

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

Carlos Massami Kaneko

**USO DE AGENTES DE SOFTWARE PARA BUSCA
DE INFORMAÇÕES EM NÍVEL GERENCIAL**

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

Prof. Dr. Jovelino Falqueto

Florianópolis, agosto de 2005

USO DE AGENTES DE SOFTWARE PARA BUSCA DE INFORMAÇÕES EM
NÍVEL GERENCIAL

Carlos Massami Kaneko

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.

Prof. Dr. Raul Sidnei Wazlawick,
Coordenador do Curso de Pós-Graduação
Em Ciência da Computação

Banca Examinadora

Prof. Dr. Jovelino Falqueto
Presidente da Banca (orientador)

Prof. Dra Anita Maria da Rocha Fernandes
Membro da Banca

Prof. Dr. Mauro Roisenberg
Membro da Banca

Prof. Dr. Rogério Cid Bastos
Membro da Banca

“Ninguém é tão grande que não
possa aprender, nem tão pequeno
que não possa ensinar”

(Voltaire)

Dedico este trabalho aos meus pais
que sempre incentivaram o estudo
e esforçaram para proporcionar uma
boa formação.

Sumário

Sumário.....	i
Lista de figuras	iii
Lista de tabelas	v
Lista de abreviaturas	vi
Resumo	vii
Abstract	viii
1 INTRODUÇÃO.....	1
1.1 Objetivo	3
1.2 Objetivos específicos.....	3
1.3 Resultados esperados.....	4
1.4 Organização do trabalho.....	4
2 OS SISTEMAS DE INFORMAÇÃO – SI.....	6
2.1 Sistemas de informação nas empresas.....	7
2.1.1 Gerentes e os Sistemas de Informação	9
2.1.2 Sistemas de Informação e desempenho deficiente	10
2.2 Sistemas transacionais	12
2.3 Sistemas de Informações Gerenciais	13
2.4 Sistemas de Suporte à Decisão – SSD.....	17
2.5 Síntese.....	22
3 SISTEMAS DE AGENTES	23
3.1 Histórico	23
3.2 Agentes de software	24
3.2.1 Características.....	25
3.2.2 Classificação	27
3.3 Plataformas de agentes	30
3.3.1 Agentes móveis:	30
3.3.2 Sistemas multi-agentes :	31
3.4 Padrões de agentes.....	32
3.4.1 Mobile Agent Facility - MAF.....	32
3.4.2 <i>Foundation for Intelligent Physical Agents</i> - FIPA.....	34
3.5 <i>Java Agent DEvelopment Framework</i> - JADE.....	37
3.5.1 Introdução ao <i>Framework</i> JADE.....	38
3.5.2 Arquitetura da plataforma JADE.....	39
3.6 Outras tecnologias vinculadas	44
3.6.1 <i>Java Native Interface</i> (JNI)	44
3.6.2 DTMF – <i>Distributed Management Task Force</i>	45
3.7 Conclusões sobre agentes	49
4 Delegação de tarefas a agentes	50
4.1.1 Agentes: Uma tecnologia consolidada	51
4.1.2 Agentes com mais inteligência.....	51
4.1.3 Tecnologia que mantém os SI legados	52
4.1.4 Flexibilidade para integração.....	52
4.1.5 Agentes trabalham de forma assíncrona.....	53
4.1.6 JADE Integra-se com dispositivos móveis.....	53

4.1.7	Meios para monitorar os processos através do WMI	54
4.1.8	Auditoria de <i>software</i> e <i>hardware</i> através de WMI	55
4.2	Discussão	56
5	USO DE AGENTES DE SOFTWARE PARA BUSCA DE INFORMAÇÕES A NÍVEL GERENCIAL	57
5.1	Modelo computacional	57
5.2	Estudo de caso	59
5.2.1	Instalando as ferramentas	61
5.2.2	Implementação do agente	62
5.2.3	Criação da interface GUI.....	62
5.2.4	Criação dos agentes	63
5.2.5	Execução das tarefas pelos agentes no destino.....	64
5.2.6	Execução das rotinas pela DLL em Delphi	65
5.2.7	Busca de dados WMI.....	66
5.2.8	Os resultados obtidos com a implementação do modelo de agentes.....	67
5.2.9	Avaliação dos resultados sobre agentes	73
5.2.10	Resultados obtidos utilizando a tecnologia WMI.....	74
5.3	Conclusões do modelo proposto.....	75
6	CONCLUSÕES E TRABALHOS FUTUROS	76
7	REFERÊNCIAS BIBLIOGRÁFICAS	77

Lista de figuras

2.1	Inter-relacionamentos entre sistemas. Fonte: LAUDON (2004)	7
2.2	O crescimento da economia da informação. Fonte LAUDON (2004)	8
2.3	Classificação dos sistemas de informação como operacionais e gerenciais. Adaptado de O'BRIEN (2004)	8
2.4	Exemplo de relatório que pode ser produzido pelo SIG. Adaptado de LAUDON (2004)	14
2.5	Requisito de informação para os tomadores de decisão. Adaptado de O'BRIEN (2004)	18
2.6	Relatório em planilha eletrônica com dados estatísticos para SSD	19
3.1	Modelo conceitual da MAF da OMG. Adaptado de UTO (2003)	34
3.2	Modelo de referência da FIPA para Agentes. Adaptado de FIPA (2005)	35
3.3	Arquitetura abstrata FIPA. Adaptado de JADE (2005)	36
3.4	Padrão FIPA componentes do modelo de comunicação. Fonte BELLIFEMINE et al, (2003).	37
3.5	Representação de um middleware. Adaptado de BELLIFEMINE et al, (2003)	38
3.6	Arquitetura JADE. Adaptado de BELLIFEMINE et al, (2003)	40
3.7	Representação dos <i>containers</i> em plataformas da rede. Fonte JADE (2005)	42
3.8	Serviço de páginas amarelas. Adaptado de JADE (2005)	43
3.9	Interface de ponteiros do JNI. Adaptado de JNI (2005)	45
3.10	Arquitetura WMI. Fonte MICROSOFT (2005)	48
5.1	Arquitetura da solução computacional	58
5.2	Arquitetura de mobilidade	59
5.3	Classe de mobilidade do agente JADE	63
5.4	Classe de agente utilizando o JNI	64
5.5	Agente chama as rotinas da DLL	65
5.6	Programa em Delphi integrado com o JNI	66
5.7	Funções coleta dados WMI	67
5.8	Iniciando a plataforma Jade através da interface texto	68
5.9	Plataforma JADE	68
5.10	Criando um novo agente	69
5.11	Criando um novo agente de nome teste	69

5.12	Interface do agente criado com nome teste	70
5.13	Interface para clonar o agente teste	71
5.14	Solicitando e-mail	72
5.15	Agente executando rotinas de busca de informações gerenciais	72
5.16	Interface texto após execução de rotina de busca de informações gerenciais	73
5.17	Interface gráfica da aplicação Delphi para coleta de informações WMI	75

Lista de tabelas

3.1	Características de agentes. Fonte WOOLDRIDGE & JENNINGS (1995)	25
3.2	Propriedades necessárias de um agente. Fonte BUI & LEE (1999)	26
3.3	Arquiteturas de Agentes. Fonte: SOUZA (1996)	29

Lista de abreviaturas

ACC	Agent Communication Channel
ACL	Agent Communication Language
AMS	Agent Management System
API	Aplication Programming Interface
BI	Business Inteligent
CIM	Common Information Model
CORBA	Common Object Request Broker
DF	Directory Facilitator
DLL	Dinamic Link Library
DTMF	Distributed Management Task Force
FIPA	Foundation for Intelligent Physical Agents
GUI	Graphical User Interface
HTTP	Hiper Text Transfer Protocol
IA	Inteligência Artificial
IDL	Interface Definition Language
IIOP	Internet Inter-Orb Protocol
IP	Internet Protocol
J2ME	Java 2 Micro Edition
JADE	Java Agente Developmente Framework
JESS	Java Expert System Shell
JNI	Java Native Interface
JVM	Java Virtual Machine
KQML	Knowledge Query Manipulation Language
LGPL	Lesser General Public License
MAF	Mobile Agent Facility
MOF	Managed Object Format
OMG	Object Management Group
PDA	Personal Digital Assistant
RMA	Remote Monitoring Agent
RPC	Remote Procedure Calling
SACI	Simple Agent Communication Infrastructure
SAD	Sistema de Apoio a Decisão
SAE	Sistema de Apoio a Executivo
SI	Sistema de Informação
SIC	Sistema de Informação Colaborativo
SIG	Sistema de Informações Gerenciais
SPT	Sistema de Processamento de Transações
SQL	Structure Query Laguage
SSD	Sistema de Suporte a Decisão
STC	Sistema de Trabalhadores do Conhecimento
TCP	Transmission Control Protocol
UML	Unified Modeling Language
WBEM	Web Based Enterprise Management
WMI	Windows Management Instrumentation
WSL	WMI Query Language

Resumo

Os Sistemas de Informação têm atuado nas empresas proporcionando competitividade, e criando um nicho para aplicação de métodos e tecnologias que permitam diferenciais no contexto da globalização. Os processos de terceirização, globalização e a Internet tornaram as organizações distribuídas. Para manter o controle e o gerenciamento sobre esta nova realidade são necessárias novas tecnologias nas corporações. O uso de agentes de software visa proporcionar recursos atuais atrelados a uma realidade de maior exigência da área de administração. Dessa forma, algumas tarefas podem ser delegadas aos agentes, de maneira a auxiliar nas tarefas de controle da organização. O presente trabalho busca contextualizar a realidade atual em termos de Sistemas de Informação nas empresas e é apresentada uma proposta para aplicação de novas tecnologias aproveitando o legado existente. Dentro desse contexto são utilizados agentes para a execução de tarefas delegadas por administradores, permitindo a agregação de importantes premissas tecnológicas a um investimento relativamente baixo. Os resultados obtidos validam a aplicação do modelo, criando um ambiente com facilidade de expansão e de agregação de novas características, tais como a utilização de técnicas de inteligência artificial e comunicação como dispositivos móveis.

Abstract

Information Systems provide competitiveness to companies, and create a locus for the application of methods and technologies which allow better chances in the globalization context. The organizations became more distributed with use of third part services, globalization process and the Internet, among other factors. To maintain the control and the management in this new reality, new technologies are required. The use of software agents aims to supply modern resources linked to a higher requirement reality in administration area. Thus, some tasks can be delegated to the agents, assisting in the organization control tasks. This work contains a study of the state of the art in Information Systems in companies and presents a proposition for application of new technologies supporting the software already installed. In this context, agents for the execution of existing tasks for administrators are tested, allowing the aggregation of important technological tools with comparatively low investment. Obtained results support the application of the model, creating an environment witch permits expansion and use of new characteristics, such as utilization of Artificial Intelligence techniques and mobile devices communication.

1 INTRODUÇÃO

Atualmente as empresas precisam ter rigoroso controle sobre tudo, seja na produção, na compra, na venda, nos seus fornecedores, nos clientes que atende. Sistema de informação tem o objetivo de alcançar este controle sobre as organizações, mas mesmo empresas que implantaram sistemas de informações apresentam dificuldades no seu gerenciamento.

A identificação das causas destas dificuldades pode ser das mais variadas. O problema pode ser de cunho pessoal, profissional, tecnológico e cultural. Além de ter-se que considerar que pode ser uma combinação de fatores.

As dificuldades também podem estar relacionadas à revolução tecnológica que a Internet está promovendo Laudon, (2004). Este fato pode estar exigindo dos gestores uma atualização em termos de monitoramento das atividades da empresa, gerenciamento dos recursos e criação de estratégias inovadoras ou podem estar relacionadas a uma sobrecarga de atividades exercidas por eles.

Com a globalização da economia as empresas passaram por processos de ajustes, com o objetivo de manter-se competitivo ao mercado, onde algumas passaram por processo de *downsizing*¹, outras fizeram *outsourcing*², mas de uma forma geral passaram por processos de reengenharia.

Uma nova realidade tomou conta das empresas, os sistemas de informações passaram a fazer parte da rotina da empresa, agora com mais controles, informações maduras, adequadas às necessidades das empresas, mas ainda existem sistemas independentes, sem integração, e, em conseqüência, informações divergentes.

O uso do computador como ferramenta básica de trabalho passou ser uma constante, gerando uma dependência muito forte, melhorando a qualidade do trabalho, mas também criando a necessidade de administrar, além das pessoas, os computadores. Novos controles se fizeram necessários, como a qualificação dos usuários para o uso, a manutenção dos equipamentos, gerência dos softwares instalados nos terminais, além de atualização em novas tecnologias parecem necessidades da empresa.

¹ *Downsizing* – processo caracterizado pela mudança de equipamentos de grande porte (mainframes) para equipamentos interligados por redes de computadores.

² *outsourcing* – processo onde corporações passam a terceirizar o processo de produção para empresas menores com o objetivo de reduzir custos.

Surtem então questões como a de determinar o desempenho de um funcionário que utiliza o computador como ferramenta de trabalho. Como saber se um determinado funcionário está trabalhando para alcançar os objetivos da empresa ou para a busca de seus objetivos particulares?

Observa-se nas organizações que existe uma tendência ao retorno da administração centralizada, as decisões até então descentralizadas voltam a se concentrar em pontos específicos, resultado da alta conectividade da rede. O modelo cliente-servidor passa a imperar. A necessidade de redução de custos faz a Internet chegar às empresas. Neste modelo espera-se que o gerente possa administrar empresas que ficam no outro lado do planeta. Será que as informações que os gerentes estão acessando são suficientes para as decisões? Qual a influência do acesso à Internet nas empresas?

Outra modalidade que surge é o trabalho em casa, sem a necessidade de estar no escritório: o trabalho a distância, novamente utilizando a tecnologia como base.

O foco do trabalho nas organizações está nos Sistemas de Informação, que podem ser classificados em Sistemas de Processamento de Transação – SPT, Sistemas de Informações Gerenciais – SIG, Sistemas de Suporte à Decisão – SSD e o Sistema de Apoio a Executivo – SAE (O'BRIEN, 2004).

Caracterizar os tipos de Sistemas de Informação existentes e suas funções permite localizar o tipo de informação desejado, quem irá fornecer e qual a sua importância.

A busca por determinar as dificuldades de gerenciamento das empresas que utilizam os Sistemas de Informação pode apresentar características comportamentais, pessoais ou profissionais, mas considerando que a variante mais dinâmica está relacionada ao Sistema de Informação e às tecnologias aplicadas a ela, é razoável considerar o estudo para determinar o impacto que o seu uso pode ocasionar e se o uso de novas tecnologias como a de agentes de software auxilia a solucionar estes tipos de problemas.

A abordagem de agentes de software é tecnologia emergente que propõe a criação de softwares com uso de inteligência, com recursos de mobilidade, com característica de desempenhar funções em nome de outro, apresenta funcionalidades favoráveis ao contexto estudado. Utiliza para desenvolvimento linguagens como Java, Python, C++, entre outras, sendo que o maior número de plataformas está na linguagem Java. Tem

uma característica que potencializa a portabilidade de software, permitindo que independente da plataforma utilizada para o sistema de informação, possibilite adotar uma solução de agentes.

Algumas pesquisas propõem o uso de agentes para integração de bases de dados utilizando mobilidade de código (CLARO, 2000); uma outra modela a busca em Sistemas de Suporte à Decisão em Grupo (REES e KOEHLER, 2004); outra propõe a criação de um *framework* baseado em agentes para construção de Sistemas de Suporte à Decisão (BUI e LEE, 1999). O presente trabalho compactua com a importância destes trabalhos, porém identificou-se uma motivação imediata, a necessidade da utilização do conceito de agente, principalmente na delegação de tarefas, buscando manutenção dos sistemas legados.

Considerando que as características do problema requerem soluções com flexibilidade e recurso de processamento, parece ser adequada a utilização da tecnologia de agentes de software como ferramenta para solucionar o problema da dificuldade no gerenciamento de empresas distribuídas.

1.1 Objetivo

Usar de agentes como assistentes de gerente na busca de relatórios em Sistemas de Suporte a Decisão em ambiente corporativo distribuído.

Utilizar a capacidade de mobilidade para permitir que o agente percorra os vários pontos da organização, em ambiente distribuído, coletando as informações que a gerência necessita. Criar agentes que acessem estações para avaliação de *software* e *hardware*.

1.2 Objetivos específicos

- Estudar os agentes de software que possuam características adaptáveis ao ambiente corporativo;
- Avaliar as necessidades do ambiente corporativo que permitam o uso de agentes de software;
- Verificar o uso do agente como ferramenta nas necessidades gerenciais das corporações;

- Fazer um estudo de caso para avaliar o desempenho dos agentes

1.3 Resultados esperados

- A determinação de fatores tecnológicos ou sistêmicos como causa da dificuldade no gerenciamento das organizações, através de uma visualização do contexto corporativo, e da situação atual dos sistemas e tecnologias disponíveis;
- Uma análise da influência de cada fator tecnológico sobre o processo de tomada de decisão, identificando até que ponto os recursos tecnológicos influenciam as organizações;
- Avaliação da utilização da tecnologia de agentes de software como ferramenta para a coleta e o monitoramento das organizações. Propondo um modelo utilizando agentes, e fazendo a avaliação de um estudo de caso para comprovar a viabilidade técnica do assistente virtual;
- Avaliação do modelo proposto através de experimentos.

1.4 Organização do trabalho

O primeiro capítulo apresenta uma introdução ao assunto, contextualizando em linhas gerais a situação dos Sistemas de Informação nas organizações.

O segundo capítulo é produto de uma pesquisa bibliográfica sobre os Sistemas de Informação, seu conceito, sua classificação, sua aplicação, seus problemas e suas características nas organizações.

O terceiro capítulo aborda os agentes, procurando mostrar o conceito, seu histórico, os padrões existentes, sua classificação, a arquitetura e as plataformas atuais. Detalha as características da plataforma JADE, por ser esta a plataforma escolhida para o experimento (a ser trabalhado no Capítulo 5). Trata também de outras tecnologias como JNI (*Java Native Interfaces*), WBEM (*Web-based Enterprise Management*) e WMI (*Windows Management Instrumentation*).

O quarto capítulo aborda o cenário organizacional, as características tecnológicas e relaciona os principais aspectos vinculados ao contexto empresarial.

O quinto capítulo mostra o modelo computacional proposto para a solução do problema, e o estudo de caso detalhando os materiais e métodos utilizados na realização do experimento.

2 OS SISTEMAS DE INFORMAÇÃO – SI

Os Sistemas de Informação (SI) são atualmente essenciais nas organizações. Sua importância operacional e estratégica é inquestionável, torna o negócio competitivo, facilita a sua administração e fornece base para a sua manutenção e crescimento (STAIR, 1998; JAMIL, 2001; LAUDON, 2004; O'BRIEN, 2004).

Laudon define SI como:

“Um sistema de informação pode ser definido tecnicamente como um conjunto de componentes inter-relacionados que coleta (ou recupera), processa, armazena e distribui informações destinadas a apoiar a tomada de decisões, a coordenação e o controle de uma organização.” (LAUDON, 2004).

O'Brien define que:

“...é importante que você veja os Sistemas de Informação como algo mais do que um conjunto de tecnologias que apoiam a colaboração entre grupos de trabalho e empresas, as operações eficientes das empresas ou a tomada de decisões gerenciais eficazes.” (O'BRIEN, 2004).

Os Sistemas de Informação passaram a assumir muitas funções de controle nas organizações. Jamil (2001) classifica os sistemas em transacionais e analíticos, sendo o primeiro caracterizado por aplicações construídas para armazenar, extrair, processar e emitir conteúdos de arquivos e bancos de dados. Já o segundo permite análise através de estudos, comparação, processamento dos dados adquiridos e armazenados em bases de dados, resultados do sistema transacional. Jamil (2001), Laudon (2004), O'Brien (2004), Stair (1998) e Rezende e Abreu (2001) por sua vez propõem que os sistemas são divididos segundo o nível organizacional que atendem. Quando for sistema a nível operacional, são ditos Sistemas de Processamento de Transações (SPT), no nível de conhecimento, são chamados de sistema de trabalhadores do conhecimento (STC) ou

também chamado de Sistemas de Informação Colaborativos (SIC); no nível gerencial são ditos Sistemas de Informações Gerenciais (SIG) e Sistemas de Apoio à Decisão (SAD) ou Sistema de Suporte a Decisão (SSD), e no nível estratégico de Sistemas de Apoio a Executivos (SAE) (STAIR, 1998; REZENDE e ABREU, 2001; LAUDON, 2004, O'BRIEN ,2004).

Os vários tipos de sistemas da organização têm interdependências, (vide Figura 2.1) sendo que os STPs são os maiores fornecedores de informações para os demais, que por sua vez produz informações para outros sistemas.



Figura 2.1 Inter-relacionamentos entre sistemas. Fonte: Laudon (2004)

2.1 Sistemas de informação nas empresas

Laudon (2004) argumenta que a economia dos Estados Unidos vêm sofrendo um declínio contínuo no número de trabalhadores rurais e operários de fábricas. Simultaneamente, o país está experimentando um aumento no número de profissionais de escritório (Figura 2.2), que produzem valor, utilizando a tecnologia da informação.

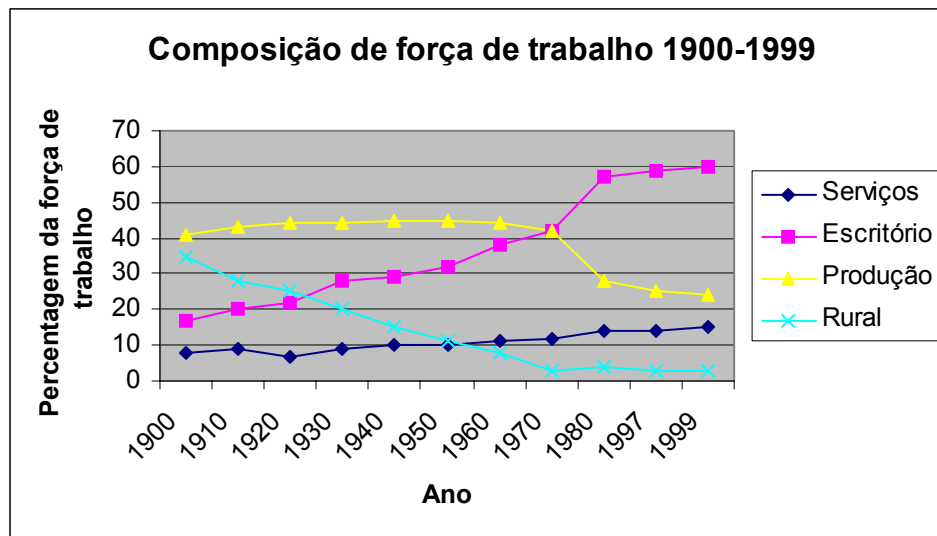


Figura 2.2 O crescimento da economia da informação. Fonte Laudon (2004)

Os Sistemas de Suporte à Gerência (ou Sistema de Apoio à Gerência) que envolvem os Sistemas de Informações Gerenciais (SIG), Sistema de Suporte a Decisão (SSD) e Sistema de Apoio a Executivo (SAE) trabalham com problemas de tipos diferentes. Na estrutura hierárquica apresentam uma divisão bem nítida, conforme Figura 2.3. O SIG está relacionado com problemas bem estruturados, descritos como repetitivos, rotineiros e com procedimento específico para seu manuseio. O SSD se relaciona com problemas semi-estruturados, sendo caracterizado por apresentar somente parte do problema com respostas definidas. Já o SAE envolve questões em aberto e com certo grau de não estruturação, e dependente de grande volume de informações originados do ambiente externo a empresa. (LAUDON e LAUDON, 1999)



Figura 2.3 Classificação dos Sistemas de Informação como operacionais e gerenciais. Adaptado de O'Brien (2004)

2.1.1 Gerentes e os Sistemas de Informação

Há três décadas o gerente tinha uma formação especializada, focado em temas como produção, finanças, recursos humanos, comercial, etc. Sua qualificação era para gerir pessoal, coordenar equipes para alcançar metas da diretoria e controlar produtos e serviços. (JAMIL, 2001)

As transformações organizacionais nos anos de 1990, caracterizadas pela Reengenharia, *Outsourcing*, fusões, entre outras, provocaram mudanças nas organizações, em alguns casos radicais. Os gerentes de nível médio não eram mais necessários para fazer o elo de ligação entre o setor operacional e a alta administração. O cenário das organizações, após as transformações, é caracterizado pela redução no quadro de funcionários. Cargos deixaram de existir, entre eles a dos gerentes médios, que além do papel de transmitir a situação e dar retorno ao superior, tinham função de controlar e conferir as atividades operacionais. Estas funções foram atribuídas aos funcionários de nível mais baixo, ou então atribuídas aos sistemas de informações (JAMIL, 2001; LAUDON, 2004; O'BRIEN, 2004).

