

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE
CIÊNCIA DA COMPUTAÇÃO

HUGO SAMPAIO FIALHO

**WORKFLAGE:
uma ferramenta de definição e criação de Workflow
baseado em agentes de software para WEICOT**

São Luís
2003

HUGO SAMPAIO FIALHO

**WORKFLAGE:
uma ferramenta de definição e criação de Workflow
baseado em agentes de software para WEICOT**

Dissertação apresentada ao Curso de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão, para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Sofiane Labidi

São Luís
2003

HUGO SAMPAIO FIALHO

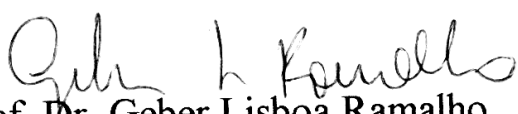
**WORKFLAGE :
uma ferramenta de definição e criação de Workflow
baseado em agentes de software para WEICOT**

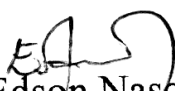
Dissertação apresentada ao Curso de
Pós-Graduação em Engenharia de
Eletricidade da Universidade Federal
do Maranhão, para obtenção do grau de
Mestre em Ciência da Computação.

Aprovada em 26 / 11 / 2003

BANCA EXAMINADORA

Prof. Dr. Sofiane Labidi (Orientador)
Doutor em Informática
Universidade de Nice, INRIA – França


Prof. Dr. Geber Lisboa Ramalho
(Membro da Banca Examinadora)


Prof. Dr. Edson Nascimento
(Membro da Banca Examinadora)

Este trabalho é dedicado, acima de tudo, a Deus,

A minha esposa Neusa e a meus filhos Higo, Amanda e Hellen.

A meus pais Antonio Sampaio e Olga de Araújo.

A toda minha família e amigos.

AGRADECIMENTOS

A Deus, pela minha existência.

Ao meu orientador Prof. Dr. Sofiane Labidi pela paciência e por acreditar no meu trabalho.

À FAPEMA, pelo financiamento da pesquisa.

À minha esposa Neusa, pelo carinho e compreensão.

Aos meus filhos Higo, Amanda e Hellen, que compreenderam a minha ausência durante este trabalho.

A meu pai, que não mediu esforços para me fazer um homem de bem.

À minha mãe, que sempre me incentivou e acredita em mim.

Aos meus irmãos, irmãs e todos os familiares que fazem parte da minha história.

À minha sogra M^a Graça Lobato pelo apoio prestado.

À coordenadoria de Pós-graduação em Engenharia de Eletricidade pelo auxílio dispensado no decorrer dos anos.

A todos aqueles que, direta ou indiretamente, colaboraram e acreditaram na realização deste trabalho.

RESUMO

Os Sistemas de Gerenciamento de Workflow (WfMS) tem sido utilizados em muitas organizações com o objetivo de reduzir, melhorar e controlar o fluxo de trabalho nos seus processos internos. Os WfMS atuais são, quase sempre, voltados para o gerenciamento de processo de negócios pouco estruturados onde o fluxo lógico das tarefas é totalmente predefinido. Porém, nem todos os processos de negócios são assim. Com o advento das novas tecnologias de comunicação entre as organizações, o ambiente do negócio está se tornando

cada vez mais dinâmico e vem exigindo processos mais complexos. Diante deste cenário, os WfMS existentes tem algumas desvantagens e limitações e necessitam de melhorias e modificações. Muitos pesquisadores têm explorado o poder da tecnologia de agente de software para prover soluções para os WfMS atuais. Há varias abordagens, algumas são melhorias com agentes, outras são baseadas em agentes. Neste trabalho apresentamos a WorkFlage como uma ferramenta para o desenvolvimento de Sistemas de Gerenciamento de Workflow baseado na tecnologia de Agentes de Software, utilizando o modelo proposto pelo projeto WEICOT. O trabalho esta organizado em 9 capítulos onde abordamos deste a tecnologia de Workflow, passando por um resumo da tecnologia de agentes, o estado da arte em Workflows baseados em agentes, indo até um estudo de caso de compra de materiais onde demonstramos a aplicação da ferramenta WorkFlage. Ao final deste trabalho faremos algumas considerações finais onde propomos melhorias e sugestões para trabalhos futuros.

Palavras-chave : Workflow, WfMS, WEICOT, Agente de Software, WorkFlage.

ABSTRACT

The Workflow Management Systems (WfMS) have been used in many organizations with the objective of reducing, improve and control the workflow in their internal process. Current WfMS are, usually, designed to the administration of loosely structured businesses process, where the logical flow of the tasks totally predefined. However, not all business processess are like this. With the coming of the new communication technologies among the

organizations, the business environment becoming more and more dynamic demanding more complex processes. Considering this scenario, existent WfMS have some disadvantages and limitations and need improvements and modifications. Many researchers have been exploring the power of software agent's technology to provide solutions for current WfMS. There are several approaches, some are improvements with agents, another are based on agents. In this work we present the WorkFlage as a tool for the development Workflow Management Systems based on to technologies of Software Agents for the model proposed by the project WEICOT. The work is organized in 9 chapters where we cover from the technology of Workflow, going through a summary of the agents' technology, the state of the art in workflow based on agents, to case study of materials purchase where we demonstrate the application of the WorkFlage tool. At the end of this work we will make some final considerations where we propose improvements and suggestions for future works.

Keywords: Workflow, WfMS, WEICOT, Software Agents , WorkFalge.

LISTA DE FIGURAS

Figura 1 - Exemplo de Workflow –Sistema de atendimento “on-line”.....	22
Figura 2 - Representação esquemática dos conceitos - WFMC	27
Figura 3 - Modelo de gatilho	37
Figura 4 - Modelo de interoperabilidade da WfMC	39
Figura 5 - Modelo de referência da WfMC	40
Figura 6 - Arquitetura básica de um agente	48
Figura 7 - Modelo conceitual dos agentes móveis.....	58

Figura 8 - Sistema de agentes	59
Figura 9 - Conceito de região	59
Figura 10 - Workflow baseado em agentes	71
Figura 11 - Workflow executado por agentes.....	72
Figura 12 - Modelo de referência do APMS	75
Figura 14 - Representação do tempo no WEICOT.....	95
Figura 15 - Arquitetura do WorkFlage	100
Figura 16 - Modelo do agente no Workflage.....	104
Figura 17 - Ciclo de mobilidade do agente.....	107
Figura 18 - Relação entre agentes.....	112
Figura 19 - Mapeamento das atividades no o agente.....	117
Figura 20 - Tela principal da ferramenta de definição	118
Figura 21 - Tela de definição de processo	122
Figura 22 - Tela de definição de atividades	124
Figura 23 - Tela de definição de tarefas	
Figura 24 - Tela de definição de perfis	
Figura 25 - Tela de código de tarefa	128
Figura 26 - Processo distribuído	132
Figura 27 - Tela do servidor WorkFlage	134
Figura 28 - Diagrama de fluxo de dados do processo de compra.....	140
Figura 29 - Carregando o processo de compras.....	144
Figura 30 - Informação do agente comprador	145
Figura 31 - Tela de resposta de coleta de preço.....	145
Figura 32 - Compra pelo menor preço.....	146
Figura 33 - Exemplo da tela de um Site	146

LISTA DE TABELAS

Tabela 1 - Exemplo de definição de Workflow utilizando script.....	23
Tabela 2 - Atributos de um papel	111
Tabela 3 - Arquivo de projeto.....	130
Tabela 4 - Arquivo de inicialização do site cliente.....	137
Tabela 5 - Atributos do agente processo.....	141
Tabela 6 - Atributos do agente comprador	142
Tabela 7 - Código da Tarefa “Consultar Menor Preço”	143

LISTA DE ABREVIATURAS E SÍMBOLOS

ACL	Agent Communication Language
ADEPT	Advanced Decision Environment for Process Tasks
AM	Agentes Móveis
APMS	Agent-based Process Management System
BT	British Telecom
CE	Comércio Eletrônico
IA	Inteligência Artificial
IAD	Inteligência Artificial Distribuída
JAT	Java Agent Template

KQML	Knowledge Query and Manipulation Language
KSE	Knowledge Sharing Effort
OMG	Object Management Group
RMI	Remote Method Invocation
RPC	Remote Procedure Call
RSA	Research Student Application
SMA	Sistema Multiagente
TI	Tecnologia da Informação
TRP	Technology Reinvestment Project
TSE	TRP Support Environment
UML	Unified Modeling Language
URL	Uniform Resource Locator
WfDL	Workflow Description Language
WFMC	Workflow Management Coalition
WfMS	Workflow Management System
XML	eXtensible Markup Language

SUMÁRIO

LISTA DE FIGURAS	6
LISTA DE TABELAS	8
ABREVIATURA E SÍMBOLOS	9
1 INTRODUÇÃO	15
1.1 OBJETIVO	15
1.2 JUSTIFICATIVA	16

1.3	ORGANIZAÇÃO DO TRABALHO	18
2	WORKFLOW	20
2.1	CONCEITO DE WORKFLOW	21
2.2	PROCESSOS	22
2.3	OUTROS CONCEITOS.....	24
2.4	FUNÇÕES DE UM WfMS.....	27
2.4.1	<i>Funções básicas.....</i>	27
2.4.2	<i>Outras funções de um WfMS.....</i>	30
2.5	TIPOS DE WORKFLOW	32
2.5.1	<i>Produção.....</i>	32
2.5.2	<i>Administrativo.....</i>	33
2.5.3	<i>Colaborativo.....</i>	34
2.5.4	<i>Ad-hoc.....</i>	34
2.5.5	<i>Web-Based Workflow.....</i>	35
2.6	MODELAGEM DE WORKFLOW	35
2.6.1	<i>Modelo de Gatilho</i>	36
2.6.2	<i>Modelo de Casati / Ceri.....</i>	37
2.7	A WORKFLOW MANAGEMENT COALITION	38
2.8	MODELO DE INTEROPERABILIDADE (WfMC)	38
2.9	MODELO DE REFERÊNCIA DA WfMC	40
2.9.1	<i>Interface 1.....</i>	
2.9.2	<i>Interface 2.....</i>	
2.9.3	<i>Interface 3.....</i>	41
2.9.4	<i>Interface 4.....</i>	42
2.9.5	<i>Interface 5.....</i>	42
2.9.6	<i>WAPI.....</i>	43
2.10	VANTAGENS E BENEFÍCIOS.....	43
2.11	CONCLUSÕES	44
3	GENTES DE SOFTWARE.....	46
3.1	INTRODUÇÃO	46
3.2	O QUE SÃO AGENTES DE SOFTWARE ?	46
3.3	CARACTERÍSTICAS	48
3.4	AGÊNCIA	51

3.5	AGENTES X OBJETOS	51
3.6	CLASSIFICAÇÃO DE AGENTES	52
3.7	LINGUAGEM DE COMUNICAÇÃO	53
3.7.1	<i>A linguagem de Comunicação KQML</i>	54
3.8	AGENTES MÓVEIS	55
3.9	CARACTERÍSTICAS E VANTAGENS DOS AGENTES MÓVEIS.....	60
3.10	EXEMPLOS DE SISTEMAS DE AGENTES MÓVEIS	61
3.11	APLICAÇÕES	62
3.12	CONCLUSÃO.....	65
4	WORKFLOW E OS AGENTES DE SOFTWARES	66
4.1	LIMITAÇÕES DOS WORKFLOWS ATUAIS	67
4.2	COMO OS AGENTES DE SOFTWARES PODEM AJUDAR?	69
4.3	WORKFLOW BASEADO EM AGENTES DE SOFTWARE	70
4.4	DUAS ABORDAGENS	70
4.4.1	<i>Workflow baseados em agentes.....</i>	71
4.4.2	<i>Workflow Aprimorado por agentes</i>	71
4.5	TRABALHOS RELACIONADOS.....	73
4.5.1	<i>APMS.....</i>	73
4.5.2	<i>DartFlow.....</i>	77
4.6	CONCLUSÃO.....	
5	O PROJETO WEICOT	
5.1	RELEVÂNCIA DO PROJETO	82
5.2	ARQUITETURA DO WEICOT	84
5.3	MODELO CONCEITUAL DO WEICOT.....	84
5.3.1	<i>Processo.....</i>	85
5.3.2	<i>Tarefas.....</i>	86
5.3.3	<i>Restrições.....</i>	87
5.3.4	<i>Estados da Tarefa.....</i>	87
5.3.5	<i>Gatilho.....</i>	87
5.3.6	<i>Relacionamento entre tarefas.....</i>	87
5.3.7	<i>Agente.....</i>	89
5.3.8	<i>Recurso</i>	90
5.4	COMPORTAMENTO COOPERATIVO	90

5.5	ASPECTOS TEMPORAIS	93
5.6	CONCLUSÃO.....	95
6	WORKFLAGE	97
6.1	ARQUITETURA PROPOSTA.....	97
6.2	MODELAGEM DO AGENTE	102
6.3	AS HABILIDADES DO AGENTE	106
6.3.1	<i>Mobilidade</i>	106
6.3.2	<i>Sensibilidade ao ambiente</i>	107
6.3.3	<i>Gerenciamento de Workflow</i>	108
6.3.4	<i>Interação com os recursos locais</i>	109
6.3.5	<i>Aprendizagem</i>	109
6.3.6	<i>Comunicação</i>	109
6.4	MODELO DE PERFIL.....	110
6.5	RELAÇÕES ENTRE AGENTES	111
6.6	BASE DE CONHECIMENTO	112
6.7	REGRAS DE PRODUÇÃO	113
6.8	CONCLUSÃO.....	114
7	FERRAMENTA DE DEFINIÇÃO	116
7.1	TELA DE DEFINIÇÃO DO PROJETO	118
7.2	TELA DE DEFINIÇÃO DO PROCESSO.....	
7.3	TELA DE DEFINIÇÃO DE ATIVIDADES	
7.4	TELA DE DEFINIÇÃO DE TAREFAS	124
7.5	TELA DE DEFINIÇÃO DE PERFIS	126
7.6	O ARQUIVO DE PROJETO	128
7.7	AMBIENTE DE EXECUÇÃO	130
7.8	O SERVIDOR WORKFLAGE	132
7.9	IMPLEMENTAÇÃO	135
7.9.1	<i>A geração da ferramenta</i>	136
7.10	CONCLUSÃO.....	137
8	ESTUDO DE CASO: PROCESSO DE COMPRA.....	139
8.1	MODELAGEM DO PROCESSO NO WORKFLAGE	141
8.2	TELAS DO PROCESSO DE COMPRA.....	144

8.3	CONCLUSÃO.....	146
9	CONSIDERAÇÕES FINAIS	148
	REFERÊNCIAS	151

1 INTRODUÇÃO

A cada dia as empresas vêm investindo muito mais em tecnologia, a fim de melhorar a produtividade e se tornarem mais competitivas e sólidas no mercado. Uma das tecnologias, chamada de **Workflow**, consiste em automatizar todo ou parte dos processos de negócio das organizações, controlando o “fluxo” de interação e a colaboração entre as tarefas que compõem estes processos.

Com os Workflows, surgiu a necessidade de um sistema que defina, crie e gerencie a execução dos Workflows, o qual chamamos de **Sistemas de Gerenciamento de Workflow** (do inglês *Workflow Management System -WfMS*).

Para o desenvolvimento de WfMS voltados para aplicações em ambientes distribuídos e dinâmicos tais como o E-Commerce, acreditamos que o paradigma de Agente de *Software* deve ter um grande significado, pois uma vez investidos de autonomia e mobilidade, os Agentes de Software podem representar fielmente os processos fora de seus limites geográficos mantendo ainda assim a integridade dos mesmos.

Este trabalho se propõe a apresentar um modelo de sistema de gerenciamento de workflow baseado em agentes de software aplicando os conceitos do projeto WEICOT, cujos detalhes veremos no capítulo 4 desta dissertação e implementar uma ferramenta para a definição de processos orientados a agentes de software. A todo este ambiente de criação e gerenciamento de workflow baseado em agentes, estamos denominando de **WorkFlage**, cujo detalhamento está descrito nos capítulos 6 e 7 deste trabalho.

1.1 Objetivo

Esta dissertação de mestrado pretende abordar o tema dos sistemas de

gerenciamento de Workflow baseado em Agentes Inteligentes, objetivando definir modelos mais flexíveis e com maior desempenho. Isto envolve: modelagem, projeto, e desenvolvimento de agentes de software voltados ao gerenciamento de Workflows adaptativos.

A Orientação Agente oferece um suporte para modelagem de conceitos como a cooperação e a comunicação nos sistemas de workflows (LABIDI; HAMMOUDI,2000). As varias ocorrências envolvendo o tema de workflow e agentes na literatura da área de Ciência da Computação, nos despertou para a necessidade de um detalhamento no assunto, o que nos levou a iniciar este trabalho. Portanto, pretendemos alcançar com este trabalho os seguintes objetivos específicos:

Pesquisar a contribuição dos Agentes de Software na área de gerenciamento de Workflow.

Definir um modelo de sistema de gerenciamento de Workflow integrando o paradigma de agentes buscando maior flexibilidade.

Desenvolver uma ferramenta para o desenvolvimento de Workflow baseado em agentes para o modelo do WEICOT.

Aplicar o sistema proposto ao domínio de E-commerce.

1.2 Justificativa

Com a popularização da Internet na década de 80, muitas foram as aplicações desenvolvidas para tirar melhor proveito da plataforma empregada, gerando assim maior agilidade e confiabilidade nos serviços prestados aos usuários.

Uma das aplicações é o chamado comércio eletrônico (e-commerce) que tem se tornado bastante atraente aos olhos de muitas organizações, as quais passaram a investir

maciçamente na obtenção de ferramentas que elevem o seu padrão de qualidade e rapidez no atendimento ao cliente.

O primeiro passo para que estas metas sejam alcançadas é o controle dos processos de negócio da empresa. A compra de materiais, por exemplo, é um processo de negócio comum às organizações e o sucesso da produtividade desta empresa, pode estar diretamente ligado ao controle de fluxo deste processo.

A Workflow Management Coalition¹ (WfMC) define um workflow como a automação de processo de negócio (*business process*), como todo ou em parte, durante o qual documentos, informações, ou tarefas são passados de um participante para outro para ação, de acordo com um conjunto de regras de procedimentos.

Devido à abordagem específica na maioria dos workflows, encontramos incorporada à sua modelagem a característica estática, o que a diferencia das características muitas vezes exigidas por aplicações como as de comércio eletrônico onde o dinamismo precisa ser implementado.

Uma tendência bem promissora consiste em ver um *workflow* como uma coleção dinâmica de processos colaborativos, o que exige um gerenciamento de seus desempenhos por componentes inteligentes reconhecidos como *Agentes de Software*.

Os agentes de software (AS) possuem características bem interessantes como autonomia, comportamento cooperativo, e eventualmente, mobilidade (LABIDI; HAMMOUDI,1999). Além disso, eles levam em consideração as características do ambiente no qual eles evoluem. Isto traz muita flexibilidade, desde que a combinação de processo possa estar diretamente baseada nos estados e decisões dos agentes, em lugar de serem predefinidos com antecedência.

¹ Organização internacional sem fins lucrativos, criada em 1993, incluindo vários distribuidores de WFMS. A WfMC tenta desenvolver padrões para sistemas de Workflow (www.wfmc.org).

Diante do acelerado crescimento e dos benefícios trazidos pela tecnologia de Gerenciamento de Workflow, além dos benefícios trazidos pelos Agentes de Software, introduzindo certo grau de autonomia e mobilidade nos Workflows e do espaço promissor introduzido pela Internet para as aplicações de comércio eletrônico, acreditamos que pesquisar a união de tais tecnologias e apresentar um estudo elaborado nos parece um ganho e uma contribuição significativa à comunidade científica.

Há atualmente grande interesse no emprego da tecnologia, agentes em todas as áreas da computação e até em outras áreas como automação e controle. Nos sistemas de workflow, foi organizado no ano de 2000 o primeiro workshop sobre o uso da tecnologia de agentes nos sistemas de workflows no congresso internacional: *International Conference on Artificial Intelligence IC-AI'2000: June 26-29, 2000, Monte Carlo Resort, Las Vegas, Nevada, USA*. Como exemplos de trabalhos na área citamos o (MERZ,1997) e (JENNINGS,1996). Porém, pouca demonstração prática é apresentada, o que pretendemos fazer neste trabalho.

Através do modelo proposto neste trabalho, é possível verificar o controle do fluxo em um processo onde os agentes supervisionam e controlam as tarefas inerentes a este processo, com o objetivo de trazer maior flexibilidade e autonomia ao sistema.

1.3 Organização do trabalho

Esta dissertação está organizada de forma a apresentar o contexto teórico no qual este trabalho está inserido, bem como a implementação de um ambiente favorável ao desenvolvimento e gerenciamento de workflow baseado em agentes. A dissertação encontra-se dividida em 9 capítulos,:

No Capítulo 1, é feita a introdução ao trabalho proposto, bem como descrevemos o objetivo, a justificativa e como este organizado este trabalho.

No Capítulo 2, mostra-se a tecnologia de workflow e seus conceitos, as funções básicas de um sistema de gerenciamento de workflow, tipos de workflow, técnicas de modelagem, os modelos de referências da WfMC, e os benefícios desta tecnologia.

No capítulo 3 abordamos a tecnologia de agentes de software, enfocando o que são, quais as suas características, categorias dos agentes, a linguagem de comunicação e as ferramentas de desenvolvimento. Enfatizamos também os Agentes Móveis, pois toda a conceitualização do trabalho está baseada nessa tecnologia.

No capítulo 4, procuramos mostrar a união das duas tecnologias tratadas nos dois capítulos anteriores. Começamos mostrando as limitações dos sistemas atuais, e em seguida mostramos como esta tecnologia está sendo aplicada nos sistemas de workflow.

No capítulo 5 Abordaremos o projeto WEICOT, sua relevância, arquitetura, modelo conceitual, etc. A nossa contribuição se propõe a implementar um gerenciamento de workflow baseado no conceito deste projeto.

No capítulo 6 tratamos da nossa contribuição propondo uma arquitetura baseada em agentes para o gerenciamento de workflow, a qual denominamos de Workflage

No capítulo 7 Apresentamos o ambiente Workflage como uma ferramenta de definição e criação de Workflow baseado na arquitetura proposta..

No capítulo 8 apresentamos um estudo de caso que demonstra a implementação de um simples processo de Compra de Materiais usando a ferramenta proposta.

Finalmente, o capítulo 9 apresenta as nossas considerações finais para o trabalho e toda a bibliografia utilizada durante a pesquisa.

2 WORKFLOW

Apresentamos neste capítulo, uma visão geral da tecnologia de Workflow. São discutidos conceitos básicos, características, aplicações, vantagens e desvantagens de sistemas baseados na tecnologia dos Sistemas de Gerenciamento de Workflow.

No mundo em que vivemos, tornou-se clara a necessidade de organizações melhorarem a cada dia os seus processos de negócios visando não somente maiores lucros, mas também a sua perpetuidade no mercado.

Diante das várias opções para atingir esses objetivos, as empresas se deparam com a necessidade de um apoio mais robusto e eficiente do que uma simples ferramenta de automação comercial para controlar o fluxo de caixa e o estoque dos produtos. Torna-se evidente a necessidade de uma profunda avaliação nos processos internos das empresas com vista a um controle total do fluxo de atividades, agilizando e garantindo, desta forma, a eficiência nos serviços e produtos prestados por esta empresa ao ambiente em que está inserida.

Há várias tecnologias que dão suporte às empresas para melhorar os processos e agilizar a informação. Além de técnicas de reengenharia dos processos organizacionais, dispomos de uma tecnologia que trata da automação dos processos propriamente ditos, e evoluiu a uma instância tal que hoje podemos nos referir a ela como um conjunto de tecnologias e metodologias de automação de processos: a *tecnologia de workflow*.

Uma das principais motivações para se pesquisar essa tecnologia vem do fato de que ela permite, quando aplicada de maneira correta, atingir os objetivos anteriormente mencionados, trazendo assim benefícios financeiros para quem a fomenta. Esse fato provavelmente servirá como grande aliado durante as primeiras pesquisas sobre o tema.

2.1 Conceito de Workflow

O workflow está, antes de tudo, envolvido com a automação de processos compostos de atividades realizadas por humanos e/ou máquinas, principalmente aquelas onde há a interação com ferramentas de TI (Tecnologia da Informação). Desta forma, pode-se entender workflow inicialmente como um meio de visualizar, analisar e melhorar os processos, sempre buscando a automação por ferramentas específicas.

Também deve-se saber que o workflow se preocupa com a automação de processos onde documentos, informações ou tarefas são trocados entre os seus participantes, de acordo com um conjunto definido de regras e objetivos. Desta forma, há o intercâmbio de elementos dos processos entre as atividades constituintes deste.

Mesmo não havendo consenso absoluto na definição dos conceitos de workflow, existe um esforço no sentido de criarem-se padrões para tal tecnologia. Isso está se concretizando com a *Workflow Management Coalition* (WfMC), um consórcio de empresas desenvolvedoras de tecnologia de workflow, que visa criar e fomentar o uso de padrões de definição e intercomunicação entre as várias ferramentas existentes. A WfMC define workflow como a:

[...]Automação de processo de negócio no todo ou em parte, durante o qual documentos, informações, ou tarefas são passadas de um participante para outro para ação, de acordo como um conjunto de regras de procedimentos.[...](WfMC,1999).

No princípio, os sistemas de workflow limitavam-se a passar os diferentes itens de trabalho para os respectivos participantes de um processo, devendo executar tais itens como tarefas. Hoje, a tecnologia amadureceu, e o processo como um todo é automatizado, desde sua definição até o fluxo das informações de controle necessárias à execução de cada uma de suas etapas. Com isso, um sistema de workflow não é mais apenas um roteador de tarefas, mas é responsável pelo processo que lhe é requisitado executar.

2.2 Processos

Processos de Negócio são peças fundamentais para o sucesso de qualquer empreendimento. Um processo pode ser entendido simplesmente como um conjunto de atividades destinadas a alcançar um determinado objetivo. Segundo a Workflow Management Coalition, um processo é:[...]“Um conjunto de um ou mais procedimentos ou atividades relacionadas, as quais coletivamente atingem um objetivo de negócios, dentro de uma organização”[...] (WfMC,1999).

Na definição de um processo, observa-se o uso de uma linguagem gráfica, onde elipses e retângulos representam atividades. As setas representam o fluxo das atividades. Na figura seguinte exemplificamos o processo de um sistema de atendimento on-line.

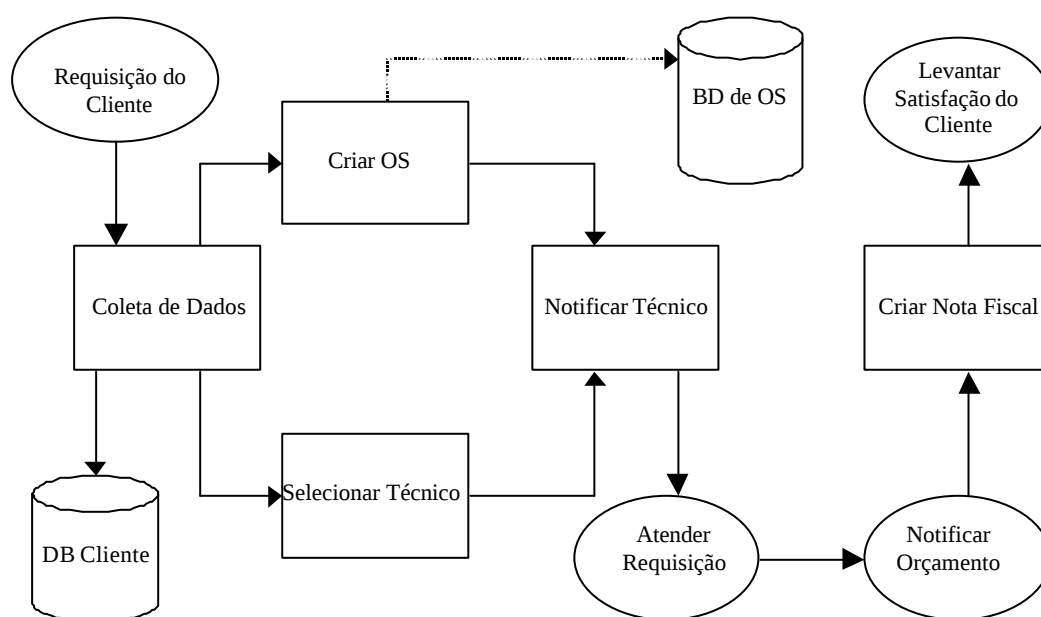


Figura 1 - Exemplo de Workflow –Sistema de atendimento “on-line”

Podemos utilizar também de algum tipo de script ou linguagem textual para a definição dos processos. Com um script, poderíamos ter algo como:

```

inicie_workflow();

    comece_atividade('Coletar dados');
    quando(ATV_CORRENTE, TERMINAR)

    comece_atividade('Criar OS') AND
    comece_atividade('Selecionar Técnico');
    quando(ATV('Selecionar Técnico'), TERMINAR)
    comece_atividade('Notificar Técnico');
    quando(ATV_CORRENTE, TERMINAR)

    comece_atividade('Atender requisição');
    quando(ATV_CORRENTE, TERMINAR)

    comece_atividade('Notificar orçamento');
    quando(ATV_CORRENTE, TERMINAR)

    comece_atividade('Criar nota fiscal');
    quando(ATV_CORRENTE, TERMINAR)

    comece_atividade('Levantar satisfação do cliente');

terminar_workflow();

```

Tabela 1 - Exemplo de definição de Workflow utilizando script

O código desse script também utiliza uma linguagem. Nele, os componentes `ATV_CORRENTE` e `TERMINAR` são pré-definidos como objetos que se referem à atividade corrente e a um de seus possíveis eventos, respectivamente, e as funções `iniciar_workflow()`, `comece_atividade()`, `terminar_workflow()`, e também outros comando não mostrado neste script, fazem parte da linguagem utilizada.

Representada de qualquer forma, a definição de processos é parte muito importante da tecnologia de workflow e, como veremos adiante, parte integrante de um sistema de workflow.

2.3 Outros conceitos

Além do conceito de *processo de negócio*, outros conceitos são necessários para compreensão da tecnologia de Workflow. São eles :

? Atividades

Uma atividade é um passo a ser dado dentro de um processo para que este se conclua. Segundo a WfMC, sua definição é:

[...]um passo lógico ou a descrição de uma parte do trabalho que contribui com a conclusão de um processo. Pode ser uma atividade manual ou uma atividade automatizada[...] (WfMC,1999).

Uma atividade também pode ser um sub-processo de um processo pai que está sendo modelado. Neste caso, este sub-processo pode ser considerado como uma atividade especial, sendo diferenciada das demais atividades que são passos indivisíveis de um processo.

? Processo de negócio

É um processo que está inserido no contexto de uma organização, ou tem como objetivo atingir metas de negócios. A WfMC define um processo de negócio como: [...]“Um tipo de processo dentro do domínio de uma estrutura organizacional e de uma política de negócios, que visa atingir objetivos de negócios”[...] (WfMC,1999).

Essa nomenclatura é um pouco estranha para os padrões brasileiros, e sua tradução, ainda mais. Não será muito utilizada, sendo o conceito de processo, puro e simples, mais adequado ao desenvolvimento desse texto.

