



**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA MECÂNICA**

**MODELO DE SIMULAÇÃO BASEADO AGENTES PARA O ESTUDO DA  
INFLUÊNCIA DE PLANOS DE PROCESSOS ALTERNATIVOS NA  
PROGRAMAÇÃO DA PRODUÇÃO EM SISTEMAS DE MANUFATURA COM  
LAYOUT FUNCIONAL**

**Tese submetida à  
UNIVERSIDADE FEDERAL DE SANTA CATARINA  
Para a obtenção do grau de  
DOUTOR EM ENGENHARIA MECÂNICA**

**IZABEL CRISTINA ZATTAR**

**Florianópolis, SC**

**2008**

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA MECÂNICA

MODELO DE SIMULAÇÃO BASEADO AGENTES PARA O ESTUDO DA  
INFLUÊNCIA DE PLANOS DE PROCESSOS ALTERNATIVOS NA PROGRAMAÇÃO  
DA PRODUÇÃO EM SISTEMAS DE MANUFATURA COM LAYOUT FUNCIONAL

IZABEL CRISTINA ZATTAR

Esta tese foi julgada adequada para a obtenção do título de

DOUTOR EM ENGENHARIA  
ESPECIALIDADE ENGENHARIA MECÂNICA

sendo aprovada em sua forma final.

---

João Carlos E. Ferreira, PhD (Orientador/Presidente da banca)

---

Eduardo Alberto Fancello, D.Sc. (Coordenador de curso)

Banca Examinadora

---

Guilherme Ernani Vieira, Dr. (PUC/PR /Relator)

---

Nílson Luiz Maziero, Dr. Eng. (UPF)

---

Paulo Jose de Freitas Filho, Dr. Eng.

---

Fernando Antônio Forcellini, Dr. Eng.

*Foi o melhor dos tempos, foi o pior dos tempos,  
foi a idade da sabedoria, foi a idade da imbecilidade, foi a época de acreditar,  
foi a época da descrença,  
foi a época da luz, foi a época da escuridão (...)*

Charles Dickens (História de duas cidades)

## DEDICATÓRIA

*Aos meus pais,  
Que em seu sono eterno  
velem por meus sonhos.  
Que em minhas realizações  
eu honre vosso nome.*

*Ao meu amado José Luiz,  
a melhor parte de mim.*

## AGRADECIMENTOS

Ao encerrar não somente um trabalho, mas sim um período de vida, o qual consumiu alguns anos e quilômetros rodados, sempre faltam palavras para descrever de forma justa o que passamos, sentimos e aprendemos.

As minhas amigas Patrícia, Elsa, Adriana e Alexandra, pela companhia, café, jantares, passeios no shopping e principalmente pela paciência e amizade, a qual somente nós, do clube da Luluzinha sabemos a importância.

Aos amigos, grandes amigos, Mikos e Julio, fiéis companheiros de laboratório, almoço, café, viagens... sem a amizade e o suporte de ambos, eu não teria conseguido. Vocês me ensinaram a importância do trabalho em equipe, bem como a importância da fé.

À Gláucia e Marcello, simplesmente meus amigos de absolutamente todas as horas, cujo amor que sinto por ambos não pode ser descrito em palavras.

Ao Sacchelli e Marcelo Teixeira, amigos e incentivadores.

Aos bolsistas de iniciação científica do laboratório GRIMA / UFSC: pelo empenho e suporte técnico.

Ao POSMEC e CAPES, que me forneceram o suporte técnico e financeiro para a conclusão desta tese.

E finalmente ao meu mentor, amigo, orientador e algumas vezes psicólogo, João Carlos E. Ferreira, sem o qual eu simplesmente teria abandonado tudo há muito tempo. Sua paciência e respeito sempre foram um caminho seguro por onde tive o prazer de trilhar durante estes anos todos.

## SUMÁRIO

<b>LISTA DE FIGURAS.....</b>	<b>10</b>
<b>LISTA DE TABELAS.....</b>	<b>13</b>
<b>RESUMO.....</b>	<b>16</b>
<b>ABSTRACT .....</b>	<b>17</b>
<b>1 INTRODUÇÃO .....</b>	<b>18</b>
1.1 Origem do trabalho .....	18
1.2 Problema de pesquisa.....	19
1.3 Objetivos do trabalho .....	21
1.3.1 Objetivo Geral.....	21
1.3.2 Objetivos Específicos.....	21
1.4 Justificativa .....	22
1.4.1 Ineditismo.....	22
1.5 Metodologia .....	22
1.5.1 Quanto à sua natureza .....	22
1.5.2 Quanto à forma de abordagem do problema.....	23
1.5.3 Quanto aos objetivos.....	23
1.5.4 Quanto aos seus procedimentos técnicos.....	23
1.6 Delimitação do trabalho .....	24
1.7 Organização da tese .....	25
<b>2 CONCEITUALIZAÇÃO: SISTEMAS BASEADOS EM MULTIAGENTES.....</b>	<b>29</b>
2.1 Introdução .....	29
2.2 Definições gerais.....	29
2.3 Definição e características dos sistemas baseados em agentes .....	30
2.3.1 O que é um agente?.....	30
2.3.2 O que são sistemas multiagentes?.....	34
2.3.3 Motivação para o uso de sistemas multiagentes.....	35
2.3.4 Arquiteturas de sistemas baseados em agentes .....	36
2.3.5 Coordenação e cooperação.....	42
2.3.6 Redes de contrato - <i>Contract-Net Protocol</i> .....	44
2.3.7 Padronização .....	47
2.3.8 Linguagens de comunicação entre agentes .....	48
2.3.9 Frameworks para o desenvolvimento de sistemas baseados em agentes....	52

2.3.10	JADE – Java Agent DEvelopment framework.....	53
<b>3</b>	<b>CICLO DE VIDA PARA O DESENVOLVIMENTO DE UM MODELO DE SIMULAÇÃO BASEADO EM MULTIAGENTES.....</b>	<b>59</b>
3.1	Ciclo de vida para o desenvolvimento de um modelo de simulação .....	59
3.2	Modelo conceitual de simulação.....	61
3.3	Modelos de simulação baseados em agentes .....	63
3.4	Metodologias para o desenvolvimento de sociedades baseadas em agentes .....	65
3.5	Introdução à metodologia PASSI.....	66
3.6	Modelo de requisitos do sistema.....	67
3.7	Modelo de sociedade de agentes.....	69
3.8	Modelo de implementação de agentes .....	70
3.9	Modelo de código.....	71
3.10	Modelo de desenvolvimento .....	73
3.11	Alterações na metodologia proposta .....	73
3.12	O conceito de validação e verificação.....	74
3.13	O conceito de validação e verificação adotado no modelo proposto.....	76
3.14	Conclusão do capítulo .....	77
<b>4</b>	<b>MODELO CONCEITUAL.....</b>	<b>18</b>
4.1	Modelo de requisitos do sistema - Sub-fase de descrição do domínio .....	79
4.1.1	Declaração do problema.....	80
4.1.2	Caracterização do ambiente modelado.....	81
4.1.3	Métricas para a análise dos resultados obtidos .....	83
4.1.4	Definição de peça e recurso produtivo no modelo de simulação proposto.....	85
4.1.5	Representação dos planos de processo no modelo de simulação proposto.....	88
4.1.6	Definição de flexibilidade no modelo de simulação proposto .....	89
4.1.7	Flexibilidade de operações .....	91
4.1.8	Flexibilidade de seqüência .....	93
4.1.9	Flexibilidade de processo .....	95
4.1.10	Flexibilidade de máquina .....	97
4.1.11	Requisitos do domínio.....	101
4.2	Conclusão do capítulo .....	106
<b>5</b>	<b>ARQUITETURA E IMPLEMENTAÇÃO DO MODELO.....</b>	<b>108</b>
5.1	Modelo de requisitos do sistema - Sub-fase de identificação dos agentes.....	108
5.2	Modelo de requisitos do sistema - Sub-fase de identificação dos papéis .....	112

5.3	Modelo de sociedade de agentes - Sub-fase de Descrição da ontologia de comunicações..	115
5.4	Modelo de sociedade de agentes - Sub-fase de descrição dos papéis	119
5.5	Modelo de sociedade de agentes - Sub-fase de descrição dos protocolos	122
5.5.1	Protocolo de negociação	123
5.5.2	Protocolo de renegociação	126
5.6	Modelo de implementação de agentes - Sub-fase de definição da estrutura multiagente	127
5.7	Modelo de implementação de agentes - Sub-fase de definição da estrutura por agente	130
5.8	Modelo de implementação de agentes - Sub-fase de definição do comportamento por agente	132
5.8.1	Agente <i>Server</i>	133
5.8.2	Agente <i>JobOrder</i>	136
5.8.3	Agente <i>JobOrder</i> – Comportamento auxiliar <i>NegotiationTimeControl</i>	141
5.8.4	Agente <i>JobOrder</i> – Comportamento auxiliar <i>WaitToNegotiate</i>	142
5.8.5	Agente <i>Resource</i>	143
5.8.6	Agente <i>Resource</i> – comportamento <i>Negotiation</i>	145
5.9	Conclusão do capítulo	148
<b>6</b>	<b>FORMULAÇÃO DA PROPOSTA DE NEGOCIAÇÃO E DESCRIÇÃO DO CASO DE TESTE.....</b>	<b>148</b>
6.1	Formulação da proposta de negociação	149
6.1.1	Duração do período de negociação	150
6.1.2	Agrupamento de operações	152
6.1.3	Cálculo do tempo de proposta e a formação da parcela de tempos internos ao sistema	154
6.1.4	A formação da parcela de tempos de máquina	157
6.2	Caso de teste para avaliação do trabalho proposto	160
6.2.1	Representação dos níveis de flexibilidade no modelo desenvolvido	162
6.2.2	Representação dos níveis de flexibilidade de máquina	167
6.3	Conclusão do capítulo	167
<b>7</b>	<b>PARÂMETROS DE SIMULAÇÃO E ANÁLISE DE RESULTADOS.....</b>	<b>169</b>
7.1	Parâmetros de simulação	169
7.2	Abordagens utilizadas para validação do modelo	174
7.3	Análise dos resultados – <i>makespan</i>	176

7.3.1	<i>Makespan</i> – flexibilidades de operação, seqüência, processos e máquina	176
7.3.2	<i>Makespan</i> – flexibilidades de operação, seqüência e máquina	180
7.3.3	Conclusões finais a respeito da variável de resposta <i>Makespan</i>	185
7.4	Análise dos resultados – <i>flow time</i>	190
7.4.1	<i>Flow time</i> – flexibilidades de operação, seqüência, processos e máquina	190
7.4.2	<i>Flow time</i> – flexibilidades de operação, seqüência e máquina	193
7.4.3	Conclusões finais a respeito da variável de resposta <i>flow time</i>	195
7.5	Análise dos resultados para as variáveis de resposta <i>makespan</i> e <i>flow time</i> utilizando seqüências dependentes do setup	200
7.6	Conclusão do capítulo	208
<b>8</b>	<b>O USO DE SISTEMAS ESPECIALISTAS ASSOCIADO À AGENTES</b>	<b>216</b>
8.1	Estrutura de dados	216
8.2	O uso de sistemas especialistas combinados a agentes	222
8.3	Implementação	224
8.3.1	Métodos <i>ask_for_Expert</i> e <i>waiting_for_Expert</i>	226
8.3.2	O agente <i>Expert</i> e sua interface com o JESS - Java Expert System Shell	229
8.4	Resultados encontrados e conclusão do capítulo	231
<b>9</b>	<b>CONCLUSÕES E RECOMENDAÇÕES</b>	<b>234</b>
9.1	Discussão	234
9.1.1	Desenvolvimento do protótipo de <i>software</i> de simulação	235
9.2	Conclusões	236
9.2.1	Conclusões sobre os grupos de simulação A, B, C e D em relação à variável de resposta <i>makespan</i> para tempos de <i>setup</i> iguais a zero	237
9.2.2	Conclusões sobre os grupos de simulação A, B, C e D em relação à variável de resposta <i>flow time</i> para tempos de <i>setup</i> iguais a zero	238
9.2.3	Conclusões sobre os grupos de simulação A, B, C e D em relação às variáveis de resposta <i>makespan</i> para tempos de <i>setup</i> diferentes de zero	238
9.2.4	Conclusões sobre os grupos de simulação A, B, C e D em relação às variáveis de resposta <i>flow time</i> para tempos de <i>setup</i> diferentes de zero	239
9.3	Contribuições	239
9.4	Recomendações para trabalhos futuros	242
<b>10</b>	<b>REFERÊNCIAS</b>	<b>243</b>

## LISTA DE FIGURAS

Figura 1.1 – Estrutura da tese .....	26
Figura 2.1 – Taxonomia de agentes.....	33
Figura 2.2 – Arquitetura hierárquica .....	37
Figura 2.3 – Arquitetura federada com diferentes abordagens: (a) facilitadores, (b) <i>brokers</i> e (c) <i>matchmakers</i> Fonte: SHEN, 2002.....	40
Figura 2.4 – Arquitetura heterárquica.....	41
Figura 2.5 – Taxonomia da coordenação .....	43
Figura 2.6 – Etapas de negociação em uma rede de contrato.....	44
Figura 2.7 – Etapas de negociação em uma rede de contrato.....	45
Figura 2.8 – Arquitetura distribuída da plataforma .....	54
Figura 2.9 – Arquitetura distribuída da plataforma JADE com múltiplos containeres.....	55
Figura 2.10 – Interoperabilidade na plataforma .....	56
Figura 2.11 – Interface inicial.....	57
Figura 2.12 – Arquitetura interna genérica de um agente em JADE.....	58
Figura 3.1 – Ciclo de vida para o desenvolvimento de um modelo de simulação .....	60
Figura 3.2 – Modelo conceitual de simulação.....	61
Figura 3.3 – Passos para o desenvolvimento do modelo conceitual .....	62
Figura 3.4 – Modelos que compõem a metodologia PASSI.....	67
Figura 3.5 – Meta-modelo representando a fase de Requisitos de Sistema da metodologia PASSI.....	68
Figura 3.6 – Modelo de sociedade de agentes descrito em função da documentação e diagramas gerados em suas sub-fases.....	69
Figura 3.7 – Modelo de implementação de agentes descrito em função da documentação e diagramas gerados em suas sub-fases.....	70
Figura 3.8 – Modelo de código descrito em função da documentação e diagramas gerados em suas sub-fases .....	71
Figura 3.9 – Modelo de desenvolvimento .....	73
Figura 4.1 – Sub-fase de descrição do domínio .....	80
Figura 4.2 – Representação da estrutura de um pedido no modelo de simulação proposto.....	87
Figura 4.3 – Representação de um plano de processo no modelo proposto.....	88
Figura 4.4 – Classificação das flexibilidades utilizadas no modelo apresentado neste trabalho.....	90

Figura 4.5 – Exemplo de plano de processo com o número de máquinas alternativas por operação.....	93
Figura 4.6 – Exemplo de uma peça com seus planos de processo alternativos.....	96
Figura 4.7 – Exemplos de planos de processo que permitem o cálculo de flexibilidade de máquina .....	99
Figura 4.8 – Operações nas peças 1 e 2 passíveis de agrupamento, sem <i>setup</i> adicional .....	100
Figura 5.1 – Sub-fase de identificação dos agentes.....	109
Figura 5.2 – Diagrama de identificação dos agentes .....	110
Figura 5.3 – Sub-fase de identificação dos papéis .....	112
Figura 5.4 – Diagrama de identificação de papéis.....	113
Figura 5.5 – Sub-fase de descrição da ontologia de comunicações.....	116
Figura 5.6 – Diagrama de descrição da ontologia de comunicação .....	117
Figura 5.7 – Sub-fase de descrição de papéis.....	119
Figura 5.8 – Diagrama de descrição dos papéis .....	121
Figura 5.9 – Envio do pedido de propostas (CFP) e suas possíveis respostas .....	123
Figura 5.10 – Fechamento de contrato ou rejeição da proposta .....	124
Figura 5.11 – A ordem se encaminha para a fila do recurso .....	125
Figura 5.12 – Cancelamento de uma ordem.....	126
Figura 5.13 – Sub-fase de definição da estrutura multiagente .....	128
Figura 5.14 – Diagrama de definição da estrutura multiagente.....	129
Figura 5.15 – Sub-fase de definição da estrutura por agente individual .....	131
Figura 5.16 – Diagrama de definição da estrutura do agente <i>Resource</i> .....	132
Figura 5.17– Sub-fase de de definição do comportamento por agente .....	133
Figura 5.18 – Descrição do comportamento principal do agente <i>Server</i> durante seu ciclo de vida.....	135
Figura 5.19 – Descrição do comportamento <i>Negotiation</i> adotado pelo agente <i>JobOrder</i> durante seu ciclo de vida .....	138
Figura 5.20 – Descrição simplificada dos estados compostos do agente <i>JobOrder</i> durante o ciclo de vida do comportamento <i>Negotiation</i> .....	140
Figura 5.21 – Descrição do comportamento auxiliar <i>NegotiationTimeControl</i> do agente <i>JobOrder</i> .....	141
Figura 5.22 – Descrição do comportamento auxiliar <i>WaitToNegotiate</i> do agente <i>JobOrder</i> .....	142
Figura 5.23 – Descrição do comportamento <i>HandleQueue</i> do agente <i>Resource</i> durante seu ciclo de vida.....	144

Figura 5.24 – Descrição do comportamento <i>Negotiation</i> adotado pelo agente <i>Resource</i> durante seu ciclo de vida .....	147
Figura 6.1 – Período de negociação, baseado em USHER (2003) .....	150
Figura 6.2 – (a) Negociação sem agrupamento de operações; (b) Negociação com agrupamento de operações .....	153
Figura 6.3 – Tempos internos ao sistema .....	156
Figura 6.4 – Tempo de proposta nas abordagens sem agrupamento x com agrupamento de operações .....	159
Figura 6.4 – Plano de processo com alternativas para a peça 10 do caso de testes .....	164
Figura 7.1 – Diferentes tipos de flexibilidades permitidas pelo modelo proposto .....	170
Figura 7.2 – Diferentes agendamentos para um mesmo grupo de simulação .....	171
Figura 7.3 – Classe que permite a configuração dos parâmetros de simulação do modelo proposto .....	172
Figura 7.4 – <i>Makespan</i> dos grupos formados por 6 peças .....	189
Figura 7.5 – <i>Makespan</i> dos grupos formados por 9 peças .....	189
Figura 7.12 – Número de trabalhos pelo tempo em um problema envolvendo chegadas simultâneas (CONWAY, 1967) .....	208
Figura 8.1 – Plano de processo da peça 9 com suas precedências e alternativas .....	218
Figura 8.2 – Estrutura de dados desenvolvida para o modelo proposto .....	221
Figura 8.3 – Integração entre o <i>framework</i> JADE e o JESS (RUSSEL e NORVIG, 1995) ..	223
Figura 8.4 – Estrutura de dados desenvolvida para o modelo proposto .....	225
Figura 8.5 – Sugestões de roteamento obtidas a partir da variável <i>makespan</i> .....	228
Figura 8.6 – Interface entre os agentes e o JESS .....	230

## LISTA DE TABELAS

Tabela 2.1 – Transferência de dados nas principais estruturas de programação (Baker, 1998).....	30
Tabela 2.2 – Vantagens e desvantagens do uso da arquitetura hierárquica em sistemas baseados em agentes.....	37
Tabela 2.3 – Vantagens e desvantagens do uso da arquitetura heterárquica em sistemas baseados em agentes.....	41
Tabela 2.4 – Grupo corrente de especificações padronizadas pela FIPA ( <a href="http://www.fipa.org/repository/standardspecs.html">http://www.fipa.org/repository/standardspecs.html</a> ; 31/07/2007) .....	48
Tabela 2.5 – Parâmetros de mensagem em FIPA –ACL (FIPA, 2002).....	51
Tabela 4.1 – Documento dos requisitos do domínio .....	102
Tabela 4.2 – Descrição geral dos casos de uso.....	105
Tabela 5.1 – Modelo formal da mensagem Inform padronizada pela linguagem FIPA – ACL.....	118
Tabela 6.1 – Tabela de tempos da peça 4.....	158
Tabela 6.2 – Número de operações diferentes por recurso.....	161
Tabela 6.3 - Operações x Recursos para a peça 1 x Tempo de processamento.....	161
Tabela 6.4 – Tempo mínimo de processamento por peça .....	162
Tabela 6.5 – Peças x níveis de flexibilidade.....	163
Tabela 6.6 – Agrupamento de peças por nível de flexibilidade dominante .....	166
Tabela 6.7 – Flexibilidade de máquina por recurso.....	167
Tabela 7.1 – Diferente níveis de setup de preparação e fixação utilizados no modelo.....	173
Tabela 7.2 – Comparação dos resultados obtidos pela abordagem proposta para a variável <i>makespan</i> para a mesma condição de flexibilidade de KIM et al. (2003) e POTTER (1997).....	178
Tabela 7.3 – Comparação dos resultados de <i>makespan</i> com a inclusão da flexibilidade de máquina com os valores obtidos por KIM et al. (2003) e POTTER (1997).....	179
Tabela 7.4 – Comparação dos tipos e níveis de flexibilidades predominantes com e sem o uso de flexibilidade de operações .....	182
Tabela 7.5 – Comparação dos resultados de <i>makespan</i> somente com as flexibilidades de operação e seqüência com os valores obtidos por KIM et al. (2003) e POTTER (1997) .....	183
Tabela 7.6 – Comparação dos resultados de <i>makespan</i> utilizando-se as flexibilidades de operação, seqüência e máquina, com os valores obtidos por KIM et al. (2003) e POTTER (1997) .....	185

Tabela 7.7 - Comparação dos resultados de <i>makespan</i> entre os grupos A, B, C e D.....	186
Tabela 7.8 – Comparação dos resultados obtidos pela abordagem proposta para a variável <i>flow time</i> para a mesma condição de flexibilidade de KIM et al. (2003) e POTTER (1997)	190
Tabela 7.9 – Comparação dos resultados de <i>flow time</i> com a inclusão da flexibilidade de máquina com os valores obtidos por KIM et al. (2003) e POTTER (1997).....	192
Tabela 7.10 – Comparação dos resultados de <i>makespan</i> somente com as flexibilidades de operação e seqüência com os valores obtidos por KIM et al. (2003) e POTTER (1997) .....	193
Tabela 7.11 – Comparação dos resultados de <i>flow time</i> utilizando-se as flexibilidades de operação, seqüência e máquina, com os valores obtidos por KIM et al. (2003) e POTTER (1997) .....	195
Tabela 7.12 - Comparação dos resultados de <i>flow time</i> entre os grupos A, B, C e D .....	197
Tabela 7.13 - Comparação dos resultados de <i>makespan</i> entre os grupos C, D, Potter e Kim et. al, para os diversos <i>setups</i> simulados .....	201
Tabela 7.14 - Comparação dos resultados de <i>makespan</i> entre os grupos A, B, Potter e Kim et. al, para os diversos <i>setups</i> simulados .....	203
Tabela 7.15 - Comparação dos resultados de <i>flow time</i> entre os grupos C, D, Potter e Kim et. al, para os diversos <i>setups</i> simulados .....	205
Tabela 7.16 - Comparação dos resultados de <i>flow time</i> entre os grupos A, B, Potter e Kim et. al, para os diversos <i>setups</i> simulados .....	207
Tabela 7.17 – Comparação do <i>flow time</i> para o grupo A sem e com seqüência dependente do <i>setup</i> .....	211
Tabela 7.18 – Comparação do <i>flow time</i> para o grupo B sem e com seqüência dependente do <i>setup</i> .....	211
Tabela 7.19 – Comparação do <i>flow time</i> para o grupo C sem e com seqüência dependente do <i>setup</i> .....	212
Tabela 7.20 – Comparação do <i>flow time</i> para o grupo D sem e com seqüência dependente do <i>setup</i> .....	212
Tabela 7.21 – Comparação do <i>makespan</i> para o grupo A sem e com seqüência dependente do <i>setup</i> .....	214
Tabela 7.22 – Comparação do <i>makespan</i> para o grupo B sem e com seqüência dependente do <i>setup</i> .....	214
Tabela 7.23 – Comparação do <i>makespan</i> para o grupo C sem e com seqüência dependente do <i>setup</i> .....	215
Tabela 7.24 – Comparação do <i>makespan</i> para o grupo C sem e com seqüência dependente do <i>setup</i> .....	215

Tabela 8. 1 – Valores da variável <i>flow time</i> obtidos com o sistema especialista para os grupos A, B, C e D nos sets 01, 12 e 24 .....	233
---	-----

## RESUMO

O presente trabalho trata do problema da execução das funções de planejamento de processo e a programação da produção de forma integrada ou paralela, ao invés da abordagem tradicional e seqüencial. É esta nova forma de execução que se apresenta como o campo ideal para o emprego de planos de processos alternativos, buscando melhorias nas métricas da programação, isto porque planos de processos pré-fixados contribuem, via de regra, para um desbalanceamento no carregamento de máquina, criando gargalos produtivos, além de gerar conflitos de agendamento quando da alocação das máquinas. O contexto apresentado foi motivador para o desenvolvimento deste trabalho, através do qual busca-se responder questões relativas à (a) como os planos de processos alternativos, gerados durante o planejamento dinâmico, podem influenciar no seqüenciamento da produção, sob a ótica das diferentes flexibilidades a serem consideradas no planejamento do processo; (b) como representar estas flexibilidades de forma adequada em um modelo de simulação; (c) quais as métricas que devem ser adotadas para a medição da performance do seqüenciamento da produção; e (d) qual o tipo de layout a ser considerado como alvo deste tipo de correlação? São estas, entre outras, as questões que levam ao objetivo traçado para este trabalho de pesquisa, que é: *Propor, implementar e validar um modelo de simulação baseado em uma sociedade multiagentes, que permita o estudo da correlação entre planos de processo alternativos, representados sob a forma de diferentes flexibilidades, e a programação da produção em um ambiente de layout funcional.* Para alcançar este objetivo foi definida uma sociedade baseada em agentes; criados protocolos de negociação e re-negociação que permitem aos agentes cooperar e coordenar suas ações; e finalmente foi desenvolvido e validado um protótipo de simulação para a integração entre planos de processos alternativos e a programação da produção aderente às normas de padronização internacionais. Os objetivos descritos foram atingidos, com o modelo de simulação proposto permitindo a combinação entre quatro diferentes tipos de flexibilidades nos planos de processo: (a) de seqüência; (b) de operações; (c) de processo; e (d) de máquina. Além disto, o modelo ainda permite o uso de precedências entre as operações. Finalmente foi desenvolvido uma extensão do modelo proposto, baseado em sistemas especialistas, que através da sugestão do roteamento inicial das peças, pode ser usado para minimizar a variabilidade do modelo.

**Palavras-chave:** integração planejamento de processo e programação da produção, planos de processos alternativos, modelo de simulação baseado em agentes.

## ABSTRACT

The present work addresses the execution problem of the process planning and scheduling in an integrated or parallel way, despite its traditional and sequential approach. This new way of execution which is presented as an ideal field for the use of alternative process plans, looking for increasing in the metric sequence, it happens because the pre fixed process plan helps, normally for the unbalance in the machine charging, creating bottlenecks, besides generating sequence conflicts when the machine is allocated. The context presented was motivated for the development of this work, through which it was searched the answer the questions related to (a) alternative process plans, generated during the dynamic planning, they can influence the production sequence, under the view of different flexibilities to be considered in the planning of the process; (b) represent these flexibilities in an adequate an useful way in the simulation way; (c) which are the metrics which should be adopted to the measurement of the sequence performance of the production; and (d) which is the kind of the layout to be considered as a target of this kind of correlation? The questions which take the target drawn to the research work up, which is: Presentation, implementation and validation of a simulation model based in the multi agent's society, which allows the correlation study between alternative process plans, represented as a flexible different way, and the production sequence in a job shop environment. To reach this objective it was defined an agent based society; creating negotiation and re-negotiation protocols which allows its agent to cooperate, and to coordinate its actions; and finally, it was developed and validated a simulation prototype validated for the integration between alternative process plans and its sequence of production according to international standard regulations. The objective described was reached with the proposed simulation model allowing the combination among four different kinds of flexibility in the process plan: (a) sequence; (b) operation; (c) process; (d) and machine. Besides this, the model allows the use of precedent between operations. Finally, it was developed in its extension to the proposed model, based in expert systems, which through the suggestion of initial routing of the parts, which can be used to minimize the model variability.

**Keywords:** integration of process planning and scheduling, alternative process planning, simulated agent based model.

---

## CAPÍTULO 1 – INTRODUÇÃO

### 1.1 Origem do trabalho

Produtos novos, cujo tempo entre o projeto e o usuário final é cada vez mais reduzido; a necessidade de aumento da variedade ofertada; uso racional da capacidade instalada em lugar do uso contínuo; a busca por custo competitivo e pela qualidade; o desafio das cadeias de suprimento, recursos ou plantas fabris distribuídas. A manufatura vem mudando suas características e isto afeta a forma como suas funções devem ser vistas e executadas.

Uma das consequências do contexto acima apresentado pode ser observada na forma como as empresas de manufatura vêm cada vez mais migrando da abordagem tradicional e sequencial de executar as funções de planejamento de processo e seqüenciamento da produção, para uma forma integrada ou mesmo paralela de execução.

As razões pela qual a integração entre planejamento de processo e seqüenciamento da produção é atraente, não somente como um campo de pesquisa, mas também como uma ferramenta competitiva para as empresas, são inúmeras. MOON e SEO (2005) citam a alta qualidade do produto e baixo custo como consequências inerentes a esta integração. Já EVERSHEIM (2005) apresenta, entre outras razões, a realização de um melhor planejamento, a utilização mais eficiente dos recursos produtivos e a possibilidade de reação eficiente no caso de alterações do status do chão-de-fábrica. Além destes, inúmeros autores (HUANG, ZHANG e SMITH, 1995; LENDERINK e KALS, 1993; USHER, 2003) concordam que o desempenho da manufatura pode ser melhorado através da integração nos mais diversos aspectos.

A integração entre planejamento de processo e seqüenciamento da produção também se apresenta como o campo ideal para o emprego de planos de processos alternativos, buscando melhorias nas métricas do seqüenciamento. Isto se deve ao fato de que planos de processos pré-fixados contribuem para um desbalanceamento no carregamento de máquina, criando gargalos produtivos, além de gerar conflitos de seqüenciamento quando da alocação das máquinas (TAN e KHOSHNEVIS, 2000).

TAN e KHOSHNEVIS, (2000) fazem uma analogia sobre o uso de planos de processo alternativos integrados ao seqüenciamento da produção, ao afirmar que o uso destes cria uma nova dimensão para o seqüenciamento, fazendo com que o mesmo deixe de ser visto como plano, formado pelos eixos máquina e tempo, e deslocando-se para um espaço tridimensional, com a criação de um terceiro eixo, chamado de método, que é a representação das diferentes

formas de processo, máquinas, tempos e outras variáveis que configuram as diferentes alternativas de fabricação.

## 1.2 Problema de pesquisa

Segundo AHMAD et al. (2001), o planejamento estático de processo trata da geração de planos alternativos genéricos e que não levam em conta as especificações que dizem respeito ao status operacional do chão-de-fábrica. Este planejamento envolve a seleção de tarefas, a programação dos processos e máquinas que tenham potencial para virem a ser utilizadas.

O planejamento de processo dinâmico, por sua vez, se dá no momento em que a peça é liberada para a fabricação. Neste momento, o plano de processos macro é recuperado e completado levando em consideração a disponibilidade dos recursos do chão-de-fábrica, além dos objetivos específicos esperados da programação (minimização de *setup*, data de entrega e outros). O planejamento dinâmico permite um tipo de replanejamento da peça escolhida tendo como base a análise dos processos de manufatura, gerando com isto um grupo ranqueado de soluções próximas ao plano de processo ótimo.

O contexto acima nos traz algumas das razões que motivaram este trabalho, questões relacionadas à como os planos de processos alternativos, gerados durante o planejamento dinâmico, podem influenciar na programação da produção, sob a ótica das diferentes flexibilidades a serem consideradas no planejamento do processo? Como representar estas flexibilidades de forma adequada em um modelo de simulação? Quais as métricas que devem ser adotadas para a medição da performance no agendamento da produção? Qual o tipo de *layout* a ser considerado como alvo deste tipo de correlação?

USHER (2003) afirma que nos anos recentes têm aumentado o interesse da comunidade científica e empresarial no desenvolvimento de sistemas dinâmicos que sejam capazes de integrar o planejamento de processos com a programação da produção, visando com isto aumentar a flexibilidade da programação.

Entretanto, segundo MONOSTORI et al. (2006), problemas que visam a integração de funções de produção distintas, como o planejamento e a programação, aumentam em muito a complexidade de decisão, o que conseqüentemente demanda por maior capacidade de processamento para sua solução.

TAN e KHOSNHEVIS (2000) explicam que ambas as funções, planejamento do processo e seqüenciamento da produção, necessitam explorar seus próprios e vastos espaços de soluções a fim de buscar uma que seja adequada. Ao se integrar essas funções, unifica-se o

espaço de solução a ser percorrido, o que provê uma base única para a efetiva combinação dos esforços das soluções, incorrendo em um aumento na complexidade do espaço de busca, causado por esta unificação.

O grau de complexidade, considerando-se apenas a ótica do seqüenciamento, pode ser ainda mais elevado, de acordo com SHEN et al. (2006), quando variáveis como recursos, operadores ou *setups* são considerados. Isto pode ser observado tomando-se como exemplo o clássico problema de um *layout* funcional com  $n$  trabalhos distribuídos em  $m$  recursos. Neste caso, o número de soluções possíveis seria dado por  $(n!)^m$ , mas ao considerarmos uma única nova variável, como  $k$  operadores, o número de soluções possíveis aumenta exponencialmente para  $((n!)^m)^k$ .

Problemas com tais características são conhecidos como *NP-Hard*, onde uma solução ótima somente pode ser alcançada através de algoritmos enumerativos, onde o tempo computacional aumenta exponencialmente com o aumento do tamanho do problema e serão foco de estudo do presente trabalho.

Arquiteturas baseadas em agentes, de acordo com DE OLIVEIRA (2005), se mostram adequadas para o modelamento de ambientes cooperativos entre entidades heterogêneas. SHEN et al. (2006) e JENNINGS (2001) explicam que abordagens baseadas em agentes possuem inúmeras vantagens potenciais quando utilizadas em sistemas de alta complexidade, provendo robustez e eficiência aos mesmos.

Entre estas vantagens, uma em particular, é de especial importância ao ambiente modelado: o uso de computação paralela através de um número variável de processadores. Isto é possível através da decomposição de um problema em vários sub-problemas, onde os agentes podem agir como componentes autônomos com a habilidade de cooperar entre si a fim de alcançar um objetivo global.

Este tipo de decomposição, que permite o compartilhamento de tarefas, é particularmente útil para a decomposição do planejamento de processo em sub-problemas, como o processamento em recursos alternativos, alocação de recursos, agendamentos de operações e outros (MONOSTORI et al., 2006).

A partir deste contexto é traçada a pergunta de pesquisa que norteia o presente trabalho:

***Uma solução baseada em agentes é adequada para o estudo da correlação entre planos de processo alternativos, representados sob a forma de diferentes flexibilidades, e a programação da produção em um ambiente de layout funcional?***

---

São estas, entre outras, as questões e considerações que levam aos objetivos traçados para o desenvolvimento do presente trabalho de pesquisa.

### 1.3 Objetivos do trabalho

#### 1.3.1 Objetivo Geral

*Propor, implementar e validar um modelo de simulação baseado em uma sociedade multiagentes, que permita o estudo da correlação entre planos de processo alternativos, representados sob a forma de diferentes flexibilidades, e a programação da produção em um ambiente de layout funcional.*

#### 1.3.2 Objetivos Específicos

A fim de cumprir o objetivo geral proposto, vários objetivos específicos deverão ser alcançados, dentre eles destacam-se:

- a. Investigar a adequação do uso de uma abordagem baseada em agentes para o estudo da correlação entre planos de processo alternativos e a programação da produção;
- b. Definir uma arquitetura baseada em agentes autônomos que irá suportar o modelo de simulação desenvolvido;
- c. Desenvolver um protocolo de negociação que permita aos agentes cooperar e coordenar suas ações a fim de permitir o roteamento das ordens de serviços entre os recursos de manufatura disponíveis no modelo;
- d. Possibilitar escalabilidade e flexibilidade ao modelo proposto, implicando que a inclusão de novos agentes, representando diferentes recursos ou trabalhos, ou que a remoção de agentes existentes, possam ser executadas sem o comprometimento das relações pré-estabelecidas;
- e. Identificar e selecionar os recursos de software necessários à implementação do modelo proposto de acordo com as especificações;
- f. Desenvolver e construir um protótipo de laboratório do modelo proposto de acordo com as especificações.

---

## 1.4 Justificativa

Um trabalho de pesquisa pode ser justificado a partir de três aspectos: quanto à sua relevância, quanto ao seu ineditismo e quanto à sua complexidade. Neste sub-item justifica-se o tema proposto a partir de seu ineditismo, uma vez que sua relevância e complexidade estão intrínsecas aos itens 1.1 e 1.2 já apresentados.

Além disto, este sub-item também tem como propósito apresentar a adequação da abordagem escolhida, uma estrutura baseada em agentes, para o desenvolvimento do modelo proposto.

### 1.4.1 Ineditismo

Inúmeros trabalhos utilizam abordagens baseadas em agentes para a solução de problemas relativos à manufatura (PARUNAK et al., 2001; RASHID et al., 2001; KIM et al. 2003; PECHOUCEK et al., 2005), e vários são os autores (DENKENA et al., 2003; IMBERTI e TOLIO, 2003; WANG et al., 2003; LIM e ZHANG, 2004) que dirigiram seus esforços ao problema específico da integração do planejamento do processo e programação da produção, mas não sob a ótica da influência das diversas flexibilidades disponíveis em um plano de processo de forma concomitante. Sendo que o próprio modelo conceitual e a especificação de projeto da organização multiagente podem ser considerados como contribuições inéditas.

## 1.5 Metodologia

Uma pesquisa pode ser classificada de quatro formas, de acordo com SILVA e MENEZES (2000): quanto à natureza; quanto à forma de abordagem do problema; quanto aos objetivos; e quanto aos procedimentos técnicos. A seguir é feita a classificação desta pesquisa.

### 1.5.1 Quanto à sua natureza

Em relação à sua natureza, uma pesquisa pode ser classificada em básica ou aplicada (SILVA e MENEZES, 2000). Esta tese tem as características de uma pesquisa básica, porque objetiva gerar conhecimentos novos úteis a partir do estudo da correlação entre planos de processo alternativos, representados sob a forma de diferentes flexibilidades, e o

---

seqüenciamento da produção em um ambiente de layout funcional. No entanto, não visa uma aplicação prática em curto prazo, embora se acredite que a mesma possa vir a ser discutida e comprovada em trabalhos futuros.

### 1.5.2 Quanto à forma de abordagem do problema

Em relação à forma de abordagem do problema, uma pesquisa pode ser classificada em quantitativa ou qualitativa (SILVA e MENEZES, 2000). Esta pesquisa não possui as características de uma pesquisa qualitativa, isto é, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito que não pode ser traduzido em números.

Por outro lado, possui características citadas por SILVA e MENEZES (2000) como sendo de uma pesquisa quantitativa: considera que tudo pode ser quantificável, o que significa traduzir em números opiniões e informações para classificá-las e analisá-las, requerendo o uso de recursos e de técnicas estatísticas.

Dentro do contexto acima apresentado este trabalho de pesquisa é predominantemente quantitativo.

### 1.5.3 Quanto aos objetivos

Em relação aos seus objetivos, uma pesquisa pode ser classificada em exploratória, descritiva ou explicativa (SILVA e MENEZES, 2000; GIL, 1993). Esta pesquisa tem características de uma pesquisa exploratória e explicativa. Exploratória porque envolve como procedimentos uma pesquisa bibliográfica e um estudo de caso, para validar o modelo de simulação proposto no presente trabalho. Explicativa porque visa identificar os fatores que determinam ou contribuem para a ocorrência dos fenômenos, aprofundando o conhecimento da realidade porque explica a razão, o “porquê”, das coisas.

### 1.5.4 Quanto aos seus procedimentos técnicos

Em relação aos procedimentos técnicos uma pesquisa pode, segundo (SILVA e MENEZES, 2000; GIL, 1993), ser classificada em bibliográfica, documental, experimental, levantamento, estudo de caso ou pesquisa *expost-facto*.

Dentro das categorias acima citadas, a presente pesquisa se classifica como: (a) bibliográfica, pois é elaborada a partir de material já publicado, constituído principalmente de livros e artigos de periódicos; (b) experimental, pois determina um objeto de estudo, selecionando-se as variáveis que seriam capazes de influenciá-lo, definindo as formas de controle e de observação dos efeitos que a variável produz no objeto e finalmente apresentando características de estudo; e (c) estudo de caso, envolvendo o estudo profundo e exaustivo de um ou poucos objetos de maneira que se permita o seu amplo e detalhado conhecimento.

## 1.6 Delimitação do trabalho

Este trabalho não tem como objetivo propor nenhum método ou ferramenta visando à busca de um método de seqüenciamento ótimo da produção. Ao invés disto, uma heurística é utilizada com a única intenção de permitir o roteamento das ordens de produção entre os recursos disponíveis no chão-de-fábrica, a ser discutido no capítulo 6, que trata da formulação da proposta de negociação.

Analisa-se no trabalho o tema da flexibilidade nos planos de processo e sua correlação com o seqüenciamento da produção. Porém, nessas análises serão avaliados apenas quatro diferentes tipos de flexibilidade: de processo, de seqüenciamento, de máquina e de operações, sendo que as flexibilidades de processo, seqüenciamento e operações visam a replicação das condições adotadas no trabalho de KIM et al. (2003) e a posterior validação do modelo aqui proposto. O tema de validação e verificação é discutido no item 3.13 do presente trabalho, enquanto os tipos de flexibilidades adotadas são apresentadas e discutidas nos itens 4.1.6 à 4.1.10.

Não é foco deste trabalho o estudo da influência de nenhum tipo de distúrbio no chão-de-fábrica, como quebra de máquinas, falta de operadores ou de matéria-prima, no seqüenciamento da produção. Porém, é importante ressaltar que devido ao mecanismo de renegociações entre ordens de serviço e recursos, tratado no item 5.5.2, a geração de nervosismo no sistema pode vir a ocorrer, o que implica na necessidade de uma solução robusta na presença de falhas.

Como o principal objetivo é modelar um ambiente de *layout* funcional para o estudo da correlação entre planos de processo alternativos e o seqüenciamento da produção, não faz parte do escopo deste trabalho uma discussão sobre os demais *layouts* produtivos.

---

Finalmente, a análise dos resultados se dará em um ambiente onde as máquinas são consideradas como disponíveis e onde todos os demais recursos necessários para a execução dos trabalhos estão disponibilizados.

## 1.7 Organização da tese

Este trabalho está estruturado em nove capítulos, sendo estes divididos em quatro temas distintos, conforme apresentado na figura 1.1. A primeira parte da tese é relativa à introdução e revisão bibliográfica, sendo composta pelos capítulos 1 e 2; a segunda parte trata da visão teórica que permeia o modelo proposto, sendo composta pelos capítulos 3 e 4. A seguir, a terceira parte, é relativa ao projeto do protótipo, sendo constituída pelos capítulos 5 e 6; e finalmente, a última parte, trata dos experimentos realizados e da conclusão final do trabalho, sendo composta pelos capítulos 7, 8 e 9.

Neste primeiro capítulo (**Capítulo 1 – Introdução**) é feita a introdução ao trabalho, discutindo-se o problema de pesquisa, os objetivos gerais e específicos, a justificativa em termos de ineditismo e complexidade, bem como suas limitações e estrutura.

O **Capítulo 2 (Conceitualização: sistemas baseados em agentes)** apresenta uma revisão de literatura sobre os conceitos que motivaram a utilização de uma abordagem baseada em agentes para o modelo proposto. É vista uma definição formal de agentes e algumas das características relevantes ao escopo do presente trabalho. As diversas arquiteturas de sistemas baseados em agentes, bem como as padronizações e diferentes formas de negociação são discutidas a seguir. Finalmente são apresentadas as linguagens de comunicação e os principais *frameworks* para o desenvolvimento de sistemas desta natureza.

No **Capítulo 3 (Ciclo de vida para o desenvolvimento de um modelo de simulação baseado em multiagentes)** são tratadas questões relativas ao desenvolvimento de um modelo de simulação baseado em multiagentes. Primeiramente é apresentada uma visão do ciclo de vida para o desenvolvimento de um modelo de simulação que não se baseia em agentes e, em seguida, é vista a adequação de agentes para o desenvolvimento de modelos de simulação e a necessidade de uma metodologia voltada especificamente para este fim. Na seqüência é apresentada a metodologia PASSI para o desenvolvimento e especificação de um modelo de simulação baseado em agentes. Finalmente é discutido o processo de validação e verificação para o modelo adotado.

