

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CURSO PÓS-GRADUAÇÃO EM CIÊNCIAS DA  
COMPUTAÇÃO**

**Marcelo José Ferrari**

**DESENVOLVENDO WORKFLOW COM AGENTES  
MÓVEIS**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

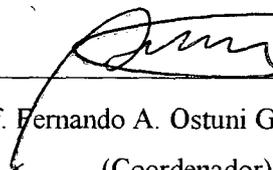
Prof<sup>o</sup> Dr. João Bosco Manguiera Sobral

**Florianópolis, Março de 2001**

# DESENVOLVENDO WORKFLOW COM AGENTES MÓVEIS

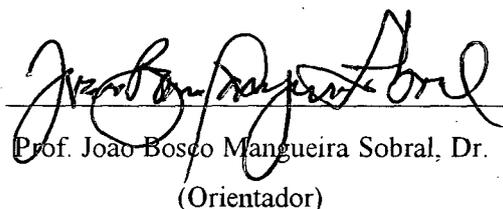
Marcelo José Ferrari

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina.

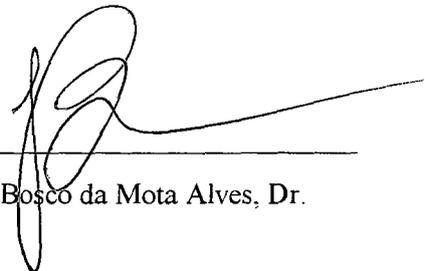


Prof. Fernando A. Ostuni Gauthier, Dr.  
(Coordenador)

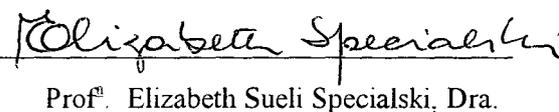
Banca Examinadora



Prof. João Bosco Manguiera Sobral, Dr.  
(Orientador)



Prof. João Bosco da Mota Alves, Dr.



Prof.<sup>a</sup> Elizabeth Sueli Specialski, Dra.

*“Não é meu desejo ser uma pessoa comum”.  
É meu direito ser diferente, ser singular, incomum.  
Desenvolver os talentos que Deus me deu.  
Não desejo ser um cidadão pacato e modesto, dependendo  
sempre de alguém.  
Quero correr o risco calculado, sonhar e construir, falhar e  
suceder.  
Recuso trocar o incentivo pela doação,  
Prefiro as intemperanças à vida garantida.  
Não troco minha dignidade por ajuda de outros.  
Não me acovardo e nem me curvo ante ameaças.  
Minha herança é ficar ereto, altivo e sem medo;  
Pensar e agir por conta própria  
e aproveitando os benefícios da minha criatividade,  
encarar arrojadamente o mundo e dizer com clareza e  
convicção:  
ISTO É O QUE EU SOU.”*

## **DEDICATÓRIA**

A meus pais, Bruno e Otilia, minha irmã Magda,

Minha amiga Cintia,

A Deus que sempre me acompanhou em todos os momentos de minha vida.

## **AGRADECIMENTOS**

A Deus por ter me dado forças para finalizar este trabalho.

Ao Professor João Bosco Manguiera Sobral, pela orientação e auxílio no desenvolvimento deste trabalho.

Aos meus pais Bruno e Otilia, que sempre me ajudaram em todos os momentos de minha vida.

A minha amiga Cinta que esteve em meu lado durante a execução deste trabalho, acreditando em mim e me estimulando para não desistir, pois as dificuldades foram muitas.

Ao meu amigo Luiz Henrique Kobarg, pela amizade, companheirismo, apoio e colaboração.

Ao amigo Fabricio, pelo auxílio no desenvolvimento do protótipo.

A empresa HBTec por me dispensar durante o horário comercial para que eu pudesse assistir as aulas.

A todos aqueles que, de maneira direta ou indireta, colaboram e acreditaram na realização deste trabalho.

## RESUMO

Um *workflow* é definido como sendo um conjunto de atividades, que podem ou não ser executadas simultaneamente, com alguma especificação de controle e fluxo de dados entre estas atividades. Um *workflow* pode representar vários processos e seus relacionamentos. Sistemas de workflow consistem de componentes de modelagem e componentes de representação do sistema. Na área de modelagem de workflow não existe, a princípio, um conjunto de modelos e técnicas que formalizem o seu desenvolvimento.

Além disso, existe ainda a necessidade dos sistemas de *workflow* manipularem e monitorarem a informação relativa ao fluxo de trabalho para gerenciar, coordenar e controlar o trabalho mais eficientemente e para proporcionar recuperação histórica do fluxo de trabalho permitindo assim monitoramento histórico (auditorias).

*Workflow* tem como um dos seus principais objetivos, minimizar o problema da coordenação do trabalho nos processos de negócios. O principal problema das atividades baseadas em processos é que não se tem o controle total das atividades que devem ser executadas. Além disso, não se têm uma visão de quais informações estão sendo manipuladas por essas atividades em um dado instante e nem quem as manipula. As técnicas de modelagem do *workflow* devem ter como objetivo básico, minimizar estes problemas.

O objetivo básico deste trabalho consiste no desenvolvimento de um protótipo de um *workflow*, utilizando agentes móveis para validar o modelo que será proposto construindo um protótipo de um workflow que valide na prática os conceitos de agentes móveis e workflow. Esta tarefa não é trivial e envolve um grande esforço de pesquisa.

**Palavras-chave:** Workflow, Agentes Móveis, Mobilidade, Trabalho cooperativo, processos de negócios e tarefas.

## ABSTRACT

A workflow is defined as being a group of activities, that they cannot or not to be executed simultaneously, with some control specification and flow of data among these activities. A workflow can represent several processes and its relationships. Workflow systems consist of modelagem components and components of representation of the system. In the area of workflow modelagem it doesn't exist, at first a group of models and techniques that formalize its development.

Besides, it exists the need of the workflow systems they still manipulate and they monitor the relative information to the work flow for gerenciar, to coordinate and to control the work more eficientemente and to provide historical recovery of the work flow allowing like this historical monitoramento (audits).

Workflow has as one of its main objectives, to minimize the problem of the coordination of the work in the processes of business. The main problem of the activities based on processes is that the total control of the activities is not had that should be executed. Besides they are not had a vision of which information are being manipulated by those activities in a dice instant and nor who manipulates them. The techniques of modelagem of the workflow should have as basic objective, to minimize these problems.

The basic objective of this work consists of the development of a prototype of a workflow, using mobile agents to validate the model that will be proposed building a prototype of a workflow that validates in the practice the mobile agents' concepts and workflow. This task is not trivial and it involves a great research effort.

**keywords:** Workflow, Mobile Agents, Mobility, I Work cooperative, processes of business and tasks.

## SUMÁRIO

### CAPÍTULO 1

<b>1. Introdução.....</b>	<b>14</b>
<b>1.1 Objetivos do trabalho.....</b>	<b>14</b>
<b>1.2 Justificativas e relevâncias.....</b>	<b>14</b>
<b>1.3 Organização do trabalho.....</b>	<b>15</b>

### CAPÍTULO 2

<b>2.1 Fundamentos do workflow.....</b>	<b>16</b>
<b>2.2 Rápida história.....</b>	<b>21</b>
<b>2.3 Tipos de workflow.....</b>	<b>21</b>
<b>2.3.1 Ad hoc.....</b>	<b>22</b>
<b>2.3.2 Produção/Transação.....</b>	<b>21</b>
<b>2.3.3 Administrativo.....</b>	<b>23</b>
<b>2.3.4 Orientados para objeto.....</b>	<b>24</b>
<b>2.3.5 Baseado no conhecimento.....</b>	<b>26</b>
<b>2.4 Estrutura do Workflow.....</b>	<b>26</b>
<b>2.4.1 Instâncias ou casos.....</b>	<b>27</b>
<b>2.4.2 Pastas.....</b>	<b>27</b>
<b>2.4.3 Documentos.....</b>	<b>28</b>
<b>2.5 Os Três Rs do workflow.....</b>	<b>28</b>
<b>2.5.1 Elementos papel.....</b>	<b>28</b>
<b>2.5.2 Regras.....</b>	<b>29</b>
<b>2.5.3 Rotas.....</b>	<b>29</b>
<b>2.5.3.1 Rota Serial.....</b>	<b>30</b>
<b>2.5.3.2 Rota paralela.....</b>	<b>30</b>
<b>2.5.3.3 Rota concorrente.....</b>	<b>31</b>
<b>2.5.3.4 Rota condicional.....</b>	<b>31</b>
<b>2.5.3.5 Rota Dependente.....</b>	<b>31</b>
<b>2.6 O Ambiente workflow.....</b>	<b>32</b>

<b>2.6.1 Hardware.....</b>	<b>32</b>
<b>2.6.2 Rede.....</b>	<b>33</b>
<b>2.6.3 Computador.....</b>	<b>33</b>
<b>2.6.3.1 Arquitetura workflow cliente-servidor.....</b>	<b>33</b>
<b>2.6.4 Software.....</b>	<b>34</b>
<b>2.7 Implantação de um sistema de workflow.....</b>	<b>34</b>
<b>2.7.1 Etapa 1 Análise de trabalho(workflow) atual.....</b>	<b>35</b>
<b>2.7.2 Etapa 2 Projetar o modelo de informação do fluxo trabalho.....</b>	<b>36</b>
<b>2.7.3 Etapa 3 Programar o modelo de informação definido.....</b>	<b>37</b>
<b>2.7.4 Etapa 4 Implantar o workflow.....</b>	<b>37</b>
<b>2.7.5 Etapa 5 Atualizar o modelo de informação implantado.....</b>	<b>37</b>
<b>2.8 Vantagem do uso da tecnologia workflow.....</b>	<b>38</b>
<b>2.9 Modelagem de workflow.....</b>	<b>39</b>
<b>2.9.1 Modelagem do workflow em redes de petri.....</b>	<b>40</b>
<b>2.9.2 Modelo de gatilhos.....</b>	<b>40</b>
<b>2.9.4 Modelo de ações.....</b>	<b>40</b>
<b>2.9.5 Modelo de interoperabilidade da WfMC.....</b>	<b>40</b>
<b>2.10 Exemplos de sistemas de workflow existentes.....</b>	<b>41</b>

## **CAPÍTULO 3**

<b>3. Utilizando agentes móveis.....</b>	<b>45</b>
<b>3.1 Histórico.....</b>	<b>45</b>
<b>3.2 Características de agentes.....</b>	<b>46</b>
<b>3.3 Características avançadas de agentes.....</b>	<b>47</b>
<b>3.4 Classificação de agentes.....</b>	<b>48</b>
<b>3.5 Tipos de agentes.....</b>	<b>49</b>
<b>3.6 Características que tornam uma aplicação apropriada para agentes.....</b>	<b>50</b>
<b>3.7 Que problemas os agentes podem resolver.....</b>	<b>51</b>
<b>3.8 Arquitetura de agentes.....</b>	<b>52</b>
<b>3.9 Por que usar arquitetura de agentes.....</b>	<b>53</b>
<b>3.10 Comunicação entre agentes.....</b>	<b>56</b>

<b>3.11 Desempenho dos agentes.....</b>	<b>57</b>
<b>3.12 Tendências.....</b>	<b>58</b>
<b>3.13 Plataforma de agentes móveis.....</b>	<b>61</b>
<b>3.14 Trabalhos relacionados.....</b>	<b>62</b>

## **CAPÍTULO 4**

<b>4. Implementação.....</b>	<b>64</b>
<b>4.1 Estrutura do workflow implementado.....</b>	<b>64</b>
<b>4.2 Modelos de dados UML.....</b>	<b>65</b>
<b>4.3 Arquitetura do protótipo.....</b>	<b>65</b>
<b>4.4 Instalação do ambiente.....</b>	<b>66</b>
<b>4.5 Configuração do ambiente.....</b>	<b>67</b>
<b>4.6 Execução do protótipo.....</b>	<b>67</b>
<b>4.7 Implementação do sistema.....</b>	<b>69</b>
<b>4.8 Fontes do sistema.....</b>	<b>72</b>

## **CAPÍTULO 5**

<b>5. Ferramentas existente no mercado.....</b>	<b>75</b>
<b>5.1 Componentes da ferramenta Workflow Builder 2.0.....</b>	<b>76</b>
<b>5.1.1 Construtor de workflow.....</b>	<b>76</b>
<b>5.1.2 Mecanismo de workflow baseado em regras.....</b>	<b>76</b>
<b>5.1.3 Dimensionamento completo.....</b>	<b>77</b>
<b>5.1.4 Notificações eletrônicas.....</b>	<b>77</b>
<b>5.1.5 Caixa pessoal de entrada.....</b>	<b>78</b>
<b>5.1.6 Integração via e-mail.....</b>	<b>78</b>
<b>5.1.7 Um workflow habilitado à internet.....</b>	<b>78</b>
<b>5.1.8 Facilidade de habilitar aplicativos no workflow.....</b>	<b>78</b>
<b>5.2 Necessidade de software e hardware.....</b>	<b>79</b>
<b>5.3 Características básicas.....</b>	<b>79</b>
<b>5.3.1 Construtor de workflow gráfico.....</b>	<b>79</b>
<b>5.3.2 Mecanismo do workflow.....</b>	<b>80</b>

<b>5.3.3 Notificações eletrônicas.....</b>	<b>80</b>
<b>5.3.4 Monitoramento gráfico e análise de desempenho.....</b>	<b>81</b>
<b>5.3.5 Arquitetura aberta e dimensionável.....</b>	<b>81</b>
<b>5.4 Implementação do protótipo utilizando a ferramenta workflow builder.....</b>	<b>81</b>

## **CAPÍTULO 6**

<b>6 Comparativo entre protótipo implementado e workflow existente no mercado..</b>	<b>87</b>
<b>6.1 Conclusões.....</b>	<b>87</b>

## **CAPÍTULO 7**

<b>7 Conclusão.....</b>	<b>80</b>
-------------------------	-----------

<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>92</b>
--	-----------

## LISTA DE FIGURAS

<b>FIGURA 1.0 – Gráfico que mostra alguns dos tipos de rotas.....</b>	<b>32</b>
<b>FIGURA 1.1 – Representação da estrutura de agentes x cliente-servidor.....</b>	<b>55</b>
<b>FIGURA 1.2 – Representação do modelo de workflow implementado.....</b>	<b>64</b>
<b>FIGURA 1.3 – Modelo lógico de dados do sistema proposto.....</b>	<b>65</b>
<b>FIGURA 1.4 – Arquitetura do protótipo.....</b>	<b>66</b>
<b>FIGURA 1.5 – Fragmento do arquivo autoexec.bat.....</b>	<b>67</b>
<b>FIGURA 1.6 – Interface principal do sistema.....</b>	<b>69</b>
<b>FIGURA 1.7 – Interface de entrada de dados dos pedidos.....</b>	<b>69</b>
<b>FIGURA 1.8 – Interface de checagem de estoque.....</b>	<b>70</b>
<b>FIGURA 1.9 – Interface de checagem de duplicatas pendentes.....</b>	<b>70</b>
<b>FIGURA 2.0 – Interface de cancelamento de pedido/itens.....</b>	<b>71</b>
<b>FIGURA 2.1 – Interface de informações do workflow.....</b>	<b>71</b>
<b>FIGURA 2.2 – Interface de acompanhamento dos pedidos.....</b>	<b>72</b>
<b>FIGURA 2.3 – Classe Principal do protótipo (Principal.class).....</b>	<b>72</b>
<b>FIGURA 2.4 – Classe Administrador (Administrador.class).....</b>	<b>73</b>
<b>FIGURA 2.5 – Classe Agente Móvel (AgenteMovel.class).....</b>	<b>74</b>
<b>FIGURA 2.6 – Classe de Informações do Agente (InformacoesAgente.class).....</b>	<b>74</b>
<b>FIGURA 2.7 – Interface inicial do workflow.....</b>	<b>82</b>
<b>FIGURA 2.8 – Interface de entrada de dados dos pedidos.....</b>	<b>82</b>
<b>FIGURA 2.9 – Interface de checagem de estoque.....</b>	<b>83</b>
<b>FIGURA 3.0 – Interface de checagem de duplicatas pendentes.....</b>	<b>84</b>
<b>FIGURA 3.1 – Interface de cancelamento de pedido/itens.....</b>	<b>85</b>
<b>FIGURA 3.2 – Interface de informações do workflow.....</b>	<b>85</b>
<b>FIGURA 3.3 – Interface de acompanhamento dos pedidos dentro workflow.....</b>	<b>86</b>

## ABREVIATURAS E SÍMBOLOS

<b>ACL – Language Communication Agents.....</b>	<b>56</b>
<b>API – Application Programming Interface.....</b>	<b>67</b>
<b>CSCW – Computer Supported Cooperative Work.....</b>	<b>17</b>
<b>DBMS – Data Base Management System.....</b>	<b>24</b>
<b>IA – Artificial Intelligence.....</b>	<b>29</b>
<b>JDK – Java Develop Kit.....</b>	<b>66</b>
<b>JVM – Java Virtual Machine.....</b>	<b>53</b>
<b>KIF – Knowledge Interchange Format.....</b>	<b>55</b>
<b>KQML – Knowledge Query Manipulation Language.....</b>	<b>55</b>
<b>OO - Object Oriented.....</b>	<b>24</b>
<b>ORB – Object Request Brokers.....</b>	<b>53</b>
<b>SAC – Sistema de Atendimento ao Cliente.....</b>	<b>19</b>
<b>WFDL – Workflow Description Language.....</b>	<b>39</b>
<b>WFMC – Workflow Management Coalition.....</b>	<b>39</b>
<b>WFMS – Workflow Management System.....</b>	<b>38</b>

# CAPÍTULO 1

## 1. Introdução

Os sistemas de workflow podem ser definidos como sistemas cujo objetivo é “auxiliar as organizações na especificação, execução, monitoramento e coordenação do fluxo de trabalho em um ambiente de escritório distribuído” [CRU98]. O fluxo de trabalho corresponde à tramitação de documentos e informações entre diversos *agentes*, pertencentes a unidades organizacionais potencialmente distintas, cada qual agregando uma parcela do trabalho necessário para que se alcance determinado objetivo de negócio.