? Sistema de Gerenciamento de Workflow (WfMS)

Um *Sistema de Gerenciamento de Workflow*, ou *Workflow Management System (WfMS)* é definido pela WfMC como “um sistema que, define, gerencia e executa ‘workflows’ através da execução de software cuja ordem de execução é controlada por uma representação computadorizada da lógica do workflow”.

Para simplificar o entendimento, podemos pensar em um WfMS como sendo um software que pode interpretar uma definição de processos (a representação computadorizada da lógica do workflow) e executá-la. Essa execução pode ser inerente a um processo que dure alguns minutos ou até muitos meses, variando em abrangência desde processos internos de uma organização até mesmo processos entre duas ou mais organizações. Mas mesmo com diferenças tão grandes entre os processos, os WfMS possuem características comuns que podem ser alvo de padronização e estudo.

? Instância do processo

Uma instância é uma representação interna da definição do processo a ser controlado pelo sistema de gerenciamento de workflow. Ele exprime as ocorrências que estão sendo processadas pelo sistema naquele momento. Quando um processo é iniciado no sistema de gerenciamento de workflow, é então criada uma *instância de processo* que por sua vez gera as instâncias das suas atividades. As Instâncias de processo e de atividades são manipuladas no controle do fluxo pelo sistema de gerenciamento de workflow.

? **Participantes**

Um participante de um Workflow é responsável pela execução parcial ou total de uma determinada instância de atividade. Um participante pode ser humano ou um software. Termos como: Ator, Agente e Usuário são bastantes utilizados na literatura para descrevê-los.

? **Itens de trabalho**

Do ponto de vista de um participante de Workflow, as atividades são representadas como uma coleção de itens de trabalho. Este conjunto de itens de trabalho é o resultado da instanciação de uma atividade no WfMS. Os itens de trabalho são apresentados ao participante em forma de lista de trabalho.

? **Regras**

As Regras definem de que forma os dados que trafegam no fluxo de trabalho devem ser processados, roteados e controlados pelo sistema de workflow.

? **Rota**

Caminho lógico que, definido seguindo regras específicas, tem a função de transferir a informação dentro do processo, ligando as atividades associadas ao fluxo de trabalho.

? Papéis

Um Papel é o conjunto de características e habilidades necessárias para um participante executar determinada(s) tarefa(s) pertencente(s) a uma atividade.

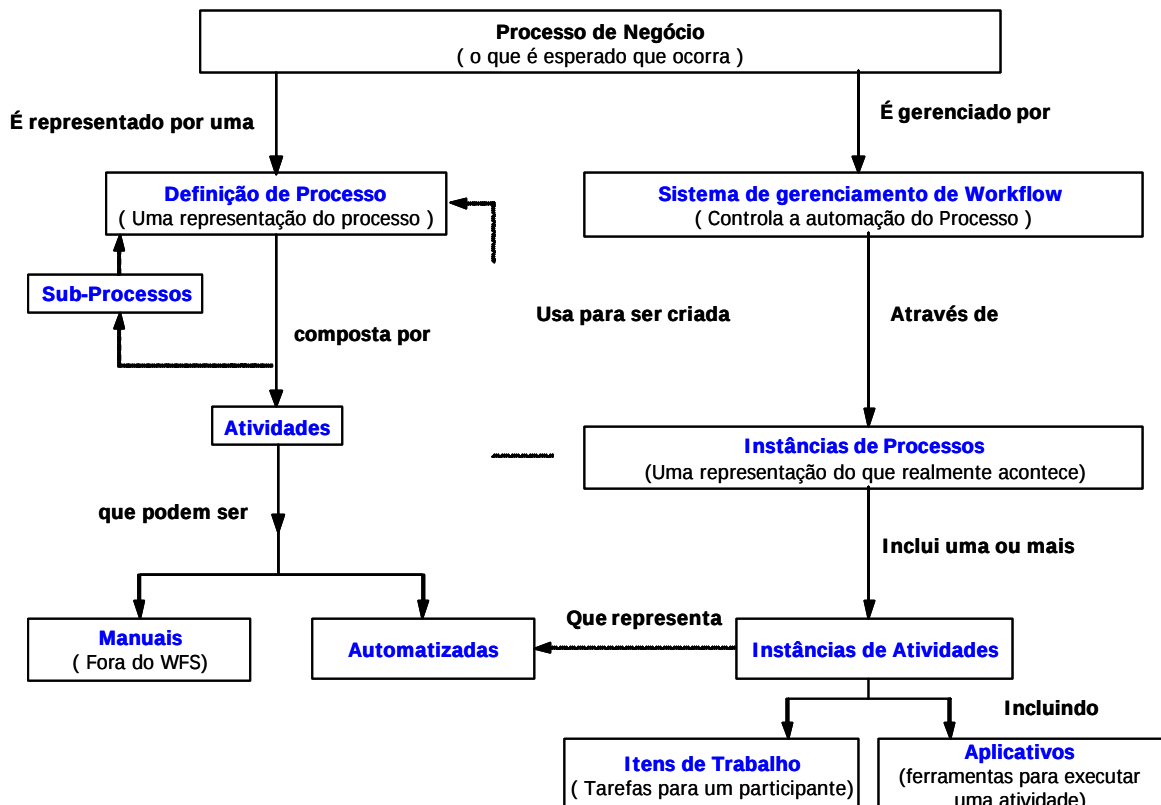


Figura 2 - Representação esquemática dos conceitos - WfMC

2.4 Funções de um WfMS

2.4.1 Funções básicas

O modelo de referência da WfMC procura dividir um WfMS em partes, sem comprometer a independência de características de implementação que o modelo deve

oferecer. Sendo assim, em um maior nível de abstração, um WfMS pode ser caracterizado por dar suporte às três áreas seguintes:

- a) Funções de tempo de criação ou construção (build-time functions): são funções responsáveis por criar as definições de processo a ser interpretada pelo restante do WfMS.
- b) Funções de tempo de execução (run-time functions): são as funções do WfMS responsáveis por gerenciar os processos de workflow em um ambiente operacional, e seqüenciar as várias atividades que fazem parte desses processos.
- c) Interações entre os participantes humanos e as ferramentas de TI: o WfMS deve ser capaz de aceitar a interação humano/TI para o processamento dos passos de uma atividade, como a produção de um documento de texto ou planilha de cálculo, aceitando seus resultados como parte da cadência do sistema.

? Funções de tempo de criação ou construção

São as funções do WfMS que resultam em uma representação computadorizada de processos. Durante este período, um processo é traduzido do seu contexto no mundo real para uma definição formal e processável, através do uso de uma ou mais técnicas de análise e modelagem (WfMC,1999).

As funções de tempo de construção podem ser implementadas através de ferramentas CASE, como editores gráficos, ou até mesmo através de scripts gerados em um editor de texto simples. A WfMC reconhece que a interface para essas funções é uma das partes que mais se distingue entre os produtos de workflow, dada a diversidade de formas de se implementar ferramentas que geram definições de processos. Mas o resultado do

processamento dessas ferramentas, ou seja, a definição do processo em si, é uma área de iminente padronização, para que as geradas por uma ferramenta possam ser visualizadas, editadas e/ou executadas em outras ferramentas de outros desenvolvedores, sem problemas.

? **Funções de tempo de execução**

As funções de tempo de execução agem como a ligação entre os processos enquanto modelados em uma representação computacional, e como tal, são vistos no mundo real, refletido nas interações, durante sua execução, com os usuários e as ferramentas de TI (WfMC,1999).

Nesta fase, a definição de um processo é interpretada por um software, responsável por criar e controlar instâncias desse processo, programar a seqüência dos passos de cada atividade ao longo de seu curso, invocar os recursos de TI e humanos necessários, entre outras funções. Há uma peça fundamental para que tudo isso aconteça: um software, o qual chamaremos de *workflow engine* ou *workflow management engine (WFM engine)*. Ele é o responsável pelas funções aqui descritas, e também componente central da arquitetura do modelo de referência da WfMC como um todo.

? **Interações entre os humanos participantes e ferramentas de TI**

Atividades individuais de um processo de workflow estão geralmente ligadas a operações humanas, freqüentemente realizadas em conjunto com ferramentas de TI (como por exemplo, editores de texto). Outro aspecto pode ser a necessidade de processamento de informações que requerem ferramentas específicas que atuam sobre estes dados. A interação dessas ferramentas com o software de controle, é necessária para invocá-las apropriadamente

e repassar os dados de forma correta.

Neste ponto, segundo a WfMC, há grandes vantagens em se possuir padronização, como a construção de uma interface comum para a interação de participantes e ferramentas de TI e a chance de se construir essas ferramentas de maneira que trabalhem, sem problemas, com vários sistemas de workflow.

2.4.2 Outras funções de um WfMS

Outras funções podem ser encontradas nos sistemas de workflow atualmente disponíveis no mercado. Entre elas podemos citar:

? **Roteamento de trabalho**

Em uma seqüência predefinida de execução de atividades, pode acontecer que na passagem de uma para outra exija uma decisão a ser tomada determinando qual a nova rota a ser adotada. O Roteamento do fluxo, nos sistemas convencionais, pode se basear em uma determinada ação tomada pelo usuário ou uma condição dos objetos manipulados na atividade corrente.

? **Invocação automática de aplicativos**

Uma vez verificada a lista de tarefas e sabendo-se qual é a próxima tarefa a ser executada, o sistema executa o aplicativo necessário para conclusão daquela tarefa. A maioria dos sistemas atuais permite especificar o aplicativo utilizado para cada atividade do Workflow.

? **Distribuição dinâmica de trabalho**

Quando definimos um processo, não determinamos um participante específico para a execução das atividades e sim definimos um papel a ser executado. No momento da instanciação do processo no WfMS, é necessário determinar qual o participante irá executar um determinado papel. Dizemos que há uma distribuição dinâmica de trabalho quando esta escolha é feita pelo WfMS.

? **Priorizar trabalho**

A capacidade de priorizar um trabalho é uma função chave nos WfMS. Nem sempre é interessante que a execução dos processos, sigam uma ordem baseadas no modelo FIFO (first-in first-out), portanto a maioria dos sistemas atuais, permitem alterar a prioridade de uma instância de processo.

? **Acompanhamento do trabalho**

Esta funcionalidade é, talvez a que mais atrai nos sistemas de gerenciamento. Saber imediatamente qual status de uma determinada atividade, com quem está no momento e quanto tempo está esperando, é uma característica fundamental em todos os sistemas de workflow. Normalmente os WfMS permitem acompanhamento “reativo”, ou seja, reporta o status corrente caso o usuário consulte.

2.5 Tipos de Workflow

Com a evolução da tecnologia e do mercado, fez-se necessário separar diferentes tipos de workflow de acordo com suas funcionalidades, para que se torne mais fácil aplicar o tipo correto para cada caso específico de aplicação dessa tecnologia.

Qualquer sistema de informação que se baseia na ordenação e controle de tarefas, pode ser automatizado usando técnicas de workflow, porém a WfMC propõe uma classificação do workflow baseada em seu objetivo final. Segundo a WfMC, pode-se distinguir os seguintes tipos:

2.5.1 Produção

O Workflow de Produção tem como objetivo principal a automação de várias tarefas similares, para a otimização da produtividade. Essa automação deve ser perseguida até que um processo que não necessite de intervenção humana seja conseguido. A única interação de pessoas no processo seria voltada para o tratamento de exceções, ou seja, dos itens que estão fora da especificação válida para o processo.

Os Workflows de produção envolvem atividades estruturadas que descrevem processos de informação complexos. Normalmente a sua execução requer um alto número de transações que acessam múltiplos sistemas de informação, e englobam processos de negócios repetitivos e previsíveis, como aplicação de empréstimo ou reivindicações de seguro.

Eles também envolvem processos repetitivos e previsíveis, mas estes são tipicamente processos de informação complexos envolvendo acesso a múltiplos sistemas de informação. Os sistemas de gerenciamento de workflow que suportam workflow de produção devem prover facilidades para definir dependências entre tarefas e controlar a execução de

tarefas com pouca ou nenhuma intervenção humana.

Nota-se que a intervenção humana neste tipo de processo é reduzida, e até mesmo o tempo necessário para tais intervenções é menor. Com isso, pretende-se ganhar produtividade sobre a execução de tarefas repetitivas, naturalmente automatizáveis, geralmente sem paradas.

Essas características conferem ao workflow de produção a capacidade de gerenciar processos bem complicados, como transferências monetárias diversas entre bancos, que hoje já não mais passam por um caixa, mas podem ser feitas diretamente de um terminal eletrônico. Também pode ser integrado mais facilmente com sistemas já existentes.

2.5.2 Administrativo

O Workflow administrativo envolve processos repetitivos e previsíveis com regras simples de coordenação de tarefas. Tanto a ordenação como a coordenação de tarefas em um workflow administrativo, podem ser automatizadas. Este tipo de workflow não engloba um processamento complexo de informações e não requer acesso a múltiplos sistemas de informação usados para suportar produção ou serviços administrativos. Sistemas de workflow administrativo são baseados tipicamente em correio eletrônico.

Workflows administrativos envolvem atividades fracamente estruturadas, repetitivas, previsíveis e com regras de coordenação de tarefas simples, tal como roteamento de uma requisição de viagens através de um processo de autorização.

Nesse tipo de workflow, a flexibilidade é mais importante que a produtividade. Sua principal característica é a facilidade em se definir os processos.

2.5.3 Colaborativo

Aqui, o principal enfoque é a capacidade de trabalho em grupo. Sistemas de workflow desse tipo são pensados para grupos que trabalham para a obtenção de objetivos comuns. Para isso, assuntos como comunicação via Internet entre os integrantes dos grupos estão incluídos. As definições dos processos não são rígidas, e podem sofrer modificações freqüentemente. Usa-se também o termo *Groupware* para indicar esse tipo de workflow. Mas nem toda aplicação de Groupware é workflow, como por exemplo videoconferências.

2.5.4 Ad-hoc

O workflow *ad hoc* é mais flexível do que os outros tipos. Permite que as definições de processos sejam modificadas de maneira fácil e rápida para se adequarem às adversidade que venham a surgir durante sua execução. O termo *Ad hoc* pode ser entendido como “livre” ou “não tão exigente”. Workflows Ad-hoc dão mais liberdade ao usuário, e não são tão exigentes quanto à segurança.

A ordenação e a coordenação de tarefas em um workflow ad-hoc não são automatizadas, mas sim controladas por humanos.

Não existe uma estrutura pré-definida para o processo, ou a mesma pode ser modificada em tempo de execução. A lógica do processo é definida pelas ações executadas pelo usuário, em tempo de execução.

Este tipo de Workflow envolve pequenos grupos de profissionais e tem a intenção de dar apoio a pequenas atividades que requerem uma solução rápida.

Workflows do tipo Ad-hoc suportam definição rápida e execução de modelos de processos menos complexos envolvendo coordenação humana. A arquitetura Ad-Hoc opera

em atividades que não permitem a previsibilidade das atividades a serem seguidas, mas que apresentam objetivos a serem alcançados. Como exemplo de workflow Ad Hoc podemos citar o desenvolvimento de software.

2.5.5 Web-Based Workflow

Um quarto tipo de workflow, embora ainda não definido pela WfMC, não poderia deixar de ser lembrado pelo seu grande significado nos dias de hoje. James Kobiellus (KOBIELUS,1997) identificou o Web-Based Workflow como um dos tipos de workflow existente.

Com o enfoque na arquitetura da internet, o Web-based workflow é distinto dos demais tipos apresentados, principalmente por causa da plataforma sobre a qual opera. Enquanto os tipos anteriores operam com vários protocolos e um grande leque de tecnologia de comunicação, este está restrito aos padrões utilizados hoje na Internet.

As soluções que utilizam o conceito do web-based workflow estão baseadas fortemente em formulários montados em HTML (HyperText Markup Language) e em páginas da web que implementam applets Java. Isso é uma das grandes vantagens dessas soluções, pois um formulário pode ser criado apenas uma vez e acessado de diversos browsers clientes diferentes. Isso gera uma uniformidade e traz facilidades na hora de criar e implementar essas soluções.

2.6 Modelagem de Workflow

A modelagem de processos deve ser utilizada para melhor entender, gerenciar e controlar os processos. O principal objetivo da modelagem de processos é representar os

processos de uma maneira clara e formal em diferentes níveis de abstração. A disponibilidade de modelos bem elaborados, possibilitam uma análise crítica das atividades existentes definindo assim, melhorias e racionalizações nos processos. As técnicas de modelagem utilizadas nos sistemas existentes são, em geral, proprietárias e os modelos diferem na teoria dos conceitos.

Várias técnicas de modelagem foram propostas com o objetivo de serem usadas para representar o comportamento dinâmico dos sistemas. Entre essas técnicas, destaca-se o *Modelo de Gatilho* e o *Modelo de Casati/Ceri* (CASATI,1995)

2.6.1 Modelo de Gatilho

O modelo de Gatilho é uma técnica simples de modelagem cujo objetivo é auxiliar a análise de workflow a fim de automatizar processos de negócio (JOOSTEN,1994). Entende-se por “gatilho” o disparo de uma atividade decorrida do término de uma atividade anterior.

Na notação deste modelo, cada atividade é representada por um retângulo contendo o nome da atividade. Um arco que aponta para determinada atividade representa que a atividade pode ser disparada pelos eventos que ocorrem como resultado de cuja atividade o arco está partindo. O modelo de gatilho é dividido em colunas, onde cada coluna contém as atividades para as quais o papel é responsável. O modelo de gatilho deve ser construído de acordo com três funções primitivas: 1) determinar os papéis de usuário, 2) descobrir quais atividades são executadas sob a responsabilidade de cada papel e 3) descobrir como cada atividade é disparada.

A figura seguinte mostra um exemplo de um modelo de gatilho, que descreve o procedimento de reclamação de um cliente. No exemplo a reclamação é registrada pelo

representante da organização que distribui o produto, que é a mesma pessoa que faz o contato pessoal com cliente.

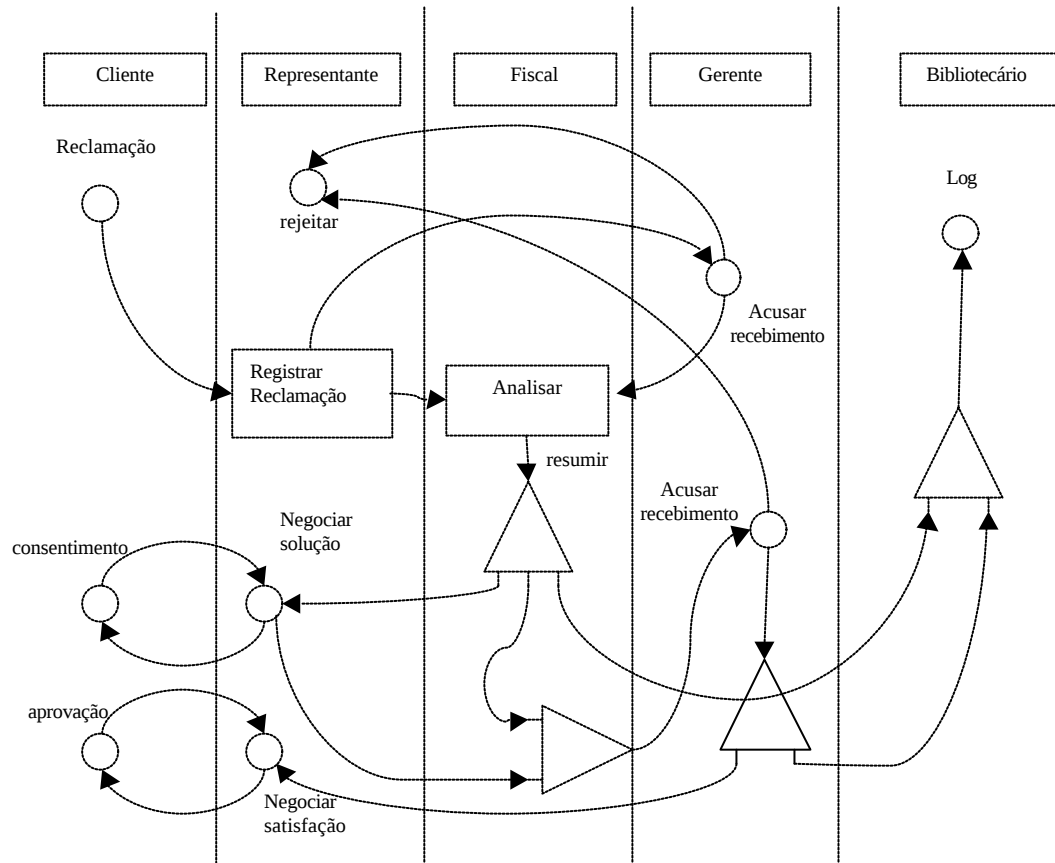


Figura 3 - Modelo de gatilho

2.6.2 Modelo de Casati / Ceri

O modelo de Casati/Ceri (CASATI,1995) é um dos mais completos para definir um processo. Além de incluir todos os conceitos já vistos de workflow, modela ainda o acesso à base externa de dados, através de SQL.

Este modelo possui uma linguagem para sua definição, denominada WfDL (Workflow Description Language) (CASATI,1995). O Modelo define cinco componentes para uma atividade: nome, descrição, pré-condições, ações e exceções. As pré-condições são expressões *booleanas* avaliadas antes da execução da atividade. As ações descrevem como os

dados são manipulados pela atividade e as exceções indicam atitudes a serem tomadas na ocorrência de determinados eventos anormais.

Neste modelo, uma atividade é representada por uma caixa, contendo quatro divisões, cada uma das quais armazena um ou mais elementos. De cima para baixo, situam-se as listas de pré-condições, o nome e descrição da tarefa, a lista de ações e a lista de exceções. Os símbolos de início e fim são denominados pelas duas linhas horizontais paralelas. Os atributos de quantidades de instâncias geradas e o número de instâncias que precisam terminar, são denotados por um retângulo colocado imediatamente antes e depois da caixa da atividade.

2.7 A Workflow Management Coalition

A Workflow Management Coalition (WfMC) é um órgão criado em 1993 com o objetivo de fornecer padrões para área de Workflow. Atualmente, a WfMC possui mais de 250 membros entre empresas fornecedoras e instituições acadêmicas. Sua missão é promover a área de Workflow, através da divulgação da tecnologia e geração de padrões para interoperabilizar diversos sistemas de Workflow. A necessidade de uma maior interoperabilidade torna-se bastante visível ao analisar-se o mercado atual de workflow, onde os sistemas de gerência de workflow diferem grandemente entre si, em termos de arquiteturas, modelos e conceitos utilizados.

2.8 Modelo de interoperabilidade (WfMC)

Este modelo foi concebido para permitir que diferentes ferramentas de modelagem de workflow e sistemas de gerência de workflow pudessem trocar informações

entre si. Desta forma, seria possível modelar um workflow utilizando uma ferramenta que suporte este padrão e implementá-lo em qualquer WfMS que entenda este padrão. Assim, um dos maiores problemas do atual mercado de workflow, que é a incompatibilidade entre os produtos de modelagem, poderia ser solucionado. A Figura 4 apresenta um esquema de como seria a relação entre as ferramentas de modelagem e os WfMS.

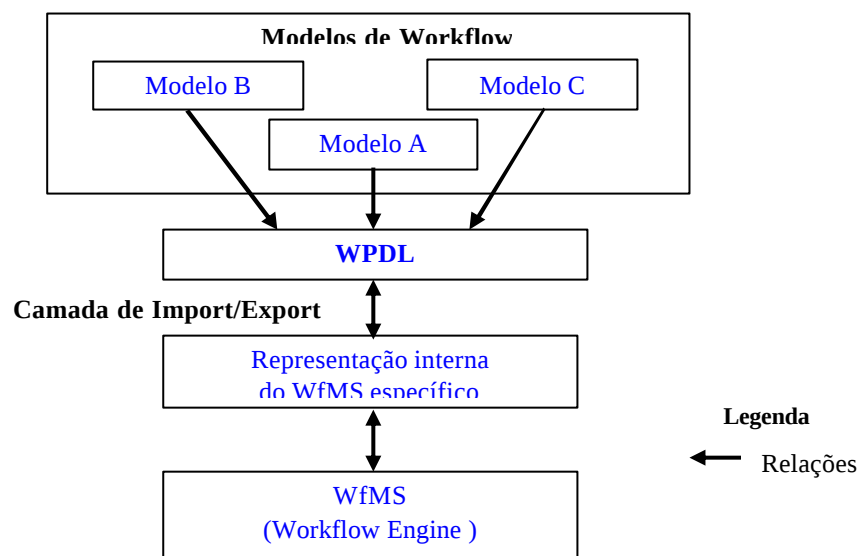


Figura 4 - Modelo de interoperabilidade da WfMC

A interoperabilidade é obtida através de dois elementos definidos em normas da WfMC, sendo :

- ? Um **Meta-modelo** contendo as informações que devem ser armazenadas sobre o workflow;
- ? Uma **Linguagem padrão** para a interoperabilidade **WPDL** (Workflow Process Definition Language)

2.9 Modelo de referência da WfMC

Com o objetivo de estabelecer padrões para a área de workflow, a WfMC desenvolveu um Modelo de Referência. Esse modelo identifica os componentes que se comunicam com suas respectivas interfaces.

O modelo de Referência da WfMC define cinco interfaces entre os componentes, além de uma interface sobre o serviço de execução de workflow, denominada WAPI (Workflow API).

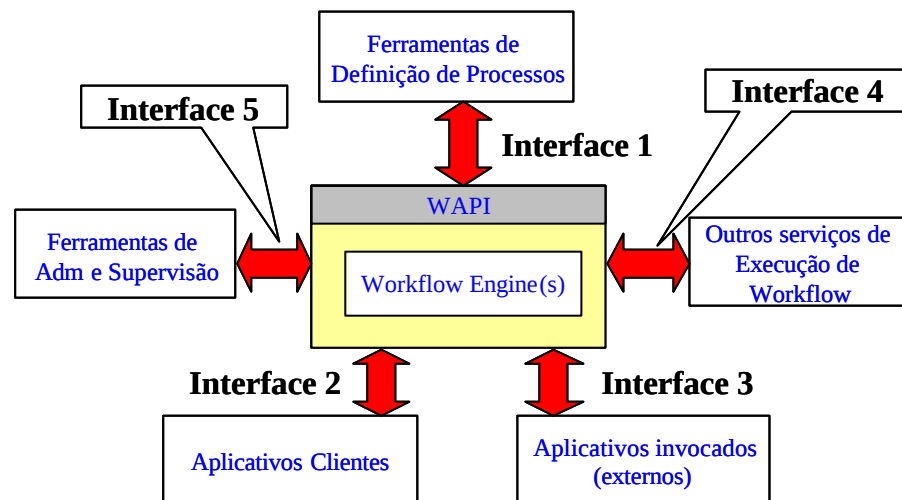


Figura 5 - Modelo de referência da WfMC

Cada interface tem sua própria característica e segue padrões definidos pela WfMC, sendo :

2.9.1 Interface 1

A interface 1 define a comunicação entre as ferramentas de definição do processo e o servidor de workflow. Ela é chamada de interface de importação/ exportação das

definições de processo de workflow e pode ser composta por um formato comum de intercâmbio ou por um conjunto de funções ou chamadas ao sistema, conhecida como API, os quais permitem que a definição de processo gerada por uma ferramenta de modelagem possa ser usada por diferentes produtos de workflow.

Envolve aspectos da definição do processo. Desenvolve uma linguagem padrão de definição de processos (WPDL) a ser suportada pelas ferramentas de definição de processo.

2.9.2 Interface 2

Para esta interface, a WfMC define um conjunto de APIs (WAPI –Workflow Application Programming Interface), as quais podem ser utilizadas por uma aplicação de Workflow para acessar o servidor e as listas de tarefas a fazer (worklists), independentemente da natureza de implementação do produto. Para isto, são providos comandos que operam as instâncias de processos, assim como manipulam as listas de tarefas a fazer.

Esta interface envolve os aplicativos clientes de workflow. Ela padroniza as chamadas do gerenciador da listas de trabalho (Ex.: recuperar um item de trabalho, declarar o início e o término da atividade, etc.)

2.9.3 Interface 3

Em algum estágio de um processo executado em um WfMS, pode ser necessária a intervenção humana, como já visto. Quando isso ocorre, geralmente há a necessidade da utilização de aplicações externas ao WfMS, que produzam dados, documentos ou outro tipo de informação que seja necessária à conclusão do processo. A interface 3 define a comunicação entre o servidor de workflow e uma larga faixa de aplicações de ambientes

heterogêneos que devem ser integrados ao sistema de workflow. Além disso, aplicações desenvolvidas especificamente para workflow também são invocadas através desta interface.

Esta interface trata da comunicação como aplicações externas, invocadas para realização de atividades específicas.

2.9.4 Interface 4

Esta interface tem como objetivo permitir que WfMS de diferentes “fabricantes” possam interagir, passando itens de trabalho, processos, atividades, entre outros elementos de workflow, entre si, e cooperar para a execução de um mesmo processo.

Há de se notar que, para esse modelo dar certo, necessita-se de que todos os sistemas de workflow integrados possam interpretar uma versão padronizada de uma definição de processos e, além disso, necessitam compartilhar os mesmos dados de controle de workflow, mantendo assim uma visão comum dos estados dos processos durante sua execução.

Esta interface provê a habilidade de se transferir partes de um processo para execução em um outro servidor de workflow. Trata da comunicação entre diversos WfMS envolvendo a administração de partes do mesmo processo.

2.9.5 Interface 5

Esta interface visa possibilitar que ferramentas de administração e monitoramento de sistemas de workflow de desenvolvedores diversos possam funcionar com diferentes WfMS. Ela visa também permitir uma visão completa do status do trabalho, fluindo através da organização, não importando em qual sistema ele está rodando.

A interface 5 vai incluir chamadas específicas à WAPI para manipular questões administrativas do workflow, inclusive em nível de segurança e autorizações. A idéia, então, é ter ferramentas de administração operando independentemente dos WfMSs que se possa ter. Trata da administração e monitoramento dos processos executados.

2.9.6 WAPI

A **WAPI** – Esta interface consiste de uma série de construções pelas quais os serviços de execução de workflow podem ser acessados.

2.10 Vantagens e benefícios

A modelagem correta dos processos de uma organização e a sua correspondente implementação através de sistemas de gerência de workflow podem trazer diversas vantagens. A seguir, enumeramos alguns pontos benéficos na implantação de um sistema de gerenciamento de workflow.

- ? *Garantir a integridade dos processos:* muitas vezes, normas e procedimentos da organização não são praticados ou o são apenas parcialmente. Vários fatores podem ser apontados como a causa desses problemas, porém, o uso de um WfMS garante que as regras dos processos não serão desrespeitadas, e que a ordenação das atividades será seguida.
- ? *Diminuição do tempo de resposta:* geralmente quando o processo da organização é manual, há um espaço de tempo relativamente alto na transferência de tarefas de um participante para outro. Como no uso de um WfMS a coordenação entre

tarefas é administrada automaticamente, assim o processo pode fluir mais rapidamente.