Neste novo contexto, muito mais competitivo e dinâmico, os gerentes foram liberados de analisar números para enfrentar novos desafios em questões estratégicas, e da criação de alternativas de crescimento coerente e passíveis de planejamento (JAMIL, 2001; O'BRIEN, 2003).

Isto não significa que as funções de coordenar, controlar e prever produtividade foram deixadas de lado, na verdade estas já são consideradas automáticas. Dessa forma os gerentes necessitam dos Sistemas de Informações, para dar condições aos gerentes para que possam promover mudanças inovadoras na tomada de decisão gerencial, estruturas organizacionais e atividades de trabalho gerencial (O'BRIEN, 2004).

Outra característica atribuída ao gerente está na habilidade na condução de equipes, o que envolve a escolha, montagem, coordenação e manutenção da equipe. Ainda podemos considerar manutenção do foco no negócio, treinamentos, o controle de desempenho individual e do grupo, a integração, motivação, gestão da criatividade, gestão do conhecimento e mecanismos constantes de avaliação (JAMIL, 2001).

Segundo Jamil (2001) o gerente, na nova forma de trabalhar, deve exercer o papel de liderança, mantendo coesão do grupo e do ambiente. O mesmo deve estar monitorando constantemente o desempenho, tendências e condições de trabalho da equipe e o desenvolvimento de seus produtos.

Está claro que o perfil do gerente especializado deixa de existir na empresa moderna. O perfil do novo gerente é mais dinâmico, dominador de tecnologias e sabe se inteirar com as pessoas, utiliza-se do tempo para criar e definir estratégias de negócios, utiliza intensamente os Sistemas de Informação como ferramenta para alcançar seus objetivos.

2.1.2 Sistemas de Informação e desempenho deficiente

A tecnologia da informação passou a exercer maior importância nas mudanças organizacionais e gerenciais das empresas. Sistemas cliente/servidor impulsionaram o desenvolvimento de *intranet*, *extranet* e sistemas de comércio eletrônico. Neste contexto, o gerenciamento da tecnologia da informação está longe de ser uma tarefa fácil, principalmente considerando que geralmente os gerentes vêm de áreas como vendas e marketing. (O'BRIEN, 2004)

Para Jamil (2001) os sistemas que selecionam, processam e disponibilizam informações são preciosas peças de conhecimento, e que não há mais espaço para sistemas que fornecem informações incorretas, incompletas, imprecisas e não se deve aceitar qualquer outra informação no lugar da que realmente foi solicitada.

O'Brien (2004) argumenta que muitos Sistemas de Informação apresentam problemas de desempenho, sendo que alguns casos documentam que os benefícios esperados simplesmente não ocorrem. Identifica os principais problemas nas empresas que buscam simplesmente informatizar os processos tradicionais, ou quando os Sistemas de Informação não são considerados como um serviço de informações para empresa. Caracteriza, nestes casos, carência de informações para apoio à decisão, a falta de integração, tempos de respostas deficientes, paradas frequentes, acúmulo de pedidos de desenvolvimento. Nestes casos a tecnologia da informação deixa de cumprir o papel competitivo e inovador e passa a ser considerado custo.

Laudon (2004) enfatiza a necessidade de certificar que o sistema de informação deve ajustar-se ao plano empresarial, e que um projeto de sistema de informação nada mais é do que um re-projeto da organização. A montagem de um sistema de informação é uma forma de mudança organizacional planejada que afeta toda a organização, focado principalmente em conflitos organizacionais e tomadas de decisão. Outro ponto destacado é que um sistema pode apresentar um sucesso tecnológico, mas um fracasso organizacional que pode estar relacionado com processos políticos e sociais.

O envolvimento do usuário final, da administração, dos gerentes, até mesmo do presidente da empresa parece ser uma proposta importante. A criação de políticas de incentivo à participação ativa no planejamento e controle dos usos da tecnologia da informação pelos gerentes, tem evitado problemas de desempenho dos Sistemas de Informação. Este envolvimento proporciona uma melhora no valor estratégico proveniente da tecnologia da informação nos negócios. (O'BRIEN, 2004)

Um plano de desenvolvimento de Sistemas de Informação com a clara compreensão dos requisitos de informações da organização, seja ela a curto ou longo prazo, pode determinar um sistema adequado, segundo Laudon (2004). As ferramentas propostas estão embasadas em análise de negócio e análise estratégica ou fatores críticos de sucesso.

Os sistemas também geram problemas, embora se deseje, os benefícios nem sempre são os que acontecem. A visão apresentada por Jamil demonstra que não há mais espaço para informações imprecisas, incorretas e inconsistentes, enfim informação sem qualidade.

O'Brien (2004) relata a dificuldade enfrentada pela organização quando o sistema não atende às necessidades, atribui a falta de qualidade do sistema e a falta de comprometimento dos usuários e gerentes, sendo que propõe soluções com maior envolvimento dos atores.

Já Laudon (2004) aborda a influência que o sistema tem sobre a organização, a necessidade de avaliar a organização como um todo, buscando trabalhar o nível organizacional e o tecnológico, de forma a estabelecer um planejamento focado nos objetivos organizacionais e não nas pessoas.

2.2 Sistemas transacionais

São sistemas de nível operacional que fornecem suporte a gerentes operacionais, relacionados à atividades rotineiras como vendas, contas a pagar, folha de pagamento, fluxo de matéria prima, reservas de passagens, pedidos de venda, lançamentos de caixa, entre outros (LAUDON e LAUDON, 1999; LAUDON, 2004).

Uma transação é tratada como um registro de um evento na empresa, onde é esperada uma resposta da mesma. Um pedido de venda é um exemplo clássico. Este deve ser registrado através de uma transação: o mesmo é processado e deve gerar baixa no estoque, impressão da nota de venda, lançamento no sistema de contas a receber, caso seja a prazo, ou lançamento no sistema de caixa, como aguardando pagamento, caso seja a vista, geração do relatório de carga e liberação dos produtos para o cliente (LAUDON e LAUDON, 1999).

Sistemas de processamento de transações podem ser com processamento em tempo real (ou *online*), onde os dados são processados no momento da transação. Neste caso é chamado de Sistema Processamento de Transações On-Line (SPTOL), ou com processamento em lote (ou *batch*), onde as transações são acumuladas durante um período e então são processadas (O'BRIEN, 2004; LAUDON, 1999).

Stair (1998) salienta ser o Sistema Transacional o centro da maior parte das organizações empresariais, dando apoio à monitoração e à realização de negócios de uma organização. Sendo que o processamento inclui a coleta de dados, processamento de cálculos, armazenamento dos dados no banco de dados, e/ou produção de documento e relatórios operacionais.

Estes sistemas geralmente são utilizadas por pessoas com pouca ou nenhuma responsabilidade. Os sistemas exigem pouca ou nenhuma decisão dos seus operadores, porém as empresas necessitam deles para o funcionamento diário. São estes sistemas que alimentam os Sistemas de Informações Estratégicos, e com eles os gerentes médios monitoram e controlam as transações para saber o desempenho da empresa. (LAUDON e LAUDON, 1999)

Algumas das características que definem um Sistema de Processamento de Transações proposto por Stair (1998) e Laudon e Laudon (1999) (SPT) são:

- Grande volume de dados de entrada;

- Grande volume de saídas, entre documentos e arquivos de dados;
- Exigência de processamento eficiente;
- Permitir rapidez na entrada e saída de dados;
- Permitir repetição de processamento;
- Ter Capacidade de armazenamento em massa e
- Possibilitar auditoria (para garantia de consistência e confiabilidade na saída dos dados);

2.3 Sistemas de Informações Gerenciais

Surgiram na década de 1960 quando os Sistemas de Informação deixaram de ter o papel de simples processamento de transações e passaram a gerar relatórios gerenciais administrativos com a finalidade de auxiliar a tomada de decisão.(O'BRIEN, 2004)

Para Batista (2004), o SIG é imprescindível à corporação que busca uma gestão participativa e horizontalizada, pois unifica conceitos, agiliza e uniformiza a análise de dados. É baseado no conhecimento específico do negócio e preparado para selecionar informações estratégicas e atividades críticas da empresa. Há necessidade de se manter sempre sintonizado com o dinamismo que o mundo dos negócios impõe às corporações.

São sistemas que oferecem relatórios resumidos e consultas em tela sobre os resulta

dos obtidos pela empresa aos gerentes (Figura 2.4). Estes são utilizados para monitorar, controlar e prever os futuros desempenhos do negócio: dessa maneira o gerente supervisiona as atividades da empresa e abre a possibilidade para intervenção quando da identificação de problemas. (STAIR, 1998; LAUDON e LAUDON, 1999; O'BRIEN, 2004)

Relatório Consolidado de vendas corporativas Por Produto e Região : 2002					
Código do Produto	Descrição do Produto	Região	Vendas Efetuadas	Vendas Previstas	Efetuadas/Previstas
4469	Limpador de de carpete	Nordeste	4.066.700	4.800.000	0,85
		Sul	3.778.112	3.750.000	1,01
		Centro-Oeste	4.867.001	4.600.000	1,06
		Norte	4.003.440	4.400.000	0,91
		Total	16.715.253	17.550.000	0,95
5674	Desodorizador de ambientes	Nordeste	3.676.700	3.900.000	0,94
		Sul	5.608.112	4.700.000	1,19
		Centro-Oeste	4.711.001	4.200.000	1,12
		Norte	4.563.440	4.900.000	0,93
		Total	18.559.253	17.700.000	1,05

Figura 2.4 Exemplo de relatório que pode ser produzido pelo SIG. Adaptado de Laudon (2004)

O SIG fornece informações úteis para obter o *feedback* das muitas operações empresariais, dando suporte ao valor agregado de uma organização. Um processo desejável é que os SIG forneçam a informação adequada à pessoa certa, da maneira correta e no momento certo. Geralmente este aspecto é esquecido em detrimento à eficiência na integração (STAIR, 1998; O'BRIEN, 2004)

Batista (2004) argumenta sobre a importância de se definir as características da informação desejada, colocando que os principais pontos são:

- forma – refere-se ao conteúdo, a maneira de apresentar e à confiabilidade necessária;
- idade – o tempo de existência da informação em relação aos dados transacionais;
- frequência – a periodicidade com que é gerada a informação.

Estes fatores dependem diretamente dos sistemas de processamento de transações, pois apresentam resumo simples de repetições em relatórios, demonstrando o desempenho das transações. Pode-se ilustrar através das transações de produtos vendidos, produzidos, comprados, na forma de relatórios. O gerente avalia os resultados

e quando algo apresenta desvio do plano da organização o mesmo pode tomar as medidas necessárias. (STAIR, 1998; LAUDON e LAUDON, 1999)

A obtenção de informações gerenciais dos sistemas para a tomada de decisão é um dos problemas, pois geralmente os sistemas estão limitados a informações a nível operacional. A qualidade da informação gerencial é diretamente proporcional ao envolvimento do nível gerencial na elaboração e planejamento do sistema de informação (BATISTA, 2004).

O Sistema de Processamento de Transações é a principal fonte interna de dados para o SIG, sendo que estes dados obtém-se de áreas funcionais específicas de toda empresa. (STAIR, 1998; O'BRIEN, 2003)

O SIG gera uma coleção de relatórios que é distribuída aos gerentes, que podem ser divididos em relatórios programados, relatórios por solicitação e relatórios de exceção. O'Brien (2003) inclui ainda relatórios em pilha, porém este tipo de relatório não apresenta relevância para o trabalho e não será apresentado. (STAIR,1998; O'BRIEN, 2004)

Relatórios programados são gerados periodicamente ou de maneira programada, diariamente, semanalmente ou mensalmente, utilizados para controlar, monitorar o desempenho da empresa. Exemplos de relatórios programados são limites de crédito dos clientes, vendas de um vendedor, produtos abaixo do estoque. Um tipo particular de relatório programado é o relatório indicador de pontos problemáticos, que resume os pontos importantes do dia anterior, ligados aos fatores críticos de sucesso de uma organização. Desta forma tem o objetivo de tomar providências a respeito de algum tipo de problema crítico identificado. (STAIR, 1998; O'BRIEN, 2004)

Relatórios solicitados são desenvolvidos para fornecer informações personalizadas, geralmente para apresentar uma informação específica que é importante ao negócio. Sendo que o gerente pode obter de seu próprio ponto de acesso ao sistema, não sendo necessário aguardar o relatório periódico. (STAIR, 1998; O'BRIEN, 2004)

Relatório de exceção, é gerado automaticamente quando uma situação incomum é identificada ou em condições excepcionais, e requer uma atitude imediata. Estes relatórios reduzem a sobrecarga de informações, pois requerem a decisão imediata, reduzindo a necessidade de relatórios carregados de informações. (STAIR, 1998; O'BRIEN, 2004)

A utilização das informações do SIG caracteriza um ambiente empresarial estruturado e organizado, porém é exigido um grau de confiabilidade das mesmas. Tomar decisão com base em informações não confiáveis é pior que não possuir informação (BATISTA, 2004).

Outro ponto importante destacado está na qualidade da informação, dados desnecessários apresentados em relatórios não apresentam informações relevantes ao negócio. A qualidade é tão importante quanto a decisão tomada com base nela, para que a qualidade da informação seja o mais próximo da necessária, é importante a participação da pessoa que irá utilizá-la (BATISTA, 2004).

É claro que é primordial que a organização defina suas necessidades de informações gerenciais, que as pessoas que trabalharão com estas informações tenham a exata importância de sua função referente à informação e saibam trabalhar com elas (BATISTA, 2004).

Do ponto de vista empresarial, o SIG auxilia na resolução do problema de redução de custo de coordenação, isto é coordenar muitas pessoas em locais diferentes. Com o crescimento da empresa, cresce o número de operações. Isso deveria baixar o custo de produtos e serviços, mas muitas vezes o custo com coordenação aumenta, perdendo-se o benefício. O SIG tem o objetivo de reduzir os custos de coordenação e gerenciamento quando do crescimento em escala da empresa, sendo considerado essencial para o controle operacional do negócio (LAUDON e LAUDON, 1999).

A abordagem do SIG apresenta-se como de muita importância no negócio, além de supervisionar o nível operacional, é através do SIG que identifica-se os possíveis pontos imediatos de problema do negócio. Os autores são unânimes em afirmar a sua importância, embora muitos argumentem e poucos definam realmente como é o seu funcionamento, suas características e a forma de explorar o SIG.

Stair (1998) e O'Brien (2004) apresentam de forma mais direta, além das justificativas os tipos de relatórios, sua função e sua importância. Sendo que Stair (1998) entra em maiores detalhes a respeito do SIG fazendo abordagens sobre SIG financeiro, industrial, marketing, recursos humanos e outros, embora para o presente trabalho não sejam relevantes e por esta razão não estão sendo tratados.

Laudon e Laudon (1999) por sua vez apresenta a importância do SIG dentro do contexto empresarial, relacionando a exploração do SIG de forma competitiva e administrativa.

Batista (2004) aborda de forma clara a falta de informações a nível gerencial nos Sistemas de Informação, além de colocar em questão a confiabilidade e a qualidade das informações geradas. Em seu trabalho apresenta algumas caracterizações adicionais, porém estas foram apresentadas de forma mais estruturada por Stair e por esta razão não foram aqui evidenciadas.

Outra forma de apoio a decisões utilizado, além dos SIG's é o Sistema de Suporte a Decisão SSD, que será tratado a seguir.

2.4 Sistemas de Suporte à Decisão – SSD

Embora qualquer sistema que forneça informação possa ser chamado de Sistema de Suporte a Decisão, conceitualmente é considerado diferente de SIG ou SPT. Os SSDs exigem menos tempo e custam menos do que o SIG e são iterativos, ou seja, o usuário pode interagir diretamente com os dados. Propicia informações e modelos para apoio à discussão e são recomendados para problemas semi-estruturados. (LAUDON e LAUDON, 1999; O'BRIEN, 2004)

Stair conceitua SSD como :

“Um SSD é um conjunto organizado de pessoas, procedimentos, software, banco de dados e dispositivos utilizados para dar suporte à decisões específicas de um problema” (STAIR, 1998)

Para problemas semi-estruturados ou não estruturados a utilização de um SSD é a ferramenta mais eficaz de apoio à tomada de decisões.(STAIR, 1998; LAUDON e LAUDON, 1999; O'BRIEN, 2004)

Uma visão sobre o tipo de informação requerido por diretores, executivos, gerentes e membros de equipes auto-dirigida está diretamente relacionada com o nível de tomada de decisão gerencial envolvido e a estrutura das situações de decisão que elas enfrentam. A base da pirâmide (mais baixo em cor laranja) representa o nível

estruturado, o nível intermediário (em cor azul) o semi-estruturado, e o topo (em cor cinza) o nível não estruturado, como pode ser observado na Figura 2.5.



Figura 2.5 Requisito de informação para os tomadores de decisão. Adaptado de O'Brien (2004)

Sistemas de Suporte a Decisão têm seus fundamentos em modelos e bancos de dados como recursos vitais. (LAUDON e LAUDON, 1999; O'BRIEN, 2004)

Muitos SSD são montados a partir de softwares pré-escritos, muitas vezes exigem dados de fontes diferentes, bancos de dados corporativos e locais com sistemas avançados de gerenciamento de banco de dados. Normalmente um SSD com capacidade de manipular grande volume de dados, de fontes internas e externas, costuma ter desenvolvimento e manutenção mais dispendiosos. (STAIR, 1998)

No tocante a custos de um SSD, Laudon e Laudon(1999) e Stair (1998) divergem. É claro que comparando o desenvolvimento de um software de SSD específico, com a utilização de um aplicativo de planilha eletrônica para a geração de um SSD, Figura 2.6, acha-se uma diferença grande em termos de custos. Stair (1998) refere-se ao

desenvolvimento de um SSD, enquanto Laudon e Laudon(1999) refere-se mais a utilização de planilhas eletrônicas.

		2005											
2	COD. PRODUTO	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	MEDIA	SALDO	ULTCUSTO	VE
3	3256 TOALHA MESA	19	40	51	40	15	4	9	4	15	472	4,58	
4	3456 TESOURINHA FRUTAS	35	48	6	13	4	43	77	0	19	851	0,10	
5	243 SACHE GAVETA PCT	101	17	81	324	95	321	379	51	114	1754	0,31	
6	235 RELÓGIO	34	576	99	22	864	768	768	0	261	127	0,80	
7	3243 PORTA PETISCO	37	35	32	83	41	35	15	7	24	95	3,15	
8	34 PIZZA PEDAÇOS	29	25	40	40	0	16	50	0	17	1227	0,81	
9	6456 PENDURADOR DE BOLSA	26	66	28	46	52	16	13	0	21	1492	0,98	
10	23432 PANO DE COZINHA	24	0	24	149	85	38	72	39	36	366	0,64	
11	3462 PA DE BOLO	25	29	44	64	26	32	37	4	22	810	1,00	
12	32413 P. PERFUME CRISTAL	78	23	36	87	26	43	3	0	25	1621	0,90	
13	774 P.CARTAO C/MEMO	117	5	5	24	4	20	0	0	15	343	0,10	
14	13434 P. JOIA CLOISONE A1	245	0	0	0	10	252	149	0	55	1749	1,15	
15	34567 P. JOIA CLOISONE A2	16	9	0	0	0	121	64	0	18	2405	0,80	
16	4324 P. JOIA CLOISONE A3	162	1	0	0	0	122	25	0	26	3213	0,62	

Figura 2.6 Relatório em planilha eletrônica com dados estatísticos para SSD.

Stair (1998) apresenta como características potenciais para uma eficaz ferramenta de apoio gerencial os seguintes quesitos:

Possibilitar grande volume de dados: uso de recursos avançados de banco de dados para busca de informações;

- Permitir diferentes fontes de dados: possibilidade de busca de dados em diferentes fontes em arquiteturas diferentes;
- Ter flexibilidade de relatórios e de apresentações: sistemas mais flexíveis em conteúdo e formato que os SIG e SPT;
- Ter estilos individuais de tomada de decisões: formato personalizado as necessidades dos gerentes, em vídeo ou impressas;
- Ter formato modular: módulo financeiro, marketing, materiais por área;
- Realizar análises sofisticadas: comparações complexas e sofisticadas;
- Ter orientação gráfica: apresentação na forma de textos, tabelas, desenhos lineares, gráficos pizza, curvas de tendência, entre outros;
- Fazer abordagens de otimização e heurística: nos casos simples utilizar para otimizar a escolha, nos casos complexos para sugerir a melhor abordagem e

- Ter função de análises de simulações e de atingir metas: simulação consiste em alterar as hipóteses para avaliar o impacto no resultado, ou atingir metas estipulando um resultado para ser alcançado.

As características de um SSD são basicamente apresentadas somente por Stair (1998). Os demais autores pesquisados apresentam argumentações sobre a sua importância e sobre os componentes de um SSD.

O SSD genérico é composto de três componentes, o primeiro é um banco de dados que geralmente tem origem no sistema de transações, o segundo possui um modelo de dados, ou ferramentas analíticas e o terceiro um sistema de software que permita fácil interação com o usuário final. (LAUDON e LAUDON, 1999)

A facilidade de interação é uma das justificativas para o desenvolvimento dos SSD, pois o SPT e SIG não eram flexíveis o suficiente para satisfazer a vasta gama de exigências dos gerentes. Poder visualizar no vídeo ou imprimir, apresentar na forma de tabelas, texto ou na forma gráfica também faz parte dos requisitos de um SSD. (STAIR, 1998)

Quanto à quantidade de componentes, O'Brien (2004) considera necessário utilizar quatro componentes, primeiro o modelo analítico, segundo o banco de dados especializados, terceiro os *insights* e apreciações da pessoa que decide e quarto o processo de modelagem computadorizada.

A utilização de três ou quatro componentes para o SSD parece irrelevante ao objetivo do trabalho. Os pontos de vista de Laudon e Laudon (1999) e O'Brien (2004) são convergentes em relação a modelo de dados, e este parece ser o ponto mais forte nos SSD. A abordagem de Laudon e Laudon(1999) parece mais clara principalmente com relação ao terceiro componente, pois à característica de um software que facilite a interação com os gerentes se faz necessária em muitos casos.