### 1.1 Objetivos do trabalho

Este trabalho vem contribuir na área de workflow e tem por objetivos:

- a) Apresentar um modelo conceitual de workflow desenvolvido a partir da tecnologia de agentes móveis.
- b) Validar o modelo proposto construindo um protótipo de um workflow que valide na prática os conceitos de agentes e workflow. Esta tarefa não é trivial e envolve um grande esforço de pesquisa. Este trabalho pretende ser uma experiência sobre o tema, dentro das atividades do grupo de Computação Distribuída do CPGCC/UFSC.

### 1.2 Justificativas e relevâncias

As soluções existentes no mercado são baseadas na arquitetura cliente-servidor. Segundo [CRU98] mesmo a tecnologia do workflow ser recente, ela provoca grandes discussões entre teóricos, cientistas e fabricantes de software com essa tecnologia. Os americanos dominam em quantidade o mercado com suas soluções (ORACLE); entretanto, existem outras opções mais avançadas fora do Estados Unidos. No Chile,

uma empresa de nome Xnear é detentora de uma tecnologia avançada que leva o mesmo nome da companhia, Xnear.

Outra alternativa para o desenvolvimento de workflow são os agentes móveis, que amenizam o problema da transmissão na rede cliente-servidor. Na rede, a largura da banda é um importante a algumas vezes raro recurso de aplicação distribuída. A transação solicitada entre um cliente e um servidor pode requerer muitas voltas sobre a rede para ser completada. Este tipo de operação cria um tráfego muito grande e consome muito da banda de transmissão. Em um sistema com muitos “clientes” e, ou mais transações, o total de requerimentos no tamanho da banda pode exceder a disponibilidade permitida, ocasionando uma performance muito ruim para as aplicações que estão envolvidas. Na utilização de agentes para buscar as solicitações ou transações e, enviando os agentes do cliente para o servidor, o fluxo na rede é reduzido. Dessa maneira, somente o que os agentes encontrarem será transmitido pela rede, tornando a velocidade de transmissão maior [THO97, SHO93].

### **1.3 Organização do trabalho**

Este trabalho está organizado em sete capítulos descritos a seguir.

O primeiro capítulo define os objetivos do trabalho, justificativas, relevâncias e organização do trabalho.

No segundo capítulo são identificadas características básicas e fatores de mais alto nível envolvidos na construção de um sistema de workflow.

O terceiro capítulo trata das questões básicas sobre agentes (são citadas características, vantagens etc) conceitos estes que serão utilizados na elaboração deste trabalho.

O quarto capítulo mostra detalhes da implementação de workflow usando o paradigma dos agentes móveis.

O quinto capítulo mostra as ferramentas existentes no mercado que desenvolvem workflow's.

O sexto capítulo mostra um comparativo entre o protótipo implementado e workflow desenvolvido com ferramenta existente no mercado.

O sétimo capítulo trata da conclusão do trabalho e mostra de forma qualitativa os resultados obtidos.

O terceiro capítulo será mostrado um exemplo de aplicação que será implementado utilizando os conceitos de agentes e workflow.

## CAPÍTULO 2

### WORKFLOW

Os sistemas de workflow se inserem na área mais abrangente de software de apoio ao trabalho cooperativo (*Computer Supported Cooperative Work - CSCW*), cujo objetivo é, como o nome indica, o de propiciar suporte computadorizado a grupos de pessoas que trabalham em conjunto para atingir um objetivo comum. Sistemas de workflow são utilizados em diversas áreas, como a de engenharia de software, a científica e organizacional: nos ateremos no presente trabalho aos voltados para organizações ou escritórios distribuídos [MWF93].

Os sistemas de workflow podem ser definidos como sistemas cujo objetivo é "auxiliar as organizações na especificação, execução, monitoramento e coordenação do fluxo de trabalho em um ambiente de escritório distribuído" [BUL92]. O fluxo de trabalho corresponde à tramitação de documentos e informações entre diversos agentes, pertencentes a unidades organizacionais potencialmente distintas, cada qual agregando uma parcela do trabalho necessário para que se alcance determinado objetivo de negócio.

A observação de que por volta de 90% desta tramitação pode ser considerada rotineira e repetitiva torna natural que se deseje suprir suporte automatizado para estas tarefas. Após um surto de interesse nos anos 70, associado à área de automação de escritórios, a pesquisa relativa a sistemas de workflow ficou estagnada durante a maior parte dos anos 80, devido ao fracasso dos sistemas pioneiros. Verificou-se que estes sistemas apresentavam uma inflexibilidade que inviabilizava seu uso prático, mesmo em situações simples e controladas [BAR98].

#### 2.1 Fundamentos do Workflow

A palavra workflow é nova; tem, se muito, uns cinco anos ou seis anos. É óbvio que, se a palavra é nova, a tecnologia também o é. Aliás. A tecnologia workflow surgiu

junto com outras tecnologias que possibilitaram o trabalho integrado, corporativo, ativo. Antes do surgimento dessas ferramentas, conhecidas genericamente como tecnologias para trabalho em grupo, as atividades realizadas com o suporte dos computadores poderiam chegar a, máximo, se complementares. Além disso, o processo, se fosse conhecido como tal, continha atividades que ou não agregavam valor ao produto ou, quando o faziam, era de modo tão insipiente que as perdas eram visíveis o olho nu [CRU98].

Os processos eram passivos, assim como os sistemas de informações que tinham sido desenvolvidos para suportá-los. Ambos cumpriam suas funções primordiais somente se cada funcionário quisesse, se quisesse, quando quisesse. Por isso, os processos e os sistemas eram passivos. Os dois existiam em estado latente. Se cada funcionário fizesse a parte que lhe competia, no tempo e no espaço correto, dentro dos limites do procedimento da atividade, o trabalho conseguia ser produtivo, gastava-se menos do que o valor agregado pela atividade, e isso se refletia no aumento da margem de contribuição da cada elo da cadeia produtiva no preço do produto final. Eram raros, porém, os processos racionalizados, com produtividade elevada, eficiente e lucrativos pela própria natureza.

Ainda hoje, os processos, na grande maioria das empresas, continuam passivos. As atividades continuam passivas. Os sistemas de informação continuam passivos. Por causa disso, os ganhos que poderiam advir com a implantação de novas tecnologias ficam aquém do esperado.

O que é ser um processo passivo?

Um processo é passivo quando as atividades que o compõem não dispõem de tecnologia que as transforme de atividade que são puxadas em atividades que puxam o trabalho. Atividades passivas esperam para serem executadas, enquanto as ativas cobram se seu responsável sua operação. Atividades passivas esperam para que o produto de sua operação seja enviado à próxima, ou próximas atividades, as ativas têm “inteligência” suficiente para o despacharem tão logo ele esteja pronto. Atividades

passivas não sabem para quem enviar o produto de sua execução; as ativas sabem. Além de tudo isso, a mesma tecnologia que transforma as atividades de passivas em ativas executa uma série de outras atividades e possibilitam que o processo seja gerenciado de modo muito eficaz [CRU98].

#### Exemplo de atividades passivas.

Um funcionário recebe uma solicitação de assistência a cliente (SAC) enviada pela atendente do telemarketing às 14h. Essa SAC pode ter vindo via papel ou via algum sistema informatizado de atendimento a cliente. O funcionário não toma conhecimento da SAC tão logo ela chega. Sequer, muitas vezes, a abre para saber do que se trata. A SAC, por sua vez, permanece aguardando que o funcionário tome uma atitude. O supervisor não fica sabendo o que está ocorrendo. E o cliente fica sem atendimento. Às 17h, quando todos vão embora, pode ocorrer de a SAC ainda não ter sido atendida... e internamente ninguém ficou sabendo!

#### O que é uma tecnologia passiva?

É toda tecnologia que não consegue fazer com que as pessoas que deveriam beneficiar-se dela a usem. São inertes, embora possuam o que de mais novo possa existir de vantagens tecnológicas. Uma tecnologia passiva suporta passivamente um processo. Uma tecnologia passiva permanece distante do usuário, não tendo como se fazer notar ou como se fazer usar, a não ser que o usuário se lembre e queira usá-la.

#### Exemplo de uma tecnologia passiva.

Um sistema de informações extraordinário, construído sobre um banco de dados orientado para objeto de última geração. Vai rodar em plataformas Risc e Intel, com tecnologia de primeira linha. Só para ser desenvolvido, custou US\$ 1.200.000,00 e levou um ano e meio para ficar pronto. Esse sistema vai controlar toda a parte administrativa, comercial e de produção de uma empresa de serviço que atende de 10 15 mil clientes por ano. Só há um probleminha: se algum dos usuários, ou todos decidirem

não utilizar corretamente o sistema, em termos de cumprimento de prazos e expectativas, tudo vai por água abaixo.

Em ambos os exemplos, ficam evidentes que, por mais que uma tecnologia seja atualizada, por si só ela não será suficiente para fazer uma atividade acontecer. Atividades podem ser atrasadas, pedidos podem deixar de ser atendidos, processos podem sofrer de graves problemas funcionais por mais que tenham sido racionalizados os fluxos de trabalho que os transpassam.

Então o que é que precisa se feito para que os processos, atividades e os procedimentos possam corresponder à expectativa que o planejamento e a organização, com que eles foram criados, prometeram? A palavra-chave é automatização. Esse é o componente que vai tornar processos, atividades e procedimentos 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! Para automatizar esses elementos, é necessário que o elo de ligação entre todos tenham uma natureza ativa, que transforme, que dê vida, que faça pulsar as operações do dia-a-dia. Essa ferramenta chama-se WORKFLOW, a seguir algumas definições de workflow:

*Workflow é a tecnologia que possibilita automatizar processos, racionalizando-os e potencializando-os por meio de dois componentes implícitos: organização e tecnologia.*

*Workflow é o fluxo de controle e informação num processo de negócio*

*Workflow é um conjunto de ferramentas que possibilita análise proativa, compressão e automação de atividades e tarefas baseadas em informação.*

*Workflow é a tecnologia que ajuda a automatizar políticas e procedimentos numa organização.*

## 2.2 Rápida História

Nem bem a tecnologia workflow começou a existir e já provoca grandes discussões entre teóricos, cientistas e fabricantes de softwares com essa tecnologia. Os americanos dominam em quantidade o mercado com suas soluções; entretanto, existem outras opções mais avançadas fora dos Estados Unidos. No Chile, uma empresa de nome Xnear é detentora de uma tecnologia muito avançada que leva o mesmo nome da companhia, Xnear [SWE95].

A definição e os tipos de Workflow variam de autor para autor. Alguns especialistas dividem Workflow em quatro tipos: *ad hoc*, orientado para a produção (*production oriented*), orientado para a administração (*administrative oriented*) e baseado no conhecimento (*knowledg-based oriented*). Já outros preferem classificar Workflow da seguinte forma: *ad hoc*, baseado em transação (*transaction based*), orientado para objeto (*object oriented*), mas com três modelos de processo: orientado para correio eletrônico, orientado para documentos e orientado para processo. A combinação entre os vários tipos de Workflow e modelos de processo faz com que a implementação dessa tecnologia seja flexível e, praticamente, abranja todas as necessidades de automatização de fluxos de trabalho [MCC92].

## 2.3 Tipos de WorkFlow

No geral, Workflow é dividido nos seguintes tipos:

- Ad hoc
- Produção ou Transação
- Administrativo
- Orientado para Objeto
- Baseado no conhecimento

### 2.3.1 Ad hoc

Qualquer que seja a corrente de especialistas que esteja discutindo Workflow, todas elas concordam com esse tipo. A expressão Ad Hoc segundo Aurélio Buarque de Holanda definido como uma expressão latina que significa ‘para isto’, ‘para este caso’. Tem as seguintes conotações. 1. De propósito, adrede. 2. Designado por se tratar de perito para executar determinada tarefa. Ex. delegado ‘ad hoc’.

Assim, podemos concluir que o Workflow do tipo *ad hoc* é aquele criado para ser usado dinamicamente por grupos de trabalho cujos participantes necessitem executar procedimentos individualizados para cada documento processado dentro do fluxo de trabalho. Embora algumas partes de um procedimento possam se reutilizadas, não existem consistências para que sejam criadas estruturas que formalizem esse procedimento. Workflow *ad hoc* é composto de regras que dificilmente se repetem, por inteiro, a cada documento que tenha que ser trabalhado pela atividade.

Um exemplo de Workflow *ad hoc* é o orientado para *e-mail*. A contribuição importante proporcionada por esse tipo está diretamente ligada à quebra de paradigma ocasionada por sua adoção. Embora seja o mais simples Workflow, o fato de implantá-lo requer uma mudança cultural na organização, nem sempre pequena nem simples. As pessoas, em qualquer empresa, estão acostumadas com papéis, muitos papéis. De repente, tudo isto passa a ser exclusivamente um meio eletrônico de se comunicarem. Os papéis somem! É complicado.

### 2.3.2 Produção/ Transação

Diferentemente do Workflow orientado para e-mail, o orientado para transação ou para produção é do tipo peso pesado nas aplicações Workflow. Isso ocorre porque o Workflow orientado para aplicações envolvem grandes quantidades de dados, muitas políticas de negócios e recursos financeiros em grande escala. Essa mistura explosiva de elementos críticos dá um caráter especial a esse tipo de Workflow e faz com que seu desenvolvimento e implementação tenha que ser cuidadosamente planejado; afinal, não

se pode colocar em risco a operação de uma empresa com a desculpa de se estar tentando melhorar o tempo e a operacionalidade de fluxo de trabalho.

Os dados tratados por um sistema de Workflow orientado para transações têm duas origens, uma no próprio fluxo. São os dados que dão início ao fluxo de trabalho; em outras palavras são os dados que põem a engrenagem em movimento, como, por exemplo, a solicitação de um cliente que quiser resolver um problema, uma típica aplicação SAC (Serviço de Atendimento aos Clientes). Outra origem está nos bancos de dados que suportam as aplicações ligadas ao sistema. Por exemplo, ao receber a solicitação de um cliente, o atendente, como um simples toque numa única tecla, recupera todas as informações existentes sobre aquele cliente nos bancos de dados da empresa. Isso evita ter que perguntar todos os dados do cliente novamente a ele, evita perda de tempo com tarefas repetitivas e organiza o envio do documento a quem, de fato, vai resolver o problema e a quem vai supervisioná-lo.

Qualquer aplicação cujo escopo seja suportar o processo de negócio de uma empresa é, por si só, uma aplicação sensível; afinal, não dá para colocar em risco, por conta de ações mal planejadas, o principal fator de existência de qualquer empresa: seu negócio. Esse tipo de Workflow envolve muitos departamentos dentro da empresa, pois os processos existentes na maioria das empresas são multifuncionais. Por isso, o nível de complexidade é grande, principalmente na fase de desenvolvimento; depois de construídos, no entanto, só fluxos de trabalho automatizados sofrem poucas alterações tanto estruturais quanto funcionais. O problema principal é que a maioria das empresas tem estruturas orientadas para funções e não para processos.

### **2.3.3 Administrativo**

Esse tipo é um pouco mais complexo que o tipo orientado para *e-mail* e mais simples que o orientado para transações. O tipo administrativo tem características de sistema de correio eletrônico, mas com algumas capacidades a mais, 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. Em princípio, todos os softwares que tratem do fluxo de documentos e formulários podem ser classificados como Workflow administrativo. Exemplo desse tipo de Workflow contém rotinas de aprovação de despesas, controle de gastos com viagens, aprovação de ordens de compras e todos os conjuntos 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. Numa tentativa clara de elevar o grau de sofisticação desses softwares, seus fabricantes incluíram nas novas versões ferramentas que permitem o processamento de documentos e formulários inexistentes nas versões anteriores. Dentro desse conjunto, estão:

- Ferramentas para a criação de formulários simplificados.
- Ferramentas para rotear formulários.
- Interação com os formulários
- Ferramentas para notificações.

Sistemas de Workflow administrativo tratam de rotinas administrativas de caráter bem simples e, mesmo tendo ferramentas que permitem certa flexibilidade de programação, não podem ser comparados a sistemas mais sofisticados, como os que veremos adiante.

#### **2.3.4 Orientados para Objeto**

Sistemas de Workflow orientados para objeto são as versões sofisticadas dos sistemas de Workflow orientados para transações. Entretanto, é bom lembrar que a tecnologia *OO (Object Oriented)* não é nova, e muito menos surgiu com o desenvolvimento da tecnologia Workflow. A tecnologia OO surgiu em 1980 como uma evolução da tecnologia relacional, que, por sua vez, revolucionou os DBMS (*Data Base Management System*) estruturais.

Embora a tecnologia OO já exista há muito tempo, infelizmente ainda não foram definidos padrões internacionais que possam ser adotados por todos os fabricantes de tecnologia Workflow OO, o que, evidentemente, dificulta sua adoção por um número maior de fabricantes de tecnologia de informação e por usuários.

O propósito da tecnologia Workflow orientado para objeto é o de permitir o desenvolvimento de aplicações mais complexas, que permitam tanto a quem as programa como a quem as usa facilidades que com outra tecnologia seria impossível de conseguir. Por isso, esse tipo de tecnologia tem certas características muito particulares, como as que veremos a seguir.

- Encapsulamento  
É a possibilidade que o software deve ter para combinar múltiplos procedimentos em uma única representação procedural.
- Direito de herança  
É a possibilidade de criar novos procedimentos baseados em regras e propriedades herdadas de procedimentos já existentes. Isso poupa tempo e evita erros nas definições dos novos elementos ou objetos.
- Integridade referencial  
Características que garante que qualquer informação adquirida ou passada por herança será igual à informação que tiver dados origem às informações descendentes.
- Procedimentos compostos  
Ferramentas que o software deve ter para permitir a construção de procedimentos aninhados em outros procedimentos.
- Bibliotecas procedurais  
Coleção de procedimentos que podem ser recombinaados e reusados em outros procedimentos de tal forma que qualquer modificação que um procedimento vier a sofrer numa biblioteca dessas se refletirá imediatamente em todos os procedimentos que dependam daquele foi modificado.

### **2.3.5 Baseado no Conhecimento**

O último tipo de Workflow que merece nossa atenção é baseado no conhecimento. Isso significa dizer que o sistema Workflow baseado no conhecimento tem características e ferramentas que permitem aprender com seus próprios erros e acertos. Qualquer sistema baseado no conhecimento tem tecnologia para ir além da execução pura e simples das regras preestabelecidas e incorporar exceções a seus procedimentos.