- ? *Integração das atividades da empresa:* com o uso de sistemas de workflow, atividades desenvolvidas por divisões diferentes da empresa podem ser conectadas, compondo um único processo. Um problema grave nas empresas, atualmente, é serem vistas como uma coleção de departamentos dispersos, e não como uma entidade única.
- ? *Manipulação eletrônica de documentos:* Com o uso de um WfMS para manipular documentos, é possível trabalhar totalmente no meio eletrônico, substituindo totalmente o registro em papel, evitando assim problemas como: armazenamento, consistência da informação, não cumprimento das regras e etc.

2.11 Conclusões

Neste capítulo abordamos os conceitos mais relevantes da tecnologia de workflow buscando uma compreensão dos seus objetivos e mostrando seus novos paradigmas. Os Sistemas de workflow se inserem no contexto geral de software cujo objetivo é o suporte ao trabalho cooperativo, onde se enfatiza a interação entre usuários, e não apenas a interação usuário/sistema.

Esta área tornou-se muito interessante pela grande variação apresentada, na prática, pelos processos, devido a diferenças de cultura empresarial, de grupo de trabalho e de agentes executores. Tal diferença se dá pela existência de casos especiais e pela evolução natural dos processos no decorrer do tempo. Em decorrência disto, novos paradigmas devem ser utilizados, em detrimento da divisão tradicional entre modelagem e execução.

Uma conclusão à qual se chega é que, como essa tecnologia se mostra em plena evolução, ainda há algumas lacunas nos conceitos e indefinições de padrões.

Atribuir aos sistemas computacionais um certo grau de autonomia e inteligência tem sido um desafio constante para os pesquisadores da área de informática nos dias atuais. Movidos por este desejo, muitos dos pesquisadores da área de desenvolvimento de sistemas buscam unir a tecnologia de Agentes de Software ao mundo dos sistemas de gerenciamento de workflow, acreditando assim, atribuir aos WfMS maior flexibilidade e autonomia.

3 AGENTES DE SOFTWARE

Apresentamos, neste capítulo, uma visão geral da tecnologia de agentes, em especial os agentes móveis. São discutidos conceitos básicos, características, aplicações, vantagens e desvantagens de sistemas baseados nessa tecnologia.

3.1 Introdução

A cada dia, os desenvolvedores buscam uma forma de embutir nas máquinas reações que as tornem semelhantes ao homem. Características como autonomia nas decisões, comportamento social e capacidade de aprender, são os maiores desafios para que o sonho de ver uma máquina agir como um homem se torne realidade (NWANA,1996).

Estamos vivendo neste momento uma introdução a um outro paradigma de desenvolvimento chamado “baseado a agentes”. Tal paradigma introduz o conceito de unidades computacionais com certo grau de autonomia para a tomada de decisão, bem como se adaptar ao meio em que vive, manter um relacionamento com outras unidades para alcançar o objetivo comum e até aprender algo novo para sua base de conhecimento.

3.2 O que são agentes de software ?

Segundo Wooldridge (WOOLDRIDGE,1995), não há definição universalmente aceita do termo agente, mas há um consenso geral de que a autonomia é a idéia central da noção de agência. Assim ele define: [...] “Um agente é um sistema computacional que está situado em algum ambiente, e que é capaz de ações autônomas neste ambiente visando atingir seus objetivos”.

Esta definição se aplica tanto a agentes humanos como a robôs e agentes de "software". Em (RUSSEL,1995), os autores definem um agente como algo que obtém conhecimento do seu ambiente através de sensores e atua nesse ambiente através de efetadores.

Uma outra definição, que pretende sintetizar o que possa haver de comum em todas as outras definições, diz que agentes de "software" são entidades computacionais persistentes e ativas que percebem, raciocinam, agem e comunicam-se em um ambiente (JENNINGS, 2000).

O grau de sofisticação de um agente está diretamente ligado à forma como ele seleciona as ações a tomar. O agente pode simplesmente reagir a eventos que ocorrem no mundo externo, utilizando um conjunto de regras "situação-ação" – é o caso extremo de um *agente reativo*.

A tomada de decisão pode implicar em um conhecimento prévio do mundo, que será memorizado pelo agente. Além disso, o agente pode ser motivado por objetivos, fazendo a escolha da ação depender do destino pretendido.

Uma noção de agente mais elaborada inclui na sua definição atitudes de informação acerca do mundo (conhecimento e crença) e pró-attitudes (intenções e obrigações) que guiam as ações do agente. Este tem a chamada arquitetura "BDI", onde a estrutura tripla <crença (*belief*), desejo (*desire*), intenção (*intention*)> toma um papel ativo no processo cognitivo do agente (JENNINGS,2000).

A arquitetura típica de um agente divide-se, de forma mais ou menos evidente, em duas partes:1) o *sistema inteligente*: parte responsável por resolver os problemas do domínio da sua especialidade, e 2)*camada de cooperação*: parte que permite ao agente comunicar e interagir com outros agentes do sistema em que está inserido.

O *sistema inteligente* é o componente que dá ao agente uma característica autônoma na resolução de problemas, capaz de tomar decisões inteligentes no que diz respeito à execução das suas tarefas. A *camada de cooperação* permite que os agentes troquem mensagens entre si, buscando uma interação e cooperação para que seja alcançado um objetivo.

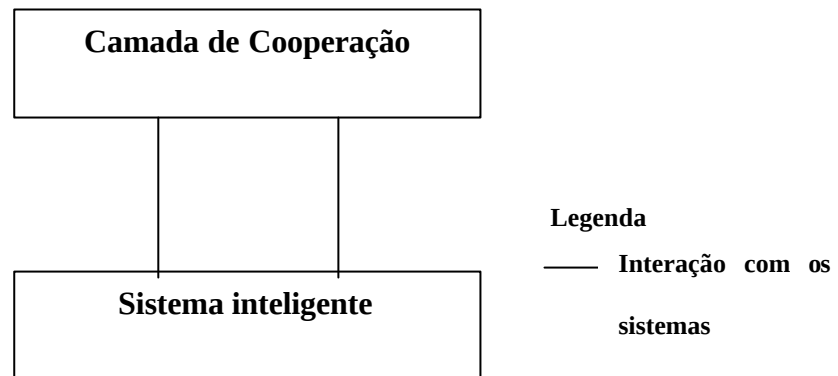


Figura 6 - Arquitetura básica de um agente

3.3 Características

Um Agente caracteriza-se por adquirir algumas habilidades que de fato os tornam diferentes dos paradigmas até então existentes. Dentre as características mais comuns dos agentes, citamos :

? Autonomia

Esta característica atribui ao agente a capacidade de agir sem a intervenção externa de humanos, mantendo assim um certo grau de controle sobre suas próprias ações e seus estados internos. Wooldridge e Jennings (ODGERS,1999a) relaciona o controle somente ao usuário final quando define que, um agente inteligente deve possuir a habilidade de praticar ações para desenvolver tarefas ou alcançar objetivos, sem necessitar da interferência do usuário final. Wooldridge (WOOLDRIDGE,1995), acrescentam que o agente, além de

possuir controle sobre seu comportamento, deve também possuir controle sobre seu estado interno.

? **Reatividade**

É a característica pela qual um agente percebe seu ambiente e responde de maneira adequada às mudanças que nele ocorrem. O ambiente pode ser um mundo físico, uma interface gráfica, uma coleção de outros agentes, a Internet, ou uma combinação desses elementos. Essa característica denota um comportamento deliberativo. Agentes deliberativos utilizam um modelo simbólico do mundo representado de forma explícita, para tomar decisões por meio de raciocínio lógico (ou pseudo-lógico), baseado na correspondência de padrões e manipulação de símbolos. Wooldridge e Jennings (WOOLDRIDGE,1995) definem a reatividade como sendo a propriedade que permite aos agentes perceberem seus ambientes e responderem adequadamente às mudanças neles ocorridas.

? **Pró-atividade:**

Agentes não apenas reagem ao ambiente, mas também são capazes de apresentar um comportamento dirigido por objetivos ao tomar iniciativas. Wooldridge define a pró-atividade como a capacidade manifestada pelo agente de exibir um comportamento direcionado a objetivos. Desta forma, o agente não age simplesmente em resposta ao ambiente mas sim, de acordo com um propósito. Para tanto, deve exibir um comportamento oportunístico, voltado para a realização de seus objetivos. Ser dirigido por objetivos significa ser intencional. O princípio da intencionalidade, no contexto de sistemas multiagentes, foi formalizado em (COHEN,1990). Portanto, que ser pró-ativo é uma consequência de *ter intencionalidade*, ou seja, por ser dirigido por objetivos, um agente também pode tomar iniciativas.

? **Habilidade social:**

Agentes devem ser capazes de interagir quando for apropriado para completar suas tarefas ou auxiliar outros agentes. Isso requer que os agentes tenham meios de comunicar suas necessidades e um mecanismo para decidir quando interações sociais são convenientes.

? **Mobilidade**

A característica de mobilidade dá ao agente a capacidade de migração adaptativa de um nó a outro de uma rede. Em cada nó, o agente interage com os agentes estáticos e com outros recursos para realizar suas tarefas. Esta característica tem sido bastante explorada nos sistemas atuais, principalmente quando executado sobre a plataforma da Internet. Por exemplo nos sistemas de comércio eletrônico. Nas próximas seções, detalharemos esta característica, uma vez que estaremos tratando especificamente dos *Agentes Móveis*.

Consideramos que as três primeiras características citadas acima sejam o mínimo necessário para que um programa possa ser classificado como um agente. As características restantes dizem respeito a facilidades que podem ou não estar presentes, dependendo principalmente do ambiente de atuação do agente. Acreditamos também que requisitos de aprendizagem podem ser introduzidos atendendo às exigências mínimas para que um agente possa rotular-se inteligente. Embora muitos agentes desenvolvam atividades sob a instrução de pessoas (por exemplo, uma busca na WEB por preços mais baixos para um determinado produto), muitos desenvolvem suas ações, baseados não em um pedido externo de um usuário, mas baseados em seu conjunto de regras e conhecimento. Esses agentes normalmente realizam tarefas de manutenção de sistemas, como atualização de bases de dados, verificação de sistemas de arquivos ou monitoramento de recursos críticos.

3.4 Agência

Conforme se verifica em muitas literaturas, denomina-se agência o conjunto de características que definem um software agente e o diferencia das demais aplicações. Segundo Gilbert (GILBERT,1996), o agente define-se em um espaço tridimensional, construído por :

- ? **Agência** - O grau de agência é intensificado à medida que o agente representa o usuário e apresenta capacidade de interagir com : dados, aplicação, serviços e agentes.
- ? **Inteligência** – é a habilidade do agente em acessar as declarações de objetos do usuário e executar as tarefas a ele delegadas. Níveis mais altos de inteligência são alcançados à medida que os agentes apresentam as seguintes características : Preferência, Raciocínio, Planejamento e Aprendizagem.
- ? **Mobilidade** - No plano da mobilidade os agentes podem apresentar as características de Estáticos e Móveis.

3.5 Agentes x Objetos

Segundo Odell (ODELL,1999), o conceito de agente apresenta semelhanças estruturais em relação ao conceito de objetos. O fato de que um agente é uma entidade que possui capacidades comportamentais e conhecimento privado, e que um objeto também possui esta mesma estrutura, demonstra características comuns entre estes conceitos. Assim, é possível afirmar que o que difere um agente de um objeto é fundamentalmente o conceito de autonomia.

Odell traça um paralelo entre os conceitos de agente e objeto mapeando o estado mental do agente para o estado interno do objeto e as capacidades comportamentais do agente para os métodos de um objeto.

Podemos concluir que os Objetos podem ser definidos como entidades de software que encapsulam algum estado, são hábeis para executar ações, ou métodos sobre seus estados e comunicam-se por passagem de mensagens. As similaridades são óbvias : modularidade e controle sobre seu estado interno, as diferenças, no entanto, parecem mais fortes (WEISS,1999).

- ? Agentes incorporam uma noção mais forte de autonomia que os objetos. Em particular, decidem por eles mesmos se executam ou não uma ação a pedido de outro agente;
- ? Agentes são capazes de comportamento flexível (reatividade, pró-atividade e sociabilidade) e o modelo padrão de orientação a objetos não diz nada sobre este comportamento;
- ? Um sistema multiagentes é inerentemente *multi-thread*, em que cada agente controla, pelo menos, uma *thread*.

Nada impede que agentes sejam programados em linguagens orientadas a objetos. Existem templates e pacotes para programação de agentes em linguagens como JAVA (ex. JAT - Java Agent Template). Ou seja, podemos tirar proveito das similaridades para explicitamente programar as características de agentes não previstas no modelo de orientação a objetos.

3.6 Classificação de agentes

Há diferentes tipos de classificação de agentes. Os agentes podem atuar de forma isolada (standalone) ou atuar como membro de uma comunidade de agente nos chamados sistemas multiagente (MAS). Diferentes dimensões definem a classificação do agente. Conforme as tarefas a serem executadas, os agentes podem ser:

- ? *Agente usuário*: especialmente projetado para ajudar o usuário ou, em alguns casos, atuar em seu nome. O Agente deve ter capacidade de se comunicar com o usuário, entender os seus interesses, suas preferências e seus hábitos.
- ? *Agente Serviço*: de forma genérica, é aplicado às tarefas, assim como recuperação de informações, indexador de web, etc.

Conforme a localização do agente, eles podem ser:

- ? *Estacionário*: Executado e atuando apenas no computador local do usuário ou em um servidor. Este tipo de agentes não desenvolve as habilidades de mobilidade.
- ? *Móveis*: Os Agentes Móveis são abstrações de software que podem migrar entre os computadores de uma rede representando os usuários em várias tarefas.

3.7 Linguagem de comunicação

A tecnologia de agentes de software despertou o interesse como ferramenta para criação de um novo modelo para sistemas de software complexos. No projeto de agentes, misturam-se muitas das propriedades tradicionais de inteligência artificial (como "raciocínio" no nível de conhecimento, flexibilidade, etc.) com as experiências obtidas nas áreas de sistemas distribuídos, teorias sobre negociação e teorias de equipe de trabalho. Para que a cooperação entre agentes de software tenha sucesso, é requerida comunicação entre eles. Uma coleção de agentes trabalhando juntos em cooperação pode ser vista como uma pequena sociedade, e o funcionamento de qualquer sociedade coerente precisa de uma linguagem comum e um meio de comunicação. Para que os atos de comunicação e a cooperação entre agentes sejam possíveis, precisa-se de uma Linguagem de Comunicação entre Agentes (em inglês: Agent Communication Language, ou ACL). Dentro de uma ACL, torna-se importante a forma como as mensagens são comunicadas, isto é, se as mensagens expressam

adequadamente seu propósito sob um ponto de vista semântico. Na continuação, será mostrado um breve resumo de algumas linguagens de comunicação entre agentes.

O projeto de uma linguagem de comunicação entre agentes utiliza normalmente uma das abordagens: procedural ou declarativa. Na abordagem procedural, a comunicação acontece através de diretivas. Tanto comandos individuais quanto programas completos podem ser transmitidos e executados no lado receptor. Linguagens baseadas em *scripts*, tais como TCL e *Telescript* (WHITE,1997), são exemplos da abordagem procedural. Na abordagem declarativa, a comunicação ocorre através da troca de estruturas declarativas, tais como definições, asserções e outras.

3.7.1 A linguagem de Comunicação KQML

Knowledge Query and Manipulation Language (KQML) é uma linguagem versátil de propósito geral que suporta a comunicação entre vários agentes com um conjunto de primitivas reservadas chamadas *performative*. A linguagem KQML adotou o termo *performative* para designar as primitivas das mensagens. A definição deste termo se refere a uma declaração feita pelo usuário emissor e que é executada pelo agente receptor, simplesmente porque o emissor a declara ou afirma. A KQML, descrita por Tim Finin em (FININ,1997), é o resultado de pesquisas feitas pelo grupo *Knowledge Sharing Effort* (KSE), cuja iniciativa teve por objetivo desenvolver fundamentos para interação e interoperabilidade em sistemas de software.

KQML adota o uso de ontologias. Ontologias são um conjunto de especificações explícitas de significado (FININ,1997), conceitos e relacionamentos aplicáveis a algum domínio específico. Desta forma, pode-se assegurar que dois agentes estejam utilizando de um mesmo “vocabulário” durante o processo de comunicação. Em outras palavras, a

utilização de uma ontologia permite a definição de um contexto único, eliminando-se a ambigüidade.

As mensagens KQML codificam a informação em três diferentes níveis arquiteturais : conteúdo, mensagem e comunicação. Sua sintaxe está baseada na linguagem Lisp e é composta de uma ação (*performative*) e parâmetros. A ordem de posicionamento dos parâmetros não é importante. Os parâmetros são codificados como pares : <palavra-chave, valor>, onde a palavra-chave é precedida pelo símbolo ":".

Uma característica interessante da linguagem KQML é que ela pode oferecer uma forma de acesso à informação, mesmo para programas que não sejam agentes.

3.8 Agentes móveis

No início dos anos 80 surgiu o conceito de RPC (*Remote Procedure Call*) (BIRREL,1984) dando origem a um novo paradigma de comunicação. Esse conceito fundamentou o modelo arquitetural dos sistemas distribuídos, que ficou conhecido como *modelo cliente/servidor*. Neste modelo, o cliente invoca procedimentos que estão armazenados no servidor, enviando-lhes os argumentos necessários para execução. Após o término da execução do procedimento, o servidor devolve ao cliente os resultados, e só então continua seu processamento, o que caracteriza um modo de comunicação síncrono.

Mais tarde, surgiu o conceito de Avaliação Remota (*Remote Evaluation*) (STAMOS,1990), que combinou RPC com a idéia de mensagens executáveis (objetos, por exemplo). Nesse processo, a idéia que prevalece é a de transferência de código e não somente de argumentos. Variações dessa idéia originaram o conceito de *Código por Demanda*, em que são transportadas referências a um código base² que o servidor fornece quando requisitado.

² O código base especifica as localizações das classes usadas por uma aplicação.

Applets e Servlets Java são exemplos de implementações desse paradigma.

Em 1994, como extensão dessas idéias, surgiu o paradigma dos Agentes Móveis, fruto de pesquisas que gerariam o sistema Telescript. O termo “*agente móvel*” foi cunhado com a publicação do artigo intitulado “*Mobile Agents*” (WHITE,1997), que apresentou a arquitetura do sistema Telescript como exemplo de aplicação desse paradigma.

As idéias apresentadas nesse artigo incentivaram as pesquisas na área em todo o mundo. Mais tarde, surgiu o sistema Agent Tcl (GRAY,1995), atualmente denominado D’Agents (GRAY,1998). Esse sistema é baseado em uma linguagem baseada em scripts chamada Tcl (OUSTERHOUT)(WELCH,1995). Depois vieram os sistemas baseados na linguagem Java (GOSLING,1997), tais como os sistemas Mole (STRAFIER,1996), da Universidade de Stuttgart, Aglets da IBM (LANGE,1998) e o sistema Ajanta (KARNIK,1998), entre outros.

Atualmente, a maioria dos sistemas existentes e dos esforços de pesquisa estão voltados para o uso da linguagem Java, principalmente por razões de portabilidade, popularidade e segurança, o que gera aceitação comercial. Existem também os sistemas projetados para suportar a execução de agentes escritos em várias linguagens, tais como os sistemas Tacoma (GRAY,2000) e Ara (ROSENCHEIN,1985).

Em 1997, foi submetido à apreciação da OMG (*Object Management Group*) um documento que constitui uma proposta para padronização de sistemas de agentes móveis. Essa proposta ficou conhecida como MAF (*Mobile Agent Facility*) e objetiva facilitar a interoperabilidade entre os sistemas baseados em agentes móveis. Descrevemos a seguir o modelo conceitual implementado pelos sistemas baseados em agentes móveis.

A execução de um agente móvel não fica restrita ao sistema em que ele foi iniciado. Agentes móveis podem se transportar para um outro sistema de agentes no qual esteja contido um objeto com o qual o agente móvel precisa interagir. Desse modo, é possível

tirar vantagem do fato de poder interagir localmente com esse objeto, utilizando seus serviços.

Um agente móvel possui capacidades e requerimentos que muitos sistemas de objetos distribuídos não suportam no momento, como o suporte a protocolos de comunicação específicos, por exemplo. Os mecanismos tradicionais,³ utilizados pela maioria dos sistemas distribuídos, não atendem às necessidades de comunicação dos sistemas baseados em agentes, e em particular dos agentes móveis.

Quando um agente viaja, seu código e estado são transferidos junto com ele. O estado de um agente pode ser entendido como sendo os valores dos atributos do agente em um dado momento. Esse estado pode ser salvo para ajudar o agente a determinar *o que* fazer quando sua execução for interrompida quando o agente for transferido para um outro sistema, e a execução for retomada no local de destino. Alguns sistemas baseados em agentes móveis utilizam o mecanismo de Serialização/Desserialização para captura do estado de execução de um agente.

Serialização é o processo em que os valores dos atributos do agente são armazenados de maneira serial, de modo que possam ser restaurados mais tarde pelo processo inverso, o da Desserialização. Por razões de segurança, os objetos serializados são identificados e verificados durante o processo de Serialização/Desserialização. Os sistemas que empregam esse método para captura de estado, como os sistemas baseados em Java, por exemplo, não dão suporte à captura dos valores das pilhas e do contador de programa.

Um *sistema de agentes* é uma plataforma que pode criar, interpretar, executar, transferir e terminar agentes. A cada agente pode ser associada uma *autoridade*, a qual poderá ser autenticada mais tarde como parte da implementação de políticas de segurança. O mesmo ocorre com um sistema de agentes. Por exemplo: um sistema de agentes com autoridade X implementa as políticas de segurança de X na proteção de seus recursos. Um sistema de

³ RPC (*Remote Procedure Call*) ou sistemas de objetos distribuídos, como CORBA (*Common Object Request Broker Architecture*), DCOM (*Distributed Common Object Module*) e RMI (*Remote Method Invocation*).

agentes é identificado, unicamente, por seu nome e endereço e em uma mesma máquina, pode haver um ou mais sistemas de agentes executando simultaneamente (OMG,2000).

O *tipo de um sistema de agentes* descreve o perfil dos agentes do sistema. Por exemplo, se o sistema de agentes é do tipo Aglet suporta Java como linguagem de agentes, usa a abstração de Itinerário para viajar e utiliza o método Serialização de Objetos Java. Portanto, para se obter a função de um sistema de agentes, é necessário especificar o perfil do agente (tipo do sistema de agentes, linguagem e método de serialização empregado) para identificar unicamente a funcionalidade desejada (OMG,2000).

Quando um agente decide migrar de um local para outro, ele só poderá viajar por entre ambientes de execução chamados *places*. Um place é um *contexto*, dentro do sistema de agentes, no qual um agente pode executar. Esse contexto é responsável por prover funções básicas, tais como controle de acesso e execução dos agentes. A Figura 7 ilustra o modelo conceitual genérico de um sistema baseado no paradigma de agentes móveis.

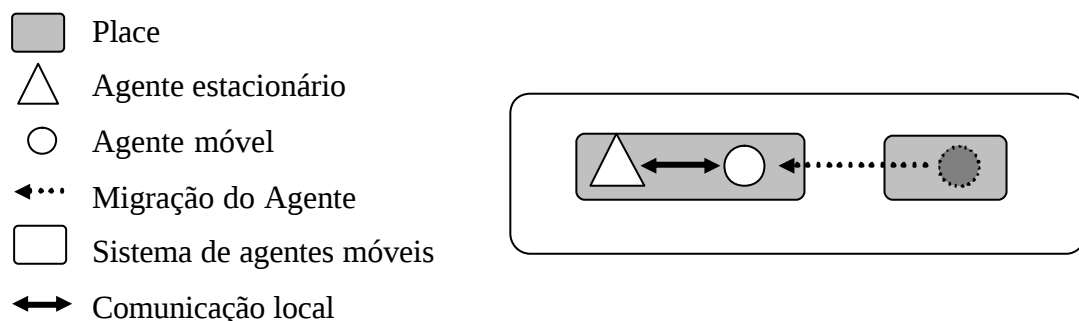


Figura 7 - Modelo conceitual dos agentes móveis.

Os places de origem e de destino podem residir no mesmo sistema de agentes, ou em sistemas diferentes que suportem o mesmo perfil de agente. Uma visão em camadas do relacionamento entre places, o sistema de agentes, o sistema operacional e a máquina é mostrada na Figura 8

A cada place é associada uma localização, que consiste do nome do place e do endereço do sistema de agentes no qual ele está inserido. Um sistema de agentes pode conter um ou mais places e cada um deles pode conter um ou mais agentes executando. Quando for solicitada a localização de um agente, será fornecido o endereço do place onde este agente está executando.

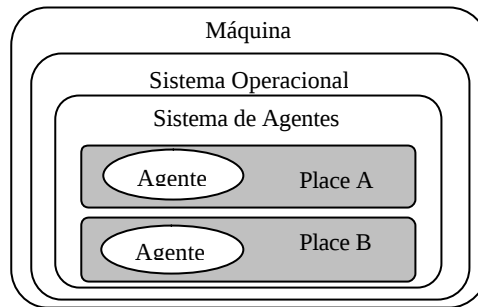


Figura 8 - Sistema de agentes

Uma *região* é um conjunto de sistemas de agentes que possuem a mesma autoridade, isto é, representam a mesma pessoa ou organização (OMG,2000). Esse conceito permite a distribuição da carga por entre vários sistemas de agentes. Também podemos entender uma região, como sendo a *rede lógica* formada pelos places (KARNIK,1998), como mostra a Figura 9.

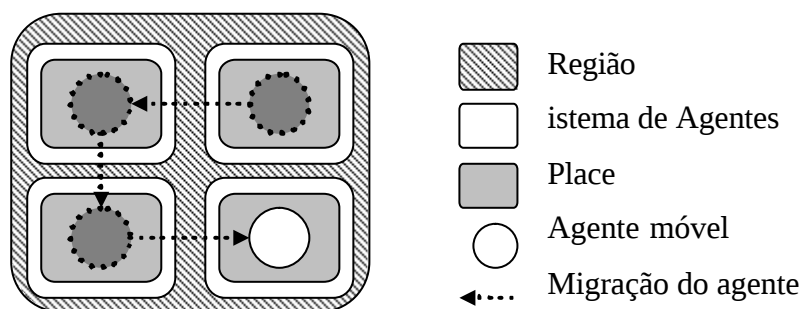


Figura 9 - Conceito de região

A seguir veremos as principais características de um agente móvel e apresentaremos algumas vantagens de sua utilização.

3.9 Características e Vantagens dos Agentes Móveis

A principal característica de um agente móvel é sua capacidade de operação *assíncrona* e *autônoma*. Isso significa que ele não precisa de uma conexão permanente com a rede. As tarefas a serem executadas podem ser encapsuladas no próprio agente, que então é lançado na rede, onde as executará de forma autônoma. O dispositivo que enviou o agente poderá desconectar-se, só voltando a conectar-se para receber os resultados das tarefas executadas. Essa característica reduz os custos de comunicação, pois os requisitos de largura de banda e latência da rede podem ser reduzidos. Também provê suporte de tolerância a falhas, pois podem decidir usar servidores alternativos, uma vez que são os agentes que decidem para *onde* e *quando* migrar. Tais características os torna mais flexíveis e adaptáveis às mudanças de seu ambiente.

Um agente móvel é capaz de encapsular seus próprios protocolos, o que permite que novos protocolos sejam instalados automaticamente, e que sejam utilizados somente quando necessários para uma interação particular. Essa característica permite que as tarefas encapsuladas em um agente possam ser executadas em qualquer local da rede. Embora tais características possam ser alcançadas por outros meios, o paradigma dos agentes móveis reúne-as de forma mais clara para os projetistas de sistemas distribuídos, pois eles apresentam um modelo conceitual simples e genérico que torna fácil projetar, implementar e manter sistemas distribuídos (LANGE,1999).

Um aspecto muito importante a ser considerado quando se pensa em agentes móveis é a segurança das aplicações. Aspectos gerais relacionados à segurança de sistemas baseados em agentes móveis serão discutidos a seguir.

3.10 Exemplos de Sistemas de Agentes Móveis

Algumas plataformas para suportar sistemas baseados no paradigma de agentes móveis foram desenvolvidas. Entre elas :

Telescript: sistema desenvolvido pela General Magic, em 1995, que utiliza uma linguagem orientada a objetos para programação de agentes, também chamada Telescript. Essa linguagem foi inspirada em C++ e Smalltalk (GOLDBERG,1983). Ela implementa conceito de classes e os mecanismos de herança múltipla, tratamento de exceções, e oferece uma biblioteca de classes para a programação de agentes móveis. Esse sistema foi o pioneiro na proposição do modelo de agentes móveis como uma alternativa para o desenvolvimento de aplicações distribuídas.

Telescript teve como principal objetivo disponibilizar comercialmente uma infraestrutura de software que induzisse ao desenvolvimento de aplicações para comércio eletrônico. O sistema oferece recursos bastante complexos para migração e segurança de agentes e para controle de acesso a recursos. Não foi um projeto comercialmente bem sucedido, principalmente pelo fato de exigir o aprendizado de uma nova linguagem em uma época em que as atenções do mercado estavam voltadas para a linguagem Java. Por este motivo, o projeto foi cancelado pela General Magic, que logo em seguida lançou o sistema Odyssey (NOBLE,1999), baseado em Java (VALENTE,2000).

Aglets: consiste em uma biblioteca de classes, desenvolvida pela IBM, para a implementação de agentes móveis. Os aglets inspiram-se no modelo Applets Java. O modelo de segurança dos aglets (KARJOTH,1997) é semelhante aos modelos dos sistemas Ara.

Ara (*Agent Remote Action*): é um projeto do *Distributed Systems Group*, da Universidade de Kaiserslautern, na Alemanha. Ara suporta agentes escritos em Tcl, Java e C/C++. Os agentes C/C++ são compilados para um tipo de bytecode para prover

portabilidade. A arquitetura do sistema é constituída por um núcleo e vários processos. Cada agente é executado como um processo independente e protegido. O servidor de agentes e os interpretadores das linguagens formam uma única aplicação executando sobre a camada do sistema operacional. Novos interpretadores podem ser acoplados. A segurança é reforçada em todas as ações de um agente, provendo autenticação, autorização e contagem do uso de recursos. Os mecanismos de segurança ainda estão sendo aperfeiçoados.

3.11 Aplicações

O paradigma de agentes móveis tem sido empregado principalmente em aplicações nas áreas de Comércio Eletrônico, Recuperação de Informações Distribuídas, Sistemas de Gerenciamento de Workflow, Disseminação de Informações, Processamento Paralelo e Gerenciamento de Redes.

Comércio Eletrônico: sem dúvida, é uma das áreas mais promissoras. Em um ambiente de comércio eletrônico, determinadas transações exigem que recursos remotos sejam acessados quase que em tempo-real. O paradigma dos agentes móveis mostra-se bastante adequado a aplicações dessa natureza, uma vez que um agente pode ser transferido para onde se encontra o recurso, e lá, interagir localmente.

Dessa maneira, as transações podem ser efetuadas com maior rapidez. Agentes representando usuários podem ser enviados a um local especial, uma espécie de *mercado virtual*, onde poderão negociar em benefício de seus usuários, fazer levantamento de preços, analisar propostas de negócios, manipular informações, etc. Uma aplicação que está sendo desenvolvida como um modelo conceitual para plataformas de comércio eletrônico é o sistema Tabican (LANGE, 1998). Essa plataforma está sendo desenvolvida pela IBM, utilizando Aglets.