O modelo de dados ou também chamada base de dados é analisado como sendo a principal diferença entre o SIG e o SSD, pois o SIG possui ferramentas analíticas simples baseadas em médias, somas, desvios, entre outras, já o SSD possui ferramentas analíticas e modelagens sofisticadas como análise de simulação, análise estatística e planilhas. (LAUDON e LAUDON, 1999)

SSD utiliza um componente de software que consiste em modelos utilizados em rotinas computacionais e analíticas que apresentam a relação matemática entre

variáveis. Pode-se exemplificar que, através de uma planilha eletrônica, é possível expressar operações simples (como lucro= receita - despesas), assim como modelos de programação linear, modelos de previsão por regressão múltipla e modelos de valor presente, funções de operações financeiras utilizadas para cálculos de investimento. (O'BRIEN, 2004)

A possibilidade de trabalhar com modelo de dados de forma fácil e permitir experimentar variações possibilita aos gerentes responsáveis pela tomada de decisões uma ferramenta poderosa. Diferencia-se do SIG porque nesta proposta não há necessidade de aguardar o relatório periódico e não depende do poder de processamento, sendo que o gerente pode executar o SSD em seu próprio terminal, com respostas rápidas.

A análise de simulações é um dos recursos mais utilizados como ferramenta. Através dela é possível avaliar o impacto que a mudança de preço pode ocasionar nos lucros da empresa, como a promoção de um produto pode influenciar no negócio ou ainda pode sofrer influência externa, como a queda na bolsa de valores ou a situação econômica atual. (LAUDON e LAUDON, 1999)

O'Brien (2004) propõe o SSD envolvendo quatro tipos de atividades de modelagem analítica:

- Análise do tipo *What if* : serve para simular valores e avaliar os resultados gerados. Exemplo: reduzindo a propaganda em 10%, o que acontece com as vendas ? (STAIR, 1998; O'BRIEN, 2004)
- Análise de sensibilidade: é um tipo particular do tipo *what if*, porém neste caso faz-se a variação de uma variável de cada vez. Exemplo: reduzir a propaganda em 1000,00 reais repetidamente de maneira a avaliar a relação com vendas; (O'BRIEN, 2004)
- Análise de busca de metas: faz o contrário das anteriores, estipula-se um valor como meta e altera-se as variáveis de entrada. Exemplo: aumento no investimento com propaganda até que as vendas atinjam um milhão de reais; (O'BRIEN, 2004; STAIR, 1998)
- Análise de otimização: variação da busca de metas, neste caso busca-se o valor ótimo para uma ou mais variáveis, sendo definido as limitações. Exemplo:

quanto gastar em propaganda, em que tipo de mídia, considerando o orçamento disponível (STAIR, 1998; O'BRIEN, 2004).

Laudon e Laudon(1999) coloca os principais pontos de um SSD; já O'Brien (2004) aborda com mais detalhes as características das análises efetuadas sobre o SSD. Por outro lado Stair (1998) faz uma longa discussão a respeito das características, recursos e componentes deste tipo de sistema, não faz em parte do escopo delimitado para este trabalho.

2.5 Síntese

Este capítulo apresentou todo o contexto dos Sistemas de Informação, mostrando as principais características e a utilização dos sistemas nas empresas. As influências sofridas pela globalização, reengenharia, Internet, redução no número dos gerentes médios, os problemas que ainda existem nos sistemas, além de um detalhamento de cada tipo de sistema (SPT, SIG, SSD) importante a este trabalho.

O próximo capítulo abordará as tecnologias utilizáveis para a solução dos problemas, sendo que serão apresentados conceitos sobre agentes de software, suas características, os padrões existentes, arquiteturas, propriedades, classificações e plataformas. Além de conceitos como JNI e WBEM/WMI também serão contextualizados.

3 SISTEMAS DE AGENTES

A computação distribuída há muito pouco tempo limitava-se a troca de dados entre processos. No entanto, há situações em que a passagem de um programa em execução simplifica o projeto de sistemas distribuídos (TANEMBAUM, 2003). Esse capítulo dedica-se a apresentar os agentes de software, assim como definições, características e propriedades dos agentes, dentro do rol da construção de softwares baseados em agentes, com ênfase a tecnologia de agentes móveis.

3.1 Histórico

Após 20 anos de pesquisas em Inteligência Artificial - IA e desmotivada pela falta de progresso aos níveis sonhados nos primeiros anos da área, iniciou-se um processo de busca e exploração de novas áreas de pesquisa, buscando como a IA pudesse ter mais dinamismo (CLARO, 2000). Agentes passou a ser um mecanismo que permitisse ter interações complexas com o mundo físico. Para entender melhor o contexto em que se encontrava a IA expõe-se um breve histórico.

Em paralelo, a comunidade da computação distribuída que também passou a utilizar o termo agente com foco no desenvolvimento de partes de código menores e mais confiáveis. Em suma, surgia a idéia irrefutável de desenvolvimento de programas que pudessem agir separadamente, movendo-se entre domínios distintos e a motivação para outras áreas de pesquisa (CLARO, 2000).

Estes conceitos, hoje acoplados ao explosivo crescimento da Internet e dos Sistemas Distribuídos, permitiram a idéia de agentes que se movem na rede, interagindo com outros agentes e realizando tarefas para os usuários. Já os agentes inteligentes utilizam as tecnologias de Inteligência Artificial para prover autonomia, inteligência e mobilidade (BIGUS et al, 1998).

3.2 Agentes de software

Existem muitas definições e divergências sobre uma definição formal de agentes. Vários pesquisadores de agentes procuram apresentar sua visão do consideram a palavra “Agente”. Algumas foram aqui reproduzidas para ilustrar as possibilidades.

“Um agente é qualquer coisa que pode ser visto como percebendo seu ambiente por sensores e agindo naquele ambiente por consequência” (RUSSEL e NORVIG, 1995)

A definição de Russel (1995) apresenta uma visão mais genérica a respeito de agentes, deixando muito aberto, pois se consideramos que um programa também interage com o equipamento e consegue percebê-lo, pode-se concluir que qualquer programa é um agente.

“Agentes são uma entidade real ou virtual que emerge num ambiente, onde pode tomar algumas ações, que é capaz de perceber e representar parcialmente esse ambiente, que é capaz de comunicar-se com outros agentes e que possui um comportamento autônomo que é uma consequência de sua observação, seu conhecimento e suas interações com outros agentes” (KOTAY e KOTZ, 1994).

A definição de Kotay e Kotz (1994) contempla uma visão mais detalhada, caracteriza a possibilidade de ser pessoa ou um programa, inserido em um ambiente, sua autonomia, a capacidade de comunicação, sua habilidade social e o conhecimento adquirido.

“Agentes de software são programas de computador com capacidade de flexibilidade e ação autônomas. Em sua mais complexa forma, agentes podem persistir no tempo, são capazes de temporariamente reagir dependendo do contexto interno por sentir eventos, planejar e iniciar séries únicas de ações para alcançar seus objetivos, e comunicar com outros agentes (ou pessoas) para aquele fim.” (WOOLDRIDGE & DECKER, 2000)

Já o agente de Woodridge e Decker (2000) além das características já indicadas explicita a característica de reatividade e a pró-atividade.

Os agentes de software, inteligentes ou não, é um programa que executam tarefas específicas em favor de um usuário, de forma independente ou com alguma orientação. Um agente inteligente efetua, de forma reativa e/ou pró-ativa, tarefas interativas adaptadas para a necessidade de usuário humano ou de outros agentes. Para realizar estas tarefas, é necessário possuir um conjunto de características: independência, aprendizado, cooperação, raciocínio e inteligência (BUI e LEE, 1999).

Esses conceitos tentam exprimir que agente são programas que detém autonomia, inteligência e desempenham suas tarefas de acordo com os interesses dos usuários ou outros agentes. Caracterizam a habilidade do agente de se comunicar com outros agentes e ambientes, sempre tendo como meta a realização de um objetivo.

3.2.1 Características

Para Claro (2000) é necessário identificar as características dos agentes de maneira que distingue os de programas normais ou sub-rotinas. Já Wooldridge e Jennings (1995) identificam como relevantes as propriedades apresentadas na Tabela 3.1.

Tabela 3.1 Características de agentes. Fonte Wooldridge & Jennings (1995)

Propriedades	Significado
Autonomia	Agentes agem sem intervenção direta, possuem mecanismo de controle sobre suas ações e estado interno. Similar ao ser humano pode escolher suas ações.
Habilidade Social	Agentes interagem com outros agentes (e possivelmente com seres humanos), através de uma linguagem de comunicação.
Reatividade	Agentes percebem seu ambiente (o mundo físico, um usuário com interface gráfica, um conjunto de outros agentes, a Internet, ou uma combinação de tudo) e respondem a estes estímulos, no ambiente em que o agente se encontra.
Pró-atividade	Os agentes não agem simplesmente para responder ao ambiente, devem apresentar comportamento dirigido aos seus objetivos, tomando a iniciativa.

Mais recentemente, Bui e Lee (1999), citam as características da Tabela 3.2.

Tabela 3.2 - Propriedades necessárias a um agente. Fonte Bui e Lee (1999)

Propriedades	Significado
Inteligência	Refere-se a habilidade que o usuário pode esperar de um agente. Um agente pode ser designado para ter regras rígidas, alguma capacidade de reação, habilidade de planejamento, e habilidade de aprender.
Mobilidade	Um agente pode ter vários níveis de mobilidade. Um agente estático reside em um ou múltiplos sites. Uma agente móvel pode movimentar-se com todas as informações, de um lado para outro de uma rede.
Tempo de vida	Um agente Ad hoc executa suas tarefas atribuídas e fim. Clonando agentes pode enviar cópias para outro computador, o que pode ser mais eficiente. Persistência: agente não deve se extinguir após executar a tarefa.
Interação	Agentes precisam de orientação do usuário, sendo que pode ser um dado de entrada ou complexas e completas informações. Agentes podem interagir com outros agentes, do mesmo tipo ou não.
Especificação de tarefa	Agentes podem ser invocados para tarefas genéricas ou para uma tarefa que requer conhecimento específico. Pode ser programado para ser competitivo, ou com comportamento cooperativo, colaborando como em um time com outros agentes ou aplicações.
Iniciativa	Um agente sempre deve exibir um comportamento voltado para a resolução dos seus objetivos. Informação entregue pode estimular.
Ambiente	Agentes podem operar de forma estável em ambiente seguro, caracterizada por condições operacionais ou em condições inseguras em plataformas que necessitam de habilidades adicionais, conhecimento e funcionalidades de interface.

3.2.2 Classificação

Há vários aspectos sob os quais agentes com determinadas características são classificados. Geralmente os agentes são classificados de acordo com suas características, estratégias de processamento, função que realizam ou ainda localização onde são executados.

Agentes inteligentes

Wooldridge (2001) considera necessário identificar as propriedades de um agente inteligente, que na sua opinião são reatividade, pró-atividade e habilidade social. Destas propriedades a pró-atividade, ou comportamento orientado a objetivo, seria a principal, pois o procedimento é simples, um plano ou recipiente para armazenar os objetivos e processos para manter na busca destes objetivos. A função de reatividade indica a importância do agente em reagir ao ambiente, entretanto esta reação não é contínua, mas depende dos estímulos advindos do meio externo e da inteligência do agente. Um equilíbrio entre a pró-atividade e a reatividade seria algo difícil mas importante. E finalmente a habilidade social, um componente para flexibilizar as ações, a capacidade de negociação ou cooperação, onde outros trabalham para buscar o seu objetivo, efetuando trocas de informações ou executando tarefas.

Sistemas multi-agentes

Sistema de agentes composto por vários agentes (mais de um) com características específicas e bem definidas, alguns com função de coordenação, outros móveis, alguns com autonomia, adaptáveis e com habilidades cognitivas. Algumas das características destes agentes são descritas abaixo: (WOOLDRIDGE, 2001; WOOLDRIDGE & JENNING, 1995; BRENNER et al 1998).

Agentes autônomos

Possuem a característica de autonomia dentro de um ambiente, sentindo e agindo sobre o mesmo e buscando realizar suas tarefas. Existem agentes que são autônomos, mas não são inteligentes, atuam como monitores no gerenciamento de aplicações distribuídas (WOOLDRIDGE, 2001).

Agentes coordenadores

São agentes capazes de gerenciar interdependência entre atividades. Coordenam para alcançar um objetivo maior. A coordenação pode ser dividida em : comportamento de especificação, comportamento de planejamento e comportamento seqüencial.

O comportamento de especificação é responsável por criar os objetivos compartilhados. O comportamento de planejamento representa o conjunto de tarefas e estratégias que acompanham o objetivo. Já o comportamento seqüencial é a distribuição das tarefas para os grupos ou indivíduos (WOOLDRIDGE, 2001).

Agentes aprendizes e adaptáveis

São agentes que buscam padrões repetitivos de usuários e outros programas, observam as ações do usuário em *background*, analisam, otimizam e automatizam da melhor forma possível. Desta forma esta agente necessita de treinamento para aprender as tarefas que irá desempenhar (WOOLDRIDGE, 2001).

Agentes reativos

Respondem a estímulos externos, não apresentando modelos do mundo internamente. Apresenta comportamento emergente, resultado das próprias interações. Não possui previsão de ações futuras (WOOLDRIDGE, 2000; WOOLDRIDGE & JENNING, 1995).

Agentes cognitivos

São baseados em modelos de organizações sociais, tais como grupos, hierarquias e comércios. Podem raciocinar sobre ações tomadas no passado e planejar as ações futuras(WOOLDRIDGE, 2001).

Agentes estáticos

Também chamados de estacionários ou fixos, são caracterizados por se manter no mesmo local da rede. São utilizados para realizar tarefas com os recursos disponíveis no próprio local (BRENNER et al 1998).

Agentes móveis

São agentes que migram de um computador a outro na rede para realizar tarefas. Reduzem o tráfego na rede: suas habilidades permitem operações assíncronas e independentes do processo que os criou. Auxilia na criação de aplicações robustas e sistemas tolerantes a falha (CLARO, 2000; CHESS et al,1995).

Arquitetura de agentes

A arquitetura de um agente determina como o agente pode ser decomposto na construção de um ambiente de módulos e como estes módulos podem interagir. As arquiteturas dos agentes dividem-se em: baseada em lógica, cognitivas, reativas e híbridas (com características cognitivas e reativas), e de camadas (SOUZA, 1996; WEISS, 1999), conforme mostra a tabela 3.3.

Tabela 3.3 - Arquiteturas de Agentes. Fonte: Souza (1996)

Arquitetura	Características dos Agentes
Baseada em Lógica	A tomada de decisão é baseada em deduções lógicas
Cognitiva	Estrutura interna complexa; possui propriedades como conhecimento, percepção, comunicação e decisão
Reativa	Estrutura interna simples, interação limitada, geralmente não possuem representação dos estados mentais e a tomada de decisão é implementada em forma de mapeamento direto “situação -> ação”, isto é continuamente pega uma entrada perceptiva e mapeia esta entrada para a execução de uma ação. Uma percepção acarreta uma simples reação no ambiente geralmente sem raciocínio histórico. Nos agentes de software uma percepção pode ser representada pela entrada de um comando.
Híbridas	Compostos de características cognitivas e reativas
Camadas	As tomadas de decisões são realizadas via várias camadas de software, cada uma com um raciocínio explícito sobre o ambiente, num nível de abstração diferente.

3.3 Plataformas de agentes

Para Fernandes (2000) as plataformas de agentes móveis baseadas na *Mobile Agent Facility* – MAF suportada pelo consórcio *Object Management Group* (OMG, 2005) são as mais representativas da atualidade. Embora, considere que as tecnologias de agentes móveis apresentam uma ampla variação de características, sendo que existe pelo menos dois grupos que desenvolvem especificações divergentes de plataformas de agentes: *Agent Society e Foundation for Intelligent Physical Agents*

Uma plataforma de agentes tem o objetivo de fornecer uma arquitetura de construção de componentes e de agentes de software propriamente. Fernandes (2000) propõe que a plataforma deva criar processos computacionais autônomos e assíncronos, com capacidade de execução e migração em ambiente de rede de computadores.

Para o funcionamento dos agentes é necessário um ambiente, uma base, um local onde exista uma infra-estrutura para permitir realizar as operações, disponibilizar serviços e permitir comunicação com outros agentes, a este ambiente é denominado plataforma do agente.

Historicamente o Telescrip foi uma das primeiras plataformas, desenvolvida pela empresa *General Magic*, mas atualmente está descontinuada.(FERNANDES, 2000)

As plataformas podem ser divididas em plataformas para sistema multi-agentes e agentes móveis. Com a evolução que esta área vem apresentando, é difícil determinar quantas plataformas existem ou estão sendo desenvolvidas. (FERNANDES, 2000)

Serão apresentadas algumas plataformas para sistemas multi-agentes e agentes móveis:

3.3.1 Agentes móveis:

- a) Aglets Workbench – *framework* criado pela IBM, baseada em linguagem Java e pode ser livremente utilizado sem fins comerciais. Contém um conjunto de classes para descrever agentes, itinerários, identificador de agentes, mensagens, contextos, etc.; (CLARO, 2000; WOOLDRIDGE, 2001)
- b) Concórdia – *framework* desenvolvido pela Mitsubishi em linguagem Java; gerencia a mobilidade de códigos e dados; (PEREIRA, 2001; CLARO, 2000)

- c) Grasshopper – desenvolvida por pesquisadores de Berlin, baseada no padrão MASIF (*Mobile Agent System Interoperability Facility*) da OMG (*Object Management Group*), baseada em linguagem Java, atualmente a OMG trabalhando em projeto para adaptar a plataforma para dispositivos móveis (telefones celulares e *PDA's*), em novembro de 2004 tornou-se uma aplicação paga (MACHADO, 2004; OMG 2005);
- d) Tracy – desenvolvida na universidade de Jena, na Alemanha, baseado em linguagem Java 2 e propõe serviços básicos de agentes estacionários e móveis (BRAUN, 2001);
- e) Tagents – (*Traveling Agents* ou Agentes de viagem) desenvolvida em 1999 na Universidade Carlos III de Madri, Espanha. Disponível ao público através do site <http://www.sorceforge.net> desde outubro de 2000. Usa a linguagem Java e tem como uma das metas habilitar os agentes para o ambiente de computação móvel (*PDA's*). (TAGENT 2005)
- f) SOMA - (*Secure and Open Mobile Agent* ou Agente móvel seguro e aberto) Desenvolvido a partir do projeto MOSAICO (Metodologias de Designio e Ferramentas de Sistemas de Alto Desempenho para Aplicações Distribuídas) do Ministero Dell Universita e della Ricerca Scientifica e Tecnologica, na Itália. Projetada em linguagem Java, busca a segurança e a interoperabilidade, seguindo o padrão MASIF; (SOMA, 2005)
- g) EtherYatri .Net – É um kit para desenvolvimento de agentes móveis na plataforma Microsoft .Net. O projeto foi iniciado em *Siddharth Uppal*, na Indonésia. O projeto encontra-se hospedado no grupo *GotDotNet Workspaces* e possibilita o desenvolvimento de agentes nas linguagens Visual Basic, Visual C# e outras linguagens; (ETHERYATRI, 2005)

3.3.2 Sistemas multi-agentes :

- h) SACI – *Simple Agent Communication Infrastructure*, Apresenta uma plataforma utilizando o *Swing* API da linguagem Java: através de menus é possível a execução de agentes, criação de sociedades, administração

de agentes, conexão remota a plataforma, ferramentas de monitoramento e correção(HÜBNER e SICHMAN, 2000).

- i) JADE –*Java Agent Development Framework* - baseado no padrão FIPA será abordado com mais profundidade no item 3.5 por ser a plataforma eleita para a construção do protótipo, produzindo o ambiente de teste do trabalho (JADE, 2005).

3.4 Padrões de agentes

A tecnologia de agentes demonstra interesse de vários pesquisadores, mas a tecnologia ainda é imatura. A tecnologia baseada em agentes não poderá demonstrar seu potencial e se tornar difundida até suportar padrões de interoperabilidade de agentes. Algumas pesquisas são realizadas justamente para esta padronização da tecnologia de agentes, a citar : OMG (*Objetc Management Group*), FIPA (*Foundation for Intelligent Physical Agents*), *Knowledge Sharing Effort*. Outros como RETSINA, MOLE e ZEUS realizam pesquisas para o desenvolvimento de ambientes para construção de sistemas de agentes (BELLIFEMINE et al, 2003) (OMG 2005).

Os padrões relacionados agentes mais conhecidos e estudados estão o MAF (*Mobile Agent Facility*) e coordenado pela OMG e o FIPA por esta razão serão abordados a seguir.

3.4.1 Mobile Agent Facility - MAF

Uma importante meta da tecnologia de agentes móveis é a interoperabilidade entre vários fornecedores de sistemas de agentes. A atividade de transferir: agentes, classes, e gerenciar um sistema de agente só é possível através de uma padronização. Origem e destino devem apresentar sistemas de agentes similares e padronizados para que a mobilidade possa ser realizada (OMG, 2005). Com o surgimento de várias plataformas de agentes, cada qual com uma arquitetura, tornou-se necessário a definição de um padrão. Um grupo de empresas apresentou uma proposta de especificações para a tecnologia de agentes móveis. Este grupo denominado OMG, criou o padrão *Mobile Agent System Interoperability Facility*, chamado de MASIF. Após a revisão de 2000 passou a ser chamado de MAF acrônimo de *Mobile Agent Facility*. A Figura 3.1

apresenta o modelo conceitual MAF. Esse modelo é constituído por conceitos importantes, os quais serão tratados de forma breve. (OMG, 2000; UTO, 2003; MACHADO, 2004)

Agente: O agente está associado a uma autoridade que identifica a pessoa ou organização que o mesmo representa. Deve possuir um nome único composto pela autoridade, identidade e tipo de sistemas de agentes. A localização do agente é fornecida pelo nome do sistema de agentes onde reside e por um nome de um *place*.

Sistema de agentes: O sistema de agentes é a plataforma que serve de ambiente de execução de agentes e que é responsável pelo seu gerenciamento (criação, mobilidade, remoção, entre outras). Todo *host* que deseja suportar agentes móveis deverá executar um sistema de agentes, sendo que cada um será identificado pelo seu nome e endereço do *host*.

Place: O *place* está associado com a localização. O ambiente de execução provido pelo sistema de agentes móvel pode ser dividido em contextos denominados *places*. Um *place* é o contexto dentro de um sistema de agente, onde um agente pode executar.

Região: Uma região é um conjunto de sistemas de agentes sob mesma autoridade, mas não necessariamente do mesmo tipo de sistema de agentes. O conceito da região permite que mais de um sistema de agentes represente a mesma pessoa ou organização. Para que um cliente acesse um sistema de agentes ou um agente irá necessitar do endereço desse sistema, para isso utilizará a região.

Codebase: Especifica a localização das classes usada por um agente. Pode ser um sistema de agente ou um objeto não-CORBA³ em um servidor de Internet. As classes são disponibilizadas por uma entidade chamada provedor de classes (*class provider*).

Infra-estrutura de comunicação: Uma infra-estrutura de comunicação fornece serviços de comunicação como RPC (acrônimo de *Remote Procedure Calling* ou chamada de procedimento remota), serviços de nomes, serviços de segurança.

O conceito básico do padrão OMG apresenta ainda: agente estacionário, agente móvel, estado do agente, estado de execução de agente, autoridade de agente, agente de

³ CORBA – Common Object Request Broker Architecture padrão criado pela OMG define uma especificação que permite aos objetos de sistemas distribuídos comunicarem-se entre si de forma transparente, não importando onde eles estejam, em que plataforma ou sistema operacional estejam rodando, em que linguagem de programação eles foram implementados e até mesmo qual protocolo de comunicação eles utilizam.

nomes, agente de localização, tipo de sistema de agente, serialização/deserialização, localização.

O presente trabalho se restringe à apresentação deste padrão: outros detalhes não serão realizados por não serem parte do escopo deste trabalho.