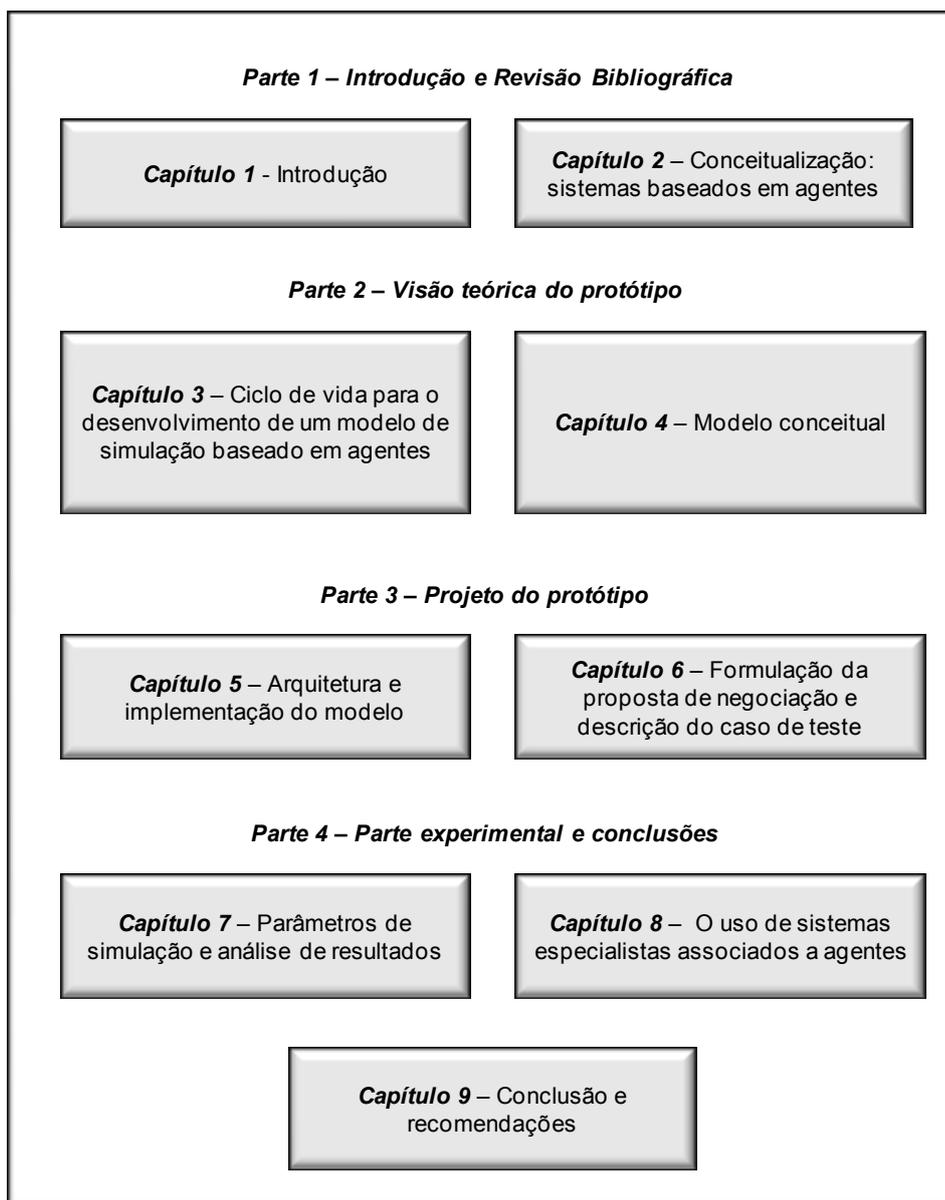


Figura 1.1 – Estrutura da tese

O **Capítulo 4 (Modelo conceitual)** trata da validação do contexto da simulação e de como as exigências para o seu desenvolvimento foram especificadas e cumpridas, bem como das bases de conhecimento que posteriormente serão utilizadas pelos agentes para compor o modelo proposto. Primeiramente é feita a descrição do problema a ser modelado, seguida da caracterização do ambiente utilizado como modelo e pela descrição dos elementos que irão compor o modelo de simulação, bem como pela representação das flexibilidades inseridas nos plano de processos das peças. O capítulo encerra com a descrição dos requisitos de domínio do modelo proposto.

O **Capítulo 5 (Arquitetura e implementação do modelo)** trata do desenvolvimento computacional do modelo de simulação proposto através das fases da metodologia PASSI,

bem como de parte de sua verificação. São apresentadas neste capítulo as fases da metodologia PASSI que tratam da verificação do modelo. Inicialmente é dada continuidade à fase de requisitos do sistema, iniciada no capítulo 4, que trata do modelo conceitual adotado. Em seguida é apresentada a fase de sociedade de agentes, responsável por modelar a solução em função de seus papéis, interações, dependências e ontologia. O capítulo se encerra com o modelo de implementação, responsável por apresentar a arquitetura proposta em função de suas classes e métodos.

**O Capítulo 6 (Formulação da proposta de negociação e descrição do caso de teste)** inicia com a formulação da proposta de negociação, usada como critério de desempate e de agendamento entre as diversas ordens que estejam concorrendo concomitantemente aos recursos disponíveis durante uma simulação. Os tempos que compõem uma negociação também serão tratados e discutidos neste capítulo. Finalmente é apresentado o caso de teste desenvolvido por YEO et al. (2003), o qual será utilizado para a validação do modelo.

**O Capítulo 7 (Parâmetros de simulação e análise de resultados)** trata da análise dos resultados obtidos através do protótipo de simulação desenvolvido em laboratório para o presente trabalho.

Inicialmente são apresentados os parâmetros de simulação utilizados, tanto para a validação do modelo proposto, quanto para as demais condições estudadas. A seguir é feita uma breve discussão a respeito dos trabalhos de KIM et al. (2003) e POTTER (1997), utilizados para a validação e comparação de resultados obtidos neste trabalho.

Finalmente são apresentados e discutidos os valores das variáveis de resposta *makespan* e *flow time* obtidos através do modelo proposto, sendo então comparados àqueles obtidos por KIM et al. (2003) e POTTER (1997), valores estes que foram obtidos a partir do caso de testes sugerido por YEO et al. (2003).

**O Capítulo 8 (O uso de sistemas especialistas associados a agentes)** tem por finalidade apresentar a estrutura do banco de dados utilizado modelo proposto, o qual foi desenvolvido em MySQL que é um sistema de gerenciamento de banco de dados que utiliza a linguagem de consulta estruturada SQL (*Structured Query Language*), a qual permite o desenvolvimento de bancos de dados relacionais.

Dentro deste contexto, será feita uma breve revisão a respeito do uso de sistemas especialistas associados a agentes, bem como da escolha do *Shell* empregado, sendo

---

finalmente descrita a implementação das novas funcionalidades no modelo proposto e discutidos os resultados obtidos.

No último capítulo, **Capítulo 9 (Conclusões e recomendações)** são apresentadas as discussões, conclusões, contribuições do trabalho, além das recomendações para trabalhos futuros.

---

## CAPÍTULO 2 – CONCEITUALIZAÇÃO: SISTEMAS BASEADOS EM AGENTES

### 2.1 Introdução

Este capítulo tem por objetivo apresentar uma revisão de literatura sobre os conceitos que motivaram a utilização de uma abordagem baseada em agentes para o modelo proposto.

Primeiramente é vista uma definição formal de agentes e algumas das características relevantes ao escopo do presente trabalho. As diversas arquiteturas de sistemas baseados em agentes, bem como as padronizações e diferentes formas de negociação são discutidas a seguir. Finalmente são apresentadas as linguagens de comunicação e os principais *frameworks* para o desenvolvimento de sistemas desta natureza.

### 2.2 Definições gerais

Muitas foram às tendências e pontos de vista que resultaram no interesse, projeto e uso de agentes inteligentes. LUGER (2004) explica que dois pontos de vista na comunidade de pesquisadores de IA, nos anos 80, influenciaram diretamente nos sistemas baseados em agentes.

O primeiro era a tradição em pesquisas da inteligência artificial, ou comunidade IAD [...] seu objetivo era compreender como solucionadores de problemas distribuídos poderiam ser efetivamente coordenados para solução inteligente de problemas.

O segundo ponto de vista foi aquele de Rodney Brooks e seu grupo no MIT, que questionava a necessidade de um esquema representacional central e, com a sua arquitetura de subordinação, mostrava como seres inteligentes poderiam evoluir de formas inferiores e básicas de inteligência.

Já WOOLDRIDGE (2002) diz que cinco foram às tendências que marcaram a história da computação, que são:

- Onipresença (do inglês *ubiquity*);
- Interconectividade;
- Inteligência;
- Delegação de tarefas; e
- Orientação ao usuário.

Já a programação, ainda de acordo com BAKER (1998) e WOOLDRIDGE (2002) evoluiu através das seguintes etapas até o projeto de agentes:

- Código de máquina;
- Linguagem Assembly;
- Sub-rotinas;
- Procedimentos e funções;
- Tipos de dados abstratos;
- Objetos, e
- Agentes.

De acordo com BAKER (1998) é adequado observar que um agente tem seu próprio sistema de valor e meios próprios para se comunicar com outras entidades ou estruturas de programação. A tabela 2.1 apresenta as diferentes formas de transferências de dados e comunicação, nas principais estruturas de programação, até chegar à tecnologia de agentes.

Tabela 2.1 – Transferência de dados nas principais estruturas de programação

Fonte: BAKER, 1998

	<b>Informação</b>	<b>Fluxo</b>
	Dados binários (hexadecimal...)	Go to
	Dados básicos (real, caractere...)	For, while, repeat...
	Dados estruturados (strings, arrays....)	Funções
	Dados abstratos (filas, sets, listas, árvores...)	Operações
	Objetos (palettes, tool bars...)	Métodos
	Agentes	Mensagens

Apesar do interesse apresentado pela academia no que se refere ao uso de agentes, não existe uma definição consensual. Segundo IMAM e KODRATOFF (1997) isto ocorre devido ao fato de que uma definição irá depender diretamente do objetivo operacional assim como do contexto do problema no qual o agente estará inserido.

### 2.3 Definição e características dos sistemas baseados em agentes

#### 2.3.1 O que é um agente?

LUGER (2004) define agente como um elemento de uma sociedade que pode perceber aspectos (frequentemente limitados) de seu ambiente e afetá-lo. Já WOOLDRIDGE (2002) os define como um sistema de computador que é capaz de ação independente em nome de seu usuário. Uma definição pragmática é dada por MAS (1997) apud PAOLUCCI e SACILE (2005), que diz que um agente é um sistema computacional que, enquanto existir, possui metas, sensores e efeitos e que decide de forma autônoma quais as ações a tomar em uma situação corrente, a fim de maximizar o progresso no cumprimento de suas metas.

A definição que permeia este trabalho foi estabelecida por WOOLDRIDGE e JENNINGS (1995) e é amplamente utilizada e aceita por pesquisadores na área de sistemas baseados em agentes (JENNINGS, 2001), nela um agente é definido “como um sistema de computador que está situado em algum ambiente e é capaz de ações autônomas e flexíveis, a fim de alcançar seus objetivos planejados”. Por flexível, entende-se que o sistema deve ser responsivo, pró-ativo e social.

WOOLDRIDGE E JENNINGS (1995) identificaram três diferentes classes de agentes:

- Agentes que executam diretamente as tarefas baseadas em regras pré-definidas e suposições;
- Agentes que executam tarefas bem definidas pelo usuário; e
- Agentes que agem voluntariamente, com informações ou serviços, sempre que julga apropriado, sem necessitar que lhe seja explicitamente solicitado algo.

SHEN et. al. (2001) classificam os agentes em diferentes modelos:

- Agentes móveis – são agentes de software que podem migrar de uma máquina para outra;
- Agentes reativos – são agentes de software simples (hardware do tipo e/ou), equipados com um mecanismo simples para reagir à mudanças no seu ambiente;
- Agentes cognitivos – são aqueles capazes de argumentos complexos;
- Agentes autônomos – são aqueles que procuram por suas próprias metas;
- Agentes de interface – são usados como intermediários entre o sistema automático e seres humanos; e
- Agentes intermediários – também conhecidos como agentes de meio, são agentes que trabalham com um papel específico em uma arquitetura complexa de agentes.

Vários são os atributos que caracterizam um agente. REZENDE (2005) diz que os principais são:

Reatividade – capacidade de reagir às mudanças do ambiente a partir do reconhecimento de um contexto conhecido;

Adaptabilidade – capacidade de adaptar-se seu processo de decisão frente a situações desconhecidas;

Mobilidade – capacidade de mover-se e ser executado em outras plataformas;

Personalidade – capacidade de personificar-se, utilizando recursos que lembrem características humanas como a emoção ou o mau humor;

Interatividade com o usuário – capacidade de interagir com usuários e, considerando os possíveis mal-entendidos, reagir às falhas de comunicação de maneira aceitável;

Ambiente de atuação – caracteriza o local onde o agente vai atuar, isto é, em ambientes fechados (*desktop*) ou abertos (Internet); e

Comunicabilidade – capacidade de interagir com outros agentes computacionais para a obtenção de suas metas.

WOOLDRIDGE E JENNINGS (1995) citam ainda como características dos agentes:

Autonomia – capacidade de executar a maioria de suas tarefas sem a intervenção direta de humanos. Além disto, deve ter um determinado grau de controle sobre suas próprias ações e seu *status* interno;

Habilidades sociais – um agente deve ser apto a interagir com outros agentes ou humanos;

Pró-atividade – ao responder ao seu ambiente, um agente deve ser oportunista, ter um comportamento direcionado a metas e ter iniciativa;

Veracidade - é a suposição que um agente não comunicará falsa informação conscientemente.

A partir das características discutidas, foi construída uma taxonomia na qual foi considerado que um agente é um tipo de sistema computacional como características próprias e especiais. Esta taxonomia é apresentada na figura 2.1.

REZENDE (2005, pg.276) explica que esta taxonomia é formada por quatro diferentes eixos:

Eixo cognitivo – um agente pode conter um modelo de representação interna do ambiente e dos outros agentes baseado em estados mentais; e um modelo racional de decisão (agente cognitivo) ou apenas agir baseado num modelo de reações aos estímulos provocados pelo ambiente (agente reativo);

Eixo de foco – um agente pode enfatizar similaridades físicas com humanos (estrutural) [...] ou pode enfatizar similaridades comportamentais;

Eixo de atuação – um agente pode atuar de forma isolada ou interagindo com outros agentes (social);

Eixo ambiental – um agente individual pode atuar no *desktop* ou em uma rede internet ou intranet.

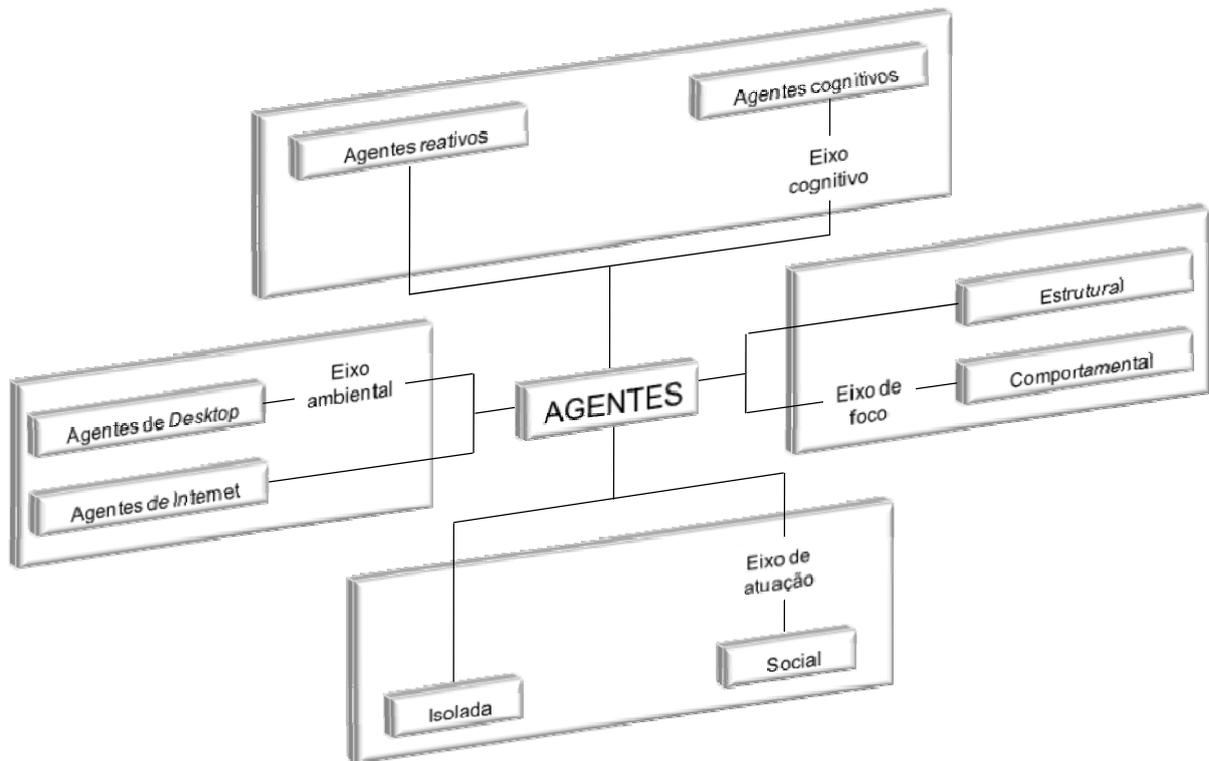


Figura 2.1 – Taxonomia de agentes  
 Fonte: REZENDE, 2005

É importante observar que não é necessário que todas as características citadas apareçam simultaneamente em um agente. No contexto deste trabalho, por exemplo, as mais importantes são: autonomia, reatividade e comunicabilidade.

Uma questão chave a ser respondida por um agente é como decidir quais as ações que devem ser executadas a fim de satisfazer da melhor possível seus objetivos pré-definidos. Em resumo, “um agente não é apenas mais um tipo de sistema computacional. Contribuem para diferenciá-lo [...] além de autonomia, ter um processo de decisão racional e um planejador de ações que concretizem suas decisões na obtenção de suas metas.” (REZENDE, 2005).

De acordo com JENNINGS (2001), ao se adotar uma visão orientada a agentes para a solução de um problema, torna-se clara a aparente necessidade de envolver múltiplos agentes. Estes múltiplos agentes têm por finalidade representar a natureza descentralizada do problema, ou seus vários locais de controle ou mesmo as diferentes perspectivas possíveis ou ainda seus interesses conflitantes.

### 2.3.2 O que são sistemas multiagentes?

Configura-se como um ambiente de atuação multiagentes, quando mais de um agente atua, seja de modo competitivo ou cooperativo, de forma a realizar um determinado conjunto de tarefas ou objetivos. Ou seja, um conjunto de agentes autônomos em atividade caracteriza o que é chamado de MAS - Sistema Multiagentes (*Multi-Agent Systems*).

Uma definição mais formal de sistemas multiagentes é dada por LUGER (2004), que diz que um MAS é um programa de computador com solucionadores de problemas localizados em ambientes interativos, que são capazes de ações flexíveis, autônomas e, ainda, socialmente organizadas que podem, mas não necessariamente, ser dirigidas para objetivos predeterminados ou metas.

A definição que permeia este trabalho é dada por MARIK e McFARLANE (2005) que explicam que um sistema multiagente é um grupo de agentes organizado e precisamente definido. Possuindo princípios de organização de comunidade e de operação, como arquitetura, tipos de mensagens, protocolos de negociação, e outros. Sendo apoiado por uma plataforma ou infra-estrutura de agentes, com registros, cancelamentos de registros, suporte para comunicação e outros.

Quanto à sua usabilidade, vários autores (BOND e GASSER, 1988; JENNINGS e WOOLDRIDGE, 1998; WOOLDRIDGE, 2002) sugerem uma série de fatores para verificar a adequação de uma solução baseada em agentes para um determinado ambiente. Este ambiente deve, entre outras características ser:

Aberto ou ao menos altamente dinâmico, incerto e complexo;

Representável por agentes como uma metáfora natural na sua modelagem (o ambiente deve ter comportamentos cooperativos e ou competitivos, que visam à solução de problemas complexos); e

Naturalmente distribuído. Isto significa ter sua estrutura de dados, controle ou conhecimento distribuída. Ambientes que possuem sistemas de dados distribuídos em diferentes controles não permitem soluções centralizadas.

Conforme LUGER (2004), uma orientação a agentes se justifica quando determinadas características são relevantes ao escopo da solução do problema abordado. Em geral são características que incluem o fato de que:

1 – Agentes são autônomos ou semi-autônomos. Isto é, cada agente tem certas responsabilidades na solução de problemas, com pouco ou nenhum conhecimento do

---

que os outros agentes fazem ou de como fazem. Cada agente realiza a sua parte [...] ou relata os seus resultados para outros agentes;

2 – Os agentes são locais. Cada agente é sensível ao seu ambiente particular e (normalmente) não tem conhecimento do domínio completo de todos os agentes;

3 – Os agentes são interativos. Isto significa que eles formam uma coleção de indivíduos que cooperam numa tarefa particular. [...] as habilidades e as responsabilidades, mesmo quando vistas como coletivas, estão distribuídas pela população de indivíduos;

4 – A sociedade de agentes é estruturada. [...] uma solução final não será vista apenas como coletiva, mas também, como cooperativa; e

5 – [...] o resultado cooperativo global da sociedade de agentes pode ser visto como maior que a soma de suas contribuições individuais.

### 2.3.3 Motivação para o uso de sistemas multiagentes

O aumento do interesse no que diz respeito às aplicações de sistemas multiagentes pode ser explicado através dos seguintes pontos (NWANA et al., 1996; OUELHADJ et al., 1998; FERBER, 1999; OLIVEIRA et al., 1998; WEISS, 1999; SHEN et al., 2001; JENNINGS 2001; SHEN, 2002):

- a) Sistemas multiagentes permitem uma analogia com o mundo real, através do trabalho em equipe;
- b) Normalmente os problemas “reais” são física e funcionalmente distribuídos, caso dos sistemas de manufatura, tráfego aéreo, sistemas de saúde, entre outros;
- c) Sistemas complexos então além do controle direto. Sua operação se dá através da cooperação entre diferentes sub-sistemas, os quais podem ter diferentes interesses ou modos de operação;
- d) Estruturas centralizadas são difíceis de serem mantidas e reconfiguradas, sendo normalmente inflexíveis e ineficientes para satisfazer as necessidades do mundo real. Além disto, estas estruturas apresentam um alto custo em presença de falhas, uma vez que o volume de conhecimento a ser gerenciada pode ser muito grande;
- e) Necessidade de integração entre diferentes sistemas (*legacy systems*) e especialistas;
- f) O mundo real é um ambiente heterogêneo e como tal utiliza de diferentes tipos e modelos de dados, os quais operam de diferentes maneiras;
- g) Sistemas distribuídos são mais robustos e confiáveis na presença de falhas, permitindo uma detecção e recuperação mais rápidas. Isto ocorre pois a falha de um ou vários agentes não necessariamente implica na falha de todo o sistema;

- 
- h) Escalabilidade e flexibilidade. Ambas as características são provenientes das estruturas abertas e dinâmicas que caracterizam sistemas multiagentes, cuja principal capacidade é a de se adaptar ao tamanho de diferentes problemas através da adição de novos agentes, sem afetar a funcionalidade daqueles já existentes;
  - i) Eficiência computacional. Agentes podem operar de forma assíncrona e em paralelo, o que pode resultar no aumento da velocidade de solução;
  - j) É possível conectar os agentes de recursos diretamente aos dispositivos que representam e com isto realizar o re-agendamento em tempo real (agendamento dinâmico). Esta técnica permite um alto grau de tolerância a erros.

#### 2.3.4 Arquiteturas de sistemas baseados em agentes

A arquitetura de um agente pode ser definida como o padrão de relacionamento entre os módulos que compõem o agente. De forma análoga, a arquitetura de um sistema multiagentes pode ser expressa como o padrão de relacionamentos entre agentes (SHEN et al., 2001).

Três são as arquiteturas de sistemas multiagentes freqüentemente utilizadas no escopo da programação da manufatura, são elas: hierárquica, federada e autônoma (SHEN, 2002; PAOLUCCI e SACILE, 2005; e SHEN et al., 2006), as quais serão apresentadas e discutidas a seguir.

Na arquitetura hierárquica, figura 2.2, os agentes operam em diferentes níveis de uma hierarquia que basicamente reproduz a tradicional distribuição de responsabilidade de uma organização de controle e manufatura, onde existe um número de unidades semi-autônomas, freqüentemente fisicamente distribuídas, cada qual com um grau de controle sobre recursos locais ou com diferentes níveis de exigências de informações.

Agentes dos níveis superiores controlam os agentes dos níveis inferiores e a comunicação é feita usualmente de forma vertical entre os diversos níveis. Este tipo de modelo não explora a potencialidade de um sistema MAS e sim adapta os agentes a uma arquitetura pré-existente de uma organização “burocrática”.

Como conseqüência a flexibilidade é reduzida e em lugar de uma solução descentralizada e distribuída, encontra-se um sistema “monolítico” e centralizado. (PAOLUCCI e SACILE, 2005).

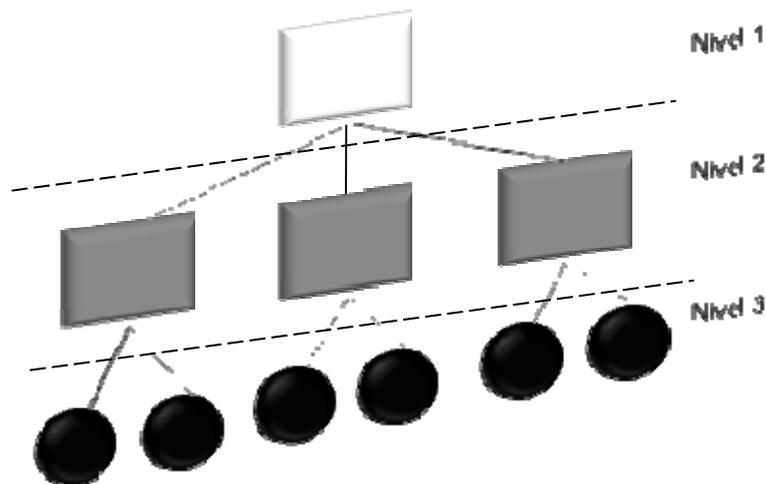


Figura 2.2 – Arquitetura hierárquica  
 Fonte: SHEN, 2002

Seu emprego mais freqüente se encontra no campo da programação distribuído assíncrono, em sistemas de gerenciamento logísticos, na arquitetura para agendamento da manufatura dinâmica e distribuída e em sistemas holônicos de manufatura. DE OLIVEIRA (2005) sintetiza as principais vantagens e desvantagens no uso da arquitetura hierárquica em sistemas baseados em agentes, tabela 2.2.

Tabela 2.2 – Vantagens e desvantagens do uso da arquitetura hierárquica em sistemas baseados em agentes  
 Fonte: DE OLIVEIRA, 2005

Vantagens	Desvantagens
Reduz a funcionalidade e complexidade de cada controlador individual; É passível de modificação através da inclusão de novos nós nos níveis já existentes; Trabalha próximo do regime ótimo em situações estáveis; Comportamento adaptativo. É possível obter feedback dos níveis inferiores e superior através de um loop fechado.	A introdução de novos níveis é difícil. Estrutura rígida. Dada a rigidez estrutural sua escalabilidade é dificultada e por vezes é impossível ampliar o sistema; Responde mal a situações de instabilidade, como quebra de recursos ou ordens imprevistas. Isto ocorre devido a demora dos altos níveis em obter informações e retornar medidas cabíveis, o que muitas vezes pode colocar todo o sistema em situação de risco; Sobrecarga de informação nos níveis intermediários devido a necessidade constante de atualização de informação vinda dos níveis inferiores nos controladores.

Alguns exemplos relevantes de sistemas de manufatura baseados em agentes que utilizam arquiteturas hierárquicas podem ser encontrados nos trabalhos de JONES e McLEAN (1986) AMRF – *Automated Manufacturing Research Facility (The NIST reference architecture)*; BUTLER e OHTSUBO (1992) ADDYMS: *Architecture for distributed dynamic manufacturing scheduling*; BURKE e PROSSER (1994) *The distributed asynchronous scheduler*; SOUSA e RAMOS (1997) *A Dynamic Scheduling Holon for*

*Manufacturing Orders*; BUSSMANN (1998) *An agent-oriented architecture for holonic manufacturing control*; e APARICIO et al. (1998) *NIIP-SMART: an investigation of distributed object approaches to support MES development and deployment in a virtual enterprise* e finalmente o trabalho de BAUER et al. (1991) *PAC – Production Activity Control* que foi aperfeiçoado por MAGLICA (1996) e chamado de PAC+. Posteriormente PAC+ foi aplicado em duas novas abordagens: *FACE Flexible Assembly Control Environment* (ONORI, 1996) e *MARK III FAS* desenvolvido por ONORI et al. (1997).

A próxima arquitetura a ser vista é chamada de arquitetura federada. Uma arquitetura federada é chamada de integral quando não há nenhum compartilhamento explícito para o armazenamento de dados ativos. Ao invés disto, o sistema armazena todos os dados em bancos de dados locais e lida com as atualizações e mudanças através da passagem de mensagens. (SHEN, 2002). Já PAOLUCCI e SACILE (2005) explicam que nesta forma de arquitetura os agentes são organizados em pequenas comunidades ou *clusters*, conforme suas funções, enquanto outros agentes fazem o papel de coordenadores de atividades. SHEN (2002) classifica os sistemas federados utilizados em manufatura em quatro abordagens predominantes:

*Facilitadores* – combinam diversos agentes relacionados em um grupo. A comunicação entre os agentes é feita por um facilitador. Cada facilitador provê a intermediação entre uma coleção local e remota de agentes, o que é feito através de dois serviços, um que especifica a saída de mensagens (*routing*) e outro que traduzem as mensagens que estão chegando para que possam ser utilizadas pelos agentes.

*Brokers* – são similares aos facilitadores, com funções adicionais como monitoramento e notificação. A diferença entre um facilitador e um *broker* é que o facilitador é responsável apenas por um grupo de agentes designado, enquanto que qualquer agente pode contatar qualquer *broker* que esteja no mesmo sistema a fim de encontrar agentes de serviço para completa uma tarefa específica.

*Matchmakers*<sup>1</sup> - é uma *superset* da arquitetura *broker*, pois utiliza o mecanismo *broker* para combinar diversos agentes.

*Mediadores* – além de funcionar como um facilitador e um *matchmaker*, o mediador assume o papel de sistema coordenador promovendo a cooperação entre agentes inteligentes e aprendendo com o comportamento dele. Esta arquitetura impõe uma hierarquia estática ou dinâmica para cada tarefa específica, o que provê, computacionalmente, simplicidade e maleabilidade. É uma arquitetura aplicável para o

---

<sup>1</sup> A FIPA - *The Foundation for Intelligent Physical Agents* tem proposto o uso de agentes de página amarela e de um diretório facilitador, que são similares aos *matchmakers*.

---

desenvolvimento de sistemas de agendamento em manufaturas distribuídas, que são ao mesmo tempo complexos e dinâmicos e que incluem muitos agentes de recursos.

Vários autores (GENESERETH e KETCHPEL, 1994; HAYDEN et al., 1999; HORLING e LESSER, 2005 e SHEN, 2006) explicam que arquiteturas federadas possuem como principal vantagem e desvantagem a coordenação de atividades via intermediadores. Isto ocorre porque o nível intermediário, aqui representado pelos facilitadores, *brokes* ou *matchmakers*, pode exercer diferentes funcionalidades, como por exemplo, tradutor, alocação de tarefas, ou monitoramento, entre outras. Isto faz com que todos os agentes possam usufruir de tais capacidades sem necessariamente ter o conhecimento necessário para executá-las, ou mesmo de ter a necessidade de monitorar os agentes que as executam. Porém, ao mesmo tempo pode gerar sobrecargas de informações. Apesar disto, este tipo de arquitetura costuma garantir estabilidade ao mesmo tempo em que provê escalabilidade. A figura 2.3 apresenta três arquiteturas federadas genéricas, mostrando a diferença entre as abordagens com facilitadores, *brokes e matchmakers*, para o desenvolvimento de sistemas baseados em agentes para a programação da produção.

Alguns exemplos relevantes de sistemas de manufatura baseados em agentes que utilizam arquiteturas hierárquicas podem ser encontrados nos trabalhos de McGUIRE et al. (1993) SHADE; PENG et al. (1998) CIIMPLEX; OUELHADJ (1998); MATURANA et al. (1999) MetaMorph e SHEN et al. (2000) MetaMorphII.

O terceiro tipo de arquitetura é chamado de autônoma ou heterárquica, cujo principal desenvolvedor é Neil Duffie (DUFFIE, 1990; DUFFIE, CHITTURI e MOU, 1988) baseado no trabalho de József Hatvany (HATVANY, 1985). Sua principal característica é a falta de um controle central, a completa ausência de hierarquias e a natureza autônoma e cooperativa de cada nó que compõe o sistema, os quais tipicamente representam os recursos ou tarefas de um chão-de-fábrica (DE OLIVEIRA, 2005). SHEN et al. (2006) argumenta que agentes autônomos devem ao menos possuírem as seguintes características:

Não possuírem gerenciamento ou controle a partir de nenhum outro agente ou humano;

Possuírem a habilidade de se comunicarem e interagirem diretamente com qualquer outro agente no sistema ou mesmo com sistemas externos;

Possuírem conhecimento a respeito dos outros agentes e do seu ambiente; e

Possuírem seus próprios objetivos e associados a estes, um grupo de motivações.

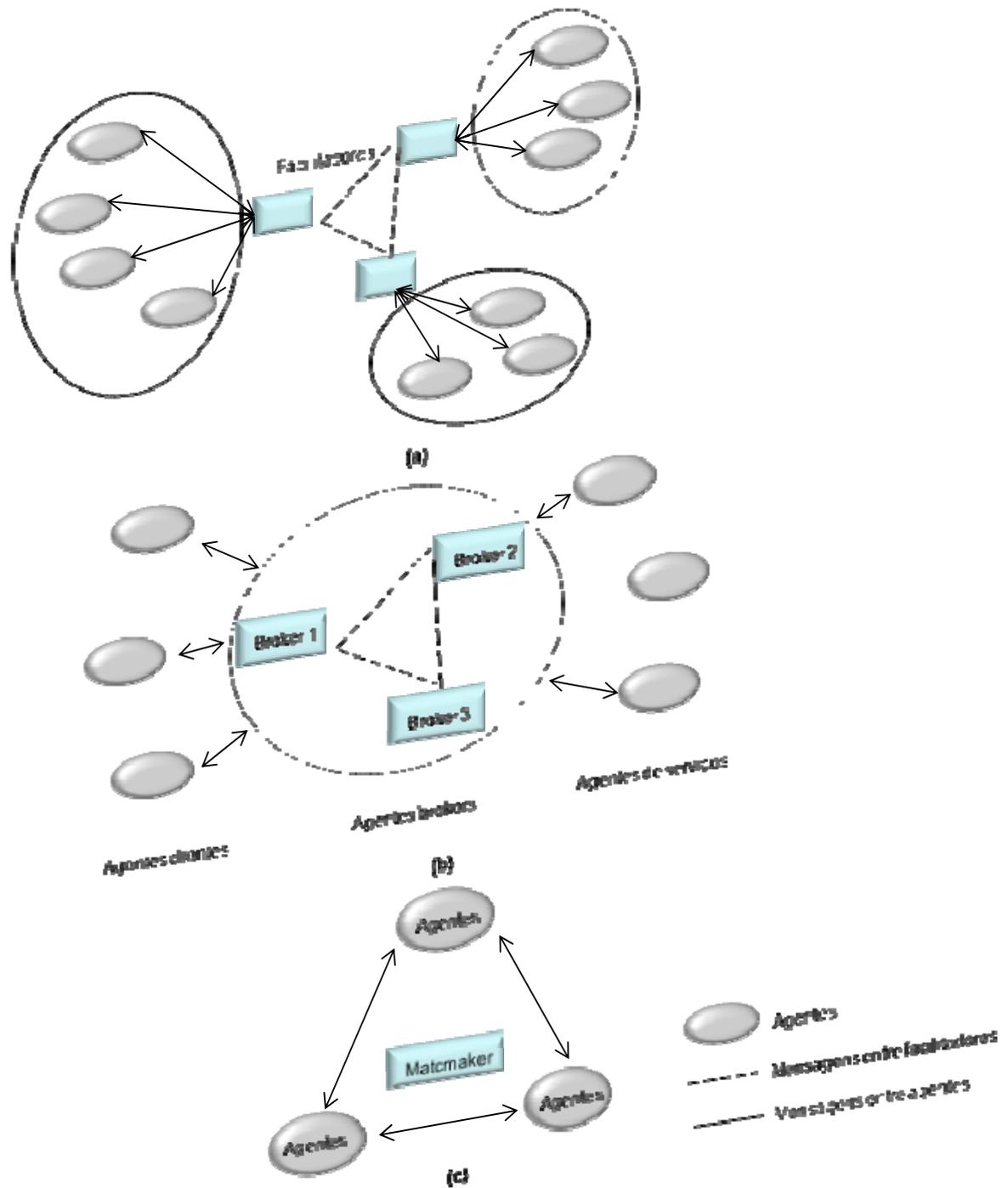


Figura 2.3 – Arquitetura federada com diferentes abordagens: (a) facilitadores, (b) brokers e (c) matchmakers  
 Fonte: SHEN, 2002

DE OLIVEIRA (2005) sintetiza as principais vantagens e desvantagens no uso da arquitetura hierárquica em sistemas baseados em agentes, tabela 2.3.

Tabela 2.3 – Vantagens e desvantagens do uso da arquitetura heterárquica em sistemas baseados em agentes

Fonte: DE OLIVEIRA, 2005

Vantagens	Desvantagens
<p>A complexidade geral de programação é reduzida através da eliminação de níveis hierárquicos;</p> <p>Reage mais rapidamente a situações fora do normal;</p> <p>Melhor tolerância a falhas. Isto ocorre pois a falha em um nó não é propagada, não tendo impacto na estratégia de controle;</p> <p>Reduz tempo e custo de desenvolvimento através da ausência de níveis supervisórios;</p> <p>Maior escalabilidade e modularidade.</p>	<p>Problemas com otimização. A independência dos nós resulta em alguma redundância, o que dificulta a melhoria da performance sem dispor de uma informação global;</p> <p>Pode ocorrer comportamento caótico. Isto ocorre devido à falta de uma entidade central de supervisão;</p> <p>A performance do sistema depende diretamente do quão bem as regras de negociação foram definidas, tendo impacto direto sobre o balanceamento do carregamento nos recursos, devido a concorrência direta entre os mesmos para a alocação de suas tarefas.</p>

O conhecimento e o controle, neste tipo de arquitetura, são distribuídos, não obstante, decisões de controle e objetivos além do escopo de cada agente, podem ser alcançadas através de negociação entre os agentes participantes, representada na figura 2.4.

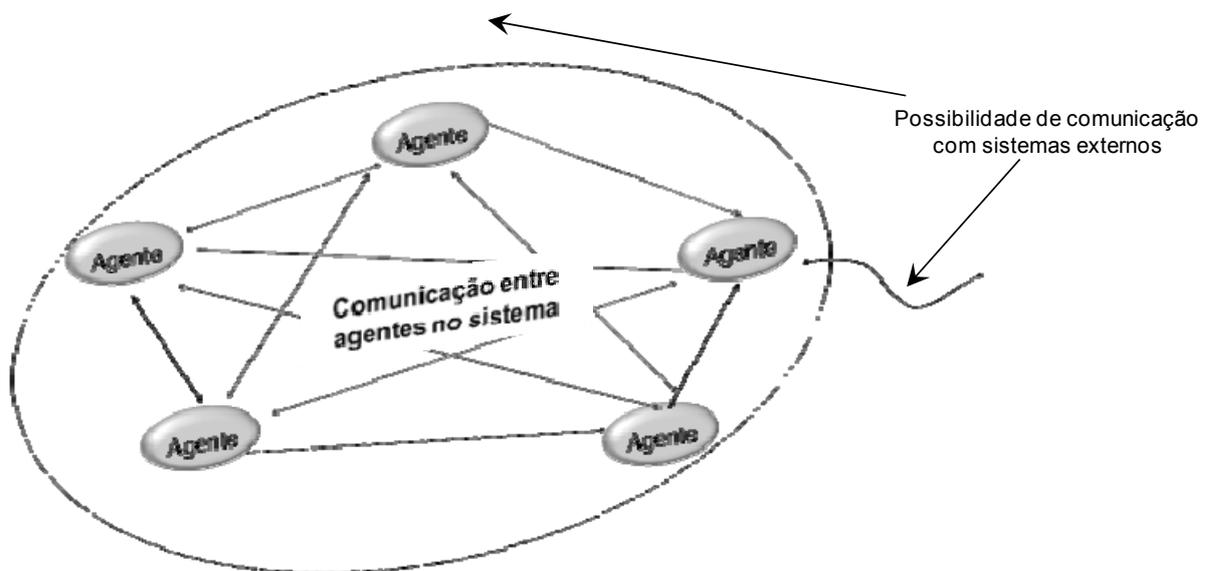


Figura 2.4 – Arquitetura heterárquica

Ao contrário das demais arquiteturas, há poucas implementações utilizando a abordagem heterárquica, sendo que sua maior contribuição está nos conceitos que foram introduzidos por DUFFIE (1990) e posteriormente empregados em aplicações industriais de abordagens multiagentes. Exemplos podem ser encontrados nos trabalhos de PARUNAK, BAKER e CLARK (1998) AARIA – *Agent Architecture*, que empregou o conceito de agentes autônomos e protocolos de negociação fixos; SHEN, MATURANA e NORRIE (2000),

---

MethaMorphII, desenvolvido através de uma combinação entre as abordagens federadas e heterárquica.

### 2.3.5 Coordenação e cooperação

Visando resultados coerentes, e independentemente de sua arquitetura, os agentes que compõem um sistema precisam ter habilidades como coordenação e comunicação. De acordo com FARATIN et. al. (1998) agentes necessitam interagir com outros agentes para alcançar seus objetivos, isto ocorre porque não possuem capacidades ou recursos suficientes para resolver sozinhos seus problemas. Outra razão para a necessidade de interação é a interdependência entre agentes situados em um ambiente em comum. JENNINGS et. al. (2001) esclarece que estas interações entre agentes podem variar deste simples trocas de informações até pedidos para uma ação particular a ser executada, o faz com que a coordenação seja um dos mais importantes aspectos em um sistema multiagentes, em especial naqueles onde metas e interesses sejam compartilhados entre todos os agentes que os compõem. SHEN et al. (2006) enfatiza ainda que sem coordenação um grupo de agentes pode rapidamente degenerar em uma caótica coleção de indivíduos.

DE OLIVEIRA (2005) explica que, a fim de coordenar suas atividades, os agentes devem seguir uma série de compromissos, os quais podem estar representados por: um grupo de regras fixas ou emergentes, por obrigações sociais ou mesmo por estilos de conversação. A figura 2.5 sumariza uma série de compromissos, cada qual correspondente a uma diferente abordagem de coordenação.

As duas principais abordagens de coordenação entre agentes são a cooperação e competição, apresentadas no segundo nível da figura 2.5. Cooperação é a coordenação entre agentes que compartilham as mesmas metas enquanto competição é a coordenação entre agentes com interesses próprios (HUHNS e STEPHENS, 1999).

Antes de dar continuidade a discussão sobre os mecanismos de coordenação entre agentes se faz necessário pontuar as seguintes considerações sob o ponto de vista adotado no presente trabalho:

a) Tendo em vista o escopo de atuação deste trabalho um modelo de simulação para a área de manufatura, a ser descrito na declaração do problema item 4.1.1, entendeu-se que uma revisão completa sobre a taxonomia da coordenação apresentada na figura 2.5 não faz parte de seu conteúdo, sendo que uma ampla discussão sob o tema pode ser encontrada em DE OLIVEIRA (2005); e

b) Considerando-se que a coordenação entre agentes através da competição tem sua maior aplicabilidade no domínio do comércio eletrônico (DE OLIVEIRA, 2005; PADGET et al., 2002), a mesma não será discutida no presente trabalho.

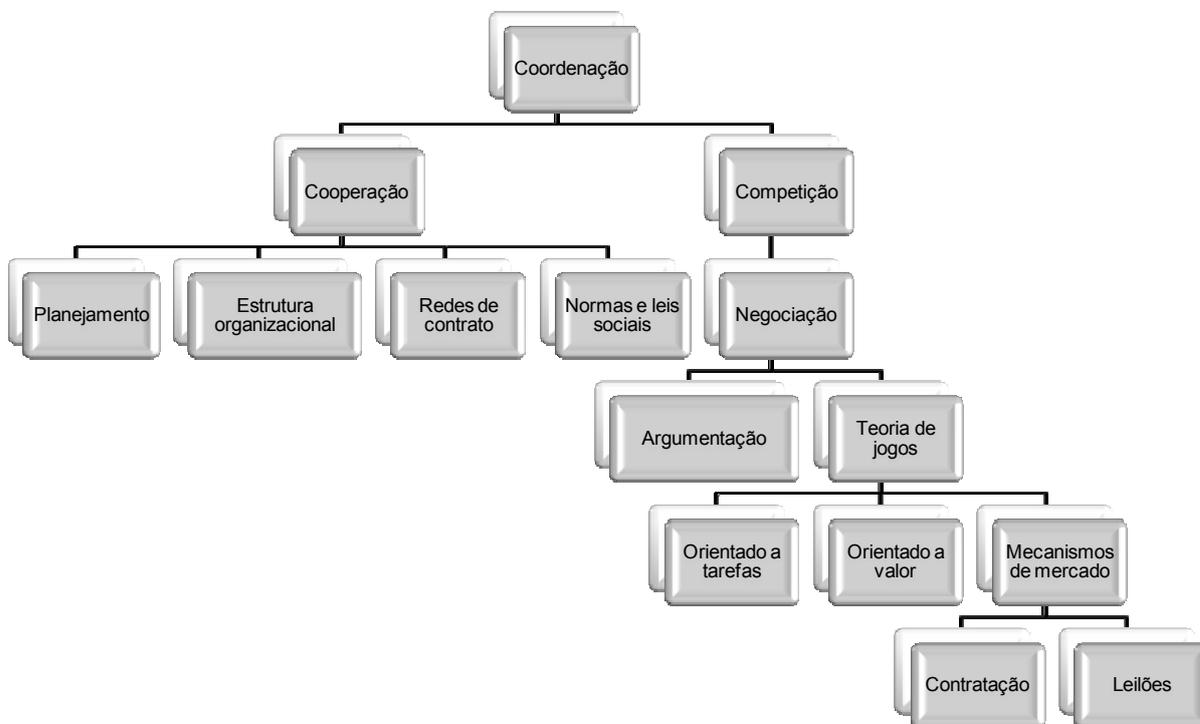


Figura 2.5 – Taxonomia da coordenação  
 Fonte: DE OLIVEIRA, 2005

A cooperação entre agentes consiste de duas fases distintas, porém complementares. Na primeira fase a tarefa ou objetivo é decomposto em sub-tarefas, que na fase seguinte são distribuídas entre as entidades participantes, esta fase também é conhecida como alocação ou distribuição de tarefas. A distribuição de tarefas, segundo OUELHADJ (2003), envolve a definição de mecanismos organizações que diz que agentes podem combinar suas habilidades para executar trabalhos coletivos.