Uma outra aplicação que explora os conceitos de mercado eletrônico e negociação é a aplicação desenvolvida pela Tryllian, o sistema *Gossip Tryllian*, 1999. Embora as negociações no sistema *Gossip* sejam destinadas a busca de informações, há uma tendência de que sejam expandidas para outros tipos de aplicações mais adequadas, como mercados eletrônicos, por exemplo. Aspectos relacionados à segurança não podem ser negligenciados em aplicações dessa natureza. Alguns requerimentos básicos de segurança para mercados eletrônicos são discutidos em (ZAPF,1998).

Recuperação de Informações Distribuídas: agentes móveis são particularmente atraentes em aplicações que requerem recuperação de informações distribuídas. Ao invés de enviar grandes quantidades de dados através da rede para extrair as informações necessárias no lado cliente, um agente móvel, incorporando algoritmos para extrair dados específicos, pode ser enviado para as fontes de informação remotas, onde a informação poderá ser extraída localmente.

Uma das maiores vantagens do emprego dessa tecnologia é a eliminação da transferência de resultados intermediários através da rede, o que diminui o tráfego em problema não solucionado nas arquiteturas dos sistemas de recuperação tradicionais. Exemplos de aplicações de agentes móveis em Recuperação de Informação são encontrados em (BREWINGTON,1999)(THEILMANN,1999)(TRYLLIAN,1999) e no serviço que propomos nesta dissertação.

Gerenciamento de Sistemas de Workflow: em aplicações de *workflow*, o fluxo de dados entre os participantes caracteriza o processamento da informação. Agentes móveis são especificamente úteis, porque provêm mobilidade, comportamento, autonomia e informação sobre o *workflow*. Os agentes ainda podem fornecer todo o código necessário para acessar os itens de informação de uma maneira semanticamente correta.

Disseminação de Informações: agentes móveis podem ser efetivamente empregados para disseminar informações para um dado número de consumidores, tais como notícias e artigos,. Esses agentes podem encapsular uma série de políticas, tais como aquelas que permitem que um determinado agente só possa ter acesso a uma dada informação após haver pago por ela. Algumas aplicações nesta área podem ser encontradas em (KAY,1998).

Processamento Paralelo: certos agentes móveis podem criar clones em cascata na rede. Um dos usos potenciais dessa tecnologia é a administração de tarefas que englobam processamento paralelo. Se uma aplicação requer muito poder de processamento, ela deve ser distribuída entre vários processadores. A infraestrutura proposta por agentes móveis mostra-se bastante plausível para alocação de recursos (WHITE,1998a).

Gerenciamento de Sistemas de Comunicação e Redes de Computadores: devido à sua inerente complexidade, os sistemas de comunicação e as redes de computadores requerem estratégias de gerenciamento cada vez mais sofisticadas para garantir níveis de performance e confiabilidade adequados. Nessas aplicações, o grande problema do uso de abordagens baseadas em modelos de gerenciamento centralizados, como SNMP (*Simple Network Management Protocol*), por exemplo, é a escalabilidade.

O uso de agentes móveis apresenta-se como uma alternativa para resolver problemas dessa natureza. Essa tecnologia permite que seja utilizada uma estratégia descentralizada para monitorar dispositivos ou aguardar por eventos específicos, efetuando alguma ação quando tais eventos ocorrem. O uso de agentes móveis permite maior flexibilidade a tais aplicações, devido à possibilidade de execução assíncrona e autônoma. Alguns exemplos de aplicação nesta área podem ser encontrados em (BALDI,1997)(BALDI,1998)(WHITE,1998b)(PULIAFITO,2000). Aspectos relacionados à segurança em aplicações de gerenciamento são discutidos em (REISER,2000). Exemplos de aplicação de agentes móveis em sistemas de telecomunicações podem ser encontrados em

(HAYZELDEN,1999) (QI,2000).

Outras áreas de aplicação de agentes móveis das quais temos conhecimento são: Acesso a Banco de Dados Distribuídos na Web (SAMARAS,1999) e Ambientes Cooperativos (CASTILLO,1998) (DIKAIKOS,1999).

3.12 Conclusão

Neste capítulo abordamos a tecnologia dos Agentes de Softwares buscando apresentar os conceitos mais relevantes e algumas das características que tornam esta tecnologia bastante atraente no desenvolvimento de sistema. Concentramos uma atenção maior ao detalhamento das características de mobilidade que possibilita a um agente de software se locomover de uma máquina para outra levando consigo, além do seu *status* corrente, o seu próprio código fonte, possibilitando uma certa auto-reprodução em ambientes distribuídos.

Ainda neste capítulo, discutimos algumas das razões que fizeram com que esta tecnologia seja bastante pesquisada, mostramos as suas vantagens, citamos alguns sistemas baseados nesse paradigma e mostramos algumas áreas bastante promissoras para o emprego dessa tecnologia.

Diante do exposto neste capítulo, podemos verificar que a tecnologia de agentes de software, principalmente os agentes móveis, está presente em muitas áreas do desenvolvimento de sistemas e com tantas características favoráveis ao ambiente destituído e dinâmico. Certamente a tecnologia de agentes de software ajudará no desenvolvimento de Sistemas de Gerenciamento de Processo de Negócios mais inteligentes e flexíveis.

No capítulo seguinte, buscaremos relacionar o paradigma dos Agentes de Software e a tecnologia dos Workflows.

4 WORKFLOW E OS AGENTES DE SOFTWARES

A tecnologia de WfMS apareceu como o estágio tecnológico mais avançado da evolução de várias técnicas como a Reengenharia, e principalmente o *Groupware* e *Workgroup* (BUSSLER,1996)(KHOSHAFIAN,1995).

Os WfMS convencionais usam modelos e representações explícitos de processo, junto com ferramentas automatizadas que apóiam a ativação e a gestão contínua de uma instância de processo (GEORGAKOPOULOS,1995).

A tecnologia de *workflow* começou a ser aplicada em áreas caracterizadas pela simplicidade dos processos envolvidos como, por exemplo, os processos administrativos. O sucesso obtido encorajou as pesquisas para desenvolvimento de técnicas que permitem estender os benefícios dos sistemas de *workflow* para novas aplicações envolvendo processos complexos executados em ambientes dinâmicos e incertos. Isto inclui justamente as classes de tarefas e ambientes que a Inteligência Artificial (IA) está investigando no sentido de controlar entidades computacionais e/ou dispositivos físicos.

Enquanto o *workflow* emergiu durante a última década, a comunidade de inteligência artificial esteve envolvida com pesquisas relacionadas a modelagem e gerência de processos durante várias décadas. Ao contrário do enfoque de *workflow* sobre processos industriais e de negócio, a comunidade de IA se motivou inicialmente por domínios que envolvem controles reativos de entidades computacionais e dispositivos físicos (por exemplo, robôs, antenas, satélites, redes de computador, comunidades de agentes, etc.). Apesar destas preocupações e perspectivas diferentes, há muito em comum entre os objetivos, exigências, e abordagens destas duas comunidades. Workflow provê os ingredientes empresariais e a infraestrutura computacional que, respectivamente, motiva e permite uma aplicação industrial da tecnologia de IA.

O Grupo de Gerenciamento de Objetos⁴ (*Object Management Group* -OMG) estabeleceu recentemente um padrão de indústria para interoperabilidade de sistemas de *workflow*. Tais padrões estão promovendo o desenvolvimento de sistemas de *workflow* especializados incorporando funcionalidades de Inteligência Artificial.

Os *workflows* tradicionais são geralmente concebidos e armazenados para uso ulterior. Isto lhes dá um caráter pré-definido e estático, e conseqüentemente uma desvantagem potencial é a pouca adaptabilidade desses *workflows*. Além disso, com o recente progresso dos *middlewares* de comunicação e das aplicações distribuídas, os workflows precisam ser mais flexíveis e com alto grau de escalabilidade.

Para atingir esses requisitos, uma tendência bem promissora consiste em ver um *workflow* como uma coleção dinâmica de processos colaborativos que precisam ser agrupados (*cluster*). O agrupamento de processos e a gerencia de seus desempenhos pode ser realizado por componentes inteligentes reconhecidos como *Agentes de Software*.

Os agentes de software (AS) possuem características bem interessantes como autonomia, comportamento cooperativo, e eventualmente mobilidade (LABIDI; HAMMOUDI,1999). Além disso, eles levam em consideração as características do ambiente no qual eles evoluem. Isto traz muita flexibilidade desde que a combinação de processo possa estar diretamente baseada nos estados e decisões dos agentes, em lugar de serem predefinidos com antecedência.

4.1 Limitações dos Workflows atuais

A atual geração de WfMS tem seu foco central na coordenação de tarefas, enfatizando basicamente a interdependência. Estes sistemas tendem a operar com um controle

⁴ Organização internacional sem fins lucrativos, que visa desenvolver padrões para a tecnologia de objetos (www.omg.org).

centralizado que monitora e gerencia os eventos ocorridos durante o fluxo original do processo, porém nos casos onde o processo é dinâmico e um determinado evento pode adotar um fluxo não planejado na sua definição, algumas adaptações necessitam ser incorporadas para a completa adequação a esta situação.

A evolução das tecnologias e as transformações ocorridas no mundo dos negócios, tais como: as mudanças na estrutura tradicional de gerenciamento nas empresas, o surgimento das redes de computadores e a popularidade da Internet, vêm influenciando ativamente no ambiente das organizações, modelando um ambiente mais dinâmico e volátil, que exige processos mais complexos. Desta forma, as desvantagens e limitações dos Sistemas de Gerenciamento de Workflow existentes, tornam-se mais evidentes. Diante das desvantagens e limitações dos WfMS atuais, podemos citar:

- ? A pouca flexibilidade durante a execução do processo;
- ? A falta de habilidade para lidar com modificações dinâmicas dos recursos e das tarefas;
- ? Falta de facilidade para redistribuir os itens de trabalhos automaticamente sempre que necessário;
- ? A falta de um tratamento próprio às exceções, especialmente durante a execução dos itens de trabalho;
- ? Habilidades limitadas para prever modificações, devido a eventos externos;
- ? Incompatibilidade entre eles: A maioria dos WfMS apresentam-se de forma isolada, o que dificilmente possibilita uma interligação entre eles.

Uma das grandes vantagens de implementar sistemas de gerenciamento de workflow baseado na tecnologia de agentes inteligentes, é a possibilidade de poder facilmente gerenciar processos não estruturados, uma vez que suas técnicas de inteligência permitem tornar o gerenciamento dinâmico.

Nos WfMS atuais, o processo tem seu fluxo de trabalho pré-estabelecido pelo modelo de origem. Ocorrendo uma exceção neste fluxo, quase sempre leva o sistema a parar suas atividades, necessitando, porém, da intervenção externa para solução do fato não previsto na modelagem interna. A seguinte, resumiremos algumas das características dos sistemas de gerenciamento de workflow baseado em agentes de software.

4.2 Como os agentes de softwares podem ajudar?

Conforme verificamos nas seções anteriores, características como dinamismo e tomada de decisões, na maioria dos casos, são carências dos sistemas de workflows atuais. Os agentes de softwares buscam exatamente a execução de software com dinamismo e introduzem através de técnicas de IA, suporte à tomada de decisões. Assim tudo leva a crer que utilizando-se das características inerentes aos agentes de softwares, os sistemas de gerenciamento de workflow podem superar algumas das suas limitações.

Muitos pesquisadores na área de workflow já vem trabalhando no sentido de agregar aos WfMS o paradigma de agentes de softwares. Eles acreditam que as características dos agentes de softwares podem melhorar substancialmente o desempenho dos sistemas de gerenciamentos de workflow, principalmente quando tratamos com sistemas distribuídos, onde os agentes móveis (AM) demonstram características que os tornam atraentes. Assim, a noção de sistemas de workflow baseados em agentes é um campo de pesquisa em evidência.

Alguns grupos de pesquisadores vem desenvolvendo sistemas de workflows baseados em agentes de softwares, e cada um desses grupos introduzem nas suas soluções características e melhorias próprias. Ainda neste capítulo, apresentaremos resumidamente algumas das soluções mais freqüentemente encontradas nas literaturas do gênero.

4.3 Workflow baseado em agentes de software

Segundo Odgers (ODGERS,1999a), um sistema de gerenciamento de workflow baseado em agente é um conjunto de componentes de software que reúne critérios a serem considerados como agentes e estão envolvidos no gerenciamento do fluxo de tarefas em um processos de negócios. E um sistema de gerenciamento no qual os agentes executam, coordenam e apóiam um processo de negócio como um todo ou em parte dele (LEI,1999). A composição da arquitetura de um sistema de gerenciamento baseado a agente, depende muito do modelo de sistema adotado para definir o processo. Assim, na maioria das soluções estudadas é possível encontrarmos diferentes tipos de agentes que gerenciam o workflow. Como por exemplo, podemos citar: O *Agente Workflow*, (ou às vezes chamados de *Agente Processo*), que é responsável por controlar uma instância do processo, O *Agente Atividade* que normalmente é móvel e incorpora o papel de um ator que executa alguma tarefa no processo e em muitos casos encontramos a presença de um agente estacionário que atua como uma interface entre os recurso locais e o agente tarefa.

4.4 Duas abordagens

Shepherdson (SHEPHERDSON,1995) enfatiza que o uso de agentes de software proativo nos sistemas de gerenciamento de workflow enriquece o trabalho distribuído, possibilitando maior controle no tratamento de eventuais falhas, bem como maior flexibilidade na interoperabilidade dos sistemas de workflows. Embora este seja um campo da pesquisa que vive em constante evolução, hoje observamos nas soluções apresentadas, duas abordagens de desenvolvimento (SHEPHERDSON,1995). Uma conhecida como *Sistemas de Workflow Baseados em Agentes*, que visa o desenvolvimento total do sistema baseado em

agentes e outra conhecida como *Sistemas de Workflow Aprimorados por Agentes*, que visa o aproveitamento da base instalada dos sistemas existentes e acrescentando apenas uma camada de agentes para “Agênciar” os serviços prestados.

Nos tópicos seguintes, fazemos uma breve descrição de cada uma das abordagens citadas.

4.4.1 Workflow baseados em agentes

Segundo Shepherdson (SHEPHERDSON,1995), os sistemas desenvolvidos que tem como filosofia esta abordagem, buscam distribuir entre os agentes o controle das três fases do gerenciamento do processo, (do inglês : provision, enactment e compensation). Desta forma, em um sistema deste tipo, os agentes de softwares têm total responsabilidade com a provisão de recursos (provisioning), a execução da instância do processo e os ajustes na solução (compensation), com cada agente gerenciando e controlando uma determinada atividade ou conjunto de atividades. Porém, o desenvolvimento de workflow baseado em agentes ainda incorre em alto custo de engenharia e o resultado de uma modificação na tecnologia instalada levará a um grande esforço de treinamento.



Figura 10 - Workflow baseado em agentes

4.4.2 Workflow Aprimorado por agentes

Esta abordagem é chamada de *Sistemas de Workflow Aprimorados por agente*, e

ao contrário dos sistemas totalmente baseados em agentes, ela busca combinar uma camada de agentes com os sistemas de gerenciamento de workflow existentes. A camada de agente é responsável tanto pela fase de provisão dos recursos como pela fase de compensação no gerenciamento de processo de negócio, enquanto que a base do sistema de gerenciamento de workflow cuida da execução do processo.

Na camada de agente, cada agente é responsável por um ou mais serviços oferecidos, onde um serviço ofertado é uma combinação de atividades do workflow e os recursos que são dependentes dele. Assim esta camada de agentes é tratada nesta abordagem como uma “agência”.

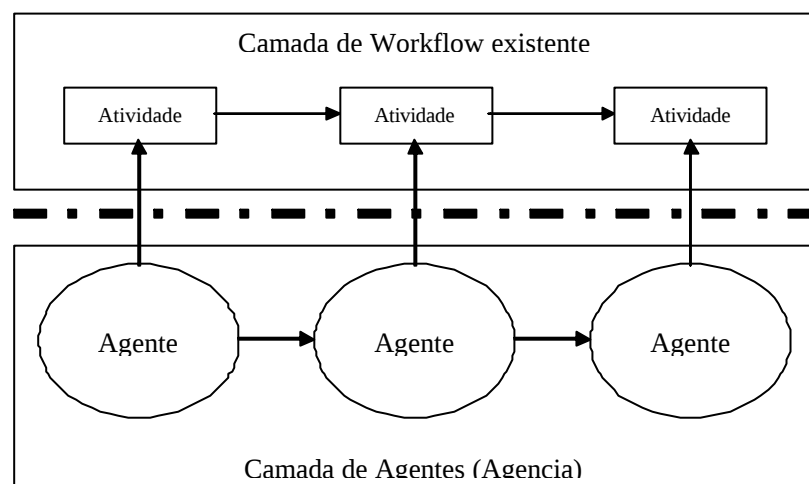


Figura 11 - Workflow executado por agentes

Os workflow aprimorados por agentes trabalham para satisfazer solicitações externas que são traduzidas como um objetivo para a sociedade de agentes (uma agência) alcançar. Os agentes envolvidos devem se relacionar, colaborando e compartilhando os recursos, podendo até dividir o objetivo principal em pequenos objetivos (sub-objetivos) para que seja negociada sua execução em outra agência conforme a necessidade.

4.5 Trabalhos Relacionados

Na tentativa de unir o paradigma de agentes de software ao gerenciamento de processos de negócios, muitos pesquisadores apresentaram soluções que merecem destaque na literatura do gênero.

Em TRP Support Enviroment (TSE), Chang (CHANG,1996) introduz diferentes tipos de agentes que gerenciam um processo de negócio. Hawryszkiewicz e Debenham (HAWRYSZKIEWYCZ,1998) descrevem um sistema de workflow baseado em agentes focalizando a inteligência e a cooperação entre agentes. Merz, Liberman e Lamersdorf (MERZ,1997) introduzem os agentes móveis para apoiar a interoperabilidade entre sistemas de gerenciamento de workflows. Eles realizaram uma implementação baseada em CORBA de um sistema de agentes distribuídos. O RSA (do Inglês : Research Student Application) (HAWRYSZKIEWYCZ,1998), é um sistema de workflow baseado em agentes experimentais desenvolvido na Universidade de Tecnologia de Sydney.

Outras soluções podem ser encontradas na literatura do gênero. Esta seção objetiva descrever alguns trabalhos relacionados ao controle do fluxo de processo envolvendo agentes de software. A seguir serão resumidos dois sistemas. Um é o Agent-based Process Management System – APMS (JENNINGS,1996) e o outro é o DartFlow (CAI,1996) (ODGERS,1999b).

4.5.1 APMS

O Agent-based Process Management System – APMS é uma abordagem de sistema de gerenciamento de processo baseado em agente de software desenvolvida pela British Telecom (BT). O APMS é apresentado como uma solução orientada a serviços, pois

como um processo, consiste em um número de atividades relacionadas que muitas vezes retrata as unidades de negócios de uma organização (Ex: do processo de Vendas). Os agentes de software representam os interesses destas unidades de negócios e comunicam-se através da rede de comunicação negociando a troca de serviços, visando alcançar o objetivo comum do processo. Desta forma, na visão orientada a serviço do APMS, os agentes ora fornecem serviços para os outros, (são provedores) ora recebem serviços dos outros agentes (são clientes). Um serviço é então um conjunto de tarefas que permite a um agente oferecer ou receber de um outro agente, algumas funcionalidades operacionais.

4.5.1.1 Modelo de Referência APMS

O modelo de referência do APMS é constituído basicamente de três componentes principais e as interfaces entre eles, os componentes, são:

- ? O Agente;
- ? A Tarefa;
- ? O Sistema Monitor APMS

A Figura 12 mostra o modelo de referência do APMS.

Os *Agentes* trocam informações entre eles através da interface *A* e gerencia as tarefas do processo através da interface *B*. O desenvolvimento, a manutenção e a administração dos serviços são executados pelo componente Monitor utilizando a interface *C*. Durante a execução do processo, algumas informações podem ser trocadas entre as tarefas pela interface *D*.

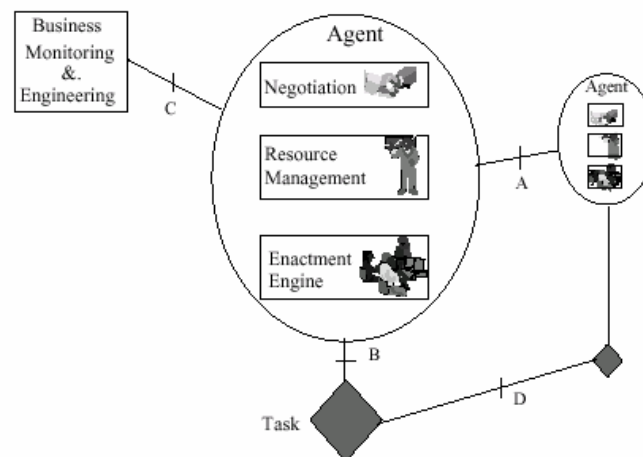


Figura 12 - Modelo de referência do APMS

Um agente APMS típicos se divide em três módulos: o módulo da negociação, o módulo de gerenciamento e o módulo de execução.

4.5.1.2 Módulo de negociação

O Módulo de Negociação é o processo através do qual dois agentes procuram uma negociação mútua e o comprometimento na entrega de um serviço. Parâmetros como a qualidade do serviço, tempo de entrega e custo do serviço, são levados em conta na fase da elaboração da negociação. O resultado de uma negociação entre cliente e provedor é um contrato de prestação de serviço contendo a lista dos valores acordados.

No APMS para a comunicação dos parâmetros, utiliza-se FIPA (FIPA,1997), uma linguagem com primitivas de negociação que especifica um padrão de comunicação para agentes e um conjunto de protocolo para apoiar a interoperabilidade.

4.5.1.3 Módulo gerenciador de recursos

O Gerenciamento de recursos é uma das vantagens do APMS. O módulo de gerenciamento de recursos de um agente garante que o agente forneça o serviço. Um recurso pode ser visto como um sistema, uma base de dados ou até uma pessoa que participa do processo de negócio. No APMS, um conjunto de recursos é atribuído ao agente para possibilitar o cumprimento do serviço contratado, assim, há a necessidade de algum controle sobre o comprometimento desses recursos, visando otimizar rapidez, eficiência e minimizar os custos e o desperdício.

4.5.1.4 Módulo de Execução

Este módulo envolve a ativação das tarefas e serviços negociados de forma que execute o combinado em contrato. Um agente servidor inicia suas tarefas e serviços quando este é solicitado por um cliente agente. Esta ativação pode envolver a execução de software ou o envio de uma mensagem via e-mail, por exemplo.

Ele pode executar vários serviços simultaneamente, portanto, deve ser *multithreaded* para permitir concorrência. Na ocorrência de uma falha durante a execução da tarefa, o agente pode executar uma ação corretiva e tentar resolver a falha.

4.5.1.5 Aplicação

Para exemplificar a utilização do APMS, podemos citar duas aplicações desenvolvidas com base nesta abordagem: o ADEPT (ALTY,1994) e o Beat (O'BRIEN,1996), ambas desenvolvidas pela BT. O ADEPT desenvolve um protótipo APMS

demonstrando a autonomia dos agentes gerenciando quase uma centena de tarefa de negócio e o BeaT estuda a utilização do APMS em tarefas de negócios em grandes organizações como a BT e considera o aspecto de engenharia do APMS.

4.5.2 DartFlow

O DartFlow (ODGERS,1999b) é um sistema de gerenciamento de workflow baseado em agentes desenvolvido pelo Dartmouth College. Ele objetiva aprimorar os WfMS comerciais de forma a resolver alguns problemas comuns aos workflows, tais como a falta de flexibilidade e o tratamento inteligente de erros. O DartFlow usa como sua IHM applets JAVA integrados ao browser da WEB e como backbone os agentes móveis. O desenvolvimento do sistema está baseado na linguagem JAVA, o que possibilita uma certa independência de plataforma, e os agentes são implementados em AgentTcl (GRAY,1995) o que torna o DartFlow flexível.

O Funcionamento do sistema DartFlow segue a seguinte sequência: Após a autenticação do usuário através de um login, uma lista de tarefas a ser executada por este usuário (worklist) é apresentada possibilitando o usuário selecionar um item e receber no browser um formulário HTML. Quando o formulário é submetido, um programa CGI é invocado, criando um novo agente que processará o formulário. Como cada agente móvel contém a sua própria descrição de processo, o DartFlow tem a capacidade de adaptar-se a cada instância de processo conforme suas necessidades específicas. O agente permanece ativo até terminar a tarefa para a qual ele foi criado. O DartFlow é composto de quatro componentes principais:

- ? *A Interface de Usuário (IU);*
- ? *O Agente Processo, que carrega os dados e a informação de controle do fluxo.*
- ? *O Agente Servidor, que implementa a maior funcionalidade como a informação de*

roteamento, o tratamento de erro, etc.

- ? O *Servidor de Lista de Trabalho* (WorkList) para comunicar o resultado do agente processo para a interface do usuário.

4.5.2.1 A interface do usuário

O DartFlow utiliza-se dos recursos disponíveis na plataforma WEB para interagir com o usuário. Applet Java, HTML, CGI, etc. formam o conjunto de recursos necessários para visualizar e submeter os formulários específicos dos itens de trabalho de uma determinada tarefa. Através desta interface, o usuário pode, dentre outras facilidades: conectar-se ao sistema submetendo um formulário de login, solicitar sua lista de tarefa, submeter alterações ao processo, etc. Cada novo formulário recebe uma identificação única através de um número ID o qual identifica o agente processo, uma vez que para cada formulário submetido, um novo agente será criado.

4.5.2.2 Agente processo

O Agente Processo no DartFlow é um agente móvel que carrega os dados e as informações do controle de fluxo no processo. Ao submeter um formulário na IU um script CGI é invocado, o qual pode criar um novo agente processo ou acordar um agente adormecido no processo. Um único agente é responsável por cada instância do processo até a sua finalização. O agente processo solicita um determinado serviço ao agente servidor, passando como argumento o conteúdo dos campos do seu formulário. O Agente Servidor após processar a solicitação do agente, retorna o resultado em um script Tcl. Baseado na resposta da sua solicitação, o agente processo pode dinamicamente modificar seu comportamento ou alterar seu conteúdo.

4.5.2.3 Agente servidor

O Agente Servidor, por sua vez, implementa a maior parte das funcionalidades do DartFlow, tais como o roteamento de informações e tratamento de erros. Dois Agentes Servidores são implementados no DartFlow, um é o *Servidor Organizacional (Organization Server)* que contém informação genérica do fluxo do processo e o outro é o *Servidor Rastreador (Tracking Server)*, que guarda a trilha de todos os Agentes Processo e a sua seqüência de atividades, assim como as modificações.

4.5.2.4 Servidor de lista de trabalho (Worklist Server)

O Servidor de lista de trabalho mantém a lista de tarefas, atualiza as informações para os clientes ativos e informa aos Agentes processo que alguma alteração ocorreu e ele está sendo solicitado a consultar novamente o Agente Server e recarregar a nova lista de trabalho. Uma conexão através de socket entre a IU e o Servidor de Lista de Trabalho, possibilita a atualização imediata da IU caso ocorra alguma modificação na lista de tarefa deste usuário.

4.5.2.5 O Fluxo do processo DartFlow

O Fluxo do Processo DartFlow envolvendo seus principais componentes, pode ser observado na figura 13 , e descrito resumidamente da seguinte forma:

Fluxo Normal :

- 1) O Agente Processo envia o tipo do processo e o ID;
- 2) O Servidor Organizacional retorna a lista de tarefa;

- 3) O Agente Processo registra-se como o Servidor Rastreador;
- 4) No fim de cada tarefa, o Agente Processo informa ao Servidor de Lista de Trabalho;
- 5) O Servidor de Lista de Trabalho atualiza a IU;
- 6) O Usuário submete novos dados, retornando para o passo 4.

Fluxo com modificações dinâmicas:

- a) No caso de uma modificação no processo, o Servidor Organizacional informa ao Servidor Rastreador sobre os usuários e processos afetados;
- b) O Servidor Rastreador informa ao Servidor de Lista de Trabalho os agentes afetados;
- c) Da próxima vez que um Agente Processo contatar o Servidor de Lista de Trabalho, este verifica se é um agente afetado pela modificação e caso seja, solicita sua visita ao Servidor Organizacional para obter nova lista.
- d) O Agente Processo contata o Servidor Organizacional;
- e) O Servidor Organizacional retorna a nova lista de trabalho.

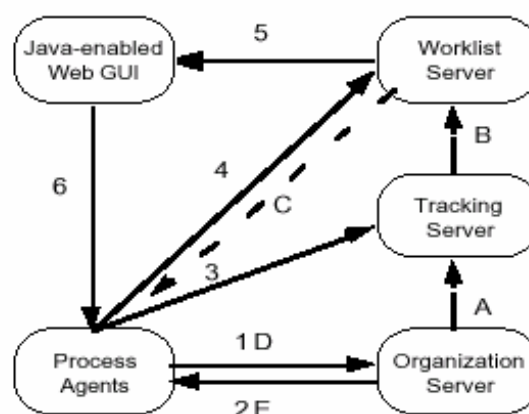


Figura 13 - Interações entre os componentes do DartFlow

4.6 Conclusão

Algumas limitações encontradas nos sistemas de gerenciamento de workflow, como: a falta de flexibilidade, necessidade de modificações dinâmicas, redirecionamento do fluxo de trabalho e outras, nos propõe a buscar novas técnicas de desenvolvimento para agregar melhorias no controle de processo. Este capítulo, após citar algumas limitações, faz uma reflexão de como os agentes de softwares podem ajudar. As diversas características dos Agentes de Softwares discutidas nos capítulos anteriores, parecem ser a solução ideal para os sistemas de controle de processo atuais. Um grande número de pesquisadores tem se dedicado ao estudo dos WfMS baseado em agentes de software. Muitas são as idéias que surgem a cada dia, e diante do crescente volume de pesquisa nesta área, acreditamos que ainda muitas idéias ainda surgirão e que estamos apenas no começo.

Neste capítulo, citamos algumas abordagens de sistemas de gerenciamento de workflow baseados em agentes de software e escolhemos duas delas para descrevermos resumidamente. Foram elas: a abordagem da British Telecom (BT) um Agent-based Process Management System – APMS e o DartFlow da Dartmouth College.

No capítulo seguinte, apresentamos o projeto WEICOT, que incorpora os aspectos temporais e cooperativos aos sistemas de gerenciamento de Workflow, possibilitando controlar processos poucos estruturados. O projeto WEICOT será a base conceitual para o desenvolvimento da nossa contribuição.

5 O PROJETO WEICOT

Após apresentarmos as tecnologias de Workflow e Agentes softwares e o estado da arte em desenvolvimento de WfMS baseado em agentes, apresentamos, neste capítulo, uma visão geral do projeto WEICOT, cuja arquitetura e definição dos aspectos temporais e de cooperação, serão a base para o desenvolvimento da ferramenta de definição de processo que nos propomos a apresentar neste trabalho.

