectadas por canais unidirecionais chamados de conexões. Cada unidade computacional possui uma única conexão de saída que pode ser dividida em quantas conexões laterais se fizer necessário, sendo que cada uma destas conexões transporta o mesmo sinal, o sinal de saída da unidade computacional. Esse sinal de saída pode ser contínuo ou discreto. O processamento executado por cada unidade computacional pode ser definido arbitrariamente, com a restrição de que ele deve ser completamente local, isto é, deve depender, somente, dos valores atuais dos sinais de entrada que chegam até a unidade computacional via as conexões e dos valores armazenados na memória local da unidade computacional”, conforme Hecht-Nielsen (1990).

“Redes Neurais Artificiais são sistemas paralelos distribuídos compostos por unidades de processamento simples – nodos que calculam determinadas funções matemáticas, normalmente não lineares. Tais unidades são dispostas em uma ou mais camadas

e interligadas por um grande número de conexões, geralmente, unidirecionais.”, conforme Braga, Ludermir e Carvalho (2000).

“Uma rede neural é um processador maciçamente paralelamente distribuído, constituído de unidade de processamento simples, que tem a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos: a) O conhecimento adquirido pela rede a partir de seu ambiente, através de um processo de aprendizagem; e b) Forças de conexão entre neurônios, conhecidas como pesos sinápticos. São utilizadas para armazenar o conhecimento adquirido”, conforme Haykin (2001).

### **3.4.2 - Estrutura para modelos de Redes Neurais Artificiais**

Segundo Haykin (2001), existem vários modelos de RNA, porém, cada modelo pode ser definido, formalmente, pelas seguintes características principais:

- um conjunto de unidades computacionais;
- um estado de ativação para cada unidade simbolizada de  $a$ ;
- uma função de saída  $F$  para cada unidade;
- um padrão de conectividade entre as unidades, o qual é definido pela matriz de pesos  $W$ ;
- uma regra de combinação usada para propagar os estados de ativação das unidades pela rede;

- uma regra de ativação usada para atualizar o estado de ativação de cada unidade, usando o valor atual do estado de ativação e as entradas recebidas de outras unidades;
- um ambiente externo que fornece informação para a rede e/ou interage com ela; e
- uma regra de aprendizado usada para modificar o padrão de conectividade da rede, usando informação fornecida pelo ambiente externo, ou seja, para modificar a matriz de pesos  $W$ .

### 3.4.3 - Topologia das Redes Neurais Artificiais

Segundo Yoneyama e Nascimento (2000), de acordo com a topologia, uma RNA pode ser classificada como *feedforward* - sem realimentação local ou *feedback* - com realimentação local ou recorrente. Em uma RNA tipo *feedforward*, uma unidade envia sua saída, apenas, para unidades das quais ela não recebe nenhuma entrada direta ou indiretamente. Em outras palavras, em uma rede tipo *feedforward* não existem laços (*loops*). Em uma rede tipo *feedback*, os laços existem. A figura 3.10 mostra os tipos de RNA: *feedforward* a e b; e *feedback* c.

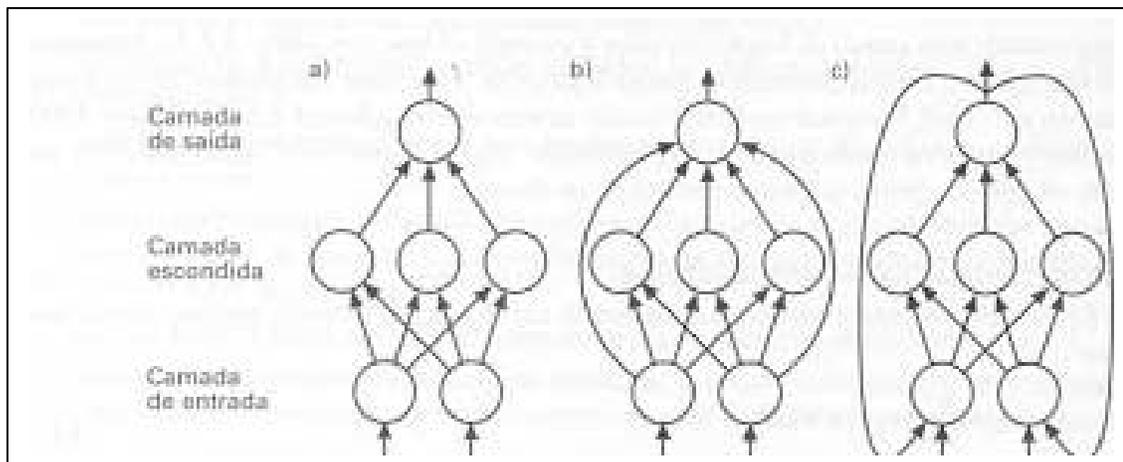


Figura 3.10 – Algumas topologias para rede neuronal artificial.

Fonte: Yoneyama e Nascimento (2000, página 118)

Uma aplicação típica de uma RNA do tipo *feedforward* é o desenvolvimento de modelos não lineares usados em reconhecimento de padrões e classificação. Uma aplicação típica de uma RNA do tipo *feedback* é como uma memória endereçada por conteúdo, onde a informação que deve ser gravada corresponde a pontos estáveis e equilíbrio estável da rede.

### 3.4.4 - Aprendizado em Redes Neurais Artificiais

Segundo Yoneyama e Nascimento (2000), a utilização de uma RNA na solução de uma tarefa passa, inicialmente, por uma fase de aprendizagem, quando a rede extrai informações relevantes de padrões de informações apresentadas para ela, criando, assim, uma representação

própria para o problema. A etapa de aprendizagem consiste em um processo interação de parâmetros da rede, os pesos das conexões entre as unidades de processamento, que guardam, ao final do processo, o conhecimento que a rede adquiriu do ambiente que está operando, além de ponderarem cada entrada recebida.

Os diferentes métodos de aprendizado podem ser classificados de acordo com a participação do supervisor ou professor no processo de aprendizado. No grau de supervisão mais forte possível, o supervisor fornece, diretamente, para a rede neural os valores dos pesos. Esse tipo de supervisão é denominado de **Muito Forte**, e as redes que a utilizam são chamadas de RNA de Peso Fixo.

Em um grau de supervisão menor, denominada **Forte**, o supervisor fornece para a RNA um conjunto de treinamento, ou seja, um conjunto de entradas e suas respectivas saídas desejadas.

Na redução seguinte do nível de supervisão, denominada **Fraca**, o supervisor faz apenas o papel de um crítico, fornecendo uma avaliação grosseira da saída da RNA, em vez de fornecer a saída desejada.

No menor grau de supervisão, denominada de **Muito Fraca**, o algoritmo de treinamento da RNA tenta descobrir categorias de dados de entrada e o supervisor participa, apenas, fornecendo os rótulos para estes agrupamentos.

### 3.4.4.1 - Aprendizado supervisionado

Método de aprendizado mais comum no treinamento das RNA, tanto de neurônios com peso, como de neurônios sem peso. São denominados supervisionados, porque a entrada e saída desejadas para a rede são fornecidas por um supervisor ou professor externo. O objetivo maior é ajustar os parâmetros da rede, de forma a encontrar uma ligação entre os pares de entrada e saída fornecidos.

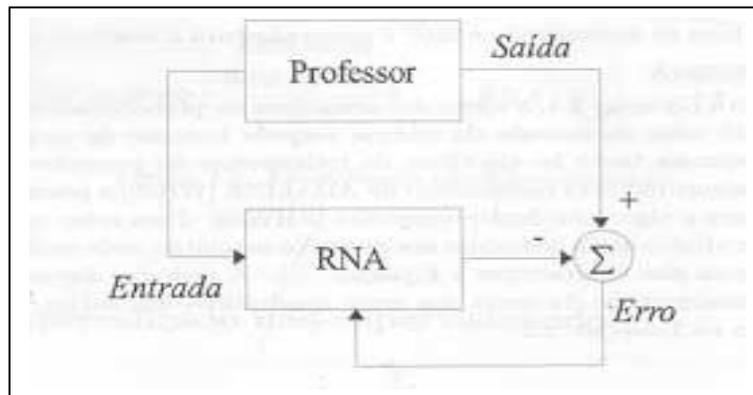


Figura 3.11 – Aprendizado supervisionado.

Fonte: Braga, Ludermir e Carvalho (2000, página 17)

A desvantagem desse tipo de aprendizado é que, na ausência do professor, a rede não conseguirá aprender novas estratégias para situações não cobertas pelos exemplos do treinamento da rede.

O aprendizado supervisionado pode ser implementado de duas formas: *off-line* e *on-line*. Para treinamento *off-line*, os dados do conjunto de treinamento não mudam e, uma vez obtida a solução para a rede, esta deve permanecer fixa. Caso novos dados sejam adicionados ao conjunto de

treinamento, um novo treinamento, envolvendo, também, os dados anteriores, deve ser realizado.

Por sua vez, no aprendizado on-line, o conjunto de dados muda continuamente, e a rede deve estar em contínuo processo de adaptação.

### **3.4.4.2 - Correção de erros**

Segundo Braga, Ludemir e Carvalho (2000), a adaptação por correção de erros procura minimizar a diferença entre a soma ponderada das entradas pelos pesos – saída calculada pela rede e a saída desejada, ou seja, o erro da resposta atual da rede.

Ainda, segundo Braga, Ludemir e Carvalho (2000), a forma genérica para alteração dos pesos por correção de erros é apresentada na equação abaixo:

$$w_i(t+1) = w_i(t) + \eta e(t) x_i(t)$$

onde  $\eta$  é a taxa de aprendizado e  $x_i(t)$  é a entrada para o neurônio  $i$  no tempo  $t$ .

Na equação apresentada, o ajuste dos pesos deve ser proporcional ao produto do erro pelo valor de entrada da sinapse, conexão entre dois neurônios propagando-se o impulso nervoso, naquele instante de tempo.

### **3.4.4.3 - Aprendizado não-supervisionado**

Segundo Braga, Ludemir e Carvalho (2000), Aprendizado não-supervisionado tem como característica não possuir supervisor ou professor para acompanhar o processo de aprendizado. Apesar da semelhança entre o aprendizado supervisionado e o aprendizado dos seres humanos, muitos dos sistemas biológicos ocorrem através do aprendizado não-supervisionado, como por exemplo: os estágios iniciais dos sistemas de audição e visão. Para estes algoritmos, somente os padrões de entrada estão disponíveis para a rede. A partir do momento que a rede estabelece uma harmonia com as regularidades estatísticas da entrada de dados, desenvolve-se nela uma habilidade de formar representações internas para codificar características de entrada e criar novas classes ou grupos automaticamente.

Este tipo de aprendizado só torna-se disponível quando existe redundância nos dados de entrada. Sem redundância seria impossível encontrar qualquer padrão em característica de dados de entrada.

### **3.4.5 - Aplicações de Redes Neurais Artificiais e Sistemas Neurais Híbridos - SNH**

As RNA podem ser utilizadas sem uma vasta gama de aplicações, podendo-se citar como exemplos de processamento de sinais: cancelamento de ruídos, reconhecimento de voz e caracteres, codificação, compressão de dados, diagnósticos médicos e outros; sistemas dinâmicos: manipuladores

mecânicos, processos industriais, veículos auto-guiados e outros; sistemas de decisão: terapêutica médica, análise financeira e outras; e numerosas outras possibilidades.

Embora RNA tenham se mostrado uma técnica eficiente para a solução de um grande número de problemas, conforme Braga, Ludermir e Carvalho (2000), é um grave erro afirmar que elas são suficientes para resolver qualquer problema de Inteligência Artificial. Assim, surgiram os Sistemas Neurais Híbridos – SNH, que combinam dois ou mais tipos diferentes de subsistemas, sendo um deles uma RNA, para formar um sistema heterogêneo com características inteligentes.

Segundo Braga, Ludemir e Carvalho (2000), a principal idéia dos Sistemas Neurais Híbridos é a de que uma única técnica, devido as suas limitações e/ou deficiências, pode não ser capaz, por si só, de resolver um dado problema. Neste caso, a combinação de duas ou mais técnicas pode levar a uma solução mais robusta e eficiente. É importante ressaltar, porém que a utilização de SNH não leva, necessariamente, a uma melhora do desempenho do sistema como um todo.

Algumas abordagens têm sido propostas para integrar RNA com: Estatística, Lógica Nebulosa, Sistemas Baseados em Conhecimento, Algoritmos Genéticos, Raciocínio Baseados em Casos, Lógica Matemática, Linguagens Formais, Agentes Inteligentes, Sistemas Tutores e Linguagem Natural.

### 3.4.6 - Extração de conhecimento

Segundo Braga, Ludemir e Carvalho (2000), as RNA são conhecidas pelo bom desempenho que, geralmente, obtêm quando utilizadas. Entretanto, para inúmeras aplicações, é importante não, apenas, o desempenho obtido, mas, também, a facilidade do usuário compreender como a rede chega às suas decisões. Mais, especificamente, quando um método é utilizado para classificação, pode ser importante entender como e por que um dado padrão de entrada foi classificado em uma dada classe. Por uma dada incapacidade de explicar como e por que a rede gera suas respostas, são as RNA, comumente, denominadas de “caixas pretas”.