As possíveis estratégias a alocação de tarefas são representadas no terceiro nível da figura 2.5, sendo que neste trabalho será utilizada uma variante das redes de contrato - *Contract-Net Protocol*, cuja discussão teórica será feita a seguir no item 2.3.6 do presente capítulo.

### 2.3.6 Redes de contrato - *Contract-Net Protocol*

De acordo com vários autores (WOOLDRIDGE, 2002; SHEN, 2002 e SHEN et al., 2006) as redes de contrato – *Contract-Net Protocol* (CNP), em suas mais variadas versões, são o mecanismo de coordenação mais frequentemente encontrados nos sistemas de manufatura baseados em agentes, em particular naquelas que tratam da alocação de recursos. Exemplos podem ser encontrados nos trabalhos de DUFFIE (1988); PARUNAK et al. (1998); OW e SMITH (1980); SHAW (1988); BUTLER e OHTSUBO (1992); OUELHADJ et al. (1998); WONG et al. (2006a); entre outros. *Contract-Net Protocol* é um protocolo de alto nível desenvolvido por SMITH (1980) baseado na metáfora de contratação de uma empresa por outra durante o desenvolvimento de determinados serviços.

BUSSMANN et al. (2004) explica que apesar de serem baseadas normalmente em um protocolo simples, as redes de contrato costumam ser eficientes para a designação de tarefas para nós individuais de uma rede, isto ocorre, pois é assumido que cada nó possui uma tarefa que precisa ser executada (por outro nó) e que existem inúmeros nós (potenciais) que podem executar esta tarefa. A figura 2.6 apresenta as três etapas que compõem uma negociação em uma rede de contrato.

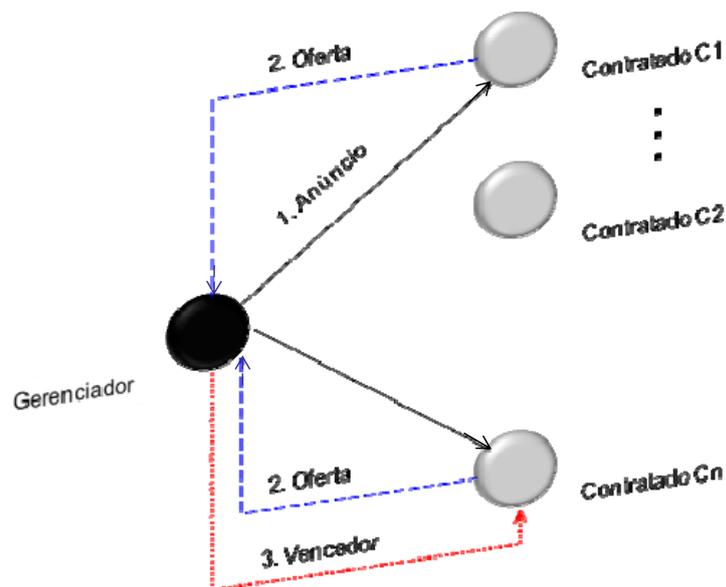


Figura 2.6 – Etapas de negociação em uma rede de contrato  
Fonte: BUSSMANN, 2005

O nó que possui a tarefa a ser executada é chamado de gerenciador e os outros nós de proponente/contratado. O gerenciador inicia o protocolo com o anúncio da tarefa aos potenciais contratados. Os contratados (proponentes) respondem com uma oferta. A

informação necessária a ser provida pela oferta é especificada pelo gerenciador quando do anúncio. O gerenciador compara as ofertas e escolhe a melhor, isto será feito de acordo com as suas preferências. O nó vencedor recebe uma comunicação de vitória (contratação para a execução da tarefa). Os outros nós podem ou não receber uma mensagem de rejeição de oferta.

A Federação para Agentes Físicos Inteligentes – FIPA<sup>2</sup>, em seu documento SC00029H de março de 2002, apresentou uma padronização adaptada do protocolo proposto por SMITH (1980), o qual chamou de Protocolo de Interação de Redes de Negociação FIPA (do inglês *FIPA Contract Net Interaction Protocol*), apresentado na figura 2.7.

O agente iniciador solicita m propostas dos agentes participantes emitindo uma ação de chamada de propostas (do inglês *call for proposals* – cfp<sup>3</sup>), onde são especificadas as tarefas, assim como qualquer condição que o iniciador estiver colocando para a execução das mesmas.

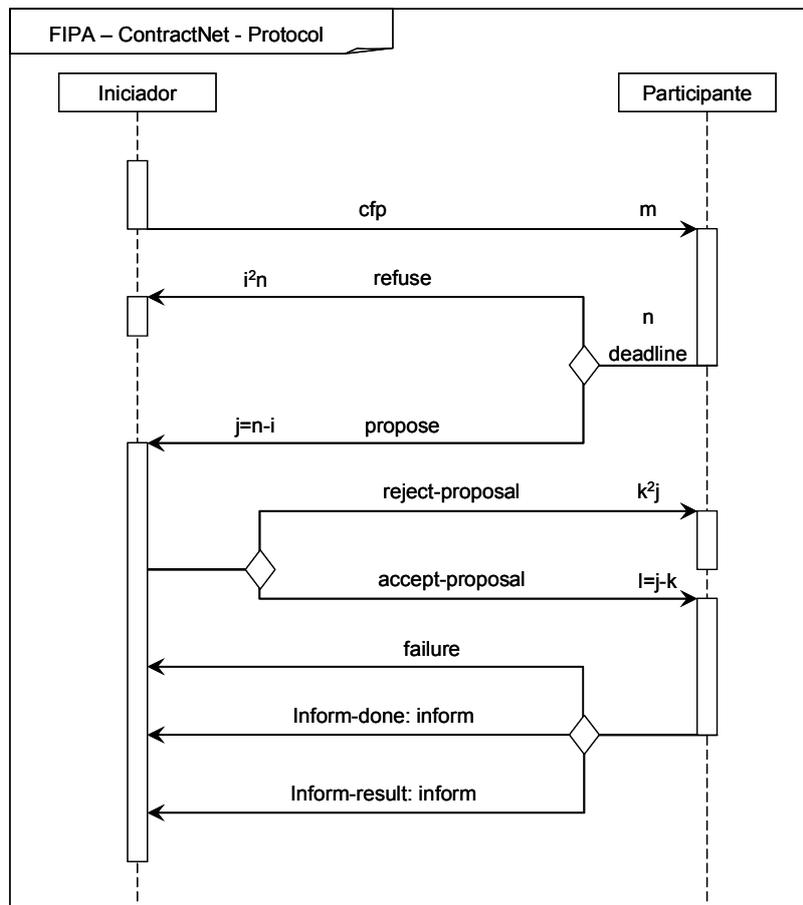


Figura 2.7 – Etapas de negociação em uma rede de contrato  
 Fonte: BUSSMANN, 2005

<sup>2</sup> FIPA - cuja existência se deve pela necessidade de padronização em aplicações baseadas em agentes, a ser discutida no item 2.3.7 do presente capítulo.

<sup>3</sup> As ações performativas *CFP*, *proposal*, *refuse*, *propose*, *accept-proposal* e *reject-proposal*, são padronizadas no documento FIPA00037.

Os agentes que receberem este *cfp* são vistos como potenciais contratados e estarão aptos a gerar  $n$  respostas. Destas  $n$  respostas,  $j$  serão ações de propostas (do inglês *proposal*) para realizar a tarefa solicitada. As propostas enviadas pelos agentes participantes incluem as pré-condições para a realização da tarefa, que podem ser um preço, um tempo, ou qualquer outra especificação solicitada. Alternativamente às propostas, existirão  $i = n-j$  participantes que poderão recusar (do inglês *refuse*) o *cfp*. A partir do momento que o tempo de resposta se encerra, o agente iniciador analisa as  $j$  propostas recebidas e a partir delas seleciona os agentes aptos para executar a tarefa, que podem ser um único agente, vários ou nenhum. Os  $l$  agentes escolhidos recebem uma mensagem de aceite (do inglês *accept-proposal*) e os  $k$  restantes, uma mensagem de não aceite (do inglês *reject-proposal*). Ao final da execução da tarefa, o contratado envia uma mensagem ao iniciador informando o *status* da solicitação, que pode ser sua finalização (do inglês *inform-done*), ou em uma versão mais elaborada são informados os resultados (do inglês *inform-result*), ou ainda uma falha (do inglês *failure*) ao completar a tarefa.

Inúmeras variações de redes de contrato podem ser encontradas na literatura, sendo que SHEN e NORRIE (1999) citam entre outros o mecanismo de voto, sua versão modificada o *Extended Contract Net Protocol* (ECNP), o *Market-Driven Contract Net*, o *B-Contract-Net*, entre outros. Ainda podem ser citados os mecanismos de ofertas orientadas às peças (DUFFIE, 1988; OW e SMITH, 1988; LIN e SOLBERG, 1992), recursos (BAKER, 1998; BUTLER e OHTSUBO, 1992, SHEN et al., 2001; BRUCCOLERI, AMICO e PERRONE, 2003) ou bi-direcionais (SAAD, 1995).

SHEN (2002) e OUELHADJ (2003) citam como principais vantagens no uso de redes de contrato:

- Alocação dinâmica de tarefas;
- Balanceamento natural de carregamento;
- Introdução ou remoção dinâmica de agentes; e
- Grande confiabilidade à falhas.

Apesar de eficientes, as redes de contrato podem apresentar problemas quando o número de agentes a serem contatados é muito grande. O tempo de atravessamento das mensagens no sistema tende a aumentar demais e com isto ocasionar efeitos indesejáveis, como congestionamento de rede e desperdício de tempo de CPU.

Para evitar estes problemas, SHEN (2002) explica que os pesquisadores têm estendido o modelo CNP tradicional com inúmeras melhorias, dentre elas:

---

Envio de mensagens de ofertas a um número limitado de nós, ao invés de para todos os nós que compõem a rede;

Antecipação de ofertas, isto é, os contratados enviam ofertas adiantadas;

Variação do tempo quando o compromisso é assumido;

Permissão de quebra de compromisso;

Permissão de vários agentes responderem como um grupo (coalizão); e

Introdução de prioridades para a resolução de tarefas.

As alterações sobre o modelo padrão de rede de contrato utilizadas no protótipo desenvolvido no presente trabalho, cujo modelo conceitual e implementação são discutidas nos itens 5.5.1 e 5.5.1, dizem respeito à ampliação do tempo de negociação entre agentes. A apresentação e discussão destas alterações estão no capítulo 7 do presente trabalho, que trata da formulação da proposta de negociação.

### 2.3.7 Padronização

O mais significativo esforço de padronização em aplicações voltadas para agentes e multiagentes vem sendo proposto e desenvolvido nos últimos anos pela FIPA - *Foundaton for Intelligent Physical Agents*.

A FIPA é uma organização não lucrativa estabelecida em agosto de 1996 e registrada em Genebra na Suíça (disponível em [www.fipa.org](http://www.fipa.org)). Seu propósito é promover o desenvolvimento de especificações de tecnologias genéricas de agentes que maximizem a interoperabilidade entre e por aplicações baseadas em agentes. A FIPA tem proposto e avaliado inúmeras padronizações e especificações, entre as quais podem ser citadas *benchmarking*, comunicação, entrada e saída de dados, representação de informações, privacidade e segurança, sociedades de agentes, ferramentas e arquiteturas, informação pessoal e personalizada, ambientes de execução para agentes móveis e gerenciamento de agentes. Em 2002, a FIPA completou um processo de padronização de um sub-grupo de todas as especificações, em um total de vinte e cinco, apresentadas na tabela 2.4.

Tabela 2.4 – Grupo corrente de especificações padronizadas pela FIPA  
 Fonte: <http://www.fipa.org/repository/standardspecs.html>; 31/07/2007

Identificador	Título
SC00001	FIPA Abstract Architecture Specification
SC00008	FIPA SL Content Language Specification
SI00014	FIPA Nomadic Application Support Specification
SC00023	FIPA Agent Management Specification
SC00026	FIPA Request Interaction Protocol Specification
SC00027	FIPA Query Interaction Protocol Specification
SC00028	FIPA Request When Interaction Protocol Specification
SC00029	FIPA Contract Net Interaction Protocol Specification
SC00030	FIPA Iterated Contract Net Interaction Protocol Specification
SC00033	FIPA Brokering Interaction Protocol Specification
SC00034	FIPA Recruiting Interaction Protocol Specification
SC00035	FIPA Subscribe Interaction Protocol Specification
SC00036	FIPA Propose Interaction Protocol Specification
SC00037	FIPA Communicative Act Library Specification
SC00061	FIPA ACL Message Structure Specification
SC00067	FIPA Agent Message Transport Service Specification
SC00069	FIPA ACL Message Representation in Bit-Efficient Specification
SC00070	FIPA ACL Message Representation in String Specification
SC00071	FIPA ACL Message Representation in XML Specification
SC00075	FIPA Agent Message Transport Protocol for IIOP Specification
SC00084	FIPA Agent Message Transport Protocol for HTTP Specification
SC00085	FIPA Agent Message Transport Envelope Representation in XML Specification
SC00088	FIPA Agent Message Transport Envelope Representation in Bit Efficient
SI00091	FIPA Device Ontology Specification
SC00094	FIPA Quality of Service Specification

Em 8 de junho de 2005, a FIPA foi oficialmente aceita pela IEEE - *Institute of Electrical and Electronics Engineer*. Entre suas especificações uma das mais relevantes a este trabalho, pode ser citada a SC00023 *FIPA Agent Management Specification*, de dezembro de 2002 e atualizada em março de 2004, que contém as especificações para o gerenciamento de agentes, incluindo o gerenciamento de serviços de agentes, ontologia e transporte de mensagens de agentes na plataforma.

### 2.3.8 Linguagens de comunicação entre agentes

Ao representar uma idéia em um dado escopo, é necessária uma linguagem que não permita diferentes interpretações e que expresse claramente o significado do que foi dito.

Linguagens especializadas têm sido desenvolvidas em campos específicos, como a STEP – *Standard for the Exchange of Product Model Data* STEP que é um padrão internacional para a integração, apresentação e o intercâmbio de dados de produtos industriais, via computador e a SGML - *Standard Generalized Markup Language* para a descrição da estrutura lógica de um documento.

De forma análoga, agentes em negociação necessitam de uma linguagem que suporte mais do que a simples troca de mensagens e, para este fim, foram criadas as linguagens de comunicação entre agentes, - ACLs (do inglês *Agent Communication Languages*), que permitem a troca de informações e conhecimento entre agentes. Ou seja, as ACLs permitem a troca de proposições, regras e ações, utilizando nos conteúdos uma ontologia específica ao domínio e expressos através do uso de um idioma escolhido.

A maioria das ACLs é baseada na teoria de dizer-agir (do inglês *speech-act theory*) proposta primeiramente por John Langshaw Austin (AUSTIN, 1955), que de forma muito resumida pode ser descrita pela expressão “dizendo algo, nós fazemos algo”. Por exemplo, um padre ao dizer “eu os declaro marido e mulher” esta realmente unindo duas pessoas em matrimônio.

Os atos da fala podem ser expressos através de palavras padronizadas, conhecidas como performativas (perguntar, dizer, requisitar), e podem mudar o estado interno de um agente. Segundo WOOLDRIDGE (2002) as categorias mais importantes de atos da fala (comunicativos) foram definidas por SEARLE (1969) e são:

Representativos - um ato representativo remete o orador à verdade de uma proposição expressa. Representa o paradigma informativo (*inform*);

Diretivos – é uma tentativa de uma das partes (a que fala) de ser ouvida pela parte que pode executar uma determinada ação. Representa o paradigma requisição (*request*);

Comprometimentos – remete o orador a um curso de ação. Representa o paradigma de promessa (*promising*);

Expressivos – expressa um estado psicológico, por exemplo, de gratidão; e

Declarativos – produzem mudanças no estado institucional das relações. Representa o paradigma declarando uma decisão ou atitude.

Conforme DE OLIVEIRA (2005) as linguagens de comunicação entre agentes ainda devem ser consideradas em três níveis, tanto sob a ótica da comunicação, quanto do compartilhamento de conhecimento, são eles: o nível de sintaxe, que descreve como os modelos internos de conhecimento são organizados e descritos. O nível de semântica, que indica como os conceitos deverão ser interpretados, e finalmente o nível de comunicação, que especifica a predisposição de um agente se comunicar com outros agentes a fim de poder trocar informações dentro de um contexto.

Neste trabalho serão apresentadas as três linguagens mais utilizadas para a padronização de sistemas orientados a agentes. Especial atenção será dada à padronização FIPA - ACL, adotada no desenvolvimento da arquitetura proposta.

---

KIF – (do inglês *Knowledge Interchange Formation*), é uma linguagem orientada a computadores para a troca de conhecimento entre programas discrepantes (disponível em <http://www.cs.umbc.edu/kse/kif/>). Possui declarações semânticas o que permite, por exemplo, entender o significado de expressões na representação sem a necessidade de um interpretador para sua manipulação. Entre outras características está a definição de objetos, funções e relações. É possível a representação de propriedades específicas, de seus relacionamentos e propriedades gerais do domínio. Em 1993 foi proposto um *draft* da *American National Standard for KIF* e em 1998 foi proposta uma primeira revisão.

KQML – (do inglês *Knowledge Query and Manipulation Language*), composta por uma linguagem e uma série de protocolos para a troca de informação e conhecimento (disponível em <http://www.cs.umbc.edu/kqml/>). Faz parte dos esforços da DARPA *Knowledge Sharing Effort*, que busca o desenvolvimento de técnicas e metodologias para a construção de grandes bases de conhecimento, que possam ser compartilhadas e que permitam reutilização. KQML é ao mesmo tempo um formato de mensagem e um protocolo de troca de mensagem e pode ser usado como uma linguagem para um aplicativo de programa, a fim de interagir com um sistema inteligente. Também permite que dois ou mais sistemas inteligentes compartilhem conhecimento para a solução cooperativa de um problema. Além disto, KQML também provê uma arquitetura básica para compartilhar conhecimento através de uma classe especial de agentes, chamada de facilitadores de comunicação, os quais coordenam as interações entre outros agentes. Em 1993 foi criada uma especificação de linguagem KQML para a comunicação entre agentes, a qual foi revista em 1996.

FIPA - ACL – conjunto de tipos de mensagens e descrições dos efeitos da mensagem sobre os agentes que a enviam e sobre os que a recebem. Possui uma semântica definida precisamente, de forma análoga a uma linguagem de descrição semântica. Foi desenvolvida e é mantida pela FIPA. A tabela 2.5 apresenta vários parâmetros utilizados nas mensagens FIPA-ACL.

O campo parâmetro, apresentado na primeira coluna da tabela, representa qual o tipo de protocolo de interação que o agente que envia a mensagem irá empregar, estabelecendo desta forma o teor da conversação entre os agentes. Na segunda coluna é apresentada a categoria de parâmetros, que é definido pela FIPA como um padrão típico para troca de mensagens. Finalmente na terceira coluna é feita uma breve descrição para a utilização de cada protocolo.

Tabela 2.5 – Parâmetros de mensagem em FIPA –ACL

Fonte: FIPA SC00037J, 2002

<i>Parâmetros</i>	<i>Categoria dos parâmetros</i>	<i>Descrição</i>
<i>performative</i>	Tipo de ato de comunicação	Denota o tipo de ato de comunicação ( <i>Accept Proposal</i> , <i>Cancel</i> , <i>Call for Proposal</i> ) <sup>4</sup>
<i>sender</i>	Participante da conversação	Identifica o remetente da mensagem, ou seja, o nome do agente no ato de comunicação.
<i>receiver</i>	Participante da conversação	Identifica a identidade dos recebedores da mensagem.
<i>reply-to</i>	Participante da conversação	Indica que a mensagem subsequente na conversação será direcionada ao agente nominado no parâmetro <i>reply-to</i> , e não ao agente nominado no parâmetro <i>sender</i> .
<i>content</i>	Conteúdo de mensagem	Denota o conteúdo da mensagem (o objeto da ação). Deve ser possível de ser interpretado pelo agente que a recebe, o que deve ser observado com cuidado, pois as interpretações podem divergir entre o remetente e o recebedor.
<i>language</i>	Descrição de conteúdo	Linguagem na qual o conteúdo da mensagem é expresso.
<i>encoding</i>	Descrição de conteúdo	Identifica a codificação do conteúdo da mensagem.
<i>ontology</i>	Descrição de conteúdo	Ontologia usada a fim de dar um significado aos símbolos nas expressões de conteúdo.
<i>protocol</i>	Controle de conversação	Protocolo de interação que o remetente emprega nas mensagens ACL.
<i>conversation-id</i>	Controle de conversação	Introduz uma expressão, a qual é usada para identificar a sucessão contínua de atos comunicativos que juntos formam uma conversação.
<i>reply-with</i>	Controle de conversação	Introduz uma expressão que será usada pelo agente que estará respondendo para identificar sua mensagem.
<i>in-reply-to</i>	Controle de conversação	Denota uma expressão que referencia uma ação prévia da qual esta mensagem é a resposta.
<i>reply-by</i>	Controle de conversação	Denota uma expressão de tempo e/ou de data que indica o prazo máximo em que o remetente gostaria de receber uma resposta.

O conteúdo de uma mensagem em FIPA –ACL pode ser composto por um ou mais parâmetros apresentados, porém os parâmetros considerados necessários para que um agente possa estabelecer uma comunicação é variável de acordo com a situação, sendo que o único parâmetro obrigatório é o *performative*. É preciso notar, porém que a maioria das mensagens em FIPA –ACL também contêm os parâmetros *sender*, *receiver* e *content* (FIPA SC00037J, 2002).

Sua grande vantagem, quando comparada a linguagem KQML, reside em uma representação semântica formal mais definida (DE OLIVEIRA, 2005). Como características comuns a ambas as linguagens podem ser citadas:

<sup>4</sup> Uma lista completa dos valores reservados de atos de comunicação pela FIPA –ACL pode ser encontrada no documento SC00037J - FIPA *Communicative Act Library Specification*, disponível em <http://www.fipa.org>.

---

Baseadas na teoria *speech act* (op. cit.);

Utilizam os mesmos níveis conceituais: conteúdo, comunicação e mensagem;

São independentes do mecanismo de transporte, do conteúdo da linguagem e da ontologia utilizada.

A FIPA-ACL ainda provê especificações para as linguagens KIF, RDF – *Resource Description Framework*, CCL – *Constraint Choice Language* e SL – *Semantic Language*.

### 2.3.9 Frameworks para o desenvolvimento de sistemas baseados em agentes

De modo análogo a outros sistemas, os sistemas multiagentes têm sido implementados em linguagens diversas como C++, Java, Lisp, Prolog e outras. Estas diversas linguagens funcionam como a base para os diversos *frameworks* de desenvolvimento de sistemas baseados em agentes. Um *framework* captura a funcionalidade comum a várias aplicações, que devem ter algo significativo em comum, como pertencer a um mesmo domínio de problema. Ou seja, um *framework* provê uma solução a uma família de problemas semelhantes. Entre os *frameworks* código aberto (*open source*) ou não, podem ser citados:

a) JADEX – desenvolvido pelo Grupo de Sistemas Distribuídos e de Informação da Universidade de Hamburgo (disponível em <http://vsis-www.informatik.uni-hamburg.de/projects/JADEx/>). É um *framework* baseado em Java, construído sobre a plataforma JADE, de acordo com as especificações da FIPA, que permite o desenvolvimento de agentes orientados à meta (modelo BDI- *Belief, Desire e Intention*) e que provê um grupo de ferramentas para o desenvolvimento, criação e testes de agentes. O principal objetivo do projeto de pesquisa JADEX é construir sobre a estrutura FIPA um nível racional de agentes, que permita a construção de agentes inteligentes. Para que isto ocorra é necessário prover uma arquitetura de agente que considere a estrutura interna do agente, a sociedade de agentes e os conceitos de IA.

b) FIPAS-OS – desenvolvido originalmente nos laboratórios da *Nortel Networks' Harlow Laboratories* na Inglaterra, trabalha com código aberto e é aderente à padronização FIPA, (disponível em <http://www.nortelnetworks.com/products/announcements/fipa/>). FIPAS-OS foi um dos primeiros *frameworks* com arquitetura aberta e aderente à padronização FIPA, sendo na época de sua criação uma tentativa de reduzir as barreiras de adoção da padronização FIPA. É implementado em JAVA e utiliza uma abordagem orientada a tarefas na estrutura interna de agentes. Provê uma arquitetura baseada em componentes que permite o

---

desenvolvimento de agentes para domínios específicos, além de suporte para os conceitos de interoperabilidade da FIPA.

c) Zeus – desenvolvida pela *British Telecom*, em JAVA, consiste de quatro componentes: uma biblioteca de agentes, ferramentas de construção de agentes, uma metodologia de desenvolvimento e ferramentas de visualização. Apesar de a plataforma ZEUS possuir um servidor para nomear agentes (ANS – *Agent Name Server*) e um diretório facilitador (DF), é somente parcialmente aderente à padronização FIPA. Cada agente da plataforma ZEUS consiste de um nível de definição, que inclui as habilidades de raciocínio e aprendizado do agente, além de suas habilidades, metas, recursos, crenças, preferências, etc. Um nível organizacional que descreve o relacionamento do agente com outros agentes e um nível de coordenação, onde o agente é modelado como uma entidade social, em termos de técnicas de coordenação e negociação.

Dentre os diversos *frameworks* disponíveis para o desenvolvimento de um sistema baseado em agentes, optou-se para o neste trabalho pela plataforma JADE - *Java Agent DEvelopment framework*, a ser descrita a seguir no presente capítulo. Sua escolha deveu-se primordialmente aos seguintes fatores:

- a) Esta plataforma foi totalmente e em total conformidade com as especificações da FIPA; e
- b) A plataforma provê uma série de funcionalidades a fim de simplificar o desenvolvimento de sistemas multiagentes, além de apresentar muito poucas restrições ao uso de seu código, *open source*. Não impondo a necessidade do uso de nenhuma arquitetura em particular, ou seja, o usuário usa e escreve em código JAVA sem o uso de nenhuma construção especial.

#### 2.3.10 JADE – Java Agent DEvelopment framework

*JADE* é um ambiente *open source* para aplicações baseadas em agentes desenvolvida em JAVA pela Itália Telecom em conjunto com a Universidade de Parma, (disponível em <http://JADE.tilab.com/>). Segundo BELLIFEMINE, et al. (2006), como esta plataforma foi desenvolvida totalmente aderente à padronização FIPA, isto lhe garante características muito interessantes, como:

- a) Sua arquitetura é baseada na coexistência de várias JVM - *JavaVirtual Machine* e pode ser distribuída entre vários *hosts* (máquinas), independente do sistema operacional que cada uma utiliza, sendo que apenas uma JVM é executada em cada *host*;
- b) Permite a implementação de aplicações multi-domínio;
- c) Oferece protocolo IIOP<sup>5</sup> para conectar diferentes plataformas de agentes;
- d) Mecanismo de transporte camaleão – definido de acordo com a situação;
- e) Suporte para que os agentes executem tarefas múltiplas, paralelas ou concorrentes, através de modelos de comportamentos (*behaviours*);
- f) A comunicação entre JVM é feita através de RMI – *Remote Method Invocation*, advinda do JAVA. A figura 2.8 apresenta a arquitetura distribuída da plataforma JADE dividida em três *hosts*, e em cada *host* encontra-se uma JVM e em cada JVM há um container de agentes, o que fornece um ambiente completo para a execução de um ou mais agentes.

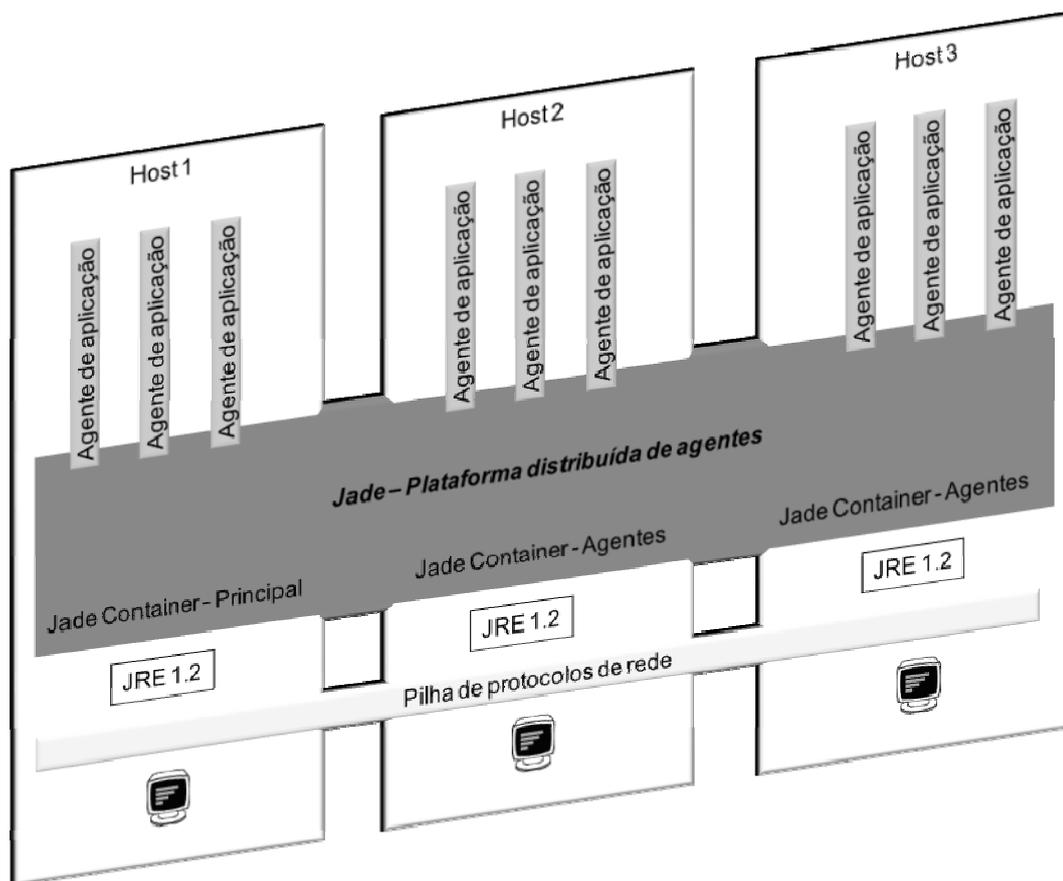


Figura 2.8 – Arquitetura distribuída da plataforma  
Fonte: BELLIFEMINE, 2006

<sup>5</sup> IIOP – *Internet Inter-ORB Protocol* – protocolo responsável por tornar possível que programas distribuídos desenvolvidos em diferentes linguagens se comuniquem através da internet (CORBA, 2001).

A figura 2.9 apresenta o mesmo conceito de *hosts* e plataformas, apresentando a inclusão de uma segunda plataforma, composta também de um JVM e de um container principal.

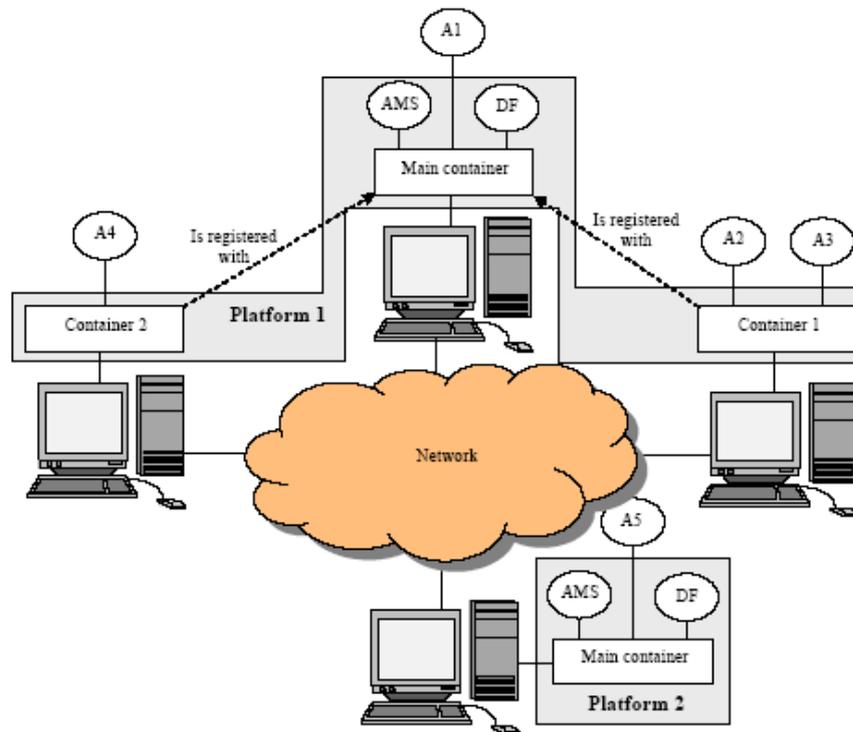


Figura 2.9 – Arquitetura distribuída da plataforma JADE com múltiplos containers  
Fonte: BELLIFEMINE, 2006

Como características da plataforma, ainda podem ser citadas o fato de:

- a) Oferecer Interface Gráfica de Usuário – GUI, que auxilia no gerenciamento e controle de vários agentes e plataformas ao mesmo tempo, inclusive remotamente;
- b) Transportar mensagens no formato FIPA - ACL dentro da mesma plataforma de agentes;
- c) Possuir uma biblioteca de protocolos FIPA – que permite a integração entre agentes da plataforma JADE;
- d) Apresentar serviços de nomes em conformidade com a FIPA – na inicialização dos agentes, estes obtêm seus GUID – *Globally Unique Identifier*, que são identificadores únicos em todo o ambiente;
- e) Automação de registros – registro e cancelamento automático de agentes com o AMS – *Agent Management System*;
- f) Integração – mecanismo que permite à aplicações externas carregarem agentes autônomos JADE;

g) Ambiente de agentes complacente à FIPA – no qual esta incluída o AMS, o DF – Diretório Facilitador e o canal de comunicação destes agentes o ACC – *Agent Communication Channel*.

A figura 2.10 apresenta as possíveis formas de comunicação que a plataforma JADE utiliza para realizar suas trocas de mensagens. O meio é escolhido de acordo com a localização do agente receptor da mensagem. É importante observar, que no canal de comunicação de agentes, existe um *cache* local, que armazena as referências dos objetos de outros *contêineres*. Estas referências são adicionadas ao *cache* sempre que uma mensagem é enviada.

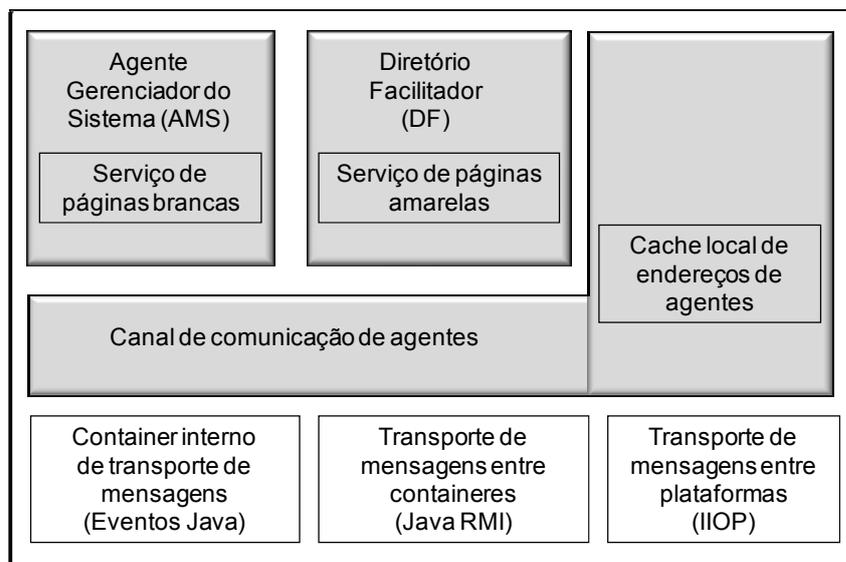


Figura 2.10 – Interoperabilidade na plataforma  
Fonte: BELLIFEMINE, 2006

A figura 2.11 apresenta a interface inicial da plataforma JADE, o RMA - Agente de Monitoramento Remoto, responsável pela administração e controle da plataforma, sendo que mais de um GUI pode ser ativado. Ainda é vista a interface de usuário do DF GUI, que é usada devido à falta do diretório facilitador no JADE. Este GUI permite um simples e intuitivo modo para controlar a base de conhecimentos, ou seja, provê o serviço de "páginas amarelas".

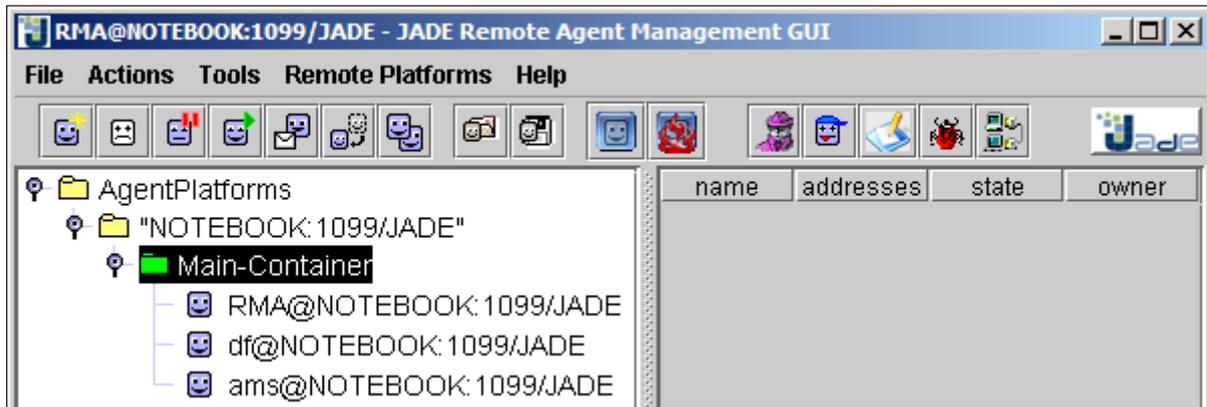


Figura 2.11 – Interface inicial  
Fonte: BELLIFEMINE, 2006

### 2.3.10.1 O agente na plataforma JADE

Segundo o documento FIPA *Abstract Architecture Specification* (SC00001L), um agente é visto como um processo computacional que implementa autonomia, comunicação e funcionalidade a uma aplicação. Ou seja, de uma forma geral a FIPA não especifica as estruturas internas de um agente, preferindo garantir a interoperabilidade através das especificações externas ao mesmo, como protocolos, linguagens e conteúdos.

Para a plataforma JADE um agente é considerado como um objeto ativo que carrega consigo uma *thread*<sup>6</sup> de controle. Ele é completamente autônomo (controla sua execução, decide quando ler as mensagens e quais ler), podendo iniciar diversas conversações simultâneas e executar várias tarefas concorrentemente.

O programador usa comportamentos (*behaviours*) para modelar as ações que o agente é capaz de executar. Os *behaviours* trabalham em modo de agendamento cooperativo, sendo que cada *behaviour* deve ceder o controle para permitir que os outros possam ser executados. Na visão do programador, um agente em JADE é simplesmente uma classe JAVA.

Na figura 2.12 é apresentada uma descrição genérica da arquitetura interna de um agente em JADE. Na parte superior têm-se os comportamentos ativos do agente que representam as ações/intenções que cada agente possui, observa-se que o modelo computacional de um agente em JADE é multitarefa, isto é, tarefas ou comportamentos, são executados de forma concorrente. Cada funcionalidade ou serviço provido por um agente deve ser implementado como um ou mais comportamentos. Na parte inferior esquerda da

<sup>6</sup> Em Java, o programador especifica que os aplicativos contêm fluxo de execução (*threads*), cada *thread* designando uma parte do programa que pode ser executado simultaneamente com outras *threads*. (DEITEL e DEITEL, 2005).

figura, tem-se uma fila privativa de mensagens ACL. Todo agente JADE tem essa fila onde decide quando e quais mensagens recebidas ler. Ao centro, tem-se o seqüenciador de comportamentos e o gerenciador do ciclo de vida.

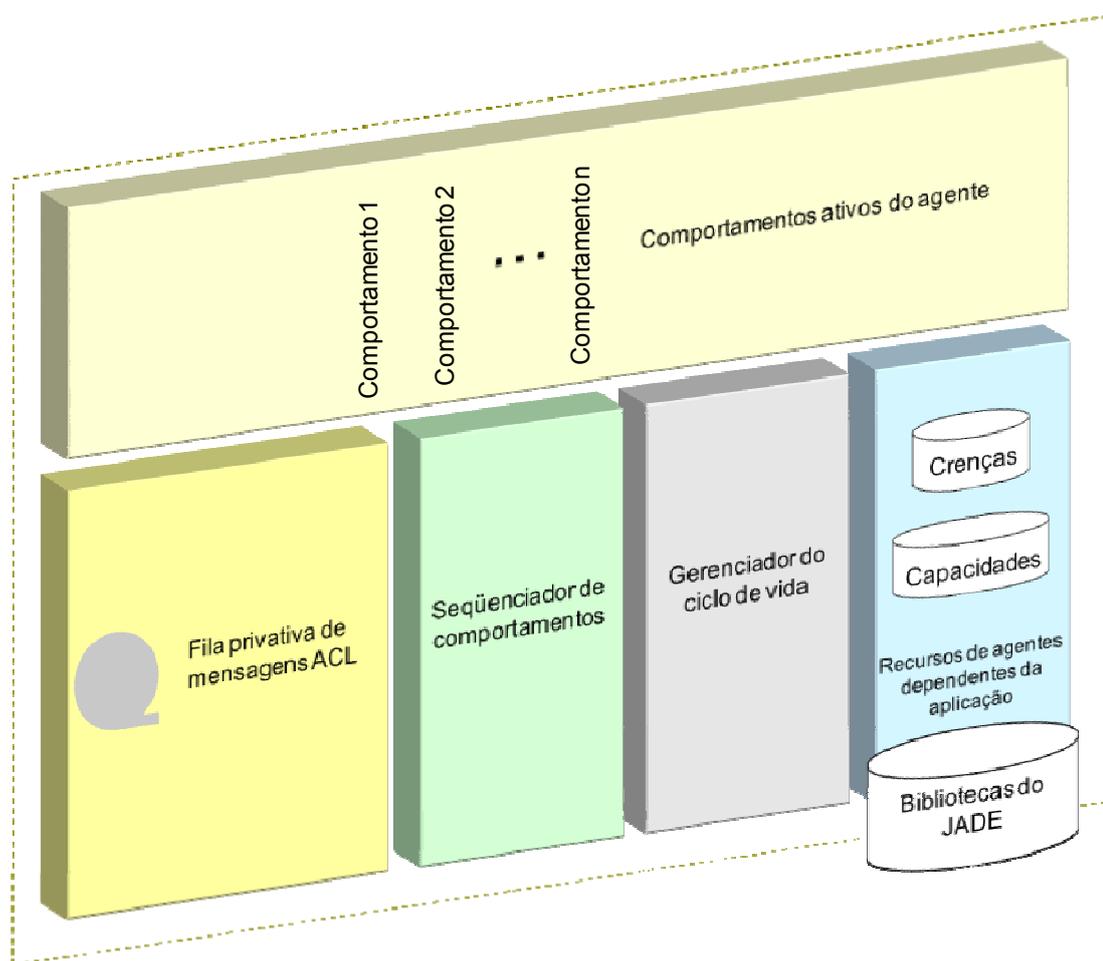


Figura 2.12 – Arquitetura interna genérica de um agente em JADE  
Fonte: BELLIFEMINE, 2006

O primeiro é responsável por seqüenciar a ordem de execução dos comportamentos (caso exista uma ordem de precedência). O segundo, gerenciador de ciclo de vida, é o controlador do estado atual do agente, é a forma que os agentes utilizam para determinar seu estado atual (ativo, suspenso, ou outro).

No lado direito da figura, tem-se os recursos de agentes dependentes da aplicação. Nesse local serão armazenadas as crenças e capacidades que o agente adquiriu na execução da aplicação.

---

## **CAPÍTULO 3 – CICLO DE VIDA PARA O DESENVOLVIMENTO DE UM MODELO DE SIMULAÇÃO BASEADO EM MULTIAGENTES**

Este capítulo trata da questão do ciclo de vida para o desenvolvimento de um modelo de simulação baseado em multiagentes. Dentro desta perspectiva é apresentada primeiramente uma visão do ciclo de vida para o desenvolvimento de um modelo de simulação que não se baseia em agentes, e em seguida é vista a adequação de agentes para o desenvolvimento de modelos de simulação e a necessidade de uma metodologia voltada especificamente para este fim. Em seguida é apresentada a metodologia PASSI para o desenvolvimento e especificação de um modelo de simulação baseado em agentes. Finalmente é discutido o processo de validação e verificação para o modelo adotado.