O projeto WEICOT⁵ (LABIDI; HAMMOUDI,1998) tem como temática principal a modelagem, concepção, e realização de um sistema de gerenciamento de processos (*Workflow Management System -WfMS*) permitindo a organização e execução de atividades pouco estruturadas e cooperativas. Para prover as características necessárias deste tipo de tarefas, o WEICOT estende os WfMS atuais integrando o conceito de cooperação e organização temporal entre as atividades de um processo.

O projeto situa-se no cruzamento das áreas de Informática (Inteligência Artificial Distribuída, Bancos de Dados e Engenharia de Software) e das ciências das organizações (Sistemas de Informação e *Office Information Systems*).

5.1 Relevância do Projeto

O suporte de trabalho de grupo e especificamente das organizações em suas atividades rotineiras foi, durante muito tempo, um grande entrave para os sistemas de informação. As pesquisas nessa área conduziram, na última década, à definição dos *workflows* como um novo paradigma ajudando à automação de processos. O interesse pelos sistemas de *workflows* é crescente e ficou claro agora que esse fenômeno vai ter um grande impacto sobre os sistemas de informação futuros. Uma visão desta tendência é a aplicação da

⁵ Workflow Environment Integrating Cooperating and Temporal Organization

pesquisa em workflow ao comércio eletrônico (ANDERSON,1999)(LABIDI,1996).

Partindo de uma perspectiva de pesquisa, a DARPA⁶ identificou o workflow como um tema de suas tecnologias básicas (*key “must have” technologies*) e está investindo pesadamente no desenvolvimento de sistemas de gerenciamento de *workflow* de segunda geração. Porém, o impacto de tal tecnologia está bem além de uma aplicação específica e é de grande interesse para comunidade de negócio em geral.

Apesar de ser relativamente recente, o domínio de workflow tornou-se rapidamente de grande importância. Enquanto há alguns anos qualquer organização restringia as suas exigências de sistemas de informação à solução cliente-servidor da sua base de dados, hoje em dia nenhuma dispensa uma solução de *workflow* otimizada, que em muitos casos condiciona a própria arquitetura da base de dados.

Prevê-se que o domínio do Workflow se torne, ao longo dos próximos tempos, um domínio aglutinador de várias tecnologias, dentre elas os Agentes de Software, havendo quem acredite que o próprio conceito de trabalho colaborativo seja diluído nas soluções de *workflow*.

Centenas de WfMSs, comerciais e outros, estão sendo desenvolvidos. Porém, poucos oferecem as funcionalidades de um verdadeiro motor de workflows (*workflow engine*). Notamos que os WfMS mais conhecidos são: ActionWorkflow (da *Action Technologies*), FlowMark (da IBM) e Visual WorkFlow (da FileNet) (GEORGAKOPOULOS,1995).

⁶ Departamento de Defesa Americano.

5.2 Arquitetura do WEICOT

A arquitetura conceitual do WEICOT esta organizada em três níveis: a) nível de interface, b) nível de ferramenta e c) nível de armazenamento.

- a) O nível de interface permite a comunicação entre os usuários e o sistema. Os usuários podem ser gerentes ou trabalhadores usuais. Este nível se divide basicamente em quatro serviços: Especificação da tarefa, Monitoramento e execução da tarefa, Organização temporal da tarefa e Visualização e browsing.
- b) O nível de ferramenta, por sua vez, implementa os quatro módulos do nível anterior e constitui o componente principal do WEICOT. Este nível consiste em duas ferramentas: a Ferramenta WfMS, que implementa as funcionalidades de gerenciamento do Workflow e a Ferramenta Agenda, que implementa as funcionalidades de organização temporal.
- c) O nível de armazenamento utiliza um sistema gerenciador de base de dados para armazenar toda a parametrização do processo de negócio modelado.

5.3 Modelo conceitual do WEICOT

O Sistema WEICOT é um modelo baseado em tarefas, (que também pode ser entendido como Atividades). Desta forma, destacamos dois aspectos importantes que o WEICOT se propõe a apresentar: o comportamento e a noção de tempo. O comportamento está associado ao modo como os agentes executam as tarefas, e a noção de tempo possibilita modelar a evolução temporal do processo.

No WEICOT, os componentes têm uma representação gráfica, o que permite uma visualização do fluxo de trabalho de maneira fácil e clara. Além da representação gráfica, cada componente possui uma descrição textual, em que são definidas as suas informações.

O Modelo proposto está definido de acordo com algumas perspectivas:

- ? *Funcional* – O modelo apresenta a unidade funcional do workflow que é o processo de negócio.
- ? *Organizacional* - O modelo apresenta os agentes e os recursos como elementos de organização.
- ? *Perspectiva de estados* – O modelo apresenta os diversos estados que as tarefas, os agentes e os recursos alcançam no decorrer do processo.
- ? *Perspectiva de relacionamento* – O modelo apresenta os tipos de relacionamento existentes entre tarefas.
- ? *Comportamental* – O modelo apresenta a definição de comportamentos que os agentes utilizam na execução das suas tarefas.
- ? *Temporal* – O modelo apresenta noções de tempo, possibilitando modelar a evolução temporal do processo.

5.3.1 Processo

O processo no modelo do WEICOT é definido como um conjunto de tarefas relacionadas entre si através de regras, a fim de atingir um determinado objetivo da organização. No WEICOT um processo é representado graficamente por um triângulo. No interior do triângulo tem-se o nome do processo.

As seguintes informações fazem parte da descrição textual de um processo. São elas:

- ? Nome do processo;
- ? Objetivo do processo;
- ? Data inicial;

- ? Data final;
- ? Responsável.

5.3.2 Tarefas

Sendo o processo um conjunto de tarefas, cabe aqui um detalhamento mais apurado do conceito de tarefas no WEICOT. A tarefa é o elemento principal do modelo WEICOT (LABIDI; HAMMOUDI,1998). A tarefa corresponde a um passo lógico dentro do processo e pode ser vista como um trabalho associado a um agente para sua execução.

A representação gráfica de uma tarefa no modelo do WEICOT é feita através de um retângulo contendo o nome da tarefa no seu interior. Para a representação de sub-tarefas, o modelo propõe um retângulo duplo.

As tarefas apresentam alguns atributos, que o modelo chama de dados cadastrais.

São eles:

- ? *Nome da tarefa;*
- ? *Objetivo da tarefa;*
- ? *Recursos;*
- ? *Entrada;*
- ? *Saída;*
- ? *Tipo da tarefa.*

O nome da tarefa indica o nome associado àquela tarefa; o objetivo descreve o que a tarefa pretende realizar; os recursos, indicam o que será necessário para executar a tarefa; a entrada descreve os dados que a tarefa utiliza na sua execução, podendo ser até a saída da tarefa anterior; a saída descreve os dados produzidos pela execução da tarefa, o que pode servir de entrada para uma outra tarefa e o tipo indica se a tarefa é simples ou composta (sub-tarefa).

5.3.3 Restrições

As restrições são expressões lógicas que serão avaliadas para verificar se a tarefa deve ser executada ou concluída. Há, no WEICOT, dois tipos de restrições: as pré-condições que avaliam a execução da tarefa e as pós-condições, que consistem em uma expressão lógica que avalia um dado como critério, para conclusão da tarefa.

5.3.4 Estados da Tarefa

Um estado da tarefa mostra como se encontram suas atividades em um determinado instante no processo. Para uma tarefa simples, têm-se os seguintes estados: *criada, alocada, executando, concluída, abortada e suspensa*. Já para as tarefas composta (sub-tarefas), acrescenta-se um novo estado que é: *Executando Sub-Tarefa*.

5.3.5 Gatilho

O modelo do WEICOT utiliza-se da técnica de disparo de gatilho para o acionamento das atividades. Um gatilho é disparado ao final de algumas tarefas fazendo com que uma outra tarefa seja iniciada obedecendo ao fluxo do processo. Um gatilho pode ser disparado automaticamente através de um agente, de uma mensagem ou de informações de limitações de tempo.

5.3.6 Relacionamento entre tarefas

O fluxo traçado pela definição de um processo exige que relacionamentos entre as

tarefas sejam cumpridos, assim a noção de relacionamento entre as tarefas é inspirada do modelo de redes de Petri (JENSEN,1992) no modelo do WEICOT. Os seguintes relacionamentos são evidentes neste modelo e merecem sua conceitualização:

- ? *Relacionamento Seqüencial* - Neste tipo de relacionamento, as tarefas são executadas na ordem seqüencial, assim, a conclusão da tarefa anterior habilita o início da tarefa posterior.
- ? *Relacionamento e-junção* – Este relacionamento implica em convergir a indicação de término de duas ou mais tarefas executadas em paralelo anteriormente, para disparar a próxima tarefa.
- ? *Relacionamento e-split* - Este relacionamento sincroniza o término de uma tarefa com a ativação de duas ou mais tarefas posteriores em paralelo.
- ? *Relacionamento ou-junção* - Este relacionamento implica em convergir a indicação de término de pelo menos uma de duas ou mais tarefas executadas anteriormente, para disparar a próxima tarefa.
- ? *Relacionamento ou-split* - Este relacionamento sincroniza o término de uma tarefa com a ativação de pelo menos uma de duas ou mais tarefas posteriores.
- ? *Relacionamento condicional* – Este relacionamento ocorre quando uma condição após o término de uma tarefa deve definir qual a próxima tarefa a ser executada.
- ? *Relacionamento interação condicional* – Este relacionamento corresponde a um ciclo que envolve a execução repetitiva de uma tarefa enquanto a condição for verdadeira.

5.3.7 Agente

No modelo, um agente é o elemento responsável pela execução das tarefas. Para cada agente será definida uma lista de tarefas.

Um determinado agente pode se encontrar em um dos quatros estados: disponível imediato, disponível, ocupado e indisponível. O estado disponível imediato indica que o agente está criado, porém ainda não tem suas tarefas disponíveis. Este é o estado inicial do agente. No estado disponível, o agente já possui suas tarefas, mas ainda não completou sua capacidade de número de tarefas. Quando um agente preenche sua capacidade de execução de tarefas, ele passa para o estado ocupado. O Estado indisponível ocorre quando o agente não pode receber uma tarefa.

Os agentes podem ser unidades computacionais e/ou seres humanas. São definidos alguns atributos para o agente. Sendo:

- ? **Nome:** denominação do agente;
- ? **Tipo:** pode ser humano ou um software;
- ? **Papel:** a cada agente é associada uma função (papel) dentro da organização;
- ? **Estado:** indica o estado de disponível imediato, disponível, ocupado, indisponível;
- ? **Composição:** indica se o agente é simples ou composto;
- ? **Limite de tarefas:** quantidade de tarefas que o agente pode ter simultaneamente alocadas para ele.

5.3.8 Recurso

Durante a execução das tarefas pelos agentes, há necessidade de utilizar algum recurso, como equipamentos, aplicativos, entre outros.

Os recursos podem ter dois estados: disponível e indisponível. O recurso estará disponível quando ele puder ser utilizado pelo agente para executar uma determinada tarefa, e indisponível quando não. O elemento recurso não é representado graficamente na descrição do processo no WEICOT. Os atributos de um recurso são:

- ? Nome do recurso;
- ? Tipo do recurso;
- ? Estado do recurso.

5.4 Comportamento cooperativo

Nos sistemas de gerenciamentos atuais, encontramos evidenciada a missão de coordenar trabalhos, porém o modelo do WEICOT estende os seus conceitos a fim de permitir, além da coordenação de tarefas, observar o comportamento cooperativo entre seus agentes.

O trabalho em grupo pode ser definido como um conjunto de pessoas que compartilham seus recursos para executar uma tarefa e obter dela o melhor resultado possível. Hoje quase que todos os processos de negócios em uma organização necessitam da cooperação para alcançar seus objetivos. O modelo do WEICOT acrescenta alguns atributos ao modelo de tarefa de forma a representar o comportamento cooperativo. Estes atributos são definidos como: *escopo, alocação, comportamento e atitude de delegação*.

O atributo de ***alocação*** determina qual agente, seja ele um agente individual (simples) ou um grupo de agentes (agente composto), é o responsável pela execução da tarefa.

O atributo ***escopo*** está relacionado com a abrangência das tarefas, ou seja, uma tarefa pode ser “*local*” (se executada por apenas um agente individual) e/ou pode ser “*global*” se executada por mais de um agente. Quando o escopo é global, é necessário um esforço coletivo na sua execução e, portanto, um comportamento cooperativo.

O atributo ***comportamento*** pode assumir vários valores, como, por exemplo: *competição, co-ação, negociação, etc.*

? *Competição*

O Comportamento de *competição* ocorre no WEICOT quando uma tarefa é alocada a um grupo de agentes onde cada um deve executar esta tarefa separadamente e será escolhido o melhor resultado entre eles tendo em vista alguns critérios de avaliação. Para exemplificar este comportamento, citamos um processo de compra de material onde a tarefa de fazer coleta de preço pode ser executada por vários compradores e aquele que obtiver o menor custo e maior rapidez na aquisição do material, é a proposta escolhida.

? *Co-ação*

O Comportamento de *co-ação*, ocorre de forma semelhante à competição, onde uma tarefa é alocada para um grupo de agentes, porém, não há uma competição entre eles e nem a escolha do melhor resultado, todo resultado é aceito e necessário para o objetivo. Para definirmos melhor este conceito, podemos citar o exemplo de um processo de avaliação de artigos, onde a tarefa de revisão é executada por alguns agentes e todos devem apresentar seu parecer quanto ao conteúdo do artigo avaliado.

? *Negociação*

O Comportamento de *negociação* ocorre quando uma tarefa é atribuída a um grupo de agentes para que seja resolvida em conjunto. Neste tipo de comportamento, não há a execução individualizada da tarefa por cada agente e sim a união de todos para executar aquela tarefa. Cada agente deve negociar uns com os outros para concluir a tarefa de modo consensual. Se não obtiverem o consenso, um agente superior deve intervir para a solução.

? *Complementaridade*

O Comportamento de *complementaridade* ocorre normalmente quando as tarefas podem ser subdivididas em pequenas sub-tarefas, podendo assim, cada sub-tarefa ser alocada a um agente simples de um grupo de agentes. Como exemplo prático, podemos citar um grupo de alunos responsáveis por apresentar uma pesquisa sobre um determinado tema. Cada um dos membros da equipe deve apresentar uma parte do conteúdo para que seja composto o trabalho final do grupo.

? *Assistência*

O Comportamento de assistência ocorre quando uma tarefa é atribuída a apenas um agente, porém, outros agentes o auxiliam na conclusão sempre que este agente precisa, ou seja, o agente que recebe a tarefa será assistido por outros que podem ajudar caso seja necessário.

O Último atributo, que representa a cooperação, é a ***delegação*** que determina se uma tarefa pode ou não ser alocada por seu agente inicial para outro agente caso haja necessidade. Durante a execução de uma tarefa, o agente pode ser solicitado a executar uma

outra mais prioritária e assim pode delegar à tarefa corrente para um outro agente. A delegação constitui também uma forma de cooperação, pois muitas vezes o esforço deve ser re-alocado para otimizar o processo.

5.5 Aspectos temporais

Além do comportamento cooperativo, uma grande contribuição incorporada pelo modelo WEICOT é uma abordagem sobre o tempo envolvido na execução de uma tarefa. Esses aspectos, aqui chamados de aspectos temporais, visam atribuir um controle mais real aos tempos gastos no ciclo de vida de uma tarefa e conseqüentemente de um processo.

O modelo do WEICOT visualiza na tarefa dois aspectos para a gestão temporal (LABIDI; HAMMOUDI,1998): o interno, onde se tem a noção de tempo associada aos diversos estados da tarefa e o externo onde se tem a noção de tempo associada à duração de cada tarefa.

A nível interno, a gestão temporal se dá pela associação de algumas datas ou horários aos estados, possibilitando retratar fielmente a evolução temporal da tarefa, mostrando o seu estado e o tempo de ocorrência.

A nível externo, a gestão temporal se dá indicando os tempos de duração da tarefa como um todo. Assim, é possível prever o tempo total do processo, bem como controlar mais ativamente as deficiências na execução do processo. Indica-se também um tempo associado a possíveis prorrogações devido à ocorrência de atraso no fluxo normal do processo. Para a gestão externa dos aspectos temporais, o modelo introduz o conceito de duração, prorrogação, tempo inicial, tempo final, tempo inicial prorrogado, tempo final prorrogado e o menor e maior tempo de duração do processo.

O tempo inicial e final representa o instante em que a tarefa começa e termina respectivamente sua execução. A duração corresponde ao intervalo de tempo entre o início e o fim da execução.

Uma prorrogação pode ser prevista no caso de atrasos. Este tempo conhecido como *tempo de prorrogação* são durações de dias ou horas pela qual será tomada como tempo inicial ou final, caso o tempo inicial não seja cumprido. O tempo de prorrogação assim como os tempos iniciais e finais, também estão associados ao início e ao fim da execução.

A representação gráfica no modelo do WEICOT é feita através da indicação do lado direito da representação da tarefa associada (para os tempos normais) e do lado esquerdo (para os tempos de prorrogação). A descrição dos tempos e duração obedecem à seguinte legenda:

- ? **D** : Duração da tarefa;
- ? **P** : Duração da prorrogação;
- ? **TI** : Tempo Inicial;
- ? **TF** : Tempo Final;
- ? **TIP** : Tempo Inicial Prorrogado;
- ? **TFP** : Tempo Final Prorrogado.

O tempo total do processo sem prorrogação é representado na notação do WEICOT pela indicação do menor tempo, e o tempo acrescido da prorrogação é representado pela indicação do maior tempo. Quando o início de uma tarefa depende do resultado de um operador e-junção, o seu tempo inicial é o maior dos tempos finais dentre as tarefas que fazem parte do operador e-junção. Quando a tarefa é a tarefa inicial do processo, o tempo inicial é zero. A Figura 14 mostra a representação gráfica de um processo utilizando a noção de tempo em nível externo das tarefas:

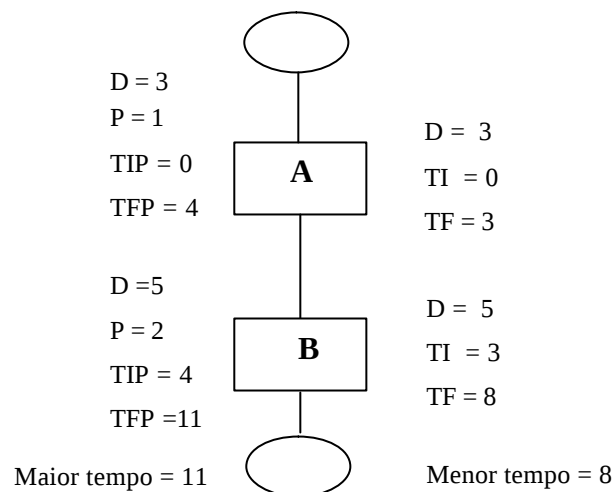


Figura 14 - Representação do tempo no WEICOT

5.6 Conclusão

Conforme verificamos nesta seção, o WEICOT é um modelo que procura incorporar conceitos temporais e de cooperação entre as tarefas nos sistemas de workflow já existentes, objetivando atingir a flexibilidade. O modelo conceitual do WEICOT, conforme definido anteriormente, é composto dos seguintes elementos: processo, tarefas, relacionamentos, agentes e recursos. Viu-se que o elemento principal do modelo é a tarefa.

Diante das definições apresentadas nesta seção, verificamos que através de atributos de tarefa como: *escopo da tarefa*, *alocação da tarefa*, *comportamento dos agentes* e *atitude de delegação*, o modelo do WEICOT incorpora os aspectos de cooperação entre os agentes na execução de suas tarefas.

Ainda nesta seção, verificamos como o modelo do WEICOT aborda os aspectos temporais no processo. A modelagem do tempo possibilita gerenciar o andamento das tarefas, monitorando a cada instante o seu estado interno e a sua duração no fluxo do processo.

O modelo do WEICOT apresenta também uma representação gráfica que denota na modelagem do processo os tempos de início e fim das tarefas, bem como possibilita

gerenciar os atrasos previstos na execução dos trabalhos.

A nossa contribuição está centrada na elaboração de uma ferramenta que, utilizando novas tecnologias e conceitos da ciência da computação e da área da Inteligência Artificial, possibilite implementar sistemas de gerenciamento de Workflow tomando como base o modelo do WEICOT, objetivando buscar maior flexibilidade na definição de um processo baseado em agentes de software.

No capítulo seguinte, apresentamos o **WorkFlage** como uma proposta para modelar e implementar um processo de negócio baseado na tecnologia de Agentes de Software para o modelo WEICOT.

6 WORKFLAGE

Apresentamos nesta seção uma arquitetura que estende o WEICOT para implementação de um sistema de gerenciamento de workflow baseado em agentes inteligentes, visando integrar as várias metodologias deste paradigma de desenvolvimento de software ao controle de processos de negócio. Estamos chamando de WorkFlage a arquitetura de um sistema de gerenciamento de workflow que incorpora uma ferramenta de forma a possibilitar a modelagem e a implementação de um processo de negócio com as características de um sistema multiagente, onde um grupo de agentes cooperam e colaboram entre si, reunindo todas as suas habilidades para controlar e gerenciar o fluxo de trabalho interno ao processo, sempre visando alcançar o objetivo proposto para cada instância gerenciada pelo WfMS.

A arquitetura posposta pela WorkFlage toma como base para a modelagem de um processo a conceitualização proposta pelo projeto WEICOT descrito anteriormente, adaptada de forma a incorporar as funcionalidades dos agentes de software.

Antes de apresentar as características da ferramenta de desenvolvimento WorkFlage propriamente dita, abordaremos nas seções seguintes a arquitetura proposta para adaptar o modelo WEICOT ao paradigma dos agentes, o modelo proposto para o agente, bem como suas habilidades, o modelo de perfil adotado pelo agente, o relacionamento entre os agentes, a base de conhecimento e o modelo de regras de produção para a aquisição do conhecimento.

6.1 Arquitetura proposta

De forma genérica, buscamos mostrar, com este modelo, uma estrutura coerente

para desenvolvimento de sistema de gerenciamento de workflow baseado em agentes sem que seja inicialmente, voltada a um domínio específico, entretanto, devido à abrangência e a flexibilidade disponibilizadas pelos agentes de software, o modelo proposto poderá também ser utilizado na modelagem de qualquer processo de negócio tais como : o comércio eletrônico, controle de tráfego, revisão de paper, etc.

Conforme comentamos anteriormente, ao incorporar os agentes de software ao controle de processo, visa-se dar uma maior flexibilidade aos sistemas que controlam processos não estruturados, onde o fluxo das atividades não é previsível, e exceções ocorrem a todo instante. Poucas são as ferramentas disponíveis no mercado para a definição de um processo de negócio baseado neste novo paradigma de programação, o que nos motivou neste trabalho a apresentar uma ferramenta que busca proporcionar a definição de um processo de negócio baseado em agentes de software de forma simples e acessível ao usuário final.

A **WorkFlage** pretende proporcionar a base para o desenvolvimento de workflow baseado em agentes, bem como despertar novas pesquisas e estimular a comunidade científica e a indústria de software a dar mais um passo na evolução tecnológica de novas ferramentas de software.

Acreditamos que uma flexibilidade bastante promissora e favorável na definição de processo seria o desenvolvimento de um sistema de workflow com os agentes voltados para “Gerenciamento” do fluxo de trabalho, ou seja, os agentes devem, além de executarem suas funções internas ao processo, gerenciar as exceções de forma a tomar as melhores decisões quanto ao novo rumo no fluxo das atividades.

Para as funcionalidades de gerenciamento de workflow, a WorkFlage incorpora um conjunto de biblioteca de classes que retrata as especificações do WEICOT (LABIDI; HAMMOUDI,1998) e para modelar a arquitetura do WEICOT com o paradigma dos agentes de software, verificamos a presença de três agentes de software atuando com funções bem

definidas no modelo do processo.

Os agentes **Processo**, **Atividade** e **Integrador**, interagem entre si compartilhando recursos e serviços rumo ao objetivo. Assim:

- ? No modelo do Workflage, o agente **Processo** atua sobre o interesse do processo como um todo, concentrando em si, todas as definições globais para o sistema de gerenciamento, tais como: Descrição do processo, Relação de Atividades, etc. Este agente recebe a configuração do objetivo geral a ser alcançado e que depois será subdividido entre os diversos agentes integrantes da sua instância. A cada instância do processo de negócio, apenas um agente processo será criado.
- ? O agente Atividade na WorkFlage acompanhará todos os passos de uma atividade não somente gerenciando os estados daquela atividade, mas também controlando o fluxo interno de tarefas para aquela atividade. Este agente recebe o objetivo geral do agente processo e busca cooperar e colaborar para uma melhor participação da sua função na obtenção de um resultado satisfatório. O agente Atividade reflete uma atribuição (função) desempenhada por um ator no processo, sendo para tanto configurado na definição do processo para exercer um determinado “papel”. Para cada instância do processo de negócio, vários agentes podem ser criados, dependendo da necessidade do processo.
- ? O agente Integrador tem a finalidade de promover o intercâmbio entre os agentes atividades e principalmente entre eles e o agente processo,

desvinculando algumas tarefas rotineiras dos agentes atividade e de processo, para que eles sejam liberados para o processamento das tarefas que envolvem maior “raciocínio”. A cada instância do processo de negócio, apenas um agente integrador será criado.

Toda atuação dos agentes se desenvolve em um ambiente com aspectos favoráveis à atuação desses agentes. Neste modelo do Workflage, o ambiente terá suas características adaptadas à instância do processo em execução, podendo ser puramente localizado ou distribuído. No contexto do comércio eletrônico a característica intrínseca ao meio ambiente se constitui distribuída. A Ferramenta gera para cada processo criado, um ambiente Servidor e um Ambiente Cliente que servirá de estrutura para execução do sistema de gerenciamento de workflow. Nesta estrutura um agente Integrador será introduzido e servirá de elo entre os ambientes, proporcionando uma espécie de gerenciamento de segurança para o tráfego distribuído dos agentes.

A Figura 15 ilustra a arquitetura do modelo proposto para o WorkFlage, com todos os seus componentes e relacionamentos entre eles.

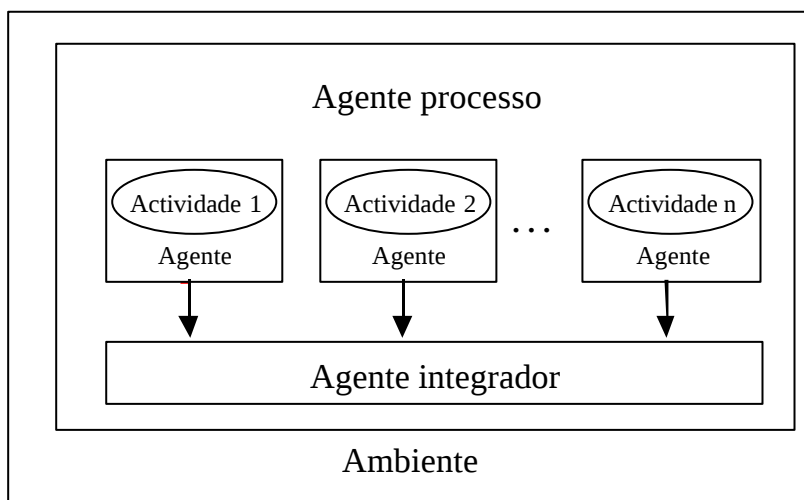


Figura 15 - Arquitetura do WorkFlage

O Desenvolvimento de um Sistema de Gerenciamento de Workflow deve partir da sua habilidade em definir um processo de negócio, ou seja, o sistema deve ter a capacidade de modelar e implementar um modelo de processo tornando-o disponível à criação de uma instância interna. Assim, antes de partirmos para a parte prática de implementação do sistema, abordamos a nossa modelagem de um processo baseado em agentes de software na WorkFlage.

Para a modelagem de processo, os WfMS atuais utilizam ferramentas de definição baseadas em técnicas de modelagem própria ao ambiente de desenvolvimento do sistema de gerenciamento. Assim, uma ferramenta, para modelar um processo de negócio cujo gerenciamento está baseado em agentes de software, não apenas deve ter a capacidade de gerar uma representação interna do processo, mas também deve agregar as novas características da tecnologia de agentes de software ao modelo. A ferramenta Workflage absorve toda a conceitualização de um processo e de forma rápida e transparente ao usuário, gera uma representação interna do processo, propiciando a criação de uma ou várias instâncias cujo controle do fluxo entre suas atividades e tarefas será gerenciado pelos agentes.

A Workflage dispõe de um conjunto de classes, pré-dispostas a instanciar os agentes com as definições mínimas das habilidades inerentes a um tipo de agente padrão. Esta biblioteca de classes constitui o “*Kernel*” desta ferramenta, ficando a cargo do usuário que vai modelar o processo na ferramenta, a configuração dos agentes conforme as necessidades das suas atribuições e funcionalidades desempenhadas no processo.

É importante ressaltar que um bom sistema de gerenciamento de workflow deve permitir uma flexibilidade na definição, manutenção e execução do processo, permitindo, assim, que a qualquer momento alterações como incluir novas atividades ou alterar atividades existentes sejam efetuadas sem perder, com isto, o controle dos processos já em andamento.

A idéia proposta para a ferramenta Workflage consiste em atribuir aos três agentes do modelo, algumas habilidades para que, somadas as propriedades inerente aos agentes de software, agreguem valores no sentido de otimizar os gerenciamento dos fluxo dos processos em uma organização.

Tendo em vista que grande parte dos processos envolve atividades distribuídas, uma das características mais exploradas dos agentes de software na modelagem do processo, será a mobilidade, o que permitirá a um processo ser executado em máquinas diferentes e em tempos diferentes sem que seja perdida a característica de gerenciamento global do processo.

No domínio do comércio eletrônico, os processos de negócios têm, quase que na totalidade, uma abrangência distribuída e sendo assim o agente necessita ter um comportamento também dinâmico. Qiming Chen (QIMING,1999) define uma arquitetura que se adapta a esta característica de dinamismo. São as chamadas “arquiteturas dinâmicas”.

6.2 Modelagem do Agente

Para compor todas as funcionalidades do modelo proposto, visualizamos uma arquitetura para os agentes, composta basicamente de três subdivisões. Tais subdivisões são tratadas aqui como: parte fixa (*kernel*), parte configurável (*base de regras*) e a parte variável (*base cognitiva*) do agente.

A parte fixa é responsável pela implementação das funcionalidades e habilidades do agente e será invariável. As funcionalidades agregadas ao desempenho do agente dependem basicamente da plataforma utilizada no desenvolvimento dos agentes. Neste trabalho buscamos desenvolver uma plataforma própria, voltada à implementação de sistema de workflow baseados na tecnologia de agentes móveis, seguindo basicamente as padronizações sugeridas no documento normativo *Móbile Agent System Interoperability*

Facilities (MASIF) do Object Management Group – OMG (OMG,1998). Este documento tem como objetivo especificar a padronização de um ambiente propício à atuação e sobrevivência de agentes móveis, bem como permitir a interoperabilidade entre agentes de diferentes fornecedores através de um conjunto de interfaces CORBA normalizada. Nesta parte introduzimos as habilidades básicas do agente e sua estrutura de comunicação com o mundo externo.