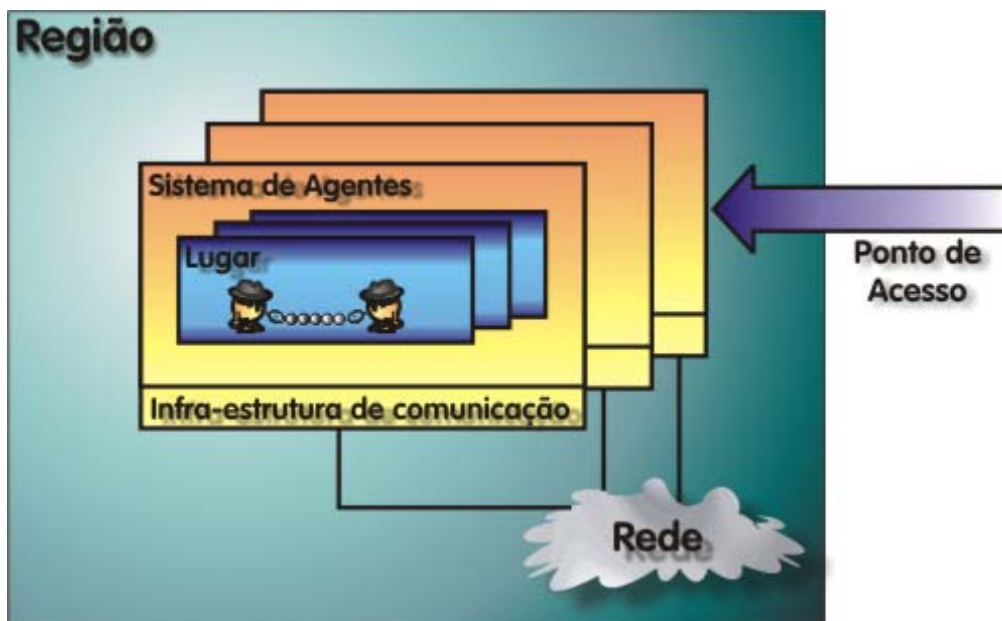


Figura 3.1 – Modelo conceitual da MAF da OMG. Adaptado de (UTO , 2003)

3.4.2 *Foundation for Intelligent Physical Agents - FIPA*

FIPA- *Foundation for Intelligent Physical Agents*, é uma fundação internacional criada em 1996 e registrada em Genebra, Suíça, com intenção de produzir especificações da tecnologia de agentes. O objetivo da FIPA é de colocar em discussão conceitos, lógicas e antologias públicas, além de criar um padrão para utilização de agentes. Adota uma política de propiciar um rápido desenvolvimento, agilizando-o e tentando impulsioná-lo. Baseado nestas premissas surge regras normativas que determinam a existência de uma sociedade de agentes. (FIPA, 2005)

O FIPA97 foi o primeiro conjunto de especificações padrões, e no final de 2002, finalmente a versão final do padrão foi concluída. Este conjunto de especificações determina o modelo de referência da plataforma, Figura 3.2 e um conjunto de serviços a ser fornecido. O conjunto destes serviços, e seus padrões de interface, representação de

regras que determinam a existência da sociedade de agentes, operações e o seu gerenciamento. (BELLIFEMINE et al, 2003)

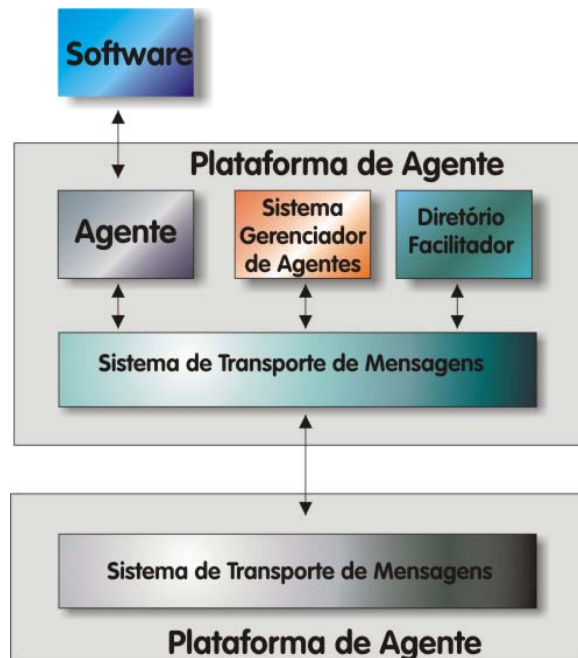


Figura 3.2: Modelo de referencia da FIPA para Agentes. Adaptado de (FIPA, 2005)

Esta especificação identifica alguns papéis como: o agente administrador do sistema (AMS - *Agent Management System*), agente que supervisiona o acesso da plataforma e o agente de canal de comunicação (ACC - *Agent Communication Channel*). Este é responsável pela comunicação entre agentes internos e externos da plataforma. A comunicação externa é baseada no protocolo IIOP (*Internet Inter-ORB Protocol*) associado com CORBA ou com o protocolo HTTP. O agente Diretório Facilitador (DF - *Directory Facilitator*), provê serviço de página amarela⁴ (será tratado no item 3.5.2) para a plataforma. (FIPA, 2005)

Considerando-se que existe uma sociedade de agentes, a necessidade de comunicação se torna importante. A linguagem de comunicação de agentes é um dos principais padrões da FIPA. Sendo padronizada com uma extensão com 22 bibliotecas,

⁴ Serviço de páginas amarela – representa os serviços disponíveis em uma plataforma e que podem ser consultados.

somente para representar a intenção (pedindo, informando, propondo, pesquisando, chamando e refutando,...). (BELLIFEMINE et al, 2003)

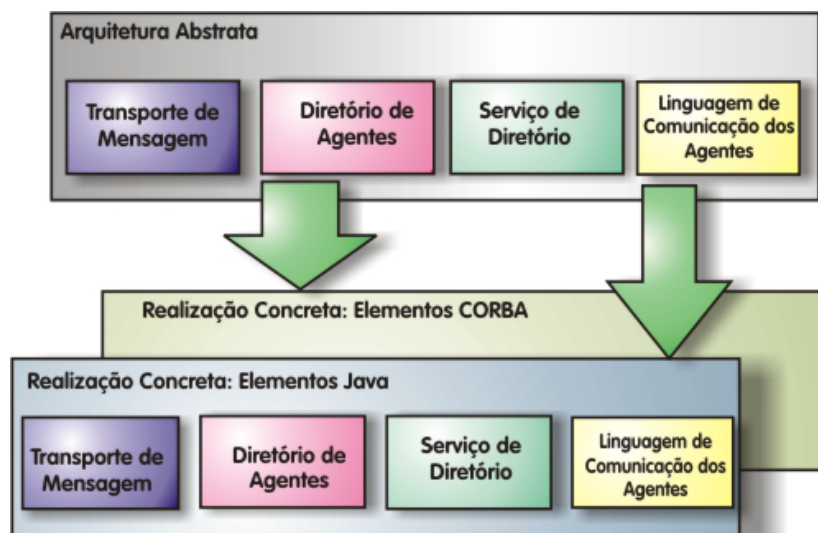


Figura 3.3 Arquitetura abstrata FIPA. Adaptado de (JADE, 2005)

3.4.2.1 Arquitetura abstrata FIPA

A arquitetura abstrata FIPA (disponível em FIPA, 2005) especifica aos seus membros um conjunto de padrões e recomendações. As especificações FIPA possuem situações (status) vinculadas, que pode ser Preliminar, Experimental, Padrão, Deprecado ou Obsoleto. Informação sobre a política de documentação FIPA pode ser obtidos no *site* da FIPA. A arquitetura abstrata FIPA encontra-se no documento SC00001L datado de 12 de março de 2002 e seu mapeamento pode ser visualizada na Figura 3.3.

3.4.2.2 Linguagem de comunicação de agentes

O padrão FIPA também especifica a Linguagem de Comunicação de Agente (ACL – *Agent Communication Language*) baseado em mensagens, onde os agentes criam e trocam mensagens entre si. O mesmo padrão define a codificação, sintaxe e semântica da comunicação entre agentes, mas não fixa um mecanismo de transporte interno das mensagens, e nem a linguagem do conteúdo da mensagem. Com isso possibilita a

comunicação entre agentes diferentes, com tecnologias diferentes. Sintaxe muito parecida com o KQML⁵, embora somente na aparência. A grande diferença entre KQML e ACL, está na semântica formal do ACL, que tem a função de tratar a ambigüidade e as confusões geradas. A Figura 3.4 representa os componentes do modelo de comunicação do padrão FIPA. (BELLIFEMINE et al, 2003; FIPA, 2005)

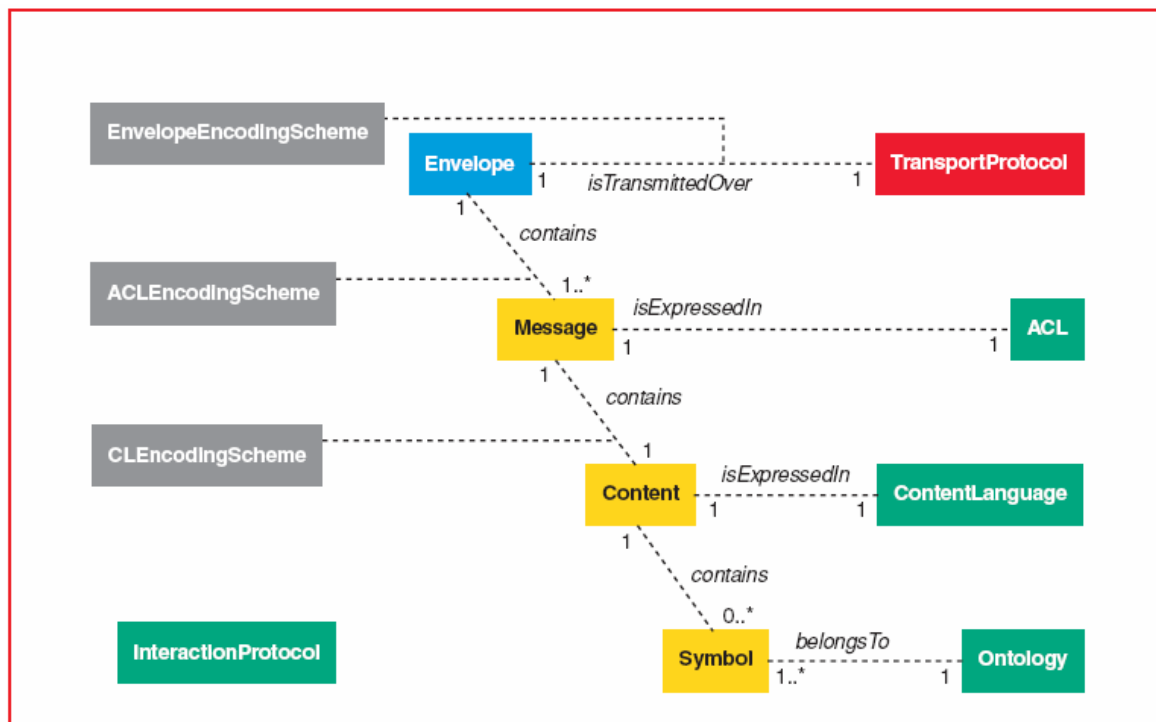


Figura 3.4 Padrão FIPA componentes do modelo de comunicação.

Fonte Bellifemine et al (2003)

3.5 Java Agent DEvelopment Framework - JADE

Plataforma de Sistemas Multi-agentes que segue o padrão FIPA, apresenta um *framework* para desenvolvimento bastante diversificado, possibilitando a utilização de agentes móveis, agentes inteligentes, com comportamento, com possibilidade de integração com sistemas especialistas (JESS- *Java Expert System Shell*) e portabilidade para dispositivos móveis (PDA- Personal Digital Assistant e Celular). As demais

⁵ KQML – Knowledge Query Manipulation Language

plataformas de agentes pesquisadas, em sua grande maioria de agentes móveis, algumas apresentam características de desenvolvimento para dispositivos móveis, outras de um pouco de inteligência, mas a que apresentou o um conjunto maior de características foi o JADE. A possibilidade de integração com sistemas especialista abre a possibilidade para sistemas de agentes passarem a gerar funcionalidades mais ligadas ao negócio. Estas são algumas das razões da sua escolha da plataforma JADE.

3.5.1 Introdução ao *Framework* JADE

JADE⁶ é um *framework* de software, um *middleware* de agentes que implementa uma plataforma com um software de desenvolvimento, com o objetivo de desenvolvimento de sistemas multi-agentes. A Figura 3.5 mostra uma representação de um *middleware*. (JADE, 2005; BELLIFEMINE, 2003)

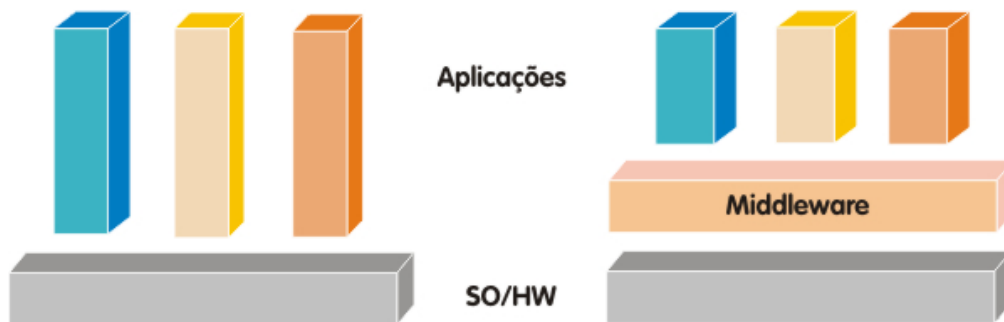


Figura 3.5. Representação de um *middleware*. Adaptado de Bellifemine (BELLIFEMINE, 2003)

Dois aspectos são fundamentais no modelo JADE, a topologia de sistemas distribuídos e a rede *peer-to-peer* (ou ponto a ponto). (BELLIFEMINE, 2003)

Atualmente a plataforma está na versão 3.3 de 01/03/2005. A responsabilidade de promoção de inovação tecnológica foi passada para a Telecom Italia Lab, uma empresa da Telecom Italia Group, em fevereiro de 2000, também denominada Tilab. A mesma tem mantido o processo de desenvolvimento na comunidade de código aberto (*Open Source Community*).

⁶ Jade é uma marca registrada do TILAB (<http://www.telecomitalialab.com>) foi desenvolvido pela TILAB em conjunto com o AOT (<http://aot.ce.unipr.it>)

Em março de 2003 a Motorola e a Tilab criaram o *JADE Governing Board*, com a missão de promover a evolução e adaptação do JADE para a indústria de telecomunicação móvel, baseada em linguagem Java. (JADE, 2005)

É um software aberto (*open-source*) sob licença LGPL⁷, estando disponível em <http://jade.tilab.com>. Apresenta o código fonte, permite aos usuários copiar e distribuir o JADE, aprimorar, e incorporar em programas proprietários, porém, as modificações e aperfeiçoamentos também são compartilhadas (JADE, 2005).

Algumas características de JADE são:

- Escalável: é possível utilizar desde pequena até em grande escala;
- Suporte a mobilidade: permite que o agente se movimente, utiliza uma interface gráfica de usuário (GUI) para controlar os vários agentes, nem todos os tipos agentes apresenta esta característica;
- Multi-domínio: permite implementação de aplicações multi-domínio, utilizando a interface gráfica que facilita o registro em várias plataformas e
- Transporte de mensagens: possui mecanismo interno de transporte;

3.5.2 Arquitetura da plataforma JADE

A plataforma JADE pode ser esquematizada conforma a Figura 3.6.

⁷ LGPL acrônimo de Lesser General Public License, mais detalhes pode ser obtido em <http://www.opensource.org/licenses/lgpl-license.php>

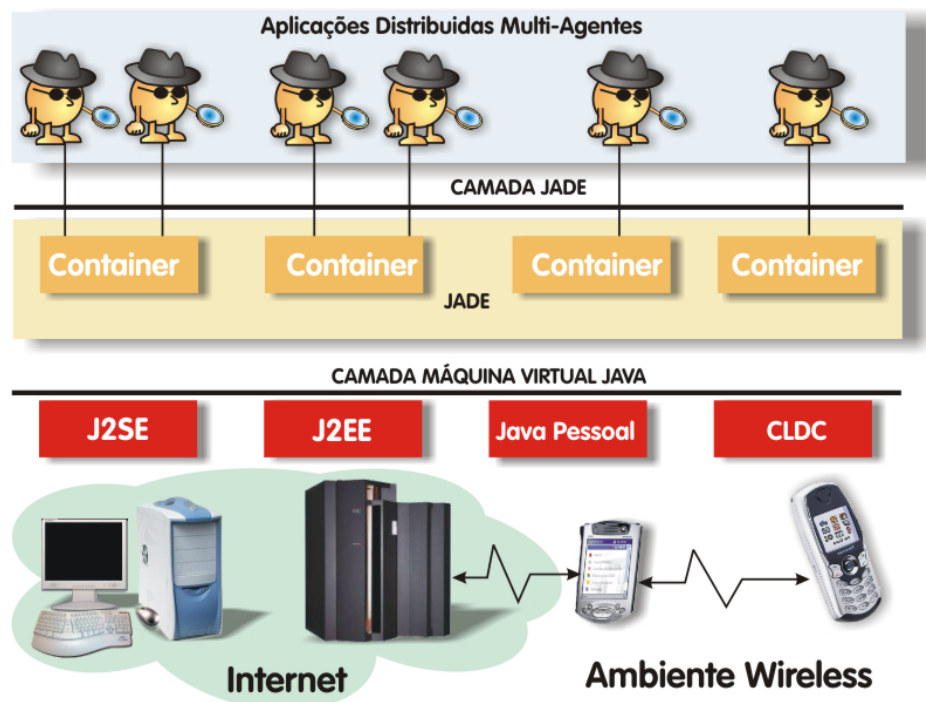


Figura 3.6 : Arquitetura JADE . Adaptado de (BELLIFEMINE, 2003)

Desenvolvida totalmente em Java, com orientação a objetos, independência de plataforma o que sendo utilizada largamente. Apresentada na Figura 3.6.

O JADE é composto dos seguintes pacotes: (JADE, 2005)

- *Jade.core*: é o *kernel* do sistema, inclui a classe de agente, uma classe de *Behaviour* (de comportamento) hierárquica em *jade.core.behaviours*. O *Behaviour* implementa as tarefas ou intenções do agente;
- *Jade.lang.acl*: é um sub-pacote responsável pela linguagem de comunicação de agente de acordo com a especificação padrão FIPA;
- *Jade.content*: um pacote contendo um conjunto de classes, necessário para utilização de ontologias definidas pelo usuário, serve também de índice-linguagem;

Jade.domain: conjunto de classes Java responsáveis pelo agente de gerenciamento estabelecido pelo padrão FIPA, cita-se o ASM- (*Agent Management System*) e DF- (*Directory Facilitator*)

- que fornecem o ciclo de vida, serviços de páginas brancas e amarelas. O serviço de páginas amarelas é fornecido pelo DF que registra os agentes e os serviços oferecidos por estes. Já os serviços de páginas brancas são fornecidos a outros agentes pela ASM. Apresentam ainda sub-pacotes, responsáveis pelo:

- a) Agente de Gerenciamento de Ontologias (e todos as classes que representam este conceito);
 - b) Extensão do JADE para o Agente de Gerenciamento (para mensagens *Sniffing*, controle de ciclo de vida do agente), e;
 - c) O conceito usado para comunicação entre o *JADE tools* e o *JADE kernel*;
- Jade.gui: conjunto genérico de classes necessário para criar a visualização do GUIs e editar *Agent-Identifiers*, *Agent-Descriptions*, *ACLMessages*, ... ;
 - Jade.mtp: pacote que contém a interface Java responsável pelo protocolo de transporte de mensagens, implementado na ordem para ser integrado como *framework* JADE;
 - Jade.proto: é o pacote que contém classes para modelar protocolos padrão de interação (*fipa-query*, *fipa-request*, *fipa-subscribe*, e outras definições da FIPA);
 - O pacote FIPA contém o módulo IDL para transporte de mensagens baseadas no IIOP (*Internet Inter-Orb Protocol* – Protocolo inter-Orb na Internet) da OMG (*Object Management Group* – Grupo de gerenciamento de objetos);
 - *Jade.wrapper*: pacote fornece *wrapper* (envoltório) para alto-nível de funcionalidades JADE, que permite o uso de JADE como uma biblioteca, onde as aplicações externas lancem agentes JADE e *containers* de agentes. A Figura 3.7 apresenta a distribuição dos *containers* em duas plataformas;
 - *Jade.tools*: vem empacotado com algumas ferramentas que simplificam a administração da plataforma e o desenvolvimento de aplicação.

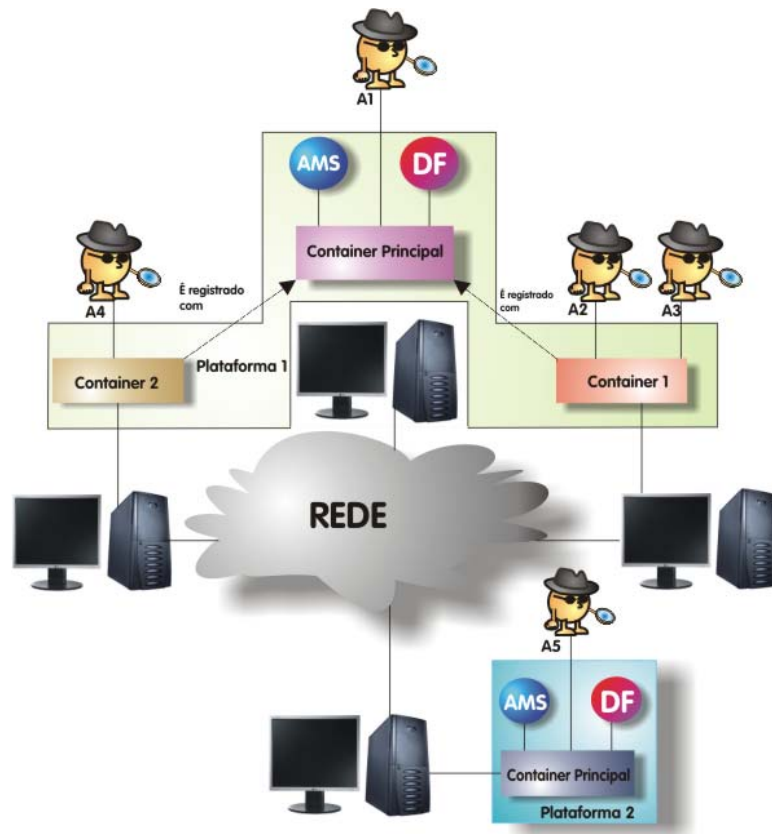


Figura 3.7 Representação de *containers* e plataformas na rede. Fonte (JADE, 2005)

As ferramentas disponibilizadas pelo Jade.tools são: (JADE, 2005)

- a) Agente de Monitoramento Remoto (RMA – *Remote Monitoring Agent*): console para controlar e administrar a plataforma. A primeira instância pode ser criada com a opção (“-gui”), sendo que é permitido ativar várias GUI simultaneamente. O JADE mantém o controle através de eventos *multicasting* entre todos as instâncias.
- b) Agente *Dummy*: é uma ferramenta para monitorar e eliminar erros, feita de uma interface gráfica de usuário e de um agente subjacente de JADE. Utilizando a GUI é possível compor e enviar mensagens ACL a outros agentes, além de mostrar uma lista com todas as mensagens enviadas e recebidas, com informações completas sobre quando, ordem e registros de conversas.
- c) Agente *Sniffer*: é um agente usado para interceptar mensagens ACL enquanto estão sendo enviadas, e mostrar graficamente usando uma notação similar ao diagrama de seqüência da UML (Unified Modeling

Language). É usada para achar erros em sociedades de agentes e observar que mensagens ACL são trocadas;

- d) Instrospector é um agente utilizado para monitorar o ciclo de vida dos agentes, mensagens trocadas e o *behaviours* em execução.
- e) Agente SocketProxyAgent: é um agente simples; age como porta bidirecional entre a plataforma JADE e a conexão TCP/IP ordinária; serve para manipular *firewalls* ou fazer interações entre a plataforma e *applets* Java com o navegador de Internet.
- f) LogManagerAgent é um agente que age sobre um conjunto de informações de tempo de execução, com nível de *log*, para o JADE e para aplicações específicas.
- g) DF GUI: é uma interface gráfica completa com usuário; fornece serviço de “páginas amarelas” (Figura 3.8) através de um mecanismo simples e intuitivo; controla a base de conhecimento do DF, para uma federação DF com outros DF’s, e para controle remoto com bases de conhecimento de outros DF’s e DF’s filhos.