Inteligência artificial é uma das tecnologias que permitem a sistemas de Workflow baseados no conhecimento aprenderem consigo mesmo. Outra tecnologia são os sistemas especialistas, desenvolvidos para poderem inferir solução a partir da vivência de fatos do cotidiano.

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 neurais, pode se a solução para as freqüentes mudanças que um fluxo deva sofrer para acompanhar a dinamicidade do processo de negócio de qualquer tipo de empresa. Todas essas tecnologias possibilitariam a um sistema Workflow baseado no conhecimento “aprender” com as exceções que fossem ocorrendo e com isso as fosse incorporando ao fluxo de trabalho. Esse tipo de Workflow seria a supra-sumo da tecnologia desenvolvida especialmente para automatizar processos. Entretanto, infelizmente, essa tecnologia ainda não está disponível, embora se saiba que vários fabricantes de software trabalham na criação de protótipos que podem vir a ser uma nova geração de sistemas Workflow.

## **2.4 Estrutura do Workflow**

Por definição, todo sistema de Workflow deve ser construído sobre uma arquitetura de cinco níveis principais. Cada um desses níveis tem diferentes funções que se complementam para formar o fluxo de trabalho automatizado. Todos têm a mesma

importância, o que significa dizer que nenhum deles pode ser excluído da estrutura. O que varia é o tamanho de fluxo em sua totalidade, que pode ter um número maior ou menor de atividades.

Os cinco elementos de um sistema de Workflow são:

- Processo (dividido em atividades);
- Instâncias ou casos;
- Pastas;
- Papéis;
- Documentos.

#### **2.4.1 Instâncias ou Casos.**

Elemento particular de um ambiente Workflow. Casos, ou instâncias, como é mais conhecido, 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 Instância. A quantidade de instâncias que podem ocorrer numa atividade está diretamente ligada ao tempo de ciclo da atividade. Por exemplo, cada nova fatura a ser emitida no faturamento é um caso. O processamento de uma única nota fiscal a se paga pelo Contas a Pagar é um caso. A execução de uma ordem de produção é um caso.

#### **2.4.2 Pasta**

O sentido de organização está presente em todos os detalhes num sistema Workflow. Para organizar e-mails, dados, informações, textos, voz e imagens foram criadas o nível que identifica o elemento Pasta, que é a reunião lógica de documentos. Uma pasta pode conter dados de todos os tipos existentes hoje, textos, imagens, sons, de uma única fonte ou várias fontes diferentes. Em um sistema Workflow os documentos trafegam pela rede, que suporta o fluxo de trabalho de forma organizada e, ao chegarem ao destino, devem ser arquivados de forma a permitir sua fácil e rápida recuperação.

### 2.4.3 Documentos

Documentos, e dados são o nível mais elementar de uma estrutura Workflow. Na verdade, o conceito de documentos, dentro de sistemas de informação, não é comum. Documentos são coleções de dados que colocados numa pasta serão usados por uma instância dentro de um processo. Documentos contêm não apenas dados, mas, também, informações sobre o formato, o processamento e apresentação desses mesmos dados. Um Objeto carrega a informação e as instruções para o processamento dessa informação num único elemento.

## 2.5 Os três Rs do Workflow

Na literatura internacional, os três elementos primários de um ambiente Workflow começam por R. Entretanto, em português os três R se transformam em dois R e um P, quando traduzimos ao pé da letra as palavras originais. Os três Rs são:

Em inglês: Roles, Rules and Routes.

Em português os 3 Rs são:

- Papéis
- Regras
- Rotas

### 2.5.1 Elementos Papel

Toda empresa tem um organograma. Seja formal, bem desenhado, com todos os níveis hierárquicos corretamente alinhados; ou seja, informal, daquele tipo que todo mundo sabe quem é quem na organização sem que um organograma tenha jamais sido escrito. Às vezes, quando é preciso, o organograma é novamente desenhado à guisa de atualização, outras vezes rabisca-se o quem e quem numa folha solta de papel, quem por

acaso estava à mão numa sala de reunião, sem que alguém se dê ao trabalho de formalizá-lo depois.

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

### **2.5.2 Regras**

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. Entretanto, um documento não é somente um conjunto de dados ou informações. Associadas a ele existem regras que especificam com clareza e exatidão a operação do documento, quais as atividades que devem recebê-lo, quais as rotas a seguir e quais cuidados especiais, se houver, o documento deve ser submetido.

Nada, num sistema Workflow acontece por acaso, mesmo se o software for quarto nível, provido de IA, a garantia que a implementação do sistema terá sucesso encontra-se no conhecimento e organização prévios das regras que sustentam o fluxo de informação no sistema Workflow. Quanto mais detalhada for a descrição das regras que suportam cada documento, mais seguro será seu processamento, não dando chance para que um papel qualquer comprometa a cadeia de valores [MED92].

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

### **2.5.3 Rotas**

Rotas para Workflow são o controle de movimentação exercido sobre os documentos. Os comandos do elemento Rotas 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 viagem é iniciada no Workflow.

A Rota é um 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 cinco tipos de Rotas e cada um desses tipos tem um propósito que não deve ser confundido como simples sofisticação da ação de transferência, na fig. 1.0 são mostrados graficamente alguns tipos de rotas. Os tipos de rotas utilizados são:

- Serial
- Paralelo
- Concorrente
- Condicional
- Dependente

#### **2.5.3.1 Rota Serial.**

Cada atividade tem apenas uma atividade anterior e apenas uma atividade posterior. Além disso, cada atividade deve ser complementada antes que o trabalho seja enviado à atividade seguinte.

#### **2.5.3.2 Rota Paralela.**

Se dá quando um grupo de atividades pode ocorrer ao mesmo tempo e tem a mesma atividade anterior e a mesma atividade posterior. Um exemplo disso é o de um documento que precise de três aprovações para ser enviado a um cliente, ou uma fatura que deve ser paga, mas só depois de aprovada por duas, quatro pessoas. Embora todas as aprovações sejam independentes entre si, elas podem ocorrer ao mesmo tempo, o que agilizaria sua aprovação.

### **2.5.3.3 Rota Concorrente**

É basicamente do mesmo tipo que o paralelo, com a diferença de que a atividade anterior e a atividade posterior devem começar ao mesmo tempo.

### **2.5.3.4 Rota Condicional**

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

### **2.5.3.5 Rota Dependente**

É uma rota necessária ao término de outra atividade. Na verdade, cada rota é dependente do término da atividade anterior. Entretanto, uma rota dependente define a necessidade do término de uma atividade anterior quando essa atividade não é a imediatamente anterior.

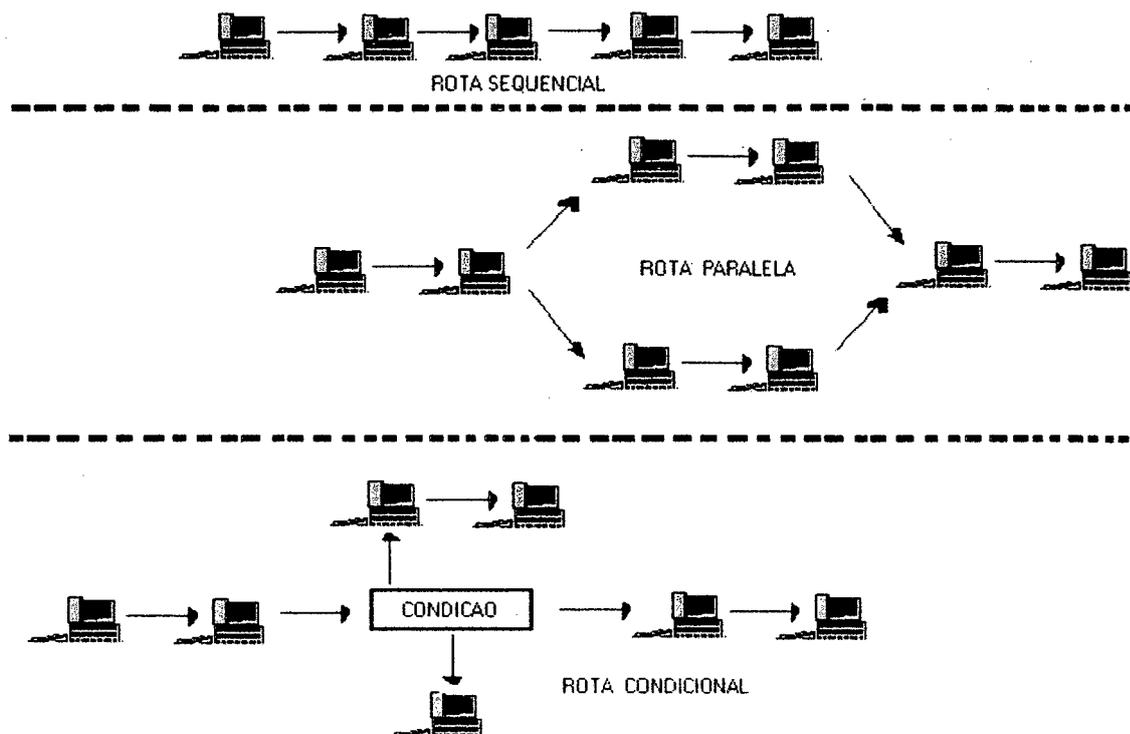


Figura 1.0 - Gráfico que mostra alguns dos tipos de rotas.

## 2.6 O Ambiente Workflow

O ambiente Workflow pode ser dividido em três partes: Hardware, software e o sistema de Workflow. Cada uma dessas partes determina o grau de sofisticação e especialização do ambiente Workflow implantado. Assim sendo, podemos concluir de antemão que a tecnologia Workflow pode ser implementada em qualquer ambiente e sobre qualquer plataforma de hardware e software. Entretanto, quanto mais simples for o conjunto hardware-software-sistema, menores serão as funcionalidades disponíveis, sendo que a presença ou a ausência de mais ou menos características tecnológicas vai se refletir no grau e na profundidade dos ganhos de produtividade do fluxo de trabalho que permeia o processo produtivo.

### 2.6.1 Hardware

A infra-estrutura que é chamada de plataforma de hardware é dividida em duas partes: computador e rede.

## **2.6.2 Rede**

A base para a existência de uma ambiente Workflow é a rede sobre a qual esse ambiente será implementado. Uma rede simples denota um fluxo de trabalho também simples, enquanto uma complexa serve a processos maiores e mais elaborados. Assim sendo, a menor rede para se implementar um sistema Workflow deve ter dois computadores interligados entre si. Esses dois computadores serão estações de trabalho correspondentes ao início e ao fim do fluxo de trabalho. Já as maiores redes suportando sistemas Workflow podem chegar a ter milhares de computadores perfeitamente integrados e sincronizados por meio da automatização do fluxo de trabalho que o sistema Workflow possibilita existir.

## **2.6.3 Computador**

Para implementar um sistema Workflow não há um computador específico. Entretanto, existe uma plataforma de hardware que melhor se adapta à natureza do workflow. Essa plataforma tem o nome genérico de cliente-servidor.

Na plataforma cliente-servidor a máquina principal responde pelo papel de servidor de dados. Talvez tenha sido esse o principal agente de mudança da antiga forma de trabalhar para a atual baseada em LANs. Antes da forma cliente-servidor de trabalhar, o compartilhamento de dados e informações era feito de forma pontual e estanque, com cada usuário passando-os, ou recebendo-os, por meio de disquetes de e para as máquinas dos outros usuários.

### **2.6.3.1 Arquitetura workflow cliente-servidor**

Qualquer aplicação que queira tirar proveito dessa arquitetura deve ser programada para ter pelo menos duas das três camadas distribuídas. Elas são:

- Comunicação com o usuário.

Essa é a camada da aplicação visível ao usuário, são as telas com as quais ele interage para executar sua parte do sistema. Essa camada é sempre executada no computador chamado de cliente e faz o papel de elo de ligação entre o usuário e o núcleo da aplicação.

- **Lógica da Aplicação.**

A camada seguinte, de cima para baixo na estrutura cliente-servidor, é a responsável pelo processamento pesado da aplicação. Ela é chamada de lógica da aplicação, porque é responsável pela parte do sistema que contém o cerne da aplicação.

- **Servidores de Arquivos e Bancos de Dados.**

A camada mais baixa da estrutura é formada pelos servidores de arquivos e de banco de dados. Nela ficam os computadores com os dados que serão acessados pelos usuários rodando aplicações na primeira camada que se utilizarão programas da Segunda camada para fazer a ligação entre a primeira e a última camada.

#### **2.6.4 Software**

Para usufruir de uma tecnologia tão avançada quanto a que está presente em todo o ambiente cliente-servidor, é necessário que o software, tanto produtos quanto aplicativos, também o sejam. Na categoria de produtos de última geração acham-se os bancos de dados orientados a objetos e que podem ser distribuídos por várias máquinas, estejam elas contíguas ou não.

### **2.7 Implantação de um Sistema de Workflow**

O ciclo Workflow tem cinco etapas, mostradas a seguir:

1<sup>a</sup> Etapa: Análise do fluxo de trabalho (workflow) atual.

- 2ª Etapa: Projetar o modelo de informação do fluxo de trabalho que se quer automatizar.
- 3ª Etapa: Programar o modelo de informação, definido e detalhando cada um dos elementos contido nele.
- 4ª Implantar o Workflow
- 5ª Atualizar o modelo de informação implantado.

### **2.7.1 Etapa 1. Análise do fluxo de trabalho (workflow) atual.**

Para projetar um novo modelo de informação é preciso partir de alguma realidade. Isto pode ser feito analisando-se o modelo atual a fim de conseguir-se os elementos necessários para projetar o novo fluxo de trabalho.

As etapas que precisam ser seguidas fazem parte de uma metodologia, entre inúmeras existentes. Em síntese pode-se dizer que são necessários três grandes momentos nessa análise:

#### **Início**

Como o processo é executado hoje

#### **Meio**

Quais as melhorias que precisam ser implantadas.

#### **Fim**

Modelo conceitual do novo processo

### **2.7.2 Etapa 2: Projetar o modelo de informação do fluxo trabalho que se quer automatizar.**

Para chegar a essa Segunda etapa é necessário que a empresa ou, mais especificamente, o processo, ou, de forma mais controlada ainda, o subprocesso, tenha

passado por algum tipo de reorganização como o mostrado na primeira etapa. A seguir as etapas descritas que devem ser cuidadosamente levantadas:

**Início**

Sob quais condições o processo inicia.

**Programação**

Descrever os tempos máximo e mínimo permitidos para cada atividade.

**Pré-atividade**

Estabelecer o montante em que determinada pessoa pode iniciar atividade certa.

**Execução**

Descrever tipos de ferramentas, aplicações, metodologias e técnicas usadas para processar um item de trabalho.

**Notificação**

Descrever as condições sob as quais as pessoas serão notificadas sobre um evento do processo.

**Pós-Atividade**

Estabelecer o momento em que determinada pessoa pode completar certa atividade e o que deve ocorrer quando ela for terminada.

**Segurança**

Descrever quem estará autorizado a participar do processo, que funções será autorizado a executar e que informações poderão ser manuseadas.

**Auditoria**

Descrever quais eventos serão auditados e em que nível de detalhamento.

### **Término**

Descrever as condições em que o processo deverá terminar ou ser interrompido.

#### **2.7.3 Etapa 3: Programar o modelo de informação, definindo e detalhando cada um dos elementos contidos nele**

Depois de projetar o modelo de informação, feito na primeira etapa, pode-se passar à seguinte, programar o modelo dentro do software escolhido para servir de ambiente Workflow.

#### **2.7.4 Etapa 4: Implantar o Workflow**

Somente depois de se adquirir experiência com a filosofia Workflow e com o software que se estiver usando deve-se tentar começar a automatizar um procedimento de forma mais arrojada, isto é, montando grandes fluxos de uma única vez. Entretanto, antes de se pensar em tecnologia, é preciso pensar em sociologia. É preciso modificar a cultura organizacional para garantir o sucesso na implantação de uma tecnologia como esta.

#### **2.7.5 Etapa 5: Atualizar o modelo de informação implantado**

A última etapa, atualização do modelo de informação implantado, divide-se em duas partes:

- a) Revisão do modelo implantado;
- b) Atualização do modelo implantado.

Uma etapa de revisão é importante à medida que se tiver a intenção de melhorar continuamente o sistema implantado. Um sistema nasce com várias deficiências e algumas delas vão-se acentuando com o passar do tempo se não tomarmos medidas que possam solucioná-las. Por isso, a etapa de revisão deve ser encarada com seriedade, pois por meio dela podemos descobrir quais as funcionalidades que não estão comportando-

se conforme o previsto, o que está aquém do que se esperava, como as pessoas envolvidas em sua utilização estão comportando-se e uma série de outros pontos que podem ser analisados.

## **2.8 Vantagem no uso da tecnologia de Workflow**

O uso de sistemas de workflow em uma organização pode trazer ao usuário diversas vantagens, tais como [AMA98]:

- **Garantia da integridade do processo:** a utilização de WFMS garante que as regras do processo não serão desrespeitadas (como por esquecimento, desconhecimento das normas, tentativa de fraude, etc.), e que a ordenação das tarefas será seguida. Processos com muitas regras, tipo análise de empréstimo, são os principais benefícios;
- **Integração das atividades da empresa:** com o uso de sistemas de workflow, atividades desenvolvidas por divisões diferentes de uma empresa podem compor um único processo. Isto é muito importante, uma vez que a empresa deve ser vista como uma entidade única e não como uma coleção de departamentos dispersos. Por exemplo, um cliente pode solicitar uma informação sobre a tramitação de um processo de seu interesse. Se este processo envolve mais de um departamento, e se não há integração entre seus respectivos sistemas de informação, somente o departamento onde o processo está, terá condições de prestar esta informação, resultando em insatisfação do cliente e prejuízo para a imagem da empresa. Com essa integração há um melhor atendimento aos clientes.
- **Eliminação do tempo de espera entre as atividades:** com o uso de WFMS, a coordenação entre tarefas são administradas automaticamente. O processo pode fluir e ser concluído mais rapidamente porque o espaço de tempo em que ele permanece em um estado “pendente” (sem que ninguém esteja trabalhando nele) é virtualmente eliminado. Para isto tornar-se verdade, vale ressaltar que a substituição do papel por meio eletrônico é fundamental;

- **Manipulação eletrônica de documentos:** a troca dos registros em papel por registros eletrônicos garante a consistência, a utilização de regras, o armazenamento e a pesquisa rápida aos registros. Essa vantagem é especialmente importante para a automação de processos baseados em documentos, como o processamento de empréstimos bancários, etc.