A Parte configurável é responsável por absolver a definição interna do processo que será implementada pela Ferramenta de Definição de Processo (FDP). Esta parte receberá na inicialização, a representação do processo de negócio gerada pela FDP e durante a etapa de execução de uma instância do processo, esta parte pode ser solicitada a alterar suas configuração devido ao dinamismo do processo, ou seja, durante a sua execução, novas atividades podem surgir atendendo requisitos de algum tratamento de exceção que venha ocorrer. Durante a definição do processo, o agente recebe um perfil para a sua funcionalidade, o que permite atribuir ao agente, suas habilidades para o exercício da função, porém este perfil serve apenas como base para iniciar suas tarefas.

A terceira e última parte, é a parte chamada “*variável*” ou “*dinâmica*” onde encontramos basicamente a base de conhecimento, a base de regras de produção e a lista de tarefas a serem executadas. Inicialmente o agente é criado com um conhecimento mínimo também fornecido na definição de seu papel quando atribuído o seu perfil. Porém, ao longo do seu ciclo de vida, esta base de conhecimento será trabalhada conforme as necessidades requeridas “variando” assim a sua configuração inicial.

Não existe fisicamente uma separação entre estas três partes, o que torna esta conceitualização um pouco abstrata. Há momentos em que as partes de configuração e variável se confundem, na verdade elas podem ser diferenciadas quanto ao tempo de configuração, ou seja, a parte configurável terá seus parâmetros preenchidos durante a

configuração em *tempo de definição do processo*, já a parte variável terá os seus parâmetros preenchidos durante a configuração em *tempo de execução*.

A Figura 16 ilustra o modelo proposto para o agente de software utilizado em um sistema de gerenciamento de workflow na arquitetura do WorkFlage.

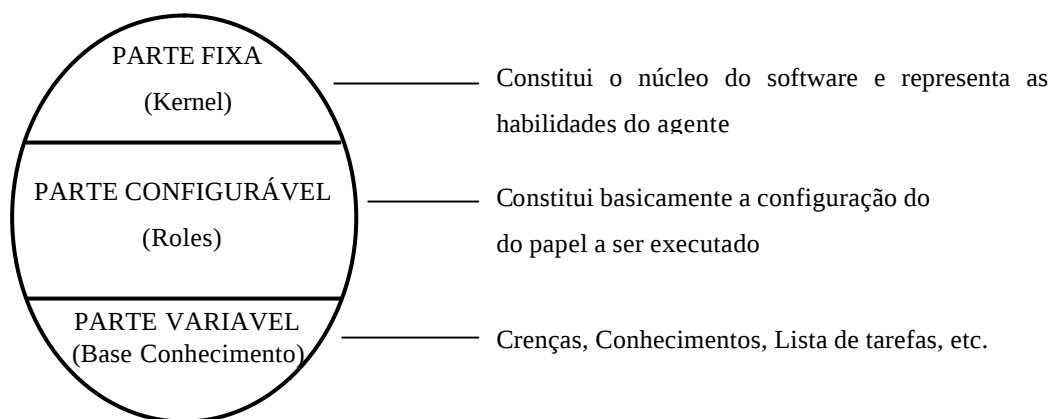


Figura 16 - Modelo do agente no Workflage

Na abordagem da WorkFlage, a cada instância da definição interna do processo, atribuímos aos agentes integrantes desta instância a função de controlar: o processo propriamente dito, suas atividades, e todo o fluxo gerado entre elas. O agente retém consigo o gerenciamento dos recursos necessários para sua função no processo. O agente processo reflete a definição do fluxo das atividades e cada agente atividade reflete um papel (função), exercido dentro do processo. Para que o agente execute plenamente seu papel, ele deve conter a qualificação e direitos necessários para a execução das tarefas desta atividade, o que será atribuído pela FDP através de um modelo de perfil.

A cada instância do processo, serão criados novos agentes para as atividades, devendo os agentes atividade receber do agente processo a sua lista de tarefa para o objetivo proposto. Os recursos necessários para criação do ambiente serão controlados por um agente integrador para o melhor desempenho das tarefas.

O modelo do agente é único e atenderá a todos os agentes envolvidos no sistema de gerenciamento de workflow, devendo apenas alterar o conteúdo da parte variável, bem como a sua configuração para o perfil em que atuará. No caso dos agentes atividades, a parte variável será composta inicialmente pela lista de tarefas para aquela atividade e a parte configurável conterá o perfil exercido por este agente e seu objetivo a cumprir. A Parte Fixa é a mesma para todos os agentes e será gerada tendo como base as classes definidas no núcleo da ferramenta de definição de processo. O Agente processo é requerido para toda as instâncias de um processo, assim, para cada instância do processo criada, deverá obrigatoriamente conter um agente processo associado, que será único e sobreviverá até a conclusão do seu objetivo.

É interessante lembrar que mesmo nos processos poucos estruturados onde a sequência das atividades não é previsível, como é o caso do comércio eletrônico, ainda assim existe uma pré-ordem estabelecida que deve ser configurada no agente processo. Podemos citar como exemplo a seguinte sequência em um processo de compra: Um agente fornecedor, não deve fornecer uma proposta sem que ele seja solicitado. Um agente comprador, da mesma forma, não deve executar o pagamento sem que tenha recebido o produto.

Os agentes são designados a exercerem uma determinada funcionalidade no sistema, porém, conforme as características do sistema modelado, alguns perfis podem ser pré-configurados e atribuídos ao agente a fim de que eles alcancem mais rapidamente todos os conhecimentos necessários para atingir os seus objetivos naquela função.

Inerente ao seu perfil, o agente recebe uma “agenda” de atribuições a ser executada por sua atividade no processo, ou seja, o agente designado a fazer compras recebe um perfil de comprador e, por conseguinte, as atribuições relativas a este perfil. Atribuições como: visitar uma loja, solicitar proposta de preço, fazer pesquisa de preço, são exemplos de atribuições que fazem parte do perfil de um agente comprador.

A Agenda constitui um conjunto de atribuições que são transformadas em tarefas a serem cumpridas pelo agente. Neste modelo, transformamos uma atribuição em um componente instanciado da classe Tarefa. Um componente ***tarefa*** constitui a unidade de trabalho exercida pelo agente.

O agente integrador é um agente especial e a sua definição fica a cargo da FDP, não sendo necessária a intervenção do desenvolvedor. Este agente deverá integrar as atividades e o processo dentro do ambiente de execução, bem como gerenciar todos os recursos locais para o sistema e os estados de execução das tarefas.

6.3 As habilidades do agente

As Habilidades dos agentes constituem parte integrante do seu kernel e são pré-requisitos para a sobrevivência do agente em uma comunidade, independente mesmo do seu papel. Nesta seção abordaremos as habilidades de um agente em relação a determinadas ações a serem tomadas. Dentre muitas outras, destacamos para o WorkFlage as seguintes habilidades:

6.3.1 Mobilidade

A Mobilidade consiste em atribuir ao agente a capacidade de locomover-se de um site para outro levando consigo os dados e o seu código, possibilitando assim grande flexibilidade no sistema. Para um ambiente onde o fluxo de trabalho e as ações a serem tomadas são incertos e imprevisíveis, a característica de mobilidade de um agente de software deixa de ser uma habilidade apenas interessante, e passa a ser imprescindível. Atualmente um aspecto distribuído caracteriza quase que a totalidade dos processos de negócios nas

organizações. Muitas empresas estão se abrindo para o mercado permitindo a criação de processos de compra e venda de forma eletrônica, de forma a possibilitar o surgimento do chamado comércio eletrônico.

Esta habilidade é conseguida através basicamente de quatros métodos internos desenvolvidos pelos agentes, sendo : viajar(), dormir(), acordar(), morrer().

A Figura 17 ilustra os métodos implementados nos kernell do agente para disponibilizar a habilidade de mobilidade no nosso modelo.

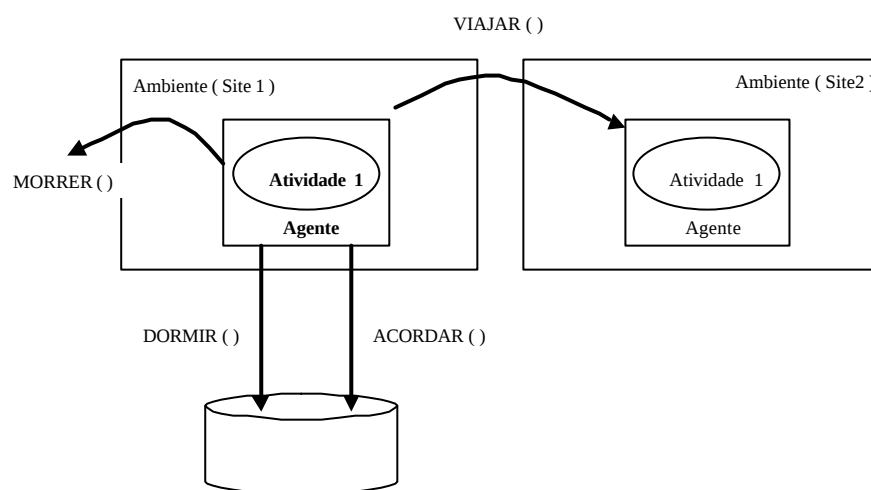


Figura 17 - Ciclo de mobilidade do agente

6.3.2 Sensibilidade ao ambiente

Através desta habilidade, um agente estabelece um relacionamento com o meio em que está inserido e, utilizando-se de um mecanismo de tratamento de eventos, ele pode perceber o mundo ao seu redor. Este mecanismo de tratamento de eventos habilita o agente a reagir a alguns eventos que ocorrem em função do fluxo normal das atividades ou de exceções que venham a ocorrer durante a execução do processo.

Além da percepção ao ambiente, o agente pode ser designado a perceber outros eventos específicos, é o caso dos componentes de um processo, onde podemos verificar que existe uma certa hierarquia na percepção dos eventos. Assim um agente processo “escuta” e reage aos eventos gerados pelos seus agentes atividades e estes, por sua vez, “escutam” e reagem aos eventos gerados pelas suas respectivas tarefas. Embora todos os agentes integrantes de uma sociedade, independentemente do tipo, percebam todos eventos, é feita uma filtragem por cada agente para que apenas os eventos destinados a ele sejam respondidos.

6.3.3 Gerenciamento de Workflow

Esta habilidade foi introduzida na parte fixa do agente, pois destina-se a modelar um conjunto de componentes (Processo, Atividade, Tarefas, etc.) que fazem parte da conceitualização dos workflows e se faz necessária para o gerenciamento dos processos. Objetos como Processo, Atividade e Tarefas, são criados pela ferramenta de definição de processo, tomando como base as características dos seus respectivos tipos implementados no kernell do agente. Para o gerenciamento de um workflow, algumas propriedades são introduzidas na definição dos componentes do processo, por exemplo: para o gerenciamento de uma atividade X, o sistema de workflow necessita conhecer suas propriedades, tais como: ID, Processo Pai, Tarefa anterior, Tarefa seguinte, Tempo previsto de execução, etc. Baseado nestas propriedades, o agente gerencia o fluxo do processo.

As definições destas propriedades estão baseadas nas orientações de implementação fornecidas pela WfMC no documento da interface 1 (WfMC,1999), sendo que algumas características inerentes ao gerenciamento por agente, baseado no modelo do WEICOT, são novas e foram introduzidas.

6.3.4 Interação com os recursos locais

O Agente deve estar apto a interagir com os recursos locais do site onde se encontra no momento. Recursos como Editor de Texto, Planilha eletrônica, I/Os, APIs e periféricos, são freqüentemente solicitados a participar das tarefas executadas pelos agentes, sempre obedecendo aos critérios de segurança pré-estabelecidos e acordados entre os integrantes do sistema. Esta habilidade permite, entre outras coisas, que o agente interaja com o usuário através de uma IHM. Muitas tarefas objetivam operar com recursos locais para poder cumprir o seu papel no processo, porém esta facilidade deve ser bem administrada, uma vez que afeta a segurança local do site. Através das configurações de “Direitos de execução” atribuída a um site, é possível impedir que um agente remoto execute algumas operações indesejadas na máquina local.

6.3.5 Aprendizagem

Esta habilidade destina-se a atribuir ao agente um certo grau de aprendizagem ao longo de seu ciclo de vida. O agente detém uma base de conhecimento e trabalha sobre uma técnica de aquisição de conhecimento baseada em fatos e regras, estabelecendo assim o seu aprendizado. Muitas plataformas de agentes utilizam técnicas de inteligência artificial bastante robustas para obter melhores resultados quanto ao aprendizado.

6.3.6 Comunicação

Os agentes devem interagir entre si demonstrando um relacionamento social entre eles. Para que este relacionamento social seja construído, é preciso que os agentes sejam

dotados de um mecanismo que possibilite uma comunicação clara e objetiva entre eles. Em determinados momentos, a cooperação entre os agentes se faz necessária para estabelecer o sucesso de um negócio, assim tão melhor será a cooperação quanto melhor for a comunicação entre os agentes. O Sistema aqui apresentado propõe a utilização do protocolo KQML (FININ,1997) para comunicação entre os agentes e uma infra-estrutura de comunicação baseada em RMI.

6.4 Modelo de perfil

Um WfMS é um sistema que define, cria e gerencia a execução do fluxo de trabalho através de software. Este sistema deve estar apto a interpretar a definição do processo e conduzir o fluxo rumo a alcançar o objetivo geral a que se propõe aquela instância (LEI,1999). Diferente dos processos automatizados com definição pré-definida, os processos dinâmicos envolvendo interesses individuais por parte dos seus agentes, baseiam-se em objetivos a serem alcançados. Um modelo completo de workflow deve abordar não somente controle de atividades, mas também como seus agentes (Atores) se relacionam com os outros em termos de conceitos, objetivos, regras, etc.

Sendo um Workflow um conjunto de atividades independentes onde cada atividade é executada por um ou vários atores para alcançar objetivos comuns, é fácil entender que os agentes exercem “*papéis*” na definição de processo, os quais são atribuídos a atividades e buscam alcançar um objetivo norteado por regras. Desta forma, um workflow pode ser visto como um conjunto de papéis se relacionando.

Na definição do processo, o WorkFlage gerará instâncias de papéis padrões, o que chamamos aqui de perfil padrão de um papel. O Perfil padrão será introduzido nos agentes na inicialização, porém este perfil será modificado ao longo da execução do sistema conforme as

necessidades de adaptação, ou seja, o objetivo e os relacionamentos de um agente podem mudar conforme sua experiência.

Em (LEI,1999) encontramos o conceito de um modelo de workflow cujo papel de cada ator define a função de um agente no workflow. Para o WorkFlage, o Modelo de Perfil é definido com base nos seguintes atributos:

Atributos	Características
Nome	Nome do perfil (função)
Descrição	Descrição resumida do perfil
Objetivo	Objetivo a ser alcançado
Qualificação	Uma ou mais pré-condição necessária para executar o objetivo
Relacionamento	Uma ou mais relação com outros papéis
Base de Conhec.	Um base de conhecimento inicial para este perfil

Tabela 2 - Atributos de um papel

6.5 Relações entre agentes

O Relacionamento entre papéis, estabelece de certa forma um modelo de prestação de serviço entre eles, onde um agente solicita serviços a outro e também fornece serviço. O agente solicitante deve conhecer os seus prestadores de serviços bem como aquele que o solicita. Este processo pode ser comparado as Redes Contratuais (Contract Net) onde a negociação entre os agentes é a troca de mensagens, baseada num dado protocolo, com vista ao estabelecimento de um acordo. Nestas redes, os agentes coordenam as suas atividades através de contratos para atingir determinados objetivos. Um agente organizador anuncia tarefas para um grupo de agentes contratantes potenciais. Estes respondem ao agente organizador com licitações para a execução das tarefas. O agente organizador avalia as propostas e escolhe a melhor, celebrando depois o contrato. No nosso modelo. os agentes

adotam este modelo de prestação de serviço, pois julgamos mais próximo dos relacionamentos nas diversas atividades em uma organização. Para desempenhar seu relacionamento, os agentes utilizarão as suas habilidades de comunicação.

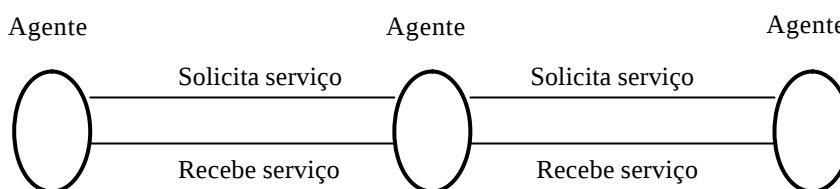


Figura 18 - Relação entre agentes

6.6 Base de Conhecimento

Durante seu ciclo de vida, os agentes lidam com bastantes informações e situações que necessitam de conhecimento prévio para agir. Diferentemente dos sistemas de gerenciamento de workflow tradicionais, onde é estabelecido um percurso preliminar do fluxo de trabalho, nos sistemas baseados em agente de software, existe a possibilidade de que este percurso seja construído on-line. A tarefa a ser executada posteriormente dependerá da ação tomada anteriormente. Por exemplo, diante de um pedido de compra de material, o agente exercendo o papel (função) de comprador, pode efetivamente comprar a quantidade solicitada ou, diante da informação que este produto já existe em estoque em uma filial da empresa, parar o processo de compra e solicitar a transferência do material para seu solicitante.

A tomada de decisão de um agente está baseada em duas importantes habilidades: a sensibilidade ao meio ambiente e o conhecimento ao seu objetivo. Ocorrendo uma alteração no fluxo normal do processo, ainda assim o alvo deve ser almejado e, para tanto, o agente toma uma decisão seguindo um outro fluxo sem perder de vista o seu objetivo. Ao longo do seu ciclo de vida, o agente absorve conhecimento, tornando-o cada vez mais apto a decidir com maior rapidez e segurança.

6.7 Regras de produção

Os sistemas de produção representa, uma das maneiras mais utilizadas para representar o conhecimento declarativo (ideal para modelar o raciocínio humano). Um sistema de produção pode ser definido como um conjunto de regras, uma memória de trabalho onde são armazenados os fatos e um algoritmo de encadeamento progressivo para gerar novos fatos a partir dos fatos existentes. Entre as várias propostas de prover a capacidade de raciocínio baseada em regras de produção, o método utilizando o paradigma de programação orientada a objeto vem sendo largamente estudado.

A possibilidade de uma entidade poder pensar ou tirar conclusões sobre um conjunto de fatos em um domínio específico lhe dá a autonomia necessária para decidir qual o próximo passo a dar, sem a necessidade de auxílio externo.

A capacidade de raciocínio permite que o agente se relacione na comunidade onde vive, podendo assim compreender uma idéia a partir de uma estrutura de dados conhecida por ele.

Sabendo os objetivos que devem ser alcançados e mantendo um conjunto de fatos e dados que permita formar uma base de conhecimento sobre um determinado domínio, um agente pode decidir, dentre as alternativas de ações possíveis, aquela que melhor lhe traga resultados.

No modelo do WorkFlage, utilizamos a integração entre objetos e regras de produção baseada na linguagem JAVA (JEOPS) (FIGUEIRA; RAMALHO,2000). O objetivo de um agente reflete uma regra imposta para determinar se foi ou não alcançado.

Uma regra pode ser definida como um conjunto de **campos**, **operadores** e **valores** constituindo uma pergunta que será submetida a uma base de conhecimento Sendo um processo de compra de material, o objetivo geral desta instância do processo de compra seria :

Ex.: “**Comprar LIVRO com preço <= R\$90,00**”

Este objetivo será transformado em uma regra da seguinte forma:

```
Regra1:: {
  Ação : "COMPRAR";
  //Campos:
    Produto: "LIVRO";
    Referencia: "3453-00";
  //Atributos:
  Preço: 90,00;
  Operador: "<=";
  Moeda: "R$"; }
```

Assim :

```
Regra 1 =  COMPRAR("LIVRO:3453-00",
                  Condição(Preço, "<=", "R$", 90,00))
```

6.8 Conclusão

Nesta seção apresentamos a modelagem da arquitetura Workflage que estende as características do modelo do WEICOT, adaptando-o a uma visão voltada ao desenvolvimento de sistemas de workflow baseado em agentes de software.

A arquitetura proposta está baseada na cooperação de três agentes básicos para o gerenciamento do workflow: O Agente Processo, o agente Atividade e um agente Integrador que será introduzido para fazer o controle do fluxo de informações e relacionamentos entre os agentes. Os Agentes executam suas funções em um ambiente favorável à interação e ao controle de fluxo de informações entre eles.

A Arquitetura do Workflage propõem também a modelagem dos agentes de software composta de três partes: Uma parte chama de *kernell*, onde está concentrada toda a funcionalidade básica do agente, uma parte que concentra todas as configurações iniciais para o agente e absorve as “regras” que compõem o perfil do agente e uma parte chamada de base cognitiva, onde será desenvolvido todo um conhecimento do agente relativo a função que exerce no processo.

No capítulo seguinte, apresentamos a ferramenta de definição que, baseada na arquitetura proposta neste capítulo, colocará em prática o desenvolvimento dos sistemas de workflow.

7 FERRAMENTA DE DEFINIÇÃO

Conforme descrevemos anteriormente, um Sistema de Gerenciamento de Workflow necessita de uma ferramenta que traduza o fluxo do processo para uma linguagem que seja compreendida por sua máquina de gerenciamento. Nesta seção apresentamos a ferramenta WorkFlage que traduz a descrição de um processo em uma representação interna para que seja gerenciado por um sistema distribuído e baseado em agentes de software.

O processo de definição tem início com a etapa de criação do projeto e deve cumprir uma seqüência de três outras etapas que serão descritas no decorrer desta seção. Para criação dos agentes, a WorkFlage pode ser configurada para ter como base um conjunto de classes de uma plataforma de desenvolvimento de agentes já existente, (como o AGLET da IBM (LANGE,1996), o ZEUS (NWANA,1999), etc.), porém desenvolvemos uma biblioteca de classes própria que será tida como default.

A Ferramenta WorkFlage foi totalmente desenvolvida na linguagem Java e gera os códigos também nesta linguagem, portanto, é necessário que os usuários tenham, nas máquinas onde vai executar o sistema, a plataforma JDK da Sun.

A WorkFlage traduz o modelo do processo em um sistema multiagente e cria também de forma automática o ambiente de execução para este processo. Um processo será transformado no agente “Processo” e suas atividades nos agentes “Atividades”. A Figura 19 mostra a transformação de um processo e suas atividades em agentes.

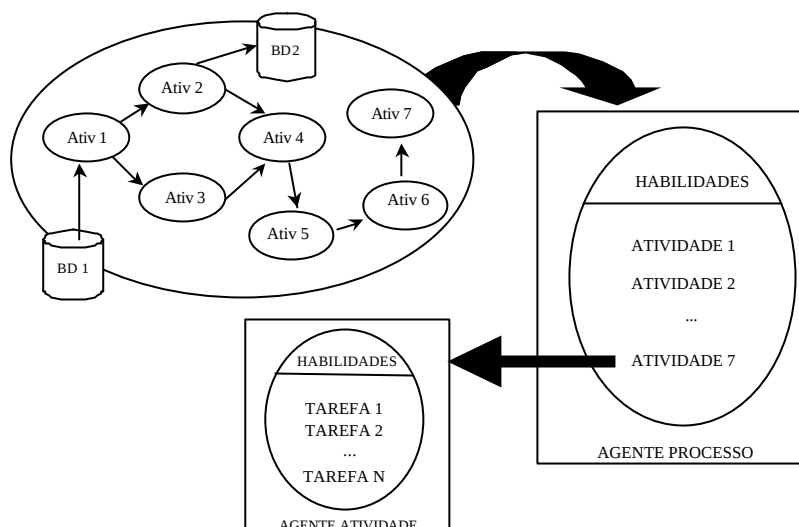


Figura 19 - Mapeamento das atividades no o agente

A definição de um processo será vista pela ferramenta como um projeto. Um projeto é criado a cada definição, o que gera um arquivo estruturado baseado na tecnologia XML para ser recarregável posteriormente, caso necessário. A Tela principal da ferramenta de definição WorkFlage, apresenta quatro Sub-Telas, representando cada uma, uma etapa do processo de definição:

- a) **Tela de definição de Projeto** – Nesta tela devem ser informadas todas as propriedades do projeto de definição;
- b) **Tela de definição do Processo** - Nesta tela serão informadas as atividades pertencentes a este processo, as propriedades do processo e o código de tela do processo.
- c) **Tela de definição de Atividades** – Nesta tela serão definidas as atividades, qual o perfil de cada atividade, qual a agenda básica para cada agente atividade e a sequência normal das tarefas;
- d) **Tela de definição de Tarefas** - Nesta tela serão definidas as diversas tarefas

que compõem este processo. Uma tarefa gera um objeto Tarefa para a agenda de uma atividade;

- e) **Tela de Perfis** – Nesta tela será disponibilizada uma opção para que o usuário carregue ou crie novos perfis para uma determinada atividade.

A Figura 20 mostra a tela principal da ferramenta e a tela de projeto. A seguir será apresentada uma descrição das funcionalidades de cada tela.

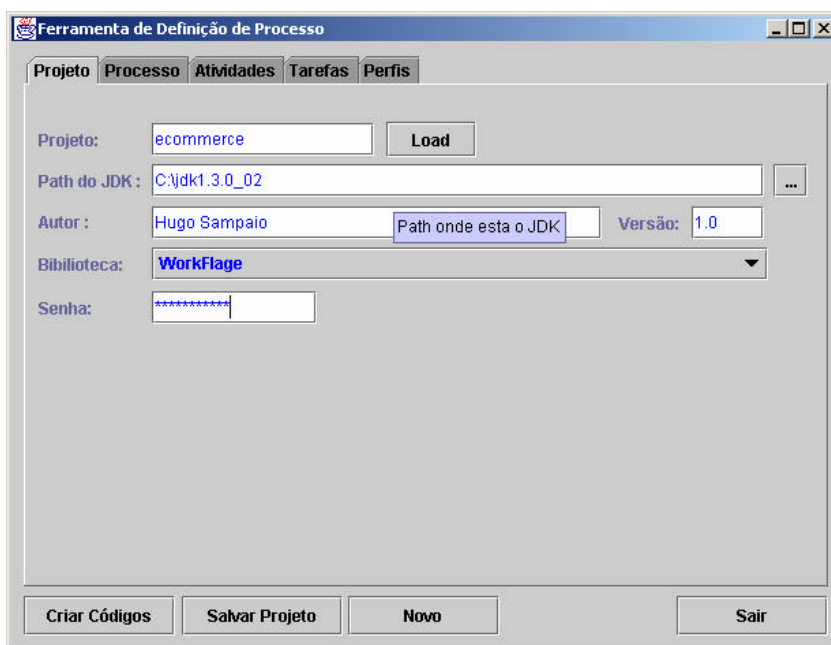


Figura 20 - Tela principal da ferramenta de definição

7.1 Tela de definição do projeto

Na tela de definição do projeto, registramos as características do processo modelado. Após concluir a definição do processo, cada projeto gera um arquivo XML com os seus atributos, um conjunto de classes que representa os agentes do processo e um arquivo de inicialização para executar o processo. Para definirmos um projeto, preenchamos na tela os seguintes campos:

- ? **Projeto** - Nome do projeto que deve ser alfanumérico e com tamanho máximo de 15 caracteres. Este nome identifica o projeto, cria uma pasta com este nome e gera um arquivo XML com as configurações deste projeto para futuras modificações.

- ? **Botão Load** - Este botão permite que seja carregado para ferramenta um projeto já criado. A carga de um projeto permite a edição do projeto original e uma nova geração.

- ? **Path do JDK** - Neste campo deve ser informado o caminho absoluto do Kit de desenvolvimento do Java. O sistema localiza o caminho atual do JDK, mas possibilita que o usuário crie em outro path.

- ? **Autor** - Este campo informa o autor da definição para controle.

- ? **Versão** - Este campo informa a versão atual da definição.

- ? **Biblioteca** - Neste campo o usuário pode escolher qual base de classes será utilizada para gerar os agentes. Definimos uma biblioteca de classes própria da ferramenta do WorkFlage.

- ? **Senha** – Este campo atribui uma senha ao projeto, de forma a impedir que outras pessoas não autorizadas alterem a definição do projeto. Esta senha deve ser alfanumérica com o máximo de 15 caracteres.

Ainda relativo ao projeto, o usuário dispõe de quatro botões, sendo:

- ? **Botão Salvar projeto** - Este botão permite que as alterações feitas em toda a ferramenta, sejam gravadas no arquivo de projeto, porém o sistema só será gerado quando solicitado pelo botão “Criar Códigos”.
- ? **Botão Criar Códigos** - Este botão não permite gerar o sistema pronto para executar. É importante lembrar que nesta versão da ferramenta, não colocamos nenhum mecanismo para debug do código Java, portanto, caso haja algum erro de sintaxe, este deve ser tratado antes da compilação na ferramenta.
- ? **Botão Novo** – Este botão permite ao usuário criar um novo projeto ou limpar toda a memória da ferramenta levando todas as variáveis internas para posição inicial.
- ? **Botão Sair** – Este botão permite ao usuário encerrar a ferramenta de definição de processo abandonando toda a edição feita até este momento caso não tenha sido gravada em arquivo.

7.2 Tela de definição do processo

A Tela de definição de processo possibilita ao usuário criar o agente processo. Um agente processo herda as características da classe CAgente da biblioteca utilizada para o desenvolvimento dos agentes, indicada na tela de projeto através da extensão da classe CProcesso do Workflage. Para cada instância do processo existe apenas um agente processo, o que dá a ele a característica de identificar a instância do processo em execução. Na tela de definição do Processo atribuímos os seguintes valores:

- ? **Nome do Processo** – Este campo define um nome para o processo. O nome do processo deve ser alfanumérico com tamanho máximo de 15 caracteres.
- ? **Início do fluxo** – Este campo indica para o processo qual atividade inicia o fluxo de trabalho no processo. Quando o valor é zero, indica que o agente processo deve iniciar.
- ? **Este processo tem tela?** – Este campo identifica se o processo tem ou não uma tela para interagir com o usuário local. Caso este processo necessite interagir com o usuário, o código da tela deve ser acrescentado em uma tela específica acionada pelo botão Código de tela.
- ? **Botão Código de tela** – Este botão chama uma tela para que seja colocado o código JAVA da tela.
- ? **Descrição** - Neste campo é feita uma breve descrição do processo.
- ? **Atividades** – Esta área de texto possibilita ao usuário criar e/ou carregar agentes atividades para o processo criado anteriormente. Cada atividade representa um agente e sua ordem na lista não indica a sequência do fluxo no processo.
- ? **Alias** – Esta área de texto, possibilita o usuário acrescentar para cada atividade criada, um “alias” que será a referência do agente no processo. A ferramenta preenche automaticamente o campo, mas o usuário pode alterar, caso ache necessário.

? **Botão Add** – Através deste botão o usuário adiciona o processo no projeto.

Figura 21 - Tela de definição de processo

7.3 Tela de definição de atividades

A Tela de definição de atividades possibilita ao usuário criar a classe dos agentes atividades no processo. Uma atividade estende a classe CAgente da biblioteca utilizada para o desenvolvimento dos agentes indicada na tela de projeto. Uma atividade representa uma função (papel) no processo e pode assumir, durante a definição, um perfil pré-formatado. Na tela de definição de Atividades verificamos os seguintes componentes:

? **Atividade** - Nesta área de texto são listadas todas as atividades existentes neste projeto, ou seja, serão listadas todas as atividades definidas anteriormente na tela de processo. Através de um menu popup, as atividades podem ser editadas.