### **3.1 Ciclo de vida para o desenvolvimento de um modelo de simulação**

Um modelo, de acordo com BALCI (2003), é uma representação ou abstração de algo, que pode ser uma entidade, sistema ou idéia. Simulação, segundo YILMAZ (2006) e BALCI (1994), pode ser definida como o processo para o desenvolvimento de um modelo cuja finalidade é a de conduzir experimentos, que tem como propósito a compreensão do seu próprio comportamento, além da avaliação de suas estratégias alternativas de operação. Um modelo de simulação pode, então, ser definido como uma representação que incorpora tempo e as mudanças que ocorrem durante um determinado período (CARSON II, 2005).

Observa-se que o ciclo de vida para o desenvolvimento de um modelo de simulação varia de autor para autor, mais em termos de nomenclatura do que em suas funcionalidades. Porém, observa-se que as etapas de formulação do problema, definição dos objetivos, modelo conceitual e validação, são comuns a todos (BALCI, 1994; FREITAS, 2001; YILMAZ et al., 2002; e YILMAZ (2006). Neste trabalho optou-se pela adoção do ciclo de vida para o desenvolvimento de um modelo de simulação proposto por YILMAZ et al. (2002) tendo em vista sua contribuição nas etapas de verificação e validação de modelos computacionais baseados em agentes, ferramenta escolhida para o desenvolvimento do modelo proposto. A figura 3.1 apresenta o ciclo de vida para o desenvolvimento de um modelo de simulação proposto por YILMAZ, 2006, formado por dois domínios: do problema e da solução. Os estágios que compõem o ciclo de vida são representados como retângulos, os processos de desenvolvimento do modelo são representados por flechas de linha cheia, e as flechas tracejadas indicam os processos de validação e verificação.

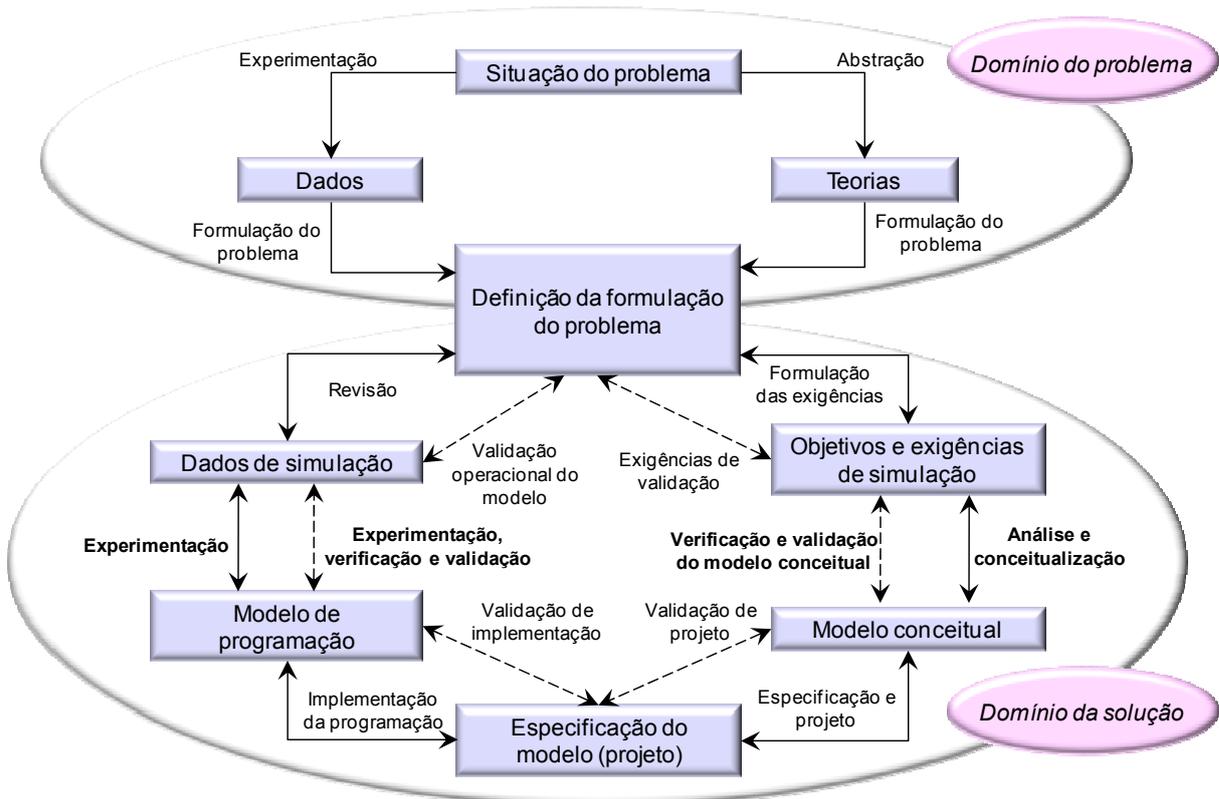


Figura 3.1 – Ciclo de vida para o desenvolvimento de um modelo de simulação

FONTE: Yilmaz, 2006

O domínio do problema inicia com o estágio da situação do problema, que irá criar abstrações para as teorias e propriedades de interesse, a fim de ajudar na elaboração de uma definição do problema. Porém, no caso de um modelo pré-existente, este deverá gerar dados que facilitem a formulação desta definição. Nesta fase, o processo de formulação do problema deverá ser guiado e estruturado por características que lhe sejam interessantes como: ambiente; interdependência; organização e outras (YILMAZ et al., 2002). O domínio da solução é formado pelos estágios de:

- (a) formulação das exigências e objetivos da simulação, que tem como principal objetivo preencher a lacuna entre o nível de definição dos objetivos gerados na definição do problema e o modelo de simulação a ser gerado;
- (b) formulação do modelo através de sua análise e conceitualização, a ser tratada no item 3.2 do presente trabalho;
- (c) especificação e implementação do modelo, que busca a transição entre o modelo conceitual em um modelo de software. Neste estágio vários paradigmas e representações de modelo podem ser utilizados, entre os quais UML – *Unified Modeling Language* (<http://www.uml.org/>), a fim de representar suas diferentes funcionalidades. É importante

notar que neste estágio deve-se escolher um modelo de representação que seja adequado para o domínio da aplicação; e

(d) experimentação e revisão, a serem discutidos no item 3.12, que trata do conceito de validação e verificação, e no item 3.13, que trata especificamente da validação e verificação adotadas no modelo proposto.

### 3.2 Modelo conceitual de simulação

Um modelo conceitual de simulação é, antes de tudo, um mecanismo ou canal que permite a comunicação entre suas fases de desenvolvimento e implementação, devendo conter o contexto de simulação, além de seus elementos e conceitos (PACE, 2000), conforme figura 3.2.

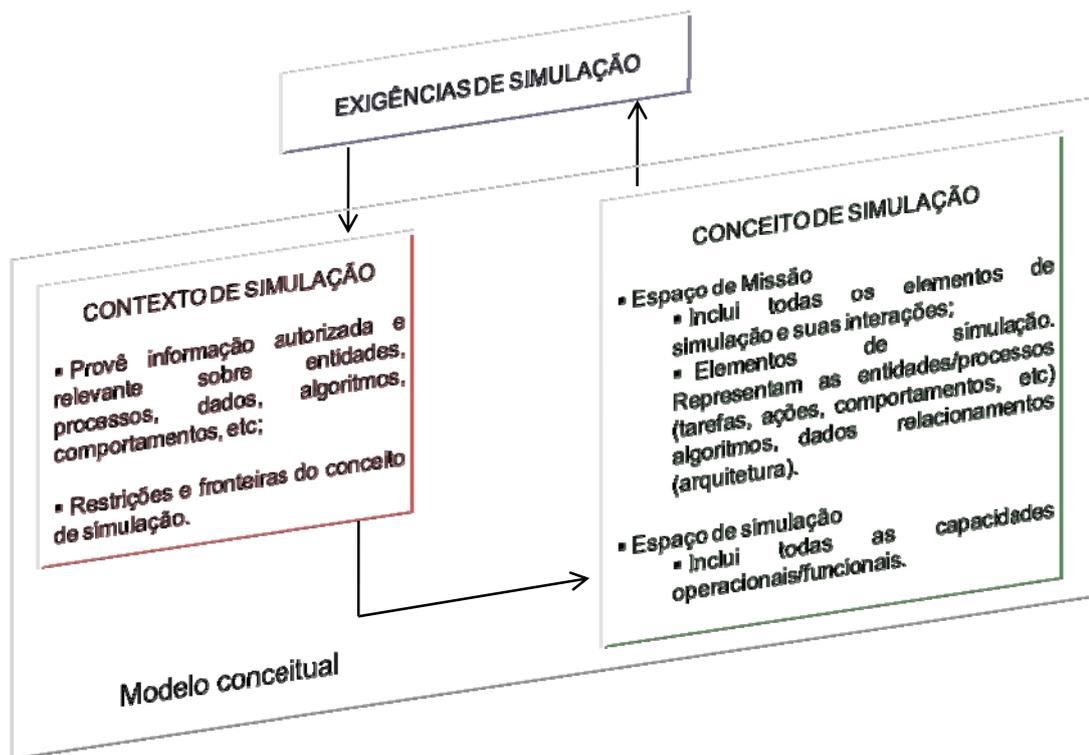


Figura 3.2 – Modelo conceitual de simulação  
FONTE: Pace, 2000

O contexto de simulação tem por objetivo fornecer informação autorizada sobre o domínio no qual a simulação esta inserida. Por exemplo, ao se representar processos físicos é necessário incluir a contextualização das leis da física e os princípios de engenharia que o regem (PACE, 2000). A informação contida no contexto de simulação estabelece as fronteiras para a construção do modelo.

Já o conceito de simulação, de acordo com YILMAZ et al. (2002), inclui a representação do espaço de missão, que é a representação de todos os elementos do domínio e as especificações de como estes elementos interagem, e a representação do espaço de simulação, que inclui a informação adicional, tanto funcional como operacionais, necessárias para que a simulação alcance seus objetivos (ver definição 3.1).

Um elemento de simulação consiste de informação descrevendo conceitos para uma entidade, uma combinação ou uma coleção de entidades. Pode descrever ainda um processo representado pela mesma. Inclui suposições, algoritmos, características, interações, dados, etc; que possibilitem identificar e descrever os seus possíveis estados, comportamentos, atributos, etc (PACE, 2000)

**Definição 3.1**  
**Elemento de simulação**

Ainda segundo PACE (2000) há quatro passos a serem seguidos para o desenvolvimento de um modelo conceitual, conforme apresentado na figura 3.3.

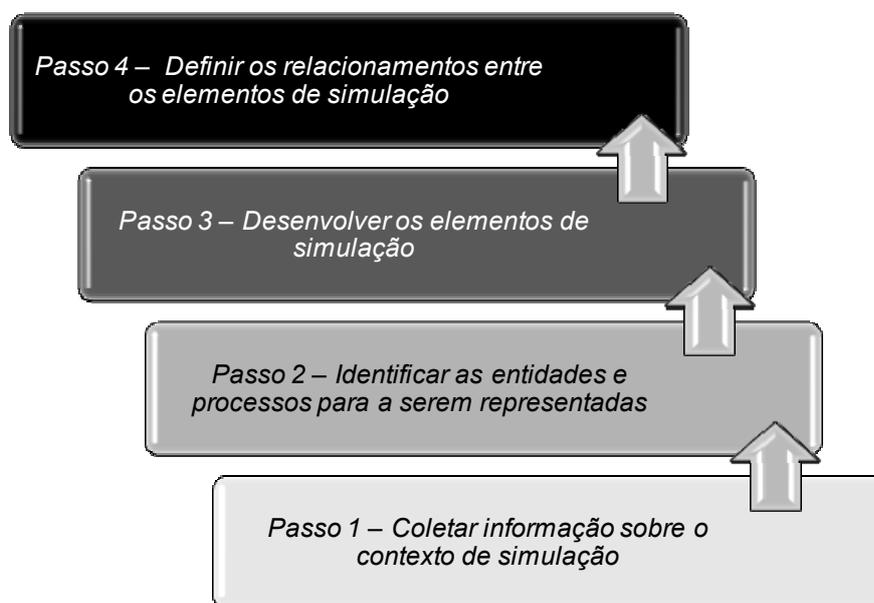


Figura 3.3 – Passos para o desenvolvimento do modelo conceitual  
FONTE: Pace, 2000

Ainda de acordo com PACE (2002), primeiramente é necessário coletar as informações relevantes sobre o contexto de simulação. Como estas informações serão a base para a identificação das entidades e processos necessários para que o modelo de simulação possa cumprir seus objetivos, é necessária atenção especial nas questões de compatibilidade e coerência do mesmo.

No segundo passo, identificação das entidades e processos, são tomadas as decisões de nível de agregação entre as entidades, suas representações como entidades singulares e como comunidades ou sub-sistemas.

O passo seguinte é a definição dos elementos de simulação, onde são tomadas as decisões sobre o nível de precisão, resolução e outras, necessárias para a representação de uma entidade ou processo. Neste passo a fidelidade da simulação é dada em função tanto do seu escopo, formado pelas entidades e processos definidos no segundo passo do desenvolvimento do modelo conceitual, quanto da sua qualidade, aqui representada em termos de qualidade e precisão.

No quarto e último passo são definidos os relacionamentos entre os elementos da simulação, a fim de garantir que as restrições e fronteiras impostas pelo contexto da simulação sejam respeitadas.

### **3.3 Modelos de simulação baseados em agentes**

O uso de agentes como ferramenta de abstração ou metáfora, segundo DAVIDSSON (2001), costuma gerar modelos para a representação de ambientes do mundo real com grande dinamismo e grau de complexidade. Agentes podem representar sistemas complexos com componentes inúmeros e distintos, ao mesmo tempo em que permitem a agregação de diferentes funcionalidades, usualmente consideradas de forma distinta, como planejamento, aprendizado e coordenação, por exemplo.

A simulação baseada em agentes, por sua vez, pode ser vista como a intersecção de três campos científicos:

- (a) computação baseada em agentes;
- (b) ciências sociais, através do estudo das interações entre as entidades; e
- (c) simulação computacional, através do uso de técnicas para a simulação de um fenômeno ou processo, como orientação a objetos, por exemplo.

Desta forma a integração entre a simulação computacional e a computação baseada em agentes resulta no que é chamado de simulação baseada em multiagentes que, segundo MACAL e NORTH (2005), também pode ser encontrada na literatura com os nomes de sistemas baseados em agentes, modelagem baseada no indivíduo e modelagem baseada em agentes.

---

Este tipo de simulação é normalmente utilizado em domínios que contenham diferentes componentes interagindo de forma diversa e complexa, e cujas propriedades do modelo não são deduzidas diretamente das propriedades de seus componentes (LUCK et al., 2004). De acordo com SHEN et al. (2006), para a execução de tarefas de otimização as abordagens baseadas em agentes provêm uma solução distribuída e inteligente, através de negociação, coordenação e cooperação. De acordo com MACAL e NORTH (2005) é apropriado pensar em simulação baseada em agentes quando:

- (a) o mundo real pode ser representado naturalmente por agentes;
- (b) as decisões e comportamentos podem ser definidos discretamente, com fronteiras;
- (c) os agentes devem mudar e se adaptar durante a simulação;
- (d) os agentes possuem relacionamentos dinâmicos com outros agentes; e o mais importante
- (e) a mudança da estrutura do processo resulta do modelo, ao invés de ser uma das entradas para o modelo.

Então, é importante notar que modelos de simulação baseados em multiagentes diferem daqueles não baseados em agentes nos aspectos relativos à autonomia de seus elementos; nas suas propriedades sociais e epistemológicas; e na necessidade do uso de mecanismos de interações (COSSENTINO, 2005). Já LUCK et al. (2004) explicam que os princípios básicos da engenharia de software e de conhecimento necessitam ser aplicados para o desenvolvimento de sistemas multiagentes de qualquer natureza, simulação ou não, mas reforçam que também é necessário que haja uma expansão dos mesmos a fim de cobrir as exigências demandadas por este novo paradigma. MACAL e NORTH (2005) dizem que modelos de simulação baseados em multiagentes não são iguais aos modelos de simulação orientados a objetos, porém ressaltam que o paradigma da orientação a objetos é uma base útil para a modelagem baseada em agentes, visto que um agente pode ser considerado um objeto com capacidades adicionais de escolha e ação.

Do contexto apresentado acima e a partir das diferenças e similaridades entre sistemas de simulação baseados em agentes e não baseados em agentes, detectou-se a necessidade da utilização de uma metodologia para a especificação e desenvolvimento orientada a agentes para o desenvolvimento do modelo de simulação proposto neste trabalho. Isto ocorre devido à necessidade de uma perspectiva de modelagem voltada ao agente e não baseada somente no processo.

### 3.4 Metodologias para o desenvolvimento de sociedades baseadas em agentes

Dentro desta linha, vários trabalhos podem ser encontrados na literatura sobre a modelagem e o projeto de organizações multiagentes, dentre os quais tem-se as seguintes metodologias: *MAS-CommonKADS* proposta por IGLESIAS (1998); *GAIA* formulada por WOOLDRIDGE et al. (1999); *MASE (Multi-Agent System Engineering)* apresentada por DeLOACH et al. (2001), *MESSAGE (Methodology for Engineering Systems of Software of AGENTS)* estabelecida no âmbito do projeto EURESCOM *project P907* (CAIRE, LEAL e RODRIGUES, 2001); *Tropos* proposta por BRESCIANI et al. (2004) e a *Prometheus* apresentada por PADGHAM e WINIKOFF (2003). Uma ampla revisão da literatura sobre metodologias para desenvolvimento de sociedades baseadas em agentes pode ser encontrada nos trabalhos de HENDERSON-SELLER e GIORGINI (2005), BUSSMANN et al. (2004) e DAM e WINIKOFF (2003) além dos citados nas primeiras linhas deste parágrafo.

Neste trabalho será adotada a metodologia *PASSI – Process for Agent Societies Specification and Implementation* para a especificação e desenvolvimento de uma sociedade baseada em agentes que integra conceitos de projeto e modelagem vindos da engenharia de software orientada a objetos e de abordagens baseadas em inteligência artificial (COSSENTINO, SABATUCCI e SEIDITA, 2003).

Esta metodologia é o resultado de estudos teóricos e experimentos desenvolvidos e apresentados em CHELLA, COSSENTINO e LoFASO (2000) e COSSENTINO, SABATUCCI e SEIDITA (2003), tendo sido inicialmente projetada para aplicações em robótica.

Sua escolha levou em consideração principalmente sua aderência a padronizações. Isto é justificado através do uso da linguagem de modelagem UML – *Unified Modeling Language*; e o uso de XML - *eXtensible Markup Language* para a representação da troca de conhecimento entre os agentes através de suas mensagens. Outro fator relevante para a sua escolha reside no fato de ser uma metodologia totalmente desenvolvida conforme os padrões recomendados pela Sua escolha levou em consideração principalmente sua aderência a padronizações. Isto é justificado através do uso da linguagem de modelagem UML – *Unified Modeling Language*; e o uso de XML - *eXtensible Markup Language* para a representação da troca de conhecimento entre os agentes através de suas mensagens. Outro fator relevante para a sua escolha reside no fato de ser uma metodologia totalmente desenvolvida conforme os padrões recomendados pela FIPA – *Fundation for Intelligent Physical Agents* (<http://www.fipa.org/>).

### 3.5 Introdução à metodologia PASSI

Na metodologia PASSI, de acordo com HENDERSON-SELLER e GIORIGINI (2005, pg. 84), o agente é considerado sob a ótica de dois diferentes aspectos. Primeiramente como uma entidade autônoma, nas etapas iniciais do projeto, sendo capaz de realizar um objetivo através de decisões autônomas, ações e relacionamentos. Posteriormente o agente é visto e tratado como uma unidade de software significativa, podendo assumir diversos papéis funcionais durante suas interações com outros agentes a fim de alcançar suas metas.

O processo de projeto é composto pelos cinco modelos, ou fases, conforme figura 3.4, listados a seguir:

- (a) modelo de requisitos do sistema responsável por sua descrição funcional;
- (b) modelo de sociedade de agentes responsável por modelar a solução em função de seus papéis, interações sociais, dependências e ontologias;
- (c) modelo de implementação, que apresenta a arquitetura em função de suas classes e métodos. É importante notar que diferentemente de uma arquitetura orientada a objetos, em uma arquitetura orientada a agentes existem dois níveis de abstração: um nível social (multiagente) e um nível individual (agente).
- (d) modelo de código; e
- (e) modelo de desenvolvimento, que apresenta a distribuição dos agentes entre as unidades de processamento (*hardware*) e sua movimentação através das diferentes plataformas disponíveis (COSSENTINO e SABATUCCI, 2003).

Há ainda a etapa de teste, que é uma verificação do comportamento de um agente em relação aos requisitos originais e sua validação, em termos de eficiência e cooperação entre os diversos níveis da sociedade.

A questão relativa à validação e verificação será considerada no item 3.12, enquanto o item 3.13 e trata especificamente da validação e verificação adotadas no modelo proposto.

Antes de iniciar o detalhamento dos modelos que compõem a metodologia PASSI, com suas respectivas sub-fases, é necessário ressaltar dois aspectos importantes:

- (a) conforme será explicado ao final deste capítulo, apesar da adequação da metodologia PASSI para o desenvolvimento e especificação de sociedades baseadas em agentes, realizou-se algumas alterações, inclusões e exclusões de sub-fases e documentações, visando facilitar a compreensão e replicação do modelo proposto; e

(b) alguns diagramas e documentos não serão detalhados no presente capítulo, tendo em vista sua melhor compreensão no contexto do capítulo 5 e 6, onde serão apresentados o modelo conceitual e a arquitetura multiagente proposta no presente trabalho, respectivamente.

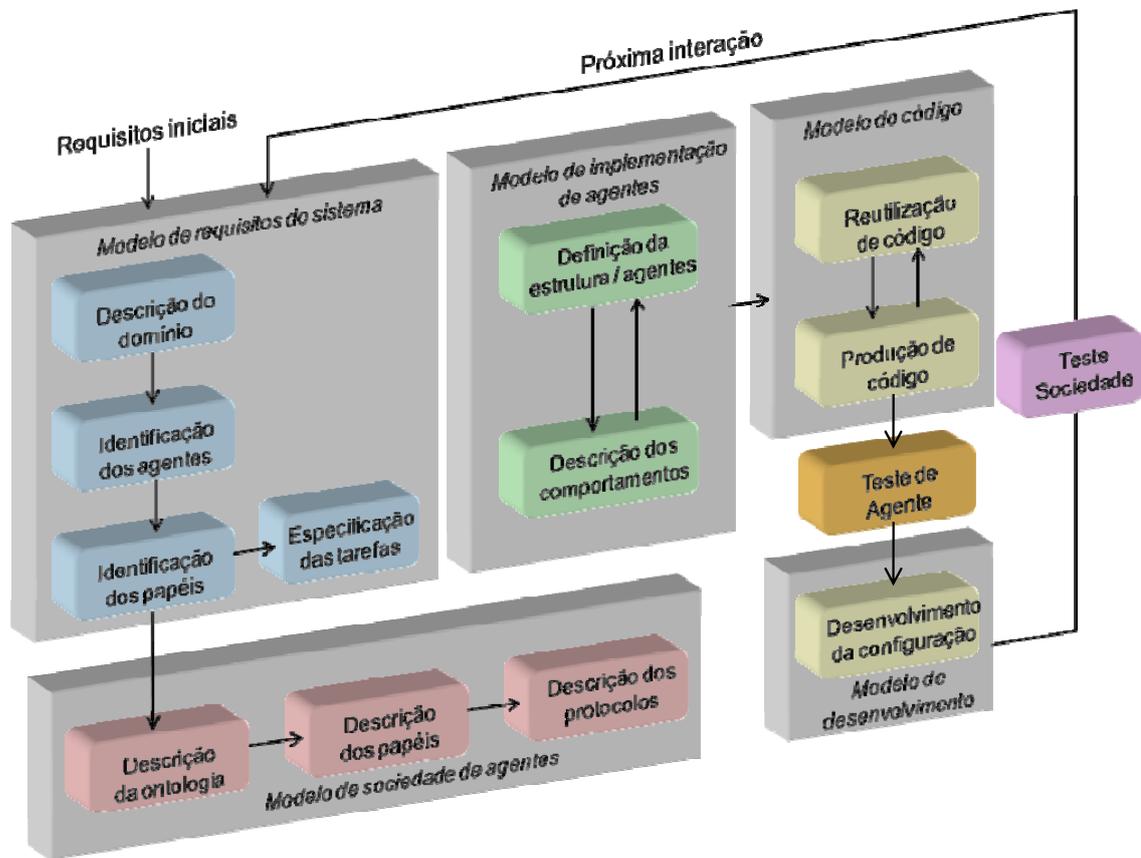


Figura 3.4 – Modelos que compõem a metodologia PASSI  
FONTE: Henderson-Sellers e Giorgini, 2005

### 3.6 Modelo de requisitos do sistema

A primeira das cinco fases que compõem a metodologia PASSI é chamada de modelo de requisitos do sistema, e tem como objetivo sua descrição funcional. É composta por quatro passos, ou sub-fases, distintos:

- (a) descrição do domínio, onde são definidos os propósitos e objetivos do estudo (FREITAS, 2001) a fim de permitir o mapeamento dos principais processos do modelo;
- (b) identificação dos agentes, que segundo MACAL e NORTH (2005) são os “tomadores de decisões”. Nesta sub-fase são especificadas as responsabilidades relativas a cada agente. Na metodologia PASSI um agente é uma unidade significativa de software, sendo uma instância da classe agente;

- (c) identificação dos papéis, ou seja, identificação de todas as tarefas necessárias de serem cumpridas por um agente a fim de alcançar um objetivo; e
- (d) especificação das tarefas, onde são descritas as capacidades de cada um dos agentes.

A figura 3.5 apresenta o meta-modelo da fase de requisitos de sistemas em função dos relacionamentos entre a documentação de trabalho e os diagramas gerados.

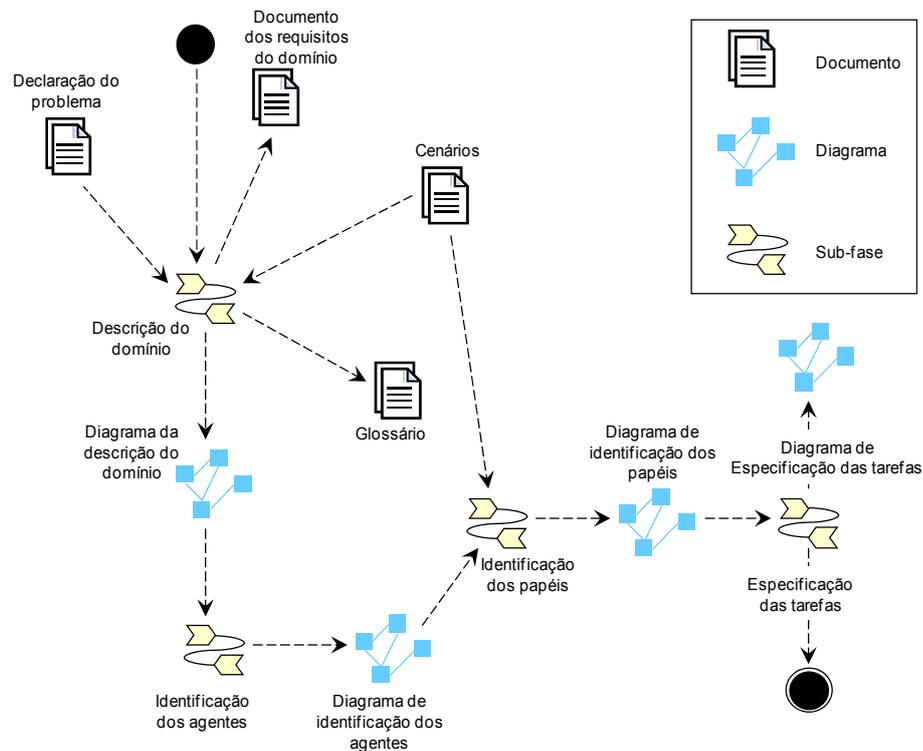


Figura 3.5 – Meta-modelo representando a fase de Requisitos de Sistema da metodologia PASSI  
 FONTE: Cossentino, Sabatucci e Seidita (2003)

Ao final desta primeira fase é produzida uma documentação, usualmente composta por modelos UML e por diversos documentos de trabalho gerados durante a execução da mesma. Os documentos de trabalho gerados nesta fase são:

- (a) a declaração do problema, que de acordo com WAZLAWICK (2004) é apenas uma descrição desestruturada, onde podem co-existir informações de nível gerencial e operacional, podendo ou não incluir detalhes sobre as tecnologias a serem empregadas;
- (b) os requisitos do domínio, que tem como base as necessidades mapeadas na declaração do problema. São registrados os tópicos relativos àquilo que o modelo de simulação proposto deve executar e sob quais condições deve fazê-lo (WAZLAWICK, 2004);
- (c) os cenários, onde são descritas as diversas interações entre os agentes que compõem o sistema; e finalmente
- (d) um glossário.

O modelo de requisitos de sistema também irá gerar quatro diagramas UML: (a) descrição do domínio; (b) identificação dos agentes; (c) identificação de papéis; e (e) especificação das tarefas.

### 3.7 Modelo de sociedade de agentes

A segunda fase que compõe a metodologia PASSI é chamada de modelo de requisitos do sistema, e tem como objetivo sua descrição funcional. A figura 3.6 apresenta o meta-modelo da fase de sociedade de agentes em função dos relacionamentos entre a documentação de trabalho e os diagramas gerados.

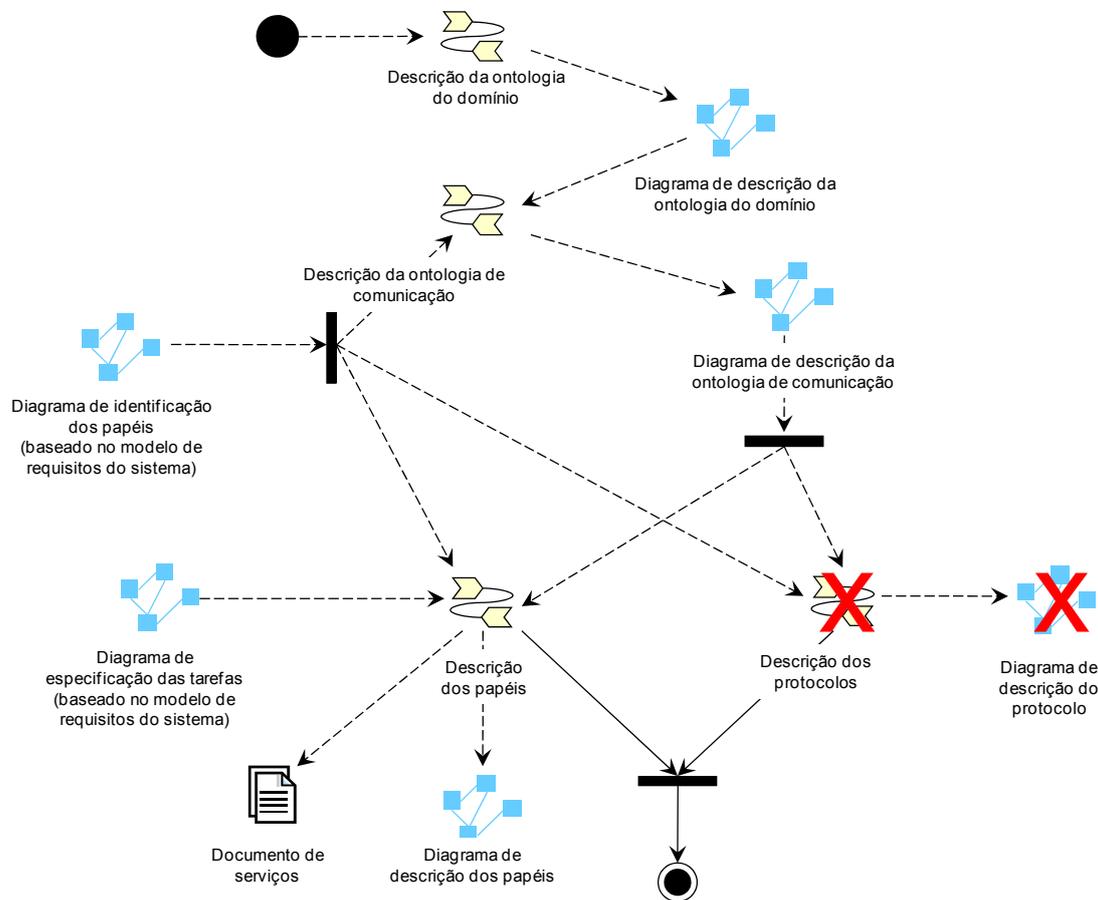


Figura 3.6 – Modelo de sociedade de agentes descrito em função da documentação e diagramas gerados em suas sub-fases

FONTE: Cossentino, Sabatucci e Seidita (2003)

Esta fase é composta por três passos ou sub-fases distintos:

(a) descrição da ontologia, composta pelas ontologias de domínio e de comunicação. O parâmetro de ontologia é usado junto com o parâmetro de idioma a fim de auxiliar na interpretação de expressões contidas nas mensagens enviadas entre agentes. Em muitas

situações, o parâmetro de ontologia será padronizado pela comunidade de agentes, e assim este parâmetro de mensagem poderá ser omitido (FIPA ACL, 2002).

- (b) descrição dos papéis, onde é feita a descrição das tarefas identificadas na sub-fase de identificação de papéis no modelo de requisitos do sistema; e
- (c) descrição dos protocolos, onde são descritos todos os protocolos que foram desenvolvidos especialmente para o modelo proposto, como por exemplo, protocolos de negociação.

### 3.8 Modelo de implementação de agentes

A terceira fase que compõe a metodologia PASSI é chamada de modelo de implementação de agentes, e tem como objetivo apresentar a arquitetura em função de suas classes e métodos em dois níveis de abstração: um nível social (multiagente) e um nível individual (agente). A figura 3.7 apresenta o meta-modelo da fase de implementação de agentes em função dos relacionamentos entre a documentação de trabalho e os diagramas gerados.

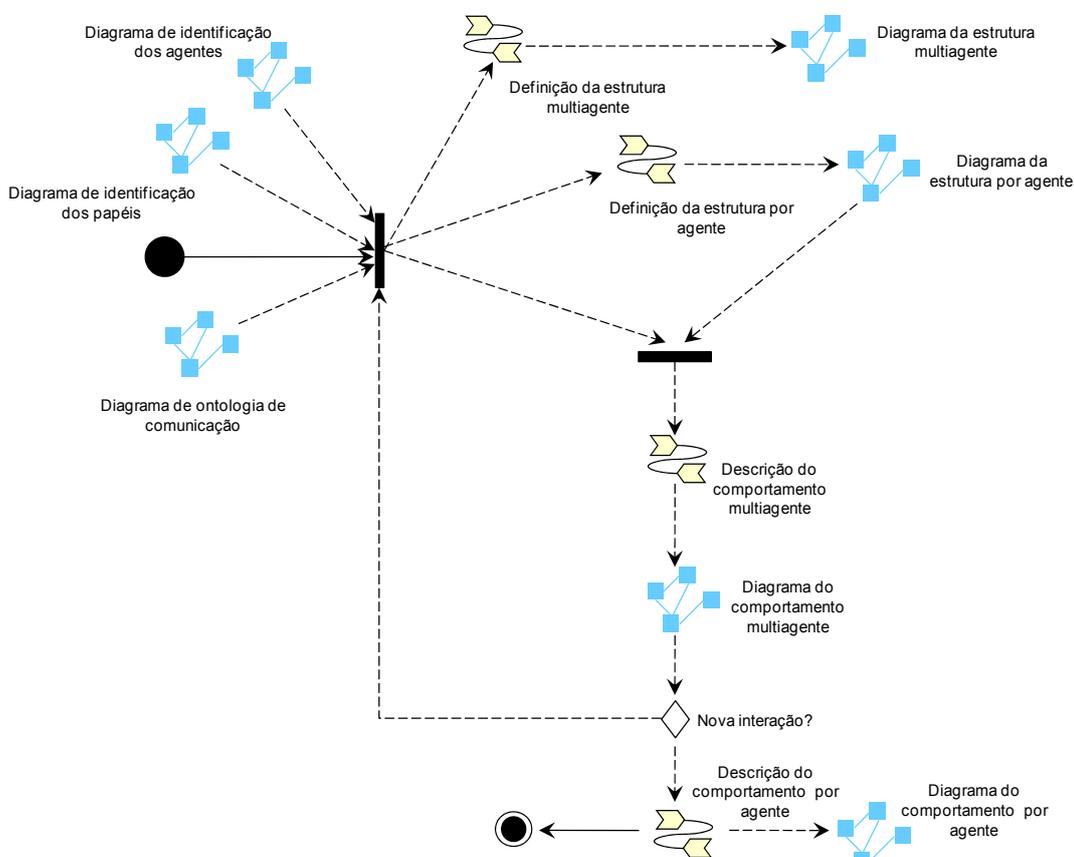


Figura 3.7 – Modelo de implementação de agentes descrito em função da documentação e diagramas gerados em suas sub-fases

FONTE: Cossentino, Sabatucci e Seidita (2003)

Esta fase é composta por três passos, ou sub-fases, distintos:

- (a) definição da estrutura multiagente. Os agentes são representados como classes, onde os atributos representam o conhecimento inerente ao agente e as operações representam as suas tarefas;
- (b) definição da estrutura por agente. As tarefas de cada um dos agentes são especificadas em função de suas sub-tarefas;
- (c) descrição do comportamento multiagente. Nesta sub-fase são representados os eventos internos aos agentes e os métodos que os invocam; e
- (d) descrição do comportamento por agente.

### 3.9 Modelo de código

Esta fase segue o desenvolvimento de código tradicional, na qual é elaborado o “corpo” do código através das informações obtidas até o momento, porém substituindo-se as classes, atributos e operações da programação orientada a objetos, por agentes, conhecimento e tarefas. A figura 3.8 apresenta o meta-modelo da fase de código em função dos relacionamentos entre a documentação de trabalho e os diagramas gerados.

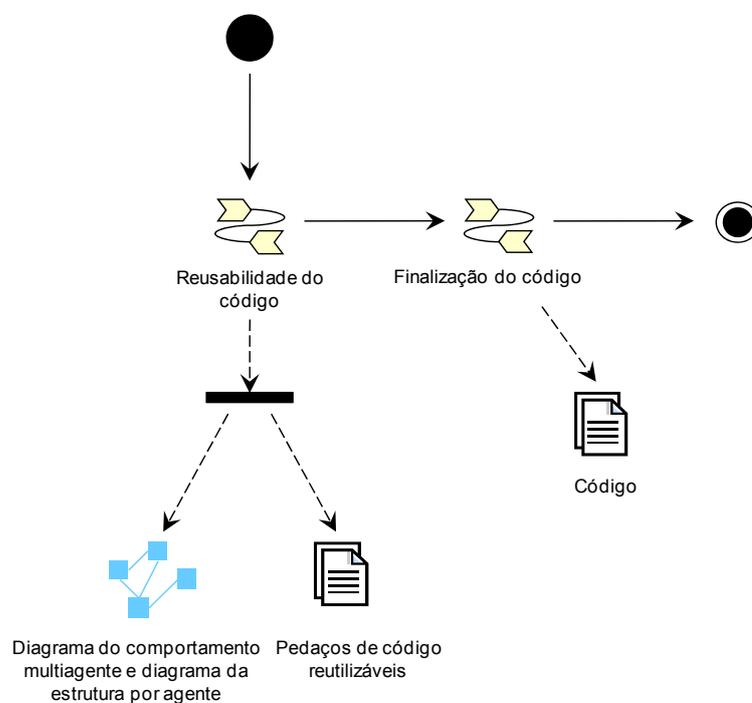


Figura 3.8 – Modelo de código descrito em função da documentação e diagramas gerados em suas sub-fases  
 FONTE: Cossentino, Sabatucci e Seidita (2003)

---

Esta fase é composta por dois passos, ou sub-fases, distintos:

- (a) reusabilidade do código;
- (b) finalização do código.

Tendo em vista que em relação às suas entradas e saídas o processo de projeto de um modelo de simulação baseado em multiagentes não difere muito do projeto de um modelo tradicional (por exemplo, orientado a objetos), também aqui é importante a padronização e a reusabilidade de partes do código (COSSENTINO et al.,2004).

Estas características de padronização e reusabilidade são chamadas *Patterns*, e são definidas por COSSENTINO et al. (2004) como a representação e implementação dos comportamentos do modelo, sendo compostas pelo modelo dinâmico do comportamento, pela sua estrutura, pelos elementos envolvidos e pelo código propriamente dito.

Ainda conforme COSSENTINO et al. (2004), do ponto de vista estrutural a metodologia PASSI classifica as *Patterns* em quatro diferentes tipos:

- (a) de ação, por exemplo, uma tarefa;
- (b) de comportamento, podendo este ser considerado como uma funcionalidade complexa que necessita de uma ou mais tarefas para seu cumprimento;
- (c) de componente, por exemplo, um agente; e
- (d) de serviço, onde duas ou mais *patterns* de componente interagem a fim de alcançar um comportamento cooperativo.

A fim de preservar as características de reusabilidade do código, COSSENTINO, SABATUCCI e SEIDITA (2003) sugerem, entre outras medidas, o uso sempre que possível de diagramas UML, além do uso de plataformas de desenvolvimento aderentes à FIPA como FIPA-OS ou JADE, por exemplo. Uma ampla revisão sobre as bibliotecas de *patterns* aplicadas na metodologia PASSI não faz parte do escopo deste trabalho, podendo ser encontrada em COSSENTINO, SABATUCCI e SEIDITA (2003) e COSSENTINO et al. (2004).

Dentro do contexto apresentado acima, deve-se ter em mente que o modelo de código e suas sub-fases são inerentes ao desenvolvimento do projeto do modelo de simulação descrito neste trabalho. Também é importante notar que a escolha das ferramentas, tanto para o desenvolvimento como para a representação do modelo, sob a ótica da reusabilidade, estão dentro dos parâmetros especificados pela metodologia PASSI. A representação desta reusabilidade é apresentada ao longo de parte do capítulo 4 e de todo o capítulo 5 e 6, não havendo um item específico para a descrição da mesma na arquitetura do modelo.

### 3.10 Modelo de desenvolvimento

Nesta fase é representada a distribuição dos agentes do modelo através das unidades de hardware que realizam seu processamento. Também é representada a migração dos agentes entre as unidades de hardware. A figura 3.9 apresenta o meta-modelo de desenvolvimento, o qual é composto por uma única sub-fase, chamada de desenvolvimento de configuração, onde são representadas, através de diagramas, as unidades de hardware disponíveis e as possíveis restrições à mobilidade e migração dos agentes.

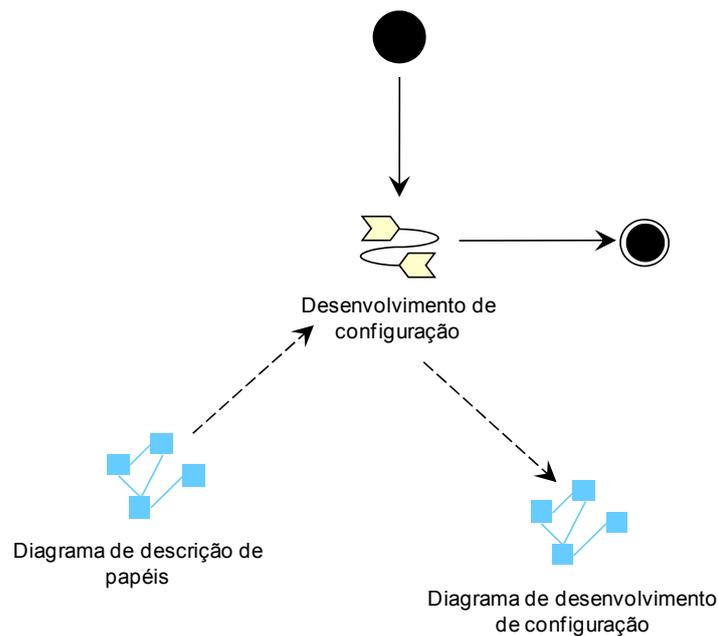


Figura 3.9 – Modelo de desenvolvimento  
FONTE: Cossentino, Sabatucci e Seidita (2003)

### 3.11 Alterações na metodologia proposta

Conforme citado no item 3.5, de introdução à metodologia PASSI, algumas alterações foram adotadas em relação à utilização da mesma.

Em relação ao modelo de requisitos do sistema foram realizadas as seguintes alterações na aplicação da metodologia PASSI:

- (a) não será utilizado um sub-item específico para glossário, sendo que as principais definições de interesse se encontram em destaque ao longo do texto;
- (b) inclusão a priori de uma descrição detalhada dos componentes de simulação que irão compor o modelo proposto, e somente após é feita a descrição do domínio, item 4.1.2, que caracteriza o ambiente modelado;

- 
- (c) tendo em vista a importância da sub-fase de identificação dos agentes e suas interações (MACAL e NORTH, 2005), e apesar da metodologia PASSI não prever nenhum tipo de documentação nesta sub-fase, optou-se por uma descrição detalhada de cada um dos agentes, com suas respectivas responsabilidades, que irão compor o modelo. Isto será apresentado no item 5.1 que trata da sub-fase de identificação dos agentes;
- (d) não será adotado o modelo UML de casos de uso para a descrição do domínio, optando-se por uma documentação mais detalhada. Para tal adotou-se a padronização de documentação sugerida por WAZLAWICK (2004), que diz que nesta fase de concepção os casos de uso só precisam ser listados e eventualmente descritos em um alto nível de abstração, podendo-se ou não indicar os atores envolvidos e os requisitos do domínio relacionados. Esta documentação é apresentada na tabela 4.1 do presente trabalho;
- (e) foi incluída uma descrição dos diferentes papéis que cada um dos agentes pode assumir durante seu ciclo de vida na sub-fase de identificação dos papéis, item 5.2;
- (f) não serão descritos os inúmeros cenários possíveis na forma de documentação. Isto ocorre tendo em vista o número elevado de combinações possíveis entre os diversos papéis funcionais que um mesmo agente pode assumir durante suas interações com outros agentes a fim de alcançar suas metas. Optou-se por utilizar para este fim o diagrama de identificações de papéis, a ser discutido no item 5.2, que trata da sub-fase de identificação dos papéis do presente trabalho; e
- (g) devido ao uso de uma documentação mais elaborada, (itens (b), (d) e (e) acima citados), não julgou-se necessária a utilização da sub-fase de especificações de tarefas, sugerida pela metodologia PASSI.