Essas vantagens permitem o alcance de alguns benefícios para a organização, tais como: melhor controle no andamento das tarefas e obtenção de uma melhor performance da organização [FAU96].

## **2.9 Modelagem de Workflow**

O desenvolvimento de uma aplicação de workflow tem início a modelagem dos processos a serem automatizados. O modelo correspondente a uma descrição do processo de forma sucinta e sem ambigüidades. O modelo captura e organiza as propriedades do processo. Estas propriedades podem ser: estáticas, que informam o modo como o processo é decomposto em tarefas e dinâmicas, que se referem ao uso e produção de relacionamento entre as tarefas. O mais importante é que o modelo fornece a base para responder perguntas sobre o processo e prevê seu comportamento [RID96].

O modelo conceitual de um Workflow permite a descrição do processo de negócios. Para cada estágio de trabalho, é necessário determinar quais as tarefas que compõem o processo, os seus relacionamentos umas com as outras, quem é o responsável pela execução de cada tarefa e quando uma deve acontecer [RID96].

Apresenta-se a seguir os seguintes modelos: o modelo de gatilhos, de redes de Petri, de Casati/Ceri, o modelo de ações e o modelo de interoperabilidade da WFMC.

### **2.9.1 Modelagem do Workflow baseada em Redes de Petri**

As redes de Petri aparecem como um bom formalismo para a modelagem de workflow. Este formalismo é independente da implantação, possui uma representação gráfica e tem uma base formal (não ambígua) dentre diversos aspectos [AAL96].

### **2.9.2 Modelo de Casati/Ceri**

Este modelo foi proposto por Casati, Ceri, Percini e Pozzi [CAS95]. Ele possui uma linguagem para sua definição chamada WFDL (Workflow Description Language). Seus elementos principais são: atividades, conexões entre tarefas e multi-atividades.

### **2.9.3 Modelo de Gatilhos**

O modelo de gatilhos ou “triggers modeling” é baseada no conceito de trigger, que pode ser entendido como o disparo de uma tarefa de workflow após o término de uma tarefa anterior.

### **2.9.4 Modelo de Ações**

O modelo de ações (também chamado de modelo ActionWorkflow) é baseado na teoria dos atos de fala [WIN87]. Este modelo utiliza uma estrutura simples, em que a tarefa é vista como uma comunicação entre dois participantes: o cliente e o executor. O cliente é aquele que solicita a realização de algo. O executor é aquele que executará o que foi solicitado pelo cliente.

### **2.9.5 Modelo de interoperabilidade da WFMC**

Este modelo foi concebido para permitir que diferentes ferramentas de modelagem de workflow e WFMC pudessem trocar informações entre si. Assim é possível modelar o workflow em qualquer ferramenta que suporte este padrão e implementá-lo em WFMC que também o suporte. Desta forma, o problema de

incompatibilidade entre a ferramenta de modelagem e a de implementação pode ser solucionado [WOR94].

## 2.10 Exemplos de Sistemas de Workflow Existentes

Serão mostrados a seguir dois sistemas de gerenciamento de workflow comerciais existentes denominados Action Workflow e Flowmark [ALV00].

O Action Workflow é um sistema de workflow da Action Technologies. Este modelo tem como elementos: o laço de workflow, as aplicações e os participantes. O laço de workflow é o elemento básico deste sistema e define o que deve ser feito. Geralmente são necessários vários laços para definirem o workflow completo. O conjunto desses laços é chamado de processo de negócios. O laço de workflow contém quatro partes: requisição, negociação, execução e aceitação. Além disso, temos dois participantes: o cliente e o executor.

Na requisição o cliente faz a proposta de trabalho a ser executado. Na negociação, cliente e executor concordam sobre o trabalho que será executado. Na execução, a ação é executada de acordo com os termos estabelecidos. Na aceitação, o cliente avalia os resultados do trabalho. A aceitação é baseada em uma condição de satisfação que é associada ao laço de workflow. Assim, o trabalho é executado com sucesso se as condições forem satisfeitas [ALV00].

As aplicações são integradas através de um mecanismo chamado STF (Standard Transaction Format). Este formato permite especificar laços de workflow. Quando uma aplicação utiliza uma especificação de workflow para o sistema de gerência de workflow, ele inicia a instância de workflow de acordo com a especificação.

Os participantes no Action Workflow são caracterizados por papéis e identidades. As identidades representam os seres humanos e os papéis agrupam os homens pelas suas funções na organização como, por exemplo, um gerente, etc.

Vários laços de Workflow podem ser associados para construir um workflow mais completo ou o processo de negócios. Estes laços podem ser conectados de forma seqüencial, condicional ou em paralelo. Na seqüência executa-se um laço após o outro. Na condição, um laço é executado se satisfaz a condição. No paralelismo cada laço de workflow pode ser iniciado várias vezes. Por exemplo, no processo de negócio de contratação de um empregado, para cada entrevista um laço de workflow deve ser iniciado.

O Action Workflow oferece três ferramentas: o Builder, a ferramenta Analyst e a ferramenta Manager. O Builder é usado para modelar os laços de workflow, os processos de negócios, os papéis e as identidades. Os elementos de modelagem disponíveis na interface gráfica são ícones representando laços de workflow e arcos para ligá-los. O conjunto de vários laços ligados por arcos forma o processo de negócios. A ferramenta Analyst é usada para executar estatísticas no processo de negócios. Permite calcular automaticamente os tempos de duração das etapas do workflow. A ferramenta Manager executa os laços de workflow e os processos de negócios. Ela permite consultar o status do workflow, instanciar novos workflow's, consultar o trabalho pendente para os usuários e diversas outras operações. O Manager é o componente central do Action Workflow.

O **FlowMark** é uma ferramenta desenvolvida pela IBM (International Business Machines) que é formada por quatro áreas de modelagem: processos, programas, estruturas de dados e Staff [ALV00].

Os programas aparecem como uma das áreas de modelagem. Eles são aplicações baseadas em computador como um editor de texto. O programa é usado pelo usuário para executar a tarefa. Os programas usados devem ser registrados no FlowMark através da definição de nome, seus parâmetros de entrada e saída e sua plataforma.

Outra área de modelagem são as estruturas de dados, que correspondem a uma lista de variáveis. Algumas delas são pré-definidas, logo, não podem se excluídas. As

outras podem ser definidas de acordo com os requisitos da aplicação. As estruturas de dados são utilizadas para modelar os parâmetros de entrada e saída de um processo. Para transferir dados de um processo para outro, os parâmetros de saída de um processo devem associar-se aos parâmetros de entrada do outro.

Os processos são associados aos usuários para execução através de vários critérios. Cita-se a exemplo, a posição do usuário na estrutura da organização. O FlowMark fornece um conjunto de construções para modelar a estrutura da empresa como níveis, pessoas, papéis, organizações junto com relacionamento papel-pessoa e organização-pessoa. Esse conjunto corresponde ao staff que é outra área de modelagem.

Os empregados que trabalham no processo definido pelo FlowMark, são identificados por um identificador único. Possuem uma senha e atributos como o primeiro nome, etc. Aqui, portanto, especifica-se que pessoas podem ser substituídas por outras, bem como quais são autorizadas a acessarem as funções de gerência do FlowMark.

São definidos os objetos chamados de papéis que definem os papéis organizacionais dentro da empresa. Um papel possui um nome e os membros associados a ele. O objeto define a estrutura de uma organização. A organização pode ser decomposta em outras sub-organizações, e possui uma pessoa associada como gerente e várias pessoas como membros.

Os processos correspondem à outra área de modelagem. São especificados através de várias construções: processo, atividade-programa, atividade-processo, bloco, pacote, conector de controle e conector de dados. O elemento de modelagem básico é o processo que define todas as atividades que devem ser executadas. A atividade pode ser uma atividade-programa ou uma atividade-processo. A atividade-programa é executada por um usuário que tem um programa associado a ela. A atividade-processo não necessita ser completada por um único usuário e tem um processo associado a ela.

A seqüência de execução entre as atividades é definida por conectores de controle. Esse controle especifica as condições de transição, de entrada e saída das atividades. Desta forma uma atividade só é iniciada ou concluída quando suas condições de entrada ou saída, respectivamente, são avaliadas como verdadeiras.

Os conectores de dados especificam os fluxo de dados entre os parâmetros de entrada e de saída das atividades conectadas. Um bloco agrupa atividades no processo. Um pacote é utilizado para iniciar um número de atividade do mesmo tipo.

O FlowMark agrupa suas ferramentas em duas categorias: de tempo de construção e de tempo de execução. As ferramentas de tempo de construção são usadas no tempo de definição do processo. Elas definem através de editores de definição de pessoas, de papéis e de organização e especificam o processo, através do editor de processos. Possuem também o editor de registros de programas, de estruturas de dados e de animação do processo.

As ferramentas de tempo de execução são usadas durante a execução do processo. O FlowMark possui uma lista de trabalho que é usada por usuários para visualizar e executar as tarefas associadas, e uma ferramenta de planejamento que é usada para iniciar várias atividades de mesmo tipo em um dado momento.

O FlowMark é um sistema centrado no banco de dados, que segue o paradigma cliente/servidor. As listas de trabalho são servidores, que executam instâncias de processos. O servidor responde às requisições dos clientes acessando o banco de dados para executar seus serviços. Este sistema é específico apenas para a definição do staff, sendo livre para ser utilizado em diversas áreas de aplicação.

O Workflow Builder 2.0 é uma ferramenta desenvolvida pela ORACLE que é formada por quatro áreas de modelagem: processos, programas, estruturas de dados e monitor de processos. Esta ferramenta será estudada em detalhes no capítulo 5.

## **CAPÍTULO 3**

### **UTILIZANDO AGENTES MÓVEIS**

Os agentes móveis formam um novo paradigma de programação na área de sistemas distribuídos. Eles possuem a capacidade de migrar de nodos em redes heterogêneas. Podem ser definidos como objetos que possuem comportamento, estados e localização. Agentes Móveis são autônomos porque após terem sido invocados eles decidem quais locações visitar e quais instruções executar.

Para algumas aplicações que suportam domínios como recuperação de informação em ambientes distribuídos, gerência de rede, comércio eletrônico ou computação móvel, a aplicação de Sistemas de Agentes Móveis é promissora.

A utilização da tecnologia de agentes móveis tem sido campo de extensas pesquisas por grandes empresas que se esforçam em disponibilizar aplicações práticas que facilitem a interação do usuário com sistemas informatizados. Enfoque especial tem sido dado ao uso da tecnologia de agentes na Internet, onde as características próprias da rede podem trazer enormes dificuldades para o uso das tecnologias tradicionais.

Sistemas de Agentes Móveis têm recebido muita atenção ultimamente. Os Agentes de mobilidade, representados remotamente aplicações e usuários, propiciam vantagens para certas aplicações como a redução do tráfego na rede, maior flexibilidade, possibilidade de balanceamento de carga e processamento assíncrono, permitindo assim o desenvolvimento de soluções mais eficientes para problemas relacionados a comércio eletrônico, computação móvel, gerência remota de recursos, data mining, entre outros.

#### **3.1 Histórico**

No início da década de 80, a comunidade de Inteligência Artificial, desencorajada pela falta de progresso, após 30 anos de pesquisa em projetos de sistema, começou a

explorar novas áreas onde sistemas de AI pudessem ter um domínio mais dinâmico. Ao invés de olhar para resultados simulados, simbólicos em mundos artificiais, começaram a explorar as possibilidades de interações complexas com o mundo físico, através de um mecanismo denominado agentes.

Vários pesquisadores como Marvin Minsk, Oliver Selfridge, Alan Kay, Nicholas Negroponte, Rodney Brooks e Pattie Maes, passaram a estudar problemas que pudessem demonstrar algum tipo de comportamento dos agentes.

Ao mesmo tempo, uma outra comunidade estava se formando usando também o termo agente. Era a comunidade de agentes de software, que estava explorando o desenvolvimento de partes de código menores e mais confiáveis. A idéia era desenvolver programas que pudessem agir separadamente, podendo mover informações entre dois ambientes distintos.

Entretanto, uma barreira surgiu para a comunidade científica, a quantidade de termos usados para descrever agentes: intelligent agents, intelligent interfaces, adaptive interfaces, knowbots, knobots, softbots, userbots, taskbots, personal agents e network agents, dentre outros [CHL95].

Pode-se definir um agente como sendo qualquer coisa que percebe seu ambiente através de sensores e atua neste ambiente através de reagentes. Atualmente, uma das definições mais aceitas descreve um agente como sendo um programa de software que auxilia o usuário na realização de alguma tarefa ou atividade. Agente é a palavra utilizada para designar uma entidade inteligente e autônoma. A palavra autônoma, neste caso, significa que cada agente possui sua própria existência, a qual não é independente da existência de outros agentes.

### **3.2 Características de Agentes**

Os agentes possuem atributos específicos que modelam seu comportamento, a seguir são mostradas algumas destas características [GEN94]:

**Autonomia:** tem a capacidade de executar ações sem a intervenção direta de operadores e possuem algum tipo de controle sobre suas ações e seu estado interno, podem decidir o que fazer quando a tarefa é executada com sucesso o falhar;

**Habilidade Social:** para conseguir seus objetivos interagem com outros agentes e em determinados momentos podem interagir com seres humanos (operadores).

**Reatividade:** os agentes detectam seu ambiente (que pode ser o mundo físico, um usuário, outros agentes, a Internet etc) e respondem aos estímulos dele recebido. Em alguns casos o agente fica adormecido e só despertará se certas mudanças em seu ambiente ocorrerem.

**Iniciativa:** os agentes não agem simplesmente em resposta ao seu ambiente, eles são capazes de atingir suas metas tomando a “iniciativa”.

**Continuidade Temporal:** os agentes são processos em execução contínua e podem assumir diferentes estados, ativos, foreground, adormecidos, em background.

**Orientação a Objetivos:** um agente deve ser capaz de trabalhar com tarefas complexas em alto nível. Decisões de como as tarefas deve ser divididas em sub-tarefas menores e em que ordem ou de que forma estas sub-tarefas devem ser executadas deve ser uma decisão tomada pelo agente.

### **3.3 Características Avançadas de Agentes**

Alguns cientistas da IA (Inteligência Artificial), consideram que os agentes devem possuir propriedades identificadas anteriormente, e sejam implementados usando conceitos mais usualmente aplicados aos seres humanos. Os agentes que se enquadram nesse grupo têm uma ou mais das seguintes características [GEN94, SHO93]:

**Mobilidade:** é a habilidade de um agente mover-se em uma rede, ocupando diferentes nós e recursos ao longo de sua execução;

**Benevolência:** supostamente agentes não têm objetivos contraditórios e todo agente conseqüentemente sempre tentará fazer o que lhe é solicitado;

**Racionalidade:** a habilidade de raciocinar durante a execução de uma tarefa é um dos aspectos chave da inteligência. Raciocinar implica que um agente tem habilidade para deduzir e extrapolar, baseado em conhecimento atual e experiências.

**Adaptabilidade:** um agente deve ter a capacidade de ajustar-se aos hábitos, métodos de trabalho e preferências de seus usuários;

**Colaboração:** um agente não deve aceitar (e executar) instruções “indesejáveis”, mas deve levar em conta que o usuário humano comete erros, omite informações importantes e/ou fornece informações ambíguas. Neste caso, um agente deve chegar estas ocorrências questionando o usuário. Um agente deve ter permissão de recusar tarefas que possam causar sobrecarregar a rede ou causar danos a outros usuários desta rede.

### 3.4 Classificação de agentes

Combinando algumas propriedades apontadas, é possível estabelecer condições necessárias (ou suficientes) para certos tipos de agentes, tal como acabamos de ver para agentes autônomos. Por outro lado, é também possível estabelecer graus na noção de agente. Por exemplo, [GEN94, SHO93] apresentam dois graus (degrees of agenthood – propriedade de ser agente).

1. **Noção Fraca de Agente:** sistema computacional com as seguintes propriedades: autonomia, habilidade social, reatividade e pró-atividade.

2. **Noção Forte de Agente:** sistema computacional que além das propriedades anteriores possui também estados mentais (racionalidade), crenças, desejos e intenções.

Vale lembrar que as propriedades de agentes e as noções de agentes são relativas ao ambiente. Por ambiente entenda-se o meio do qual o agente obtém informação através de sensores e sobre o qual o agente pode atuar. Por exemplo, a propriedade de mobilidade é dependente do ambiente no sentido que o ambiente tem que responder a ações de mobilidade da parte dos agentes.

### 3.5 Tipos de Agentes

Os agentes podem ser classificados em várias categorias, de acordo com suas características. A seguinte classificação é apresentada por [THO97,GEN94]:

**Conselheiro:** oferece ajuda e treinamento. Ensina os passos iniciais para usar um determinado sistema. Fornece suporte contínuo, observando todas as ações do usuário, as quais ele pode interceptar e pedir confirmação. Pode ser consultado para mostrar como executar uma atividade particular, ou então, surgir métodos alternativos e mais rápidos para executá-la.

**Guia:** ajuda a navegação em bancos de dados e hipermídia. Classifica, recupera e filtra grandes quantidades de informações, apresentando somente os dados relevantes e importantes aos usuários, no formato personalizado. Fornece caminhos apropriados para o usuário navegar pelo banco de dados, e auxilia-o caso sintá-se “perdido”.

**Empregado:** executa as atividades tediosas ou repetitivas. Atividades são executadas imediatamente, e algum tipo de feedback pode ser fornecido tanto pelo usuário como pelo próprio agente.

**Representante:** trabalha a ausência do usuário. De certa forma, seria parecido ao agente empregado citado anteriormente, exceto pelo fato de que as atividades não precisam ser

imediatamente executadas ou então, são executadas somente após eventos específicos. Por exemplo, pode fazer backups de arquivos de madrugada ou fazer pedidos de compras, caso algum produto atinja o limite mínimo no estoque.

**Comunicador:** trabalha com outros usuários e seus agentes, para assim, conseguir executar as atividades as quais é incumbido. Pode, por exemplo, organizar reuniões, reunindo recursos e pessoas. Ou então, pode reunir um grupo de agentes para que assim possam executar uma atividade mais complexa.