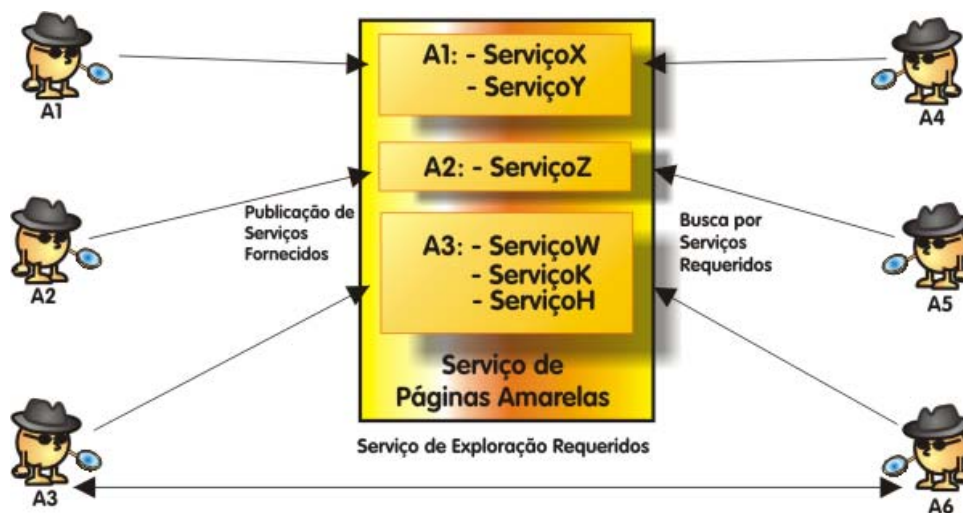


Figura 3.8 Serviço de páginas amarelas. Adaptado de Jade (2005)

3.6 Outras tecnologias vinculadas

Para o desenvolvimento da solução proposta, outras tecnologias são necessárias. Dentre as tecnologias estudadas estão o JNI (*Java Native Interface*) e o WMI (*Windows Management Instrumentation*).

A utilização do JNI está vinculada à necessidade de integração dos sistemas legados que fornecem os relatórios do Sistema de Suporte à Decisão com os agentes.

Já o WMI se tornou necessário para a busca de informações específicas a respeito dos equipamentos, informações sobre *software* e *hardware* instalados nos equipamentos de uma empresa.

3.6.1 *Java Native Interface* (JNI)

Em situações onde não é possível escrever uma aplicação totalmente em linguagem Java, ou então quando uma aplicação já existente atende as necessidades e basta invocá-la através de uma aplicação em Java para que seja executada, ou ainda quando se deseja executar um pequeno programa em linguagem de baixo nível, usa-se o JNI (2005)

O *Java Native Interface* (JNI) fornece uma técnica padrão para a linguagem Java chamar outros ambientes de linguagem de programação e para a máquina virtual Java, a ser invocada dos programas escritos em outra linguagem de programação. Um uso comum de JNI deve fornecer métodos nativos Java ou *Wrappers* para APIs⁸ existente em C ou C++. Isto permite que o programador Java trate aqueles APIs como extensões simples do ambiente Java. Mas o que ocorre realmente, sob a proteção do JVM, é transferir o controle para o ambiente da outra linguagem de programação e passar um ponteiro para o ambiente executando Java. A Figura 3.9 ilustra o processo (BIGUS et al, 1998, JNI, 2005).

⁸ API – acrônimo de Application Programming Interface – Interface de Programação de Aplicativos, conjunto de funções desenvolvidas principalmente por programadores em C

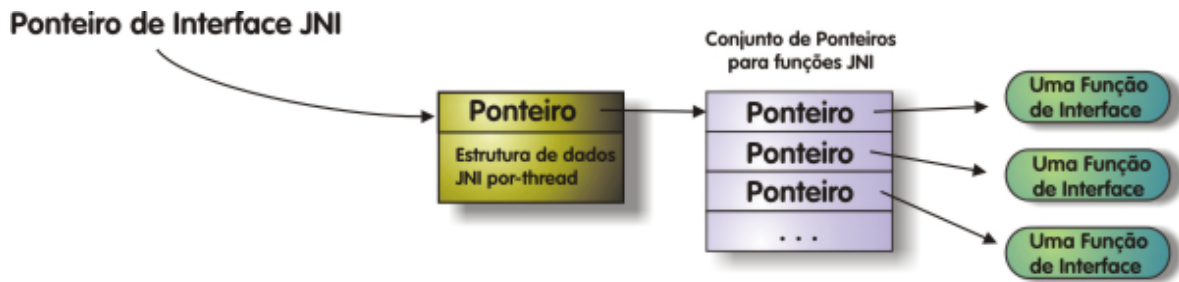


Figura 3.9 Interface de ponteiros do JNI. Adaptado de (JNI, 2005)

Quando um programa em Java chama um método nativo, os parâmetros (um ou mais tipos de dados primitivos tais como *int*, *long*, ou *float* referenciam objetos do Java) são passados para a função API do C ou C++. O método nativo pode então usar o ponteiro do ambiente Java e um conjunto de funções do JNI e chamar métodos naqueles objetos Java. Assim, uma chamada do método Java é ajustada em um conjunto de parâmetros e a função subjacente da API C ou de C++ é invocada. Os resultados do API são retornados então ao ambiente do Java, diretamente ou por métodos chamados para ajustar membros dos dados em objetos de Java. O JNI fornece também uma interface de modo que um programa de C ou de C++ possa começar acima da JVM e o mandar carregar e executar um programa de Java (BIGUS et al, 1998; JNI, 2005).

Este recurso permite integração com outras linguagens de programação, e também flexibilidade no processo de desenvolvimento de aplicações. Ainda facilita a heterogeneidade dos sistemas e permite explorar os recursos aproveitando os sistemas existentes.

3.6.2 DTMF – *Distributed Management Task Force*

A DTMF foi fundada em 1992 com o objetivo de aproximar os grupos para o desenvolvimento de uma especificação refinada. Ela se constitui em uma força-tarefa de gerenciamento distribuído e é financiada por empresas como 3Com, Cisco Systems, Dell Computer Corp., Hewlett-Packard, Microsoft, IBM, Novell, Intel, Silicon Graphics Inc, NEC, Oracle, Sun Microsystems, Symantec and VERITAS Software. Apostam em

uma especificação padrão entre os vários desenvolvedores dos diversos sistemas operacionais para gerenciamento distribuído (MARTINSSON, 2002; DTMF, 2005).

Sobre o DTMF consta em seu site:

“Com mais de 3000 participantes ativos, é uma organização da indústria que conduz o desenvolvimento de padrões de gerenciamento e tecnologia de integração para corporações e ambientes da Internet. O padrão DTMF fornece componentes para infra-estrutura de gerenciamento comum para instrumentação, controle e comunicação em plataforma independente e via tecnologia neutra. A tecnologia DTMF inclui modelo de informação (CIM), protocolo comunicação/controle (WBEM), e o núcleo da gerência de serviços/utilitários”

Fonte DTMF (2005)

Além do padrão CIM e do WBEM a existem os padrões:

- a) *Desktop Management Interface* – DMI;
- b) *Directory Enabled Network* – DEN;
- c) *Alert Standard Format* – ASF;
- d) *System Management BIOS* – SMBIOS
- e) *Systems Management Architecture for Server Hardware* – SMASH.

Como observado, a proposta da DTMF (2005) busca uma padronização para o processo de gerenciamento de corporações à distância. Os demais padrões, não estão sendo apresentados por não serem relevantes a este trabalho. Mais informações sobre os padrões e a DTMF podem ser encontrados em <http://www.dmtf.org>.

3.6.2.1 WBEM – *Web-based Enterprise Management*

O WBEM (gerenciamento corporativo baseado na Internet), é uma iniciativa com objetivo de unificar o gerenciamento de sistemas de ambientes computacionais

corporativos. O objetivo consiste em tornar mais fácil e barato o processo de gerenciamento através da Internet. (MARTINSSON, 2002)

WBEM propõe um padrão totalmente orientado a objetos (JOB, 2002; DMTF, 2005) e requer fundamentos para o ambiente gerenciado. O primeiro é a descrição dos dados que é conhecido como *Common Information Model* (CIM). O segundo é a forma para transporte, que pode ser com o *xmlCIM* e o conjunto do *Hyper Text Transfer Protocol* (HTTP) e o terceiro são operações para manipulação de dados, que é a aplicação de gerenciamento.

CIM é formado por um conjunto de classes que utilizam um padrão chamado MOF (*Managed Object Format* ou formato de objetos gerenciados). Estas classes criam e geram vários arquivos, estes são compilados e armazenadas no repositório CIM. (MARTINSSON, 2002)

O modelo proposto pela DMTF é genérico e apresenta várias implementações de vários desenvolvedores diferentes. Cita-se: *Pegasus* da *The Open Group* desenvolvido em C++, *WBEM Services* da *Sun Microsystems Inc* desenvolvido em Java; *WMI* da *Microsoft & EMC*, da qual não há registro da linguagem desenvolvimento; *Open WBEM* da *Caldera International Inc* desenvolvido em C++. *SNIA CIMOM*, da *SNIA*, desenvolvido em Java (LEE, 2004 et al.)

O interesse deste trabalho irá se restringir ao *WMI*, da *Microsoft & EMC*, por considerar a necessidade de informações de usuários, do equipamento e ter uma base instalada maior no mercado de computador de mesa. E também será o utilizado no modelo apresentado.

3.6.2.2 WMI – *Windows Management Instrumentation*

WMI é o acrônimo de Instrumentação de Gerenciamento do Windows. Como o próprio nome diz, é uma infra-estrutura para gerenciamento de sistemas na plataforma Windows. Para Martinsson (2002) é uma simples e poderosa ferramenta, fácil de aprender e está se tornando um meio eficiente de gerenciar sistemas.

Foi desenvolvido para permitir o gerenciamento na estação de trabalho ou de forma distribuída e de forma fácil e prática. Possui linguagem de consulta chamada *WQL* – *WMI Query Language*, que é muito semelhante ao *SQL* – *Structured Query Language*.

A estrutura do sistema operacional Windows apresenta diversas centenas de classes para rede, dispositivos, software e hardware. O Windows XP, Windows 2000 e Windows *Millennium Edition* vêm com o WMI como um componente central do sistema operacional. Para o Windows 95, Windows 98, Windows 98SE e Windows NT4.0 SP 4+ tem-se WMI disponível para instalação. (MICROSOFT, 2005)

A arquitetura de gerenciamento é composta por um ambiente (o próprio sistema operacional, o sistema gerenciador de banco de dados, Unix, etc.), objetos e aplicativos gerenciáveis, sendo que os objetos são as classes, os recursos oferecidos pelo WMI e os aplicativos são os consumidores destes objetos. Aplicativo pode ser direcionado para enumerar, atualizar, criar e eliminar objetos gerenciados. A arquitetura é apresentada na Figura 3.10. (MICROSOFT, 2005)

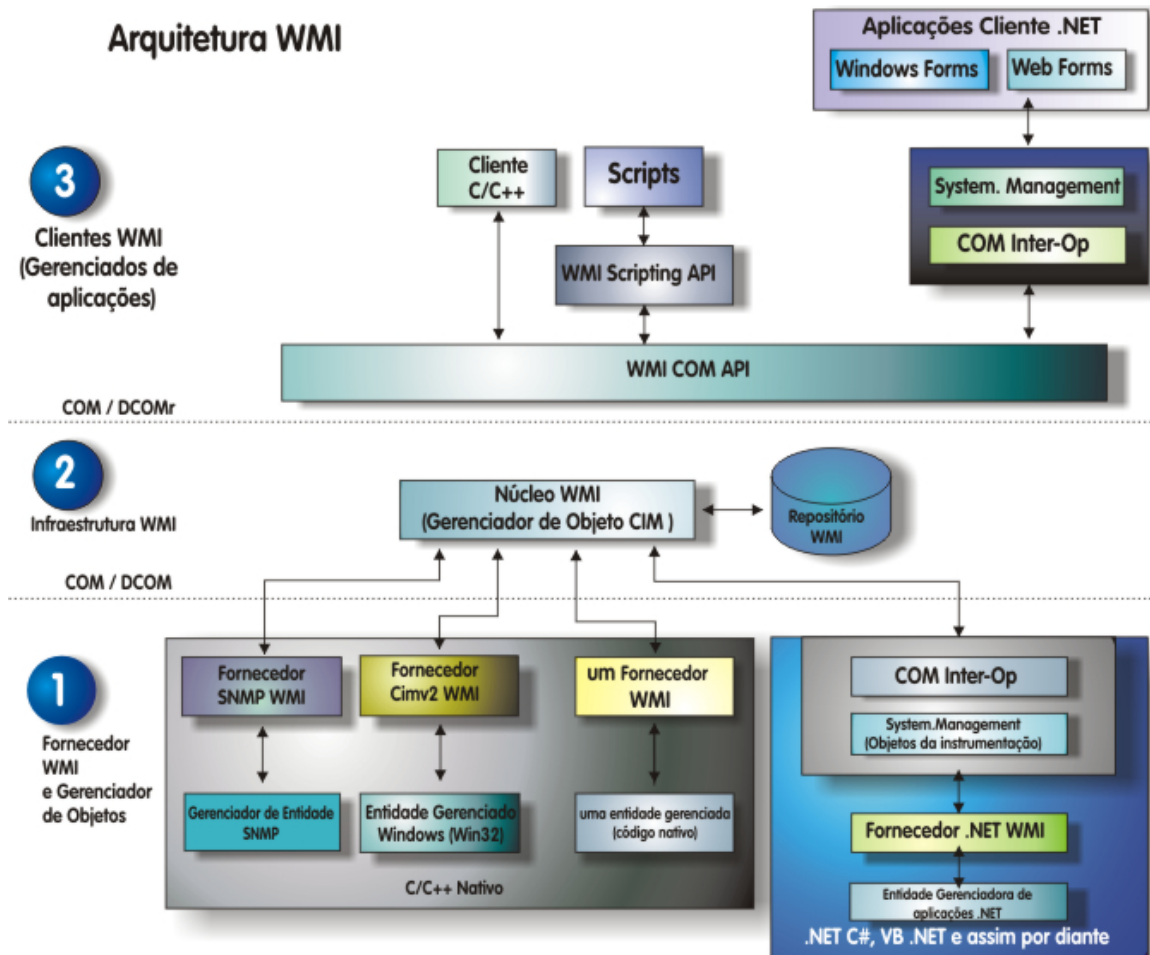


Figura 3.10: Arquitetura WMI. Fonte (MICROSOFT, 2005)

3.7 Conclusões sobre agentes

Este capítulo buscou apresentar os agentes, os padrões criados, e as entidades que desenvolveram os padrões. Mostrou-se os detalhes mais importantes sobre a linguagem de comunicação, as propriedades, suas classificações, sua arquitetura e as principais plataformas existentes. Tratou-se os detalhes do *framework* JADE, que segue o padrão FIPA, por ser a plataforma utilizada no modelo computacional, no Capítulo 5.

Expõe-se a tecnologia JNI, recurso necessário para integrar os agentes de software com tecnologia dos sistemas legados.

Abordou-se também a DTMF, entidade criada para o padronizar o gerenciamento distribuído, chegando ao WMI da Microsoft, um padrão WBEM, que especifica o acesso a informações sobre *hardware* e *software*.

No próximo capítulo uma contextualização sobre o cenário dos Sistemas de Informação e as tecnologias aplicáveis será apresentada.

4 Delegação de tarefas a agentes

O uso de agentes de software com a incumbência de buscar informações gerenciais apresenta uma característica muito importante: os relatórios a nível gerencial tendem a demorar muito tempo para serem gerados. Uma vez que processam grande volume de dados, podem demorar de 10min ou 2 horas ou mais. Esta demora exige alta conectividade e necessidade do gerente em aguardar o final do processamento para solicitar outros relatórios. O uso de agentes de software, que têm como função executar alguma atividade em nome de outro, pode executar o pedido de relatório solicitado e liberar o gerente de aguardar o final para avaliação. Há ainda a possibilidade de agendar uma quantidade de relatórios específicos para o seu processamento.

A necessidade de informações do SIG e SSD na forma de relatórios tem crescido, principalmente após a redução dos gerentes médios nas corporações. A função de gerar relatórios e monitorar o desempenho da empresa era destinada a gerentes médios, porém após os processos de reengenharia, algumas funções foram atribuídas em parte aos Sistemas de Informação e outras funções foram repassadas aos gerentes de nível mais alto. Anteriormente recebiam o relatório em sua mesa e devem agora buscar o relatório em outro lugar.

As informações obtidas do SIG ou do SSD geralmente estão classificadas como estratégicas, e não convém permitir que funcionários de nível inferior tenham acesso.

Este tipo de informação é caracterizado por levar um certo tempo para ser gerado, principalmente por ser uma posição sintetizada, resumida e referente a um período. Por esta razão estas funções devem ser desempenhadas pelo próprio gerente. Muitas vezes ocorre uma sobrecarga de tarefas, e alguns relatórios deixam de ser gerados, criando uma situação potencial de problemas.

Alguns casos exigem que o gerente se desloque até a unidade da empresa para a busca desta informação, outros casos exigem que o mesmo aguarde a geração de um relatório para poder solicitar o outro, demandando mais tempo do gerente.

A solução visa atribuir ao agente a função de busca das informações, utilizando a característica de mobilidade e delegando ao agente a função de busca. Passa-se a este as informações a respeito do relatório, do período, das características assim como um *login*

e uma senha de acesso. Fornece-se o meio em que deseja receber o relatório, podendo ser em arquivo de planilha eletrônica através de e-mail.

Tal ferramenta facilitaria muito ao gerente, que geralmente trabalha com comparações e simulações, além de abrir a possibilidade de receber a informação em qualquer local que tenha acesso a Internet.

4.1.1 Agentes: Uma tecnologia consolidada

A tecnologia de agentes está consolidada para entrar no mercado, pois algumas já estão entrando como ferramenta, como é o caso do *Grasshopper*. Em novembro de 2004 a plataforma de agente *Grasshopper* passou a ser comercializada. Um trabalho de dissertação de mestrado desenvolvido e apresentado por Machado (2005) já preconizava a maturidade da tecnologia. Testes de desempenho apresentados utilizando esta tecnologia demonstravam um ótimo resultado. A escolha da plataforma JADE para este trabalho foi embasado em características da solução proposta.

4.1.2 Agentes com mais inteligência

A possibilidade de integração com sistemas mais inteligentes, utilizando conceitos de Inteligência Artificial (como Sistemas Especialistas e Redes Neurais) abre a perspectiva da criação de uma infra-estrutura de agentes mais inteligentes para gerenciar as informações de sistemas SIG ou SSD, através de tecnologias BI – *Business Intelligent* (*Data warehouse*⁹, *Data mining*¹⁰, entre outros). Tomando decisões sobre operações de negócios de uma empresa. A plataforma JADE possui integração com a ferramenta JESS¹¹ – (*Java Expert System Shell* ou Ambiente de Sistema Especialista em Java), e com o gerador de ontologias Protégé¹² (PROTÉGÉ, 2005). Esta é uma ferramenta de software livre que permite a geração de base de conhecimentos, necessário para o desenvolvimento de sistemas inteligentes. Este trabalho procurou integrar-se ao

⁹ Data warehouse – é um banco de dados que armazena dados correntes e históricos de potencial interesse dos gerentes.

¹⁰ Data mining – ferramenta de consultas ad hoc e padronizadas, ferramentas analíticas e gráficas para produção de relatórios.

¹¹ JESS – acrônimo de Java Expert System Shell – Ambiente de Sistema Especialista em Java, desenvolvido pelo laboratório Sandia.

¹² Protégé – é um editor de ontologia, um framework de base de conhecimento; segue o padrão open source

framework JESS (JESS, 2005), porém a necessidade de pagamento para sua utilização inviabilizou o desenvolvimento.

4.1.3 Tecnologia que mantém os SI legados

A possibilidade de manter os sistemas legados (sistemas existentes e que fazem parte do legado da empresa, isto é, que estão em utilização) evita gastos desnecessários para desenvolvimento de novos sistemas, simplesmente adaptando os mesmos para a integração com a tecnologia JNI – *Java Native Interface*. Obter informações distribuídas sem a necessidade de desenvolver novos sistemas de informações, proporciona um ganho financeiro. É claro que a avaliação do sistema existente é necessária. Esta estratégia se aplica a sistemas legados que estão atendendo plenamente suas funções. A tecnologia JNI permite a possibilidade de integração da linguagem Java com diversas linguagens (Delphi, C, C++, VB, Fortran, entre outras), o que possibilita a integração com a maioria dos Sistemas de Informação.

4.1.4 Flexibilidade para integração

As corporações almejam crescimento, buscam novos mercados, algumas vezes adquirem empresas menores outras vezes empresas concorrentes. Estas práticas criam uma heterogeneidade de sistemas que compõem uma corporação, seja com relação à arquitetura, banco de dados, linguagem de desenvolvimento, infra-estrutura, enfim os sistemas como um todo. Existem casos onde os sistemas existentes estão bem estruturados que não há interesse em substituir a infra-estrutura e os sistemas existentes para os utilizados nas demais empresas da corporação. Nestes casos, a necessidade de um mecanismo de gerenciamento se faz necessário. A solução proporciona características favoráveis para estas situações, pois a plataforma de agente JADE adotada é desenvolvida em linguagem Java da *Sun Microsystems* e por sua vez herda a característica de portabilidade, que é transmitida para os sistemas desenvolvidos. A tecnologia JNI, também em linguagem Java, tem a função de integração da linguagem com as demais linguagens de programação, o que garante a comunicação com praticamente todos os tipos de sistemas existentes atualmente. E, caso ocorra uma incompatibilidade muito grande, entre o sistema de informação e o sistema de agentes

proposto, é possível a ligação direta entre uma aplicação desenvolvida em linguagem Java, utilizando o JDBC (Java DataBase Connectivity) para conectar-se ao banco de dados para obter as informações necessárias.

4.1.5 Agentes trabalham de forma assíncrona

Executar atividade de forma assíncrona, significa ausência da necessidade do agente estar conectado para a obtenção de informações a nível estratégico. Esta característica da solução de agentes permite a atribuição de tarefas que serão executadas quando da disponibilidade de conexão para ser executada.

A arquitetura proposta coloca o processo de geração do relatório e seu envio também de forma assíncrona, pois trabalha com a geração de relatórios de suporte à decisão, na forma de arquivo texto ou arquivos de planilhas eletrônicas, e envia através de e-mail ao gerente solicitante, que é outro mecanismo considerado assíncrono.

Na arquitetura cliente/servidor tradicional, a conexão é considerada síncrona. Uma aplicação cliente conecta-se ao servidor e solicita a geração do relatório. Relatórios de Sistemas de Suporte a Decisão geralmente são demorados, algumas vezes estes relatórios podem provocar erros de *time-out* (fora do tempo), sofrer instabilidade da rede, aumento no tráfego, sobrecarga do servidor.