Casos como: controle de usinas nucleares; sistema de navegação de aeronaves; diagnóstico médico; e análise de crédito necessitam de uma transparência de seus métodos.

É importante distinguir a extração de conhecimento e de regras, pois, nem sempre, o conhecimento obtido precisa, necessariamente, estar na forma de regras, apesar de acontecer comumente.

As principais vantagens relacionadas à utilização de técnicas de extração de regras como parte das operações realizadas por RNA, são:

- exploração de dados e dedução de novas teorias;
- facilitar a aceitação pelo usuário;
- melhora da generalização das soluções da rede;
- integração com sistemas simbólicos; e
- redefinição da rede.

## 3.5 – Agentes Inteligentes

### 3.5.1 - Define Agentes Inteligentes

Segundo Russel e Norvig (1995), um agente é uma entidade que possa perceber seu ambiente através de sensores e age sobre esse ambiente através de “efetadores”. A Figura 3.12 apresenta um esquema representando um agente genérico.

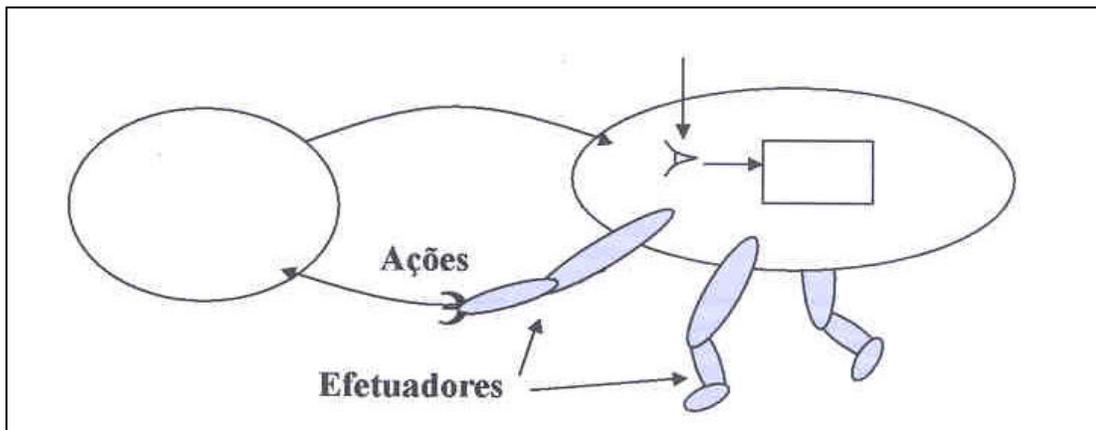


Figura 3.12 – Representação de um agente genérico.

Fonte: Russel e Norvig (1995, página 27)

Ainda, segundo Russel e Norvig (1995), um agente racional é aquele que realiza ações corretamente, ou seja, as que causarão maior sucesso ao agente, trazendo dificuldade na determinação de sua avaliação, em especial, quanto a decisão de *como* e *quando* avaliar o sucesso do agente.

É utilizado o termo Medida de Desempenho para o *como*, ou seja, o critério que determina o quanto o agente obteve de sucesso. Tal questão

detém uma subjetividade implícita. Alguns agentes, devido a suas peculiaridades, estariam impossibilitados de responder corretamente, assim, somente, observadores externos teriam a capacidade de estabelecer um padrão para a definição de sucesso em um ambiente.

O *quando* da avaliação do desempenho, também, é importante, pois poderemos medir o desempenho durante o tempo dedicado ao trabalho. Assim, torna-se importante distinguir a onisciência e a racionalidade de um agente.

Conforme Russel e Norvig (1995), um agente onisciente conhece o resultado real de suas ações, e pode agir de acordo; mas onisciência é praticamente impossível na realidade. Um agente racional deverá se preocupar com o sucesso esperado de acordo com o que foi percebido.

Em suma, o que é racional em qualquer momento dado depende de quatro coisas:

- a medida de desempenho que define o grau de sucesso;
- tudo que o agente conseguiu perceber até então, definido como seqüência de percepção;
- o que o agente sabe sobre o ambiente; e
- as ações que o agente pode executar.

Tais dependências levam a definição de um agente racional ideal: para cada possível seqüência de percepção, um agente racional ideal deve fazer qualquer ação que seja esperada para maximizar sua medida de

desempenho, com base na evidência fornecida pela seqüência de percepção e qualquer conhecimento embutido que o agente possua, ou seja, realizar ações para se obter informações úteis é uma parte importante da racionalidade.

### **3.5.1.1 - Mapeamento de seqüências de percepção**

Percebe-se que o comportamento de um agente depende, apenas, de sua seqüência de percepção até o estágio atual. Diante disso, pode-se descrever qualquer agente particular a partir de uma tabela de ações ou lista em resposta a cada seqüência de percepção possível. Tal lista é chamada de mapeamento das seqüências de percepção para ações. A especificação de qual ação um agente deve tomar em resposta a qualquer seqüência de percepção dada, fornece um projeto para um agente ideal.

Em um agente muito simples, como a função raiz quadrada numa calculadora, a seqüência de percepção é uma seqüência de acionamentos de teclas representando um número, e a ação é a apresentação do número na tela. Um mapeamento ideal é aquele que quando a percepção é um número positivo  $x$ , a ação correta é mostrar um número positivo  $z$ , tal que  $z^2 = x$ , com aproximação de 15 casas decimais.

Na Figura 3.13, a seguir, é apresentado parte do mapeamento ideal e um programa simples que implementa o mapeamento usando o método de Newton.

Percepção $x$	Ação $z$
1.0	1.000000000000000
1.1	1.048808848170152
1.2	1.095445115010332
1.3	1.140175425099138
1.4	1.183215956619923
1.5	1.224744871391589
1.6	1.264911064067352
1.7	1.303840481040530
1.8	1.341640786499874
1.9	1.378404875209022
...	...

```

function SQRT( $x$ )
   $z \leftarrow 1.0$ 
  repeat until  $|z^2 - x| < 10^{-15}$ 
     $z \leftarrow z - (z^2 - x) / (2z)$ 
  end
  return  $z$ 

```

Figura 3.13 – Parte do mapeamento ideal para um problema – Raiz Quadrada

Fonte: Russel e Norvig (1995, página 30)

### 3.5.1.2 - Autonomia

Na definição de um agente racional ideal, também, deve ser contemplado o “Conhecimento Embutido”. Se as ações de um agente são completamente baseadas em conhecimento embutido, de maneira que o agente não precise considerar suas seqüências de percepção, diz-se que tal agente carece de autonomia.

Um comportamento de um agente pode ser baseado na sua própria experiência e no conhecimento embutido utilizado na construção do agente para o ambiente particular em que opera.

Conforme Russel e Norvig (1995), um sistema é autônomo na medida em que seu comportamento é determinado por sua própria experiência. É razoável que se forneça um conhecimento inicial a um agente com inteligência artificial, assim como a habilidade de aprender.

### **3.5.2 - Estrutura de Agentes Inteligentes**

O trabalho de Inteligência Artificial é projetar um programa agente: uma função que implementa o mapeamento do agente e das percepções para as ações.

Segundo Russel e Norvig (1995), um agente é a concatenação de sua arquitetura e dos seus programas. Sendo arquitetura definida como algum dispositivo de computação. Obviamente, que a arquitetura deverá aceitar e executar o programa escolhido.

Segundo Maes (1991), um agente pode ser decomposto, pois sua construção é o resultado da concatenação de módulos componentes e suas respectivas interações. A total montagem dos módulos e suas interações têm de prover a resposta da questão que o sensor e/ou os estados internos do agente se baseiam para determinar a ação a ser executada. Segundo Wooldridge e Jennings (1999b), a arquitetura compreende as técnicas e algoritmos que suportam a tudo isto.

Em geral, a arquitetura realiza a percepção a partir de sensores disponíveis ao programa, executa o programa e alimenta as escolhas de ação do programa para os efetadores enquanto são gerados.

Segundo Russel e Norvig (1995), os programas agentes têm uma forma muito simples, conforme pode ser verificado na Figura 3.14. Cada agente utiliza uma estrutura de dados interna que será atualizada à medida que chegarem novas percepções. Estas estruturas de dados serão operadas pelos procedimentos de tomada de decisão do agente para gerar uma escolha de ação, que é, então, passada para a arquitetura para ser executada.

```

Function AGENTE-ESQUELETO(percepção) returns ação
  Static: memória – a memória do mundo do agente

  memória ? ATUALIZA-MEMORIA(memória, percepção)
  ação ? ESCOLHE-MELHOR-AÇÃO(memória)
  memória ? ATUALIZA-MEMORIA(memória, ação)

  Return ação

  *** Em cada inovação, a memória do agente é atualizada para refletir a nova percepção, a
  melhor ação é escolhida, e o fato de que a ação foi tomada é armazenada na memória. A
  memória persiste de uma inovação para outra.***

```

Figura 3.14 – O agente esqueleto.

Fonte: Russel e Norvig (1995, página 32)

Deve-se observar, ainda segundo Russel e Norvig (1995), na Figura 3.14, que se pode definir o mapeamento do agente como uma função da seqüência de percepções para ações, o programa agente recebe, apenas, uma percepção simples como entrada. É responsabilidade do agente construir a seqüência de percepções na memória. A medida de desempenho, no agente esqueleto, de estar aplicada externamente para julgar o

comportamento do agente, e é sempre possível alcançar um alto desempenho sem o conhecimento explícito da medida de desempenho.

### 3.5.3 - Mapeamento das percepções em ações

Para uma melhor avaliação do mapeamento, segundo Russel e Norvig (1995), consideraremos quatro tipos de programas agentes:

- agente de reflexo simples;
- agente que mantém registros do mundo;
- agente baseado em metas; e
- agente baseado em utilidade.

#### 3.5.3.1 - Agente de reflexo simples

Diante da impossibilidade de armazenar uma tabela de referência com todas as entradas/saídas possíveis e imagináveis, deve-se resumir armazenando certas ocorrências comuns de associações de entrada/saída. A partir do processamento de uma entrada, se pode estabelecer uma condição que ative alguma conexão/ação estabelecida no programa agente. Este tipo de conexão/ação é definido como **regra de condição-ação** ou **situação-ação** ou **produção** ou **se-então**, escrita da seguinte forma:

**SE percepção ENTÃO ação**

Humanos, também, têm muitas dessas conexões, algumas das quais são respostas aprendidas e algumas delas são reflexos involuntários. Na Figura 3.15 é mostrada a estrutura de um agente de reflexo simples de forma esquemática, mostrando como regras de condição-ação permitem ao agente fazer uma conexão entre percepção e ação. Na parte final da figura é apresentado, respectivamente, o seu programa agente, onde se observa que a função CASA-REGRAS retorna a primeira regra no conjunto de regras.

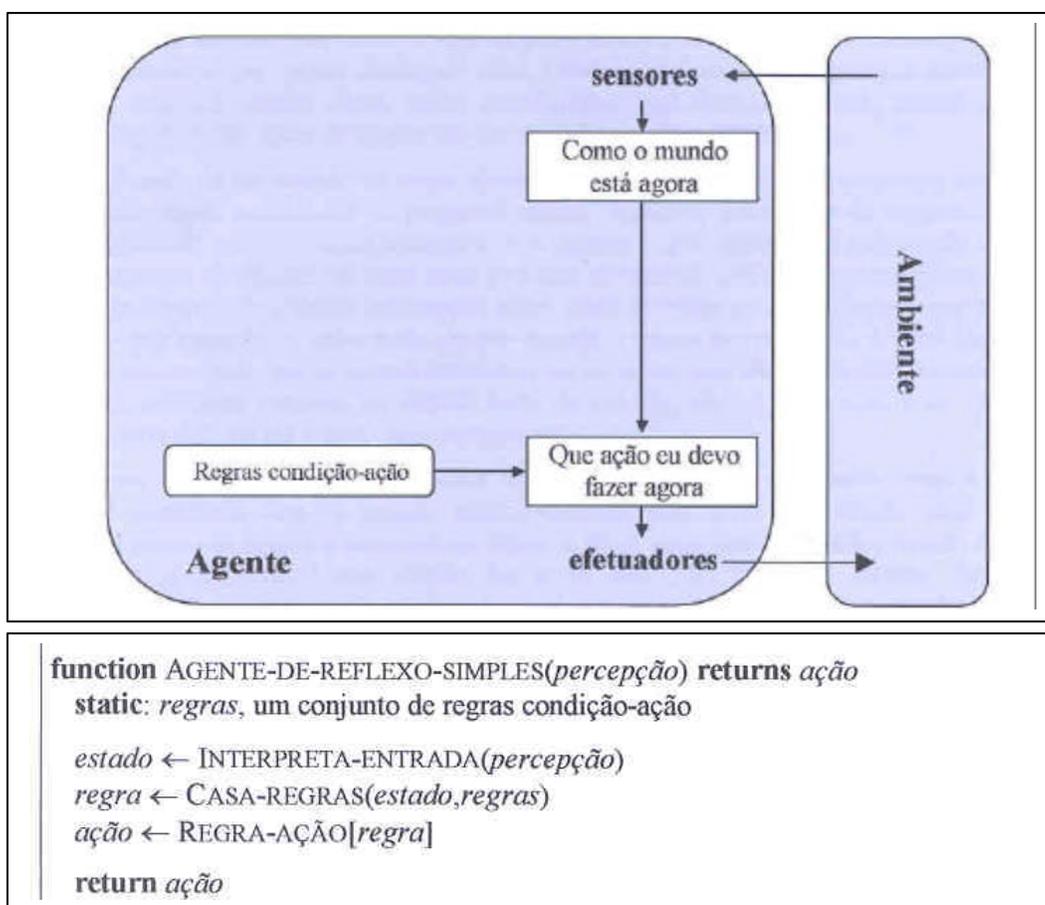


Figura 3.15 – Diagrama esquemático de um agente de reflexo simples

Fonte: Russel e Norvig (1995, página 35)

### **3.5.3.2 - Agente que mantém registros do mundo**

A necessidade desse tipo de agente justifica-se quando da necessidade de ter-se acesso ao estado completo do mundo para definir-se uma ação, e não, somente, quando da ocorrência de uma condição pré-definida, como no caso da regra condição-ação.