### **3.6 Características que tornam uma aplicação apropriada para agentes.**

Tomando como bases às características comuns dos agentes, pode-se identificar um conjunto de características que tornam uma tarefa ou aplicação suscetível a uma abordagem com base na tecnologia de gentes. A seguir as propriedades especificadas por [THO97, SHO93]:

**Adaptação:** tarefa que requer um certo grau de adaptabilidade; o agente necessita desenvolver habilidades para executá-la aprendendo melhores ou novos meios. O que também inclui métodos para evitar falhas e se adaptar as próprias necessidades, desejos e objetivos pessoais do usuário.

**Pesquisa:** a tarefa não é completamente definida, o agente deve considerar uma grande quantidade de possíveis soluções, escolhendo uma das mais adequadas de acordo com sua experiência.

**Demonstração:** a tarefa envolve aprendizado e treinamento. Isto inclui ensinar os usuários a usar ferramentas de software de maneira mais eficaz e também, por outro lado, fornecer explicações de que o próprio agente está fazendo.

**Ajuda:** a tarefa requer um certo grau de cooperação entre o usuário e o agente. O agente poderia fazer críticas construtivas ao modo de trabalhar do usuário, ou dar “dicas” sobre como utilizar melhor os recursos do sistema.

**Autonomia:** a tarefa requer atenção constante ou regular, mas pouca ou nenhuma entrada ou interação. Dessa forma, delegar esta tarefa seria muito útil e benéfico. Um exemplo será o monitoramento de sistemas simples, onde uma mudança no comportamento poderia gerar a execução automática de alguma tarefa ou ação.

**Assincronia:** a tarefa tem um intervalo significativo entre seu início e término. Este intervalo poderia ser devido ao tempo de processamento de grandes quantidades de informação ou mesmo a falta de informações vitais em um determinado momento.

### **3.7 Que problemas os agentes podem resolver.**

A tecnologia de agentes resolve, ou promete resolver segundo [THO97, SHO93], muitos problemas de diferentes partes. Agentes móveis resolvem o importuno problema da transmissão na rede cliente-servidor. Na rede, a largura da banda é importante e algumas vezes raro recurso de aplicação distribuída. A transação solicitada entre um cliente e um servidor pode requerer muitas voltas sobre a rede para ser completada. Este tipo de operação cria um tráfego muito grande e consome muito da banda de transmissão. Em um sistema com muitos “clientes” e, ou mais transações, o total de requerimentos no tamanho da banda pode exceder a disponibilidade permitida, ocasionando uma performance muito ruim para as aplicações que estão envolvidas. Na utilização de agentes para buscar as solicitações ou transações e, enviando os agentes do cliente para o servidor, o fluxo na rede é reduzido. Dessa maneira, somente o que os agentes encontrarem será transmitidos pela rede, tornando a velocidade de transmissão maior.

A arquitetura toma decisões sobre onde um pedaço em particular da funcionalidade pode residir, baseado na banda de transmissão, tráfego na rede, volume da transmissão,

número de cliente-servidor e demais fatores. Se estas estimativas são justificáveis, ou a arquitetura toma uma decisão errada, a execução da aplicação poderá ser comprometida.

Infelizmente, uma vez que sistema tenha sido desenvolvido e a execução medida, isto frequentemente dificulta ou impossibilita as mudanças de autor e fixa os problemas. Arquiteturas baseadas em agentes móveis são potencialmente muito menos suscetíveis a estes problemas. Algumas decisões devem ser feitas para melhorar o tempo gasto com desenvolvimento e, o sistema é mais fácil de ser modificado depois de ser construído.

Essa arquitetura de agentes suporta adaptações da rede podendo fazer um novo desenho automaticamente. A arquitetura de agentes também pode resolver problemas criados por intermitência ou má qualidade da conexão com a rede. Atualmente, algumas aplicações na rede são pesadas para completar a transação ou obter localização de informação. Se a conexão cair, o cliente deve frequentemente reiniciar a transação do ponto de partida. Com a tecnologia de agentes o cliente poderá obter as informações, mesmo que a conexão não esteja ativa. O cliente pode trabalhar off-line. Os agentes podem completar as transações e retornar os resultados para o cliente quando for estabelecida a conexão.

A tecnologia de agentes também poderá resolver em alguns problemas na obtenção de informação, fazendo o computador realmente pensar por nós. Esta é uma solução difícil. A comunidade da inteligência artificial, tem lutado intensamente por mais de duas décadas. Este potencial, entretanto, é imensurável.

### **3.8 Arquitetura de agentes**

Mobilidade é uma característica comum na definição de agentes. A mobilidade do agente está em um objeto ativo que pode mover dados e códigos de funcionamento por muitos lugares dentro de um sistema distribuído [SHO93].

Agentes móveis devem possuir habilidades para executar operações em qualquer máquina dentro da rede, independentemente do tipo de processador ou do sistema

operacional. Com esta tecnologia, os agentes não precisam estar instalados em todas as máquinas que poderão utilizá-lo, eles automaticamente se movem para as máquinas quando precisam executar uma tarefa.

Portanto, esta é uma característica principal do sistema de agentes móveis, semelhante ao empregado pelo Java Virtual Machine (JVM). O dinamismo natural das classes e objetos Java combinam com habilidades avançadas da rede, fazendo do Java altamente qualificado para ser usado como plataforma de agentes móveis. Junto com as classes applets são lidas dinamicamente do servidor Web direto para browser, as classes são lidas em tempo real na rede e transmitidas para uma outra localização.

O sistema de agentes proporciona uma estrutura na qual os agentes móveis podem trabalhar. Junto com objetos distribuídos dentro do sistema baseado em CORBA contando com ORB (Objet Request Brokers), os agentes solicitam agentes hospedeiros, os quais proporcionam um ambiente no qual os agentes poderão ser executados. No caso das aplicações JAVA, o ambiente é uma JVM contendo servidor de objetos nos quais os agentes realizam suas operações. Outros agentes podem estar presentes e eles podem se comunicar entre si dentro das implementações.

### **3.9 Por que usar arquitetura de agentes**

A arquitetura de agentes possui vantagens consideráveis sobre a cliente-servidor e sistemas de objetos distribuídos. São três importantes vantagens [THO97]:

1. O desempenho de processamento com agentes é alto, reduzindo assim o volume de transmissão na rede e incrementando a execução. No CORBA, a funcionalidade se equivale, mas realizar chamadas de métodos remotos repetidos no servidor de objetos não é possível, porque os objetos CORBA não podem mover-se através da rede em tempo real.
2. A operação dos agentes independe da aplicação a qual o agente foi invocado. Os agentes operam assincronamente, evitando que a aplicação cliente aguarde

para obter os resultados. Isso é importantíssimo para usuários móveis que não estão sempre conectados a rede.

3. O uso de agentes permite melhorar a nova funcionalidade dentro dos sistemas para execução em tempo real. O sistema de agentes contém próprios mecanismos de distribuição de software automáticos. Como o CORBA não foi construído para suportar códigos móveis, novas funções devem ser instaladas manualmente.

Naturalmente, um sistema não agente pode exibir estas características com algum trabalho. Mas os códigos móveis podem suportar uma transferência de código executável para uma localização remota para uma execução assíncrona desde o início. A arquitetura de agentes móveis poderá ser recomendada para sistemas onde as características abordadas são requisitos essenciais.

Agentes móveis resolvem o problema de transmissão pela rede que a arquitetura cliente-servidor não soluciona. As solicitações ou transações do cliente para o servidor, e as, repetitivas perguntas/respostas são eliminadas como mostra a fig. 1.1:

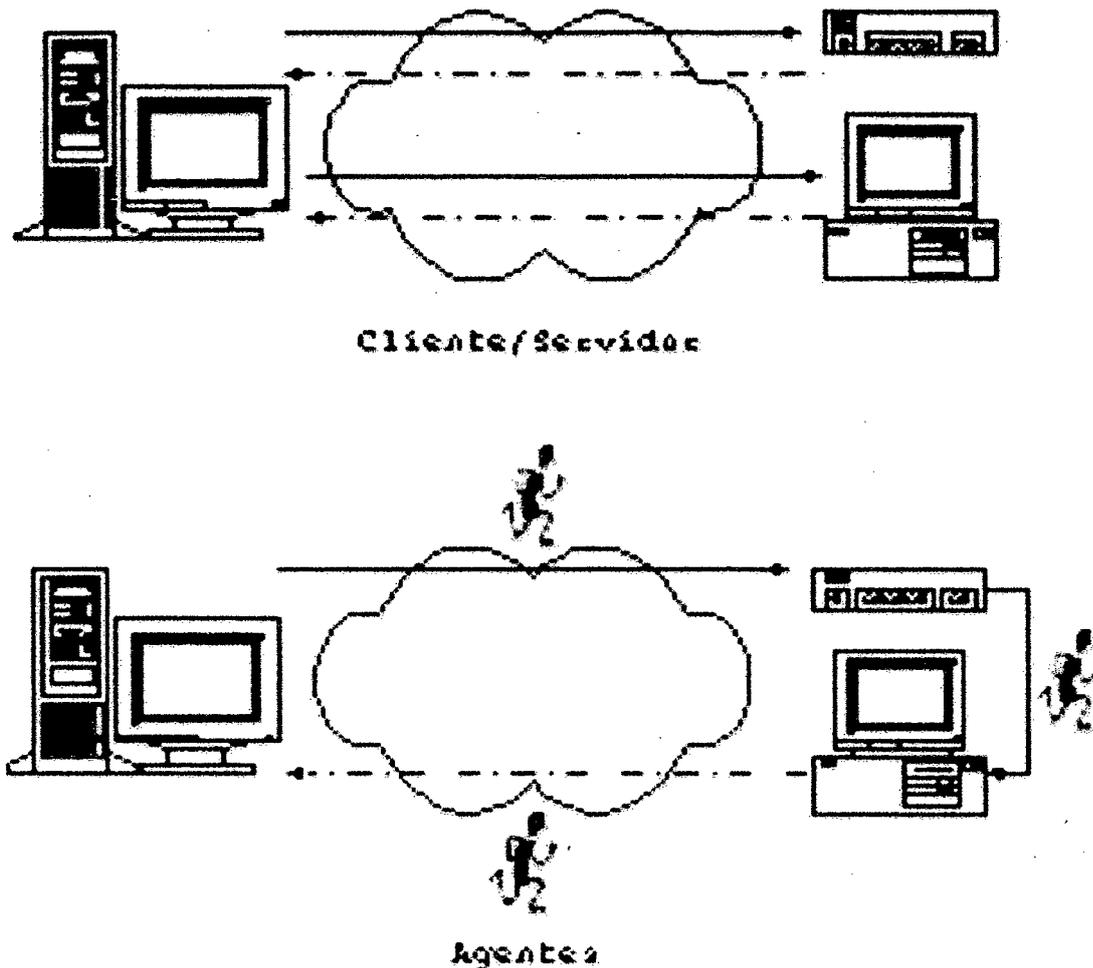


Figura 1.1 – Representação da estrutura de agentes x cliente-servidor

Agentes reduzem o risco na execução. Agentes deixam a decisão sobre a localização do código (Client vs. Server) para ser empurrado, ou enviado, no final da rotina do processo quando é possível saber mais sobre a finalidade da aplicação. Na realidade, a arquitetura deixa sempre para depois da construção do sistema e em operação para trocas de informações. Agentes reduzem também problemas originados por intermitência ou não dificuldades de conexão. Eles são construídos vagarosamente e lentamente para trabalhar “off line” e os resultados destas comunicações são formulados na retaguarda e apresentados quando a aplicação está “on line”.

### 3.10 Comunicação entre agentes.

Como há uma grande variedade de software no mercado, criados por diversas pessoas, dos mais diferentes países, obviamente há uma enorme diferença entre a forma e estrutura de seus programas. Como os agentes se propõem a comunicar entre os aplicativos do usuário, sejam quais forem, há uma necessidade de criar-se um padrão de comunicação. Logicamente, é necessário que primeiro se crie um padrão entre os próprios agentes, para depois de criado e testado, passem a usá-lo também nas aplicações comuns.

Segundo [THO97] criar um padrão não é uma tarefa simples. Há necessidade de ter-se um grupo de desenvolvimento. Este grupo tem que levar em consideração a heterogeneidade dos programas, escritos em diferentes momentos, por pessoas distintas, em diferentes linguagens, com interfaces também diferentes. Para não ficar restrito aos problemas decorrentes da criação de um padrão, discute-se a seguir um exemplo de tentativa de definição de padrão.

O grupo DARPA Knowledge Sharing Effort, usa a abordagem declarativa, que é baseada na idéia que a comunicação pode ser modelada com a troca de afirmações declarativas, sendo, porém, ao mesmo tempo compactada e suficientemente expressiva para comunicar uma grande quantidade de tipos de informações.

Este grupo definiu uma Linguagem de Comunicação de Agentes (ACL), que satisfaz estas necessidades. A ACL pode ser mais bem imaginada como sendo construída de três partes – seu vocabulário, uma linguagem interna chamada KIF (Knowledge Interchange Format) e uma outra linguagem chamada KQML (Knowledge Query and Manipulation Language). Uma linguagem ACL é uma expressão KQML na qual os argumentos são termos ou sentenças em KIF, formadas por palavras encontradas no vocabulário da ACL. O vocabulário da ACL é armazenado em um dicionário com as palavras apropriadas às áreas comuns de aplicações. Cada palavra tem notações formais (em KIF) e descrição em inglês. É um dicionário aberto, permitindo acréscimos de

palavras nas áreas já existentes ou então de novas áreas que possam surgir. Tem-se então uma linguagem padrão de comunicação entre agentes.

Mas como ficam os programas já escritos, ou seja, as aplicações que o usuário tem e que gostaria que agentes lhe auxiliassem. Temos, então, algumas abordagens: Transdutor – a aplicação envia a mensagem, o transdutor a transforma em ACL e passa ao agente. Quando da resposta, o transdutor converte na linguagem da aplicação. Tem-se alguns problemas: tempo de resposta (a tradução de ida e volta), a base de conhecimento do transdutor (quanto mais souber, mais aplicações poderá traduzir, e mais espaço ocupará).

Mas a vantagem de funcionar em qualquer aplicativo. Empacotador – insere-se código dentro da aplicação para permitir a comunicação. Isto permite ao empacotador examinar diretamente as estruturas de dados do programa e mudá-las. Torna a comunicação mais eficiente, por não ter que traduzir e diminuir a comunicação serial, além de permitir que os programas que antes não podiam se comunicar, agora tem esta facilidade.

Porém, necessita que o código original esteja disponível. Reescrita – é a atitude mais drástica, rescrever o programa todo. Obviamente aumenta em muito a eficiência e capacidade do programa, muito além das anteriores, mas altera de tal forma o código, que praticamente o programa “ganha” uma nova versão.

### **3.11 Desempenho dos agentes**

O termo de desempenho identifica a eficiência de um agente. Entretanto, é necessário estabelecer um padrão para determinar o que seria ter êxito para um agente, uma vez que não existe nenhuma medida fixa satisfatória. Este padrão deve ser usado para medir o desempenho dos agentes em seu ambiente, já que eles devem atingir suas metas da melhor forma possível [THO97, HAR96].

Alguns aspectos devem ser levados em consideração como, por exemplo, quando medir. O desempenho do agente deveria ser medido ao longo de todo seu tempo de atuação. Os agentes poderiam ser subjetivamente indagados sobre o desempenho, porém considerando que os agentes poderiam estar em alguns casos, impossibilitados de responder e em outros casos inclinados a se iludir, o padrão para a medida do desempenho deve ser objetivo e imposto com um pouco de autoridade.

### **3.12 Tendências**

Predizer qual será o papel dos agentes no futuro, e como eles serão construídos, não é uma tarefa fácil. Não só porque a tecnologia de agentes ainda está em fase de desenvolvimento, mas também porque a maioria dos fatores e grupos que estão envolvidos exercem influência uns sobre os outros. Entretanto, já existem várias aplicações baseadas em agentes que facilitam a vida dos usuários na Internet, proporcionando o rápido desenvolvimento de áreas como comércio eletrônico, ensino a distância, gerenciamento de sistemas, correio eletrônico, busca de informação na WWW, dentre outros [DAV95].

As universidades, centros de pesquisa e um número considerável de companhias, como a IBM e Microsoft, estão fazendo pesquisa na área de agentes. Para ter certeza que seus projetos de pesquisa receberão financiamento mais adiante, muitos pesquisadores e desenvolvedores destas companhias (mais isto também é aplicável para outros grupos, até mesmo alguns não comerciais) estão enfocando hoje em dia aplicações de agentes bastante básicas, como as que conduzem a resultados demonstrável dentro de um tempo definido. Exemplos deste tipo de aplicações [DAV95]:

- ◆ Agentes que parcialmente ou completamente automatizam correio eletrônico;
- ◆ Agentes que filtram e/ou procuram por novas informações que podem ser interessantes para seus usuários;

- ◆ Agentes que agendem e organizam reuniões por meio de listas fornecidas pelas pessoas envolvidas ou baseado na informação das agendas e compromissos de cada participante.

A seguir são identificadas algumas áreas de aplicação onde a tecnologia de agentes é (ou será) usada. Estas áreas são [DAV95]:

**Gerenciamento de Sistemas e Redes:** esta foi uma das primeiras áreas de aplicações a empregar a tecnologia de agentes inteligentes. O uso crescente de arquiteturas cliente-servidor elevou a complexidade dos sistemas em operação, principalmente em redes locais. As arquiteturas de agentes empregadas são em sua maioria não inteligentes, entretanto sistemas inteligentes encontrariam muita aplicação em níveis mais altos de abstração, por exemplo, aprendendo a reagir a determinados padrões no comportamento dos sistemas. Além disso, poderiam ser também empregados no gerenciamento dinâmico de grandes configurações.

**Acesso e Gerenciamento Móvel:** à medida que a computação vai se tornando cada vez mais distribuída e difusa, surge a necessidade dos usuários empregarem tecnologias móveis, tais como comunicação sem fio. Os usuários desejam não apenas conectar-se a partir de qualquer lugar e também não sofrer as restrições de largura de banda por vezes impostas pelas telecomunicações. A manipulação da transmissão da informação é uma área promissora para o emprego de agentes.