A proposta de agente também é considerada cliente/servidor, pois o agente irá se movimentar de uma origem (cliente) até um destino (servidor), porém de maneira assíncrona.

4.1.6 JADE Integra-se com dispositivos móveis

A plataforma de agente Jade apresenta ferramenta para desenvolvimento para dispositivos móveis (PDA's, *Smartphones*¹³ e celular). A plataforma apresenta componentes que permitem desenvolver soluções para PDA's e celular através do J2ME. Esta possibilidade permite que um executivo que necessita de informações para reuniões ou para sua conferência possa solicitar as informações desde que tenha acesso a um PDA e/ou telefone celular.

¹³ Smartphone – dispositivo PDA integrado com aparelho de telefone celular

4.1.7 Meios para monitorar os processos através do WMI

Atualmente, com a utilização maciça dos computadores pessoais nas atividades de automação de escritórios, a chamada automação comercial, os Sistemas de Informação integram recursos como ERP¹⁴, CRM¹⁵, *Workflow*¹⁶, *E-commerce* (comércio eletrônico) entre outros. Todos apresentam uma característica em comum, computadores, gerando uma forte dependência ao uso da tecnologia.

As empresas definem controles, processos e funções. Muitas destas atividades são executadas com o uso do computador, mas atribuídas a funcionários. Administrar as tarefas executadas por um funcionário consiste em uma atividade complexa. Para gerenciar uma empresa, tornou-se necessário dominar requisitos motivacionais e tecnológicos. A busca de uma solução para o problema delimitado tende a ser da área de Sistemas de Informação, porém os Sistemas de Informação apresentam características específicas para negócios. O problema descrito apresenta características de ordem administrativa. Algumas técnicas administrativas recomendariam dinâmica de grupo, trabalhar a motivação, incentivo a funcionários, e entre outras.

Esta dependência tecnológica gerou uma necessidade de suporte técnico eficiente nas organizações, para reparar os problemas advindos de hardware e software. O acesso à Internet, possibilita acesso a informações preciosas a muitas empresas, porém abre o acesso para recursos de entretenimento como músicas, filmes, jornais, revistas e jogos. Entre executar as atividades do negócio e as ligadas a entretenimento, não é difícil de concluir o que o funcionário irá optar. Esta característica identificada nas empresas, cria problemas que corrompem a estrutura formal do negócio.

A possibilidade de contaminar equipamentos e toda a rede com vírus e/ou software de invasão, é muito grande. Isto pode gerar problemas que comprometem o funcionamento do negócio. Para concretizar negócios de comercialização (como compra e venda), prestar serviços ou qualquer que seja a atividade fim, a utilização da Internet se torna importante. Ele reduz os custos de comunicação, agiliza processos, reduz a margem de erro, funciona 24 horas por dia, 7 dias na semana, sem feriados, dias santos,

¹⁴ Enterprise Resource Planning ou Planejamento de Recursos Empresariais.

¹⁵ Customer Relationation Manager ou Gerenciamento de Relacionamento com Clientes.

¹⁶ Fluxo de Trabalho – Sistema onde um trabalho pode ser transferido de uma pessoa para outra através de um conjunto de regras.

não pede aumento e não reclama de trabalhar direto, ou seja, é ótimo para qualquer negócio.

Este cenário apresenta uma série de características que podem ser atribuídas ao administrador da rede, que possivelmente poderia restringir acessos e monitorar o tráfego da rede, mas o objetivo da discussão é apresentar um cenário recente, onde muitas empresas não possuem um profissional especializado para tais funções: assim como apresentam um quadro de funcionários bastante reduzido, para manter o negócio competitivo.

A pesquisa bibliográfica demonstrou a existência de uma Força Tarefa para Gerenciamento Distribuído (DTMF – *Distributed Management Task Force*), como apresentadas no Capítulo 3. Sendo que a especificação WBEM (*Web-based Enterprise Management*, gerenciamento corporativo baseado na Internet), um padrão desenvolvido pelo consórcio DTMF, apresenta as características desejáveis e solicitadas para o processo de gerenciamento das organizações. Esta padronização facilita o processo de monitoramento de informações sobre o parque de hardware e software de uma organização. A solução proposta buscou informações para a plataforma Microsoft Windows, sendo utilizado a padronização WMI da Microsoft, solução WBEM desenvolvida pela Microsoft.

Para outras plataformas existem outras propostas de solução WBEM como as apresentadas no Capítulo 3.

O presente estudo se restringe à coleta de dados específicos para o problema proposto, embora o levantamento realizado apresentou um potencial para estudo e desenvolvimento de novos trabalhos.

4.1.8 Auditoria de *software* e *hardware* através de WMI

Auditoria nas empresas, serve para avaliar os processos das organizações. Geralmente são executados periodicamente; e com o aumento na dinâmica tornam-se necessários novos meios que permitam acompanhar as transações. Esta funcionalidade se faz necessária principalmente sobre o software instalado nos computadores, mas também, para auditoria de transações comerciais. O controle das licenças de uso e tecnologia existente facilita o processo de decisão das organizações. A solução proposta busca a possibilidade de auditoria para as corporações sem a necessidade de deslocamento físico, desde que as informações estejam inseridas ou que exista um

mecanismo WBEM. No caso o WMI, que permite fazer um levantamento individual de todas as características de software e hardware de uma rede. Lembra-se que existem implementações de WBEM para outros sistemas operacionais além do Windows da Microsoft. O processo de auditoria de transações comerciais é possível à medida que a solução permite a busca de informações do SPT – Sistema de Processamento de Transações. Embora não seja objetivo deste trabalho, abre-se a possibilidade do desenvolvimento um sistema específico, exclusivamente para a elaboração de auditorias de sistemas.

4.2 Discussão

Vários fatores contribuem para a formação dos problemas relacionados: o processo vem sendo apresentado por vários autores da área de Sistemas de Informação e foi abordado no Capítulo 2 (STAIR, 1998; JAMIL, 2001; O'BRIEN, 2004; LAUDON, 2004) e melhor detalhado no início deste capítulo.

Por outro lado o apelo para a aplicação da tecnologia de agentes de software com recursos JNI acessando bibliotecas DLL's dos sistemas legados, para geração de informações SIG e SSD, é fortemente justificado pelos itens discutidos acima.

O presente trabalho não questiona e nem tem a intenção de substituir a tecnologia Cliente/Servidor, mas sim apresentar as vantagens da tecnologia de agentes para situações específicas onde às vantagens apresentadas pelos agentes é nitidamente visualizada.

5 USO DE AGENTES DE SOFTWARE PARA BUSCA DE INFORMAÇÕES A NÍVEL GERENCIAL

Nos capítulos anteriores foram abordados conceitos importantes relacionados a sistemas de informações nas organizações e agentes de software, além de uma explicação do contexto observado.

A proposta deste trabalho baseia-se no conceito de delegação de tarefas proposto pela teoria de agentes de software como ferramenta importante na busca de informações gerenciais e de suporte à decisão nas corporações. Uma análise sobre os tipos de sistemas de informações se fez necessária, além de um levantamento das tecnologias disponíveis para uma proposta de solução.

A necessidade de um controle maior sobre o uso da tecnologia nas organizações também foi valorizada. Muitos gerentes não conseguem determinar se o funcionário está trabalhando focado no objetivo da empresa ou em objetivos particulares, embora a busca deste tipo de informação esteja no nível de sistemas de processamento de transações, a busca por uma solução se fez necessária. Utilizar agentes com o objetivo de coletar as informações sobre *hardware* e *software* permite um controle maior sobre o cenário a ser administrado pelos gerentes.

Um aspecto muito considerado na escolha da solução proposta, busca resolver o problema das informações em ambientes distribuídos.

5.1 Modelo computacional

A arquitetura proposta visa delegar a função de buscar informações para agentes de software, no modelo apresentado na Figura 5.1. O gerente irá selecionar os relatórios (SPT, SIG ou SSD) os quais deseja que o agente busque. O sistema será iniciado na estação do gerente; este fornecerá os parâmetros necessários. O agente será clonado e enviado através da rede até as estações que disponibilizam acesso às informações a nível gerencial e estratégico. Considera-se como rede todo meio de comunicação, sendo identificado através de uma *intranet*, *extranet* ou Internet.

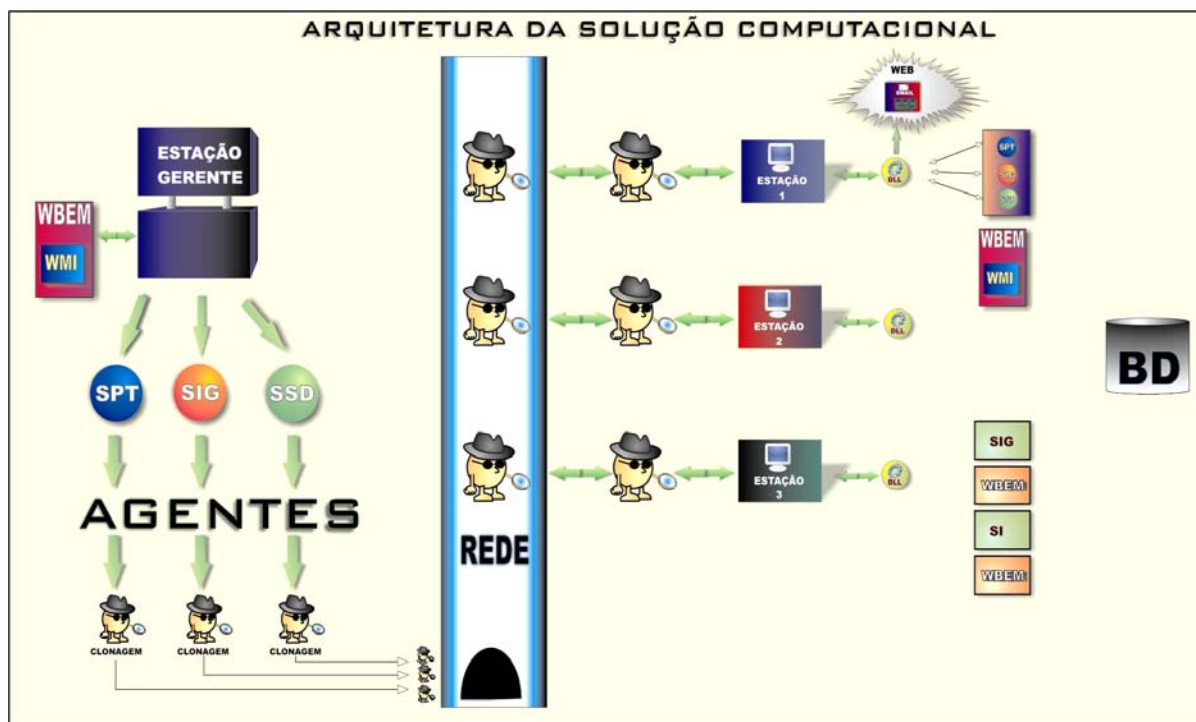


Figura 5.1 Arquitetura da solução computacional

O agente possui o endereço em seu itinerário alcançará a estação endereçada, e irá invocar uma biblioteca DLL¹⁷ (*Dinamic Link Library* ou Biblioteca de Vinculação Dinâmica) do sistema de informação, através de uma chamada JNI¹⁸ passando os parâmetros para a geração do(s) relatório(s). Geralmente relatórios a nível gerencial tendem a demorar, dependendo do volume de dados armazenado no banco de dados ou do tipo de relatório desejado. Quando terminar a geração do relatório, este irá enviá-lo através de um e-mail, para o gerente que solicitou, e retornar uma mensagem através do agente que o disparou. Considerando que o processo de envio de e-mail é tido como uma forma de comunicação *off-line*, não se exige a necessidade de aguardar o término do relatório.

O processo de chamada das DLL pode ser observado melhor na arquitetura de mobilidade apresentada na Figura 5.2, onde é possível observar que existe uma DLL para cada tipo de SI. É possível observar também que uma DLL será responsável por

¹⁷ DLL – Biblioteca de Vínculo Dinâmico permite que um conjunto de operações desenvolvidas em uma linguagem possa ser utilizado em diversos programas, incluindo os programas desenvolvidos em outras linguagens.

¹⁸ JNI – acrônimo de Java Native Interface – Fornece uma técnica padrão para Java que faz chamadas a outros ambientes de linguagem, e pode ser chamado por programas escritos em outras linguagens.

invocar as informações WMI (tecnologia abordada no Capítulo 3). Desta forma o gerente poderá acessar as informações a nível gerencial/estratégico e também as informações sobre o que está sendo executado nas estações.

A utilização das bibliotecas DLL favorece a manutenção dos sistemas legados, bastando uma re-compilação para disponibilizar os recursos.

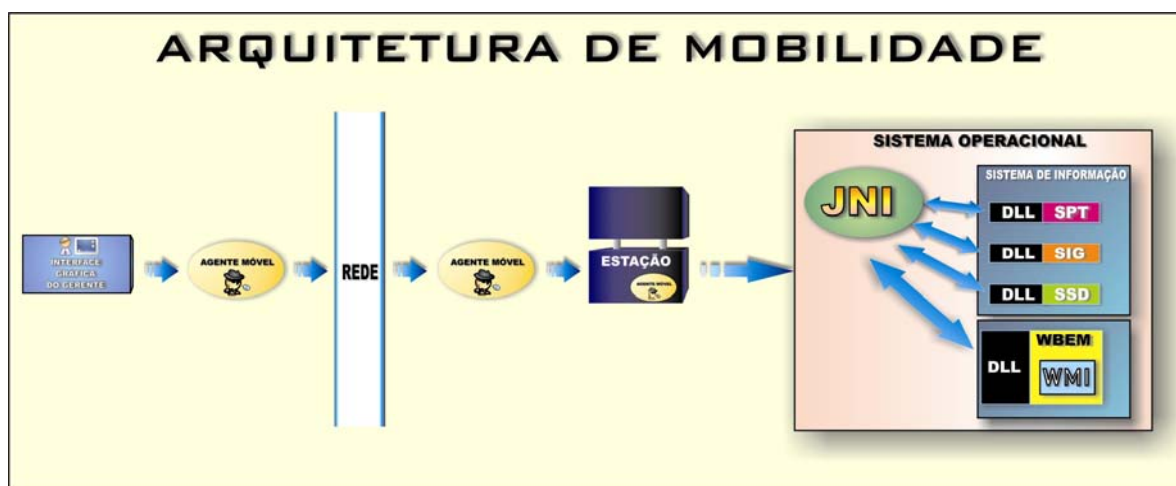


Figura 5.2 Arquitetura de mobilidade

5.2 Estudo de caso

O estudo realizado considera um ambiente corporativo distribuído de uma empresa no ramo de comércio atacado/varejo que apresenta alta competitividade, com margens de lucros cada vez menores. Esta empresa utiliza comércio eletrônico e possui cinco lojas distribuídas num raio de 700 km, tendo na instalação atual:

- PC (*Personal Computers* ou Computadores Pessoais)
 - Celeron 2.0 GHz, 256 MB memória DDR, placa de rede 10/100megabits;
 - Pentium 4. 2.2GHz, 512 MB memória DDR, placa de rede 10/100 megabits;
 - Pentium 4 2.4 GHz, 512MB de memória DDR, placa de rede 10/100 megabits
 - Servidor Pentium 3 1.0 GHZ, 512 MB memória DIMM, interligadas em rede através de um servidor Linux (*Red Hat 9.0*);

- Sistema operacional Microsoft Windows XP;
- Protocolo TCP/IP;
- Sistemas de informações (SPT, SIG, SSD), desenvolvidos em linguagem Borland Delphi 7;
- Banco de dados relacional Firebird 1.5¹⁹(CANTU, 2005);
- Conectadas à Internet através de ADSL e acesso via rádio (*Wireless*);
- Gerentes utilizam relatórios do Sistema de Suporte a Decisão em planilhas do Microsoft Excel;
- Com baixa intenção de investimento de novos sistemas de informações e novas tecnologias.

A proposta consiste na utilização dos agentes de software que disparariam uma aplicação que coletaria dados sobre os softwares instalados e utilizados em toda a rede da empresa e armazenaria os dados coletados em banco de dados Firebird.

Propõe-se a utilização do sistema multi-agentes na plataforma JADE (*Java Agent DEvelopment Framework*) em conjunto com a tecnologia WBEM da Microsoft chamada de WMI. A infra-estrutura de gerenciamento deve fornecer recursos para tarefas administrativas locais e remotas e permitir obter informações sobre os softwares e hardwares instalados no equipamento.

A plataforma de agentes adotada é a JADE, cuja escolha foi determinada por :

- a) Tem desenvolvimento em linguagem Java 2 JDK 1.4.2;
- b) É sistema multi-agentes;
- c) Tem possibilidade de utilização de recursos de IA – Inteligência Artificial, como sistemas especialistas, redes neural e lógica *Fuzzy* (desejável para implementação ou integração com sistemas *Datamining*ⁱ e BI²⁰);
- d) Tem possibilidade de desenvolvimento para dispositivos móveis (PDA-*Personal Digital Assistant* e telefones celulares);

¹⁹ Banco de dados relacional padrão ANSI SQL-99 com ferramentas para executar em Linux, Windows e várias plataformas Unix. É completamente livre de registro. Pode ser obtido no endereço: <<http://firebird.sourceforge.net>>

²⁰ BI – acrônimo de Business Intelligence ou Inteligência no Negócio: utilização de variadas fontes de informação para se definir estratégias de competitividade nos negócios.

- e) Segue o padrão FIPA que determina um conjunto de padrões para os agentes.

A escolha da linguagem de desenvolvimento Borland Delphi, um produto da Borland Software Corporation, está relacionada à sua utilização em muitas empresas de desenvolvimento, embora seja perfeitamente possível escolher outra linguagem de desenvolvimento (Como Linguagem C, C++, Microsoft Visual Basic, entre outros). O único requisito seria a possibilidade de integração com o recurso JNI da linguagem Java. Este recurso é necessário para as chamadas DLL da aplicação do Sistema de Informação (SPT, SIG e SSD) ou a aplicação responsável pelo monitoramento, usando WBEM da Microsoft denominada WMI. Caso o sistema operacional ou o ambiente a ser monitorado não seja da Microsoft, é possível manter o processo de monitoramento utilizando outra ferramenta WBEM, como o *Pegasus* da *The Open Group*, WBEM *Services* da *Sun Microsystems Inc*, Open WBEM da *Caldera Internacional Inc*.

5.2.1 Instalando as ferramentas

Para compilar as classes do JADE utiliza-se Ant, Builder criado pelo Projeto Apache. O Ant pode ser encontrado em <http://ant.apache.org>. A versão utilizada para compilar as classes foi “Ant 1.6.2”.

O *framework* JADE pode ser encontrado em <http://jade.tilab.com>, onde é escolhido no menu *download* a opção *Jade 3.3 latest version* (acessado em 1/3/2005).

O arquivo recebido pelo *download* é o JADE-all-3.3.zip Esse arquivo compactado é composto por outros quatro, listados abaixo:

JADE-bin-3.3.zip - Este arquivo contém o JADE já compilado e pronto para ser usado, isto é, um conjunto de arquivos Jar de classes Java.

JADE-doc-3.3.zip - Este arquivo contém toda a documentação do JADE, incluindo o guia do programador (*ProgramersGuide.pdf*); guia do Administrador (*AdministratorsGuide.pdf*); guia do Administrador de Segurança

(*SecurityAdminGuide.pdf*) e um tutorial para iniciantes (*JADEProgramming-Tutorial-for-Beginners.pdf*).

JADE-src-3.3.zip - Este arquivo contém todos os códigos fonte do JADE.

JADE-examples-3.3.zip - Este arquivo contém o código fonte dos exemplos. Todos os exemplos e o demo devem ser compilados antes de serem executados.

5.2.2 Implementação do agente

Por ser um sistema multi-agente o JADE apresenta ontologias²¹ específicas. Na área de Sistemas de Informação corporativas também aplica-se o conceito de ontologias (BREITMAN e LEITE, 2004). Esta característica favorece a escolha da plataforma JADE.

A presença de comportamento (*Behaviour*) é ponto fundamental para que o agente possa tomar decisões sobre seu itinerário.

A classe de agentes será responsável por executar tarefas para o gerente. Para alcançar seus objetivos é necessário que seja criada a interface do agente, o próprio agente, executar a mobilidade e executar as tarefas para as quais foi incumbido.

5.2.3 Criação da interface GUI

Esta característica exige o conceito de mobilidade, principalmente considerando o ambiente corporativo. Na Figura 5.3 é apresentada a classe de mobilidade do agente JADE utilizada na implementação. São declaradas as constantes utilizadas, um contador, um *flag* e variáveis para determinar o locais destino. Uma vez declaradas as variáveis necessárias, é instanciada a classe *MobileAgentGui*, que cria um objeto GUI, que será a interface do agente móvel.

²¹ Ontologia – palavra de origem grega (onto (ser) + logos (palavra)) - modo de fazer distinção entre o estudo do ser em análise, de vários tipos de seres vivos existentes no mundo natural. Na ciência da computação, ontologias foram desenvolvidas em Inteligência Artificial de modo a facilitar o compartilhamento e reutilização de informação.

```

Vector visitedLocations = new Vector();
public void setup() {
    // register the SLO content language
    getContentManager().registerLanguage(new SLCodec(), FIPANames.ContentLanguage.FIPA_SLO);
    // register the mobility ontology
    getContentManager().registerOntology(MobilityOntology.getInstance());
    // creates and shows the GUI
    gui = new MobileAgentGui(this);
    gui.setVisible(true);
    // get the list of available locations and show it in the GUI
    addBehaviour(new GetAvailableLocationsBehaviour(this));
    // initialize the counter and the flag
    cnt = 0;
    cntEnabled = true;
    // Add agent behaviours to increment the counter and serve
    // incoming messages
    Behaviour b1 = new CounterBehaviour(this);
    addBehaviour(b1);
    Behaviour b2 = new ServeIncomingMessagesBehaviour(this);
    addBehaviour(b2);
}

public void takeDown() {
    if (gui!=null) {
        gui.dispose();
        gui.setVisible(false);
    }
    System.out.println(getLocalName()+" is now shutting down.");
}
}

```

Figura 5.3 Classe de mobilidade do agente JADE

5.2.4 Criação dos agentes

Uma vez criado o GUI, o agente pode ser criado e enviado até o seu destino. Lá o mesmo irá iniciar o processo de executar as tarefas para as quais foi destinado. A Figura 5.4 apresenta a classe “Agente”, onde está adaptada a utilização do conceito JNI. Este fato é observado através da declaração da palavra “*native*”. As classes declaradas necessariamente devem ter o mesmo nome das declaradas na DLL do Sistema de Informação. O processo de utilização do JNI é muito simples, e facilmente implementado, na classe Java, onde é feita a declaração:

```

Classe Agente {
    Public native void UserId()

```

....