- ? **Agenda com execução seqüencial** – Uma atividade pode ter a execução de sua lista de trabalho “agenda” de forma *seqüencial* ou não. Esta caixa possibilita ao usuário indicar para a ferramenta de processo que a atividade selecionada tem ou não a execução seqüencial. Quando uma atividade tem uma tela, obrigatoriamente a execução não será seqüencial, logo este campo é automaticamente desabilitado na criação de uma tela para atividade.
- ? **Menu popup** – Este menu possibilita executar algumas operações sobre a atividade selecionada.
- ? **Criar tela** - Uma atividade gera um agente atividade e caso este agente necessite interagir com o usuário, uma interface homem máquina deve ser criada. Esta opção do menu popup chama uma tela para incluir o código JAVA para a tela do agente atividade.
- ? **Renomear** – Esta opção do menu possibilita trocar o nome do agente atividade selecionada. Esta ação exige que o projeto seja recompilado.
- ? **Apagar** – Esta opção do menu possibilitar apagar um agente atividade. Esta ação remove todas as referências no projeto a esta tarefa.
- ? **Editar** – Esta opção menu possibilita a edição da atividade selecionada. Uma tela será apresentada com o código completo do agente atividade, permitindo a alteração dos atributos para esta atividade.
- ? **Botão Atribuir perfil** – Este botão chama uma tela de perfis para que seja atribuído um perfil já cadastrado para esta atividade.

- ? **Botão Atribuir tarefas** – Este botão chama uma tela para que seja possível incluir, nesta atividade, algumas tarefas. Uma atividade pode criar novas tarefas ou carregar um perfil com tarefas já pré-definidas.

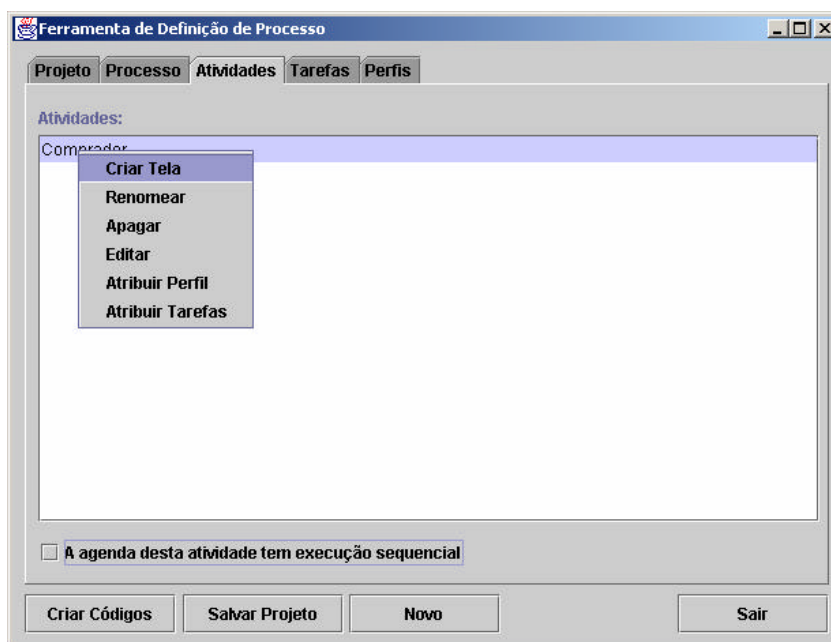


Figura 22 - Tela de definição de atividades

7.4 Tela de definição de tarefas

A Tela de definição de tarefas possibilita ao usuário criar as tarefas do processo. Uma tarefa participa de uma ou várias atividades no processo e durante a execução de atividade, apenas uma tarefa será executada por vez, o que exige que internamente a atividade, o fluxo de trabalho seja sequencial. Na tela de definição de Tarefas verificamos os seguintes componentes:

- ? **Tarefa** - Nesta área de texto são listadas todas as tarefas existentes neste projeto. A partir deste campo, através de um menu popup, as tarefas podem ser editadas.

- ? **Menu popup** – Possibilita executar algumas operações sobre a tarefa selecionada.
- ? **Código** – Este menu chama uma tela para a inclusão do código JAVA para a tarefa. Este código representa o corpo de uma classe que será instanciada da classe CTarefa e herda todas as características de uma tarefa para o WorkFlage.
- ? **Renomear** – Este menu possibilita trocar o nome da tarefa selecionada. Esta ação exige que o projeto seja recompilado.
- ? **Apagar** - Este menu possibilitar apagar uma tarefa. Esta ação remove todas as referências no projeto a esta tarefa.
- ? **Editar** – Este menu possibilitar a edição da tarefa selecionada. Uma tela será apresentada permitindo a alteração dos atributos para esta tarefa.

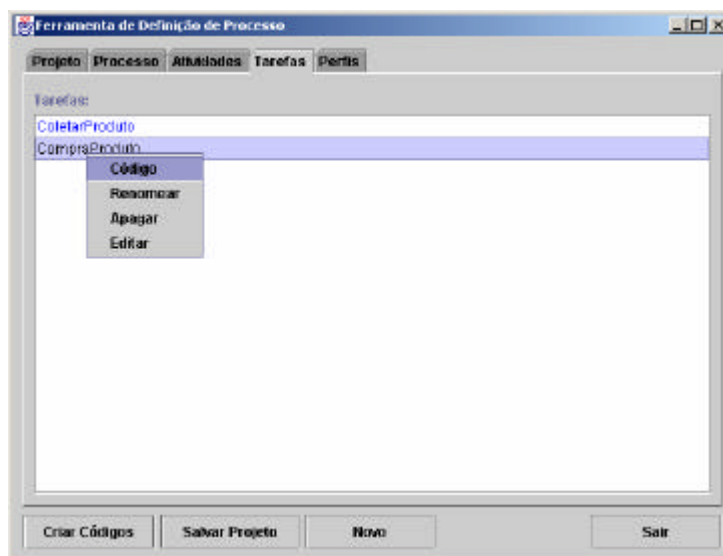


Figura 23 - Tela de definição de tarefas

7.5 Tela de definição de perfis

A definição de um perfil facilita a criação dos agentes. Um perfil basicamente é utilizado para modelar o agente na hora da sua criação. Ele reúne as habilidades específicas para um determinado agente. Em um sistema de compras, por exemplo, nem todos os agentes necessitam negociar preço ou efetuar a compra do produto, este tipo de característica se enquadra mais com o agente comprador. Na WorkFlage, a tela de perfil possibilita criar um “modelo” para os agentes. Estes modelos (perfis) não são exclusivos de um projeto, pois a ferramenta de definição cria um banco de perfis que podem ser utilizados em outros projetos. Os campos básicos para a tela de perfis são os seguintes:

Ferramenta de Definição de Processo

Projeto | Processo | Atividades | Tarefas | **Perfis**

Nome: Load

Descrição:

Autor:

Qualificação:

Variáveis internas:

```
CProduto produto = new CProduto();
String acao = new String(Comprar);
int contador = 0;
```

Perfis:

Comprador	Renomear Apagar Editar Atribuir Tarefas
Vendedor	

Deletar Add

Criar Códigos Salvar Projeto Novo Sair

Figura 24 - Tela de definição de perfis

- ? **Nome** – Este campo define um nome para o perfil. O nome do perfil deve ser alfanumérico com tamanho máximo de 15 caracteres.
- ? **Autor** - Este campo informa o autor da definição para controle.

- ? **Qualificação** - Este campo indica uma tarefa que deve ser executada na inicialização do agente para que ele tenha algum conhecimento prévio. Por exemplo: Para o agente comprador iniciar suas tarefas, ele deve antes saber quais os sites de venda disponíveis e qual o tipo de produto oferecido por cada um.
- ? **Variáveis internas** – Neste campo, podem ser previamente indicadas algumas variáveis que devem fazer parte do modelo do agente. A sintaxe utilizada é a mesma da linguagem Java.
- ? **Perfis** – Lista dos perfis no banco de perfis da ferramenta WorkFlage.
- ? **Menu popup** – Este menu possibilita executar algumas operações sobre o perfil selecionado.
- ? **Renomear** – Esta opção do menu possibilita trocar o nome do perfil selecionado. Caso este perfil seja utilizado no projeto, após renomear, o projeto deve ser recompilado.
- ? **Apagar** – Esta opção do menu possibilita apagar um perfil. Esta ação remove todas as referências no projeto a este perfil. Caso este perfil seja utilizado no projeto, após esta ação o projeto deve ser recompilado.
- ? **Atribuir tarefa** – Através desta opção do menu popup, será atribuído um conjunto de tarefas que fazem parte deste perfil. Esta opção exige que as tarefas envolvidas já estejam criadas.

A figura 25 mostra um exemplo da tela de código de uma tarefa. Esta tela será apresentada quando a opção “código” do menu popup for selecionada. Uma tela semelhante a esta será mostrada quando selecionada a opção “Criar tela” do menu popup da tela de

definição de atividades.

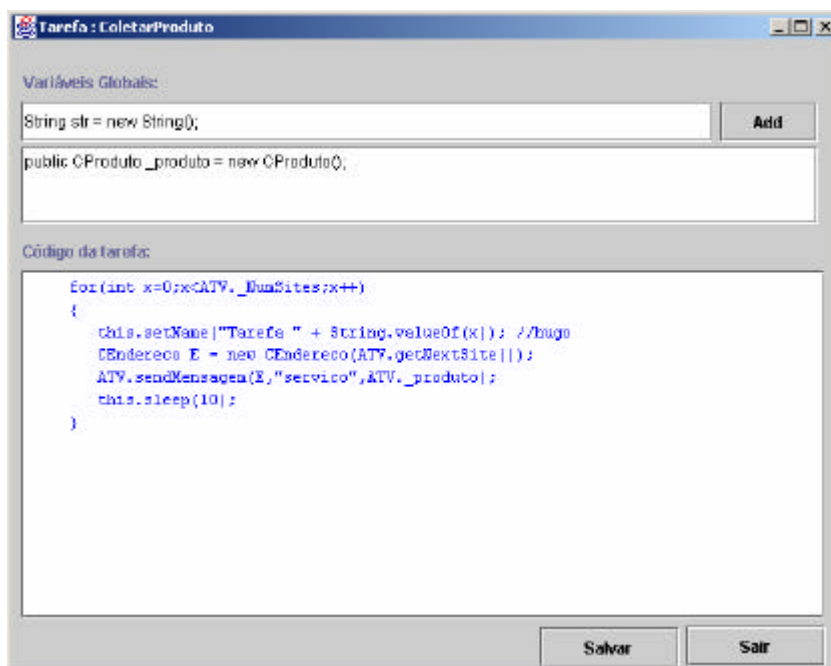


Figura 25 - Tela de código de tarefa

7.6 O Arquivo de projeto

Após a criação de um projeto, um arquivo XML será criado representando todos atributos do projeto. Este arquivo será carregado quando o botão “Load” for selecionado na tela de projeto e todos os campos do projeto, inclusive os campos das telas de Tarefa,Atividade e Processo serão preenchidos. A carga de um projeto possibilita também a criação de referências entre todos os integrantes do projeto. O código abaixo mostra um exemplo de um arquivo XML de um projeto.

```

<?xml version="1.0" encoding="UTF-8" ?>

- <ecommerce>
- <PROJETO>
    <NomeProjeto>ecommerce</NomeProjeto>
    <PathJDK>C:\jdk1.3.0_02</PathJDK>
    <Autor>Hugo Sampaio</Autor>
    <Biblioteca>WorkFlage</Biblioteca>
    <Versao>1.0</Versao>
</PROJETO>
- <PROCESSO>
    <NomeProcesso>Processo</NomeProcesso>
    <TemTela>0</TemTela>
    <InicoFluxo>0</InicoFluxo>
    - <Descricao>
        - <![CDATA[
            Este processo foi definido para exemplificar a utilizacao da
            Ferramenta WorkFlage. Ele modela um processo de compra de
            Insumos para uma empresa.
        ]]>
    </Descricao>
</PROCESSO>
- <ATIVIDADES>
    <ATIVIDADE nome="Comprador" alias="Comprador" idxatv="0"
    execContinua="0"/>
</ATIVIDADES>
- <TAREFAS>
    <TAREFA nome="ColetarProduto" alias="Coleta" idxatv="0" idxtar="0" />
    <TAREFA nome="CompraProduto" alias="Compra" idxatv="0" idxtar="1" />
</TAREFAS>
<VARIABLES_ATIVIDADES />
- <VARIABLES_TAREFA>
    - <VARIABLE idxtar="0">

```

```

- <![CDATA[
    public CProduto _produto = new CProduto();
]]>

</VARIABEL>

+<VARIABEL idxtar="1">

<CODIGOS_ATIVIDADES />

- <CODIGOS_TAREFAS>

- <COD_TAREFA idxtar="0">

- <![CDATA[
    for(int x=0;x<ATV._NumSites;x++)
    {
        this.setName("Tarefa " + String.valueOf(x)); //hugo
        CEndereco E = new CEndereco(ATV.getNextSite());
        ATV.sendMensagem(E, "servico", ATV._produto);
        this.sleep(10);
    }
]]>

</COD_TAREFA>

+ <COD_TAREFA idxtar="1">

+ <TELAS_ATIVIDADES>

<TELAS_TAREFAS />

+ <VARIAVEIS_TELA_ATIVIDADE>

<VARIAVEIS_TELA_TAREFA />

+ <CODIGOS_TELA_ATIVIDADES>

<PERFIS />

</ecommerce>

```

Tabela 3 - Arquivo de projeto

7.7 Ambiente de Execução

A definição de um processo está vinculada à criação de um ambiente que favorece a sobrevivência dos agentes. Este ambiente é baseado na arquitetura cliente servidor, onde temos uma máquina servidora e as demais máquinas que compõem a rede, como cliente deste servidor (também chamados de sites).

Todas as máquinas que participam de um processo, constituem um *contexto* no sistema e devem estar executando ou um site cliente ou um site servidor. Somente um site servidor pode iniciar um processo. Um site cliente, ao inicializar, deve obrigatoriamente informar a qual site servidor ele deve se conectar. Com a carga de um processo em um servidor, os agentes são criados e recebem imediatamente as informações do contexto atual. Ao longo de suas vidas, os agentes podem migrar para outros sites, conforme as suas necessidades, recebendo na sua chegada as informações do contexto atual onde ele está. Um agente sempre deve estar ligado a um contexto, seja um servidor ou um site cliente.

A conexão dos sites ao servidor forma um backbone virtual que é entendido e utilizado pelos agentes como canal de comunicação. Todos os componentes do sistema (agentes e sites), recebem um endereço único que é composto pelo endereço do contexto atual mais um número ID. Através de um endereço único, o agente é identificado tornando possível o mecanismo de troca de mensagens entre os agentes e os sites.

No primeiro momento, um processo que foi definido é criado em uma única máquina e depois será estendido (utilizando a característica de mobilidade dos agentes) às demais.

Após a distribuição de todas as atividades nas máquinas integrantes do processo, o agente processo deve iniciar a primeira atividade. Uma mensagem é enviada para o agente integrador que, por sua vez, informa a todos os agentes que estão “dormindo” que o processo está ativo. Se na definição existir uma atividade sem dependência, esta será iniciada imediatamente mesmo que esteja em máquinas diferentes.

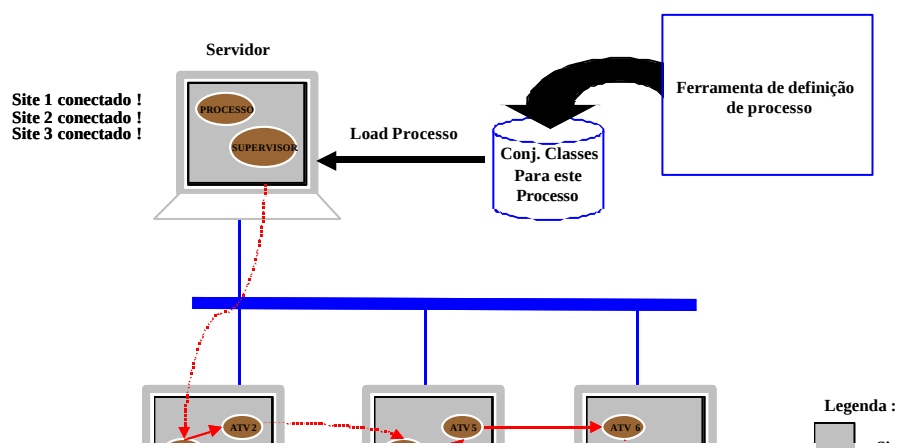


Figura 26 - Processo distribuído

Se algum problema de comunicação ocorrer, o agente supervisor deve consultar a definição interna do processo, para providenciar que uma outra máquina assuma o seu papel, desde que esta máquina tenha os recursos necessários para execução das atividades da máquina sem comunicação.

Durante a execução, os agentes atividade devem passar a cada instante o status das suas tarefas para o agente integrador que providenciará a atualização do monitor do processo. O monitoramento do sistema será feito através de um servidor.

Os agentes atividade podem, a qualquer momento, decidir migrar para outra máquina se achar que os recursos na outra máquina estão mais disponíveis do que na máquina atual. A Figura 26 mostra a arquitetura física do WorkFlage implementada em um processo distribuído.

7.8 O Servidor Workflage

Baseado nas orientações do MASIF, o WorkFlage implementa o conceito de *Place* como unidade básica para formação do ambiente multiagente. Um place caracteriza um contexto no qual o agente está inserido e através do qual o agente pode desempenhar todas suas funcionalidades e habilidades. Fora do place, o agente não tem vida, pois é no place onde

está todo o ambiente favorável à sua sobrevivência.

Uma definição de processo gerada pelo WorkFlage está baseada na definição de dois tipos de Place: o *Place Servidor* e o *Place Cliente*. (também chamado de Site).

O *Place Site* constitui a unidade base de formação de um contexto. Este place tem um elo de comunicação que o interliga aos outros Places e, desta forma, gera um barramento que será o canal de comunicação do ambiente WorkFlage. Um Place Site, ao ser criado, recebe automaticamente um endereço próprio e deve obrigatoriamente conhecer o endereço do seu servidor.

O *Place Servidor* constitui o ponto de partida para execução de uma definição de processo. Somente no place servidor é possível carregar uma instância do processo. O Servidor gera todos os agentes passando para eles o contexto onde estão inseridos. Embora tenha algumas funções diferenciadas do Site, o Servidor detém todas as funções de um Site, o que permite que ele também possa funcionar como um Place Site.

A Figura seguinte mostra a tela do servidor WorkFlage, bem como suas funcionalidades.

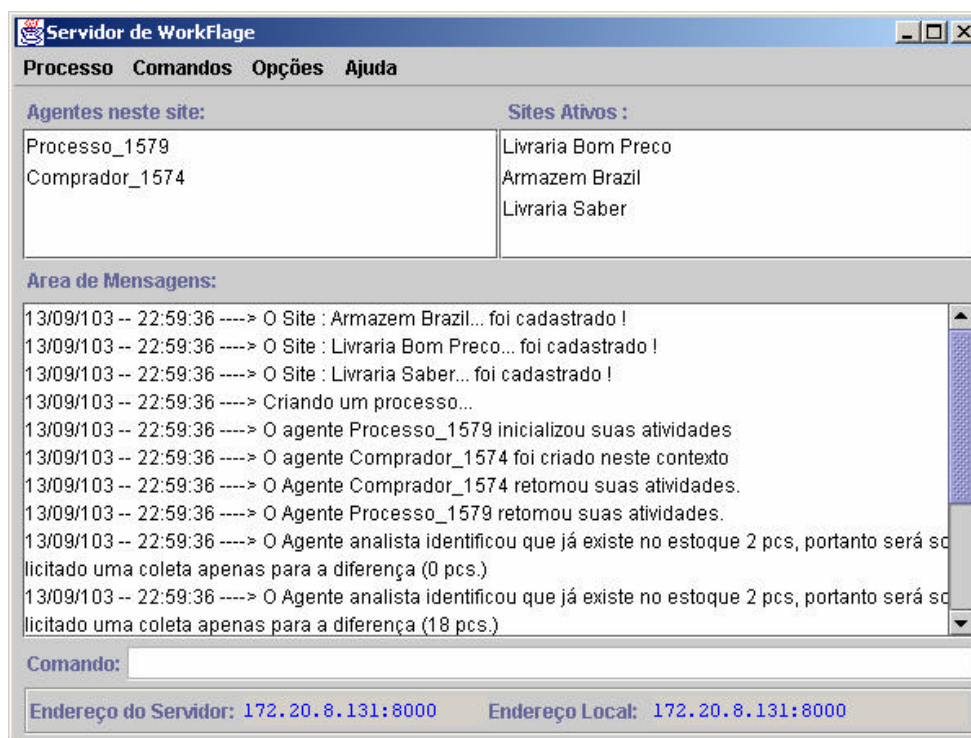


Figura 27 - Tela do servidor WorkFlage

A tela do servidor do WorkFlage, assim como a tela do Site, está dividida em três áreas, sendo Área de Endereço dos Agentes, Área de Endereços dos Sites e Área de Mensagem.

A área de endereço dos agentes mostra os endereços de todos os agentes que estão ativos no site. Quando um agente chega no site, ele é registrado na base de agentes e o seu endereço é mostrado nesta área. É através desta área que o usuário pode interagir com o agente.

A área de endereço dos sites mostra todos os sites cadastrados no sistema e ativos. Quando um site é criado em uma máquina qualquer, ele solicita o cadastro do seu servidor. Após o cadastro na base de sites do servidor, o agente integrador percebe que escuta uma alteração no barramento e informa a todos os participantes a nova estrutura. Neste momento, os sites cadastram o novo site na área de endereço. Da mesma forma, quando um site sai do backbone, todos recebem a informação para retirá-lo da sua base de site.

Na área de Mensagem, o sistema mostra todas as mensagens referentes ao funcionamento do sistema, bem como algumas mensagens enviadas pelos agentes para que seja apresentada nesta área.

Na barra de menu do servidor, encontramos os seguintes itens de menu:

- a) **Processo** – Disponibiliza acesso às operações relativas ao manuseio dos processos, tais como: Abrir, Criar novo, Carregar, etc.
- b) **Comandos** - Disponibiliza alguns comandos pré-formatados para o sistema de WorkFlage, tais como : Imprimir ficha do agente, Limpar a área de

mensagem,etc.

- c) **Opções** – Neste menu é possível acessar a ferramenta de definição de processo e outras opções para configurar o ambiente.
- d) **Ajuda** - Este menu disponibiliza acesso a alguns tópicos de ajuda para a operação do WorkFlage.

O Servidor disponibiliza ainda um campo de comandos, onde o usuário pode enviar comando que necessite de passagem de parâmetros. Por exemplo, o envio de uma mensagem para outro site, e um menu pop-up que, quando ativo na seleção de um agente, apresenta um série de opções de comando para aquele agente, tais como: enviar para outro site, dar um ping no agente, etc.

7.9 Implementação

A ferramenta de definição de processo WorkFlage, foi desenvolvida baseada na linguagem JAVA (JDK2) e tem como núcleo um conjunto de classes baseado no documento normativo MASIF do OMG que dar uma característica de mobilidade aos agentes. Durante o desenvolvimento da ferramenta, outras plataforma de desenvolvimento foram observadas, tais como Aglet (LANGE,1996) e a SOMA (Secure and Open Mobile Agent) desenvolvida na Universidade de Bolonha na Itália (BREWINGTON,1999). Embora ambas sigam as orientações da MASIF, nós optamos por construir um conjunto de classes de forma a adaptá-las para a utilização de agentes de software em Sistemas de Gerenciamento de Workflow.

O conjunto de classes que compõem as funcionalidades do Workflage estão reunidas em um pacote chamado “workflage”. Este pacote é composto de 19 classes onde citamos as mais significativas:

CAgente - Classe abstrata para geração do agente do Workflage;

CAtividade - Classe que modela um atividade do processo;

CContexto - Classe que representa o contexto em que o agente está envolvido;

CEndereco - Classe auxiliar para tratamento de endereços dos sites;

CEventoDeWorkflage - Classe que modela todos os eventos do Workflage;

CLoaderDeProcesso - Permite a carga de um processo no ambiente do Workflage

CMensagem - Classe que faz o tratamento das mensagens entre agentes;

CProcesso - Classe que modela um agente processo;

CServidor- Classe que gera o servidor do WorkFlage;

CSite - Classe que gera os sites do WorkFlage;

CTarefa - Classe que modela uma tarefa no processo;

7.9.1 A geração da ferramenta

Após a definição de um processo, a ferramenta cria uma estrutura de arquivo na raiz do JDK e gera automaticamente os seguintes arquivos e estrutura:

- ? Caso ainda não exista a estrutura do sistema de WorkFlage, a ferramenta cria uma pasta chamada “WorkFlage” com algumas sub-pastas internas no path raiz do JDK. Esta é a pasta HOME do sistema;
- ? Para cada definição de processo gerada pela ferramenta, uma pasta com o nome do projeto é criada dentro da sub-pasta Público na raiz da pasta HOME. (Ex.: C:\<%HOME%>\Public\<nome do projeto>);
- ? Um arquivo **Servidor.bat** necessário à inicialização do site servidor será criado;
- ? Um arquivos **Cliente.bat** necessário à inicialização do site cliente. Este arquivo, juntamente com a pasta Site, deve ser distribuído entre todos as máquinas clientes integrantes do processo;
- ? Todas os arquivos “.class” que representam o processo estarão na pasta

C:\<%HOME%>\classes;

- ? Uma pasta contendo os arquivos fontes (.java) das classes do processo estarão na pasta C:\<%HOME%>\src\<nome do projeto>;
- ? Os demais arquivos necessários, devem estar na pasta C:\<%HOME%>\bin.

Estamos buscando a cada dia melhorar o nível de interação entre a ferramenta de definição e o usuário, portanto pretendemos disponibilizar o código fonte, como uma forma de difundir os estudos na área e agregar significativas melhorias.

```
@echo off
SET JDK_HOME=C:\jdk1.3.0_02
SET APP_NOME=WorkFlage\cliente
@rem -----
@SET APP_HOME=%JDK_HOME%\%APP_NOME%
@SET JDK_CLASS=%JDK_HOME%\lib;
SET JAR_CLASS=%APP_HOME%\classes\WorkFlage.jar;
SET APP_CLASS=%APP_HOME%\classes;
@SET APP_PUBLICO=%APP_HOME%\publico;
@SET APP_TEMP=%APP_HOME%\temp;
@rem -----
SET LOCAL_CLASSPATH=
    %JDK_CLASS%%APP_CLASS%%JAR_CLASS%%APP_PUBLICO%%APP_TEMP%
@rem -----
@%JDK_HOME%\bin\java -classpath "%LOCAL_CLASSPATH%"_
    workflage.CSite "127.0.0.1:8000" "Nome do Site"
```

Tabela 4 - Arquivo de inicialização do site cliente

7.10 Conclusão

Neste capítulo apresentamos o WorkFlage como uma proposta de arquitetura para o desenvolvimento de Sistemas de gerenciamento de Workflow baseados em Agentes de Software incorporando todas as características do modelo do WEICOT.

Juntamente com a arquitetura Workflage, estamos implementando uma ferramenta de definição de processo que possibilita modelar, implementar e executar um processo de negócio baseado em agentes de software.

Neste capítulo apresentamos um modelo para o agente utilizado no WEICOT e

introduzimos a habilidade de gerenciamento de workflow através de uma classe Java totalmente desenvolvida para o tratamento dos eventos de um workflow. Apresentamos também a arquitetura distribuída de hardware que é utilizada pelo WorkFlage bem como a estrutura de execução dos agentes.

Ao final deste capítulo, apresentamos a ferramenta de definição de processo, descrevemos as telas da ferramenta e algumas características da implementação.

Concluimos que, com o avanço tecnológico e a continuidade do trabalho em um futuro bem próximo, teremos uma base de perfis de atividade mais rica, o que nos possibilitará uma maior abrangência de modelagem.

No capítulo seguinte, apresentamos uma modelagem de um processo de compra de material como estudo de caso para demonstrar a utilização da ferramenta proposta.

8 ESTUDO DE CASO: processo de compra

Para demonstrar a aplicabilidade da ferramenta de definição de processo proposta, nós modelamos, como estudo de caso, um processo de compra de material para uma organização. Neste modelo, nós identificamos a presença de três papéis (funções) sendo; a função de Analista, a função de Comprador e a função do Almoxarife. Cada papel no processo compartilha o mesmo objetivo geral do processo que é adquirir o material solicitado pelo usuário com o menor custo, melhor qualidade e da forma mais rápida possível. O processo segue o seguinte fluxo normal de operação:

Um usuário que deseja comprar um certo material para a empresa, faz uma solicitação da compra deste material informando através de um formulário as suas necessidades e possibilidades de custo, a quantidade desejada, a descrição do produto, o modelo e o prazo de entrega previsto. Esta solicitação gera um pedido de Material (PM) e é enviada para um analista de compra que, após receber a aprovação da compra pela chefia imediata do usuário, faz uma avaliação geral do processo e prepara um dossiê identificando o número do processo, o código do produto dentro dos padrões da empresa, a situação do estoque para este produto, informações das últimas compras, etc. Após a análise feita pelo Analista, o dossiê é encaminhado ao comprador para efetuar a compra. O Comprador, por sua vez, relaciona todos os fornecedores possíveis, envia uma coleta de preço para cada um deles, recebe as propostas de fornecimento, solicita análise técnica do usuário e emite a ordem de compra. Após a efetivação da compra, o material a ser enviado para a companhia deve ser recebido no estoque pelo Almoxarife. O Almoxarife tem a função de receber o produto no almoxarifado, associar o material recebido com a sua ordem de compra, solicitar aceitação do usuário e atualizar o processo que será encerrado.

Algumas exceções podem ocorrer durante o fluxo normal do processo, assim, para uma total automação e otimização, necessitamos incorporar um certo grau de raciocínio para decidir qual a próxima tarefa a ser executada. Como exemplo podemos citar que: antes de emitir o pedido de compra, o Analista deve verificar se o material solicitado já existe no estoque para decidir se ele compra ou se faz uma solicitação de retirada do material do estoque. Se o material já existe no estoque e a sua quantidade for inferior à solicitação do usuário, o Analista deve tomar a decisão de comprar apenas a diferença entre a quantidade em estoque e a quantidade solicitada pelo usuário. Para cada compra de material, é criada uma nova instância do processo, que deve ter uma identificação única. O Diagrama abaixo representa o fluxo dos dados no processo de compra entre os agentes Analista e Comprador.