Esse tipo de agente precisará manter alguma informação de estado interno, de maneira que possa distinguir entre estados do mundo que geram a mesma entrada perceptiva, mas que, apesar disso, sejam significativamente diferentes, determinando ações diferentes.

A atualização da informação de estado interno exige que dois tipos de conhecimento sejam codificados no programa agente. Primeiro, será necessária alguma informação de como o mundo evolui independentemente dos agentes. Segundo, será necessária alguma informação sobre como as próprias ações dos agentes influenciam o mundo.

Na Figura 3.16 é apresentada a estrutura de um agente de reflexos, mostrando como a percepção corrente é combinada com os antigos estados internos para gerar a descrição atual do estado corrente. Na parte final da figura é apresentado, respectivamente, o seu programa agente, onde se

observa que a função ATUALIZA-ESTADO é a responsável pela criação das novas descrições de estado interno.

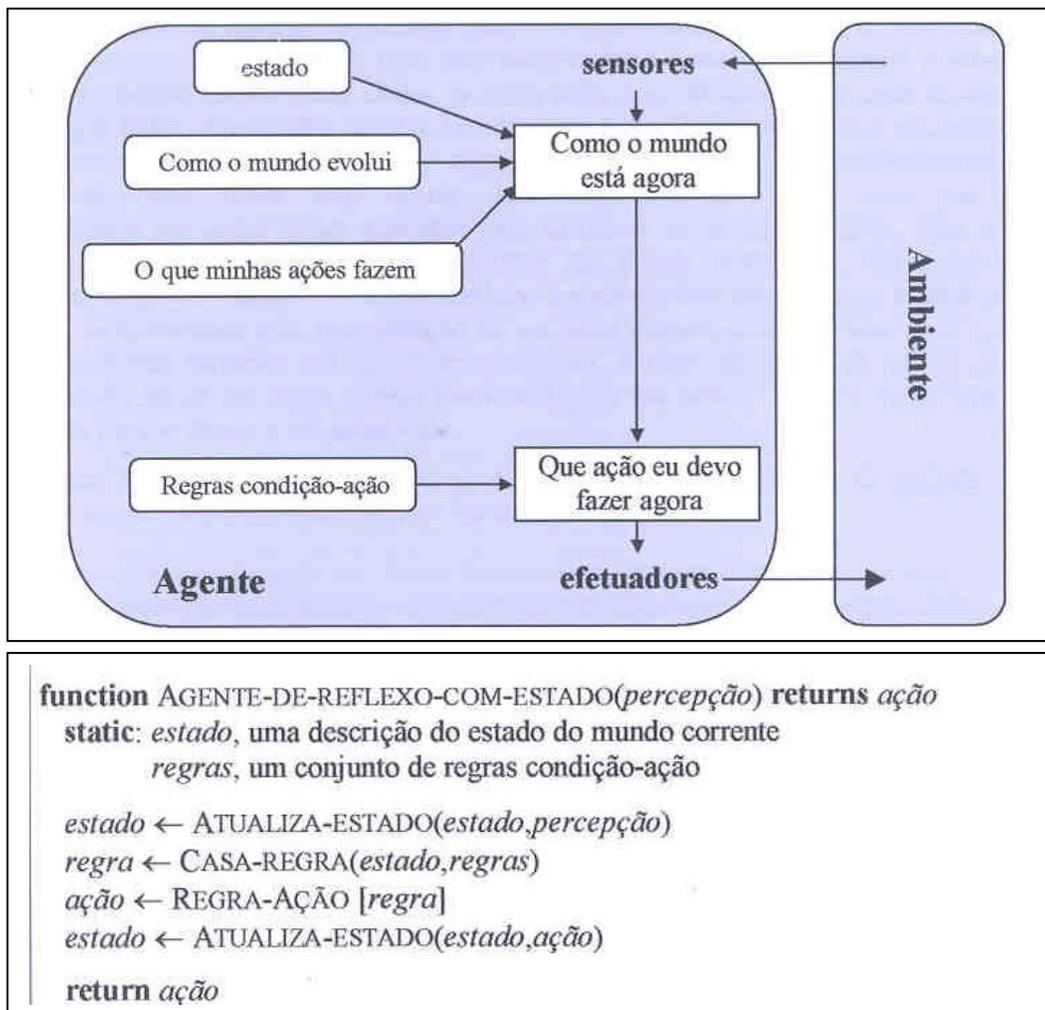


Figura 3.16 – Diagrama esquemático de um agente de reflexo com estado interno.

Fonte: Russel e Norvig (1995, página 37)

### 3.5.3.3 - Agente baseado em metas

O conhecimento sobre o estado atual do mundo nem sempre é suficiente para se decidir que ação executar. Assim, além da descrição do estado corrente, o agente pode precisar de alguma informação-meta, que descreveria situações que são desejáveis.

O programa agente pode combinar essas metas desejáveis com a informação sobre os resultados das ações possíveis, com o objetivo de escolher ações que atinjam a meta. Algumas vezes, isso será simples, quando a meta é atingida a partir da aplicação imediata de uma ação, outras vezes, não será tão simples, quando o agente tiver que considerar longas seqüências de mudanças e voltas para encontrar o resultado esperado.

Note que a tomada de decisão desse tipo é, fundamentalmente, diferente das regras condição-ação já descritas, pois envolve consideração do futuro, com as possíveis indagações: “o que vai acontecer se eu fizer isto?” ou “isto satisfará?”.

O agente baseado em metas é bem mais flexível, apesar de parecer menos eficiente. Essa flexibilidade, também, é registrada com respeito a alcançar destinos diferentes, podendo, inclusive, alterar seu comportamento.

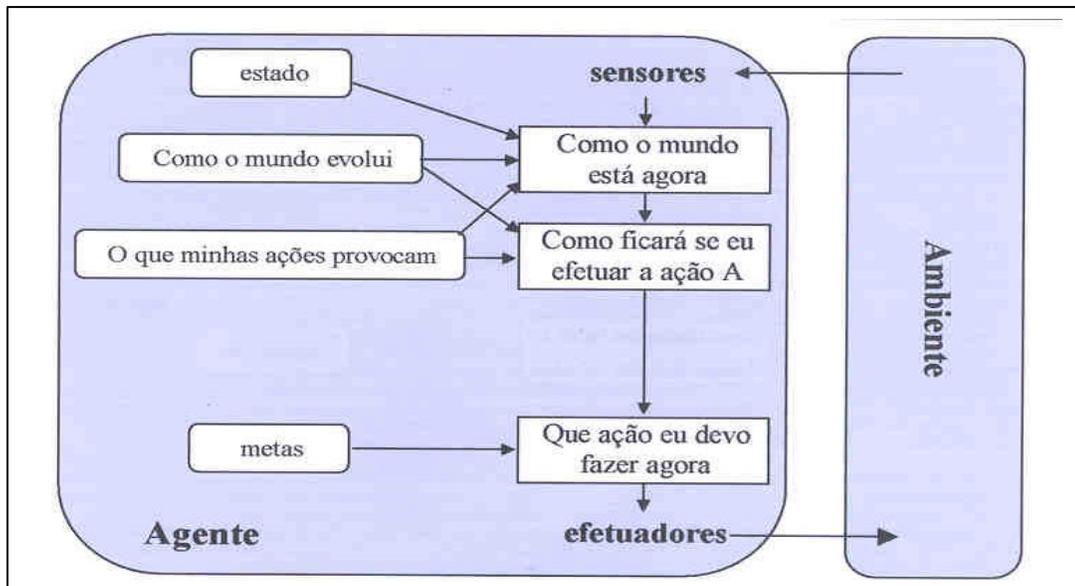


Figura 3.17 – Diagrama esquemático de um agente com metas explícitas.

Fonte: Russel e Norvig (1995, página 38)

### 3.5.3.4 - Agente baseado em utilidade

As metas, por si só, podem não ser suficientes para gerar comportamento de qualidade. Metas fornecem, apenas, uma distinção crua entre os estados “satisfeito” e “insatisfeito”, enquanto uma medida de desempenho mais geral pode permitir uma comparação entre diferentes estados do mundo, de acordo com o grau de satisfação do agente, se tivesse alcançado sua meta. A terminologia mais apropriada para este caso é dizer que se um estado do mundo tem preferência sobre outro, então maior utilidade para o agente ele apresenta.

Utilidade é, entretanto, uma função que mapeia um estado em um número real, que descreve o grau de satisfação associado. Uma especificação completa da função de utilidade permite decisões racionais em

dois tipos de casos onde as metas apresentam problemas. Primeiro, quando existem metas conflitantes, apenas, algumas delas podem ser alcançadas, assim, a função utilidade especifica a negociação apropriada. Segundo, quando existem várias metas que o agente pode almejar, e quando nenhuma delas pode ser alcançada com certeza. A função utilidade fornece uma maneira pela qual o agente, diante desse fato, pode satisfazer-se com o que foi possível alcançar.

Exemplificando, um agente que possui uma função de utilidade explícita e que então pode tomar decisões racionais, mas terá que comparar as utilidades alcançadas por diferentes cursos de ações. Em alguns casos, a função utilidade pode ser traduzida em um conjunto de metas, tal que as decisões tomadas por um agente baseado em metas sejam idênticas àquelas tomadas por um agente baseado em utilidade.

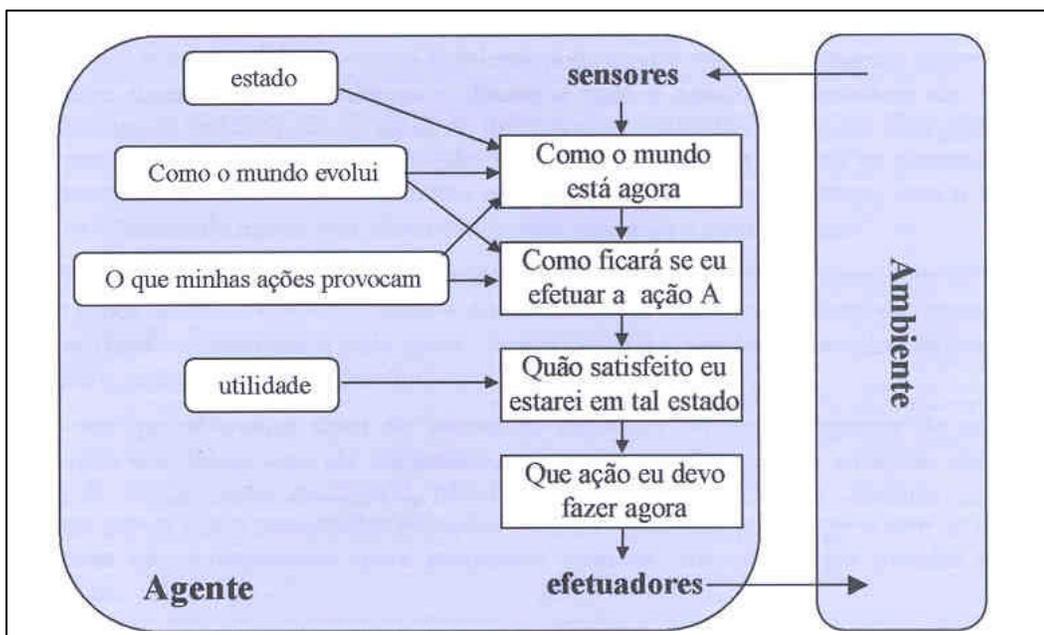


Figura 3.18 - Diagrama esquemático de um agente baseado em utilidade.