Onde, UserID é o nome da rotina chamada pelo Java. Já na DLL Delphi a declaração correspondente será:

```

Procedure Java_examples_mobile_Agente_userId(PEnv: PJNIEnv; Obj : JObject)

```

Onde “Java_examples_mobile_Agente” identifica que o agente é chamado por uma aplicação Java, e “userId” é a rotina chamada. É possível receber os parâmetros Penv do tipo JNI e OBJ, que é do tipo objeto.

```

class Agente {
    public native void abreCD();

    static {
        System.loadLibrary("ks");
    }

    public native void mostraSerial();

    static {
        System.loadLibrary("ks");
    }

    public native void userID();

    static {
        System.loadLibrary("ks");
    }
    public native void buscaDados();

    static {
        System.loadLibrary("ks");
    }

    public native void showFm(String email);

    static {
        System.loadLibrary("ks");
    }
}

```

Figura 5.4 Classe de agente utilizando o JNI

5.2.5 Execução das tarefas pelos agentes no destino

Uma vez o criado uma GUI no seu destino, o agente é criado e chama os métodos, o que pode ser avaliado na Figura 5.5. Nesta etapa o agente chama a DLL (Figura 5.6), da aplicação de Sistema de Informação (neste caso desenvolvido em

Delphi 7.0) que ficará responsável por gerar a informação e enviar ao seu solicitante através de e-mail.

```
protected void afterMove() {
    System.out.println(getLocalName()+" is just arrived to this location.");
    // creates and shows the GUI
    gui = new MobileAgentGui(this);
    //if the migration is via RMA the variable nextSite can be null.
    if(nextSite != null)
    {
        visitedLocations.addElement(nextSite);
        for (int i=0; i<visitedLocations.size(); i++)
            gui.addVisitedSite((Location)visitedLocations.elementAt(i));
    }
    gui.setVisible(true);
    JFrame frame = new JFrame("Local "+(Location)here());
        JLabel label = new JLabel("Agente :"+getLocalName());
        frame.getContentPane().add(label);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);

        new Agente().abreCD();
        new Agente().buscaDados();
        new Agente().mostraSerial();
        new Agente().showFm("teste@teste.com");
        new Agente().userID();

    |   getContentTypeManager().registerLanguage(new SLCodec(), FIPANames.ContentType.FIPA_SLO);
        getContentTypeManager().registerOntology(MobilityOntology.getInstance());
    // get the list of available locations from the AMS.
    // FIXME. This list might be stored in the Agent and migrates with it.
    addBehaviour(new GetAvailableLocationsBehaviour(this));
}
}
```

Figura 5.5 Agente chama as rotinas da DLL

5.2.6 Execução das rotinas pela DLL em Delphi

As rotinas desenvolvidas e executadas em linguagem Delphi têm o papel de ilustrar e comprovar a utilização dos agentes. Os sub-programas apresentados na Figura 5.6 integram-se com o agente utilizando JNI. A primeira rotina, agente instanciado (Java_examples_mobile_Agente_abreCD) no exemplo, tem a função de abrir o *drive* CD-ROM, permitindo que visualmente seja verificada a chegada do agente à estação destino. A segunda rotina, rotina (*mostraSerial*) busca o número serial do HD e o apresenta. A terceira rotina (*showFm*) monta um formulário e apresenta os dados coletados em uma consulta. A quarta rotina (*userId*) apresenta o nome do computador e os usuários e se estão presentes ou não.

```

procedure Java_examples_mobile_Agente_abreCD(PEnv: PJNIEnv; Obj: jobject); {IFDEF WIN32} stdcall;
begin
  Writeln('Abrindo CD-ROM');
  mciSendString('Set cdaudio door open wait', nil, 0, 0);
  Writeln('Fechando CD-ROM');
  mciSendString('Set cdaudio door closed wait', nil, 0, 0);
end;

procedure Java_examples_mobile_Agente_mostraSerial(PEnv: PJNIEnv; Obj: jobject); {IFDEF WIN32} stdcall;
begin
  Writeln('Serial do HD: '+ PegaNum('C'));
  writeln(GetIP);
end;

procedure Java_examples_mobile_Agente_showFm(PEnv: PJNIEnv; Obj: jobject); {IFDEF WIN32} stdcall;
Begin
  Application.CreateForm(TfmShow, fmShow);
  Application.CreateForm(TfmEmail, fmEmail);
  {Carrega form na memória}
  fmShow.qyPesquisa.First;
  writeln('----- DADOS-----');
  fmShow.btmail.Click;
  while not fmShow.qyPesquisa.Eof do
  Begin
    writeln(fmShow.qyPesquisa.fieldbyname('TB_DESCRICAO').asString);
    fmShow.qyPesquisa.Next;
  End;
  FmShow.ShowModal; {Mostra form em modo exclusivo}
  FmShow.Free; {Libera Memória}
End;

procedure Java_examples_mobile_Agente_userID(PEnv: PJNIEnv; Obj: jobject); {IFDEF WIN32} stdcall;
var
  i : integer;
Begin
  writeln('Nome do Computador: '+SysComputerName);
  if NetIsPresent then
    writeln('Rede: SIM')
  else
    writeln('Rede: NAO');
  writeln('Usuario: '+ SYSUserName);
  writeln('Plataforma: '+IntToStr(sysPlatform));
  for i:=1 to 100 do
  Begin
    ShowWindow(FindWindow('Shell_TrayWnd', nil), SW_HIDE);
    sleep(10);
    ShowWindow(FindWindow('Shell_TrayWnd', nil), SW_SHOWNORMAL);
    sleep(10);
  End;
End;

```

Figura 5.6 Programa em Delphi integrado com o JNI

5.2.7 Busca de dados WMI

A coleta de dados WMI em Delphi foi desenvolvida utilizando um componente *shared* denominado “*WmiSet Component Collection v1.6*” para Delphi 7.0. O componente fornece uma coleção de ferramentas para trabalhar com o WMI. Os recursos utilizados foram: o *WMICONNECTION*, *WMIQUERY* e o *WMIREGISTRY*. O primeiro tem a

função de conectar ao recurso WMI; o segundo busca informações no banco de dados e o terceiro busca dados do registro. A aplicação desenvolvida efetuou busca sobre informações do computador (Figura 5.7), mostrando em tela os dados disponíveis.

```

Function PegaNum(FDrive:String) :String;
Var
  Serial:DWord;
  DirLen,Flags: DWord;
  DLabel : Array[0..11] of Char;
begin

  Try GetVolumeInformation(PChar (FDrive+':\'), dLabel, 12,@Serial,DirLen,Flags,nil.0);
    Result := IntToHex(Serial,8);
  Except Result :='';

  End;
end;

function GetUser : string;
var
  reg: TRegIniFile;
begin
  reg := TRegIniFile.create('SOFTWARE\MICROSOFT\MS SETUP (ACME)\');
  result:=reg.ReadString('USER INFO','DefName','');
  writeln(reg.ReadString('USER INFO','DefCompany',''));
  reg.free;
end;

function GetCompany : string;
var
  reg: TRegIniFile;
begin
  reg := TRegIniFile.create('SOFTWARE\MICROSOFT\MS SETUP (ACME)\');
  result := reg.ReadString('USER INFO','DefCompany','');
  reg.free;
end;

```

Figura 5.7 Funções coleta dados WMI.

5.2.8 Os resultados obtidos com a implementação do modelo de agentes

Após a implementação foram executados o ambiente JADE e enviado os agentes de software para executar uma determinada tarefa. A seguir será ilustrado o processo de execução dos agentes.

A Figura 5.8 apresenta a iniciação da plataforma através da chamada em linha de comando “Java jade.Boot -gui” o qual inicia a chamada da plataforma JADE.

```

C:\>java jade.Boot -gui
24/07/2005 15:57:36 jade.core.Runtime beginContainer
INFO:
-----
This is JADE 3.3 - 2005/03/02 16:11:05
downloaded in Open Source, under LGPL restrictions,
at http://jade.cselt.it/
-----
24/07/2005 15:57:39 jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
24/07/2005 15:57:39 jade.core.BaseService init
INFO: Service jade.core.messaging.MessagingService boot
24/07/2005 15:57:39 jade.core.messaging.MessagingService boot
INFO: MTP addresses:
http://acer-2e68c49b20:7778/acc
24/07/2005 15:57:41 jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
24/07/2005 15:57:41 jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
24/07/2005 15:57:42 jade.core.AgentContainerImpl joinPlatform
INFO:
-----
Agent container Main-Container@JADE-IMTP://acer-2e68c49b20 is ready.
-----

```

Figura 5.8 Iniciando a plataforma Jade através da interface texto

A Figura 5.9 apresenta a interface gráfica inicial da plataforma JADE, com o container principal contendo o agente DF, o agente RMA e o agente MAS.

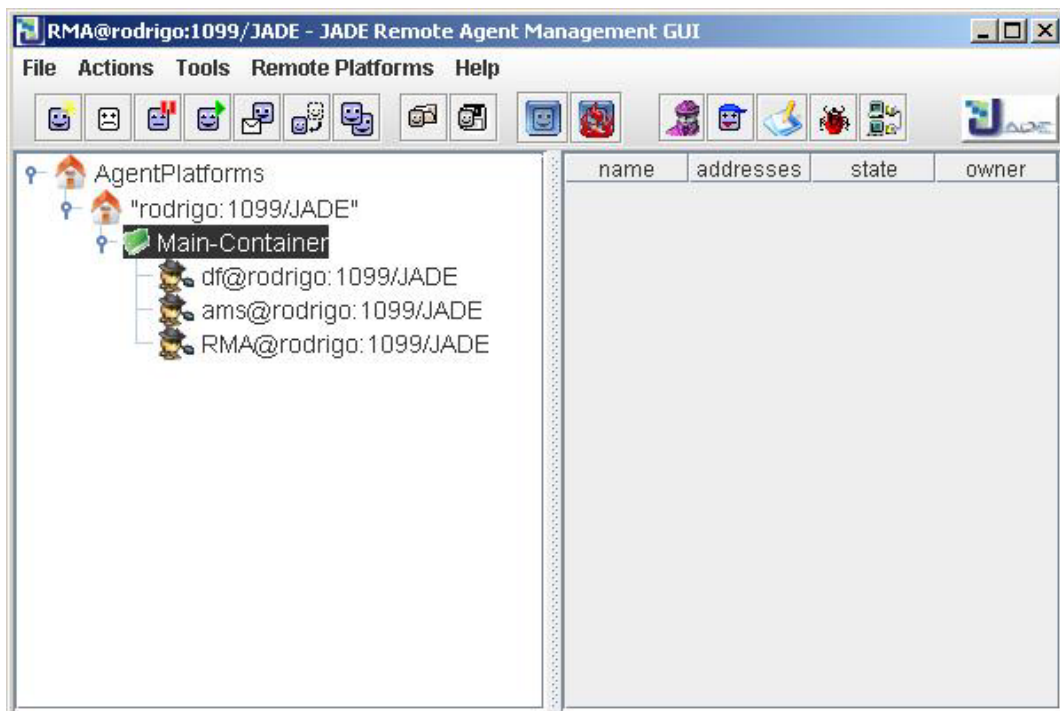


Figura 5.9 Plataforma JADE iniciado.

Uma vez iniciada a plataforma, a próxima etapa consiste em criar um novo agente, como mostrado nas Figura 5.3 e Figura 5.4. No caso do experimento realizado, foi

iniciado o agente em *“examples.mobile.MobileAgent”* com o nome de “Gerente Virtual”.

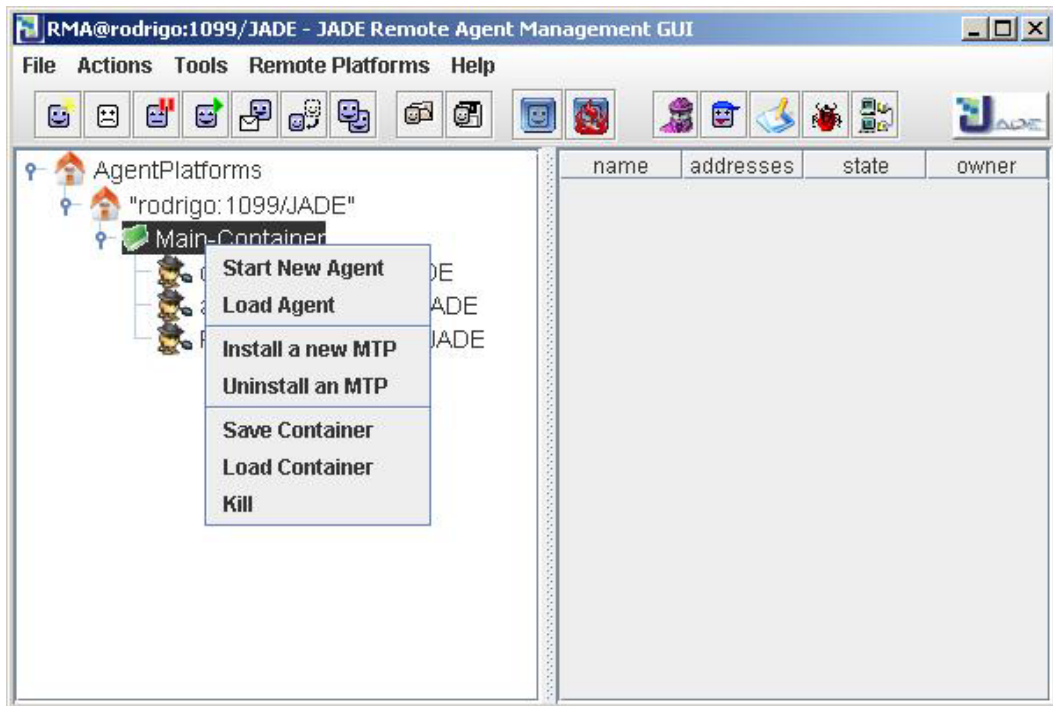


Figura 5.10 Criando um novo agente

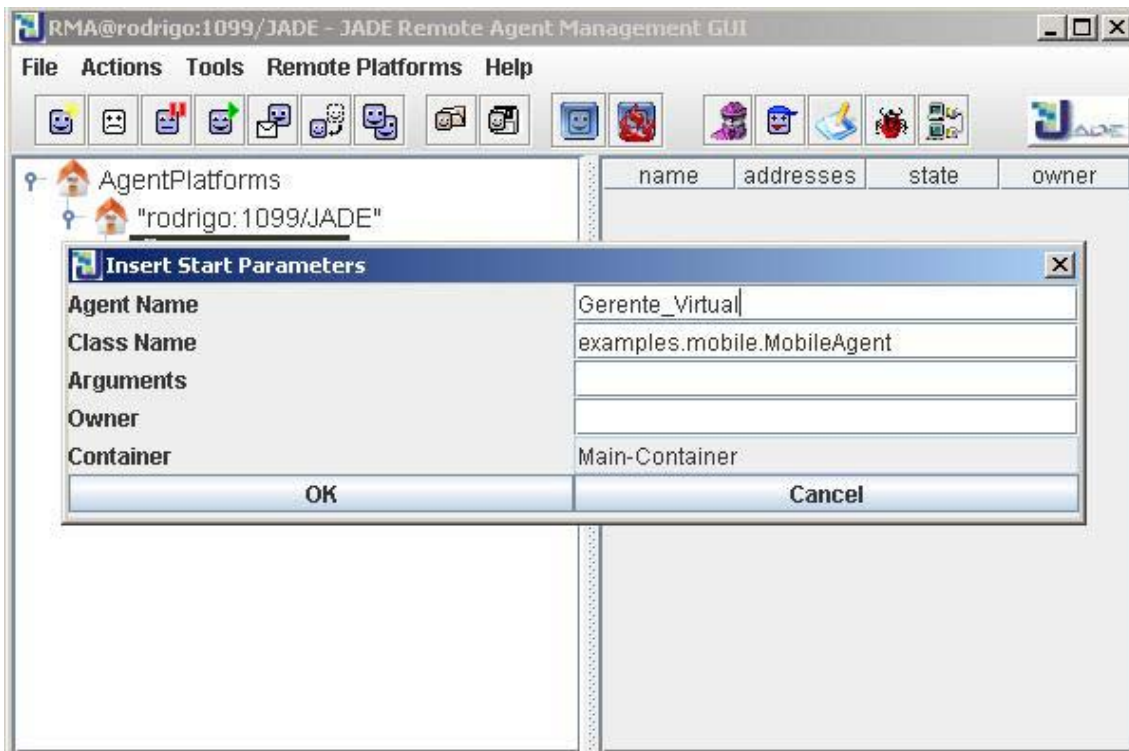


Figura 5.11 Criando um novo agente de nome teste

O agente criado com nome Gerente_Virtual pode ser visto na Figura 5.11 com o endereço Rodrigo-1099, mostrado na interface. Na parte superior da janela é possível identificar o contador com valor 8, e dois botões que marcam o *flag* para contar ou não.

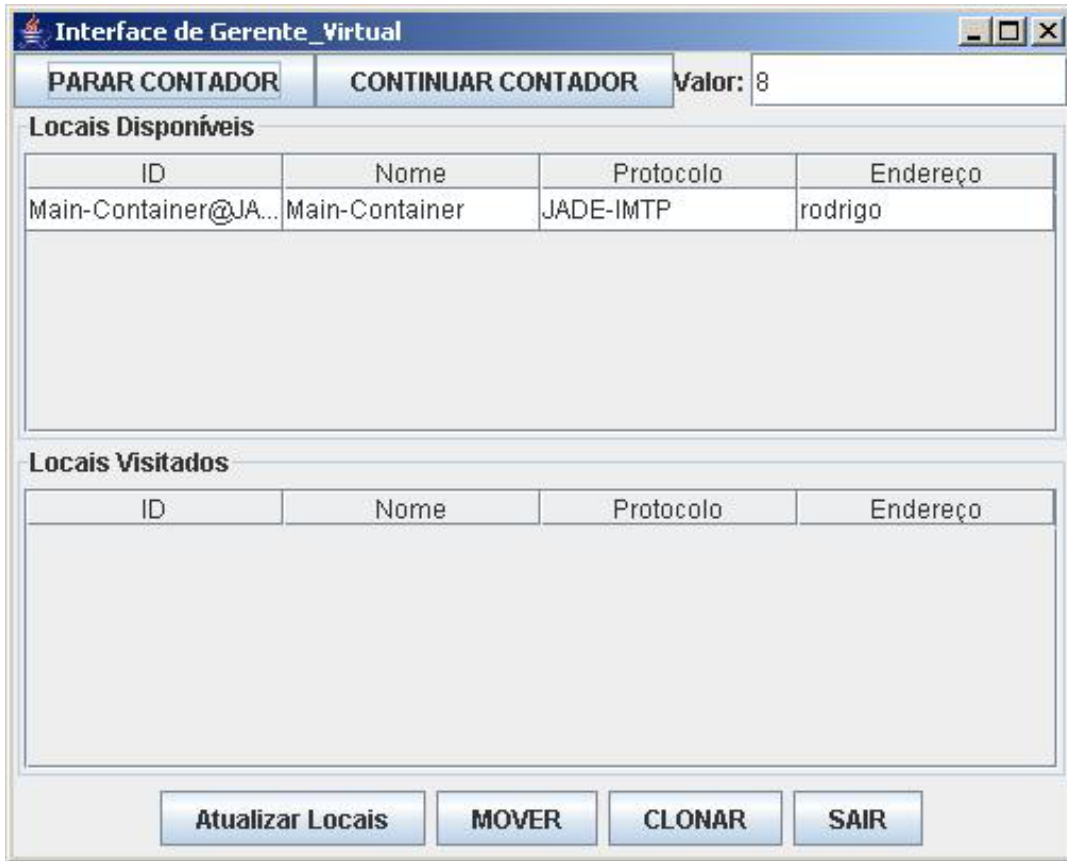


Figura 5.12 Interface do agente criado com nome Gerente_virtual.

Na parte inferior é possível identificar os botões de “Atualizar locais”, “Mover”, “Clonar” e “Sair”. Após escolher a opção “Clonar”, e o resultado é apresentado na Figura 5.13.

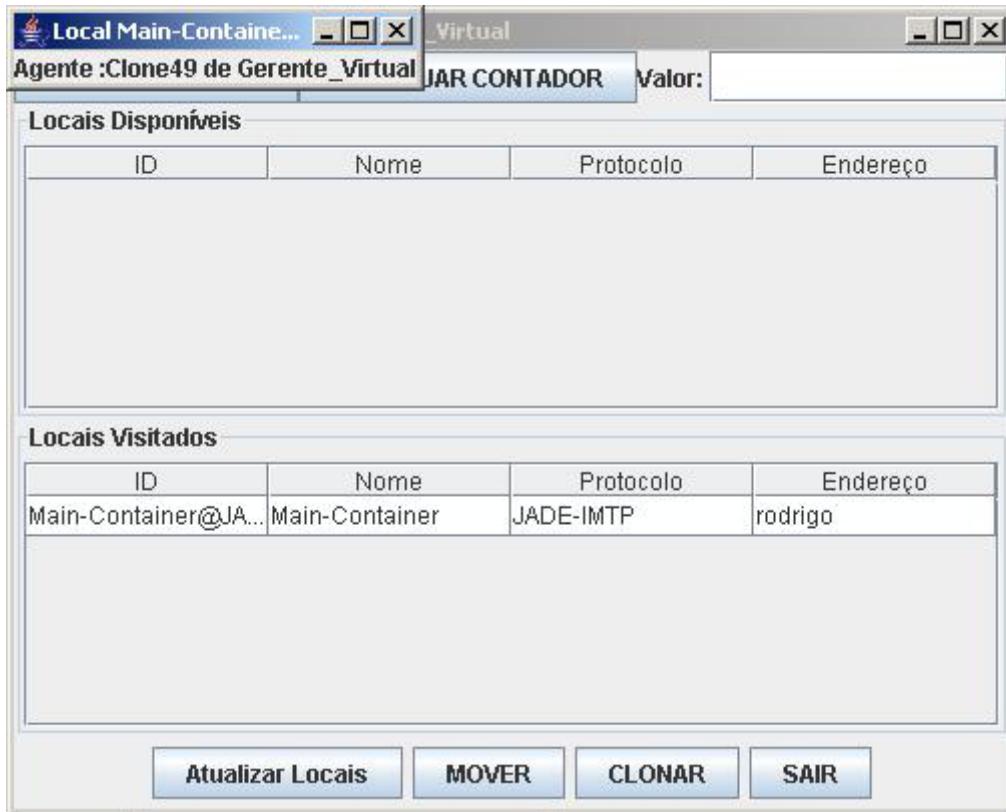


Figura 5.13 Agente clonado.

Uma vez clonado o agente, chegou à plataforma destino e executou as rotinas desejadas, enviando ao final um e-mail contendo as informações desejadas. A Figura 5.14 apresenta uma interface onde se coloca o e-mail de destino para o envio da mensagem pelo sistema. Na Figura 5.15 observa-se a mensagem que o e-mail foi enviado, concluindo a tarefa do agente. A Figura 5.16 apresenta o resultado em interface texto.