**Correio Eletrônico e Troca de Mensagens:** agentes vem sendo empregados nesta área já há algum tempo, priorizando mensagens e automaticamente organizando o correio eletrônico de seus usuários. Os agentes inteligentes podem facilitar todas essas funções, por exemplo, por meio de regras que poderiam ser inclusive deduzidas a partir de padrões de comportamento observados em seus usuários.

**Acesso e Gerenciamento da Informação:** esta é uma área de grande atividade, tendo em vista a rápida popularização da Internet e a exploração disponível a seus usuários. Aqui agentes inteligentes podem ser empregados não apenas na pesquisa e filtragem de

informação, mas também na categorização, priorização, disseminação seletiva, anotação e no compartilhamento cooperativo de documentos e informação.

**Colaboração:** é uma área em rápido crescimento onde os usuários trabalham juntos em documentos compartilhados na rede. Aqui é necessário não apenas uma infra-estrutura que permita o compartilhamento robusto e escalável de dados e outros recursos, mas também funções que permitem gerenciar equipes e o produto de seu trabalho. O exemplo mais conhecido de aplicações deste tipo é o Lotus Notes.

**Gerenciamento Administrativo:** inclui o gerenciamento de fluxos de trabalho e também áreas de suporte, como por exemplo, a integração entre computadores e serviços de telefonia, onde processos são definidos e então automatizados. Nestas áreas os usuários necessitam não somente tornar os processos mais eficientes, como também reduzir os custos dos humanos. Agentes inteligentes poderiam ser empregados aqui para identificar e automatizar processos de possível interesse do usuário.

**Comércio Eletrônico:** esta é uma área em acelerado crescimento, que é alimentado pela popularização da Internet. Os consumidores, em busca de produtos e serviços necessitam de informações sobre o que estão comprando (inclusive especificações técnicas, configurações viáveis, etc). Os comerciantes necessitam localizar (e atrair) clientes, oferecer suporte especializado sobre seus produtos, controlar e realizar o follow-up de suas vendas, etc.

Tanto os consumidores quanto os comerciantes necessitam automatizar sua participação neste “mercado eletrônico”. Agentes inteligentes poderiam ser empregados aqui de diversos modos. Por exemplo, poderiam “ir as compras” para seus usuários, coletar especificações de um determinado produto e retornar com sugestões de compras que atendessem descrições recebidas. Também poderiam atuar como assistentes de vendas, fornecendo aconselhamento sobre os produtos e tentando solucionar possíveis problemas e dificuldades da disseminação de interfaces gráficas, para muitas pessoas, os computadores continuam difíceis de usar. Por outro lado, o usuário.

**Interfaces Inteligentes:** Apesar da disseminação de interfaces, para muitas pessoas, os computadores continuam difíceis de usar. Por outro lado, à medida que a população de usuários cresce e se diversifica as interfaces se tornam mais e mais complexas para acomodar hábitos e preferências variadas. Agentes de interface inteligentes poderiam, por exemplo, monitorar as ações do usuário para desenvolver um modelo com suas habilidades e automaticamente ajudá-lo quando os problemas surgirem.

### 3.13 Plataformas de Agentes Móveis

Plataformas de agentes móveis em software suportam a criação de processos computacionais autônomos e assíncronos, capazes de execução e migração continuada entre computadores em rede. O aspecto que mais diferencia entre a migração de agentes e a transferência de código móvel é que os agentes transferem não apenas o código móvel é que os agentes transferem não apenas o código quando migrando entre hosts, mas também o estado da execução. Sendo assim, no contexto de agentes móveis, tende a desaparecer a diferenças entre cliente e servidores, e todos os equipamentos que suportam a execução de agentes são chamados de hosts. As tecnologias mais representativas da atualidade são baseadas na especificação da Mobile Agent Facility – MAF\OMG (GMD at alli, 96), suportada pelo consórcio Object Management Group – OMG[OMG].

As tecnologias têm uma ampla variação de características e existem pelo menos dois outros grupos que desenvolvem especificações divergentes de plataformas de agentes: Agent Society [AGSOC] e Foundation for Intelligent Physical Agents – FIPA[FIPA]. Dada a rapidez e fluidez com que a área se desenvolve é difícil se fazer uma descrição atualizada de quantas plataformas diferentes existem ou estão sendo criadas.

Uma das plataformas de referência mais difundidas, é Aglets Workbench [AGLETS], implementada pela IBM, e que serviu como base para especificação da MAF\OMG. Outra plataforma que adere ao MAF\OMG, é Odyssey, desenvolvida pela General Magic [GENMAGIC] uma das pioneiras no desenvolvimento de plataformas de

agentes com a já descontinuada plataforma Telescript. A seguir uma descrição mais detalhada de algumas destas plataformas:

- **Odyssey** - A General Magic Inc. criou o primeiro sistema de agentes móveis comercial, chamado Telescript. Entretanto, o Telescript teve pouco sucesso, pois era baseado em uma linguagem e arquitetura de rede proprietárias. A popularidade da Internet motivou a General Magic a reimplementar um sistema de agentes móveis chamado Odyssey, baseado em Java. Este sistema simplesmente mapeou os conceitos do Telescript em classes Java, permitindo os desenvolvedores a criar suas próprias aplicações.
- **Concórdia** - Concebido pela Mitsubishi constitui-se de um *framework* para o desenvolvimento e gerenciamento de aplicações de agentes móveis. O Concórdia compreende múltiplos componentes, todos escritos em Java, na qual são combinados para prover um ambiente para aplicações distribuídas. Este sistema é simples, e requer somente uma implementação padrão da máquina virtual Java, e seu ambiente é composto de um servidor e um conjunto de agentes. O Concórdia provê mecanismos de segurança para execução segura de agentes, além de suportar mecanismos de *checkpoint* para tolerância a falhas.
- **Voyager** - Criado pela ObjectSpace, consiste de uma plataforma ORB (*Object Request Broker*) implementado em Java e com suporte a agentes. Voyager implementa os mecanismos tradicionais de troca de mensagens, somados à capacidade de objetos moverem-se através da rede como agentes. Uma desvantagem de Voyager é não oferecer mecanismos de segurança contra agentes não autorizados.

### 3.14 Trabalhos Relacionados

Há forte expectativa de que os agentes de informação sejam uma área de grande desenvolvimento nos próximos anos. Davies e Weeks [DAV95] projetaram e implementaram no BT Labs o agente Jasper (Joint Access to Stored Pages with Easy Retrieval). Os agentes Jasper trabalham para um usuário, grupo ou comunidade de

usuários, sendo capaz de armazenar, recuperar, sumarizar e informar outros agentes sobre informação útil encontrada na WWW. À medida que um usuário interage com seu agente Jasper, um perfil de seus interesses é construído dinamicamente e representado por palavras-chave. Um agente Jasper é capaz de sugerir páginas WWW interessantes a seus usuários, com base no seu perfil e no perfil do grupo ou comunidade em que se encontram inseridos.

Diversos outros agentes de informações foram construídos, em particular para a filtragem de informações, como por exemplo, o Webwatcher [AMS95], RBSE Spider [EIC94] e InfoAgent [GIA95]. Segundo [NWA96] o principal problema encontrado em agentes de informações estáticos é manter seus índices atualizados em um ambiente de elevada dinamicidade. Alguns pesquisadores [EIC94] [ETZ94] levantam também algumas questões relacionadas com a ética de agentes de informação. É provável que a próxima geração de tais agentes venha a empregar em grande escala a propriedade da mobilidade. Serão capazes de navegar na WWW e armazenar sua topologia em uma base de dados, por exemplo, em seu local de origem. Observa-se ainda que, se o agente de informação é do tipo estático, encontra-se mais próximo dos agentes de interface do que os agentes móveis, compartilhando com esses problemas similares de Modelagem e construção.

## CAPITULO 4

### IMPLEMENTAÇÃO DE WORKFLOW USANDO O PARADIGMA DOS AGENTES

Este capítulo tem o objetivo de descrever a arquitetura, instalação, configuração e forma de execução do protótipo implementado neste trabalho. Este sistema foi desenvolvido, utilizando-se a linguagem de programação JAVA.

#### 4.1 Estrutura do Workflow Implementado

O sistema proposto foi definido contendo o seguinte fluxo dentro do workflow: Processo Digitação dos Pedidos, Processo Análise do Estoque e Análise de Crédito e Consolidação do Pedido. O modelo é mostrado na Fig. 1.2:

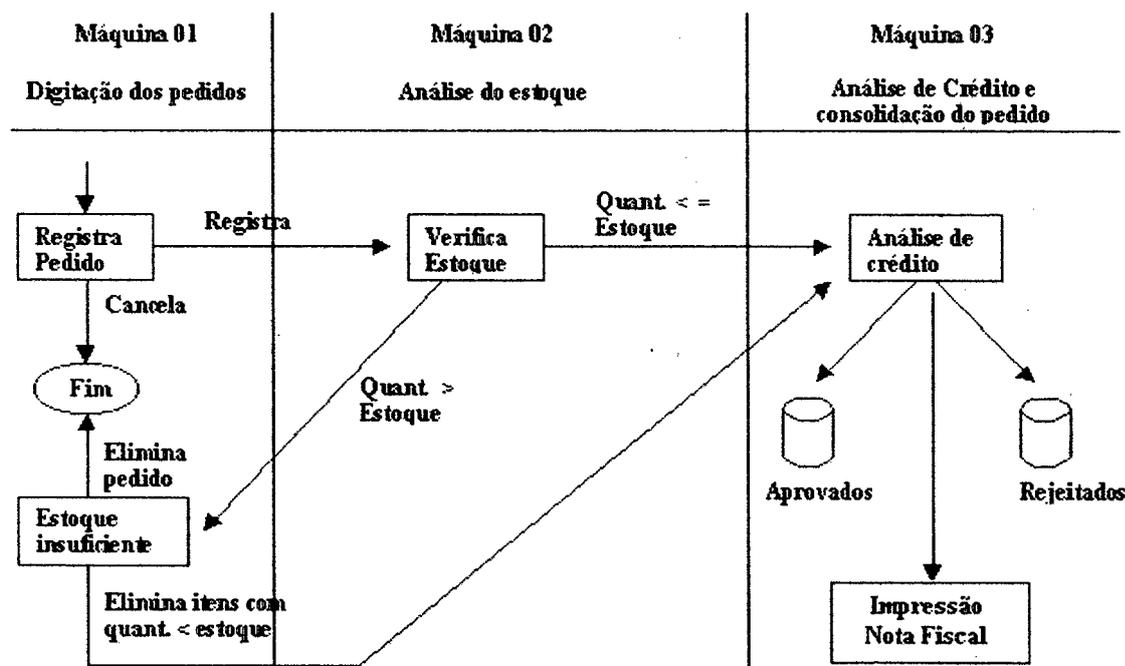


Figura 1.2 – Representação do modelo de Workflow implementado.

## 4.2 Modelo de dados - UML

O sistema proposto foi definido contendo as seguintes entidades, mostradas na Fig. 1.3:

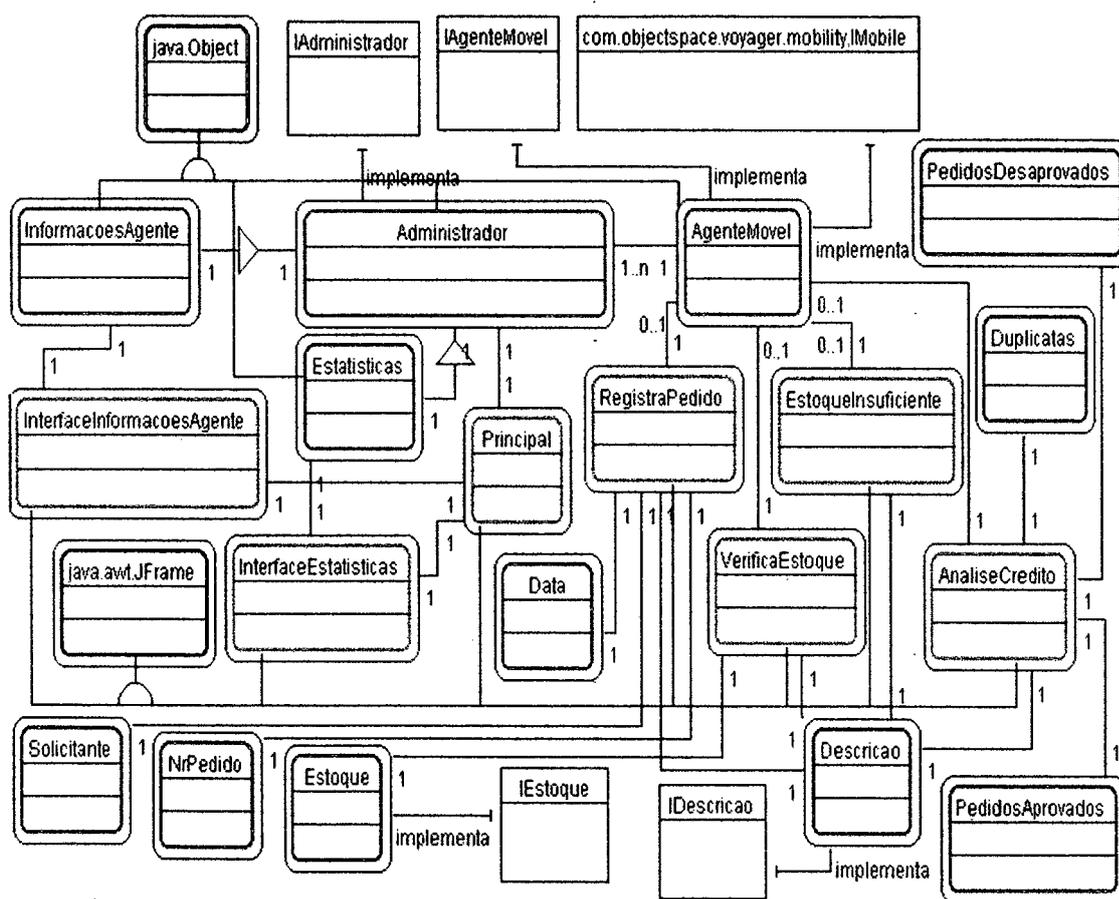


Figura 1.3 – Modelo lógico de dados do sistema proposto.

## 4.3 Arquitetura do protótipo

A arquitetura do protótipo é composta basicamente por um conjunto de três (3) máquinas interligadas através de um *Object Request Broker* (ORB), como ilustrado na fig. 1.4:

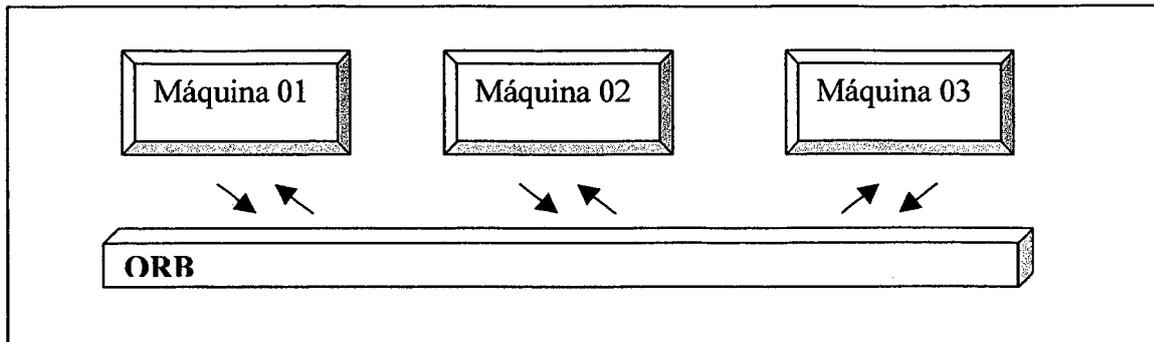


Figura 1.4: Arquitetura do protótipo.

O ORB utilizado foi o Voyager versão 3.3 e a linguagem de programação utilizada foi o Java, versão 1.2.

A versão 1.2 da linguagem Java foi escolhida pelo fato de ser pré-requisito para a execução do Voyager 3.3.

#### 4.4 Instalação do ambiente

Para que a execução do protótipo ocorra de forma correta. Ao instalar o mesmo, deve-se adotar a seguinte estrutura em todas as máquinas pertencentes à arquitetura:

- Deve-se criar um diretório chamado `voyager` logo após o diretório raiz de cada máquina, e no mesmo diretório deve-se instalar o ORB Voyager;
- Deve-se criar um diretório chamado `jdk1.2` logo após o diretório raiz de cada máquina, e no mesmo diretório deve-se instalar o Java Develop Kit 1.2 (JDK 1.2), e;
- Para o protótipo deve-se criar um diretório chamado `WorkFlow`, e neste diretório deve-se instalar o protótipo, ou seja, copiar todos os arquivos pertencentes a package `WorkFlow` para este diretório.

Depois de executado estes passos deve-se configurar o ambiente.

## 4.5 Configuração do ambiente

Após a instalação do Voyager e do JDK em todas as máquinas pertencentes a arquitetura, é necessário editar o arquivo `autoexec.bat` para que cada máquina reconheça o ORB, a linguagem Java e o local em que se encontra o protótipo. Na fig. 1.5 pode-se visualizar um fragmento do arquivo `autoexec.bat` [SUN97].

```
:  
:  
:  
set PATH=c:\jdk1.2.2\bin;c:\voyager\bin;%path%;  
set CLASSPATH=c:\voyager\lib\voyager.jar;  
:  
:  
:
```

Figura 1.5: Fragmento do arquivo `autoexec.bat`

A variável de ambiente `PATH`, tanto para o Voyager, como para o JDK, deve ser setada ao diretório dos executáveis, ou seja, ao diretório `bin`.

A variável de ambiente `CLASSPATH` deve ser setada para o arquivo `voyager.jar`, a fim de informar onde estão as classes que fazem parte da API do Voyager.

## 4.6 Execução do protótipo

Com a instalação e configuração terminadas, pode-se então dar início a execução do protótipo.

Na máquina 01 (ver fig. 1.1), máquina que dá início ao processo de `WorkFlow`, deve-se digitar na linha de comando, no diretório raiz, o seguinte comando:

```
java WorkFlow Principal <endereco_01> <endereco_02> <endereco_03>
```

Isto irá por em execução a classe *Principal* que recebe como parâmetros o endereço da máquina 01, máquina 02 e máquina 03.