Em relação ao modelo de sociedade de agentes, foi adotada a seguinte modificação: exclusão do documento de descrição de papéis, que foi antecipada para o item 5.2, que trata da sub-fase de identificação dos papéis no modelo de requisitos do sistema.

### **3.12 O conceito de validação e verificação**

A credibilidade dos resultados de um modelo de simulação não depende somente de quão corretamente o modelo foi projetado, mas também da acuracidade na formulação do problema. Modelos de simulação cuja validação e a verificação não são demonstradas correm o risco de disseminar erros ou conhecimento impreciso (YILMAZ, 2006).

Validação é definida por BANKS et al. (1987) como a demonstração de que um modelo computadorizado satisfaz os seus objetivos e exigências com a precisão necessária ao seu domínio de aplicação, e segundo BALCI (1994) a validação relaciona-se à construção do modelo correto para o escopo escolhido. De uma perspectiva operacional, MISER (1993) define que o processo de validação é uma forma de assegurar que uma teoria ou modelo descreve um fenômeno determinado, sendo adequado ao uso ao qual será aplicado.

Já o processo de verificação, de acordo com YLMAZ (2006) e BALCI (1994), trata da construção do modelo, ou seja, da acuracidade em transformar um problema formulado em uma especificação de modelo. De acordo com MISER (1993), a verificação é o processo pelo qual se assegura que o modelo construído é, na verdade, aquele que se pretendia construir inicialmente. É importante notar que MISER (1993) ainda sugere que a verificação pode ser considerada uma parte do processo de validação.

MACKENZIE et al. (2002) e BALCI (1994) explicam que as técnicas de validação e verificação devem ser empregadas durante todo o ciclo de desenvolvimento do modelo de simulação, desde a formulação do problema até a apresentação de seus resultados. Este tipo de abordagem permite a identificação e retificação de deficiências ao longo de todo processo.

DAVIDSSON et al. (2007) sugerem duas abordagens para a validação de modelos de simulação baseados em agentes: a qualitativa, baseada na opinião de um especialista do domínio sobre o modelo de simulação; e a quantitativa, através da comparação das saídas com as métricas do mundo real modelado.

Ainda sob uma perspectiva qualitativa, YLMAZ (2006) explica que uma grande variedade de métodos pode ser aplicada para avaliar a fidelidade de um modelo conceitual, dependendo do contexto no qual o mesmo está inserido. O autor propõe o uso de casos de teste previamente preparados por especialistas no domínio, os quais permitem avaliar a extensão na qual o conhecimento social representado no modelo conceitual reflete aqueles elaborados pelos especialistas humanos.

BALCI (1994) explica que as técnicas estáticas para a verificação e validação de modelos de simulação são amplamente utilizadas. Isto ocorre porque neste tipo de verificação não é necessária a execução do modelo completo. De acordo com o autor, o próprio compilador da linguagem escolhida para o desenvolvimento do modelo pode ser tratado como uma verificação. As técnicas estáticas são recomendadas para a obtenção de uma grande variedade de informações sobre a estrutura do modelo, sua sintaxe e consistência global.

Finalmente, é importante que se faça uma clara distinção entre a credibilidade do modelo e a credibilidade dos resultados obtidos pela simulação. BALCI (1994) explica que a credibilidade de um modelo é julgada tendo em vista a definição do modelo e os objetivos do

estudo. Já a credibilidade dos resultados é avaliada tendo em vista os resultados existentes, por exemplo, do sistema real modelado.

### 3.13 O conceito de validação e verificação adotado no modelo proposto

De um modo geral, o processo de validação e verificação adotado neste trabalho segue a perspectiva orientada ao processo, proposta por YLMAZ (2006), para a verificação e validação de processos sociais em modelos computacionais baseados em agentes.

A validação do modelo proposto será realizada através da demonstração da viabilidade do modelo de simulação baseado em multiagentes, tendo como base a especificação do projeto referente ao modelo conceitual. Durante este processo, especial atenção será dada aos elementos-chave do modelo proposto: a representação dos componentes de simulação; os agentes; e os protocolos de negociação desenvolvidos especialmente para este trabalho.

Será demonstrado que o modelo proposto, na forma de um protótipo de laboratório, foi implementado e que os diversos agentes propostos foram criados utilizando-se diferentes ferramentas de apoio, entre as quais o ambiente para o desenvolvimento do modelo proposto.

A verificação do modelo proposto consiste em avaliar a viabilidade e a credibilidade geral do modelo e deve concentrar-se nas especificações do projeto do modelo e seu respectivo desenvolvimento computacional. Deste modo, a verificação tem por escopo assegurar que o modelo tenha sido corretamente implementado do ponto de vista computacional a partir das especificações de projeto, enfatizando os processos sociais dos agentes e a consistência da organização multiagente (YLMAZ, 2006).

Tendo em vista que no escopo deste trabalho será desenvolvido um protótipo de simulação que tem como objetivo principal estudar a correlação entre planos de processo alternativos e a programação da produção em um ambiente de *layout* funcional, optou-se pela adoção de uma ótica quantitativa, a fim de verificar a credibilidade dos resultados obtidos pela simulação.

De acordo com DAVIDSSON et al. (2007), se uma nova abordagem é desenvolvida para a solução de um problema, e este problema já havia sido solucionado previamente através de outras abordagens, os resultados podem e devem ser comparados. Segundo os autores, desta forma se verifica a adequabilidade da abordagem sugerida para a resolução do problema. Tendo em vista este contexto, será apresentada neste trabalho uma comparação entre os dados obtidos pelo modelo proposto e aqueles obtidos por KIM et al. (2003) e

---

POTTER (1997). Todas as abordagens utilizaram o mesmo caso de teste, sugerido por YEO et al. (2003), como dados de entrada do problema.

### 3.14 Conclusão do capítulo

Neste capítulo tratou-se da questão do ciclo de vida para o desenvolvimento de um modelo de simulação baseado em multiagentes. Dentro desta perspectiva foi apresentada a visão de YLMAZ et al. (2002), tendo em vista sua contribuição nas etapas de verificação e validação de modelos computacionais baseados em agentes, ferramenta escolhida para o desenvolvimento do modelo proposto.

Foi visto ainda, que o uso de agentes como ferramenta de abstração ou metáfora, costuma gerar modelos para a representação de ambientes do mundo real com grande dinamismo e grau de complexidade. Isto ocorre, pois agentes podem representar sistemas complexos com componentes inúmeros e distintos, ao mesmo tempo em que permitem a agregação de diferentes funcionalidades, usualmente consideradas de forma distinta, como planejamento, aprendizado e coordenação (DAVIDSSON, 2001).

Dentro deste contexto, foi observado que os modelos de simulação baseados em multiagentes diferem daqueles não baseados em agentes nos aspectos relativos à autonomia de seus elementos; nas suas propriedades sociais e epistemológicas; e na necessidade do uso de mecanismos de interações (COSSENTINO, 2005), o que resulta em uma perspectiva de modelagem voltada ao agente e não somente baseada no processo.

Detectou-se a necessidade da utilização de uma metodologia para a especificação e desenvolvimento orientada a agentes para o desenvolvimento do modelo de simulação proposto, sendo que neste trabalho optou-se pela adoção da metodologia PASSI – *Process for Agent Societies Specification and Implementation* (COSSENTINO, SABATUCCI e SEIDITA, 2003), sendo que sua escolha baseou-se principalmente em sua aderência a padronizações e aos padrões recomendados pela Sua escolha levou em consideração principalmente sua aderência a padronizações. Isto é justificado através do uso da linguagem de modelagem UML – *Unified Modeling Language*; e o uso de XML - *eXtensible Markup Language* para a representação da troca de conhecimento entre os agentes através de suas mensagens. Outro fator relevante para a sua escolha reside no fato de ser uma metodologia totalmente desenvolvida conforme os padrões recomendados pela FIPA – *Fundation for Intelligent Physical Agents* (<http://www.fipa.org/>).

---

Finalmente foi discutido o processo de validação e verificação para o modelo adotado, visto que a credibilidade dos resultados de um modelo de simulação não depende somente de quão corretamente o modelo foi projetado, mas também da acuracidade na formulação do problema.

Tendo em vista que no escopo deste trabalho será desenvolvido um protótipo de simulação que tem como objetivo principal estudar a correlação entre planos de processo alternativos e a programação da produção em um ambiente de *layout* funcional, optou-se pela adoção de uma ótica quantitativa, a fim de verificar a credibilidade dos resultados obtidos pela simulação. Para este fim será apresentada uma comparação entre os dados obtidos pelo modelo proposto e aqueles obtidos por KIM et al. (2003) e POTTER (1997). Todas as abordagens utilizaram o mesmo caso de teste, sugerido por YEO et al. (2003), como dados de entrada do problema.

---

## CAPÍTULO 4 – MODELO CONCEITUAL

Este capítulo trata da validação do contexto da simulação e de como as exigências para o seu desenvolvimento foram especificadas e cumpridas, bem como das bases de conhecimento que posteriormente serão utilizadas pelos agentes para compor o modelo proposto. Para isto, adotou-se as definições apresentadas por PACE (2000), anteriormente citadas no item 3.2, relativo ao modelo conceitual de simulação.

Conforme citado no item 3.13, que trata dos conceitos de validação e verificação adotados, a validação do modelo será feita através da demonstração de sua viabilidade teórica, tendo como base a especificação do projeto referente ao modelo conceitual, porém não sendo limitado pelo mesmo.

Uma relação direta foi feita entre a definição de contexto de simulação, explicitada no item 3.2, e o modelo de requisitos de sistema da metodologia PASSI, tratado no item 3.6.

O capítulo inicia com a descrição do problema a ser modelado, seguida da caracterização do ambiente utilizado como modelo e pela descrição dos elementos que irão compor o modelo de simulação, bem como pela representação das flexibilidades inseridas nos plano de processos das peças. O capítulo encerra com a descrição dos requisitos de domínio do modelo proposto.

### 4.1 Modelo de requisitos do sistema - Sub-fase de descrição do domínio

Conforme citado no item 3.6 que trata do modelo de requisitos do sistema na metodologia PASSI, a sub-fase de descrição do domínio tem como principal objetivo o detalhamento funcional da sociedade baseada em agentes, iniciando com a declaração do problema. A figura 4.1 situa esta sub-fase dentro da metodologia PASSI.

As sub-fases, diagramas e documentos assinalados com uma letra “x” em vermelho, indicam as modificações realizadas no modelo de requisitos do sistema a fim de adequá-lo ao modelo de simulação proposto. Estas modificações foram especificadas anteriormente no item 3.11, que trata das alterações na metodologia PASSI.

Ao se iniciar um projeto que envolva simulação, é importante considerar que normalmente este tipo de modelo é criado a fim de se obter respostas para determinadas questões ou para dar suporte a uma ou mais decisões. DAVIDSSON et al. (2007) classifica os modelos de simulação baseados em multiagentes em quatro diferentes categorias: predição, responsáveis por prognósticos futuros; verificação, responsáveis por determinar a validade de

---

uma teoria ou hipótese; treinamento, responsável por melhorar as habilidades de um usuário em determinado domínio; e análise, responsável por estudar uma determinada teoria a fim de refiná-la.

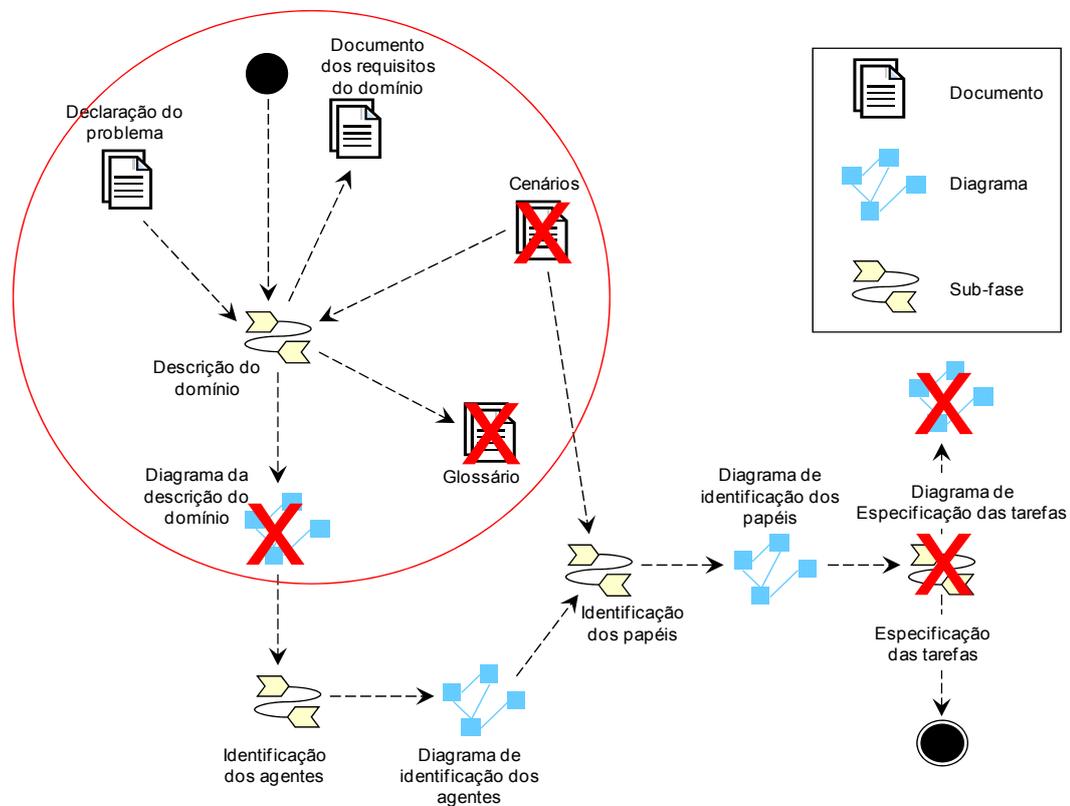


Figura 4.1 – Sub-fase de descrição do domínio  
 FONTE: Cossentino, Sabatucci e Seidita (2003)

Dentro deste contexto, o modelo aqui proposto e apresentado pode ser classificado como um modelo de análise e verificação, sendo que a hipótese em questão corresponde ao estudo em profundidade da questão da correlação entre planos de processo alternativos e a programação da produção em um ambiente de *layout* funcional. Busca-se, através deste estudo, comprovar ou refinar a teoria existente a respeito do fenômeno aqui especificado.

#### 4.1.1 Declaração do problema

Este modelo de simulação tem por propósito capturar os aspectos referentes a um ambiente de *layout* funcional, também conhecido como por processo. Este será composto por diferentes recursos e ordens de serviços.

---

O principal objetivo deste modelo será o de permitir o estudo da correlação entre planos de processo alternativos, representados neste modelo sob a forma de diferentes flexibilidades, e o seqüenciamento da produção, em um ambiente de *layout* funcional.

O modelo permitirá a negociação entre ordens e os recursos, a fim de possibilitar a programação das ordens de produção. Os resultados obtidos nestas simulações serão analisados através das métricas de *makespan* e *flow time*, a serem descritas no item 4.1.3 do presente capítulo. Além disto, o modelo calculará automaticamente a média, desvio padrão e percentuais relativos às métricas já citadas, que deverão estar agrupadas por rodadas de simulações.

Através do modelo de simulação proposto também será possível a configuração de alguns parâmetros de entrada de dados por parte do usuário, como os recursos a serem disponibilizados por rodada de simulação e a composição dos pedidos. Finalmente, o usuário deverá ter a possibilidade de configurar os parâmetros de como será apresentada a visualização dos dados e parâmetros como *sets*, *setups* e número da simulação, além de permitir a escolha entre diferentes flexibilidades de planos de processo.

Após a realização das atividades acima descritas, será feita uma descrição formal do contexto de simulação, representado através de seus elementos e dos conceitos que o regem. É importante lembrar que o contexto de uma simulação tem por objetivo fornecer informação autorizada sobre o domínio na qual a mesma esta inserida, estabelecendo as fronteiras para a construção do modelo (PACE, 2000), conforme citado no item 3.2, que trata do modelo conceitual de simulação.

#### 4.1.2 Caracterização do ambiente modelado

Problemas de agendamento diferem entre si no número de trabalhos ou ordens a serem processados, no modo como estes trabalhos chegam ao chão-de-fábrica e na ordem na qual as diferentes máquinas aparecem no processamento de suas operações individuais (CONWAY et al., 2003).

Quanto ao intervalo de chegada, os problemas são classificados como estáticos ou dinâmicos. CONWAY et al. (2003) define os problemas de natureza estática como aqueles onde as ordens a serem processadas chegam ao chão-de-fábrica de maneira simultânea. Neste caso, os recursos sempre estarão disponíveis e aguardando para serem utilizados. GRAVES (1981) explica que um agendamento estático é definido dentro de um grupo finito de especificações, sendo que não é permitida a alteração posterior de uma especificação, bem

---

---

como são proibidas as inclusões de novas exigências ao sistema. Já nos problemas dinâmicos, ordens chegam de forma intermitente, podendo ou não obedecer a uma distribuição de tempo conhecida (CONWAY et al., 2003). As especificações e exigências do sistema não são totalmente conhecidas, podendo haver distúrbios como mudanças de exigências, quebras de máquinas ou antecipações de tarefas, por exemplo (GRAVES, 1981).

Em relação à forma que as máquinas irão processar as operações de um determinado trabalho, os problemas podem ser classificados em por produto ou processo.

PINEDO (2005), GRAVES (1981) e CONWAY et al. (2003) explicam que em um sistema de manufatura por produto, também conhecido como *flow shop*, os trabalhos ou ordens devem seguir essencialmente o mesmo caminho entre as máquinas, ou seja, não é permitida a mudança na seqüência de operações. O emprego deste tipo de sistema de manufatura no modelo desenvolvido não foi considerado interessante, pois restringiria o uso da flexibilidade de seqüência, a ser discutida no item 4.1.8 do presente capítulo.

Por sua vez, os sistemas por processo, também conhecidos como funcionais ou *job shop*, podem ser definidos de maneira genérica como aqueles em que  $n$  ordens são arbitrariamente seqüenciadas em  $m$  máquinas (PINEDO, 2005) e (CONWAY et al., 2003). GRAVES (1981) explica que em problemas desta natureza não existem restrições nas exigências associadas a cada uma das operações, o que significa que uma tarefa pode ser executada em um ou em vários recursos, sem obedecer a nenhum critério específico. Neste tipo de sistema de manufatura também não há necessariamente restrições quanto:

- a) Ao uso de diferentes máquinas para o processamento de uma mesma operação, desde que haja capacidade tecnológica para isto;
- b) À mudança nas seqüências de operações, desde que respeitadas às precedências entre as mesmas; e
- c) À substituição de operações ou grupos de operações (caminhos), desde que a possibilidade tenha sido prevista no plano de processo a ser executado e que haja da capacidade tecnológica dos recursos.

Este tipo de problema é conhecido como *NP-hard* (*non deterministic polynomial-time hard*), onde a solução do problema não é determinada em tempo polinomial e sim exponencial, apresentando uma elevada complexidade de resolução por consumirem excessivo espaço de memória e elevado tempo de processamento. Normalmente este tipo de sistema está associado à alta variedade de peças e a lotes de pequeno volume, tendendo ao lote unitário (SHAFAEI e BRUNN, 2000).

---

---

A fim de modelar a estrutura de um ambiente *job shop*, é necessário apresentar algumas considerações e restrições que delimitam seu escopo. No modelo proposto neste trabalho foram utilizadas padronizações já conhecidas e aceitas na literatura para este fim (GAO et al., 2007); (DOMINIC et al., 2004); (KIM et al., 2003) e (CONWAY et al., 2003):

É dado um grupo de tarefas ou ordens, sendo que estas tarefas são independentes umas das outras;

- a) Todas as tarefas estarão simultaneamente disponíveis no tempo igual a zero;
- b) Cada tarefa tem uma seqüência de operações conhecida;
- c) Cada operação poderá ser processada em apenas um recurso por vez;
- d) Os tempos de processamento, as relações de precedência e restrições de capacidade também deverão ser conhecidas;
- e) A realização de cada operação requer um recurso ou uma máquina selecionada de um grupo de máquinas existentes no *job shop*;
- f) Cada máquina está continuamente disponível, sem nenhuma divisão significativa de escala de tempo e sem considerar indisponibilidades temporárias, causadas por quebras ou manutenções;
- g) Uma vez iniciada uma operação, ela é executada até seu término;
- h) Cada máquina pode executar apenas uma operação por vez;
- i) Após uma operação ser processada em uma máquina, é imediatamente enviada para a próxima máquina;
- j) O tempo de transporte entre as máquinas é ignorado; e
- k) Não é permitida a sobreposição de operações (*overlapping*), o cancelamento de uma operação (*preemption*), nem a sua divisão (*split*).

Dentro do contexto acima descrito, adotou-se como modelo um sistema de manufatura estático e funcional. A principal razão da escolha por este tipo de sistema está diretamente relacionada à necessidade do uso de diferentes flexibilidades para a execução dos planos de processo que serão utilizadas no desenvolvimento e validação do modelo proposto.

#### 4.1.3 Métricas para a análise dos resultados obtidos

Partindo da premissa que a saída esperada do modelo de simulação proposto é a programação, surge a necessidade de se possuir medidas concretas de se avaliar sua performance. De acordo com GRAVES (1981), existem dois tipos de critérios para se avaliar

---

um problema de agendamento, os critérios baseados em custos e os baseados em desempenho. GRAVES (1981) ainda explica que em ambientes *job shop*, em que a produção visa atender diretamente a clientes e não é voltada a estoques, é freqüente a avaliação baseada em critérios de desempenho.

Entre as inúmeras medidas de desempenho encontradas na literatura, podem ser citados o nível de utilização dos recursos produtivos, o percentual de tarefas atrasadas (*tardiness*), a média ou o máximo tempo de fluxo de trabalho (*flow time*), o tempo total de conclusão (*makespan*), datas de entrega (*due date*) e a quantidade de trabalho em processo (*work in process*) (PINEDO, 2005), (GRAVES, 1981) e (CONWAY et al., 2003). Como, por exemplo, nos trabalhos de (GAO et al., 2007) que tinha por objetivo a redução do *makespan*, minimização do carregamento de máquina e minimização do carregamento total; (WONG et al., 2006a, 2006b) e (KIM et al., 2003) redução de *makespan* e *flow time*; e (DOMINIC et al., 2004) redução de *makespan* e carregamento de máquina.

Dentro deste contexto e partindo-se da premissa que o presente trabalho tem como objetivo o estudo da correlação entre planos de processo alternativos e a programação da produção, observou-se a necessidade de se optar por ambos ou por um dos dois critérios acima citados. Adotou-se, então, uma linha de pesquisa focada nas métricas de desempenho, não fazendo parte do escopo deste trabalho os critérios baseados em custo.

Neste trabalho optou-se como critério de desempenho o tempo de fluxo de trabalho (*flow time*) e o tempo total de conclusão (*makespan*). Reforça-se que os critérios adotados visam a replicação das condições adotadas no trabalho de KIM et al. (2003) e a posterior validação do modelo aqui proposto, conforme justificado no item 3.13, que trata do conceito de validação e verificação adotado.

O tempo de fluxo  $F_i$  (*flow time*), corresponde ao tempo que o trabalho  $i$  está em processamento no chão-de-fábrica. Ou seja, é o tempo compreendido entre o momento que o trabalho  $i$  está pronto (disponível) para ser iniciado e o instante do término de seu processamento, conforme equação 4.1.

$$F_i = C_i - r_i \quad (4.1)$$

Onde para um dado trabalho  $i$  temos:

$F_i$  - tempo de fluxo

$C_i$  – tempo de conclusão

$r_i$  – tempo inicial

---

Ainda associado a este critério, pode ser calculado o tempo de fluxo médio (*mean flow time*), dado pela equação 4.2.

$$MF = \frac{1}{n} \sum_{i=1}^n Fi \quad (4.2)$$

Onde para um dado trabalho  $i$  tem-se:

$n$  – número de trabalhos presentes no chão-de-fábrica

Já o tempo total de conclusão (*makespan*) representa a diferença de tempo entre o início da execução do primeiro trabalho e a finalização do último, em um dado grupo de trabalhos, conforme equação 4.3.

$$C_{máx} = \sum_{i=1}^n Ci \quad (4.3)$$

Onde para um dado trabalho  $i$  tem-se:

$n$  – número de trabalhos presentes no chão-de-fábrica

$Ci$  – tempo total de conclusão

PINEDO (2005) explica que a minimização do tempo total de conclusão (*makespan*) está diretamente relacionada à maximização do tempo de atravessamento de um sistema de manufatura, onde este tempo refere-se à taxa de saída do sistema para um grupo finito de trabalhos. Além disto, ao se minimizar o *makespan*, maximiza-se a utilização dos recursos produtivos.

CONWAY (1967) explica que a redução do máximo tempo de fluxo (*flow time*) esta associada à redução da soma dos  $n$  tempos de *setup*. Já a sua média, é diretamente proporcional ao número de trabalhos simultâneos no chão-de-fábrica, conforme será apresentado no capítulo 8 do presente trabalho.

#### 4.1.4 Definição de peça e recurso produtivo no modelo de simulação proposto

Usualmente em modelos de manufatura, um recurso é considerado simplesmente como uma máquina, sendo que uma tarefa que pode ser executada nesta máquina é descrita como um trabalho. Trabalhos, por sua vez, podem ser formados por uma operação ou por várias

operações, que normalmente precisam ser realizadas em diferentes máquinas (PINEDO, 2005).

No protótipo desenvolvido neste trabalho, os dois principais elementos de simulação modelados são recurso produtivo e peça.

Neste modelo optou-se por considerar como recurso produtivo apenas as máquinas que compõem o layout representado. Não serão considerados como recursos os operadores, meios de transporte e outros.

Um recurso é representado pelos atributos de identificação e nome, que devem ser únicos, além do seu tipo e de sua capacidade, que é representada pelas operações que é capaz de realizar, identificadas por seus respectivos nomes e tempos. O *layout* descrito por este modelo no presente trabalho é formado por quinze diferentes recursos com capacidades diversas.

Já a peça pode ser considerada no modelo de duas formas distintas, dependendo do cenário em que se encontre:

- a) Como parte integrante de um pedido, como, por exemplo, na entrada de dados pelo usuário; e
- b) Como uma ordem, considerando que é a ordem e não a peça que interage com os recursos durante uma negociação.

Um cenário representa uma seqüência concreta de interação entre o modelo e os atores (COSSENTINO, SABATUCCI E CHELLA, 2003).

Definição 4.1  
Cenário

As definições pertinentes para a compreensão do elemento peça no modelo de simulação desenvolvido são:

- a) Pedido – corresponde a um grupo de ordens. É definido pelos atributos **identificação das peças que o compõem**; e seus respectivos **tamanhos de lote**;
- b) Ordem – é a menor unidade de um pedido. Cada ordem corresponde a uma única peça, sendo definido no modelo pelos atributos **identificação da peça**, que deve ser única; e **tamanho de lote**; e
- c) Peça – elemento definido pelos atributos **tipo** (rotacional ou prismática); **identificação**, que deve ser única; e as **operações que compõem seu plano de processo**.

Já com relação à estrutura de um elemento peça, as definições pertinentes se classificam em:

- a) Plano de processo – na visão do modelo é o conjunto de operações necessárias, com suas respectivas restrições e alternativas, para executar uma peça;
- b) Operação – corresponde a uma unidade do plano de processo. Seus atributos são **nome**, que deve ser único; **relações de precedências**; **alternativas (OR)**; e **quantidade**, que define quantas vezes a operação deverá ser realizada para a manufatura de determinada peça.

As relações de precedências indicam a seqüência exata na qual as operações que compõem o plano de processo de uma peça devem ser executadas, enquanto as alternativas (“OR”) indicam os caminhos alternativos para a manufatura de uma peça. O conceito de alternativas será detalhado no item 4.1.9 deste capítulo, que trata da flexibilidade de processo. Dentro deste contexto, um pedido representa uma ou mais ordens, sendo cada ordem refere-se a uma única peça, conforme representado na figura 4.2.

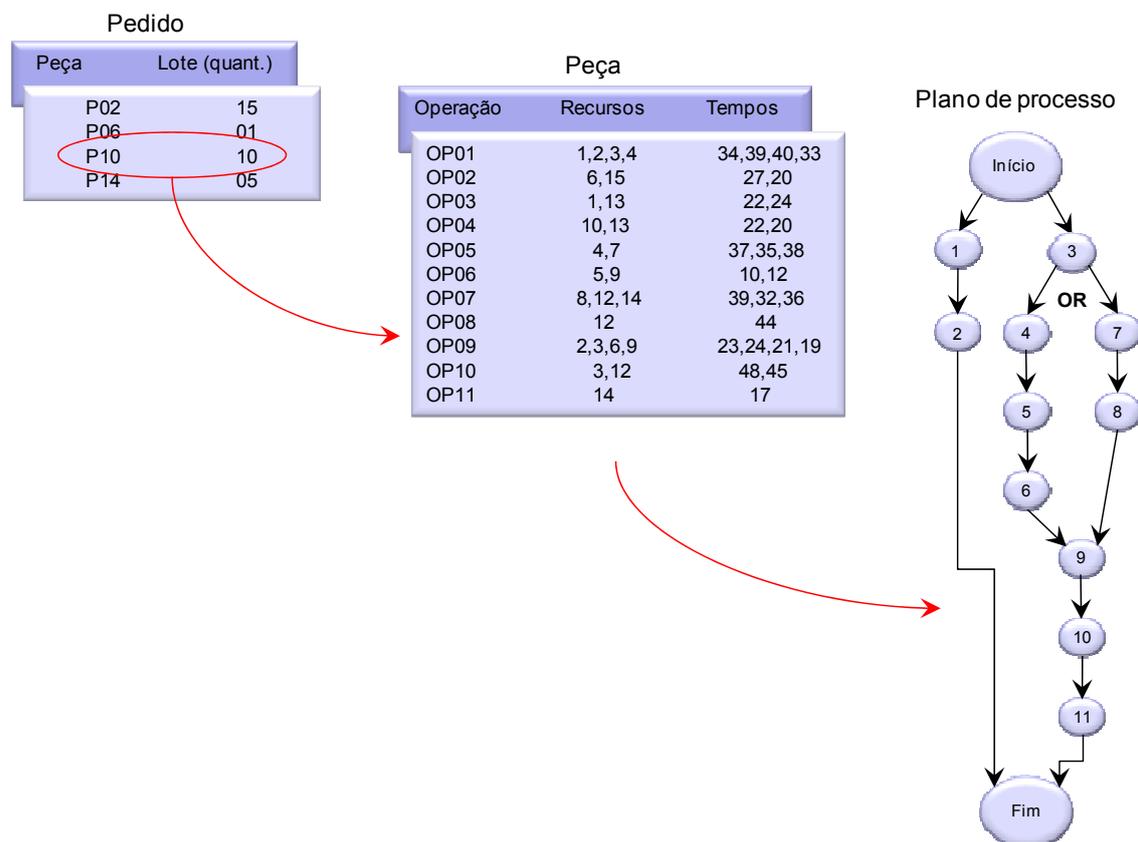


Figura 4.2 – Representação da estrutura de um pedido no modelo de simulação proposto

Cada peça é composta por um plano de processo e cada plano de processo poderá ser decomposto em uma ou mais operações, que poderão estar ou não inter-relacionadas através de relações de precedências e alternativas de processamento (“OR”).

4.1.5 Representação dos planos de processo no modelo de simulação proposto

Devido à complexidade da estrutura de planos de processos apresentada, que permite um elevado número de combinações possíveis dentro de um mesmo plano de processo, optou-se neste trabalho pela representação baseada no uso de grafos E/OU (DE MELLO e SANDERSON, 1990). A figura 4.3 apresenta um exemplo de um plano de processo utilizado para a fabricação de uma das peças utilizadas no caso de teste proposto por YEO et al. (2003) e utilizado neste trabalho.

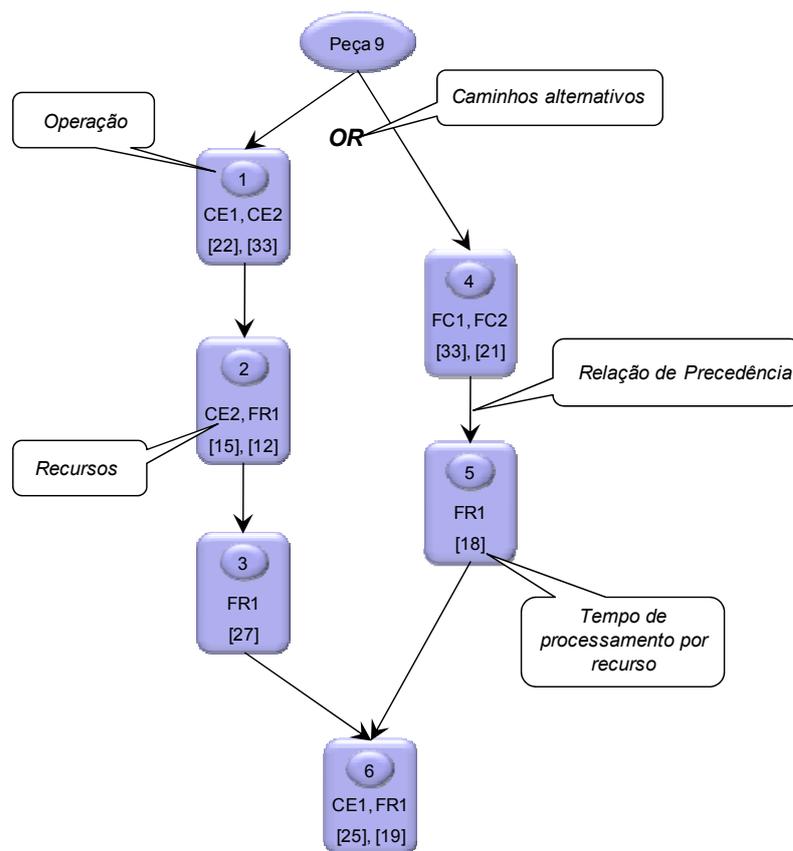


Figura 4.3 – Representação de um plano de processo no modelo proposto

Neste tipo de representação são identificadas as operações que compõem um plano de processo, bem como os recursos nos quais as operações serão executadas e seus respectivos tempos de processamento. Já as relações de precedência entre operações são indicadas por flechas que partem da operação precedente para a operação obrigatoriamente posterior. Finalmente poderá existir ou não a indicação “OR” no plano de processo, o que indica a existência ou não de flexibilidade de processo na peça analisada.

É importante notar que o número elevado de combinações permitidas nos planos de processo é consequência direta dos quatro diferentes tipos de flexibilidades utilizadas no

---

modelo (de processo, agendamento, operações e de máquina), a serem descritas nos próximos itens, e que um aumento no grau de qualquer uma destas flexibilidades implica em um aumento direto no grau de complexidade do espaço de solução.

#### 4.1.6 Definição de flexibilidade no modelo de simulação proposto

Flexibilidade em manufatura é definida, em termos gerais, como a habilidade que um sistema possui para se adaptar rapidamente a quaisquer mudanças provenientes de fatores considerados relevantes, entre os quais o produto, o processo, o carregamento ou ainda a quebra de uma máquina (BEACH et al., 2000); (TSUBONE e HORIKAWA, 1999) e (SETHI e SETHI, 1990).

Uma visão do papel da flexibilidade dentro da cadeia de suprimentos é dada por ZHANG et al. (2003), CHEN et al. (1992) e LEONG et al. (1990), que definem a flexibilidade de manufatura como a habilidade que um sistema tem de produzir uma variedade de produtos, satisfazendo a demanda exigida e mantendo alta qualidade e elevado desempenho.

Ao permitir um maior grau de flexibilidade, um sistema de manufatura pode, de acordo com BENGTTSSON (2001), se adaptar melhor às oscilações de demanda, além de possibilitar um melhor aproveitamento de sua capacidade instalada. A flexibilidade de manufatura permite novos graus de liberdade ao sistema, tornando-se posteriormente uma vantagem competitiva (KOSTE, 2000).

JAIKUMAR (1986) enfatiza que a flexibilidade é restrita ao seu domínio específico, e que este domínio pode ser definido em função dos seus produtos e processos. Já BEACH (2000) explica que a mesma pode ser vista de modo multidimensional e não como uma variável independente que pode ser definida e mensurada isoladamente. BENGTTSSON (2001) por sua vez diz que a flexibilidade de um sistema de manufatura é formada pela soma de todas as flexibilidades que o sistema contém.

Porém, mapear as flexibilidades contidas em um sistema de manufatura pode ser uma tarefa confusa, visto que a nomenclatura da mesma não é um consenso. BROWNE et al. (1984) sugerem uma taxonomia que classifica em oito diferentes categorias as flexibilidades de manufatura, as quais são: máquina, processo, produto, roteamento, volume, expansão, operações e produção. Já SETHI e SETHI (1990) mapearam em torno de 50 diferentes termos relativos a tipos de flexibilidade, muitos dos quais correspondem a uma mesma definição.

---

Neste trabalho optou-se por adotar a nomenclatura proposta por BENJAAFAR e RAMAKRISHNAN (1996), tendo-se o cuidado de relacionar as definições utilizadas no modelo a outros autores, independentemente da nomenclatura adotada pelos mesmos. Esta classificação divide as flexibilidades em dois tipos: relativas ao produto, ou ao processo.

As flexibilidades relativas ao produto tratam de toda a variedade de opções de manufatura inerentes ao produto, como, por exemplo, a presença de alternativas em planos de processo. Já as flexibilidades relativas ao processo, tratam da capacidade do processo em se ajustar a diferentes condições de operações, ou seja, de assumir diferentes funções, como no caso de máquinas multifuncionais.

Neste trabalho são considerados três tipos de flexibilidades relativas ao produto: de operação, seqüência e processo. A escolha destas flexibilidades visa a replicação das condições adotadas no trabalho de KIM et al. (2003) e a posterior validação do modelo aqui proposto, conforme justificado no item 3.13, que trata do conceito de validação e verificação adotado.

Além das flexibilidades de produto, foi adotado um tipo de flexibilidade relativa ao processo de manufatura, que é a flexibilidade de máquina, a ser descrita no item 4.1.10 do presente capítulo. Esta flexibilidade relativa ao processo de manufatura, ao contrário das flexibilidades de processos citadas anteriormente, não faz parte das condições adotadas por KIM et al. (2003), e é utilizada no presente trabalho para contribuir no estudo da correlação entre planos de processos alternativos e programação da produção. A figura 4.4 apresenta a estrutura de hierarquia de flexibilidades utilizadas no modelo de simulação apresentado neste trabalho.

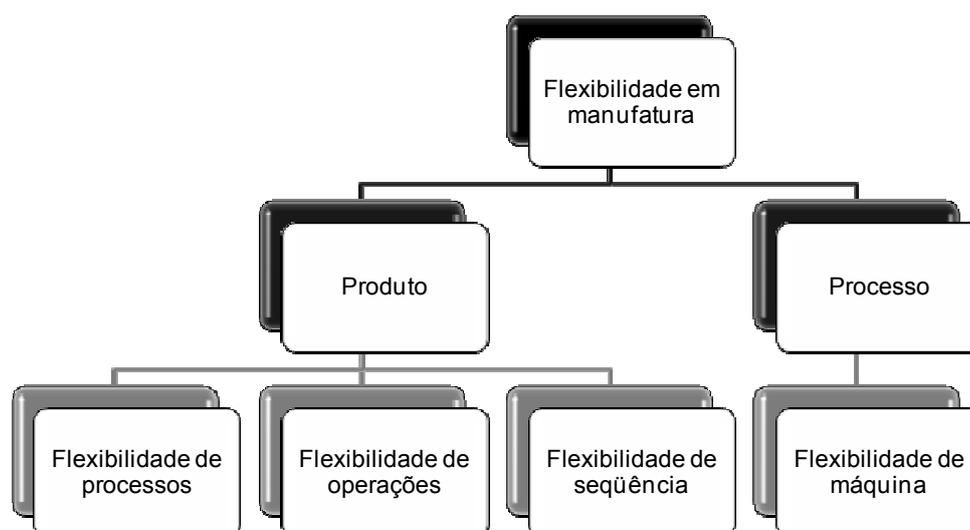


Figura 4.4 – Classificação das flexibilidades utilizadas no modelo apresentado neste trabalho  
FONTE: Adaptado de Benjaafar e Ramakrishnan (1996)

---

Apresenta-se, na seqüência, as definições de cada uma das flexibilidades utilizadas no modelo, sua importância em um sistema de manufatura, algumas das métricas existentes e uma explicação da métrica adotada para cada um dos tipos. Reforça-se que um estudo mais aprofundado a respeito das métricas utilizadas para a classificação dos diferentes tipos de flexibilidades utilizadas neste modelo não faz parte do escopo deste trabalho. Um extenso trabalho neste sentido pode ser encontrado em SHEWCHUK e MOODIE (1998), bem como em BENJAAFAR e RAMAKRISHNAN (1996) e SETHI e SETHI (1990).

#### 4.1.7 Flexibilidade de operações

Flexibilidade de operações, também chamada de *routing flexibility*, é definida por ZHANG et al. (2003); BEACH et al. (2000) e SETHI e SETHI (1990) como a habilidade de um sistema de manufatura em lidar com uma quebra de máquina a fim de continuar produzindo um dado grupo de peças. Segundo os autores, esta habilidade existe somente se as peças envolvidas permitirem o uso de rotas alternativas para o seu processamento através do sistema de manufatura.

Porém, é importante observar que diferentes rotas podem ser descritas através do uso de diferentes máquinas, operações ou seqüências, o que pode gerar várias interpretações para a definição acima. Neste contexto, fica clara a necessidade de se especificar de que forma a rota alternativa será obtida.

Neste trabalho optou-se por utilizar a definição apresentada por BENJAAFAR e RAMAKRISHNAN (1996), e LIN e SOLBERG (1992), que relacionam a flexibilidade de operações ao uso de máquinas alternativas e seus respectivos tempos de processamento.

A aplicação deste tipo de flexibilidade em um sistema de manufatura possibilita ao mesmo lidar com distúrbios ocasionados, por exemplo, por quebras de máquinas, trazendo robustez ao mesmo, ou seja, não ocorre uma redução drástica na taxa de produção do sistema quando da mudança da disponibilidade dos recursos (ZHANG et al. 2003), (SETHI e SETHI, 1990), (PARKER e WIRTH, 1994) e (BROWNE et al., 1984).

Robustez é definida por PARKER e WIRTH (1994) como a capacidade de um sistema de manufatura absorver mudanças e continuar operando com a mínima turbulência. Os autores ainda relacionam que o aumento do nível de flexibilidade de operações em um sistema de manufatura está diretamente relacionado ao aumento das rotas de processamento, ao número de operações que cada máquina pode realizar e à disponibilidade das máquinas.

---

Inúmeras métricas foram sugeridas para classificar o nível de flexibilidade de operações em um sistema de manufatura. SETHI e SETHI (1990) sugerem como métrica o número médio de possíveis formas de cada tipo de peça que pode ser processada em um dado sistema. Já CHATTERJEE et al. (1987) e CHUNG e CHEN (1989) usam como métrica a relação entre o número de operações concluídas e o número de operações alternativas para a execução de uma peça. TSUBONE e HORIKAWA (1999) empregam a relação entre as operações para a execução de um determinado plano de processo que podem ser processadas em uma máquina específica, pelo número de operações que podem ser processadas em máquinas alternativas.

Conforme citado no item 4.1.6, que trata da definição de flexibilidade no modelo de simulação proposto, a métrica adotada para mensurar a flexibilidade de operações no modelo proposto é a mesma adotada por KIM et al. (2003) e é dada pela relação entre a média do número de máquinas alternativas disponíveis em cada uma das operações e o número total de operações que irá compor o plano de processo. Isto faz com que o número de planos de processo disponíveis para uma dada peça tenda a aumentar exponencialmente com o aumento do número de máquinas alternativas.

De acordo com o critério citado acima, pode-se classificar um plano de processo de uma dada peça em três diferentes níveis de flexibilidade de operações: baixo (valores entre 1,0-2,0); médio (valores entre 2,1-3,5) e alto (acima de 3,6 inclusive). A figura 4.5 apresenta um exemplo de plano de processo com máquinas alternativas por operações.