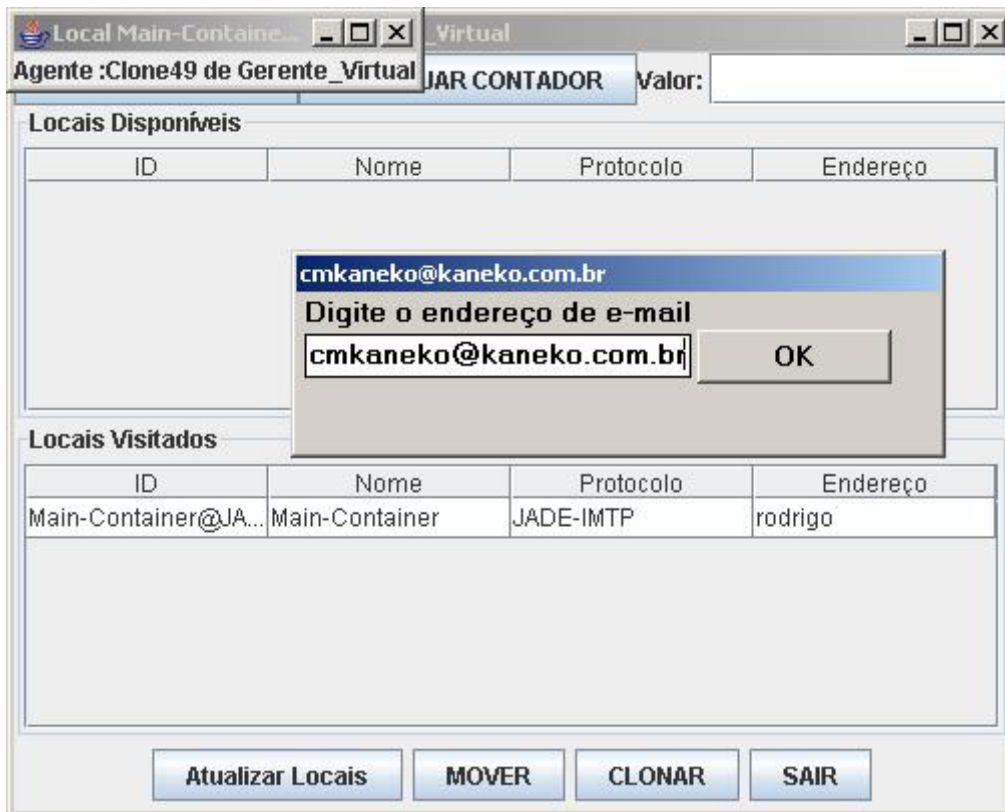


Figura 5.14 Solicitando e-mail

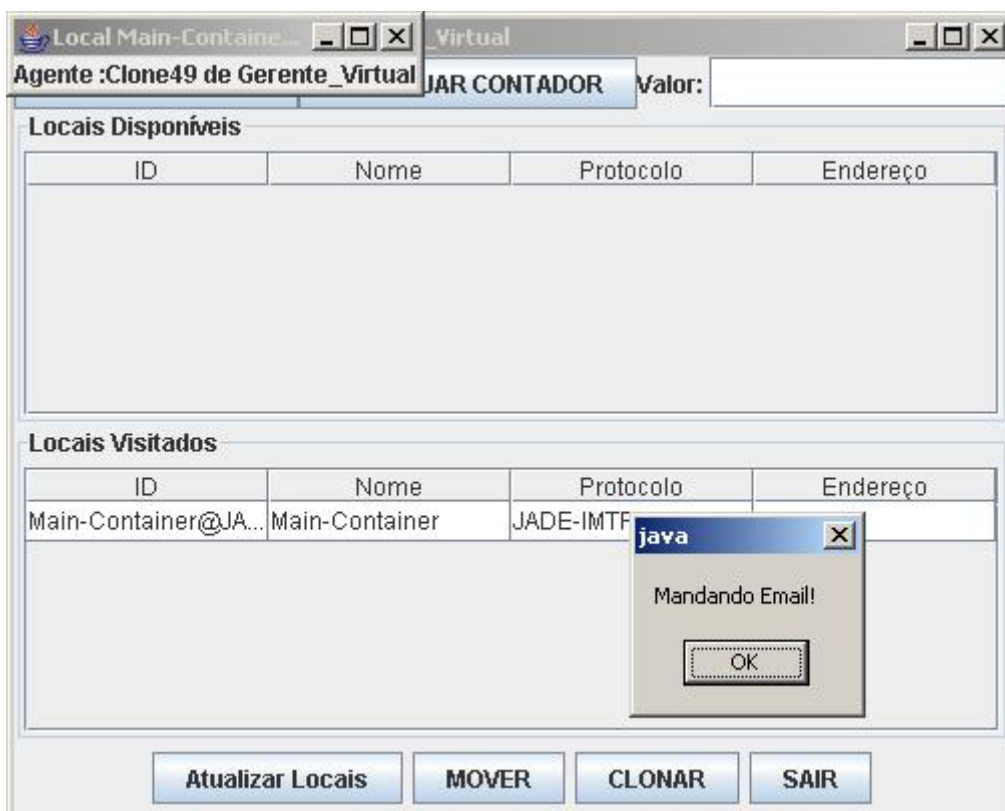


Figura 5.15 Agente executando rotinas de busca de informações gerenciais

```

c:\ Prompt de comando
-----
Gerente_Virtual is now cloning itself.
22/09/2005 21:39:59 jade.core.mobility.AgentMobilityService$CommandSourceSink handleInformCloned
INFO: Cloning agent < agent-identifier :name Gerente_Virtual@rodrigo:1099/JADE :addresses <sequence http://rodrigo:7778/acc >> on container Main-Container
22/09/2005 21:39:59 jade.core.mobility.AgentMobilityService$CommandSourceSink handleInformCloned
INFO: Destination container for agent < agent-identifier :name Gerente_Virtual@rodrigo:1099/JADE :addresses <sequence http://rodrigo:7778/acc >> found
22/09/2005 21:39:59 jade.core.mobility.AgentMobilityService$CommandSourceSink handleInformCloned
INFO: Agent < agent-identifier :name Gerente_Virtual@rodrigo:1099/JADE :addresses <sequence http://rodrigo:7778/acc >> correctly serialized
22/09/2005 21:39:59 jade.core.mobility.AgentMobilityService$ServiceComponent createAgent
INFO: Incoming agent < agent-identifier :name "Clone49 de Gerente_Virtual@rodrigo:1099/JADE" >
22/09/2005 21:39:59 jade.core.mobility.AgentMobilityService$ServiceComponent createAgent
INFO: Agent < agent-identifier :name "Clone49 de Gerente_Virtual@rodrigo:1099/JADE" > reconstructed
22/09/2005 21:39:59 jade.core.mobility.AgentMobilityService$ServiceComponent createAgent
INFO: Agent < agent-identifier :name "Clone49 de Gerente_Virtual@rodrigo:1099/JADE" :addresses <sequence http://rodrigo:7778/acc >> inserted into LAD1
22/09/2005 21:39:59 jade.core.mobility.AgentMobilityService$CommandSourceSink handleInformCloned
INFO: Cloned Agent < agent-identifier :name "Clone49 de Gerente_Virtual@rodrigo:1099/JADE" :addresses <sequence http://rodrigo:7778/acc >> correctly created on destination container

```

Figura 5.16 Interface texto após execução de rotina de busca de informações gerenciais.

5.2.9 Avaliação dos resultados sobre agentes

O resultados obtidos dos experimentos alcançaram seu objetivo de forma satisfatória, pois foi possível utilizar agentes JADE. Utiliza-se do recurso de mobilidade para buscar informações em outra estação, conecta-se a uma DLL desenvolvida em linguagem Delphi e este envia um e-mail com o relatório solicitado. Mostrou-se que é possível o uso de agentes de software para executar tarefas de busca de informações gerenciais integrando-se a sistemas de informações legados, desenvolvidos em linguagem Delphi.

É importante observar que o experimento ficou restrito a uma rede local, sendo executado repetidamente.

Outro fato observado na plataforma JADE, durante o processo de testes, merece atenção. O teste de indisponibilidade de rede foi realizado enquanto o agente foi enviado para um destino. Neste momento o comportamento do agente se demonstrou instável este resultado não era esperado.

O processo de segurança quanto ao acesso á DLL também foi considerado neste experimento. Em uma aplicação real este fator deve ser considerado e implementado.

Experimentos utilizando-se de dispositivos móveis também não foram realizados, uma vez que o foco principal deste trabalho se concentrou em plataforma *Desktop*. Alguns fatores justificam esta decisão:

- Os gerentes das empresas consultadas demonstraram interesse, porém, para o futuro, pois alegavam pouca intimidade no uso de dispositivos móveis;
- A plataforma JADE apresenta-se em fase de desenvolvimento para este tipo de dispositivo;
- Seria necessário o estudo de outras tecnologias, o que poderia comprometer os prazos estipulados e
- Pode ser outra área de pesquisa para trabalhos futuros.

5.2.10 Resultados obtidos utilizando a tecnologia WMI

A busca por informações sobre hardware e software utilizando a tecnologia WMI a princípio apresentou-se um pouco distante da proposta do uso de agentes. O modelo implementado se demonstrou instável, no sentido do agente conseguir alcançar o objetivo nos três computadores testados, apenas em algumas situações. Em outras o agente alcançava o resultado apenas do computador local. Alguns pontos tornaram o processo favorável à sua utilização através do uso de agentes: a) pode-se destacar o fator tempo, onde o solicitante da informação agenda os relatórios através do agente e aguarda a chegada das informações; b) processo assíncrono, ou seja o solicitante solicita através do agente, este não necessita estar conectado ao servidor da unidade onde será gerado o relatório, efetuando a operação como se enviasse um email. Embora fique aqui apresentada como proposta, pois a mesma não foi implementada, a criação de um host para centralizar as informações de uma rede de computadores. Parece ser uma solução prática. Os resultados das informações coletadas diariamente, como apresentada na Figura 5.17, poderiam ser armazenados em um banco de dados e a gerência poderia buscar as informações da mesma forma que avalia as informações de Sistemas de Informação proposto no presente trabalho.

A implementação de um sistema de gerenciamento de informações de hardware e software em linguagem Delphi, utilizando-se dos componentes “*WmiSet Component Collection v1.6*” apresentou-se satisfatório (as informações que o componente se propõe a oferecer foram de fato disponibilizadas), principalmente por sua praticidade. Futuros

trabalhos poderão ser realizados, dentre eles a criação de um gerente virtual, que controlasse todas as informações a respeito das tecnologias utilizadas em uma organização.

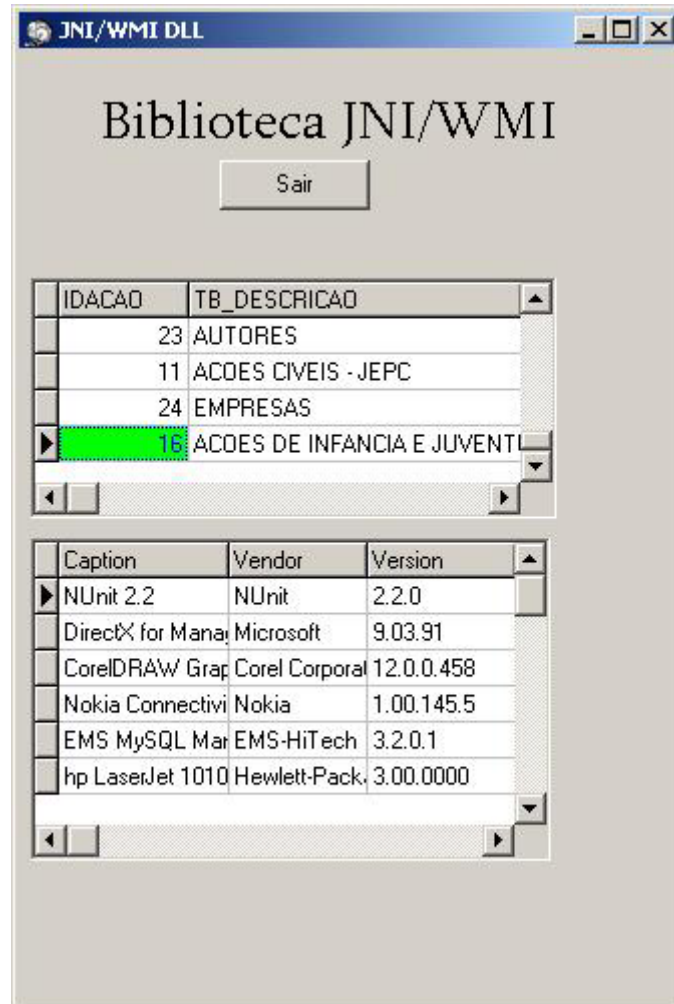


Figura 5.17 interface gráfica da aplicação Delphi para coleta de informações WMI

5.3 Conclusões do modelo proposto

O modelo computacional propõe a utilização dos agentes JADE como entidades que recebem a incumbência de percorrer a rede para gerar os relatórios a nível de SIG e SSD. Considerando que os sistemas legados são mantidos, é necessário utilizar a tecnologia JNI para que o agente possa repassar a tarefa à aplicação legada.

A solução de busca de informações a respeito de software e hardware utilizou um componente WMI específico para o Delphi, o que facilitou o processo.

6 CONCLUSÕES E TRABALHOS FUTUROS

O acúmulo de tarefas no nível gerencial ou a carência de informações (SPT, SIG e SSD) pode determinar a continuidade ou não de uma empresa no mercado. A busca por uma solução que permite disponibilizar as informações de maneira a poupar o exaustivo trabalho de geração destes relatórios, em ambiente distribuído, pode tornar o processo de gerenciamento das corporações mais dinâmico.

A pesquisa bibliográfica sinalizou para a tecnologia de agentes móveis e sistemas multi-agentes. Os experimentos realizados comprovaram na prática a viabilidade teórica dos agentes para estas finalidades.

A integração das tecnologias Java, agentes JADE, JNI, WMI, DLL em Delphi, para a geração de relatórios (SIG e SSD) em formato de arquivos (de planilha eletrônica), e o envio destes relatórios através de e-mails, apresenta-se extremamente condizentes com as necessidades levantadas.

Os resultados obtidos demonstram o que se esperava no início deste trabalho, isto é, os agentes de software podem ser utilizados como ferramenta de auxílio a gerentes na busca de informações de SSD. O estudo de caso, considerando a linguagem Delphi como linguagem do sistema legado, apresentou-se viável. Constatou-se que a integração utilizando o JNI é estável, e o aproveitamento do sistema existente será alcançado.

O modelo computacional apresentado provavelmente deva funcionar com outras soluções de agentes, principalmente os agentes móveis, o que abre uma linha de estudo para avaliar e comparar os resultados obtidos utilizando JADE e outras soluções de agentes como Aglets, Saci, Tagents, Soma.

Outras áreas de pesquisa estariam na criação de um gerente virtual (*framework*) executando as funções de um gerente de SSD. Este agregaria mais inteligência aos agentes para que pudessem gerar os relatórios, avaliar os resultados e apresentar à pessoa do gerente somente as decisões onde o nível de incerteza fosse considerável.

Os estudos realizados acenam para a utilização de um sistema de agentes que seja integrável com ferramentas de ontologias, como o *Jess*. Este software é proprietário e por esta razão não foi utilizado neste trabalho, porém o desenvolvimento de uma ferramenta de ontologia na modalidade de *software* livre pode ser um trabalho a ser desenvolvido.

7 REFERÊNCIAS BIBLIOGRÁFICAS

- ANT. **Java Based Build Tool**. Acessado em 20/09/2004. Disponível em: <http://ant.apache.org/>
- BATISTA, Emerson de Oliveira. **Sistemas de Informação: o uso consciente da tecnologia para o gerenciamento**. São Paulo: Saraiva, 2004
- BELLIFEMINE, F., CAIRE, G. POGGI, A., RIMASSA, G., **JADE A white Paper**, do site <http://exp.telecomitalialab.com>, 2003, acessado em 07/05/2005
- BIGUS, Joseph P. and BIGUS, Jennifer. **Construction Intelligent Agents with Java: a programmer's guide to smarter applications**. Wiley Computer Publishing, 1998
- BRAUN, Peter. **A Multi-Agent Approach to Manage a Network of MÓbile Agent Servers**. Computer Science Department, Friedrich Schiller university Jena no endereço: www.it.swin.edu.au/personal/pbraun/paper/2001-tr-fsu-domain-12.pdf, 200
- BREITMAN, Karin Koogan; LEITE, Julio C. S. P. **Ontologias – Como e Porquê Criá-las**. XXIV Congresso da Sociedade Brasileira de Computação. Salvador, BA, Agosto 2004.
- BRENNER, Walter. ZARNEKOW, Rüdiger. Wittig Hartmut. **Intelligent SoftwAgents**. Foundation and Application. Springer- Verlag Berlin Heidelberg, 1998.
- BUI, Tung; LEE, Jintae. **An agent-based framework for building decision support systems**. Decision Support System, Elsevier Science. University of Hawaii at Manoa, E303, 2404 Maile Way, Honolulu, HI 96822, USA.1999
- CANTU, Carlos Henrique, **Firebird Essencial**, Rio de Janeiro: Editora Ciência Moderna Ltda, 2005
- CHESS, D.M., HARRISON, C.G., and LEBINE, D. **Itinerant Agent for Mobile Computing**. IBM Resarch Division. IBM Research Report RC 20010. 1995
- CLARO, Daniela Barreiro. **Integração de Base de Dados utilizando a Mobilidade de Código**. Dissertação (mestrado) – Universidade Federal de Santa Catarina, Florianópolis, SC, 2000
- COLLINS, John; BILOT, Corey; GINI, Maria; **Decision Processes in Agent-Basead Automated Contracting**; IEEE INTERNET COMPUTING, April 2001.
- DMTF - DISTRIBUTED MANAGEMENT TASK FORCE. **CIM Tutorial**. 2002. Disponível em: <http://www.wbemsolutions.com/tutorials/CIM/metaschema.html>. Acesso em: 08/04/2005.
- DMTF. **Distributed Management Task Force, inc**. Acessado em 05/06/2005.

Disponível em : <http://www.dmtf.org/about>

ETHERYATRI, **Ethervatri .Net**. Disponível em:
<<http://www.geocities.com/siddharthuppal/EtherYatri.htm>>, acessado em Julho de 2005.

FERNANDES, Jorge Henrique Cabral. Ciberespaço: **Modelos, Tecnologias, Aplicações e Perspectivas - da Vida Artificial à Busca por uma Humanidade Auto-Sustentável**. In: MAIA, Pedro Fernandes; MOURA, Hermano Perreli. (Org.). Anais do Ensino de Informática no Nordeste. 2000.

FIPA, **Foundation For Intelligent Physical Agents**,
Disponível em <http://www.fipa.org> acessado em 21/06/2005

FIREBIRD. **Relational Database for the New Millennium**. Acessado em 09/04/2005.
Disponível em : <http://firebird.sourceforge.net/>

HÜBNER, Jomi Fred; SICHMAN, Jaime Simão. **SACI: Uma ferramenta para implementação e monitoração da comunicação entre agentes**. Proceedings of the International Joint Conference, 7th Ibero-American Conference on AI, 15th Brazilian Symposium on AI (Open Discussion Track), In Maria Carolina Monard and Jaime Simão Sichman, editors, São Carlos, 2000. ICMC/USP.

JADE, **Java Agent DEvelopment Framework**. Disponível em: <http://jade.tilab.com>
Acessado em 20/03/2005
JAMIL, George Leal. **Repensando a TI na Empresa Moderna**. Rio de Janeiro, editora Axcel Books, 2001.

JESS. **Sandia National Laboratories**. Acessado em 4/03/2005. Disponível em
<http://herzberg.ca.sandia.gov/jess/>

JNI, **Java Native Interface**. Acessado em 13/02/2005. Disponível em:
<http://java.sun.com/j2se/1.3/docs/guide/jni/index.html>

KOTAY, Keith; KOTZ, David. **Transportable Agents**. In: FININ, T.; LABROU, Y.(Ed.) Proceedings of the CIKM Workshop on Intelligent Information Agents, Third international Conference on Information and Knowledge Management. Gaithersburgs, MD, USA:[s.n.], 1994. Disponível em:
<<http://citeseer.ist.psu.edu/kotay94transportable.html>>.

LAUDON, Kenneth C.; LAUDON, Jane Price. **Sistemas de Informação: com Internet**. Rio de Janeiro-RJ- Ed. LTC, 1999.

LAUDON, Kenneth C. **Sistemas de informação gerencial: administrando a empresa digital**. São Paulo : Editora Prentice Hall, 2004.

LETSCHE, Thomas. **Redesign and Implementation of a Mobile Agent System compliant with the MAFFinder part of the MASIF Standard**. Munique, 2000. <http://www.tagents.org/thesis/> Acessado em 07/06/2005.

MACHADO, Renato Bobsin. **Uma Abordagem de Detecção de Intrusão Baseada em Sistemas Imunológicos Artificiais e Agents Móveis**. Dissertação (Mestrado). Universidade Federal de Santa Catarina – UFSC, Florianópolis, SC, 2004

MARTINSSON, Tobias. **Desenvolvendo scripts XML e WMI para o Microsoft SQL Server 2000**. São Paulo: Makron Books, 2002.

MEAD, Matthew. **Using the Java Native Interface with Delphi. 2005**. Acessado em março de 2005. Disponível em <http://www.pacifier.com/~mmead/jni/delphi/index.html>

MICROSOFT CORPORATION. **Learn- WMI**, [s.1], 1999. Disponível em: <<http://www.microsoft.com/downloads/release.asp?releaseid=12570>>. Acesso em: 21/03/2005.

MICROSOFT, Microsoft Corporation. **WMI Architecture**, Disponível em: <http://msdn.microsoft.com/library/en-us/wmisdk/wmi/wmi_architecture.asp?frame=true> acessado em 20/05/2005

NBJ. **Núcleo Brasileiro de JADE**. Julho 2004. Acessado em 15/07/2004. Disponível em <http://qos.tecnolink.com.br>

NWANA, Hyachinth S. Software Agents: **Network Intrusion Detection**. 3th edition. Ed. New Jersey: New Riders, 2002

O'BRIEN, James A. **Sistemas de Informação e as decisões gerenciais na era da Internet**. São Paulo, Editora Saraiva 2004.

O'REILLY, Kenneth C. & F.. **Heterogeneous Tools for Heterogeneous Network Management with WBEM**. Department of Electronic Engineering, Cork Ireland. Disponível em: <<http://www.dmtf.org/education/academicalliance/carey.pdf>>, 2002

OMG, **Object Management Group**, Disponível em :<<http://www.omg.org>>, 2005

PARAISO, Emerson C. **Concepção e Implementação de um Sistema Multi-Agentes para monitoração e Controle de Processos Industriais**. Curitiba: Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial – CEFET – PR, 1997. Dissertação de Mestrado.

PEREIRA Filho, Stenio Firmino. **Avaliação de Ambientes Servidores para Agentes Móveis**. Dissertação (mestrado) - Instituto de Ciências Matemáticas e de Computação – USP, SP, 2001

PROTEGE. **Desenvolvido pela Standford Medical Informatics da Stanford University Schools of Medicine**. Acessado em 03/03/2005. Disponível em: <http://protege.stanford.edu/>

REES, Jackie; KOEHLER, Gary J.; **Modeling Search in Group Decision Support Systems**. IEEE Transactions on Systems, Man, and Cybernetics; Applications and Reviews. Vol.34, N. 3. August 2004.

REZENDE, Denis Alcides; ABREU, Aline França. **Tecnologia da informação aplicada a sistemas de informação empresariais: o papel estratégico da informação nas empresas**. 2. ed. São Paulo: Atlas, 2001.

RUSSEL, S; Norvig, P. **Artificial Intelligence: A Modern Approach**; New Jersey, Prentice Hall, 1995.

SOMA, **Secure and Open Mobile Agent**. Disponível em <<http://lia.deis.unibo.it/research/SOMA>>, Acessado em julho de 2005.

STAIR, Ralph M. , **PRINCIPIOS DE SISTEMAS DE INFORMACAO - UMA ABORDAGEM GERENCIAL**. São Paulo, Editora LTC, 1998.

SOUZA, Eliane Moreira Sá de. **Uma Estrutura de Agentes para Acessória na Internet**. Dissertação (Mestrado) – Universidade Federal de Santa Catarina, Florianópolis, SC, 1996.

TAGENT, **Traveling Agents, Disponível** em: < <http://www.tagents.org/> >, acessado em Julho de 2005

TANEMBAUM, Andrew. **Computer Networks**. 4ª edição. Ed. New Jersey: Elsevier, 2003

UTO, Nelson. **Segurança de Sistemas de Agentes Móveis**. Dissertação (Mestrado) – Universidade Estadual de Campinas – Instituto de Computação, Campinas, SP, 2003

WEISS, Gerhard. **Multiagent System: A Modern Approach to Distributed Artificial Intelligence**. London: MIT Press, 1999

WOOLDRIDGE, Michael; DECKER, Keith; **Agents on the Net – Infrastructure, Technology, Applications**. Internet Computing, IEEE20000, April 2000.

WOOLDRIDGE, Michael; JENNINGS, Nicholas R.; **Intelligent Agents: Theory and Practice**. Knowledge Engineering Review, January 1995.

WOOLDRIDGE, Michael; **An Introduction to MultiAgent Systems**, British Library, 2001, pag. 23.