Nas outras máquinas, também no diretório raiz, deve-se iniciar o ORB Voyager com a seguinte linha de comando:

*Voyager* <endereço\_02>, na máquina 02;

*Voyager* <endereço\_03>, na máquina 03.

## 4.7 Implementação do Sistema

Este capítulo tem por objetivo apresentar o protótipo desenvolvido para validar os conceitos vistos anteriormente. O sistema é composto de três módulos: Informar Pedidos, Informações do Pedido e Informações do Workflow. A interface principal do sistema é mostrada na fig. 1.6.

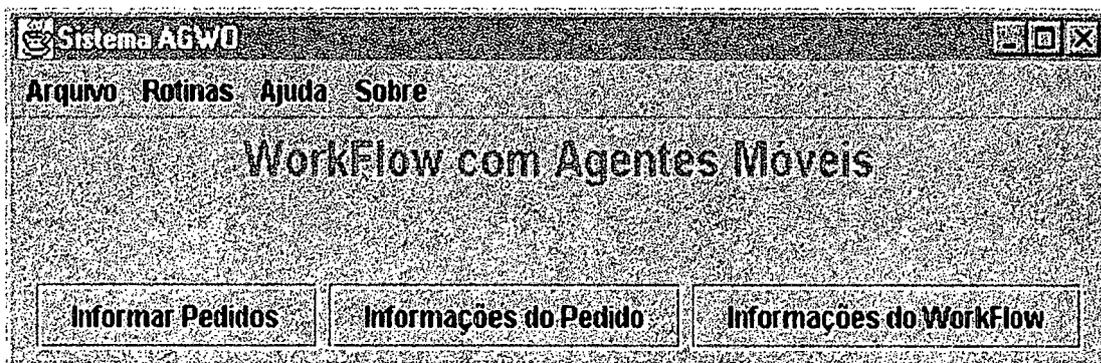


Figura 1.6 - Interface principal do Sistema.

A interface Informar Pedidos permite a inclusão dos dados relativos aos pedidos conforme é visto na fig. 1.7.

Cod.	Descrição	Quant.
1	Item 1	10
5	Item 5	10

Figura 1.7: Interface de entrada de dados dos pedidos.

A interface verifica estoque faz a checagem do estoque de cada item do pedido e permite que o operador valide as quantidades mostradas. A fig. 1.8 mostra estas opções.

Cod.	Descrição	Quant.	Estoque	
1	Item 1	10	100	<input checked="" type="checkbox"/>
5	Item 5	10	35	<input checked="" type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>

**Confirmar**

Figura 1.8: Interface de checagem de estoque.

A interface de análise de crédito faz a checagem de pendências de duplicatas do cliente e permite ao operador aprovar ou desaprovar o pedido. A interface de análise de crédito é vista na fig. 1.9.

Cod.	Descrição	Quant.
1	Item 1	10
5	Item 5	10

Duplicatas pendentes: 0

**Aprova**                      **Desaprova**

Figura 1.9: Interface de checagem de duplicatas pendentes.

A interface de estoque insuficiente mostra os itens do pedido com a quantidade solicitada, caso o item esteja com quantidade em estoque insuficiente o check box estará desmarcado. Nesta interface o operador pode cancelar os itens com estoque insuficiente ou cancelar o pedido. A interface de estoque insuficiente é vista na Fig. 2.0.

Cod.	Descrição	Quant	
5	Item 5	100	<input checked="" type="checkbox"/>
7	Item 7	10	<input type="checkbox"/>

Figura 2.0: Interface de cancelamento de pedido/itens.

A tela de consulta de informações do workflow permite ao operador do sistema consultar informações referentes aos pedidos que passaram ou estão passando pelo workflow. A tela de consulta de informações do workflow é vista na fig. 2.1.

Pedidos Processados:	0
Pedidos em andamento:	2

Figura 2.1: Interface de informações do workflow.

A tela de consulta informações do pedido, permite ao operador do sistema acompanhar o andamento dos pedidos pelo workflow (ver fig. 2.2).

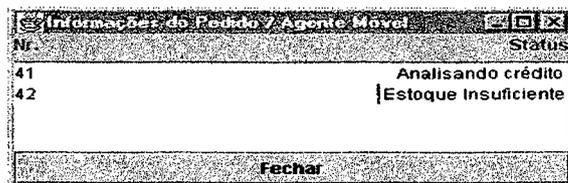


Figura 2.2: Interface de acompanhamento dos pedidos.

## 4.8 Fontes do Sistema

Neste capítulo descrevemos as principais partes do código fonte do protótipo implementado. A fig. 2.3 mostra o código fonte do programa principal.

```

Public class Principal extends JFrame {

    public Principal (String maquina_01, String maquina_02, String
                    maquina_03){
        super();
        try
        {
            init();
            iniciaServidor(maquina_01, maquina_02, maquina_03);
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
    private JButton bPedido = new JButton("Informar Pedidos");
    private JButton bInforma = new JButton("Informações do Pedido");
    private JButton bEstatistica = new JButton("Informações do Workflow");
    private JLabel lPrincipal = new JLabel("Workflow com Agentes Móveis");

    public static void main (String args[]){
        if (args.length == 3)
            new Principal(args[0],args[1],args[2]);
        else
        {
            System.out.println("Erro! ");
            System.out.println("Deve-se informar o endereço das maquinas
                                Pertencentes ao Workflow ");
            System.out.println("Sintaxe: ");
            System.out.println("java Workflow.Principal <maq_01> <maq_02>
                                maq_03> ");
            System.out.println("Exemplo: ");
            System.out.println("java Workflow.Principal
                                //localhost:7000 //localhost:8000 //localhost:9000 ");
        }
    }
}

```

Figura 2.3 - Classe principal do protótipo (Principal.class)

Apresentamos a seguir parte do código da classe Administrador, que é responsável pelo gerenciamento do Workflow de modo geral, veja fig. 2.4:

```
public class Administrador implements Iadministrador
{
    private static Administrador administrador_;
    private IAgenteMovel agente;
    private IMobility mobility;
    private String maquina_01, maquina_02, maquina_03;
    private int id;
    private InformacoesAgente informa;
    private Estatisticas estatisticas;
    private NrPedido nrPedido = new NrPedido();

    public static Administrador getAdministrador(){
    return administrador_;
    }
}
```

Figura 2.4 - Classe Administrador (Administrador.class)

A seguir mostramos parte do código-fonte da classe Agente móvel, responsável pela mobilidade dos agentes dentro do Workflow, ver fig. 2.5.

```
public class AgenteMovel implements IAgenteMovel, IMobile,
Serializable{

    private IMobility mobility;

    private int id;
    private String nrPedido;
    private String solicitante;
    private String data;
    private Vector cod_prod;
    private Vector qtd_prod;
    private Vector item_ok;
    private String status = "0";

    private VerificaEstoque verificaEstoque;
    private EstoqueInsuficiente infEstoque;
    private AnaliseCredito credito;

    private Iadministrador adm;
    private IEstoque estoque;

    private String localAdm;

    public void recebeEnderecoAdm (String local){
        this.localAdm = local;
    }
}
```

```

public void recebeId (int id){
    this.id = id;
}

```

Figura 2.5- classe Agente Móvel (AgenteMovel.class)

Apresentamos a seguir parte do código-fonte da classe informações do agente, responsável pelas informações dos agentes dentro do workflow, ver fig. 2.6.

```

public class InformacoesAgente {

    private Vector id;
    private Vector nr;
    private Vector status;

    private static InformacoesAgente informacoes_;

    public static InformacoesAgente getInformacoesAgente(){
        return informacoes_;
    }

    public InformacoesAgente (){
        id = new Vector();
        nr = new Vector();
        status = new Vector();
        informacoes_ = this;
    }

    public void colocaValores (int p_id, String p_nr, String p_status)
    {
        id.addElement(new Integer(p_id).toString());
        nr.addElement(p_nr);
        status.addElement(p_status);
    }

    public void alteraNr (int p_id, String p_nr){
        for(int i=0;i<id.size();i++)
            if(id.elementAt(i).equals(new Integer(p_id).toString()))
                nr.set(i,p_nr);
    }

    public void alteraStatus (int p_id, String p_status){
        for(int i=0;i<id.size();i++)
            if(id.elementAt(i).equals(new Integer(p_id).toString()))
                status.set(i,p_status);
    }

    public Vector pegaId(){
        return id;
    }

    public Vector pegaNr(){
        return nr;
    }

    public Vector pegaStatus(){
        return status;
    }

}

```

Figura 2.6 – Classe de Informações do Agente (InformacoesAgente.class)

## **CAPÍTULO 5**

### **FERRAMENTAS EXISTENTES NO MERCADO**

Hoje em dia, processos de negócios processam e direcionam vários tipos de informação para várias pessoas ao mesmo tempo de acordo com regras dinâmicas de negócios. O workflow da Oracle processa automaticamente e direciona qualquer tipo de informação, de acordo com as regras de negócio as quais você pode facilmente alterar, para qualquer pessoa dentro ou fora de sua empresa. O workflow da Oracle faz parte das Aplicações Oracle, um conjunto integrado de soluções para negócios.

Com tanta informação disponível e em tantas formas diferentes, como conseguir que a informação certa chegue a pessoa certa no tempo certo? O workflow da Oracle permite que você forneça, a cada pessoa envolvida num processo ativo de negócios workflow, toda a informação de que eles precisam para formular a resposta apropriada. A versão atual do workflow permite que você capte informações de todos os tipos de mídia, incluindo documentos texto, imagens, áudio e vídeo, e as associe com informações workflow em aplicativos Oracle. O workflow Oracle pode direcionar essas informações suporte para cada tomador de decisões em seu workflow.

A habilidade de adaptar rapidamente seus processos de negócio a um ambiente rapidamente mutável é vital. O workflow Oracle permite que você modele e mantenha os seus processos de negócios, utilizando um construtor gráfico de workflow. Diferentemente de sistemas workflow que simplesmente direcionam documentos de um usuário para outro com algumas etapas de aprovação, O workflow Oracle permite que você modele e automatize sofisticados processos de negócios. Você pode definir processos que se repetem e se dividem para fluxos paralelos e então se encontrem no final do workflow. Com o Workflow Oracle você pode usar todo o poder da PL/SQL, a linguagem do servidor Oracle, para processar automaticamente informações de acordo com as mais sofisticadas regras de negócio.

Seus processos de negócio não incluem somente usuários de aplicativos Oracle, mas também outros usuários dentro de sua empresa e até mesmo clientes, fornecedores, parceiros de negócios de fora de sua empresa. O workflow Oracle estende o alcance da automação de processos de negócios por toda a empresa e adiante para se incluir qualquer e-mail ou usuário de internet. O workflow Oracle permite que as pessoas recebam, analisem, e respondam a notificações de itens no aguardo de sua atenção via aplicativos Oracle ou qualquer sistema padrão de e-mail ou navegador.

## **5.1 Componentes da ferramenta Workflow Builder 2.0**

### **5.1.1 Construtor de Workflow**

O construtor workflow gráfico do Workflow da Oracle permite que você crie, visualize ou modifique um processo de negócio com simples operações de arrastar e soltar. Usando o construtor, você pode criar e modificar objetos de workflow reutilizáveis, incluindo atividades, regras, mensagens e usuários. A qualquer momento você pode adicionar, remover, ou mudar atividades do workflow, ou definir novas relações de pré-requisitos entre as atividades. Você pode facilmente trabalhar com um modelo sumário do seu workflow, expandindo atividades dentro do workflow a medida que maiores níveis de detalhes forem sendo necessários. E, Você pode operar o construtor de um PC de mesa ou até de um laptop não conectado. O workflow da Oracle vem completo com uma biblioteca de atividades predefinidas para aplicativos Oracle e organizadores de workflow que você pode facilmente moldar às suas necessidades em específico ou incluir em qualquer um dos seus próprios workflows.

### **5.1.2 Mecanismo de workflow baseado em regras**

Um mecanismo de workflow presente no servidor Oracle monitora os estados do workflow e coordena o direcionamento de atividades para um processo. Mudanças no

estado do workflow, tais como o termino de atividades do workflow, são transmitidas ao mecanismo através de uma API PL/SQL. Baseado num workflow de regras flexivelmente definidas, o mecanismo determina quais atividades devem ser processadas e então as processa. Ele suporta sofisticadas regras de workflow, incluindo looping, rearranjo, fluxos paralelos e subfluxos.

### **5.1.3 Dimensionamento Completo**

O workflow da Oracle permite que você defina e se refira a suas próprias atividades no seu workflow. Sem modificar nenhum código de aplicativos do Oracle, você pode ter sua própria atividade executada sempre que o workflow do Oracle detectar que os pré-requisitos da sua atividade estiverem preenchidos. Essa poderosa capacidade permite que você adicione qualquer número de processamentos de negócios adicionais e atividades de notificação a qualquer workflow.

### **5.1.4 Notificações eletrônicas**

O workflow Oracle permite que você notifique usuários em seus workflows para dar conta de atividades que não possam ser automatizadas, tais como aprovações para requisições ou pedidos de venda. Notificações eletrônicas podem ser direcionadas para usuários individuais ou para classes de usuários. Cada notificação contém todas as informações de suporte que o usuário precisa para tomar a decisão, e permite que eles escolham uma de uma seleção de respostas apropriadas. O workflow Oracle interpreta cada resposta de notificação e então continua a processar as atividades do workflow até que ou a resposta de um usuário seja necessária ou então que o workflow esteja completo.

Uma notificação direcionada a um grupo pode ser atuada por qualquer um dos usuários com esta função. Após resposta de qualquer um dos membros do workflow, o processo do workflow continua até a próxima atividade. Os usuários do workflow da Oracle podem revisar, analisar, e responder as notificações dele em até três dias. Usuários regulares de aplicativos Oracle podem usar um formulário de notificações

comum de dentro de aplicativos Oracle. Para estas pessoas, as quais não usam regularmente aplicativos Oracle, ou àquelas que não são de fácil acesso, o Workflow da Oracle pode também mandar notificações através de qualquer sistema de e-mail ou navegador da internet.

#### **5.1.5 Caixa pessoal de entrada**

Usuários regulares de aplicativos Oracle podem rever, analisar, e responder a comunicados numa forma de comunicação bem comum, ou Caixa de Entrada. Usuários podem rever informações de auxílio, especificar uma resposta apropriada, ou mandar em frente para outro usuário a notificação. As informações de suporte e respostas apropriadas são dependentes do contexto, sendo este determinado pela natureza da notificação.

#### **5.1.6 Integração via e-mail**

Pessoas que não usam regularmente aplicativos Oracle, ou que não são de fácil acesso, podem rever notificações usando qualquer aplicativo de e-mail. Uma notificação via e-mail pode conter um anexo HTML que abre um navegador padrão através do qual o usuário pode então rever as informações suporte e responder a essa notificação.

#### **5.1.7 Um workflow habilitado à internet**

Qualquer usuário com acesso a internet pode também ser incluído em um workflow. Usuários da internet podem acessar uma página pré-configurada da rede para ver as suas notificações surpreendentes, então navegar em páginas adicionais, dinamicamente geradas, a fim de rever informações de suporte ou a fim de responder à notificação.

### **5.1.8 Facilidade de habilitar aplicativos no workflow**

O Workflow da Oracle (Oracle Workflow) é a principal tecnologia usada para habilitar aplicativos Oracle num workflow. A arquitetura flexível do Oracle Workflow torna fácil habilitar num workflow qualquer aplicativo baseado em Oracle. Utilizando simples chamadas PL/SQL, um aplicativo consegue informar o Workflow Oracle de que atividades pré-definidas foram concluídas. E o Workflow Oracle consegue chamar qualquer função aplicativa PL/SQL, de atividades em um processo de workflow.

### **5.2 Necessidades de software e hardware**

O mecanismo do Workflow Oracle roda em toda plataforma suportada por Aplicativos Oracle. O construtor de workflows gráficos requer Windows 95 ou NT. O formato de aplicativos Oracle de notificações comuns requer uma configuração smartclient de aplicativos Oracle. O componente de notificações via e-mail requer o Oracle Office, O UNIX sendmail, ou um aplicativo de e-mails subordinado a MAPI e uma plataforma que suporte um desses aplicativos de e-mail; Suporte à anexos HTML requer um aplicativo de e-mail que suporte anexos. As notificações de rede e componentes de monitoração requerem um Servidor de Rede Oracle e um navegador que suporte Javascript, Frames e Java.

### **5.3 Características básicas**

#### **5.3.1 Construtor de workflow gráfico**

- Pode-se adicionar, editar e ver definições dos processos de negócios;
- Abrir menus de propriedades
- Layout da tela configurável pelo usuário
- Definições de atividades e processos do workflow reutilizáveis
- Suporte a subprocessos
- Pode-se mover definições de processos entre a central de dados e um novo arquivo

- Suporte para versão e data efetiva para definições de processo
- Suporte multi-idiomas

### **5.3.2 Mecanismo do workflow**

- Um mecanismo baseado em servidor implementa definições de processo em tempo real
- Presente no Servidor Oracle 7 e versões seguintes
- Roda funções PL/SQL como se fossem atividades
- Suporta atividades manuais e automáticas
- Processamento de timeout
- Suporta situações complexas de particionamento e agrupamento
- Condições de erro ativam workflows que lidam com o erro
- Suporte para cancelar, suspender ou pausar um processo
- Delega atividades pesadas para mecanismos secundários a fim de otimizar o tempo de resposta rápido.
- API simples para sinalizar mudanças no estado da aplicação
- API simples para suporte de dados de aplicativos
- Janela de processos pré-definidos
- Atividades pré-definidos do workflow
- Suporte multi-idiomas

### **5.3.3 Notificações eletrônicas**

- Notificações de assuntos e respostas a processos de usuários
- Notificações via caixa de entrada do Oracle, e-mail ou rede.
- A Conexão da Caixa de entrada do Oracle com aplicativos Oracle forma notificações via e-mail com anexos HTML
- Manda notificações para usuários individuais ou para as partes responsáveis pelas atividades

- Formatos de notificação por e-mail, baseados em aspectos definidos pelos clientes;
- Possibilidade de passar em frente notificações
- Anexos multimídia
- Suporte multi-idiomas

#### **5.3.4 Monitoramento gráfico e análise de desempenho**

- Tela gráfica baseada em rede para mostrar e administrar processos e atividades
- Impressos e relatórios padrão para monitorar processos e atividades
- Cria relatórios padrão nos histogramas

#### **5.3.5 Arquitetura aberta e dimensionável**

- Usa APIs abertas para extração e importação de dados
- Pode utilizar fontes de dados alheias ao Oracle, tais como sistemas Legados.