Fonte: Russel e Norvig (1995, página 39)

### 3.5.4 - Define Ambiente

Baseado no que foi citado anteriormente, se pode afirmar que as ações são realizadas pelo agente sobre o ambiente, que fornece estímulos em turnos aos agentes. Segundo Russel e Norvig (1995), existem vários tipos de ambientes que afetam de forma específica o desenvolvimento dos agentes. São eles:

- **Acessável:** Se os sensores de um agente dão acesso ao estado completo do seu ambiente, então diz-se que este ambiente é acessível ao agente,. É conveniente, porque o agente não precisa manter qualquer estado interno para manter a informação sobre o mundo;
- **Determinístico:** Se o próximo estado do ambiente é determinado completamente pelo estado atual e as ações selecionadas pelos agentes, diz-se que o ambiente é determinístico;
- **Episódico:** Neste tipo de ambiente, a experiência do agente está dividida em episódios. Cada episódio consiste na percepção e consequente ação do agente. A qualidade da ação depende, apenas, do episódio em si, porque episódios subseqüentes não dependem de que ações ocorram em episódios anteriores;
- **Estático/Dinâmico:** Se o ambiente pode mudar enquanto o agente estiver atuando, então diz-se que o ambiente é dinâmico para o

agente, do contrário ele é estático. Ambientes estáticos são fáceis de se tratar porque o agente não precisa ficar observando o mundo enquanto toma uma decisão sobre uma ação, nem precisa se preocupar com a passagem do tempo; e

- **Discreto/Contínuo:** Se existe um número distinto, definido, claramente, de estímulos e ações, diz-se que o ambiente é discreto. O jogo de xadrez é um exemplo típico de ambiente discreto, pois existe um número fixo de movimentos possíveis a cada turno.

Diferentes tipos de ambientes requerem diferentes programas agentes para obter-se uma maior eficiência.

Na Figura 3.19 é apresentada a relação básica entre os agentes e o ambiente. Este simulador toma um ou mais agentes como entrada e os organiza repetidamente, dando o estímulo correto a cada um deles e recebendo a ação como resposta. A simulação, então, atualiza o ambiente baseado nas ações e, possivelmente, outros processos dinâmicos no ambiente que não são considerados como agentes, por exemplo, a chuva.

```

Function EXECUTA-AMBIENTE(estado, ATUALIZA-FN, agentes, terminação)
  Entradas:      estado – o estado inicial do ambiente
                  ATUALIZA-FN – função para modificar o ambiente
                  Agentes – um conjunto de agentes
                  Terminação – um predicado para testar quando

  For each agente in agentes do
    PERCEPÇÃO[agente] ? RECEBE-PERCEPÇÃO(agente,estado)
  End
  For each agente in agentes do
    AÇÃO[agente] ? PROGRAMA[agente](PERCEPÇÃO[agente])
  end

  regra ? CASA-REGRAS(estado,regas)

  Return terminação(estado)

  *** Para cada agente fornece sua percepção, recebe uma ação e então atualiza o ambiente***

```

Figura 3.19 – Programa simulador de um ambiente básico.

Fonte: Russel e Norvig (1995, página 41)

No exemplo apresentado na Figura 3.19 que não é possível medir a performance do agente. Para tanto, faz-se necessário colocar um programa medidor específico.

Na Figura 3.20 é apresentado um programa simulador com uma medida de desempenho para cada agente.

```

Function EXECUTA-AVALIAÇÃO-DE-AMBIENTE(estado, ATUALIZA-FN, agentes,
                                     terminação, DESEMPENHO-FN) returns scores
    Local variables: scores – um vetor do mesmo tamanho que agentes, todos 0

    Repeat
      For each agente in agentes do
        PERCEPÇÃO[agente] ? RECEBE-PERCEPÇÃO(agente,estado)
      End
      For each agente in agentes do
        AÇÃO[agente] ? PROGRAMA[agente](PERCEPÇÃO[agente])
      end

      estado ? ATUALIZA-FN(ações,agentes,estado)
      scores ? DESEMPENHO-FN(scores,agentes,estado)

    until terminação(estado)

    Return scores

```

Figura 3.20 – Programa simulador de um ambiente básico com medida de desempenho.

Fonte: Russel e Norvig (1995, página 42)

## 4. APLICAÇÃO

### 4.1 – Aplicações correlatas

Apresenta-se, a seguir, o resultado de extensa pesquisa sobre publicações relativas a aplicações correlatas a este estudo, ou seja, de técnicas de IA em processos organizacionais.

#### 4.1.1 - *Agent-Oriented Software Engineering*

A publicação de Wooldridge e Jennings (1999a) propõe a implementação de uma coleção de interação de agentes autônomos, ou seja, de um sistema multi-agentes, na análise, projeto e implementação de software, possibilitando um promissor ponto de partida para a construção de software.

Um sistema orientado a agente é definido como um sistema encapsulado, que a partir do ambiente e da capacidade de flexibilidade, executa ações no ambiente de forma a atender aos objetivos do projetista.

A primeira contribuição apresentada é a importância dos sistemas multi-agentes na futura geração de software e o quanto isso representa no salto em conhecimento. A segunda é o registro do impacto no ciclo de vida da construção do software quando se adota um agente-orientado.

O papel do construtor de software é prover estruturas e técnicas que simplifiquem a complexidade inerente a esta construção. Os principais mecanismos para se gerenciar essa complexidade são:

- **Decomposição:** Técnica básica de dividir um grande problema e pequenas partes, gerenciáveis e isoladas. A decomposição auxilia por limitar o escopo do projetista, definindo a porção do problema que necessita ser considerado;
- **Abstração:** Processo de definição de um modelo simplificado de um sistema que tem muitos detalhes e propriedades. Esse mecanismo auxilia, pois limita o escopo de interesse em um determinado tempo. A atenção pode ser enfocada em aspectos importantes do problema, que mereça ser detalhado com maior relevância; e
- **Organização:** Processo de identificação e gerenciamento dos inter-relacionamentos entre os vários componentes necessários para a resolução do problema.

Na Figura 4.1 é apresentada uma visão que engloba os principais mecanismos citados anteriormente.

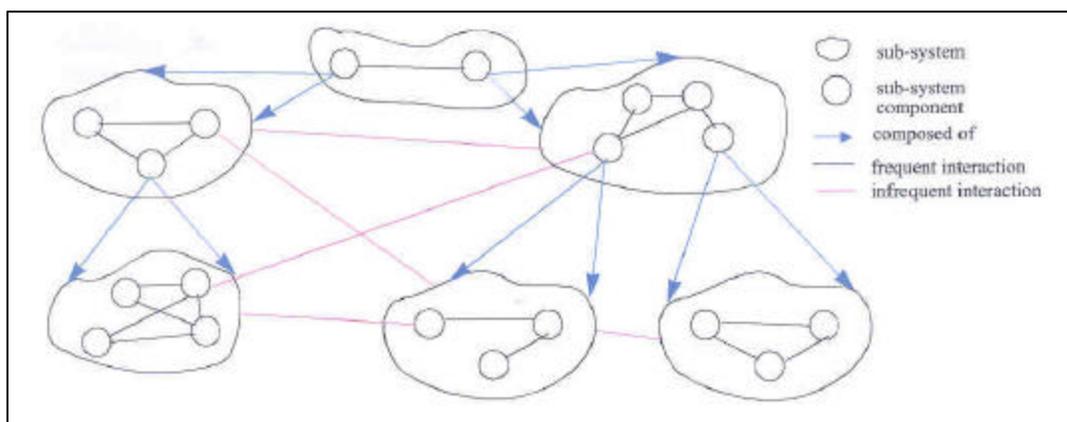


Figura 4.1 – Visão de um sistema complexo legal.

Fonte: Wooldridge e Jennings (1999a, página 3)

#### 4.1.2 - Um agente de processos de negócios

A publicação de Debenham (2000) propõe a utilização de um sistema multi-agentes inteligente com a finalidade de construção de processos de gerenciamento de negócios.

Um sistema multi-agentes é a associação de autônomos trabalhando cooperativamente, de forma a manter uma interação com o ambiente. Os agentes inteligentes poderiam ser autônomos, cooperativos ou adaptativos. Desta forma, uma arquitetura de agentes pode ser projetada especificamente para um aplicação de processo de negócios.

Citando Wooldridge e Jennings, indicou-se a tecnologia de agentes inteligentes para a aplicação na área de gerenciamento de processos de negócios, frisando que os mesmos referem-se a processos não pré-

definidos, e que, usualmente, não fazem parte de uma rotina natural e que necessitam de alto grau de iniciativa do sistema para atingir sua conclusão.

Qualquer atividade deve ter um plano para a sua conclusão, também, denominada *goal*. A condição de sucesso (SC) é definida como o procedimento que, quando executado, tenta alcançar o *goal*. A condição de sucesso (SC) é um procedimento, a execução desse procedimento pode ser bem sucedida (v), falha (?) ou abortada (A). Se a execução da condição de sucesso resultou em uma opção não planejada, esse resultado é denominado desconhecido (?). A Figura 4.2 esquematiza esse plano.

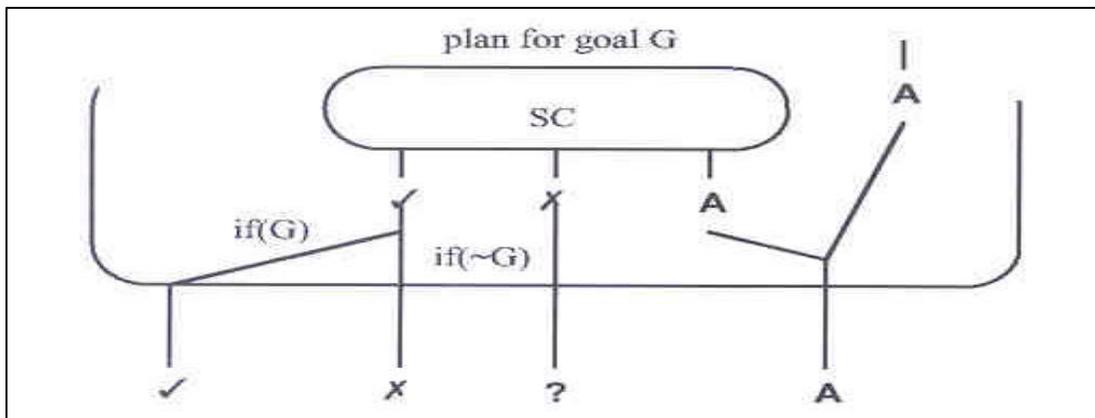


Figura 4.2 – Plano de uma condição de sucesso.

Fonte: Debenham (2000, página 2)

Sabemos que no ambiente real, em qualquer momento, um agente poderia invocar um procedimento não-determinístico, sendo que para sua execução seria necessária a construção de um agente adaptativo. Neste caso, somente, através da utilização da probabilidade, poderíamos ter uma

medida de desempenho. Na publicação citada, é demonstrada, matematicamente, a determinação da medida de performance.

#### **4.1.3 - Modelo e metodologia de IA em um projeto de processo de negócio.**

A publicação de Koubarakis e Plexousakis (1999) propõe a utilização de IA e dos conceitos que permitem uma análise do negócio, como objetivos, *goals*, papéis, atores, ações, processos, responsabilidades e restrições, para capturar conhecimento sobre um empreendimento, de forma intuitiva ou formal.

O problema da representação, análise e gerenciamento do conhecimento de uma organização e seus processos é de alta relevância na aplicação de IA no projeto de processos de negócio, análise e até da reengenharia do processo. IA pode prover tecnologia para representação e automação do processo, além de ferramentas que suportam processo de reengenharia.

A representação formal de negócios, sua organização e seus processos, usando técnicas de IA, é bastante conhecida e proposta por diversos autores, sendo inserida no trabalho em uma metodologia proposta pelos autores da publicação.