Na figura 4.5 primeiramente são mapeadas todas as máquinas capazes de executar uma determinada operação (por exemplo, a operação 1 pode ser processada nas máquinas CE1 e CE2). Isto é repetido para cada uma das operações que compõem o plano de processo. Em seguida é feito o somatório do número total de máquinas alternativas presentes no plano de processo para cada operação. Este número é posteriormente dividido pelo número total de operações, neste caso, seis, calculando-se então o nível de flexibilidade de operações do mesmo, conforme apresentado nas equações 4.4 a 4.6.

$$\text{Flexibilidade de operações} = \frac{\sum (\text{n. de máquinas alternativas por operação})}{\text{n. total de operações do plano de processo}} \quad (4.4)$$

$$\text{Flexibilidade de operações} = \frac{\sum (2+2+1+1+1+2)}{6} \quad (4.5)$$

$$\text{Flexibilidade de operações} = 1,5 \text{ (nível de flexibilidade de operação baixo)} \quad (4.6)$$

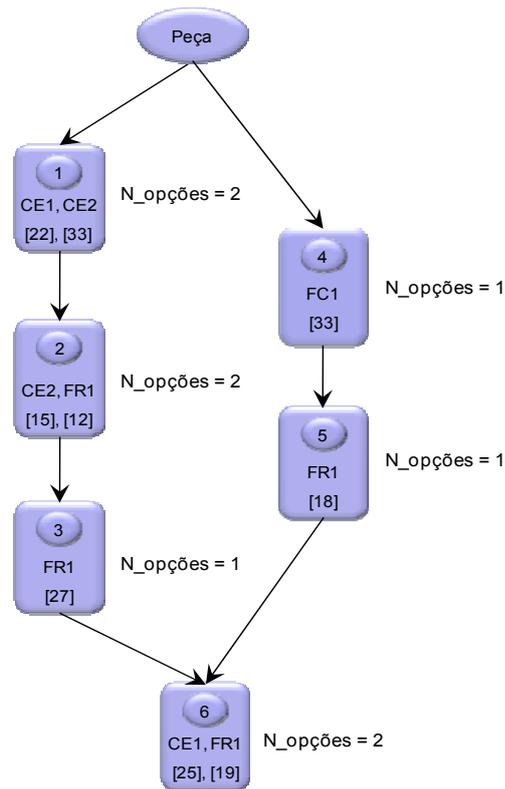


Figura 4.5 – Exemplo de plano de processo com o número de máquinas alternativas por operação

Além de proporcionar robustez, conforme já citado, outra vantagem do uso da flexibilidade de operações é a de permitir um melhor balanceamento dos carregamentos nas máquinas que compõem o sistema de manufatura (SETHI e SETHI, 1990), o que resulta em um maior aproveitamento da capacidade das máquinas instaladas e no aumento do volume de produção (ZHANG et al., 2003).

#### 4.1.8 Flexibilidade de seqüência

De acordo com BENJAAFAR e RAMAKRISHNAN (1996) e BROWNE et al. (1984), flexibilidade de seqüência pode ser definida como a habilidade de mudar a seqüência em que as operações que compõem um plano de processo serão realizadas, desde que sejam respeitadas suas relações de precedência.

Ao contrário de outras flexibilidades, a flexibilidade de seqüência não é necessariamente dependente das capacidades de processo, sendo freqüentemente uma

propriedade da peça, refletindo as relações intrínsecas entre as *features*<sup>7</sup> de manufatura da mesma (BENJAAFAR e RAMAKRISHNAN, 1996).

O principal propósito da utilização da flexibilidade de seqüência em um sistema de manufatura é o de elevar a taxa de ocupação das máquinas disponíveis. Além disto, uma maior disponibilidade de máquinas e operações diminui o tempo de espera entre operações, diminuindo conseqüentemente o *flow time* da peça (BENJAAFAR e RAMAKRISHNAN, 1996).

PARKER e WIRTH (1994) mencionam que o aumento do nível de flexibilidade de seqüência está diretamente relacionado ao aumento das operações intercambiáveis em relação ao número total de operações em um plano de processo de uma peça, e também ao aumento do número de máquinas capazes de executar estas operações intercambiáveis.

Entre as métricas para classificar o nível de flexibilidade de seqüência em um sistema de manufatura destaca-se aquela definida por BENJAAFAR e RAMAKRISHNAN (1996), que diz que a flexibilidade de seqüência de uma peça é dada pela divisão do número de seqüências possíveis pelo número total de operações.

A métrica adotada neste modelo é a mesma utilizada por KIM et al. (2003) e é apresentada na equação 4.7.

$$Flexibilidade\_de\_sequencia = 1 - \frac{soma\_máquinas\_alternativas}{\binom{n(n-1)}{2}} \quad (4.7)$$

Para a aplicação desta fórmula, primeiramente é selecionada a seqüência de operações na qual a peça será processada, escolhendo-se para o cálculo aquela com o maior número de operações possíveis. Após esta seleção, é feita a soma de todas as alternativas de máquinas possíveis de cada uma das operações que compõem o plano de escolhido. Esta soma não deve levar em consideração a primeira operação do plano, somente suas operações subseqüentes. Finalmente, é calculado o valor de  $n$ , que é o número total de operações necessárias para a manufatura da peça no plano de processo escolhido.

No exemplo da figura 4.5 duas seqüências de operações diferentes podem ser adotadas para a manufatura da peça: a primeira é composta pelas operações (1, 2, 3, 4, 5 e 6) e a segunda pelas operações (4, 5, 1, 2, 3 e 6).

---

<sup>7</sup> Uma *feature* de manufatura pode ser entendida como uma forma geométrica definida por um conjunto de parâmetros que compreende um significado especial para engenheiros de projeto e fabricação. (SHAH e MÄNTYLÄ, 1995).

A soma de máquinas alternativas pela primeira seqüência de operações é igual a sete, recursos CE2 e FR1 para a operação 2, recurso FR1 para a operação 3, recurso FC1 para a operação 4, recurso FR1 para a operação 5 e recursos CE1 e FR1 para a operação 6. Lembrando que a primeira operação, neste caso operação 1, deve ser desconsiderada para o cálculo.

Aplicando-se a mesma lógica a segunda opção de seqüência de operações (4, 5, 1, 2, 3 e 6), é obtida uma soma de máquinas alternativas igual a oito.

Como para o cálculo da flexibilidade de seqüência deve-se usar a seqüência de com o maior número de operações possíveis, opta-se pelo uso da segunda rota alternativa (4, 5, 1, 2, 3 e 6), ou seja, um número de máquinas alternativas iguais a oito, para um número de operações mínimas para o processamento da peça ( $n$ ) igual a seis. Desta forma, o cálculo do nível de flexibilidade de seqüência para a peça é dado nas equações 4.8 e 4.9:

$$\text{Flexibilidade\_de\_sequencia} = 1 - \frac{8}{\left(\frac{6(6-1)}{2}\right)} \quad (4.8)$$

$$\text{Flexibilidade\_de\_sequencia} = 0,47(\text{nível\_da\_flexibilidade\_de\_sequencia\_médio}) \quad (4.9)$$

De acordo com este critério de classificação, pode-se ter três diferentes níveis de flexibilidade de seqüência no plano de processo de uma dada peça: baixo (valores entre 0-0,40); médio (valores entre 0,41-0,70) e alto (entre 0,71-1,0), onde um valor próximo de 1 indica um menor número de relações de precedências entre as operações, o que aumenta o número de alternativas no plano de processo.

#### 4.1.9 Flexibilidade de processo

Flexibilidade de processo é definida por BENGSTOON (2001) como a habilidade de se produzir uma peça de diferentes formas, através da mudança na seqüência de suas operações ou sua substituição. SETHI e SETHI (1990) explicam que a flexibilidade de processo pode ser vista como uma propriedade da peça, significando que a mesma pode ser produzida através de planos de processo alternativos. Neste caso, um plano de processo alternativo pode ser obtido através da troca ou substituição de algumas operações.

SETHI e SETHI (1990) explicam, ainda, que o aumento do nível de flexibilidade de processo está diretamente relacionado ao uso de componentes padronizados, ao projeto de superfícies que permitam fácil acesso por diferentes operações e, quando possível, ao uso de projetos modulares. A figura 4.6 apresenta um exemplo de uma peça que permite o uso de planos de processo alternativos, cuja representação no gráfico é feita pela palavra *OR*.

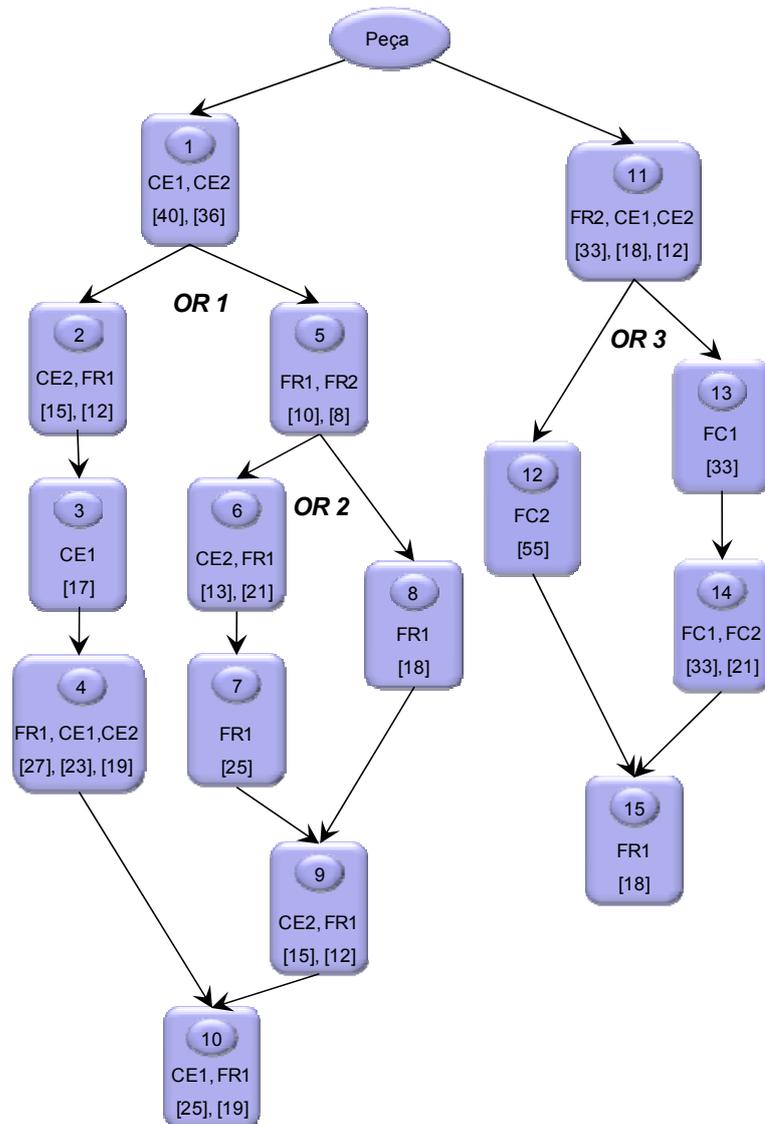


Figura 4.6 – Exemplo de uma peça com seus planos de processo alternativos

O nível flexibilidade desta peça é igual ao número de processos alternativos existentes, que neste caso será igual a doze, sendo eles:

- a) Primeiro plano de processo – composto pelas operações 1, 2, 3, 4, 10, 11, 12 e 15;
- b) Segundo plano de processo – composto pelas operações 1, 2, 3, 4, 10, 11, 13, 14 e 15;

- 
- c) Terceiro plano de processo – composto pelas operações 1, 5, 6, 7, 9, 10, 11, 12 e 15;
  - d) Quarto plano de processo – composto pelas operações 1, 5, 6, 7, 9, 10, 11, 13, 14 e 15;
  - e) Quinto plano de processo – composto pelas operações 1, 5, 8, 9, 10, 11, 12 e 15;
  - f) Sexto plano de processo – composto pelas operações 1, 5, 8, 9, 10, 11, 13, 14 e 15;
  - g) Sétimo plano de processo – composto pelas operações 11, 12, 15, 1, 2, 3, 4 e 10;
  - h) Oitavo plano de processo – composto pelas operações 11, 13, 14, 15, 1, 2, 3, 4 e 10;
  - i) Nono plano de processo – composto pelas operações 11, 12, 15, 1, 5, 6, 7, 9 e 10;
  - j) Décimo plano de processo – composto pelas operações 11, 13, 14, 15, 1, 5, 6, 7, 9 e 10;
  - k) Décimo primeiro plano de processo – composto pelas operações 11, 12, 15, 1, 5, 8, 9 e 10; e
  - l) Décimo segundo plano de processo – composto pelas operações 11, 13, 14, 15, 1, 5, 8, 9 e 10.

De acordo com KIM et al. (2003), BENGSTOON (2001) e SETHI e SETHI (1990), a flexibilidade de processo pode ser medida diretamente através do número de planos de processos alternativos que a peça possui.

De acordo com este critério de classificação, KIM et al. (2003) estabelece três diferentes níveis de flexibilidade de processo para uma peça dada: baixo (valores entre 1,0-2,9); médio (valores entre 3,0-5,9) e alto (valores acima de 6,0 inclusive). Como o número de planos de processo da peça é igual a sua flexibilidade de processo, ou seja, igual a seis, considera-se que a mesma possui um nível de flexibilidade de processo alto.

#### 4.1.10 Flexibilidade de máquina

De acordo com TSUBONE e HORIKAWA (1999), flexibilidade de máquina é a habilidade de uma máquina realizar diferentes operações requeridas para uma determinada peça ou grupo de peças, através da troca rápida de ferramentas. Outra definição, mais generalista, é estabelecida por GUPTA (1993), CHEN et al. (1992), SETHI e SETHI (1990) e ZHANG et al. (2003), para os quais a flexibilidade de máquina é a habilidade de um equipamento em realizar diferentes operações de forma econômica e efetiva. Pode, ainda, ser

---

definida como a habilidade de mudanças entre operações com o mínimo de atrasos ou *setups*<sup>8</sup> (PARKER e WIRTH, 1994).

Em uma perspectiva mais abrangente, a flexibilidade de máquina em um sistema de manufatura provê condições para que outras flexibilidades se tornem economicamente viáveis. RANTA (1998) diz que se as máquinas somente permitirem *setups* difíceis e caros, nenhum tipo de função exercida por um *software* pode ajudar, fazendo com que a flexibilidade de máquina se torne vital para a viabilidade econômica das outras flexibilidades inerentes ao sistema de manufatura em questão.

Em uma perspectiva em termos do seu próprio nível de contribuição, a flexibilidade de máquina dá condições para a diminuição do tamanho dos lotes de fabricação (RANTA, 1988), menores inventários e maior utilização da capacidade instalada (HUTCHINSON, 1984), além de menor *lead time*, tanto para os produtos já sendo fabricados, quanto na introdução de novos produtos (TSUBONE e HORIKAWA, 1999) e (CARTER, 1986).

De acordo com SETHI e SETHI (1990), o primeiro aspecto que deve ser mensurado na flexibilidade de máquina é o número de operações diferentes que uma máquina pode executar sem a necessidade de custo ou de tempo adicionais. Uma visão econômica da flexibilidade de máquina é apresentada na métrica de SON e PARK (1987), que sugerem uma relação entre o valor agregado da quantidade de produtos completados por um sistema de manufatura e o custo das máquinas ociosas. LAM (1988) sugere uma métrica baseada na relação entre o investimento para o lançamento de um novo produto e o valor investido na compra do equipamento para executá-lo.

A visão da métrica a ser utilizada para classificar a flexibilidade de máquina que orienta o presente trabalho é baseada na linha sugerida por SETHI e SETHI (1990), onde somente o aspecto tecnológico, e não o econômico é considerado. Neste contexto, TSUBONE e HORIKAWA (1999) sugerem como métrica a relação entre o número de operações que podem ser processadas sem requerer mudança de *setup* e o número total de operações processadas, para uma determinada máquina, conforme apresentado na equação 4.10.

$$M_{jm} = \frac{Rm}{Nm} \text{ para } m = 1, 2, \dots, M \quad (4.10)$$

Onde:

---

<sup>8</sup> Máquinas precisam freqüentemente ser reconfiguradas ou limpas entre o processamento de diferentes trabalhos ou operações. Este tipo de atividade é conhecido como *changeover* ou *setup* (PINEDO, 2005).

$R_m$  = número de operações de uma mesma peça que podem ser processadas na máquina  $m$  sem necessidade de novo *setup*;

$N_m$  = número total de operações de uma mesma peça que podem ser processadas na máquina  $m$ .

Observou-se que apesar de terem proposto a relação acima, TSUBONE e HORIKAWA (1999) não especificam critérios para classificar os diferentes níveis de flexibilidade de máquina. Neste trabalho arbitrou-se que uma máquina deverá estar situada em três diferentes níveis de flexibilidade: baixo (valores entre 0,15-0,25); médio (valores entre 0,26-0,35) e alto (valores acima de 0,36 inclusive). A figura 4.7 apresenta duas peças que compõem um mesmo pedido de produção.

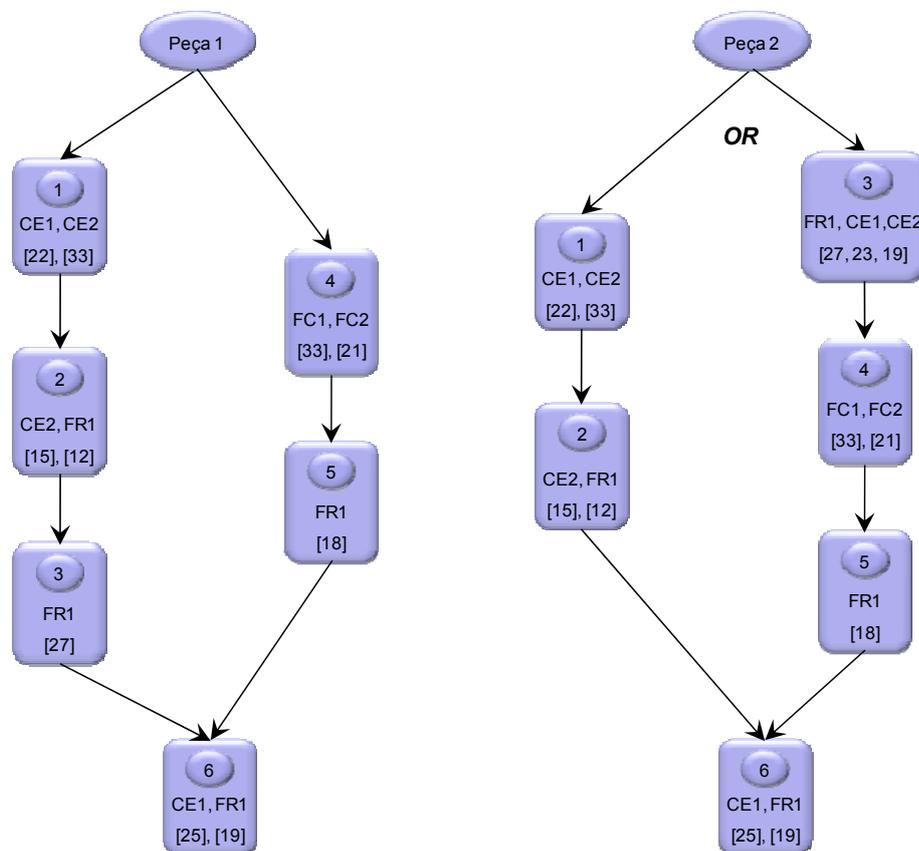


Figura 4.7 – Exemplos de planos de processo que permitem o cálculo de flexibilidade de máquina

A partir dos planos de processo apresentados na figura 4.7, pretende-se a título de exemplo calcular a flexibilidade de máquina relativa ao recurso FR1. É importante observar que para este cálculo sempre deverão ser consideradas todas as peças de uma família, grupo ou pedido, que sejam consideradas relevantes para o recurso analisado.

Para se obter o nível de flexibilidade de máquina, primeiramente é feita a soma de todas as operações que podem ser processadas no recurso FR1 sem necessidade de um novo *setup*. A figura 4.8 apresenta as operações relativas a este passo.

Observa-se que na peça 1, as operações 2, 3 e 4 podem ser executadas no recurso FR1 sem a necessidade de um novo *setup*, ou seja, podem ser agrupadas. Já na peça 2, respeitando-se a flexibilidade de processo (OR) do plano de processo, tem-se a possibilidade de agrupamento das operações 2 e 6, e das operações 5 e 6. É importante notar que todas as operações são consideradas para este cálculo, não sendo considerada nenhuma escolha prévia do caminho de processamento. Porém, deve-se subtrair uma vez a operação 6, que está duplicada.

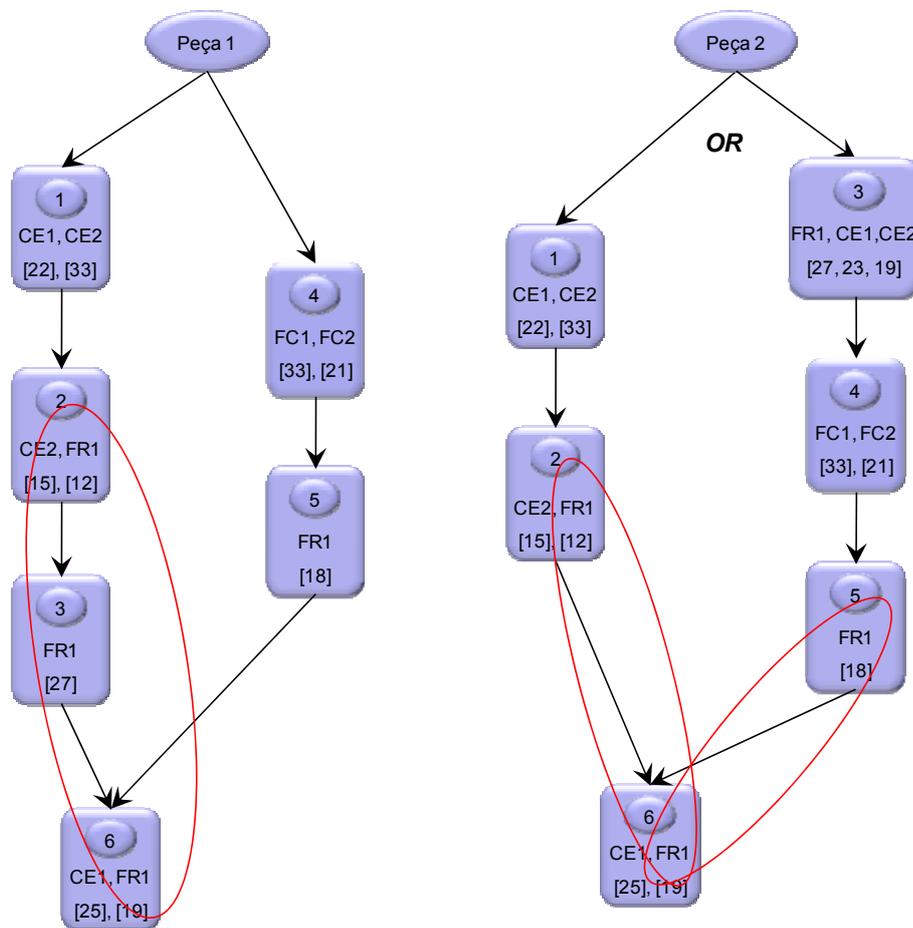


Figura 4.8 – Operações nas peças 1 e 2 passíveis de agrupamento, sem *setup* adicional

Assim, a soma das operações que podem ser agrupadas pelo recurso FR1 nas peças 1 e 2 corresponde a seis. Já o total das operações passíveis de execução no mesmo recurso é igual a oito. O valor da flexibilidade de máquina é apresentado na equação 4.11.

---

$$M_{fm} = \frac{R_{FR1}}{N_{FR1}} = \frac{6}{8} = 0,75 \text{ (nível de flexibilidade de máquina alto)} \quad (4.11)$$

De acordo com o contexto apresentado acima, a flexibilidade de máquina neste trabalho está diretamente relacionada ao tempo de *setup*, que por sua vez está diretamente relacionado ao tempo total de conclusão (*makespan*).

De acordo com PINEDO (2005), ao minimizar-se o *makespan* freqüentemente minimiza-se o tempo de *setup*, pois o *makespan* considera todos os tempos envolvidos no processamento de uma peça, como preparação de máquinas e a fixação de peças entre diferentes operações.

Seqüências dependentes de tempo de *setup*, de acordo com PINEDO (2005) e CONWAY (1967), são aquelas onde não é aceitável assumir que o tempo requerido de preparo antes da próxima tarefa a ser processada seja independente da tarefa que está sendo concluída em um dado recurso. Um exemplo disto são as operações de pintura, onde antes da troca de cada cor de tinta a ser utilizada, é necessário efetuar a limpeza dos tanques de tinta, a fim de se eliminar resquícios da cor anterior.

A dependência ou não na seqüência de operações influencia no tempo de processamento do trabalho. Em sistemas onde o tempo de *setup* é independente da seqüência, este pode ser incluído diretamente na parcela de tempo de processamento da operação. Porém, ao se constatar a dependência, é necessário criar o que é chamado de matriz de *setup*, a qual representa o tempo requerido para mudança de um trabalho *i* para um trabalho *j* em um mesmo recurso (CONWAY, 1967).

Neste trabalho, o tempo de *setup* que irá compor a matriz de *setup* relativa a uma peça será dividido entre *setup* de preparação e fixação. *Setup* de preparação é aquele que diz respeito à preparação do recurso para o processamento de uma peça, sendo um exemplo disto a troca de ferramentas. Já o *setup* de fixação refere-se à fixação da peça, em um dado recurso, a fim de permitir a execução de uma ou mais de suas operações.

Finalmente, é importante observar que, de acordo com CONWAY (1967), muitas vezes situações onde as seqüências são dependentes de tempo de *setup* são tratadas como critério dominante para se gerar ou avaliar um agendamento, conforme será visto no capítulo 6 no item que trata do protocolo de negociação desenvolvido para o presente trabalho.

#### 4.1.11 Requisitos do domínio

---

Ainda como parte do modelo de requisitos do sistema na metodologia PASSI, é necessário o registro de todos os tópicos relativos ao que o modelo de simulação proposto deve executar e sob quais condições deve fazê-lo (WAZLAWICK, 2004). Este documento é chamado de requisitos do domínio e é apresentado, na tabela 4.1.

Um requisito do domínio representa uma característica que o sistema deve apresentar. Esta característica pode ser funcional, descrevendo interações entre o sistema e seu ambiente de forma independente de sua implementação. Ou pode ser não-funcional, como, por exemplo, uma restrição a sua performance (COSSENTINO, SABATUCCI e CHELLA, 2003).

**Definição 4.2**  
Requisito do domínio

Na tabela 4.1 são apresentados, primeiramente, os campos correspondentes à especificação do requisito funcional, como seu código, nome, uma breve descrição do mesmo e sua categoria funcional, evidente ou oculto ou não. Adicionalmente a cada requisito funcional, poderá haver ou não uma lista de requisitos não-funcionais, com seu respectivo código, nome, restrição, categoria, seu nível de obrigatoriedade e permanência (WAZLAWICK, 2004). Uma revisão abrangente a respeito desta documentação encontra-se em WAZLAWICK, 2004.

Tabela 4.1 – Documento dos requisitos do domínio

REQUISITOS DO DOMÍNIO				
F1 Receber dados do usuário				Oculto ( )
Descrição: Permitir que o usuário entre com dados de pedido no modelo de simulação.				
Requisitos Não Funcionais:				
Nome	Restrição	Categoria	Desejável	Permanente
NF 1.1 Garantir que o usuário entre com dados coerentes	O usuário somente poderá entrar com combinações dos dados fornecidos.	Segurança		X
NF 1.2 Garantir que o usuário não modifique dados existentes no modelo de simulação	O usuário não terá permissão para incluir ou excluir dados. O usuário somente poderá parametrizar os valores pré-definidos pelo sistema.	Segurança		X

F2 Armazenar dados				Oculto ( x )
Descrição: Registrar permanentemente os valores medidos pela simulação, como makespan e flow time				
Requisitos Não Funcionais:				
Nome	Restrição	Categoria	Desejável	Permanente
NF 2.1 Armazenar dados relativos a cada simulação.	Armazenar em uma base de dados os valores obtidos em cada uma das simulações, como makespan e flow time.	Especificação		X

F3 Armazenar dados transitórios				Oculto ( x )
Descrição: Registrar de forma transitória uma lista com a identificação das peças que fazem parte de uma rodada de simulação.				
Requisitos Não Funcionais:				
Nome	Restrição	Categoria	Desejável	Permanente
NF 3.1 Armazenar dados transitórios relativos a cada uma das simulações.	Armazenar, durante o tempo de simulação, o nome de cada uma das peças que compõem o pedido.	Especificação		X
NF 3.2 Resetar dados transitórios	Assegurar que os dados transitórios são resetados ao final de cada simulação.	Especificação		X

F4 Apresentar dados ao usuário				Oculto ( )
Descrição: Permitir que o usuário visualize os dados armazenados de forma permanente.				
Requisitos Não Funcionais:				
Nome	Restrição	Categoria	Desejável	Permanente
NF 4.1 Modos de visualização	Os dados armazenados serão disponibilizados para visualização na forma de gráficos e tabelas temporais. A visualização deve ser configurável pelo usuário. Os parâmetros sets, setups e número de simulação, devem ser alteráveis, além de permitir a escolha entre diferentes flexibilidades.	Interface		X

F5 Permitir parametrizações por parte do usuário				Oculto ( )
Descrição: Permitir que o usuário parametrize variáveis do modelo antes de cada simulação.				
Requisitos Não Funcionais:				
Nome	Restrição	Categoria	Desejável	Permanente
NF 5.1 Garantir que o usuário parametrize as variáveis de simulação de forma coerente.	O usuário não poderá entrar com dados de formatação incorreta e valores com magnitude incoerente. Por formatação entendem-se casas decimais, algarismos, letras, etc.	Interface		X

F6 Permitir a representação do mundo real				Oculto ( x )
Descrição: Permitir a representação de entidades heterogêneas que representem peças e/ou recursos com diferentes capacidades.				

F7 Realizar operações matemáticas				Oculto ( x )
Descrição: Permitir operações matemáticas básicas (soma, subtração, multiplicação, divisão) e cálculos estatísticos a partir das medidas obtidas.				
Requisitos Não Funcionais:				
Nome	Restrição	Categoria	Desejável	Permanente
NF 7.1 Impossibilitar cálculos absurdos.	O modelo não deve permitir a realização de operações absurdas, alertando de sua ocorrência.	Segurança		X

F8 Escalabilidade e robustez				Oculto ( x )
Descrição: Possibilitar escalabilidade e robustez ao modelo de simulação proposto				
Requisitos Não Funcionais:				
Nome	Restrição	Categoria	Desejável	Permanente
NF 8.1 Permitir a inclusão de novas entidades	A inclusão de novas entidades, representando diferentes recursos ou ordens de serviços, será efetuada sem o comprometimento das relações pré-estabelecidas.	Tolerância às falhas		X
NF 8.2 Permitir a exclusão de entidades	A remoção de entidades, representando diferentes recursos ou ordens de serviços, será efetuada sem o comprometimento das relações pré-estabelecidas	Tolerância às falhas		X

F9 Permitir o uso de computação paralela				Oculto ( x )
Descrição: permitir o uso de várias unidades de processamento ou processadores para executar uma computação de forma mais rápida.				
Requisitos Não Funcionais:				
Nome	Restrição	Categoria	Desejável	Permanente
—	—	—	—	—

F10 Autonomia				Oculto ( x )
Descrição: O modelo deve operar sem intervenção de humanos após a parametrização e entrada de dados.				
Requisitos Não Funcionais:				
Nome	Restrição	Categoria	Desejável	Permanente
—	—	—	—	—

F11 Tomada de decisões				Oculto ( x )
Descrição: O modelo de simulação proposto possuir algum tipo de inteligência que permita tomada de decisão de forma autônoma, ou seja, independente da intervenção do usuário.				
Requisitos Não Funcionais:				
Nome	Restrição	Categoria	Desejável	Permanente
—	—	—	—	—

A fim de continuar com a descrição funcional do modelo, são mapeados seus grandes processos (ou casos de uso), que são suas principais atividades de consulta e transformação da informação (WAZLAWICK, 2004). A tabela 4.2 apresenta o nome do processo e uma breve descrição.

Tabela 4.2 – Descrição geral dos casos de uso

LISTAGEM GERAL DOS CASOS DE USO	
Nome	Descrição
Iniciar programa	O usuário inicializa o software e coloca o modelo de simulação em funcionamento.
Disponibilizar planos de processo	Solicita o resgate das informações de planos de processo em uma base de dados.
Informar planos de processos	Recupera e disponibiliza ao usuário, através de uma interface gráfica, as informações relativas aos planos de processo.
Entrada do pedido	O usuário acessa uma interface gráfica onde é feita a seleção das peças que compõem o pedido. O usuário somente pode escolher entre peças e opções de tamanho de lote pré-determinadas. Após a seleção de cada peça, seu plano de processo com precedências é exibido, somente em modo de visualização.

Iniciar a identificação da simulação	É necessário que os recursos sejam avisados que a simulação tem início e que enviem o tempo de setup de cada uma das peças que compõem o pedido.
Parametrizar simulação	O usuário seleciona e disponibiliza os dados de simulação: flexibilidades, recursos, sets e setups.
Converter pedidos em ordens por peça	Após a entrada do pedido, o mesmo é decomposto em várias ordens de produção. Cada ordem de produção deve ser referente a apenas um tipo de peça.
Sincronizar chão-de-fábrica	Deve ser criado um mecanismo que permita a sincronização da entrada das ordens de produção no chão-de-fábrica.
Iniciar simulação	Os recursos e as ordens por peça devem reconhecer a liberação para início da negociação no chão-de-fábrica.
Informar capacidade das máquinas	Recuperar, a partir de uma base de dados, as informações relativas à capacidade de cada um dos recursos.
Solicitar propostas	As ordens de produção devem iniciar a negociação para a sua manufatura com os recursos.
Gerar propostas	Os recursos devem gerar propostas para a manufatura das ordens de produção.
Comparar propostas	As ordens de produção devem comparar todas as propostas recebidas por parte dos recursos e selecionar a que melhor atende às suas necessidades.
Contratar recurso	A partir da análise e comparação das propostas recebidas, as ordens de produção efetuam a contratação de um recurso para sua manufatura.
Processar ordem	O recurso deverá reconhecer sua contratação por uma ordem de produção e iniciar a simulação da mesma.
Finalização da ordem	Deve ser detectado o término da manufatura de uma ordem de produção, a fim de ser registrada sua baixa na simulação.

## 4.2 Conclusão do capítulo

Este capítulo tratou da validação do contexto da simulação e de como as exigências para o seu desenvolvimento foram especificadas e cumpridas, bem como das bases de conhecimento foram utilizadas pelos agentes para compor o modelo proposto.

Inicialmente a validação do modelo foi feita através da demonstração de sua viabilidade teórica, tendo como base a especificação do projeto referente ao modelo conceitual, para este fim foi descrito o domínio no qual o modelo proposto esta inserido. Dentro deste contexto, o modelo aqui proposto e apresentado foi classificado como um modelo de análise e verificação, sendo que a hipótese em questão corresponde ao estudo em profundidade da questão da correlação entre planos de processo alternativos e a programação da produção em

---

um ambiente de *layout* funcional e estático. Busca-se, através deste estudo, comprovar ou refinar a teoria existente a respeito do fenômeno aqui especificado.

Diante da necessidade de se possuir medidas concretas de se avaliar a performance de um modelo de simulação, optou-se por adotar os critérios baseados em desempenho (GRAVES, 1981), sendo que entre as inúmeras métricas encontradas na literatura, duas foram escolhidas para o presente trabalho: a média ou o máximo tempo de fluxo de trabalho (*flow time*) e o tempo total de conclusão (*makespan*).

Ainda como parte da validação teórica do modelo, foi feita a descrição dos elementos que compõem o modelo de simulação, recursos e peças, bem como de suas operações, planos de processo, pedidos e ordens.

Finalmente foi definido e discutido o conceito de flexibilidade em manufatura. BENGTTSSON (2001) diz que a flexibilidade de um sistema de manufatura é formada pela soma de todas as flexibilidades que o sistema contém, porém, neste trabalho, optou-se por adotar a nomenclatura proposta por BENJAAFAR e RAMAKRISHNAN (1996), que classifica as flexibilidades em relativas ao produto e ao processo. Desta forma, neste trabalho foram considerados três tipos de flexibilidades relativas ao produto: de operação, seqüência e processo e uma flexibilidade relativa ao processo de manufatura, que é a flexibilidade de máquina.

Finalmente, como parte do modelo de requisitos do sistema na metodologia PASSI, foi elaborado o registro de todos os tópicos relativos ao que o modelo de simulação proposto deve executar e sob quais condições deve fazê-lo, sendo ou não este requisito uma característica funcional (WAZLAWICK, 2004).

---

---

## CAPÍTULO 5 – ARQUITETURA E IMPLEMENTAÇÃO DO MODELO

Este capítulo trata do desenvolvimento computacional do modelo de simulação proposto através das fases da metodologia PASSI, bem como de parte de sua verificação.

Conforme citado no item 3.13, que trata dos conceitos de validação e verificação adotados, a verificação do modelo proposto concentra-se tanto nas especificações do projeto do modelo, tema já discutido no capítulo 4, quanto do seu desenvolvimento computacional. Deste modo, a verificação tem por escopo assegurar que o modelo tenha sido corretamente implementado do ponto de vista computacional, com especial atenção aos processos sociais dos agentes e sua consistência como organização multiagente.

Para este fim, são apresentadas neste capítulo as fases da metodologia PASSI que tratam da verificação do modelo. Inicialmente é dada continuidade à fase de requisitos do sistema, iniciada no capítulo 4, que trata do modelo conceitual adotado. Em seguida é apresentada a fase de sociedade de agentes, responsável por modelar a solução em função de seus papéis, interações, dependências e ontologia. O capítulo se encerra com o modelo de implementação, responsável por apresentar a arquitetura proposta em função de suas classes e métodos.

### 5.1 Modelo de requisitos do sistema - Sub-fase de identificação dos agentes

Nesta sub-fase é feita a identificação dos agentes que irão compor o modelo de simulação proposto. A figura 5.1 destaca a sub-fase de identificação dos agentes, dentro do modelo de requisitos do sistema.

Nesta etapa, os estereótipos de relações entre casos de uso mapeados no item 4.1.11 que trata dos requisitos do domínio, são convertidos em comunicação, pois os diferentes pacotes (agentes) podem se relacionar somente desta forma (COSENTINO E SABATUCCI, 2003).

Para uma melhor compreensão da forma como um agente é mapeado a partir dos casos de uso, é interessante ter em mente a definição que rege o escopo e a futura identificação dos mesmos. Como já citado no item 3.5 do presente trabalho, na metodologia PASSI o agente é considerado sob a ótica de dois diferentes aspectos, primeiramente como entidade autônoma e posteriormente como unidade de software.

---

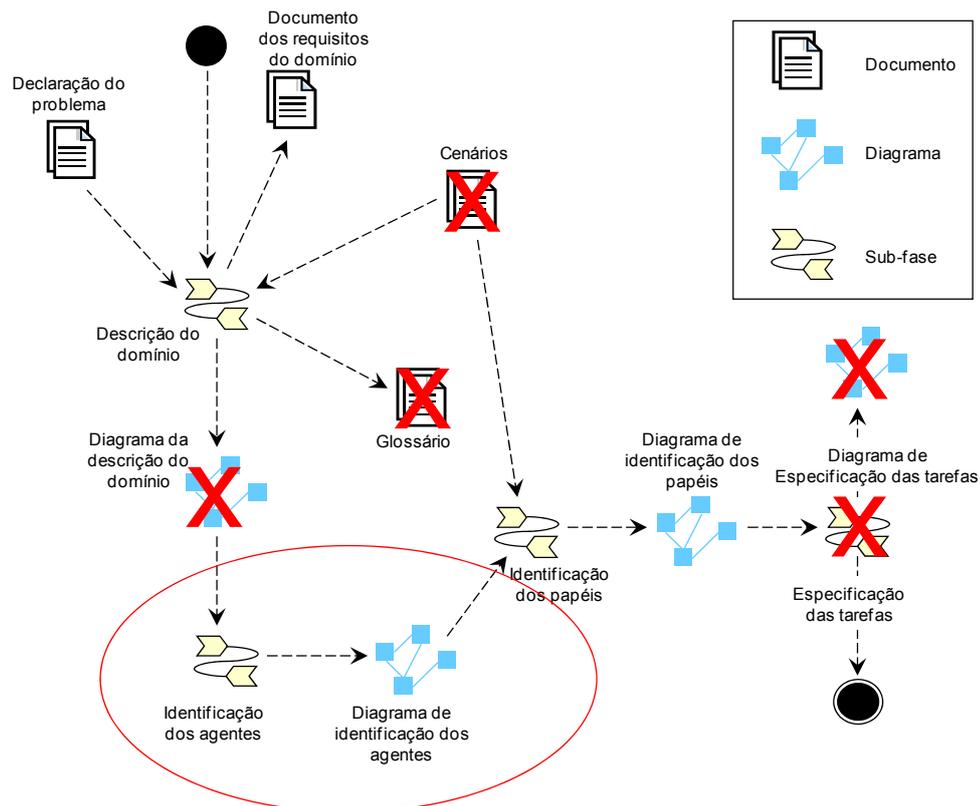


Figura 5.1 – Sub-fase de identificação dos agentes  
 FONTE: Cosentino, Sabatucci e Seidita (2003)

Agente é uma entidade autônoma composta por papéis e que possui conhecimento. Pode ser visto em diferentes níveis de abstração. No modelo de requisitos do sistema, agentes são vistos como uma agregação lógica de funcionalidades (COSSENTINO, SABATUCCI E CHELLA, 2003).

Definição 6.1  
 Agente na visão da metodologia PASSI

Um papel é visto como uma coleção de tarefas realizadas por um agente para alcançar um sub-objetivo ou oferecer algum serviço a outros membros da sociedade (COSSENTINO, SABATUCCI E CHELLA, 2003).

Definição 6.2  
 Papel de um agente

A figura 5.2 apresenta o diagrama de identificação de agentes. Neste modelo foi mapeada a necessidade de cinco diferentes tipos de agente: *Server*, *JobOrder*, *Resource*, *InputOrder* e *Sync*<sup>9</sup>.

<sup>9</sup> Optou-se por nomear os agentes identificados em inglês, a fim de manter uma padronização com os softwares e linguagens de programação utilizadas no desenvolvimento do modelo. Esta mesma padronização será utilizada para nomear os comportamentos, papéis e demais termos relacionados ao modelo aqui proposto e desenvolvido.

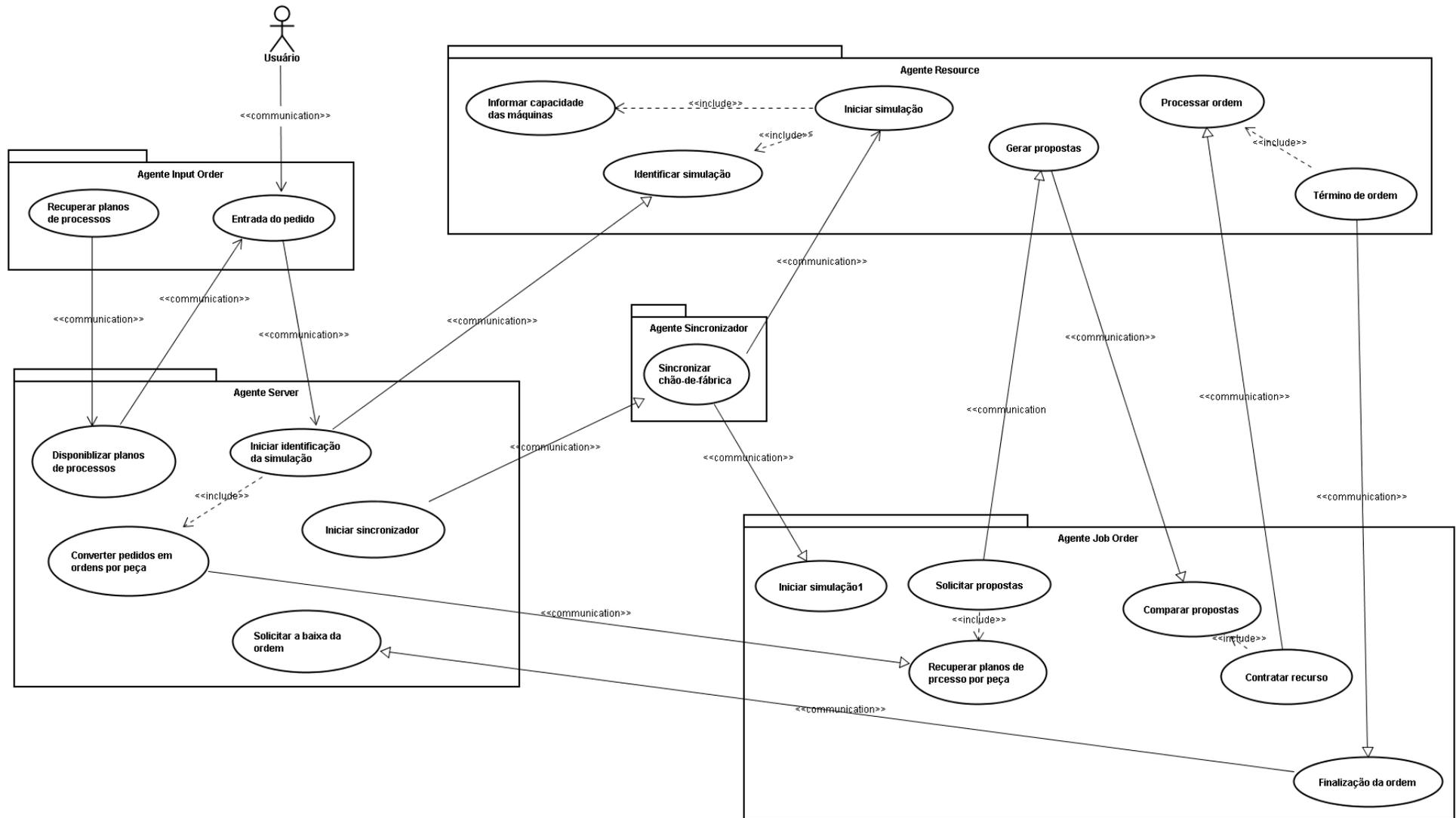


Figura 5.2 – Diagrama de identificação dos agentes

---

A figura 5.2 também representa as comunicações entre os agentes, cuja direção sempre vai do iniciador para o participante. A seguir é apresentada uma descrição das funcionalidades mapeadas para cada um dos agentes que compõem o modelo proposto.