### **5.4 Implementação do protótipo utilizando a ferramenta Workflow Builer 2.0**

Para validar os conceitos da ferramenta *Workflow Builer 2.0* desenvolvemos a mesma aplicação utilizada no protótipo com agentes. A aplicação foi escrita utilizando a linguagem PL/SQL, linguagem esta que gera código HTML. A aplicação tem início com a interface principal como é mostrado na fig. 2.7:

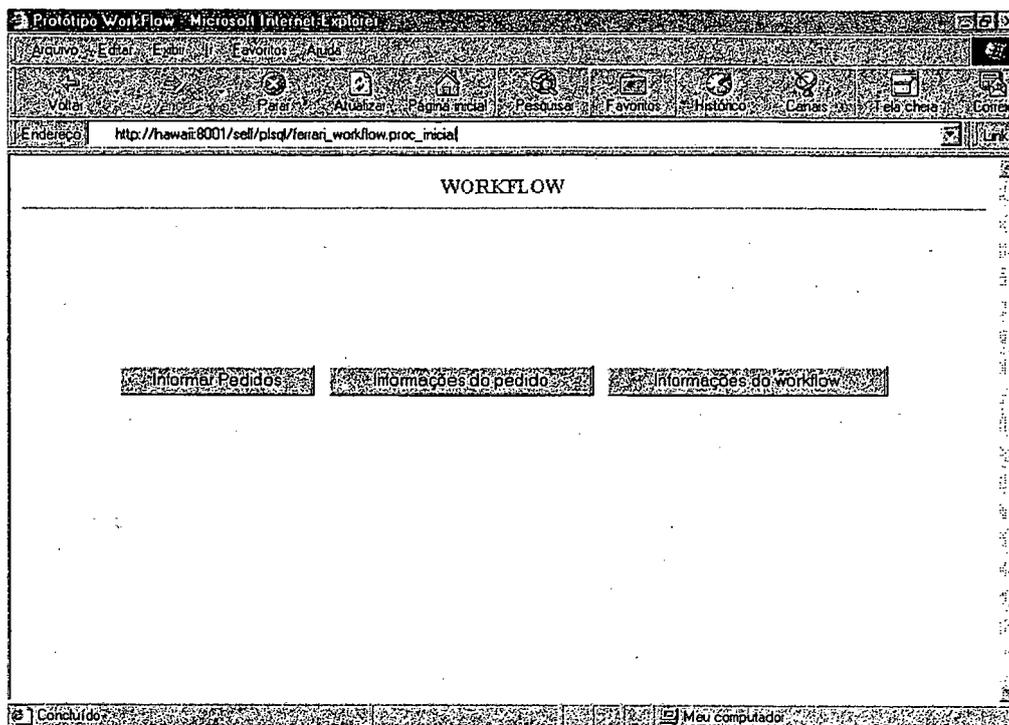


Figura 2.7: Interface inicial do workflow.

A interface Informar Pedidos permite a inclusão dos dados relativos aos pedidos conforme é visto na fig. 2.8.

Nr.:  Data:

Solicitante:

Cod	Descrição	Quant
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

Figura 2.8: Interface de entrada de dados dos pedidos.

A interface verifica estoque faz a checagem do estoque de cada item do pedido e permite que o operador valide as quantidades mostradas. A fig. 2.9 mostra estas opções.

The screenshot shows a web browser window titled "Prototipo WorkFlow - Microsoft Internet Explorer". The address bar contains the URL "http://hawaii8001/self/plsql/terrai\_workflow.estoque\_in". The main content area is titled "CADASTRO PEDIDO". It features a form with the following elements:

- Input field for "Nr.:"
- Input field for "Data:"
- Input field for "Solicitante:"
- A table with the following structure:

Cod	Descrição	Quant	Estoque

At the bottom of the form, there are two buttons: "Registra" and "Cancela". The browser's status bar at the bottom shows "Concluído" and "Meu computador".

Figura 2.9: Interface de checagem de estoque.

A interface de análise de crédito faz a checagem de pendências de duplicatas do cliente e permite ao operador aprovar ou desaprovar o pedido. A interface de análise de crédito é vista na fig. 3.0.

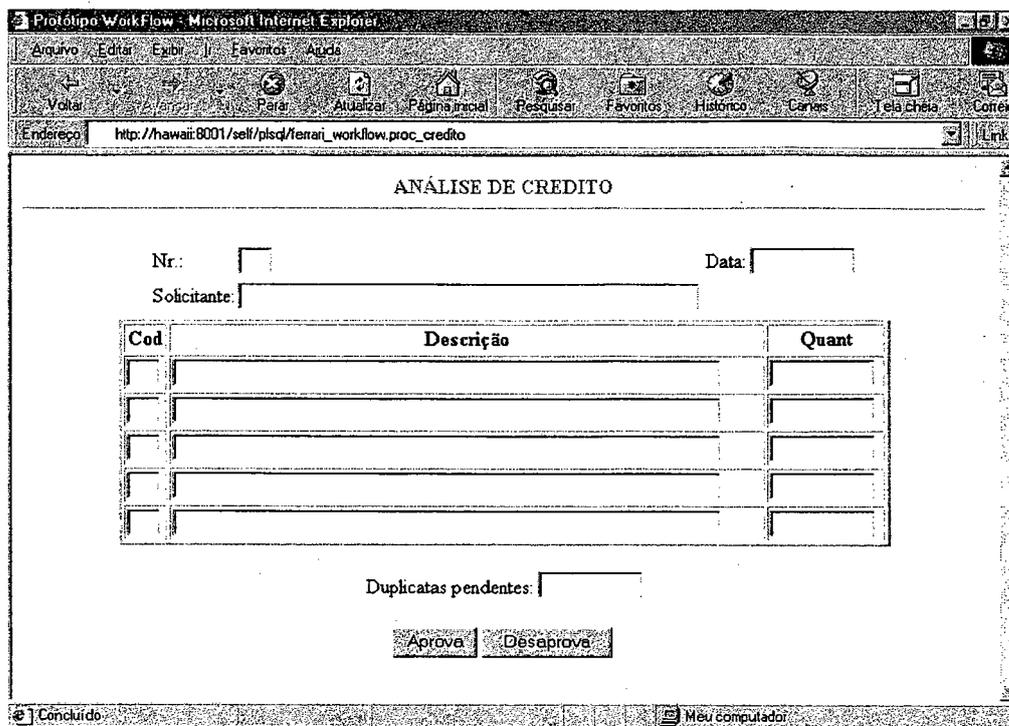


Figura 3.0: Interface de checagem de duplicatas pendentes.

A interface de estoque insuficiente mostra os itens do pedido com a quantidade solicitada, caso o item esteja com quantidade em estoque insuficiente o check box estará desmarcado. Nesta interface o operador pode cancelar os itens com estoque insuficiente ou cancelar o pedido. A interface de estoque insuficiente é vista na Fig. 3.1.

**CADASTRO PEDIDO**

Nr.:  Data:

Solicitante:

Cod	Descrição	Quant

Figura 3.1: Interface de cancelamento de pedido/itens.

A tela de consulta de informações do workflow permite ao operador do sistema consultar informações referentes aos pedidos que passaram ou estão passando pelo workflow. A tela de consulta de informações do workflow é vista na fig. 3.2.

**INFORMAÇÕES DO WORKFLOW / ESTATÍSTICA**

Pedidos Processados:

Pedidos em andamento:

Figura 3.2: Interface de informações do workflow.

A tela de consulta informações do pedido, permite ao operador do sistema acompanhar o andamento dos pedidos pelo workflow, veja fig. 3.3.

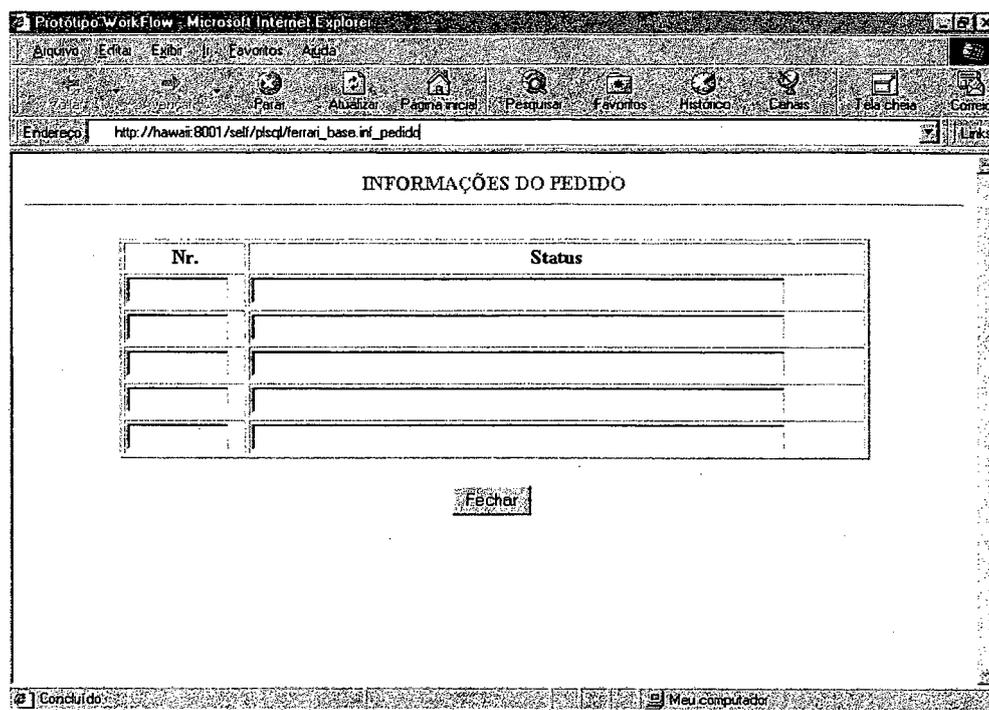


Figura 3.3: Interface de acompanhamento dos pedidos dentro workflow.

## CAPÍTULO 6

### UM COMPARATIVO ENTRE PROTÓTIPO IMPLEMENTADO E WORKFLOW DESENVOLVIDO COM FERRAMENTA EXISTENTE NO MERCADO

Para validar o protótipo do Workflow implementado e fazer um comparativo foi desenvolvido o mesmo Workflow em uma ferramenta existente no mercado. Como apresentado no capítulo 4 o exemplo desenvolvido utilizou a ferramenta Workflow da Oracle (*Workflow Builder 2.0*), o sistema tem as mesmas interfaces do anterior, mesmas funções mesmos módulos, porém implementado em linguagem de programação diferente (PL/SQL-ORACLE).

#### 6.1 Conclusões:

Desenvolvemos duas aplicações com ferramentas diferentes obtivemos os seguintes resultados:

a) O protótipo implementado utilizando a ferramenta *Oracle Workflow Builder 2.0* da Oracle mostrou as seguintes características, visto que o mesmo utiliza todas as facilidades e conceitos de banco de dados relacionais, para armazenar e recuperar as informações do workflow:

- Privacidade dos dados: São controles que devem ser estabelecidos para que os dados do workflow possam ser acessados somente por pessoas autorizadas.
- Dependência de plataforma Oracle: A ferramenta *Oracle Workflow Builder 2.0* roda somente em ambiente Oracle (banco de dados).
- Segurança de dados: O conceito de segurança abrange conceitos como procedimentos de validação e controle, garantia de integridade e controle de acesso, que visam resguardar as informações de uma possível perda ou

destruição de dados por falha de programa que compõe o workflow ou de equipamento.

- Independência de dados: Consiste na capacidade de permitir que haja evolução na descrição de dados da empresa, como a criação de uma nova estrutura lógica decorrente de uma nova aplicação, ou inclusão de um dado novo numa estrutura existente, sem que os sistemas ou aplicação que compõe o workflow tenham de ser alteradas.
- Garantia de integração dos dados: Consiste na capacidade de evitar que aplicações ou usuários concorrentes realizem atualizações sobre os dados, tornando-os inconsistentes. Essa capacidade traduz-se na introdução de recursos no sistema que inibam o acesso a um dado por uma aplicação quando esse mesmo dado já tiver sido lido para modificação por outra aplicação.
- *Oracle Workflow Builder 2.0*: Possui ferramentas para modelagem, criação e gerenciamento do workflow.

b) O protótipo implementado utilizando agentes móveis mostrou as seguintes características:

- O desempenho de processamento com agentes é alto, reduzindo assim o volume de transmissão na rede e incrementando a execução. No voyager, a funcionalidade se equívale, mas realizar chamadas de métodos remotos repetidos no servidor de objetos não é possível, porque os objetos voyager não podem mover-se através da rede em tempo real.
- A operação dos agentes independe da aplicação a qual o agente foi invocado. Os agentes operam assincronamente, evitando que a aplicação cliente aguarde para obter os resultados. Isso é importantíssimo para usuários móveis que não estão sempre conectados a rede.
- Uso de agentes permite melhorar a nova funcionalidade dentro dos sistemas para execução em tempo real. O sistema de agentes contém mecanismos de distribuição de software automáticos.

- Encapsulamento de protocolo, cada máquina em um sistema distribuído possui seu próprio código necessário para implementar a transferência de dados. Porém, novos requisitos de segurança e eficiência demandam mudanças no protocolo que podem ocasionar problemas na manutenção do código existente. Agentes móveis, por outro lado, podem mover-se para máquinas remotas a fim de estabelecer canais de comunicação baseados em protocolos proprietários.
- Independência de plataforma, redes de computadores, geralmente são heterogêneas, tanto na perspectiva de hardware como a de software. Agentes móveis são independentes da máquina e também da rede, sendo dependentes somente do seu ambiente de execução, não dificultando a integração de sistemas.

## CAPÍTULO 7

### 7. Conclusão

A tecnologia de workflow foi criada visando minimizar problema de coordenação de trabalho em grupo disponibilizando a modelagem e o suporte computacional para o processo de negócio. O workflow surgiu como um novo paradigma que auxilia no controle da automação de processos.

O workflow está intimamente relacionado com o CSCW e groupware. Pode-se entender o workflow como pertencente à área do CSCW, uma vez que ambos tem em comum a idéia do processo de grupo que corresponde ao processo de negócio. O workflow é classificado como um tipo de groupware, pois corresponde a uma ferramenta que auxilia o trabalho em grupo, uma vez que gerencia o fluxo de trabalho entre os membros de um grupo com um objetivo comum.

Os trabalhos de grupo são de natureza cooperativa, pois necessitam da cooperação dos participantes na sua realização. Entretanto os sistemas de workflow existentes focalizam mais a coordenação sem levar tanto em consideração o comportamento cooperativo adotado pelos agentes que executam as tarefas.

A utilização de agentes móveis no desenvolvimento do workflow mostrou-se viável, pois os agentes móveis independem da aplicação a qual o agente foi invocado. Os agentes operam assincronamente, evitando que a aplicação cliente aguarde para obter os resultados. Isso é importantíssimo para usuários móveis que não estão sempre conectado a rede.

Outra vantagem da utilização de agentes móveis no desenvolvimento do workflow foi a de que a arquitetura de agentes pode resolver problemas criados por intermitência ou má qualidade da conexão com a rede. Atualmente, algumas aplicações na rede, como por exemplo, Workflow são pesadas para completar a transação ou obter localização de informação. Se a conexão cair, o cliente deve freqüentemente reiniciar a transação do

ponto de partida. Com a tecnologia de agentes o cliente poderá obter as informações, mesmo que a conexão não esteja ativa. O cliente pode trabalhar off-line.

A principal e mais óbvia extensão foram a de se produzir um modelo detalhado e uma arquitetura a partir dos conceitos apresentados, construindo-se um protótipo que valide na prática estes conceitos. Esta tarefa não foi trivial e envolveu um grande esforço de pesquisa principalmente relacionado ao gerenciamento de objetos conforme os requisitos apresentados.

Como continuação deste trabalho, alguns outros aspectos e construção de workflow com agentes móveis podem ser considerados. A experiência deste trabalho mostra que se pode dar continuidade a esta pesquisa, trabalhando-se no sentido de se obter outras experiências.

## 8. REFERÊNCIAS BIBLIOGRÁFICAS

- [AMA98] AMARAL, Vinicius L. do; GRALA, Aderson Santos; LIMA, José Valdeni de. WorkFlow e Gerência de Documentos. Universidade Federal do Rio Grande do Sul. Apresentado no XVIII Congresso da Sociedade Brasileira de Computação. P.23-69.
- [ALV00] ALVES, Ana Clenir F. da Costa; Definição de um Modelo de Workflow Integrando a Cooperação e a Organização Temporal. UFMA. 2000. P.11-103.
- [BAR98] BARTHELMESS. Paulo. Sistemas de Workflow: Análise da Área e Proposta de Modelo. Instituto de Computação IC – UNICAMP 1998. P.12-33.
- [CHL95] CHESS, D.M.; HARRISON, C. G.;LEBINE, D;. Itininerant Agents for Mobile Computing IBM Reserach Report RC 20010, IBM Research Division, 1995. P.97-103.
- [CRU98] CRUZ, Tadeu. Workflow: Uma Tecnologia que vai revolucionar processos. São Paulo: Atlas, 1998. P.10-70.
- [DAV95] DAVIES, N.J.; WEEKS, R.: Jasper: Communicating Information Agents. Proceedings od the 4th International Conference on the World Wide Web. USA, December, 1995. P.121-150.
- [FAU96] FAUSTMMAN, Gert. Workflow Management and Causality trees. In Proceedings of the Second International Conference on the Design of Cooperative Knowledge Based System (COOP'96), France, June 12-14, 1996. P.40-76.
- [GEN94] GENESARETH, M.; KETCHPEL, S.: Software Agents. Communications of ACM. n.7, v.4, july 1994. P.48-53.

[HAR96] HARRISON, C.; CHESS, D.; KERSHENBAUM, A. Mobile agents: are they a good idea? In: Second International Workshop of Mobile Objects Systems. Proceedings... Linz: [s.n.], 1996. P. 25-45.