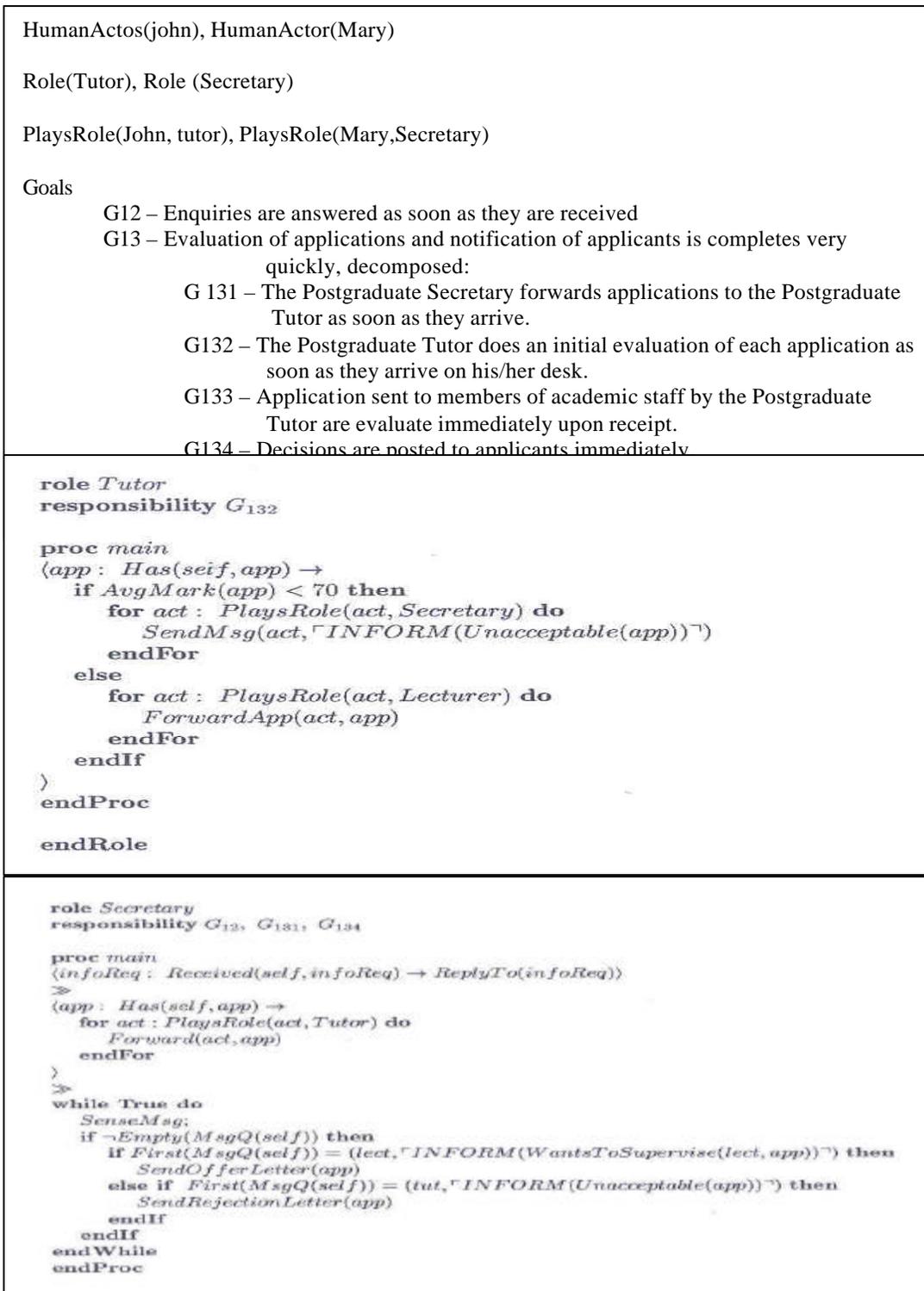


Figura 4.3 – Representação formal de um processo de negócio

Fonte: Koubarakis e Plexousakis (1999, página 11)

#### **4.1.4 - Integração do Conhecimento para construção de memória organizacional**

A publicação de Reimer (1999) propõe a utilização de IA para formalizar o conhecimento da memória organizacional.

Segundo Porter (1999), atualmente, o conhecimento é o mais importante bem das organizações. Um avançado gerenciamento do conhecimento requer que armazenemos, na maioria das vezes, em um repositório central, a memória corporativa ou organizacional.

A memória organizacional (OM) tem dois papéis principais. O primeiro, é armazenar as funções e ações passivas. Isto pode ser requerido por um usuário que necessite de uma informação específica. O segundo papel, é a adoção de um sistema ativo que dissemine o conhecimento para todos que necessitem.

O sistema ativo deve ser do tipo baseado no conhecimento, isto é, tem seu conhecimento, explicitamente, representado em uma base de conhecimentos, ou em um sistema de gerenciamento documental com palavras chaves e/ou componente de recuperação textual.

Como exemplo, é apresentada a arquitetura do sistema EULE2, Figura 4.4, sistema baseado no conhecimento que suporta o trabalho de escritório:

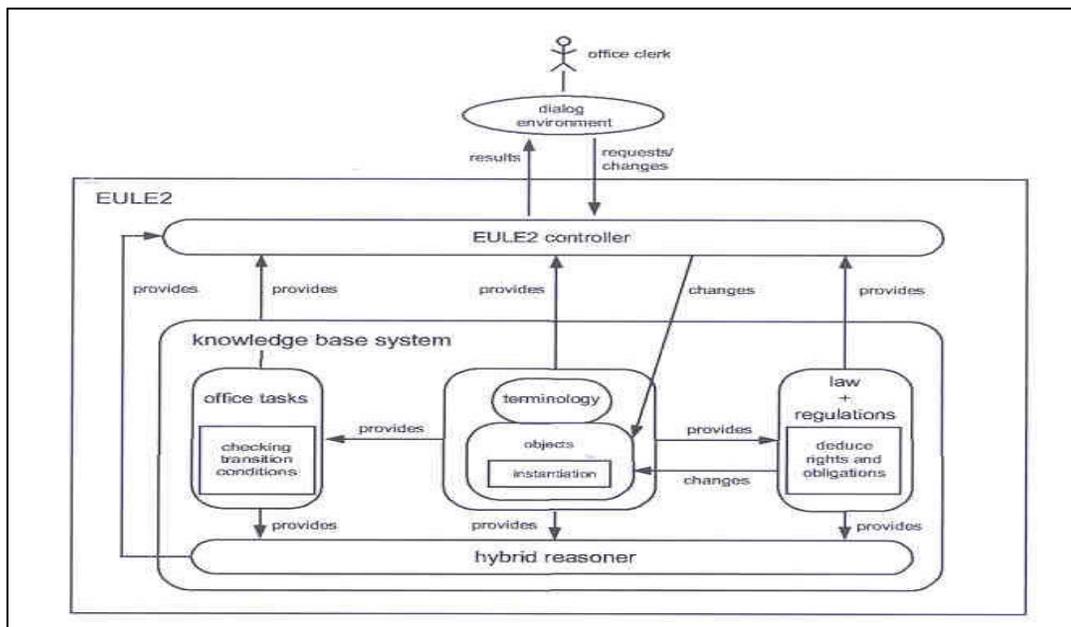


Figura 4.4 – Arquitetura do Sistema EULE2

Fonte: Reimer (1999, página 4)

A funcionalidade do Sistema EULE2 requer a representação de:

- tarefas de escritório, principalmente, consistidas de:
  - uma ordenação parcial de ações;
  - para cada ação o efeito dela.;
- instâncias para manipulação e concepção; e
- todas as leis e regulamentações que devem ser obedecidas nas tarefas do escritório.

Cada um dos três tipos de conhecimentos requer uma representação formal própria. A representação do conhecimento sobre as tarefas do escritório deve ser representada por um gráfico espaço-estado com

condições de ordenação de transição. A representação do conhecimento sobre a concepção e instâncias deve ser construída em uma terminologia lógica. A representação do conhecimento sobre leis e regulamentos deve ser codificada sobre cláusulas lógicas, onde se distingue a integridade de restrições que não podem ser violadas e regras dedutivas que derivem novos valores atribuídos.

O maior problema da solução EULE2 é a integração de diversas representações de conhecimento com diferentes níveis de formalização, conforme apresentado na Figura 4.5, devendo, assim, priorizar as bases de conhecimento a serem mantidas pelo sistema.

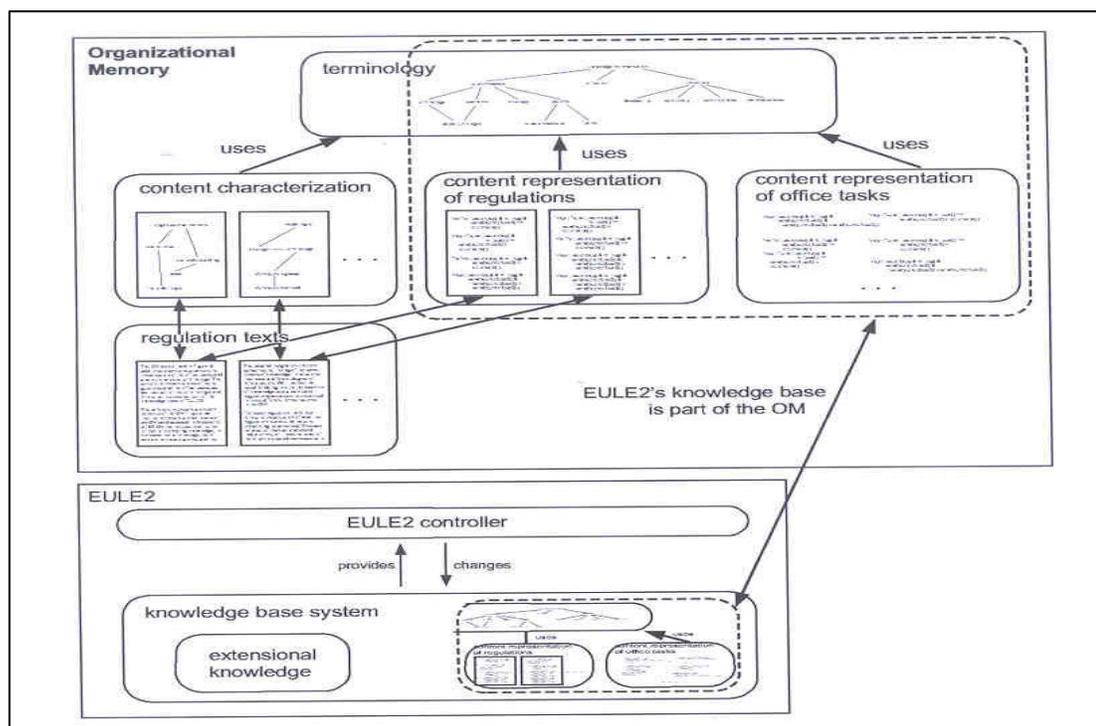


Figura 4.5 – A arquitetura de uma memória organizacional.

Fonte: Reimer (1999, página 14)

A utilização do conhecimento na construção da memória organizacional completa-se com a integração dessa memória com um sistema *Workflow*.

Um sistema *Workflow*, conforme já explanado anteriormente, coordena tarefas onde mais de um ator é envolvido. Conhecer-se anteriormente, as sub-tarefas que cada tipo de ator deve executar e gerenciar-se o fluxo de controle entre essas pessoas, possivelmente acompanhados de documentos eletrônicos, torna-se mais simples a construção da memória organizacional.

Possivelmente, a memória organizacional, implementada neste caso com o sistema EULE2, deve possuir mais conhecimento sobre as tarefas dos escritórios, sobre as leis relevantes e sobre os regulamentos da companhia. Assim, este conhecimento deve ser aproveitado pelo Sistema de *Workflow*, conforme apresentado na Figura 4.6.. Na Figura, entende-se HLL como sendo a Linguagem de Representação de Alto Nível, contida no sistema EULE2.

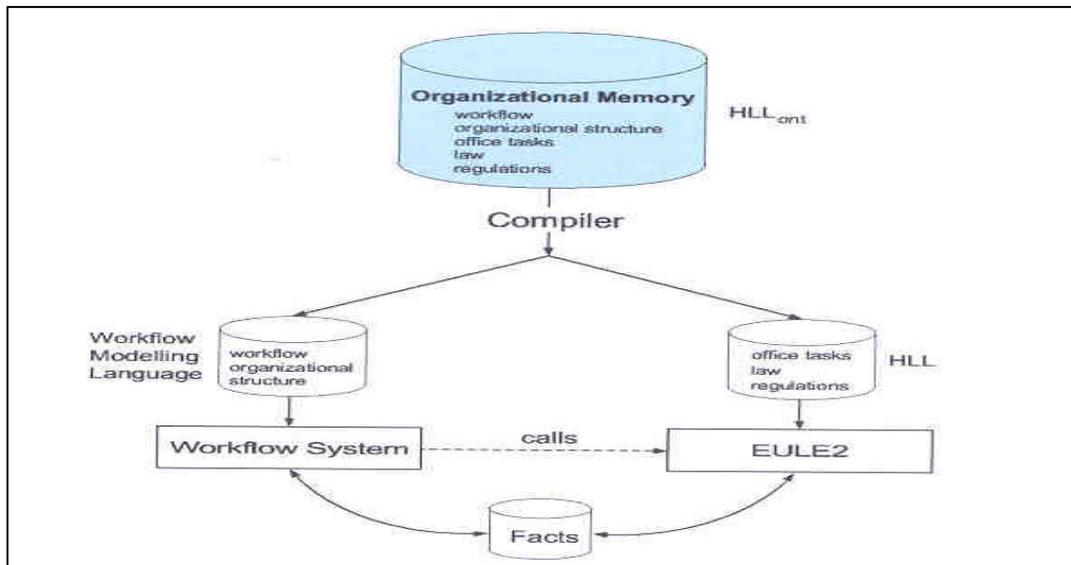


Figura 4.6 – Integração do sistema *Workflow* com o sistema EULE2.