Funções do agente *InputOrder* :

- a) Recuperar os planos de processos, referentes a cada uma das peças a serem simuladas disponibilizando-os ao usuário; e
- b) Receber os pedidos enviados pelos usuários e repassá-los ao agente *Server*.

Funções do agente *Sync*:

- a) Sincronizar a liberação das ordens, permitindo o início da negociação entre os agentes *JobOrder* e *Resource*.

Funções do agente *Server*:

- a) Disponibilizar os planos de processos ao agente *InputOrder* quando solicitado;
- b) Receber os novos pedidos enviados pelo usuário através do agente *InputOrder*;
- c) Preparar os agentes *Resources* para o início da simulação;
- d) Manter uma lista de todas as ordens em processamento na simulação para organizar e controlar o andamento da simulação;
- e) Instanciar os agentes *JobOrders*, que são a representação das ordens de fabricação por peça; e
- f) Instanciar o agente *Sync*, o qual é responsável por sincronizar os agentes *JobOrders* no chão-de-fábrica.

O agente *JobOrder* representa cada uma das ordens, já definidas anteriormente no item 4.1.4, e tem por funções:

- a) Recuperar os planos de processos referentes a peças da correspondente a ordem; e
- b) Negociar a programação das peças no chão-de-fábrica através da comparação das propostas enviadas pelos agentes *Resources*.

O agente *Resource* representa cada uma das entidades recursos, já definidas anteriormente no item 4.1.4, e tem por funções:

- a) Recuperar os dados de capacidade de cada um dos recursos a partir de um banco de dados, bem como os tempos de processamento de cada uma das operações que compõem os planos de processos das peças;
-



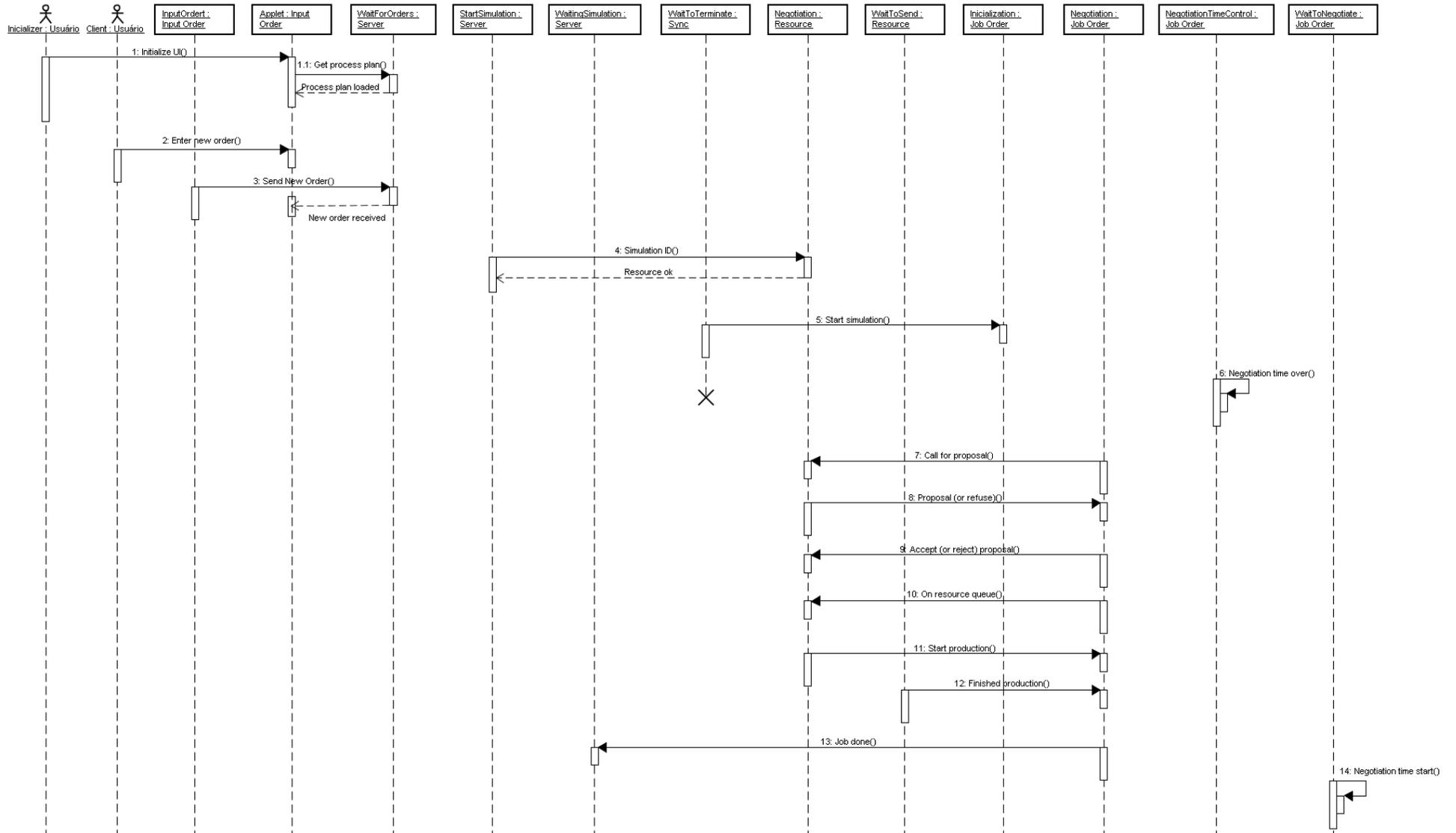


Figura 5.4 – Diagrama de identificação de papéis

---

É preciso fazer uma distinção entre os diagramas de seqüência convencionais e os diagramas utilizados para a identificação de papéis. Usualmente, diagramas de seqüência iniciam devido a uma ação provocada ou requerida por um ator, porém na metodologia PASSI, como os agentes são autônomos e ativos, eles podem servir de “gatilho” para um novo cenário, e os atores podem ou não aparecer na seqüência.

A seguir são descritos os diversos papéis que cada agente pode assumir durante o seu ciclo de vida. É importante notar que posteriormente alguns destes papéis serão convertidos em comportamentos, sendo que os principais serão detalhados no decorrer deste capítulo.

Papéis que podem ser assumidos pelo agente *InputOrder* durante o seu ciclo de vida:

- a) *Applet* - interface entre o sistema e o usuário, tanto para a inicialização do modelo quanto para a entrada de novos pedidos; e
- b) *InputOrder* – possui duas funções distintas: solicitar a recuperação resgate dos planos de processo das peças no banco de dados e disponibilizá-los ao usuário; e enviar os novos pedidos ao agente *Server*;

Papéis que podem ser assumidos pelo agente *Sync* durante o seu ciclo de vida:

- a) *WaitToTerminate* – tem por objetivo enviar uma mensagem indicando o início da simulação para todas as ordens, representadas pelos agentes *JobOrder*, que venham a fazer parte de uma dada simulação. Seu ciclo de vida inicia com a sua instanciação (criação) e se encerra imediatamente após o envio da mensagem.

Papéis que podem ser assumidos pelo agente *Server* durante o seu ciclo de vida:

- a) *WaitForOrders* – o agente permanece aguardando uma solicitação de recuperação dos planos de processo das peças, ou aguardando a entrada de um novo pedido;
- b) *StartSimulation* – o agente solicita a identificação do tempo de *setup* de todos os recursos disponíveis; e
- c) *WaitingSimulation* – o agente permanece no aguardo até que a última ordem seja concluída a fim de encerrar a simulação.

Os papéis assumidos pelo agente *Server*, que posteriormente serão convertidos em comportamentos, serão descritos com maiores detalhes no item 5.8.1 do presente capítulo.

Papéis que podem ser assumidos pelo agente *JobOrder* durante o seu ciclo de vida:

---

- 
- a) *Inicialization* – libera as ordens (agentes *JobOrders*) para que elas sejam sequenciadas no chão-de-fábrica, através da negociação com os recursos (agentes *Resources*).
  - b) *Negotiation* – tem como principal função o tratamento das mensagens recebidas por este agente no decorrer de um ciclo de negociação;
  - c) *NegotiationTimeControl* – é responsável por controlar o tempo que o agente *JobOrder* irá permanecer aceitando mensagens provenientes dos agentes *Resource*; e
  - d) *WaitToNegotiate* – é responsável por controlar o tempo que o agente *JobOrder* irá permanecer no estado *WaitForNegotiation*.

Os papéis *Negotiation*, *NegotiationTimeControl* e *WaitToNegotiate* assumidos pelo agente *JobOrder*, que posteriormente serão convertidos em comportamentos, serão descritos com maiores detalhes nos itens 5.8.2 a 5.8.4 do presente capítulo.

Papéis que podem ser assumidos pelo agente *Resource* durante o seu ciclo de vida:

- a) *Negotiation* - tem como principal função o tratamento das mensagens recebidas por este agente no decorrer de um ciclo de negociação;
- b) *WaitToSend* – este agente permanece bloqueado até que uma ordem seja finalizada, enviando ao próprio agente *Resource* uma mensagem avisando o término da mesma;

O papel *Negotiation* assumido pelo agente *Resource* será descrito com maiores detalhes no item 5.8.6 do presente capítulo.

O próximo modelo da metodologia PASSI a ser discutido e implementado é o modelo de sociedade de agentes, descrito no item a seguir.

### **5.3 Modelo de sociedade de agentes - Sub-fase de Descrição da ontologia de comunicações**

A sub-fase de descrição da ontologia de comunicação tem por objetivo representar as interações sociais entre agentes, sendo apresentada sob a forma de diagrama de classes UML.

Segundo COSENTINO (2005), ao se projetar este diagrama deve-se usar como base a fase de identificação dos agentes, descrita no item 5.1 do presente trabalho, porém introduzindo-se uma classe para cada agente identificado e estabelecendo-se uma associação para cada comunicação entre dois agentes. Também deve ser criada uma estrutura de dados

---

em cada agente, a fim de armazenar os dados trocados. A figura 5.5 destaca a sub-fase de descrição da ontologia de comunicações, dentro do modelo de sociedade de agentes.

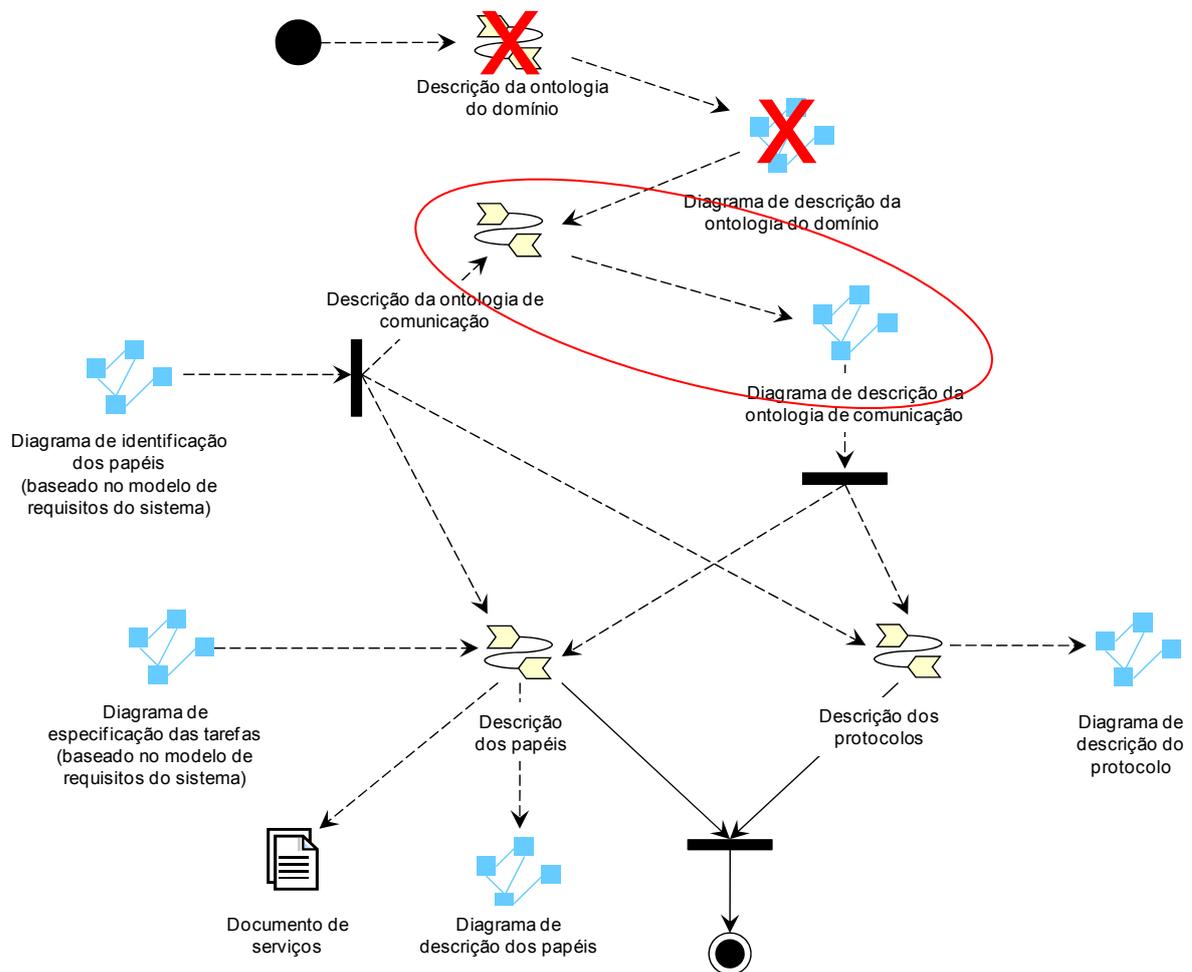


Figura 5.5 – Sub-fase de descrição da ontologia de comunicações  
 FONTE: Cossentino, Sabatucci e Seidita (2003)

Comunicação é definida como a interação entre dois agentes. Cada comunicação é descrita em termos de ontologia, conteúdo de linguagem e protocolo de interação (COSENTINO, SABATUCCI E SEIDITA, 2003).

**Definição 6.3**  
**Comunicação**

A figura 5.6 apresenta o diagrama de descrição da ontologia de comunicação do modelo de simulação proposto. Cada agente, representado na cor amarela no diagrama, é descrito em função de seu conhecimento. A partir deste conhecimento é estabelecido um relacionamento para cada dois agentes que se comuniquem entre si, sendo que em cada um destes relacionamentos são representados os papéis exercidos pelos agentes durante a comunicação.

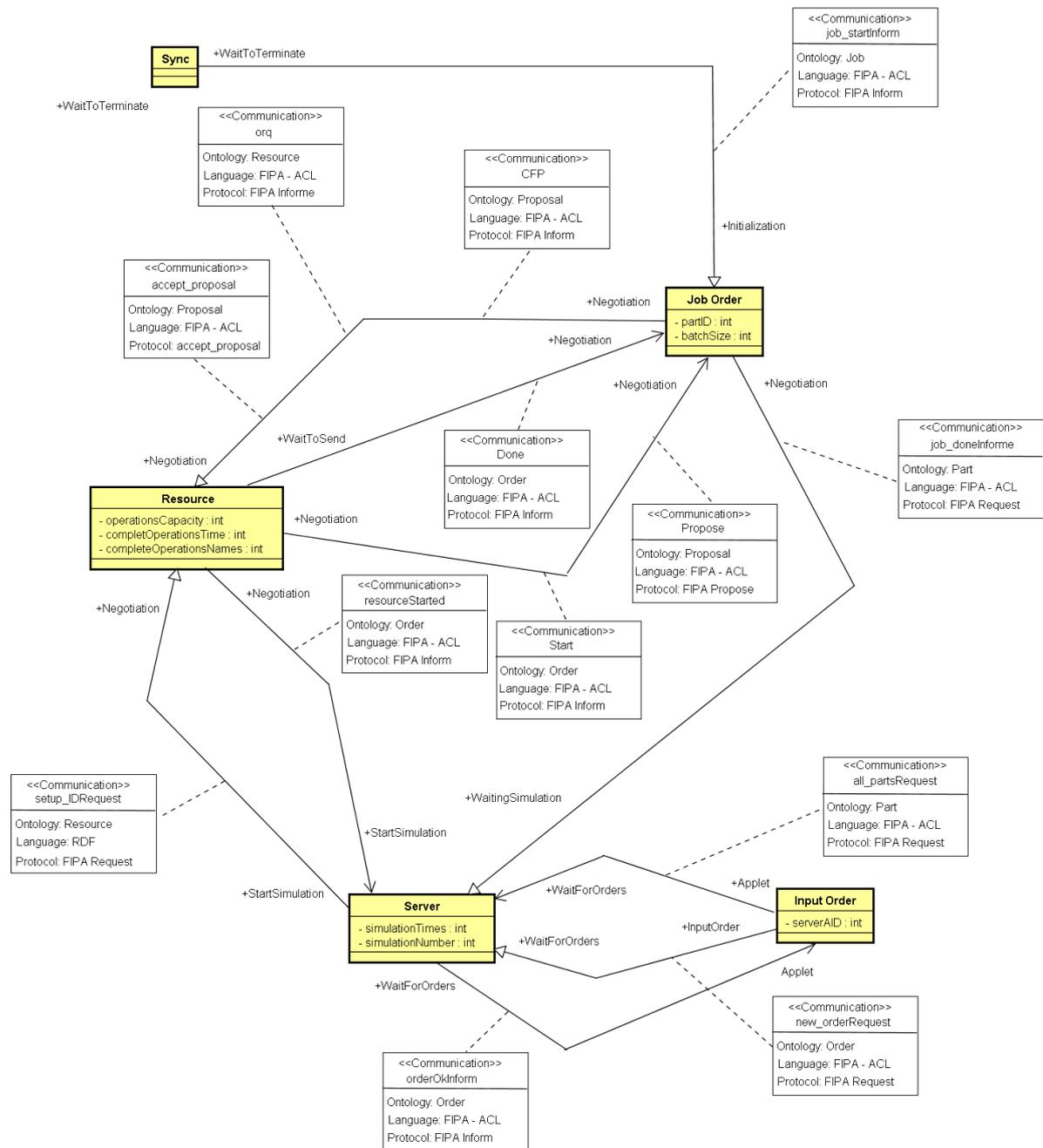


Figura 5.6 – Diagrama de descrição da ontologia de comunicação

Como consequência disto, cada comunicação, representada na cor branca no diagrama, é caracterizada por três diferentes atributos: sua ontologia; sua linguagem, que é o conteúdo da mensagem; e seu protocolo, neste caso, referindo-se a algum dos protocolos de interação FIPA.

Um exemplo da representação descrita na figura 5.6 pode ser visto entre os agentes (representados na cor amarela) *JobOrder* e *Server*. O agente *JobOrder* comunica ao agente *Server* o término do processamento de uma ordem através de uma mensagem *Order\_Done*.

A ontologia desta mensagem se refere ao elemento de simulação **Peça**, especificado no item 4.1.4, que trata da definição de peça e recurso produtivo no modelo de simulação apresentado. Ou seja, uma peça, composta por operações produtivas e representada por um número de ordem, dentro de uma simulação identificada, está sendo finalizada por um recurso produtivo também identificado.

Já a representação desta mensagem é baseada na FIPA – ACL (*Agent Communication Language*), sendo representada pelo protocolo de interação *Inform*, padronizado pelo documento FIPA - *Communicative Act Library Specification* (SC00037J, 2002).

O protocolo de interação *Inform*, por sua vez, indica que a mensagem *Order\_done* enviada contém três propriedades:

- a) A proposição enviada é verdadeira;
- b) Tem a intenção que o agente receptor acredite que a proposição enviada é verdadeira; e
- c) Não acredita que o agente receptor tenha algum conhecimento da verdade da proposição.

As primeiras duas propriedades definidas acima são diretas: o agente que envia é sincero, e de alguma maneira gerou a intenção que o receptor deveria saber a proposição (talvez tenha lhe sido perguntado). A última propriedade está relacionada à semântica do ato.

O agente que recebe a mensagem, neste caso o agente *Server*, deve por sua vez acreditar que:

- a) O agente que a enviou acredita que a proposição é verdadeira; e
- b) O agente que a enviou deseja que o receptor acredite na verdade da proposição.

Tabela 5.1 – Modelo formal da mensagem Inform padronizada pela linguagem FIPA – ACL

FONTE: FIPA - Communicative Act Library Specification SC00037J, 2002

Modelo Formal	$\langle i, \text{inform } (j, \phi) \rangle$ FP: $B_i\phi \wedge \neg B_i(B_j\phi \vee U_j\phi)$ RE: $B_i\phi$
Exemplo	Agent <i>i</i> informs agent <i>j</i> that (it is true that) it is raining today.  (inform :sender (agent-identifier :name i) :receiver (set (agent-identifier :name j)) :content "weather (today, raining)" :language Prolog)

O modelo formal de uma mensagem *Inform* é descrito na tabela 5.1. A padronização FIPA em sua especificação abstrata de arquitetura, do inglês FIPA - *Abstract Architecture Specification* (SC00001L, 2002), define “comunicação” como atos de fala, do inglês *speech acts*, conceito introduzido por SEARLE (1969), e os agrupa em vários protocolos de interação que definem a seqüência das mensagens esperadas. As especificações detalhadas dos parâmetros de mensagens FIPA podem ser encontradas nos documentos FIPA - *ACL Message Structure Specification* (SC00061G, 2002) e FIPA - *Communicative Act Library Specification* (SC00037J, 2002), não sendo parte do escopo deste trabalho discuti-las.

### 5.4 Modelo de sociedade de agentes - Sub-fase de descrição dos papéis

Na sub-fase de descrição dos papéis, um diagrama de classe UML é utilizado para representar as interações sociais entre os agentes. A figura 5.7 apresenta o meta-modelo da sub-fase de descrição dos papéis em função dos relacionamentos entre a documentação de trabalho e os diagramas gerados.

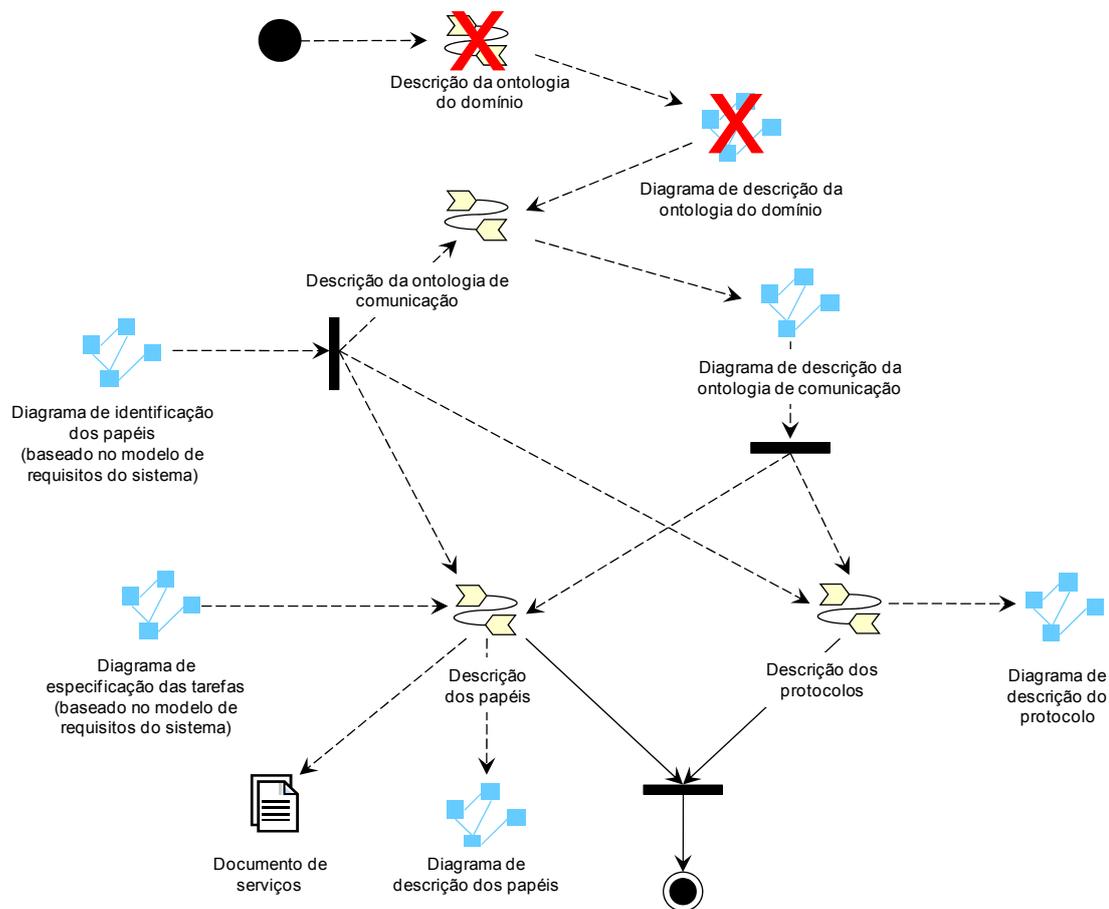


Figura 5.7 – Sub-fase de descrição de papéis  
 FONTE: Cossentino, Sabatucci e Seidita (2003)

Cada agente é representado como um pacote que contém diferentes classes e cada classe representa os diferentes papéis que o agente poderá assumir durante seu ciclo de vida. É importante notar que na metodologia PASSI a sintaxe e a notação para diagramas de classe UML foram adaptadas, a fim de representar conceitos relacionados a agentes (COSSENTINO, 2005).

Os papéis são obtidos através da composição de várias tarefas em um comportamento resultante. A fim de mostrar quais tarefas são necessárias para compor o comportamento desejado, as tarefas são alocadas no compartimento de operações da classe. Cada tarefa é então relacionada a uma ação ou a um grupo de ações, sendo que a lista de tarefas descreve o que um papel pode executar. Também são introduzidas no gráfico de papéis algumas relações adicionais denominadas de dependências entre agentes.

Dependência é a relação ente dois papéis de diferentes agentes, ou seja, implica que um agente depende de outro a fim de alcançar algum de seus objetivos (COSSENTINO, SABATUCCI E SEIDITA, 2003).

Definição 6.5  
Dependência

Estas dependências são representadas por dois tipos distintos: (a) de serviço (*service*), quando um papel depende de outro a fim de alcançar um de seus objetivos; e (b) de recurso (*resource*), quando um papel depende de outro para a disponibilização de uma entidade. Ambas as dependências são representadas no diagrama por linhas tracejadas e com seu tipo expresso entre parênteses (CONSENTINO, 2005).

A figura 5.8 apresenta o diagrama de descrição de papéis para o modelo proposto, onde as linhas tracejadas que conectam diferentes papéis de um mesmo agente são identificadas como linhas de mudança de papel (*ROLE CHANGE*) e descrevem uma relação de dependência do segundo papel em relação ao primeiro. As conversações entre dois papéis são representadas por linhas sólidas usando os mesmos nomes das relações estabelecidas no diagrama de ontologia de comunicação, descrito no item 5.3 do presente capítulo.

Por exemplo, na figura 5.8 o agente *Server* é representado na forma de um pacote com o mesmo nome. Este pacote é formado por três classes distintas, *WaitForOrders*, *StartSimulation* e *WaitingSimulation*, que correspondem aos papéis que este agente assume (já definidos no item 5.2 do presente trabalho). Estes papéis estão relacionados entre si através de um *CHANGE ROLE*, que indica a mudança de um para o outro durante o ciclo de vida do agente.

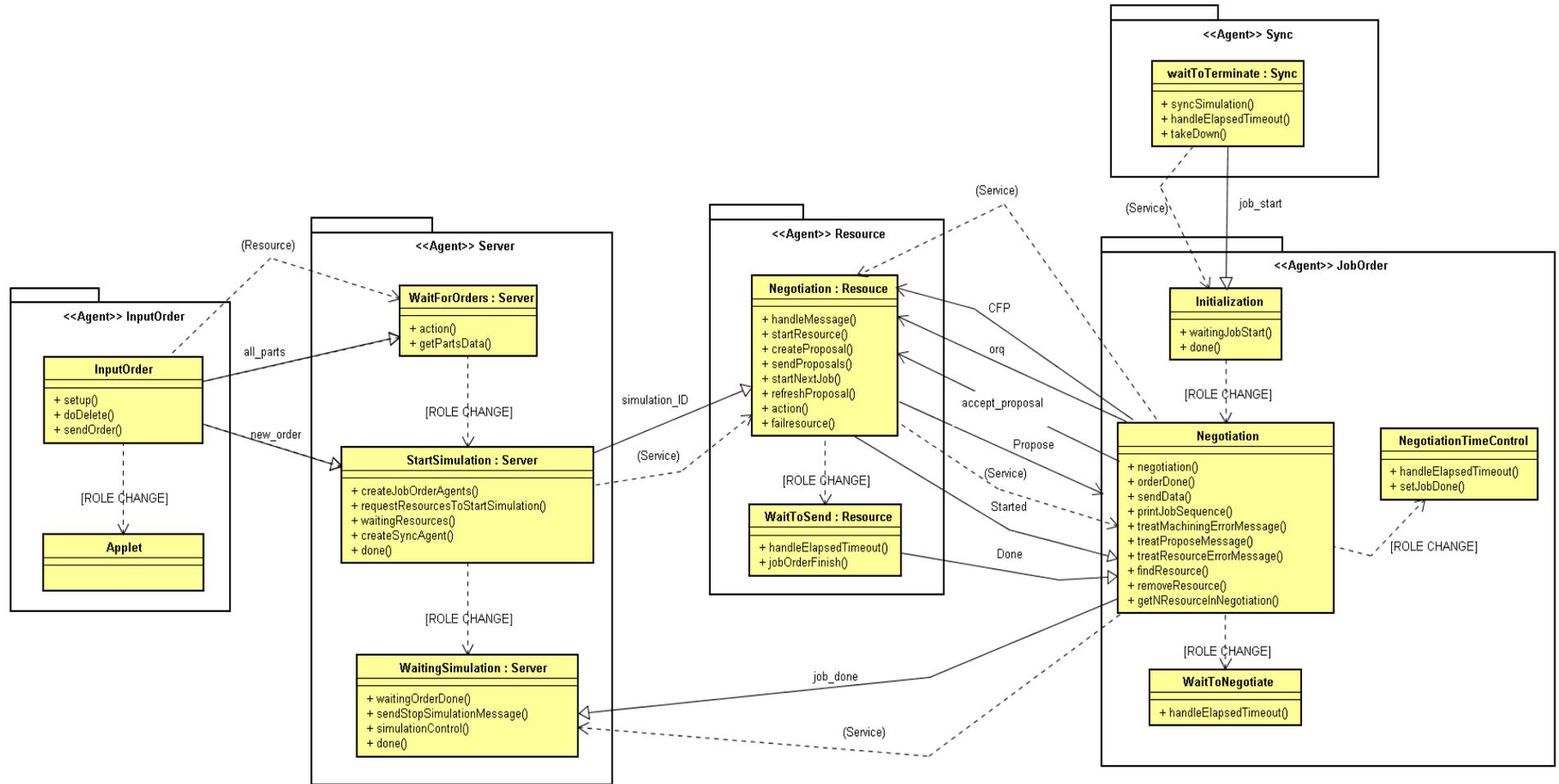


Figura 5.8 – Diagrama de descrição dos papéis

---

Para facilitar a compreensão do funcionamento de um papel, será utilizada como exemplo a classe que representa o papel *StartSimulation*, pertencente ao agente *Server*. Este papel é instanciado a partir de uma mensagem *New\_Order* enviada pelo agente *InputOrder*, indicando a entrada de um novo pedido, sendo formada pelas tarefas:

- a) *createJobOrdersAgents*, responsável por instanciar os agentes *JobOrder* a partir do pedido realizado pelo usuário;
- b) *requestResourcesToStartSimulation*, responsável por solicitar dos agentes *Resources* os tempos de *setup* previstos;
- c) *waitingResources*, responsável por aguardar a confirmação dos recursos que estarão disponíveis na simulação;
- d) *createSyncAgent*, responsável por instanciar o agente *Sync*; e
- e) *done*, responsável por controlar a lista de ordens já executadas.

Ainda segundo o diagrama da figura 5.8, o papel *StartSimulation* se comunica com o papel *Negotiation* do agente *Resource* através da mensagem de *Simulation\_ID*, solicitando os tempos de *setup* previstos para as ordens a serem simuladas. Também é possível observar que o papel *StartSimulation* possui uma dependência do tipo *service* em relação ao agente *Resource*. Neste caso específico, esta dependência implica que uma rodada de simulação somente será iniciada caso os agentes *Resources*, escolhidos para participar da mesma, já estejam instanciados. A mesma lógica de representação se aplica aos demais agentes do modelo que fazem parte do diagrama de descrição dos papéis na figura 5.8.

## 5.5 Modelo de sociedade de agentes - Sub-fase de descrição dos protocolos

Conforme visto no item 5.3, que trata da descrição da ontologia de comunicações, cada interação entre dois agentes é descrita em termos de ontologia, conteúdo de linguagem e protocolo de interação (COSENTINO, SABATUCCI E SEIDITA, 2003). Porém, nem sempre estes protocolos são suficientes para a representação da arquitetura do modelo adotado. Quando isto ocorre, protocolos especiais devem ser desenvolvidos e representados.

A metodologia PASSI sugere o uso de diagramas de seqüência em AUML - *Agent-based Unified Modeling Language* para a representação de protocolos especiais (FIPA MODELING: INTERACTION DIAGRAMS - WORKING DRAFT, 2003). Dentro desta premissa, dois diferentes protocolos foram desenvolvidos especialmente para o modelo de simulação proposto, sendo descritos a seguir.

---

### 5.5.1 Protocolo de negociação

O primeiro protocolo trata da negociação das ordens entre os agentes *JobOrder* e *Resource*. Ele tem como objetivo permitir a programação das ordens de produção entre os recursos alocados no chão-de-fábrica simulado. Para que isto ocorra, cada ordem representada por um agente *JobOrder* deve negociar com todos os recursos, aqui representados pelos agentes *Resources*, disponíveis em uma dada simulação.

Os principais conceitos que fazem parte de uma negociação são:

- a) Ordem e recurso, já descritos no item 5.1.4, que trata da definição dos elementos de simulação no modelo proposto, e
- b) Agentes *JobOrder* e *Resource*, já identificados no item 5.1 e 5.2 do presente capítulo.

Além destes, outro conceito relevante para a negociação é a proposta, cuja formulação será descrita no item 6.1. Nesta fase do trabalho, a fim de facilitar a compreensão do protocolo a ser descrito, uma proposta será definida de forma simplificada como a soma do tempo total de processamento de uma ordem por um recurso.

Uma negociação é formada pelo agente iniciador, neste caso o agente *JobOrder*, e pelos agentes participantes, neste caso os agentes *Resources*. A mesma tem início com o envio de uma mensagem CFP (*Call For Proposals*) por parte do agente iniciador, solicitando propostas para o processamento de suas operações aos agentes participantes, conforme apresentado na figura 5.9.

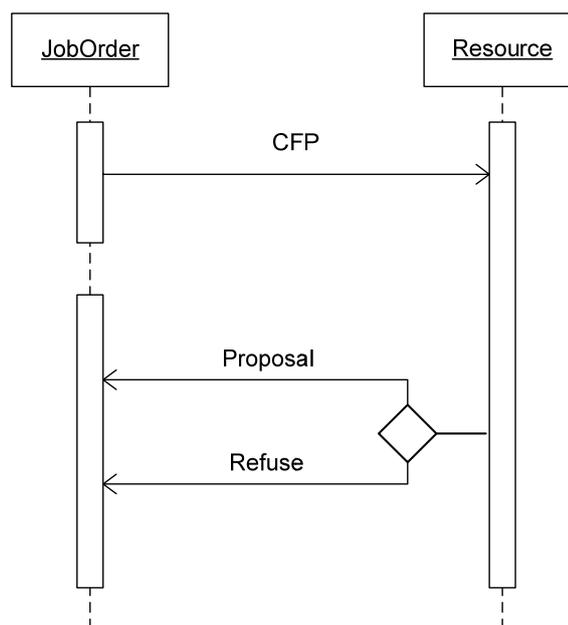


Figura 5.9 – Envio do pedido de propostas (CFP) e suas possíveis respostas

Neste caso pode haver o retorno de duas mensagens, distintas e excludentes, por parte dos agentes participantes:

- a) *Proposal* – o agente *Resource* envia uma proposta com seu tempo estimado de processamento; ou
- b) *Refuse* – o agente *Resource* recusa a ordem. Esta recusa é decorrente da falta de capacidade tecnológica por parte do recurso para a execução da ordem.

Após receber uma proposta (*Proposal*), o agente iniciador (*JobOrder*) pode retornar duas mensagens, também distintas e excludentes, ao agente participante (*Resource*):

- a) *Accept* – o agente *JobOrder* envia um aceite ao agente *Resource*, concordando com a proposta recebida e fechando o contrato para o seu processamento naquele recurso; ou
- b) *Reject* – o agente *JobOrder* rejeita a proposta recebida. Esta rejeição é decorrente do recebimento anterior, por parte do agente *JobOrder*, de uma proposta melhor, ou seja, com menor tempo de processamento, para a execução da ordem. Isto pode ocorrer quando o agente *JobOrder* negocia em paralelo com vários recursos.

A figura 5.10 apresenta a mensagem de resposta do agente *JobOrder*, contratando ou rejeitando a proposta feita pelo agente *Resource*.

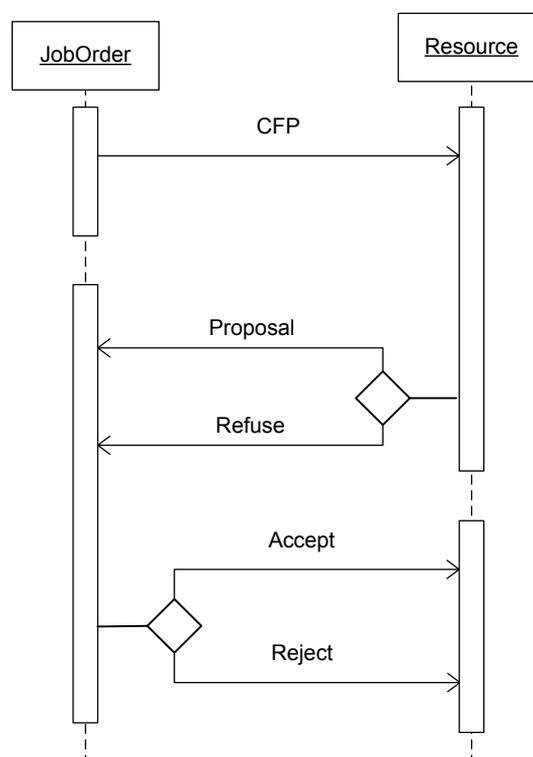


Figura 5.10 – Fechamento de contrato ou rejeição da proposta

Considerando-se que até este momento a ordem está em processamento no seu recurso atual, espera-se que a mesma termine sua negociação com o próximo recurso antes do término de seu processamento no recurso em que se encontra, evitando-se, desta forma, a criação de filas. A forma como o tempo de negociação é controlado e seu mecanismo completo serão discutidos no item 6.1.1 do presente trabalho.

Somente após o término do processamento da ordem no recurso atual o agente iniciador enviará uma mensagem ORQ (*On Resource Queue*) para o agente participante (*Resource*), indicando o seu encaminhamento para a fila de processamento do recurso contratado conforme figura 5.11.

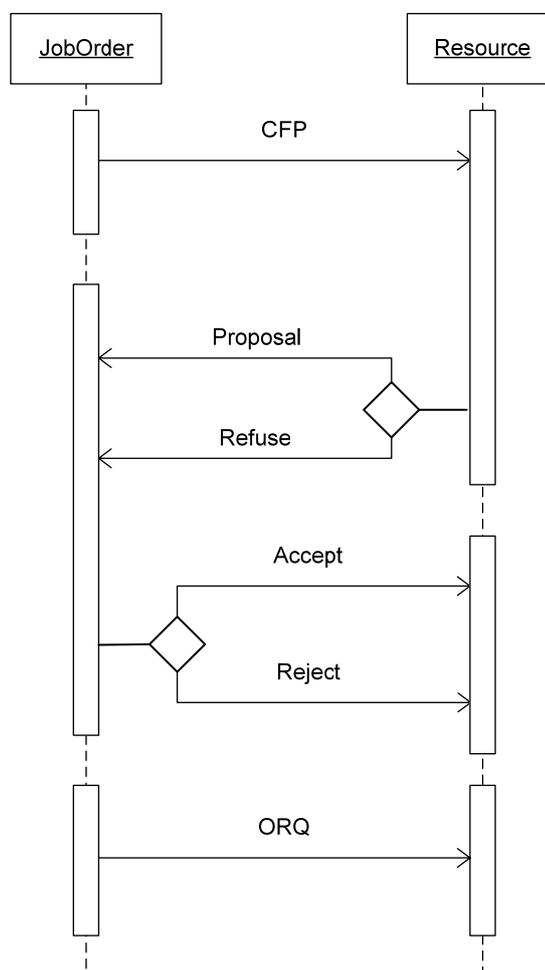


Figura 5.11 – A ordem se encaminha para a fila do recurso

É importante ressaltar que todas as mensagens performativas utilizadas neste protocolo, com exceção da mensagem ORQ (*On Resource Queue*), já possuem a sua semântica definida pelo documento FIPA - *Communicative Act Library Specification* (SC00037J, 2002). No caso da mensagem ORQ, sua padronização foi baseada nos documentos FIPA – ACL (*Agent*

*Communication Language*) e FIPA - *Communicative Act Library Specification* (SC00037J, 2002).

### 5.5.2 Protocolo de renegociação

O segundo protocolo trata do mecanismo de renegociação das ordens entre os agentes *JobOrder* e *Resource*. Para compreender a lógica de renegociação é importante primeiramente notar que o agente *JobOrder* possui dois objetivos distintos e, quando possível, complementares: (a) garantir um recurso para o seu processamento; e (b) escolher o menor tempo de processamento possível entre todas as propostas recebidas. Ou seja, o agente pode vir a escolher uma proposta que momentaneamente lhe seja conveniente, mas que devido à natureza dinâmica da negociação, pode deixar de ser interessante a partir do recebimento de uma nova proposta.

Quando isto ocorre, o agente iniciador *JobOrder* deve enviar ao agente *Resource*, contratado anteriormente, uma mensagem *Disconfirm*, cancelando a contratação do mesmo, conforme ilustrado na figura 5.12.

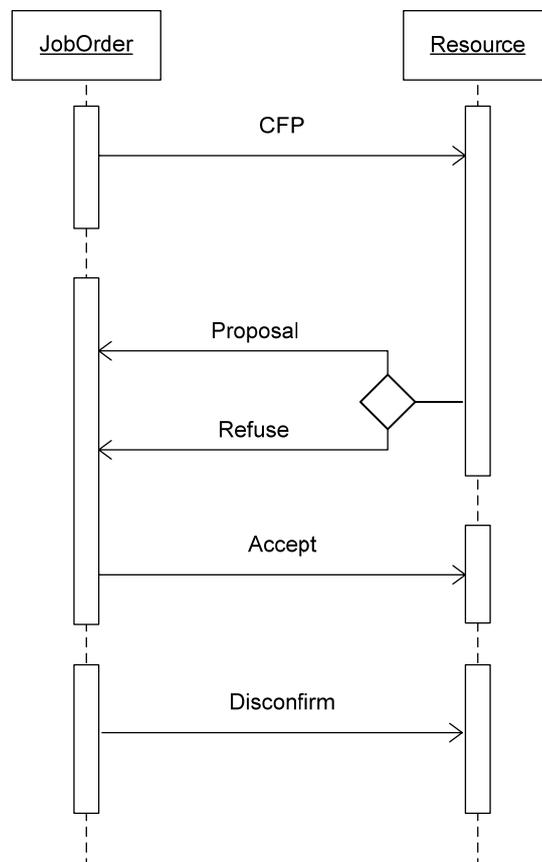


Figura 5.12 – Cancelamento de uma ordem

---

Após o recebimento da mensagem *Disconfirm*, o agente *Resource* continuará a buscar cumprir o seu objetivo, que é o de competir pelas ordens no chão-de-fábrica, enviando novas propostas aos agentes *JobOrder* com quem já tenha negociado anteriormente. Para que isto ocorra, sempre que em uma negociação o agente *JobOrder* enviar uma mensagem *Reject* a um agente *Resource*, obrigatoriamente deve enviar também o valor da proposta (melhor) que gerou a recusa do recurso em questão. Esta proposta é, então, armazenada em uma variável de tempo interna do agente *Resource* chamada *BestProposal*.

Um cancelamento de contratação, aqui representado por uma mensagem *Disconfirm*, faz com que ocorra uma alteração na fila de trabalhos que já contrataram o recurso, podendo alterar para menos o tempo de processamento de novas peças. Quando isto ocorre, o agente *Resource* que é responsável pela sua fila de processamento, recalcula o novo tempo de proposta possível e o compara com os valores armazenados na variável *BestProposal*. Caso o novo tempo calculado seja menor que algum dos valores armazenados na variável *BestProposal*, é enviada uma nova proposta ao *JobOrder* correspondente.

A mensagem *Disconfirm* já possui sua semântica definida pelo documento da FIPA - *Communicative Act Library Specification* (SC00037J, 2002).

Após gerar os documentos e diagramas relativos ao modelo de sociedade de agentes, estes serão utilizados como base para o desenvolvimento do modelo de implementação de agentes, conforme previsto na metodologia PASSI. O modelo de implementação de agentes já foi descrito no item 3.8 do presente trabalho.