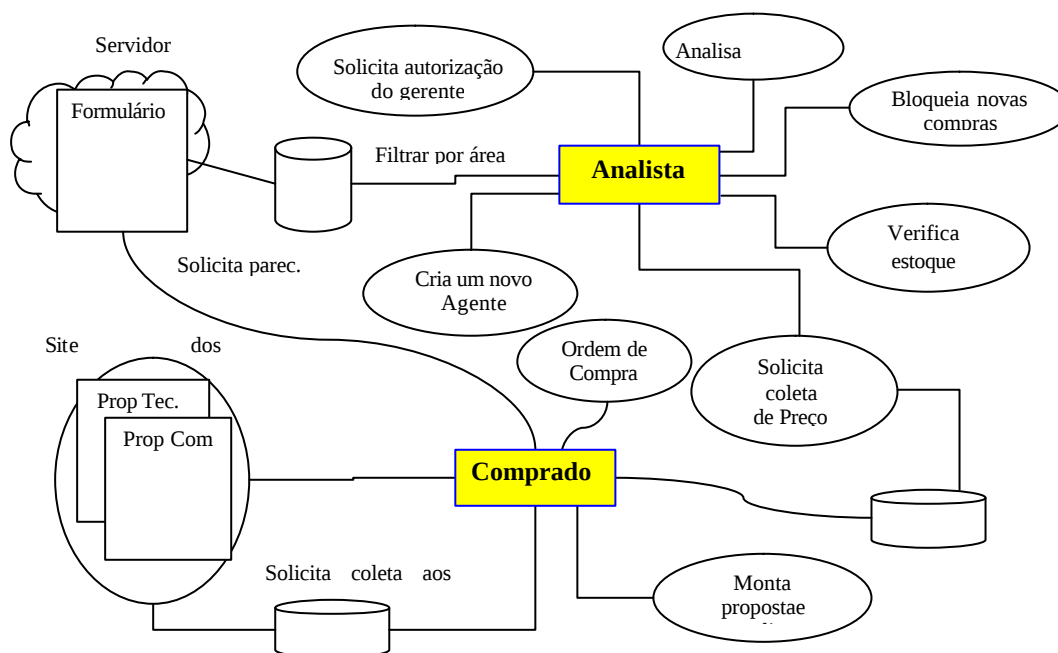


Figura 28 - Diagrama de fluxo de dados do processo de compra

8.1 Modelagem do processo no WorkFlage

Para cada papel identificado no processo, será criado um agente de software, portanto, teremos: o agente Analista, o agente Comprador e o agente Almoxarife. Para a representação interna do processo será criado um agente Processo com as seguintes configurações:

Agente Processo	
Atributos	Características
Nome	Processo de Compra de Materiais
Descrição	Aquisição de contribuições para uma companhia
Objetivo geral	Adquirir o material solicitado pelo usuário com: menor custo, de melhor qualidade e no período mais rápido possível.
Proprietário	Hugo Sampaio
Nº de papéis:	03

Tabela 5 - Atributos do agente processo

Algumas configurações como: ID, Endereço Local, serão adquiridas automaticamente pela ferramenta de definição do processo. Para cada papel desempenhado no processo, devem ser identificados os seus atributos, segundo o modelo de papéis. Na tabela seguinte, mostraremos os atributos do papel do agente Comprador:

Papel: Comprador	
Atributos	Características
Nome	Comprador
Descrição	Atributos para o agente comprador no processo de compra.

Objetivo	Comprar o produto solicitado pelo usuário, atendendo pré-requisitos do objetivo geral da instância do processo.
Qualificação	Incorporar um perfil de comprador
Relacionamento	Agente Analista, Agente Processo e Usuário
Obrigações	- Todo o conjunto de tarefas do perfil comprador - Solicitar parecer técnico do usuário - Solicitar coleta de preço - Visitar sites - Ler novas PM's - Conhecer todos os fornecedores - Conhecer todos os sites ativos

Tabela 6 - Atributos do agente comprador

Para este exemplo de modelagem na ferramenta WorkFlage, entendemos a necessidade de que alguns perfis pré-formatados de agentes já estejam cadastrados na base do WorkFlage. A elaboração de perfis não está sendo aprofundada neste trabalho, o que julgamos ser um ótimo tema para trabalhos futuros.

Durante a criação de uma nova instância, as necessidades do usuário serão transformadas em uma regra de produção, para que através da base de conhecimento do agente, o objetivo seja alcançado (Ex.: "Comprar LIVRO com preço<=R\$ 90,00").

Para cada papel desempenhado no processo, um agente atividade é criado e suas obrigações refletem um conjunto de tarefas a serem desempenhadas pelo agente. Cada tarefa criada na ferramenta gera um objeto tarefa que vai compor a lista de tarefas (agenda) do agente. Diferentemente dos workflows tradicionais, onde o fluxo de tarefas segue uma ordem prefixada, aqui o agente deve decidir qual a melhor tarefa a ser executada com base na análise

de uma série de condições do momento.

Para exemplificar a modelagem de uma tarefa, mostramos abaixo a modelagem da tarefa “ConsultarPreço” executada pelo agente comprador.

```
//=====
// Tarefa: ConsultarPreço
//      Esta tarefa possibilita ao agente consultar o
//      preço de um produto solicitado pelo cliente
//      através do objetivo.
//=====

synchronized private void ConsultarPreço(CAgente age)
{
    try
    {
        if(this.trabalhando)
        {
            this.AtualizarStatus(INICIAR_TAREFA, this);
            PAUSE(300);
            CEndereco SITE = new CEndereco(this.getUmSite());
            if(this.SiteTipo=="vendas")
            {
                SITE.SendMessage("solicitaPreco", this.MeuObjetivo);
                GetServiço(SITE, this.RegistrarMemoria("PRECO", SITE));
                this.AtualizarStatus("FIM TAREFA");
                this.RegistrarMemoria("VISITADO", SITE);
                this.VIAJAR(getNextSite());
            }
        }
        else
        {
            this.AtualizarStatus(INICIAR_TAREFA, this);
            PAUSE(300);
            this.IniciarTrabalho();
            this.ConsultarPreço(this);
            ...
        }
    }catch (Exception ex){}
}
```

Tabela 7 - Código da Tarefa “Consultar Menor Preço”

8.2 Telas do Processo de Compra

Um pedido de material pode ser executado instanciando a classe de processo de compra. Isto, pode ser feito através do menu Processo/Abrir, de onde será possível carregar o Processo desejado. A Figura 29 mostra a inicialização de um processo de compra.

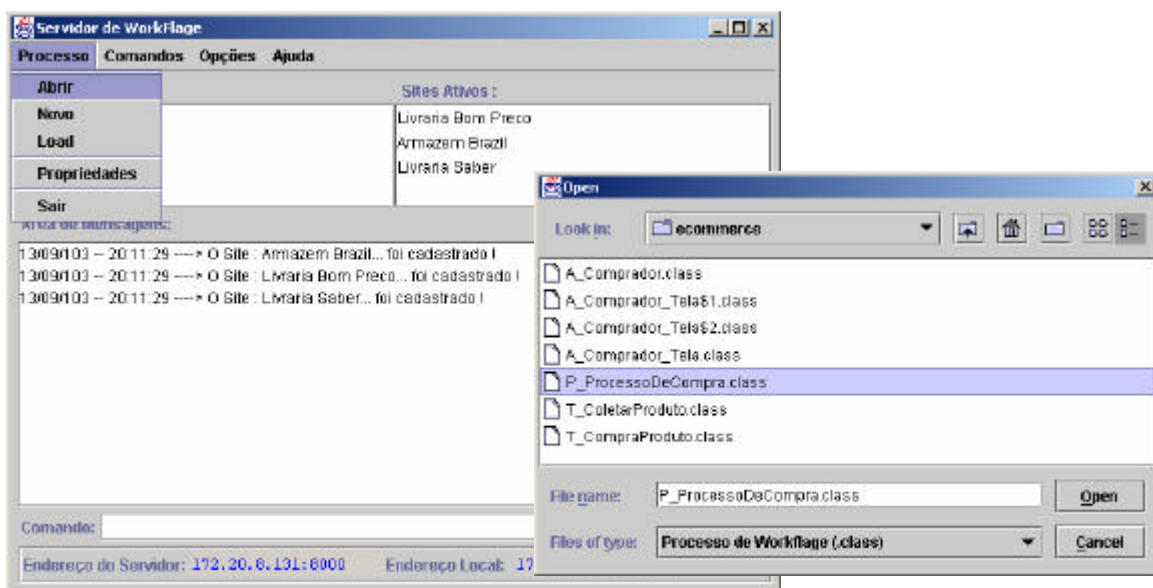


Figura 29 - Carregando o processo de compras

Após abrir um processo, o servidor de Workflage providencia a criação de todos os agentes atividades e dá início à execução das tarefas. Neste momento, o formulário para aquisição do produto será apresentado ao usuário. As informações do produto serão preenchidas nesta tela, que dará início ao processo de coleta de preço. A Figura 30 mostra a tela do agente “Comprador”, que será preenchida conforme a necessidade do usuário.

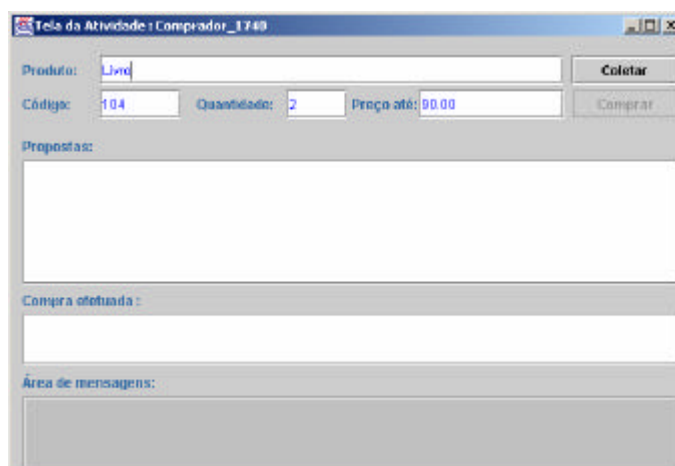


Figura 30 - Informação do agente comprador

Após o preenchimento das informações do produto desejado, o usuário solicita ao agente a coleta de preço. Antes porém, o agente analista verifica se a aquisição deste produto é favorável e repassa para o agente comprador para que este faça uma coleta de preço nos sites ativos no momento. Caso o produto já esteja no estoque, o agente analista pode parar o processo se a quantidade estocada for suficiente ou adquire apenas o suficiente para atender a solicitação do usuário. O Agente Comprador visita todos os sites ativos no momento e solicita uma coleta de preço para o produto solicitado. O Agente comprador recebe todas as coletas de preço e monta um proposta para ser avaliada pelo usuário, que deve solicitar ou não a compra pelo menor preço. Na figura 31 é mostrada a tela de coleta para o usuário.

Site	Produto	Código	Quant	Preço
Amazon Brasil	Livro	104	2	12.50
Livraria Bom Preço	Livro	104	2	10.50
Livraria Sabar	Livro	104	2	32.50

Figura 31 - Tela de resposta de coleta de preço

O Agente Comprador, neste caso, foi orientado à compra pelo menor preço do mercado, portanto o site “*Livraria Bom Preço*” será o vencedor da compra. Veja na figura 32 a compra efetuada:

Site	Produto	Código	Quant	Preço
Livraria Bom Preço	Livro	104	2	10.50

O Agente comprador, comprou o produto : Livro:104 Estocando no e-mail para o site http://12-20-8.131.8008/ para efetuar a compra

Figura 32 - Compra pelo menor preço

Após a escolha da melhor proposta, o agente comprador passa uma mensagem para o agente integrador, que se encarregará de enviar um e-mail para o site vencedor. Durante todo o processo de compra, o site será atualizado com as informações do processo, o que permite ao fornecedor saber como anda o processo de compra. A Figura 33 mostra uma tela de um site.

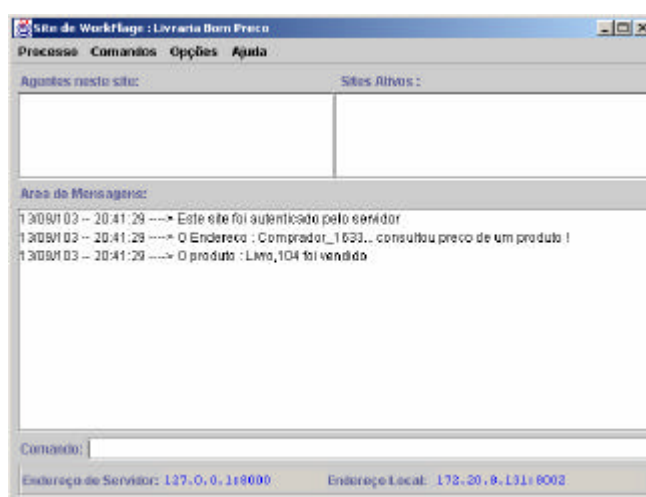


Figura 33 - Exemplo da tela de um Site

8.3 Conclusão

A utilização de agentes em Comércio Eletrônico permite a criação dos chamados mercados virtuais, onde os usuários delegam tarefas aos seus agentes que se encarregam de executá-las. Em vez de compradores e vendedores, podemos ter agentes de software compradores e agentes de software "vendedores", que se comunicam e negociam automaticamente entre si.

O sistema apresentado nesta seção obviamente necessita de um número maior de funcionalidades. Porém, este se propõe apenas a demonstrar como definir um processo de

negócio na ferramenta de definição de processo proposta neste trabalho.

Diante do exposto, podemos concluir que com o avanço das pesquisas na área da Inteligência Artificial e as inovadoras técnicas atuais de desenvolvimento de sistemas, é plenamente possível promover a interação humana com os chamados códigos inteligentes, a fim de prover ganhos consideráveis no controle do processo de negócios nas organizações.

Concluimos ainda que, quanto mais avançamos nas pesquisas, um novo leque de possibilidades para soluções de problemas, até então impossíveis, abre-se deixando-nos cada vez mais próximos do sonho de tornar a ficção uma realidade.

9 CONSIDERAÇÕES FINAIS

As Organizações estão readaptando e repensando a forma como estão fazendo os seus negócios. Para criação de novos modelos de negócios e implantação de nova forma de gerenciamento, a organização deve estar calçada em informações firmes e precisas para que não incorra em erros e prejuízo, mantendo-se sempre presente em um mercado cada vez mais agressivo e exigente.

Diante do acirrado mercado globalizado, as empresas estão buscando readaptar os seus processos de negócios visando a sua sobrevivência e permanência no mercado. Com o crescente progresso das tecnologias de informações e com o advento da Internet nos últimos anos, muitas são as ferramentas disponíveis para implementar melhorias neste sentido. As tecnologias de Workflow e Agentes de Software vem sendo estudadas por vários grupos de pesquisadores no mundo e alguns trabalhos já propõem a união das duas tecnologias para solucionar algumas limitações e agregar melhorias ao controle do fluxo de processo de negócio.

Desta forma, as empresas atuais tem investido muito em melhorias para o controle dos seus processos. Os Sistemas de Gerenciamento de Workflow têm ocupado uma boa posição no ranking das ferramentas de apoio ao gerenciamento. Os WfMS atuais algumas vezes são ineficientes devido às suas limitações.

A Tecnologia de Agente de software também desponta nos dias atuais como uma tecnologia revolucionária, que introduz os conceitos da Inteligência Artificial a pequenas unidades de software, tornando-as autônomas, independentes e inteligentes.

Nesta dissertação abordamos inicialmente as tecnologias dos Workflows e dos Agentes de Softwares, onde concluímos que algumas limitações encontradas nos Sistemas de Gerenciamento de Workflow, como: a falta de flexibilidade, necessidade de modificações

dinâmicas, redirecionamento do fluxo de trabalho e outras, são perfeitamente solucionáveis com a introdução do paradigma de Agentes de Software na tecnologia dos Workflow. Neste trabalho observamos que as atuais pesquisas nesta área apontam para um futuro bastante epleto de novidades, tais como a inclusão do Comércio eletrônico no âmbito dos sistemas de gerenciamentos de workflow.

Para o tratamento dos processos poucos estruturados, necessitamos incorporar alguns recursos que possibilitem o gerenciamento baseado no controle do tempo durante a execução do processo. O projeto WEICOT propõe um modelo de Sistema de Gerenciamento de Workflow que introduz os aspectos de temporalidade e cooperação entre os agentes. Este modelo foi adotado como base para o desenvolvimento de uma ferramenta e arquitetura que possibilitasse o gerenciamento dos processos pouco estruturados através dos agentes de software.

Este trabalho apresentou uma ferramenta de definição de processo que possibilita aproveitar as diversas características dos agentes de softwares para prover mobilidade e dinamismo aos WfMS. Durante a evolução deste trabalho, buscamos mostrar as fases de definição, execução e controle de um workflow baseado em agentes. A arquitetura e a ferramenta de definição proposta neste trabalho foi demonstrada em uma implementação prática no domínio do comércio eletrônico, onde mostramos o processo de compra de insumos para uma empresa.

Para este trabalho, implementamos a ferramenta de definição com o mínimo de funcionalidades, pois objetivávamos mostrar a possibilidade de criação de uma ferramenta mais próxima da realidade empresarial. Visualizamos que muito ainda deve ser feito para que esta ferramenta se torne uma ferramenta apta ao propósito comercial. Como perspectivas futuras deste trabalho podemos citar:

- ? A definição e desenvolvimento de uma base de perfis para compor o *kernell* da ferramenta de definição do processo;
- ? O acréscimo de funcionalidades para melhor se adaptar às reais necessidades das organizações;
- ? Implementação do mecanismo de acompanhamento de processo via web;
- ? Introdução da interface 4 do modelo de referência da WfMC, possibilitando a interoperabilidade da arquitetura do WorkFlage com outras ferramentas de gerenciamento de workflow já existente.

Portanto, podemos concluir que ainda há muito o que fazer na área de workflow baseado em agentes e que muito ainda pode ser feito, não somente no que se refere à consolidação da tecnologia, mas também na área de desenvolvimento de sistema. Isso certamente irá acarretar mudanças contínuas nos esquemas de classificação, plataformas de operação, modos de execução, e na tecnologia de desenvolvimento de sistemas de gerenciamento de workflow como um todo, aproximando cada vez mais a tecnologia das necessidades da humanidade.

REFERÊNCIAS

- ALTY J. L., D. GRIFFITHS, N.R. JENNINGS, et al “**ADEPT-Advanced Decision Environment for Process Tasks**: Overview and Architecture”, proceedings of the BSC Expert System 94 Conference, Cambridge, UK, 1994, pp 359-371.
- ANDERSON, M. Workflow Interoperability – Enabling E-Commerce. WPMC whitepaper, 1999.
- BALDI, M., GAI, S. and PICCO, G. P. “*Exploiting code mobility in decentralized and flexible network management*”. In Proceedings of the 1st International Workshop on Mobile Agents, Berlin, Germany. April 1997.
- BALDI, M., PICCO, G.P. *Evaluating the tradeoffs of mobile code design paradigms in network management applications*, ICES98. 1998.
- BIRREL, A. and NELSON, B. *Implementing remote procedure calls*, in ACM Transactions on Computer Systems 2,1, pp. 39-59. 1984.
- BREWINGTON, B., et al *Mobile agents in distributed information retrieval*. Chapter 15, pages 355-395 in “intelligent Information Agents” edited by Mathias Klusch. Springer Verlag. 1999.
- BUSSLER, C., JABLONSKI, S. Workflow Management: Modeling Concepts, Architecture and Implementation. London: Internacional Thomson Computer Press, 1996.
- CAI T., GLOOR P. A. and NOG S. DartFlow: A Workflow Management System on the Web Using Transportable Agents, PCS-TR96-283, Dartmouth College, 1996.
- CASATI, F.; CERI, S.; PERNICI, B.; POZZI, G. Conceptual Modeling of Workflow. Proceedings of OO-ER Conference. Gold Coast, Australia, 1995.
- CASTILLO A., KAWAGUCHI, M., PACIOREK, N. and WONG, D. *Concordia as enabling technology for cooperative information gathering*. In Japanese Society for Artificial Intelligence Conference. June 1998.
- CHANG J. W., SCOTT C. T. **Agent-based Workflow**. TRP Support Environment (TSE), 5th International World Wide Web Conference, 1996.
- COHEN, P. R., LEVESQUE, H.J. *Rational interaction as the basis for communication*. In Cohen, P. R., Morgan, J., and Pollack, M.E., editors, Intentions in Communication, pages 221-256. The MIT Press: Cambridge, MA. 1990.

DIKAIKOS, M.D., GUNOPOULOS, D. **FIGI**: *the architecture of an internet –based financial information gathering infrastructure*. In Proceedings of the International Workshop on Advanced Issues of E-Commerce and Web-based Information Systems. IEEE- Computer Society. April 1999.

FIGUEIRA F. C., RAMALHO, G. **Jeops** : The Java Embedded Object Production System. In M. Monard e J. Sichman (eds.). LNAI Series, vol. 1952, pp 52-61. London: Springer-Verlag, 2000.

FININ T., LABROU Y., MAYFIELD J. *KQML as an agent communication language*, Bradshaw J. ed., Software agents, MIT Press, Cambridge. 1997.

FIPA - Foundation for Intelligent Physical Agents, "Agent communication language. Technical report". FIPA97, Parts 103 (Normative Standards), October 1997, (<http://www.fipa.org/repository/index.html>).

GEORGAKOPOULOS, D., HOMICK, M. An overview of Workflow Management: from Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel Databases, 3. pp. 119-153. Kluwer Academic Publishers. Boston. 1995.

GILBERT, D., MANNY, A., BETTY, A. "White paper on Intelligent Agents", IBM Corporation, Research Triangle Park, NC, 1996.

GOLDBERG, A., ROBSON, D. *Smalltalk 80: The language and its implementation*, AddisonWesley, Reading, MA. 1983.

GOSLING J., JOY B., STEELE G. *The java language specification*, Addison Wesley. 1997.

GRAY, R.S. **AgentTcl**: A Transportable Agent System, Proceedings of the CIKM Workshop on Intelligent Information Agents, 4th International Conference on Information and Knowledge Management(CIKM), December 1995.

GRAY, R.S. et al *D'Agents: security in multiple-language, mobile-agent system*. In G. Vigna, editor, Mobile agents and security, volume 1419 of Lecture Notes in Computer Science, pages 154--187. Springer-Verlag, New York, NY. 1998.

GRAY, R.S., et al. **Mobile agents: motivations and state of art**. Technical Report TR2000-365, Dept. of Computer Science, Dartmouth College. 2000.

HAWRYSZKIEWYCZ I., DEBENHAM J. A workflow System Based on Agents , Database and Expert Systems Applications, 9th International Conference, DEXA'98 Vienna, Austria, August 24-28, 1998 Proceedings, pp 135-144.

HAYZELDEN, A., BIGHAM, J. *Agent technology in communications systems: an overview*. Knowledge Engineering Review Journal, Vol.14:3, pp. 1-35. 1999.

JENNINGS N. R., et al. Agent-based Business Process Management, International Journal of Cooperative Information Systems, 5(2&3), 1996, pp 105-130.

JENNINGS, N.R., WOOLDRIDGE, M. *Agent-oriented software engineering*. In Handbook of Agent Technology (ed. J. Bradshaw) AAAI/MIT Press. 2000.

JENSEN, K. *Colored Petri Nets: Basic concepts, Analysis Methods and Practical Use*. Vol 1. Springer Verlag, Berlin 1992.

JOOSTEN, S. Trigger Modelling do Workflow Analysis. Proceedings Com'94: Workflow Management, Challenges, Paradigms and Products. Viena, Austria, Out. 1994.

KARJOTH G., LANGE B.D., OSHIMA M. *A security model for aglets*. IEEE Internet Computing, , pp 68-77. July - August 1997.

KARNIK, N. *Security in mobile agent systems*, PhD dissertation. Computer Science and Engineering University of Minnesota. 1998.

KAY, J., et al *At postmaster: a system for agent collaboration and information dissemination*. In Proceedings of the 2nd International Conference on Autonomous Agents, pages 338--342, Minneapolis, MN, ACM. 1998.

KHOSHAFIAN, S., BUCKIEWICZ, M. Introduction to Groupware, Workflow, and Workgroup Computing. U.S.A: John Wiley & Sons, Inc, 1995.

KOBIELUS, J. G. Workflow Strategies; IDG Books – Foster City – CA – 1997.

LABIDI, S., HAMMOUDI, S., Cooperative Workflow Management in WEICOT. In: International Conference on the Design Of Cooperative System, (COOP'98)., 1998, Cannes. Proceedings of the third International Conference on the Design of Cooperative System, (COOP'98). 1998.

LABIDI, S., HAMMOUDI, S., GANNOUN, L., Integrating Cooperation and Temporal Organization in Workflow Management. In: The 2000 Inter Nternational Conference on Artificial Intelligence (ICI'2000)., 2000, Las Vegas. The 2000 International Conference on Artificial Intelligence (IC-AI'2000).. Las vegas.: World Scientific Engineering Society, SREA Press, 2000. p.229-235.

LABIDI, S., HAMMOUDI, S., WEICOT: A Distributed Workflow Environment. In: Spanish Symposium of Distributed Computation (SEID'99)., 1999, Santiago de Compostela. Proceedings of the Spanish Symposium of Distributed Computation (SEID'99). 1999.

LABIDI, S.. Cooperative Work Modeling. In: Third International Conference on Concurrent Engineering Research and Pplication, CE96., 1996, Toronto. Third International Conference on Concurrent Engineering Research and Application, CE96. 1996.

- LANGE, D.B., CHANG D.T. *IBM Aglets Workbench: a White Paper*, IBM Corporation, September 1996, <http://www.trl.ibm.co.jp/aglets>.
- LANGE, D.B., OSHIMA, M. *Seven good reasons for mobile agents*. Communications of ACM, 42 (3) :88-89. March 1999.
- LANGE, D.B., OSHIMA, M. *Programming and deploying java mobile agents with aglets*. Addison-Wesley. 1998, pp 2-5.
- LEI Y., BEAT F. Schimd, "A Conceptual Framework for Agent Oriented and Role Based Workflow Modeling", Institute for Media and Communications management, Univesity os St. Gallen, Switzerland, 1999.
- MERZ, M. et al. Using Mobile Agents to support Interorganizational Workflow Management. Distributed Systems Group - Computer Science Dept. - Hamburg University 551-572. 1997.
- NOBLE, B. D., SATYANARAYANAN, M. *Experience with adaptive, mobile Applications in Odyssey*, Mobile Networks and Apps. 1999.
- NWANA, H. et al *ZEUS: a toolkit for building distributed multi-agent systems*. In Applied Artificial Intelligence Journal, Vol 13 (1) , pp. 129-186. 1999.
- NWANA, H.S. **Software agents**: an overview", Knowledge Engineering Review, 11, No 3, Sept. 1996, pp 1-40
- O'BRIEN ,P. D., WIEGAND, W. E. Agents of Change in Business Process Management", BT Technical Journal, Intelligent Software Systems Special Issue, Oc-tober 1996.
- ODELL, J. *Agents and objects: how do they differ?*, working paper v2.2. September, 1999.
- ODGERS B. R., et al. Technologies for Intelligent Workflows: Experiences and Lessons, Proceedings of Agent-Based Systems in the Business Context, AAAI 1999 Workshop, July 1999.
- ODGERS B. R., SHEPHERDSON J. W., THOMPSON S. G. Distributed Workflow Coordination by Proactive Software Agents, Proceedings of Intelligent Work-flow and Process Management, IJCAI-99 Workshop, August 1999.
- OMG, **MAFIS**: Mobile Agent Facility Specification, Object Management Group, 00-01-02.pdf (<ftp://ftp.omg.org/pub/docs/formal/00-01-02.pdf>).1998
- OMG, The Object Management Group, *The mobile agent system interoperability facility*, OMG TC Document orbos/Jan/2000. URL: <http://www.omg.org.2000>
- OUSTERHOUT, J. *Tcl: an embeddable command language*, in Usenix Conference Proceedings, pp. 133-146, Winter. 1990.

PULIAFITO, A., TOMARCHIO, O. *Using mobile agents to implement flexible network management strategies*. Computer Communication Journal, 23;8 :708—719. April 2000.

QI, H., IYENGAR, S. S., CHAKRABARTY, K. *Multi-resolution data integration using mobile agents in distributed sensor networks*, Submitted to *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews*. October, 2000.

QIMING C., et al. *Multi-Agent Cooperation, Dynamic Workflow and XML for E-Commerce Automation*, HP labs, California, USA, 1999.

REISER, H., VOGT, G. *Security requirements for management systems using mobile agents*. Proceedings of the Fifth IEEE Symposium on Computers / & Communications. Pages 160-165. Jul 2000.

ROSENCHEIN, J. S., GENESERETH, M. R. *Deals among rational agents*. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI – 85) , pages 91-99, Los Angeles, CA.1985.

RUSSEL, S. J., NORVIG, P., *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey .1995.

SAMARAS, G., et al. *Mobile agent platforms for web databases: a qualitative and quantitative assessment*, Proceedings of First International Symposium on Agent Systems and Applications and Third International Symposium on Mobile Agents (ASA/MA 99) , Palm Springs, California. October 1999.

SHEPHERDSON, J. W, THOMPSON S.G., ODGERS B R, *Decentralised workflows and software agents* ,1995.

STAMOS, J., GIFFORD, D. K. *Remote evaluation*, ACM Transactions on Programming Languages and Systems 12,4, pp. 537-565. 1990.

STRAFIER, M., BAUMANN, J., HOHL, F. *Mole - A java based mobile agent system*. In Proceedings of the ECOOP'96 workshop on Mobile Object Systems. 1996.

THEILMANN, W., ROTHERMEL K. *Maintaining specialized search engines through mobile filter agents*. Proc. 3rd Int. Workshop on Cooperative Informative Agents (CIA'99) , Uppsala, Sweden, M. Klusch, O. Shehory, G. Wei.,1999

TRYLLIAN, A. Whitepaper. *Mobile agents: the work force of the new millennium*. November, 1999.

VALENTE, M. T et al. *Linguagens para computação móvel na internet*. UFPE, Brasil, SBLP. 2000.

WEISS G., WOOLDRIDGE M., Intelligent agents. In G. Weiss, editor, *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.

WELCH, B. B. *Practical programming in Tcl and Tk*. Prentice-Hall, Upper Saddle River, New Jersey. 1995.

WfMC: "Workflow Management Coalition Terminology & Glossary", Document Number WfMC-TC-1011, Document Status - Issue 3.0, Feb 1999.

WHITE, J. *Mobile agents* In Software Agents, Bradshaw, J. Ed., MIT Press. 1997.

WHITE, T., Bieszczad, A. and Pagurek. B. *Distributed fault location in networks using mobile agents*. In Proceedings of the 3rd International Workshop on Agents in Telecommunication Applications IATA'98, Paris, France. July 1998.

WHITE, T., PAGUREK, B., OPPACHER, F. *Connection management using adaptive mobile agents*. In H.R. Arabnia, editor, Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'98) , pages 802--809. CSREA Press. 1998.

WOOLDRIDGE, M., JENNINGS, N. *Intelligent agents: theory and practice* . The Knowledge Engineering Review. 1995.

ZAPF, M., MUELLER, H., GEIHS, K. *Security requirements for mobile agents in electronic markets*. Lecture Notes in Computer Science, 1402:205-217. 1998.

Fialho, Hugo Sampaio

WorkFlage: Uma Ferramenta de definição e criação de Workflow baseado em Agentes de Software para WEICOT / Hugo Sampaio Fialho. – São Luís, 2003

156 f.: il.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal do Maranhão, 2003.

1. Engenharia de Sistemas 2. Workflow. 3. WfMS 4. Agentes de Software. I. Título

CDU 004.414.2