Fonte: Reimer (1999, página 17)

Para ficar mais clara a integração dos dois sistemas, na Figura 4.7, detalhou-se essa cooperação.

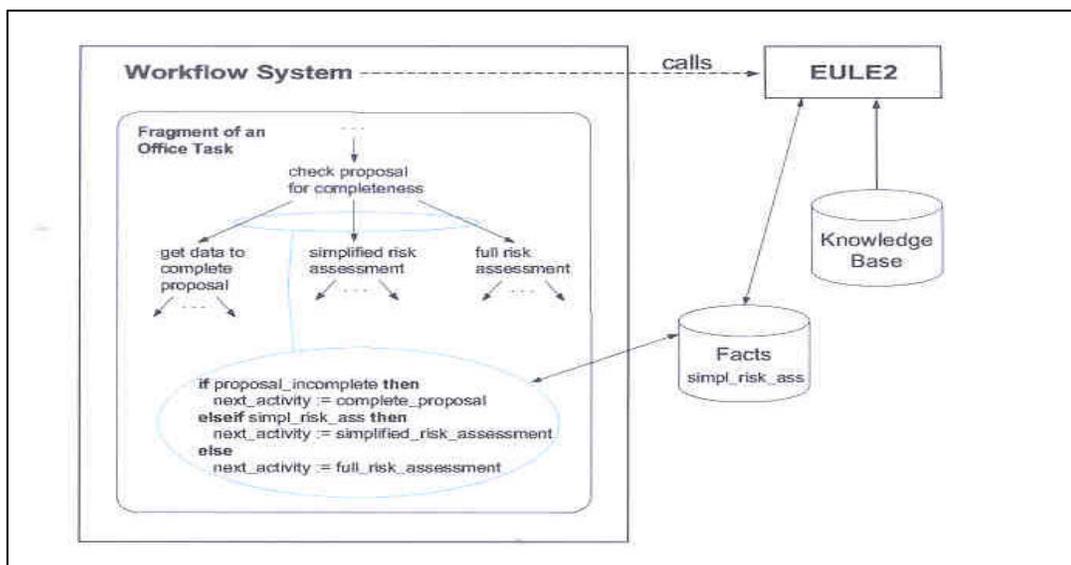


Figura 4.7 – Detalhamento da integração do sistema *Workflow* com o sistema EULE2.

Fonte: Reimer (1999, página 18)

### **4.1.5 - Suporte Inteligente para modelagem de empreendimentos**

A publicação de Stader e Jarvis (1998) propõe a integração da modelagem de empreendimentos, tão necessária ao gerenciamento das mudanças das organizações, com técnicas de IA, desde a representação do conhecimento, ontologia e técnicas de visualização de modelagem de processos, sistemas *workflow* inteligente e tecnologia de coordenação. As técnicas podem ser combinadas em um ambiente integrado sobre uma arquitetura baseada em agente.

O resultado do trabalho é a combinação de mecanismos de suporte de gerenciamento de tarefas com modelos de empreendimentos, diretamente, controlados pela operação da organização.

Os requisitos para a modelagem de empreendimentos são:

1. **integração** da informação obtida por diferentes visões do empreendimento; da informação das tarefas de suportam as ferramentas de negócio; e das conexões estabelecidas pelas ferramentas;
2. **comunicação** em diferentes níveis: entre as pessoas, que asseguram os modelos de empreendimento da organização; entre as pessoas, que asseguram alguma contribuição nas tarefas comuns, entre as tarefas que a informação pode ser relevante; e

entre as ferramentas usadas para desempenhar as tarefas, com dados relevantes que devem ser disponibilizados;

3. **flexibilidade**, sendo importante para permitir que as organizações se adaptem as mudanças do ambiente, dos processos e do uso de ferramentas; e
4. **suporte** para cuidar de detalhes técnicos e garantir que a flexibilidade não resulte em confusão, e que os processos sejam executados efetivamente.

Diferentes aspectos de uma organização e do ambiente devem ser contemplados pela modelagem de empreendimentos:

- a estrutura da organização;
- os papéis e responsabilidades das pessoas que compõem a organização;
- os processos que suportam a organização;
- a informação que é usada;
- o fluxo da informação;
- a capacidade requerida ou disponível;
- os artefatos que são produzidos;
- o mercado que a organização opera;
- etc.

Em IA, a integração de ferramentas de software tem nos sistemas baseados em agentes, sua área mais interessada neste tema.

Na Figura 4.8 é apresentada uma arquitetura de componentes de empreendimento composta de:

- *Procedure Builder* ou construtor de procedimento para capturar modelos de processos;
- *Agent Toolkit* ou ferramenta para desenvolvimento de agentes;
- *Task manager* ou gerenciador de tarefas para integrar, visualizar e suportar as tarefas. Utiliza técnicas do *Workflow* inteligente para chamar e usar os agentes; e
- *Enterprise Ontology* ou ontologia de empreendimento para a comunicação.

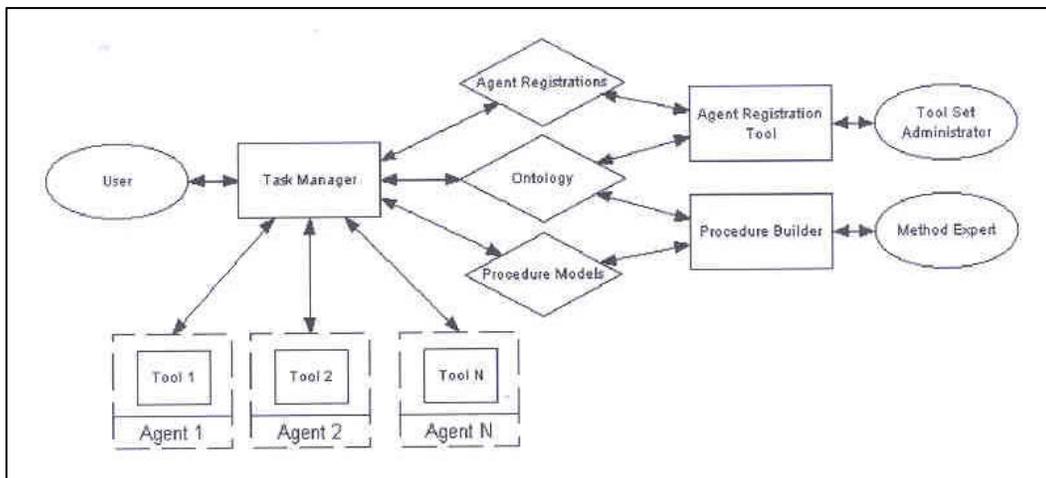


Figura 4.8 - Arquitetura de componentes de um empreendimento.

Fonte: Stader e Jarvis (1998, página 5)

#### 4.1.6 – Ontologia para suportar o gerenciamento de desenvolvimento de novos produtos na indústria química

A publicação de Moore *et al* (1999) propõe método de desenvolvimento ontológico para que sistemas de *Workflow* inteligente possam suportar o gerenciamento de processos dinâmicos, por exemplo: o desenvolvimento de novos produtos na indústria química.

Citando Uschold e Gruninger, expõe as características chaves da ontologia como:

- a ontologia de vários domínios de interesse identifica e descreve precisamente as concepções do domínio e os relacionamentos entre as concepções descritas;
- determina importantes termos e essas definições estão de acordo com todos os participantes do domínio;
- a ontologia é utilizada para especificar independentemente da aplicação foco do desenvolvimento, ou seja, reutiliza em outras aplicações de outros domínios; e
- a ontologia pode formalizar o suporte a comunicação entre sistemas da tecnologia da informação como a comunicação entre humanos.

A ontologia identifica relevantes conceitos do sistema *Workflow* Inteligente para trabalhar processos:

- **Artifacts ou artefatos:** Componente físico e equipamento que compõe a planta de desenvolvimento;

- **Information ou informação:** Tipos de informações que são usadas e transferidas durante o trabalho do processo;
- **Tasks ou tarefas:** Descrição das tarefas e processos que compõe o trabalho do processo;
- **Organisational Structure ou Estrutura Organizacional:** Estrutura organizacional afetada pelo processo. Inclui a especificação dos atores envolvidos no processo;
- **Capabilities ou Capacidade:** Descrição do domínio específico dos participantes do processo;
- **Agents ou Agentes:** Descrição dos participantes do processo, sejam humanos ou sistemas de agentes inteligentes; e
- **Computer Support ou Suporte Computacional:** Infra-estrutura computacional, hardware e software, que participa do processo.

O relacionamento entre esses conceitos é apresentado na Figura 4.9, a seguir:

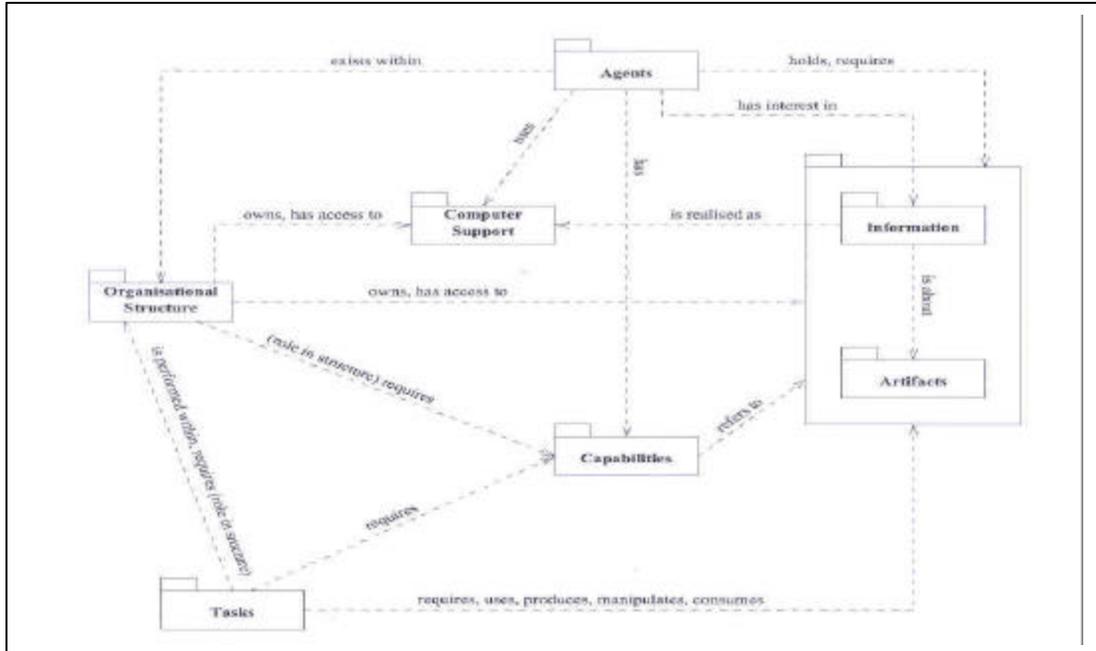


Figura 4.9 – Relacionamento entre Ontologias.

Fonte: Moore *et al* (1999, página 4)

#### 4.1.7 – Estrutura de satisfação restrita

A publicação de Nareyek (1999) propõe a utilização da técnica satisfação restrita para automatizar um sistema *Workflow*, em especial, de um caso de planejamento contínuo.

A programação restrita enfoca problemas combinatórios complexos. Esse tipo de situação é também presenciado alguns sistemas *Workflow*, em especial, aqueles que têm que contemplar todos os tipos de restrições de forma integrada, como por exemplo: o tempo, o espaço ou restrição de recursos das atividades planejadas.

O problema de satisfação restrita consiste na definição de variáveis  $X = \{X_1, X_2, \dots, X_n\}$ , onde cada variável é associada a um domínio  $D_1, D_2, \dots,$

**Dn** e com restrições  $C = \{ C1, C2, \dots, Cn \}$  de cada variável. As restrições podem ser relações entre as próprias variáveis.

O uso de variáveis com restrição possibilita o planejamento em tempo-real do sistema *Workflow*, e possibilita um mecanismo para inclusão de conhecimento com dependência de domínio. A aplicação, assim, estará integrada de forma on-line com as mudanças do ambiente, devido a sua atualização contínua.

#### **4.1.8 – KPM – Knowledge-Based Process Management**

A publicação de Lander *et al* (2000) descreve o KPM<sup>tm</sup>, como um ambiente baseado em conhecimento para projetar, gerenciar e executar processos que integram aspectos de modelagem de processos, sistema *Workflow* e gerenciamento de processos.

O KPM é aconselhado, especificamente, para domínios de projetos dinâmicos e, naturalmente, incertos, situações não contempladas pelos tradicionais sistemas *Workflow*. O KPM aplica várias tecnologias de IA, incluindo heurística dinâmica e reprogramação de processos em execução.

O KPM foi desenhado balanceando a necessidade de flexibilidade de reação na execução dos processos que necessitam de conhecimento, de detalhamento e decisão pró-ativa, característica dos humanos.

O desenvolvimento de sistemas *Workflow* tem deixado, cada vez, a modelagem estática de processos, para ser utilizado em processos

complexos e de ambiente dinâmico, em especial de captura e gerenciamento de incerteza e mudanças.

Várias áreas de IA são relevantes para o sistema *Workflow* inteligente e o gerenciamento de processos, em especial a programação dinâmica para suporte, a execução de atividades com interações complexas e mutantes com o mundo externo.

A integração da IA e o *Workflow* tem sido buscada de forma intensa, em especial na interseção conseguida através da construção de agentes inteligentes, proporcionando um controle das incertezas comuns nos processos.

A estrutura do KPM detém mecanismos de regulação do estado do processo, buscando instâncias de processos similares, possivelmente, executados de forma concorrente no ambiente. A dinâmica e incerteza natural do processo de construção requerem uma solução que incorpore aspectos tradicionais da modelagem de processos, sistemas *Workflow* e tecnologias de gerenciamento de projetos aplicadas em conjunção com programação dinâmica, durante a execução do processo.

A programação dinâmica do KPM é baseada na programação genérica fundamentada em agentes inteligentes, também, denominados sistemas quadro-negro, que têm a capacidade de captar ordens, operações, recursos e definição de recursos, reservas e restrições variadas.

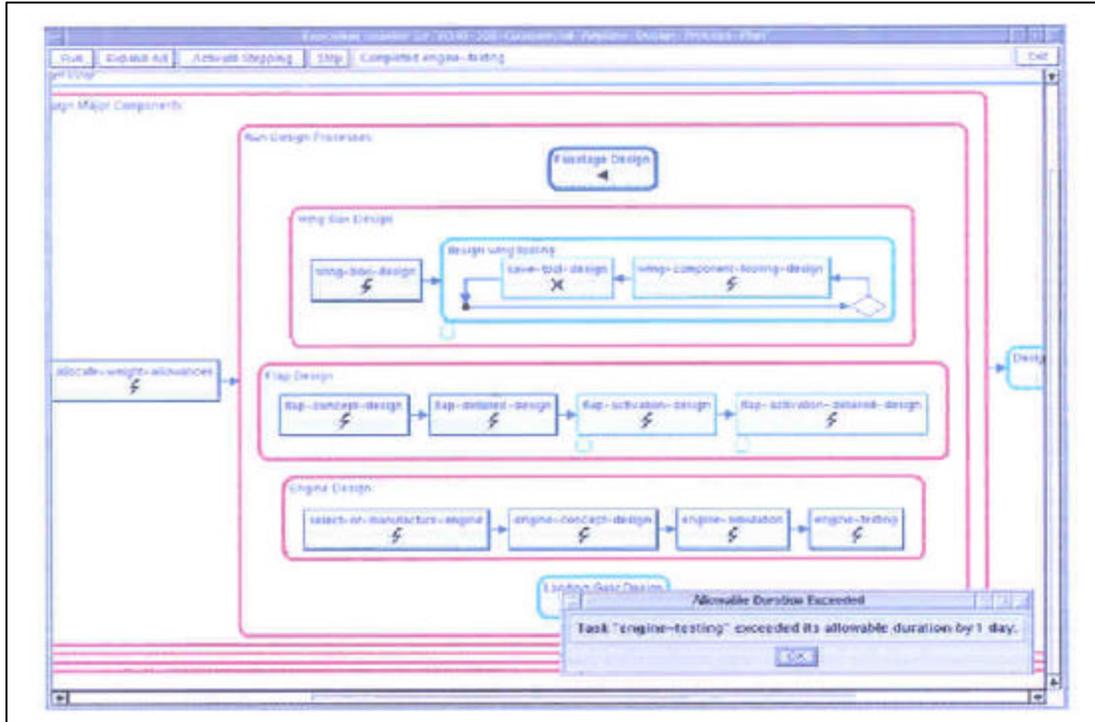


Figura 4.10 – Visão do monitor de execução do KPM.

Fonte: Lander *et al* (2000, página 6)

#### 4.1.9 – SWIM – Um sistema baseado em IA para gerenciamento organizacional

A publicação de Berry e Drabble (2000) descreve o SWIM<sup>tm</sup> Smart Workflow for Information Surveillance and Reconnaissance Management, como um sistema que projeta complexos sistemas, negócios e software, controlado pelo gerenciamento de um sistema *Workflow*.

Esse sistema explora recentes avanços da IA no controle reativo, programação e execução contínua. O SWIN estende aos sistemas *Workflow*

uma resposta a ambientes dinâmicos e incertos, para o controle de processos, envolvendo entidades dinâmicas.

O mercado, cada vez mais competitivo, utilizando a automação e utilizando a informação on-line, tem mostrado interesse na reengenharia e na automação dos processos de negócios. A comunidade dos sistemas *Workflow* advoga o uso explícito de modelos e representação de processos. Cada vez mais, há a necessidade de que os sistemas, qualquer que sejam eles, tenham flexibilidade e busquem caminhos inteligentes. Ao mesmo tempo, a comunidade da IA tem trabalhado com controles reativos para exploração de técnicas de gerenciamento de processos inteligentes, encontrando requisitos de adaptabilidade para ambientes dinâmicos e incertos.

O SWIM é um sistema multi-agente para controle e gerenciamento de tarefas complexas de ambientes dinâmicos e incertos. O sistema suporta os dois diferentes modelos de execução: **direta**, quando as ações do processo são executadas no próprio sistema e **indireta**, quando o supervisor da execução da ação submete a mesma a uma coleção de entidades distribuídas de execução.

A arquitetura do SWIN é compatível com a WfMC, mas com algumas características derivadas da aplicação da IA, sendo composta por:

- *Dinamic Process Manager* – responsável pela operação e comportamento global do sistema. Gerencia a criação de instâncias dos processos, sua decomposição dinâmica, sua consolidação e

atualiza as atividades do DEO – Dynamic Execution Order, cria o plano de monitoração e recupera as falhas;

- *Dynamic Process Selector* – responsável pela seleção, adaptação e evolução do processo;
- *Process Library Server* – responsável pela manutenção dos processos e dos construtores de blocos de processos;
- *Interface* – suporta a interação entre o usuário e o SWIM;
- *DEO Scheduler* – responsável pela alocação das atividades para processamento dos agentes;
- *Process Server* – provém um repositório para armazenamento dos múltiplos processos em organização;
- *Process Monitor* – responsável pela monitoração do estado de execução das atividades e processos; e
- *SIMFLEX Simulate Flexible Execution* – ambiente de simulação para teste, avaliação e demonstração da capacidade de gerenciamento do sistema *Workflow* dinâmico.

Na Figura 4.11 é apresentado o diagrama funcional da arquitetura do SWIM.

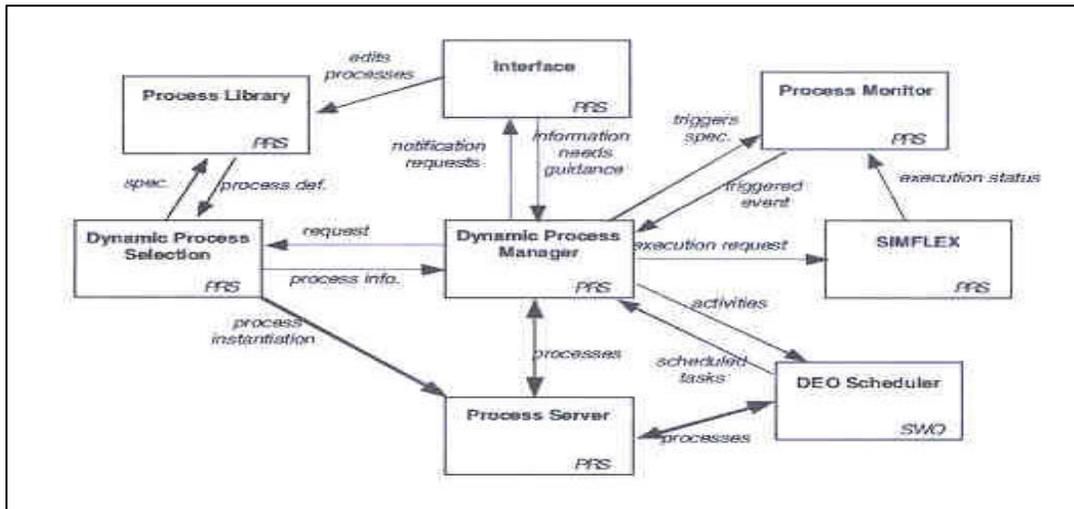


Figura 4.11 - -Arquitetura do SWIM.

Fonte: Berry e Drabble (2000, página 3)

#### 4.1.10 –Tecnologia para sistemas *Workflow* Inteligente: Experiências e Lições

A publicação de Odgers *et al* (2000) descreve a pesquisa realizada para a construção de um sistema *Workflow* Inteligente baseado em máquinas de aprendizado, decisão-ação, a partir de múltiplos critérios qualitativos e gerenciamento e resolução ontológica. Na três linhas de pesquisa, a área de IA, que suporta as soluções apresentadas, é a aplicação de agentes inteligentes.

Três classes de problemas impedem o crescimento da aplicação dos sistemas *Workflow*.

1. Informações Organizacionais: Os modelos de informação de produto e trabalho diferem em cada negócio, as regras do negócio, a estrutura organizacional e a divisão de responsabilidade;
2. Controle de comportamento descentralizado: O controle e monitoração centralizada dos sistemas *Workflow* são problemáticos quando o negócio é descentralizado; e
3. Balanceamento: A ferramenta de *Workflow* não ativa um fluxo de trabalho para ser gerenciado em um caminho pró-ativo.

#### **4.1.11 –Inteligência, flexibilidade e suporte real para processos de negócios**

A publicação de Kalläk *et al* (1998) descreve pesquisa combinando sistemas *Workflow*, orientação a objeto e técnica de construção de sistemas especialistas. O trabalho apresenta a força da integração entre processos de negócios, objetos de negócios e aplicabilidade na técnica baseada em conhecimento.

As diferentes bases de conhecimento devem ser gerenciadas em uma arquitetura de agentes inteligentes baseada na linguagem de manipulação e consulta de conhecimento – KQML. A aplicação específica de processos de negócios e suas regras devem ser representadas como dados, isto é, instância de objetos, e não em códigos.

A integração da tecnologia de desenvolvimento orientada a objeto, facilitando o desenvolvimento; o uso de armazenamento e recuperação de

conhecimento; e a aplicação da tecnologia *Workflow* criam um ambiente poderoso e integrado para a construção de sistemas que suportam um empreendimento mais efetivo que uma regular ferramenta de orientação a objeto, ou uma ferramenta de sistemas especialista, ou uma ferramenta *workflow*.

Na Figura 4.12 é apresentado o modelo conceitual para a relação entre ao gerenciamento *Workflow*, objetos de negócios e representação de regras de negócio, também denominado gerenciamento *Workflow* orientado a objeto.

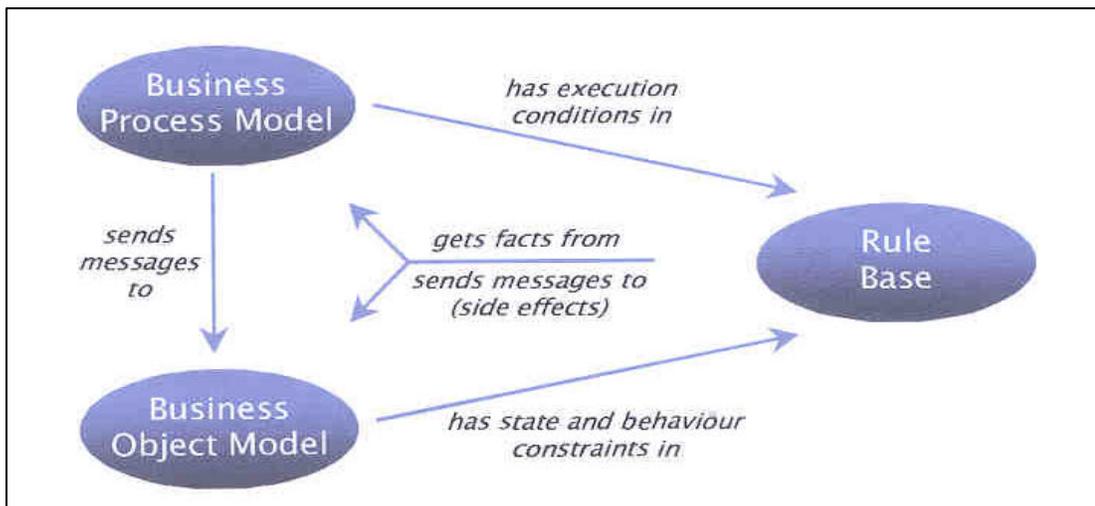


Figura 4.12 – Modelo conceitual do gerenciamento *Workflow* orientado a objeto.

Fonte: Kalläk *et al* (1998, página 5)

## 4.2 – Aplicação proposta – Ouvidoria Inteligente

Na documentação do SGO - Sistema de Gestão da Ouvidoria, em anexo a este trabalho, em seu Diagrama da Estrutura Processual da ANEEL – Agência Nacional de Energia Elétrica, é apresentada a estrutura por gestão de processos, na qual a ANEEL está implementada. No macroprocesso Planejamento e Gestão Administrativa nos subprocessos de nível 1, observa-se a existência da Gestão Técnica da Informação que, conforme Diagrama dos Subprocessos da Gestão Técnica da Informação, mantém um subprocesso de nível 2 denominado Desenvolver Sistemas de Informações, que tem em um de seus subprocessos de nível 3, o SGO ou, simplesmente, Ouvidoria.

É, exatamente, sob este sistema de informação, o subprocesso Ouvidoria, que está focada a aplicação proposta, denominada **Ouvidoria Inteligente**.

O subprocesso Ouvidoria compreende as atividades:

- Controle de Processos de Importação e Exportação de Dados das Bases Atento e Contax;
- Controle de Acesso de Usuários do SGO - Sistema de Gestão da Ouvidoria;
- Atendimentos a Usuários; e
- Estudo de Viabilidade de Incorporação de Rotinas e Análise da Estrutura do banco de Dados do SGO.

Conforme apresentado no referido Anexo, todas as atividades são descritas, definido seu respectivo procedimento de execução e diagramado os atores e sub-atividades envolvidas.

A proposta denominada **Ouvidoria Inteligente** propõe a aplicação de sistemas multi-agentes inteligentes, trabalhando de forma cooperativa, com a finalidade de manter uma interação com o ambiente, além de possibilitar uma integração com base de conhecimento.

Já, atualmente, o subprocesso Ouvidoria encontra-se implementado em um sistema *Workflow*. Por que não transformá-lo em um sistema *Workflow* inteligente, utilizando agentes inteligentes, conforme proposto anteriormente.

Na Figura 4.13 é apresentado o fluxo de trabalho definido pelo sistema *Workflow* para o SGO, desenvolvido no software Keyflow, ferramenta padrão da ANEEL para esse tipo de sistema.

Das aplicações correlatas estudadas, a que mais aproxima-se da transformação desejada é a publicada por Kalläk *et al* (1998), definindo o gerenciamento *Workflow* orientado a objeto.

A aplicação de sistemas multi-agentes inteligentes, trabalhando de forma cooperativa, com a finalidade de manter uma interação com o ambiente, possibilitaria a flexibilidade necessária para o tratamento de mudanças e incertezas. A integração com bases de conhecimentos possibilitaria aos sistemas *Workflow* uma definição, sem precedentes, de caminhos e ações, novamente, flexibilizando sua interação com o ambiente.



## 5. CONCLUSÕES E RECOMENDAÇÕES

A flexibilização necessária aos sistemas *Workflow* para o gerenciamento das mudanças e tratamento das incertezas, transformando em ***Workflow Inteligente***, é possível com o uso de multi-agentes inteligentes e com o armazenamento e recuperação de conhecimento.

A aplicação de sistemas *Workflow* inteligentes, ou baseado no conhecimento, em modelos empresariais repetitivos, reforça as conseqüências positivas do controle pró-ativo das atividades inerentes ao negócio.

A aplicação de técnicas de IA aos sistemas *Workflow* traz como principal contribuição a flexibilização, além do suporte real à característica de pró-atividade do sistema.

O caso proposto, Ouvidoria Inteligente, devido ao alto volume de transações, aproximadamente 6000 (seis mil) solicitações de ouvidoria diárias, necessita, realmente, de ter a propriedade de flexibilização da forma de tratamento, seja por uma recuperação de conhecimento, ou pela adaptabilidade às mudanças organizacionais.

A velocidade no tratamento, a certeza do caminho de atuação, além do ambiente volátil faz com que a intervenção humana seja preponderante, caso não sejam criados mecanismos inteligentes de tratamento a informação.

No caso de processos empresarias, há um consenso nas publicações estudadas do uso mais adequado de agentes inteligentes, conforme preconizado por Wooldridge e Jennings (1999b).

Somado a isto, a utilização de bases de conhecimento faz com que os sistemas *Workflow* tornem-se adaptativos.

Finalizando, recomenda-se a implementação em processos similares ao SGO, utilizando o gerenciamento *Workflow* orientado a objeto proposto por Kalläk (1998), construindo os modelos de objetos de negócios, a partir de linguagens orientadas a objeto, como o Java, tendo como finalidade a comprovação prática das conclusões aqui alcançadas.

## 6. BIBLIOGRAFIA

- Braga, A ., Ludermir, T. e Carvalho, A **Redes neurais artificiais – teoria e aplicações**. Rio de Janeiro: LTC, 2000.
- Berry, P. e Drabble, B. **SWIN: an AI-based system for workflow enabled reactive control**. Nova York: DARPA, 2000.
- Charniak, E. e McDermott, D. **Artificial Intelligence**. Nova York: Addison-Wesley, 1985.
- Chorafas, D. **Sistemas Especialistas: aplicações comerciais**. São Paulo: McGraw-Hill, 1988.
- Costa, A . e Edelweiss, N. Um ambiente para desenvolvimento de protótipos de bancos de dados dedutivos temporais. **Anais/Simpósio Brasileiro de Banco de Dados**. 1989. Páginas 163 a 173.
- Cruz, T. **Workflow, a tecnologia que vai revolucionar o processos produtivo das manufaturas na organização moderna**. São Paulo: Editora Atlas, 1998.
- D´Alleyrand, M. **Workflow em sistema de gerenciamento eletrônico de imagens**. São Paulo: CENADEM, 1995.
- Davenport, T. M. **Reengenharia de processos**. Tradução: Waltenair Dutra. Rio de Janeiro: Editora Campus, 1994.
- Debenham, J. A business Process Agent. **Research Index**. Australia, 2000. Disponível em: <http://www.researchindex.com>. Acesso em: 05 maio 2002.
- Durkin, J. **Expert Systems: Design and development**. Ohio: Intelligent Computer Systems, 1993.
- Edelweiss, N. Banco de dados temporais: teoria e prática. **Anais XVII jornada de atualização em informática**. 1998. Páginas 225 a 282.
- Elmari, R. Recent advances in temporal databases. **Anais/Simpósio Brasileiro de Banco de Dados**. 1991. Páginas 17 a 26.
- Filho, L. **Sistemas Workflow em processos empresariais, aplicando técnicas de processos produtivos fabris e modelagem em redes de Petri**. Dissertação de mestrado apresentada na Universidade Mackenzie. São Paulo, 2000.

Fruscione, J. **Workflow Automatizado**. Tradução: Fernando Luis Bär. São Paulo: Editora CENADEM, 1996.

Hammer, M. e Staton, S. **The reengineering revolution**. New York: HarperBusiness, 1994.

Harrington, H. .J. **Aperfeiçoamento de Processos Empresariais**. Tradução: Luiz Liske. São Paulo: Makron Books, 1993.

Haykin, S. **Redes Neurais: princípios e prática**. Porto Alegre: Bookman, 2001.

Hecht-Nielsen, R. **Neurocomputing**. Nova York: Addison-Wesley, 1990.

Hronec, S. **Sinais Vitais**. São Paulo: Makron Books, 1994.

Jackson, M. e Twaddle, G. **Business process implementation – building workflow systems**. London: Addison Wesley Longman, 1997.

Jensen, C.S. **A consensus glossary of temporal database concepts**. New York: Sigmod Record, 1994.

Kalläk, B., Pettersen, T. e Ressem, J. Intelligence, flexibility, and real support for business processes. In: **OOPSLA '98 Workshop**, Canada, 1998. Disponível em: <http://www.researchindex.com>. Acesso em: 05 maio 2002.

Khoshafian, S. e Buckiewicz, M. **Introduction to groupware, workflow and workgroup computing**. New York: John Wiley, 1995.

Kobelius, J. **Workflow strategies**. FosterCity-CA: IDG Books, 1997.

Kock, W. **Gerenciamento eletrônico de documentos**. São Paulo: CENADEM, 1998.

Kodratoff, Y. **Leçons d'Apprentissage Symbolique Automatique**. Toulouse: Capedues-Editions, 1986.

Koubarakis, M. e Plexousakis, D. Business process modeling and design: AI models and methodology. **Research Index**. Greece, 1999. Disponível em: <http://www.researchindex.com>. Acesso em: 05 maio 2002.

Koulopoulos, T. **The Workflow Imperative**. New York: ITP Books, 1995.

Lander, S., Corkill, D. e Rubinstein, Z. **KPM: Knowledge-Based process management**. Amherst: Knowledge Technologies International, 2000.

Lucena, C. **Inteligência artificial e engenharia de software**. Rio de Janeiro: PUC/RJ, 1987.

Maes, P. The agent network architecture. **Sigart Bulletin**, Cambridge-MA, 1991.

Mizrahi, V. **Treinamento em linguagem C++**. São Paulo: Makron Books, 1994.

Moore, J., Stader, J., Chung, P., Jarvis, P. e Macintosh, A. Ontologies to support the management of new product development in the chemical process industries. In: **International Conference on Engineering Design**, Munich 1999. Disponível em: <http://www.researchindex.com>. Acesso em: 05 maio 2002.

Mylopoulos, J. **On knowledge base management systems**. Nova York: Springer-Verlag, 1986.

Nareyek, A. Applying local search to structural constraint satisfaction. **Research Index**. Berlin, 1999. Disponível em: <http://www.researchindex.com>. Acesso em: 05 maio 2002.

Navathe, S. e Tanaka, A. Modeling and design of events and rules in databases. **Anais/Simpósio Brasileiro de Banco de Dados**. 1994. Páginas 31 a 50.

Nilsson, N. **Principles of artificial intelligence**. Nova York: Morgan Kaufmann, 1986.

Odgers, B., Thompson, S., Shepherson, J., Cui, Z., Judge, D. e O'Brien, P. **Technologies for intelligent workflows: experiences and lessons**. Suffolk: Intelligent Business System Research, 2000.

Pessanha, K. **Correio eletrônico e workgroup computing**. São Paulo: Berkeley, 1996.

Porter, M. **Competição – Estratégias competitivas essenciais**. Rio de Janeiro: Campus, 1999

Poyssick, G. e Hannaford, S. **Workflow reengineering**. Mountain View-CA: Adobe Press, 1996.

Ramakrishnan, R. e Ullman, J. **A survey of research on deductive database systems**. Nova York: ,1993.

Randall, D. **Knowledge-based systems in artificial intelligence**. Nova York: McGraw-Hill, 1982.

Reimer, U. Knowledge integration for building organizational memories. **Research Index**. Zurich, 1999. Disponível em: <http://www.researchindex.com>. Acesso em: 05 maio 2002.

Reisig, W. Place/transition systems. **Computer Science**. (Springer-Verlog), 1987.

Rich, E. **Artificial Intelligence**. Nova York: McGraw-Hill, 1983.

Russel, S. e Norvig, P. **Artificial Intelligence: A modern approach**. Nova York: Prentice Hall, 1995

Sandri, S. e Correa, C. **Lógica Nebuloso**. São Paulo: INPE, 1998.

Scultey, J. **Workflow**. Boston: Delphi Consulting Group, 1997.

Scultey, J. **Workflow**. Boston: Delphi Consulting Group, 2001.

Silva, E. e Menezes, E. **Metodologia da pesquisa e elaboração de dissertação**. Florianópolis: Laboratório de Ensino a Distância da UFSC, 2001.

Stader, J. e Jarvis, P. Intelligent support for enterprise modelling. In: **13<sup>th</sup> biennial European Conference on Artificial Intelligence**, Brighton, 1998. Disponível em: <http://www.researchindex.com>. Acesso em: 05 maio 2002.

Votre, V. **C++ explicado e aplicado**. São Paulo: Edição independente, 1998.

White, T. e Fischer, L. **The workflow paradigm**. Alameda-CA: Future Strategies Inc, 1994.

Winston, P. **Artificial Intelligence**. Nova York: Addison-Wesley, 1977.

Wooldridge, M. e Jennings, N. Agent-oriented software engineering. **Research Index**. United Kingdom, 1999a. Disponível em: <http://www.researchindex.com>. Acesso em: 05 maio 2002.

Wooldridge, M. e Jennings, N. Intelligent Agents: theory and practice.. **Research Index**. United Kingdom, 1999b. Disponível em: <http://www.researchindex.com>. Acesso em: 05 maio 2002.

Yoneyama, T. e Nascimento, C. **Inteligência Artificial em controle e automação**. São Paulo: Editora Edgard Blucher, 2000.

## **ANEXO**

### **DOCUMENTAÇÃO DO SISTEMA DE GESTÃO DA OUVIDORIA**