## **5.6 Modelo de implementação de agentes - Sub-fase de definição da estrutura multiagente**

A primeira sub-fase do modelo de implementação de agentes refere-se à definição da estrutura multiagente. A figura 5.13 apresenta seu meta-modelo em função dos relacionamentos entre a documentação de trabalho e os diagramas gerados.

Para a definição da estrutura multiagente são utilizados os dados obtidos nas sub-fases de identificação dos agentes, identificação dos papéis e ontologia de comunicação, discutidos nos itens 5.1, 5.2 e 56.3 respectivamente. A partir destas entradas, um diagrama de classes UML representa cada um dos agentes anteriormente identificados como classes, onde os atributos são relativos ao conhecimento inerente ao agente e as operações representam as suas tarefas.

---

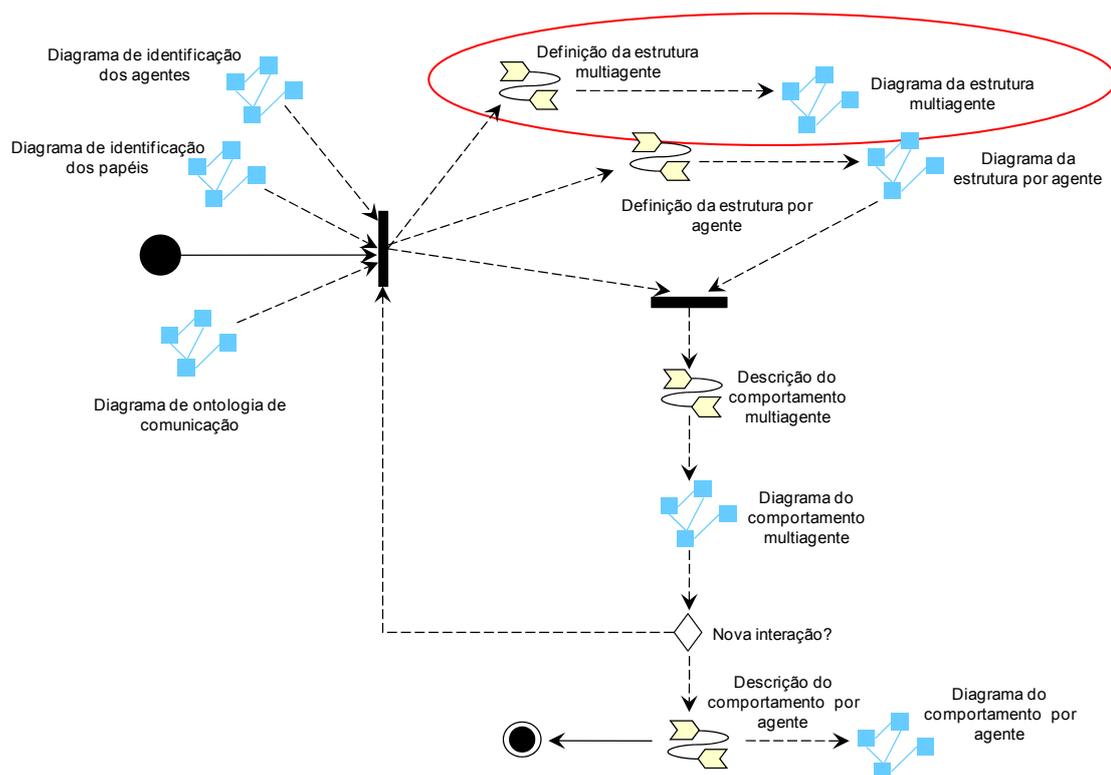


Figura 5.13 – Sub-fase de definição da estrutura multiagente  
 FONTE: Cossentino, Sabatucci e Seidita (2003)

Tomando como exemplo a classe que representa o agente *Resource* na figura 5.14, o conhecimento interno do agente é representado pelos atributos que compõem a base de conhecimento dos recursos. Esta base de conhecimento permite ao recurso interagir com o modelo de simulação. Estes atributos são:

- a) *id* – identificação do recurso (*integer*);
- b) *name* – nome do recurso (*integer*);
- c) *type* – tipo do recurso (*integer*);
- d) *setupindex* – tempo total de *setup* por operação no recurso (*integer*), utilizado somente para as operações constantes no vetor *operationsCapacity*;
- e) *operationsCapacity* – representa as operações que o recurso é capaz de realizar (*vector*);
- f) *completeOperationsTimes* – é o tempo de processamento por operação no recurso (*vector*), utilizado somente para as operações constantes no vetor *operationsCapacity*; e
- i. *completeOperationsNames* – são os nomes das operações constantes no vetor *operationsCapacity* (*vector*<sup>10</sup>).

<sup>10</sup> A classe *Vector* de Java fornece os recursos de estruturas de dados semelhantes a arrays, ou seja, que podem redimensionar dinamicamente a si mesmas DEITEL e DEITEL (2005).

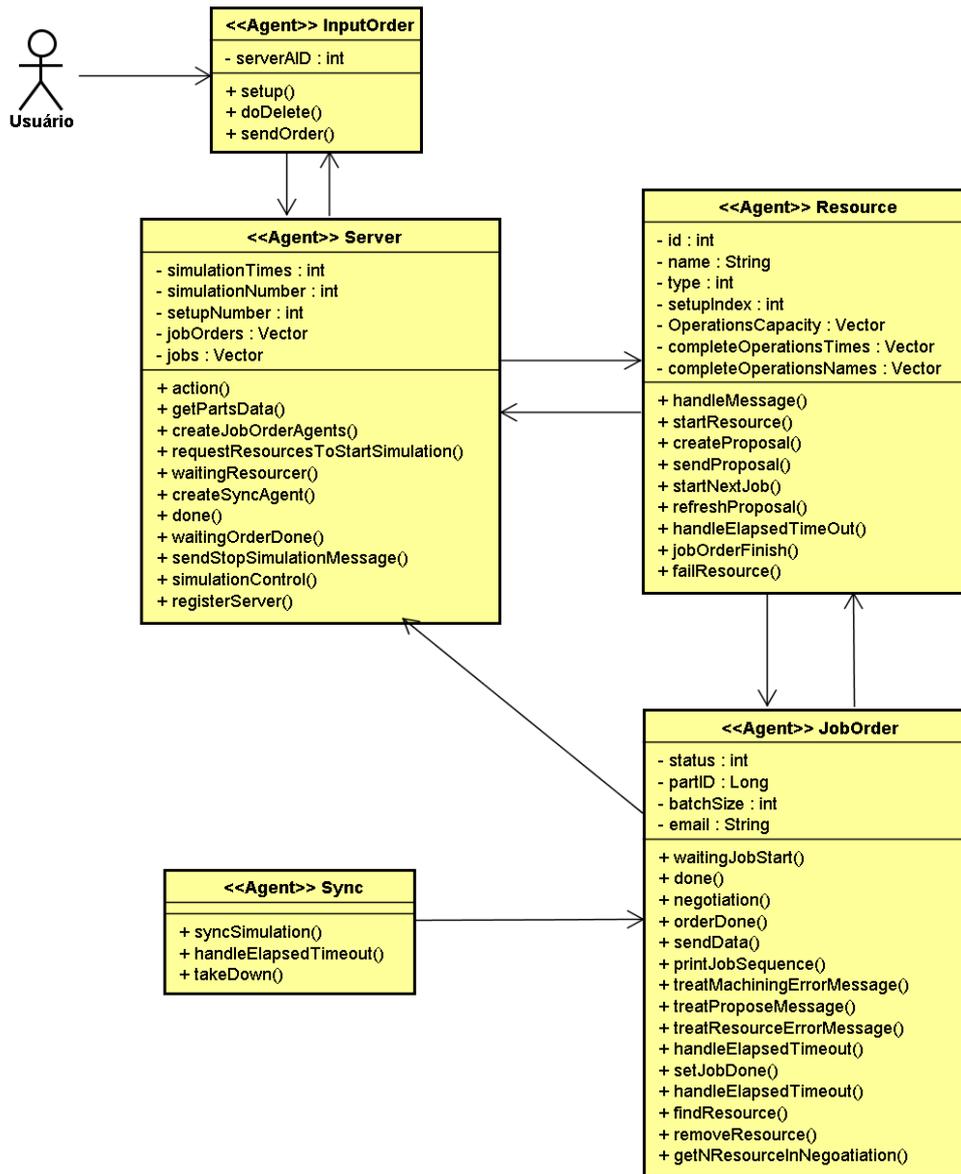


Figura 5.14 – Diagrama de definição da estrutura multiagente

Já as operações que compõem uma classe referem-se às tarefas que o agente *Resource* deve realizar para o cumprimento de seus papéis *Negotiation* e *WaitToSend*, já descritos no item 5.2 do presente trabalho. Os métodos são os seguintes:

- a) *handleMessage* – o agente *Resource* permanece no aguardo para o tratamento das novas mensagens enviadas pelos agentes *JobOrder*, e este método será detalhado no item 5.8.6 do presente capítulo;
- b) *startResource* – esvazia a fila do recurso, e este método será detalhado no item 5.8.5 do presente capítulo;
- c) *createProposal* – cria uma proposta para o processamento de uma ordem;

- 
- d) *sendProposal* – envia uma proposta de processamento de ordem para o agente *JobOrder* responsável (isto é, o agente que enviou o CFP – Call For Proposal);
  - e) *startNexJob* – inicia o processamento do próximo trabalho da fila do recurso;
  - f) *refreshProposal* – atualiza a lista de propostas recebidas pelo recurso;
  - g) *handleElapsedTimeOut* – inicializa um contador interno, e o mesmo será discutido no item 5.8.3 do presente capítulo;
  - h) *jobOrderFinish* – envia uma mensagem ao agente *JobOrder* avisando do término de uma ordem; e
  - i) *failResource* – envia uma mensagem ao agente *Server* avisando sobre uma falha de um recurso.

Ainda no diagrama da figura 5.14 pode-se observar diferentes níveis de representação de conhecimento nos agentes, o que caracteriza a heterogeneidade do modelo desenvolvido. Um exemplo disto é o agente *Sync*, que ao contrário do agente *Resource*, não possui nenhum conhecimento interno, sendo composto somente por um grupo de tarefas a serem executadas a fim de cumprir seu único comportamento *WaitToTerminate*, descrito no item 5.2 do presente capítulo.

Além do conhecimento e das tarefas de cada um dos agentes, a fim de melhor descrever as interações com o ambiente, os atores também são representados no diagrama de definição da estrutura multiagente. Um exemplo disto é apresentado na figura 5.14, onde tem-se o usuário como um ator inicializador. Porém, conforme já citado, os agentes são autônomos e ativos, podendo servir de “gatilho” para um novo cenário ou ação. Um exemplo disto é o agente *Sync*, que é instanciado diretamente pelo agente *Server*, conforme já citado no item 5.1 do presente capítulo.

## **5.7 Modelo de implementação de agentes - Sub-fase de definição da estrutura por agente**

Nesta sub-fase é representada a estrutura interna de cada um dos agentes individualmente. Para isto, são especificadas as sub-tarefas necessárias ao cumprimento de cada uma das tarefas mapeadas anteriormente na estrutura multiagente. A figura 5.15 apresenta o meta-modelo da sub-fase em função dos relacionamentos entre a documentação de trabalho e os diagramas gerados.

---

De forma análoga à definição da estrutura multiagente, na definição da estrutura por agente são utilizados os dados obtidos nas sub-fases de identificação dos agentes, identificação dos papéis e ontologia de comunicação, discutidos nos itens 5.1, 5.2 e 5.3 respectivamente.

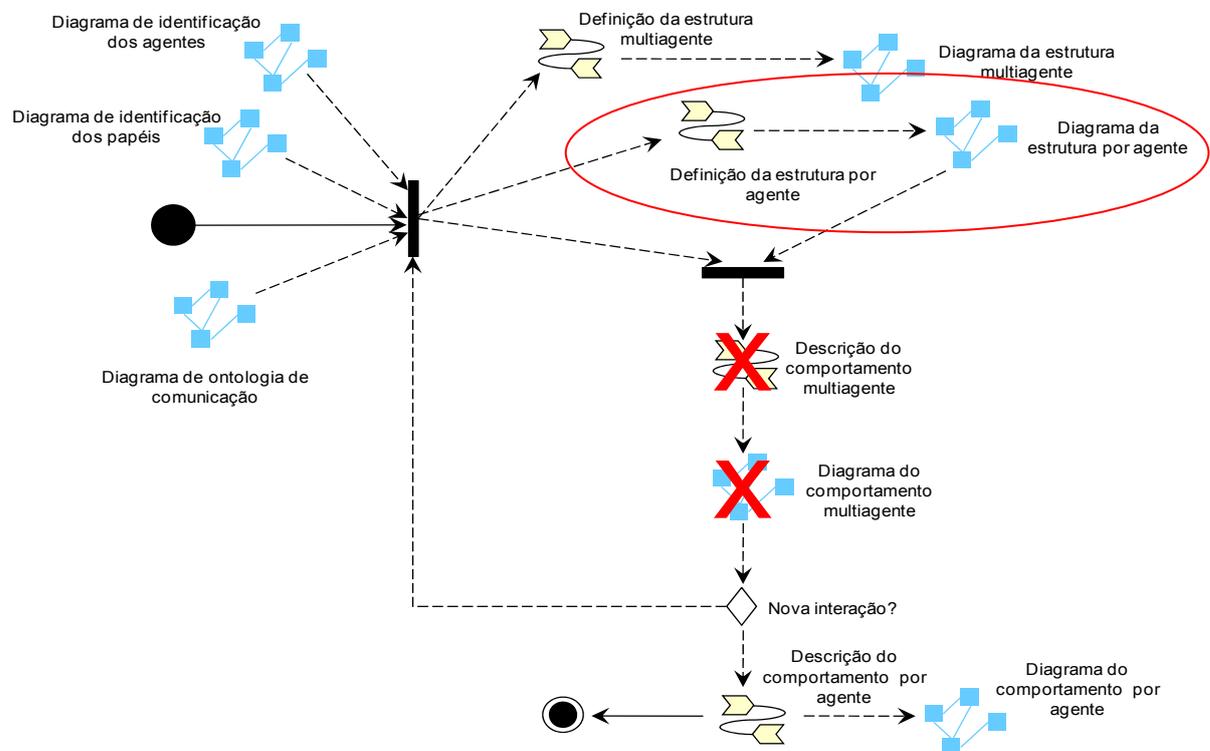


Figura 5.15 – Sub-fase de definição da estrutura por agente individual  
 FONTE: Cossentino, Sabatucci e Seidita (2003)

A figura 5.16 representa a estrutura interna do agente *Resource*, onde as tarefas *handleMessage*, *startResource*, *createProposal*, *sendProposal*, *startNexJob*, *refreshProposal*, *handleElapsedTimeOut*, *jobOrderFinish* e *failResource* são decompostas em função de suas sub-tarefas.

Como exemplo do funcionamento de uma classe do diagrama de definição da estrutura do agente *Resource*, representado na figura 5.17, será detalhada a tarefa *jobOrderFinish*, cujo objetivo final é o de enviar uma mensagem ao agente *JobOrder* avisando do término de uma ordem.

Esta tarefa é composta por três sub-tarefas distintas denominadas *getDone*, *getJobSequence* e *send*. A tarefa *jobOrderFinish* inicia com o recebimento de uma mensagem *Job\_Done*, que é uma mensagem interna, ou seja, enviada pelo próprio agente *Resource* para ele mesmo, indicando o término do processamento de uma ordem. Este comportamento será melhor descrito nos itens 5.8.5 e 5.8.6 do presente capítulo.

A partir do recebimento desta mensagem, o agente *Resource* captura o vetor *jobSequence*, composto pela operação ou operações relativas à ordem finalizada. Isto é feito através da sub-tarefa *getJobSequence*. Finalmente, o agente *Resource* envia ao agente *JobOrder* através da sub-tarefa *send*, uma mensagem *Done*, indicando o fim do processamento da ordem naquele recurso.

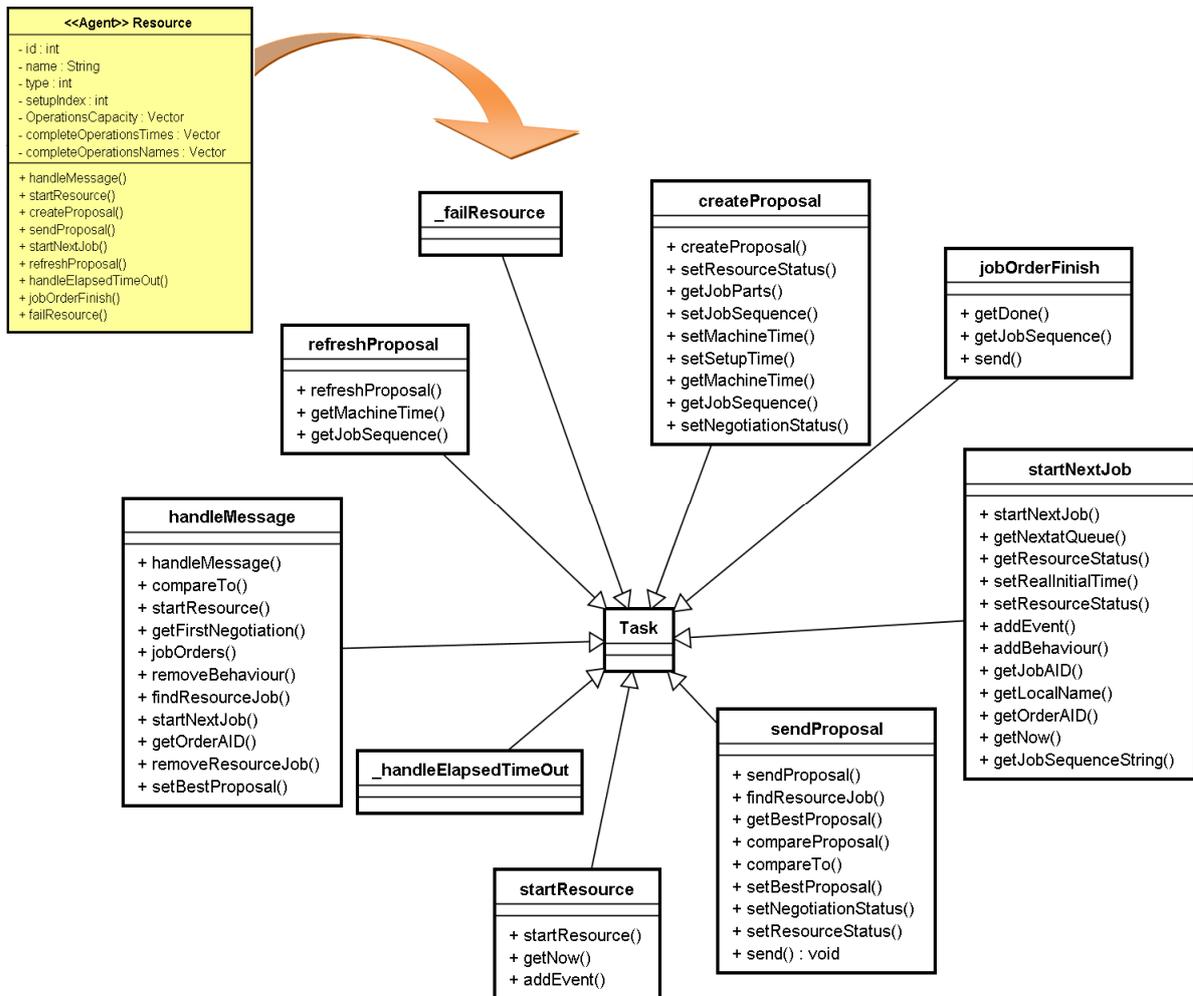


Figura 5.16 – Diagrama de definição da estrutura do agente *Resource*

## 5.8 Modelo de implementação de agentes - Sub-fase de definição do comportamento por agente

A metodologia PASSI não sugere nenhuma ferramenta específica para descrever o comportamento por agente. Optou-se, então, por manter uma padronização com o restante da metodologia, usando-se descrição através de texto e diagramas de estado UML. A figura 5.17

apresenta o meta-modelo da sub-fase de definição do comportamento por agente em função dos relacionamentos entre a documentação de trabalho e os diagramas gerados.

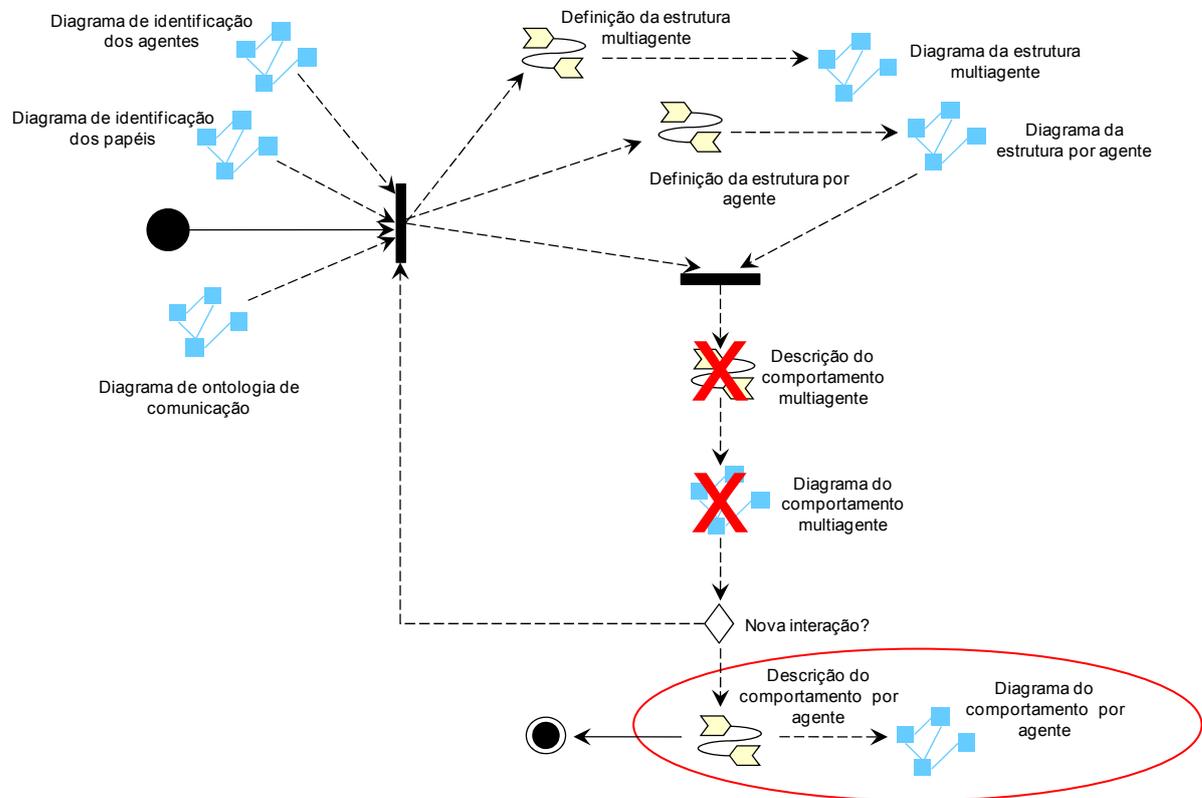


Figura 5.17– Sub-fase de de definição do comportamento por agente  
 FONTE: Cossentino, Sabatucci e Seidita (2003)

A seguir são apresentados os principais comportamentos dos agentes que compõem o modelo de simulação proposto.

### 5.8.1 Agente *Server*

Como citado anteriormente no item 5.1, o agente *Server* tem as seguintes responsabilidades: (a) disponibilizar os planos de processos ao agente *InputOrder* quando solicitado; (b) receber os novos pedidos enviados pelo usuário através do agente *InputOrder*; (c) preparar os agentes *Resources* para o posterior início da simulação; (d) manter uma lista de todas as ordens em processamento na simulação para organizar e controlar o andamento da simulação; (e) instanciar os agentes *JobOrders*, que representam as ordens de fabricação por

---

peça; e (f) instanciar o agente *Sync*, o qual é responsável por sincronizar os agentes *JobOrders* no chão-de-fábrica.

O ciclo de vida deste agente é composto por três comportamentos distintos, *WaitForOrders*, *StartSimulation* e *WaitForSimulation*, e por três diferentes estados, *WaitingOrders*, *StartingSimulation* e *WaitingSimulation*. A figura 5.18 apresenta o comportamento principal do agente *Server* durante seu ciclo de vida.

Ao ser instanciado, o agente *Server* assume o estado *WaitingOrders*, iniciando o comportamento *WaitForOrders*. Este comportamento é cíclico, ou seja, o agente *Server* permanece apto a receber novos pedidos do agente *InputOrder* durante todo o seu ciclo de vida. O comportamento *WaitForOrders* é interrompido quando: (a) for solicitado o resgate dos planos de processo das peças pelo agente *InputOrder*, através de uma mensagem *AllParts*; ou (b) receber uma solicitação de novos pedidos, também gerada pelo agente *InputOrder*, através de uma mensagem *New\_Order*, sendo que esta solicitação permanecerá armazenada até que ocorra o início da simulação.

No caso de uma mensagem *AllParts*, o agente *Server* irá recuperar de um banco de dados os planos de processos referentes a todas as peças disponíveis para simulação, enviando-as posteriormente ao agente *InputOrder* na forma de uma lista de objetos.

Ao iniciar uma simulação, o agente *Server* passa do estado *WaitingOrders* para o estado *StartingSimulation*, alterando também seu comportamento para *StartSimulation*. Neste estado, o agente *Server* transforma o pedido feito pelo usuário, o qual foi transmitido ao agente *Server* através da mensagem *New\_Order*, em diversas ordens de serviço, uma para cada peça envolvida no pedido original. Isto é feito através do método *createJobOrderAgents*, que irá gerar um agente *JobOrder* para cada ordem de serviço. Além disto, é feita uma solicitação para a identificação do *setup* de todos os recursos disponíveis através da mensagem *SetupID*. Essa mensagem contém o percentual de tempo de *setup* do tempo total de processamento a ser usado naquela simulação. Dessa forma, cada recurso consegue separar a parcela de *setup* e a parcela de processamento.

Após o envio da mensagem *SetupID*, o agente *Server* muda seu estado para *WaitingForResources*, permanecendo assim até que todos os agentes *Resources* disponibilizados no instante do envio da mensagem *SetupID* retornem uma mensagem de *Start*. Somente após isto ocorrer é que o agente *Server* irá instanciar o agente *Sync*, através do método *createSyncAgent*, encerrando o comportamento *StartSimulation* e adotando o comportamento *WaitForSimulation*, passando assim para o estado *WaitingSimulation*.

O estado *WaitingSimulation* permanece bloqueado até a chegada de uma mensagem *Order\_Done*, enviada pelo agente *JobOrder*, quando uma ordem de fabricação é finalizada.

---

Sempre que uma mensagem *Order\_Done* é recebida, o agente *Server* verifica se esta mensagem corresponde à última ordem a ser processada entre as que compunham aquela simulação, o que é feito pelo método *waitingOrderDone*.

**SERVER AGENT**

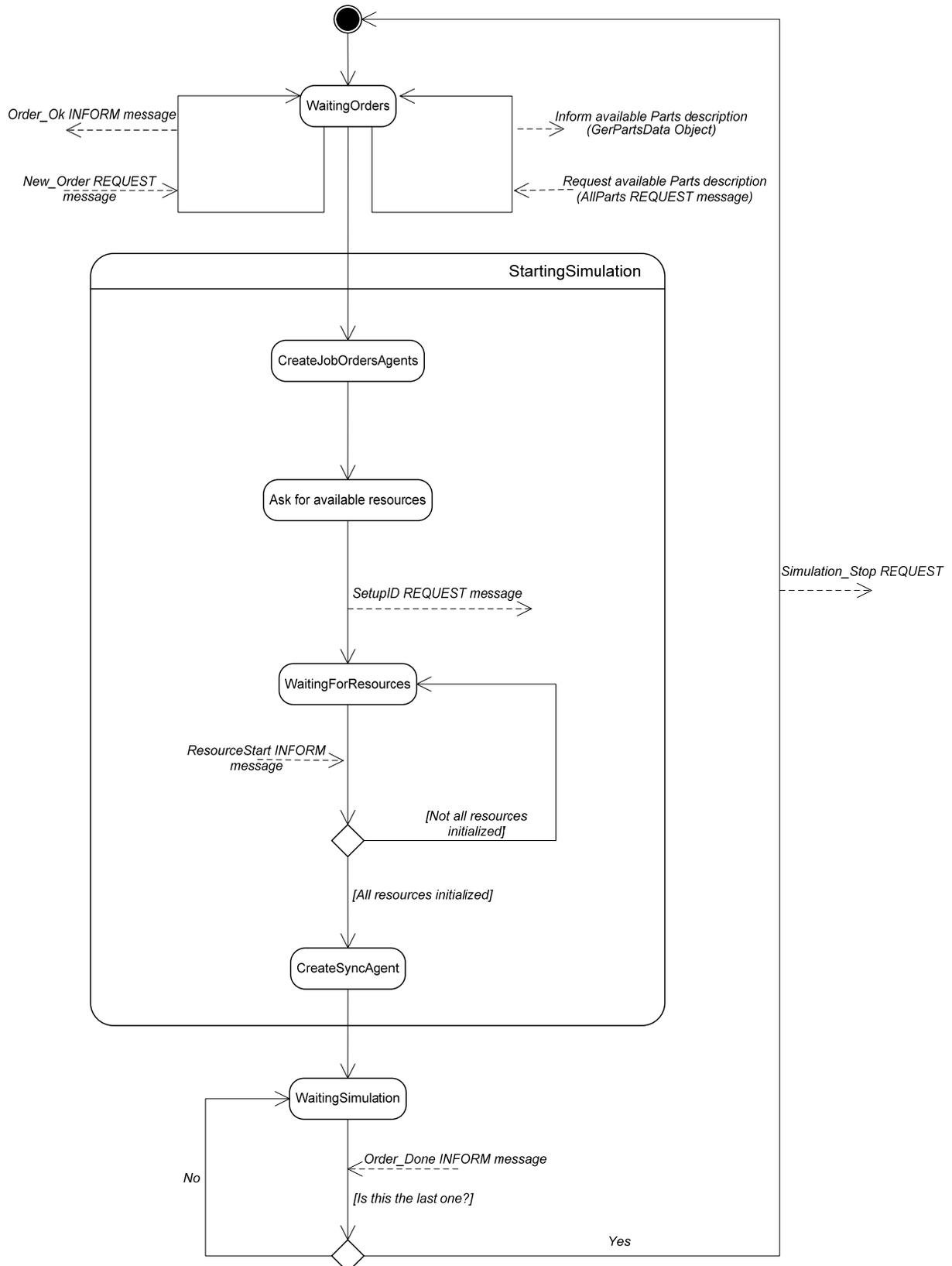


Figura 5.18 – Descrição do comportamento principal do agente *Server* durante seu ciclo de vida

---

Caso não seja a última ordem em processamento na simulação, o agente *Server* irá permanecer no estado *WaitingSimulation*, aguardando até que a última ordem seja finalizada para então enviar uma mensagem *Simulation\_Stop* a todos os recursos disponíveis indicando o fim da simulação.

A fim de facilitar as replicações necessárias de cada simulação, foi criado um mecanismo que não está mostrado no comportamento acima. Após o envio da mensagem *Simulation\_Stop* o agente *Server* deverá escolher entre adotar dois estados: (a) caso o usuário tenha configurado o programa para um número pré-determinado de simulações automáticas, o agente retornará ao estado *StartingSimulation*; (b) caso contrário o agente retornará ao estado *WaitingOrders*. Independentemente do estado que será adotado pelo agente, o comportamento *WaitForSimulation* será encerrado.

### 5.8.2 Agente JobOrder

Conforme citado no item 4.1.4, uma peça é a menor unidade de um pedido, ou seja, cada ordem corresponde a uma única peça, sendo definido no modelo pelos atributos: identificação da peça, que deve ser única; e tamanho de lote. Também como citado anteriormente no item 5.2, este agente tem por principais responsabilidades: (a) recuperar os planos de processo referentes a cada uma das peças a serem simuladas; e (b) negociar a programação das peças no chão-de-fábrica através da comparação das propostas enviadas pelos agentes *Resources*.

O ciclo de vida deste agente é composto por quatro comportamentos distintos: *Initialization*, *Negotiation*, *NegotiationTimeControl* e *WaitToNegotiate*, sendo que os dois últimos comportamentos auxiliam o comportamento *Negotiation*. Devido à sua maior complexidade, somente os comportamentos *Negotiation*, *NegotiationTimeControl* e *WaitToNegotiate* serão explicados em detalhes.

Os estados adotados durante o ciclo de vida do agente *JobOrder* são divididos em duas categorias distintas: negociação e físicos. É importante notar que esta diferenciação entre as categorias de estado de negociação e físicos foi criada em virtude dos agentes *JobOrder* trabalharem com estados compostos, ou seja, um agente *JobOrder* pode negociar as operações de seu plano de processo ao mesmo tempo que acompanha a fabricação de outras peças.

Os estados de negociação são divididos em *CallingForProposals* (CFP), *WaitingForProposals* (WFP), *WaitingForNegotiation* (WFN), *WaitingDoneMessage* (WDM) e, finalmente, *OnResourceQueue* (ORQ). Já os estados físicos se dividem entre *Producing*,

---

---

Done e Idle. A figura 5.19 apresenta a descrição do comportamento Negotiation, que é o comportamento principal do agente JobOrder, e os estados adotados durante seu ciclo de vida também são mostrados na figura.

Ao ser instanciado, o agente *JobOrder* assume o estado físico *Idle* e o estado de negociação *CallForProposal*, assim permanecendo até o recebimento de uma mensagem *Job\_Start* do agente *Sync*, o qual significa que o agente *JobOrder* deve iniciar o envio dos pedidos de propostas para a execução da peça que ele representa. Isto é feito através da mensagem *CFP (Calling For Proposals)* enviada a todos os agentes *Resources* disponíveis na simulação.

Assim que a mensagem é enviada, o agente passa para o estado de negociação *WaitForProposal*, permanecendo até que o seu comportamento auxiliar *NegotiationTimeControl* (a ser apresentado no item 5.8.3) avise do término do tempo de negociação, que marca o final do recebimento das propostas enviadas pelos agentes *Resources*. Nesse caso, é enviada uma mensagem *CANCEL* que sinaliza o encerramento de negociação com todos os recursos disponíveis na simulação.

Durante o tempo de negociação, o agente *JobOrder* poderá receber dois tipos de mensagem por parte dos agentes *Resources*: (a) *REFUSE*, que significa que o recurso que a enviou não possui capacidade de executar nenhuma operação presente no plano de processo da peça representada pelo agente *JobOrder*; (b) *PROPOSE*, representa a proposta, contendo o tempo de processamento previsto, a ser explicada no capítulo 6 do presente trabalho. A mensagem *PROPOSE* é enviada por quantos sejam os agentes *Resources* com capacidade de executar alguma operação do plano de processo da peça.

No caso do recebimento de uma mensagem *PROPOSE*, a mesma será tratada através do método *TreatProposeMessage*, que busca comparar as propostas recebidas, armazenando seu valor e comparando-as com propostas futuras. Duas alternativas são previstas neste caso: (a) se a proposta não for aceita, o agente *JobOrder* retorna ao estado *WaitForProposal* e envia uma mensagem *REJECT\_PROPOSAL* ao agente *Resource* que gerou a proposta; (b) se a proposta for aceita, uma mensagem *ACCEPT\_PROPOSAL* é enviada ao agente *Resource* que gerou a proposta.

---

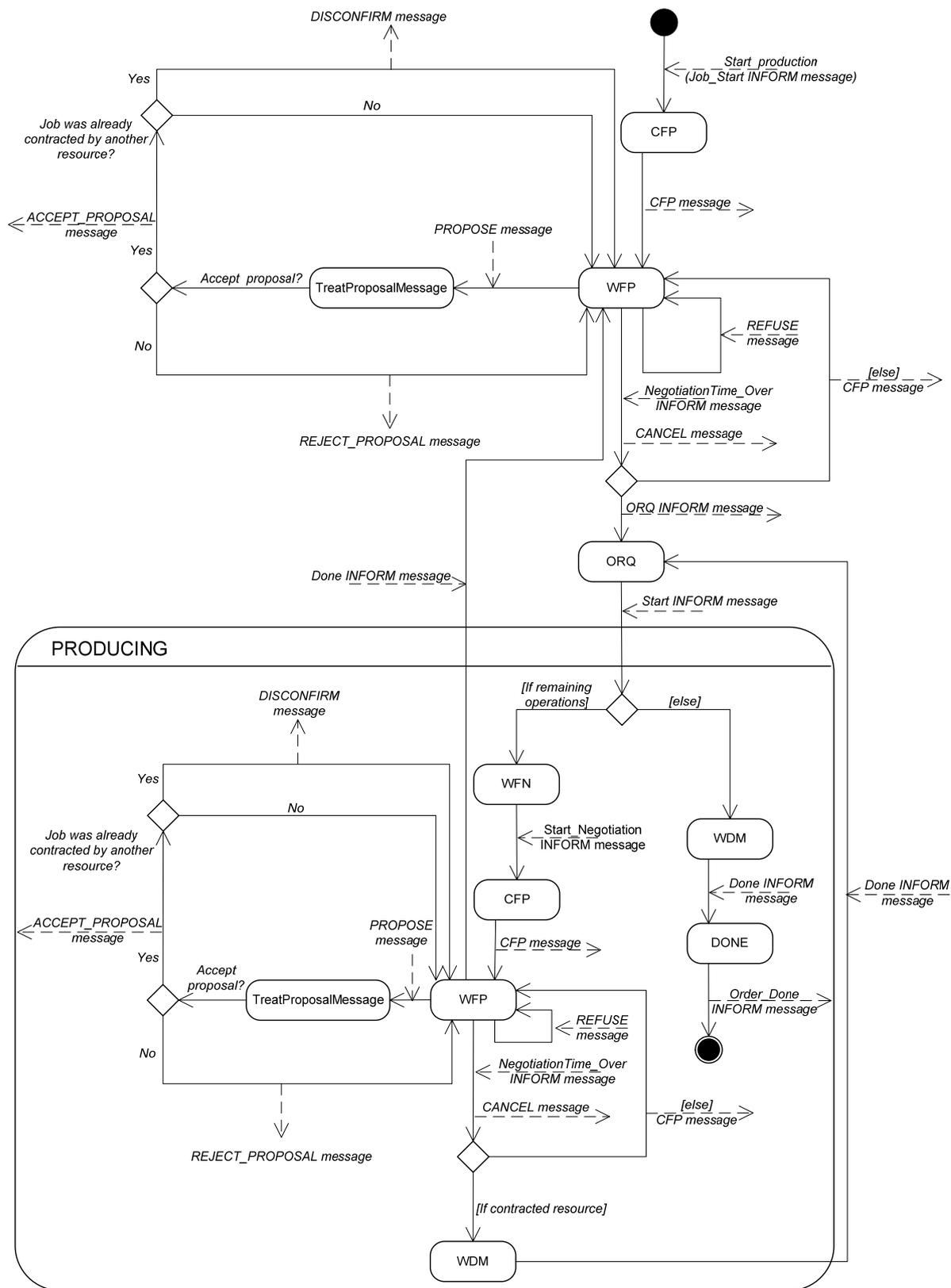


Figura 5.19 – Descrição do comportamento *Negotiation* adotado pelo agente *JobOrder* durante seu ciclo de vida

---

Após a proposta ser aceita, uma nova decisão deve ser tomada: (a) se a proposta aceita for a primeira a ser recebida pelo agente *JobOrder* naquela rodada de negociação, o mesmo retornará ao estado *WaitForProposal* até o término do tempo de negociação; (b) porém, pode ocorrer do agente *JobOrder* já ter aceito anteriormente uma proposta, sendo, neste caso, necessário enviar uma mensagem *DISCONFIRM* ao agente *Resource* que havia enviado a proposta aceita anteriormente e, somente após, retornar ao estado *WaitForProposal*. Com isto evita-se que o agente *JobOrder* permaneça com dois agentes *Resource* contratados simultaneamente para a execução da mesma peça ou operação.

Ao término do tempo de negociação, o agente *JobOrder* tem duas opções:

- a) Se este agente não tiver contratado nenhum agente *Resource*, ou seja, não tiver aceitado nenhuma das propostas recebidas, é enviada uma nova mensagem *CFP* aos agentes *Resource*, permanecendo o agente *JobOrder* no estado *WaitForProposal*; ou
- b) No caso do agente *JobOrder* já ter aceito uma proposta, ou seja, havendo contratado um agente *Resource*, é enviada uma mensagem *ORQ* (*On Resource Queue*) ao agente contratado, informando que o agente *JobOrder* está se encaminhando para a fila do agente *Resource*, passando o estado de negociação do agente *JobOrder* a ser *OnResourceQueue*.

O agente *JobOrder* irá permanecer no estado de negociação *OnResourceQueue* e no estado físico *Idle* até que receba uma mensagem *Start* vinda do agente *Resource* contratado. Esta mensagem indica o início da produção da ordem naquele recurso. Quando isto acontece, o estado físico do agente *JobOrder* passa a ser *Producing*. Novamente duas são as opções possíveis neste ponto:

- a) Caso a operação cujo início foi sinalizado pela mensagem *Start* seja a última a ser processada para o término da ordem, o agente *JobOrder* muda seu estado de negociação para *WaitDoneMessage*, onde permanece até o recebimento da mensagem *Done*. Esta mensagem indica o término da ordem no recurso. Ao receber esta mensagem, o estado de negociação do agente *JobOrder* passar a ser *Done*, e é enviada uma mensagem *Order\_Done* ao agente *Server*, indicando que a ordem representada pelo agente *JobOrder* foi finalizada, encerrando assim o ciclo de vida deste agente; ou
  - b) Se a operação negociada não for a última necessária para o término da ordem, o agente *JobOrder* muda seu estado de negociação para *WaitForNegotiation* permanecendo neste estado até receber uma mensagem *StartNegotiation* do comportamento auxiliar *WaitToNegotiation* (ver item 5.8.4 do presente trabalho), que indica o início do ciclo de negociação, conforme descrito anteriormente.
-

É importante notar que a função principal do estado *WaitForNegotiation* e do estado auxiliar *StartNegotiation* é evitar que sejam geradas filas pelo tempo de negociação entre o agente *JobOrder* e os agentes *Resource*. Isto ocorre porque a negociação da próxima operação a ser processada fica agendada para iniciar antes do término do processamento da operação atual.

A principal diferença entre o ciclo de negociação nos estados físicos *Idle* e *Producing* ocorre se houver o recebimento de uma mensagem *Done*, explicada anteriormente, durante o estado de negociação *WaitForProposal* no estado físico *Producing*. Neste caso, a mensagem será tratada como uma exceção, uma vez que, por alguma razão, a negociação entre os agentes *JobOrder* e *Resource* não foi concluída antes do término da operação corrente, o que implicará no retorno do agente *JobOrder* ao estado físico *Idle* e ao estado de negociação *WaitForProposal*.

O ciclo de negociação se repetirá até que a última operação a ser processada para o término da ordem seja finalizada, encerrando assim o ciclo de vida deste agente. Devido à complexidade do comportamento *Negotiation*, a figura 5.201 apresenta um esquema simplificado das mudanças que ocorrem nos estados físicos e de negociação durante o ciclo de vida do agente *JobOrder*, relacionando-as com seus eventos causadores.

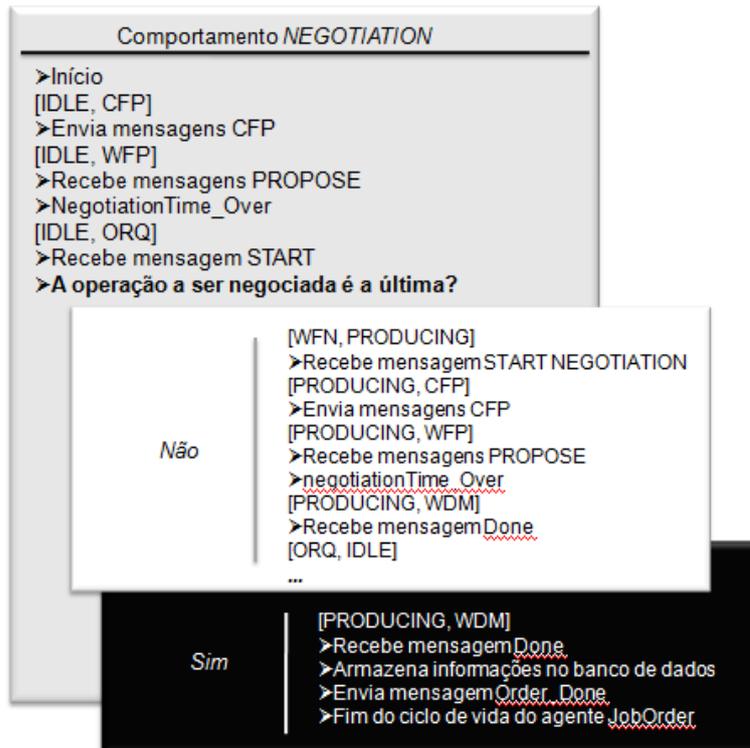


Figura 5.20 – Descrição simplificada dos estados compostos do agente *JobOrder* durante o ciclo de vida do comportamento *Negotiation*

### 5.8.3 Agente *JobOrder* – Comportamento auxiliar *NegotiationTimeControl*

Conforme citado anteriormente, além do comportamento principal, o agente *JobOrder* também conta com comportamentos auxiliares, sendo o primeiro deles *NegotiationTimeControl*. Este comportamento é responsável por controlar o tempo que o agente *JobOrder* irá permanecer aceitando mensagens *CFP* dos agentes *Resources*. A figura 5.21 apresenta o comportamento auxiliar *NegotiationTimeControl* do agente *JobOrder*.

Negotiation Time Control Behaviour

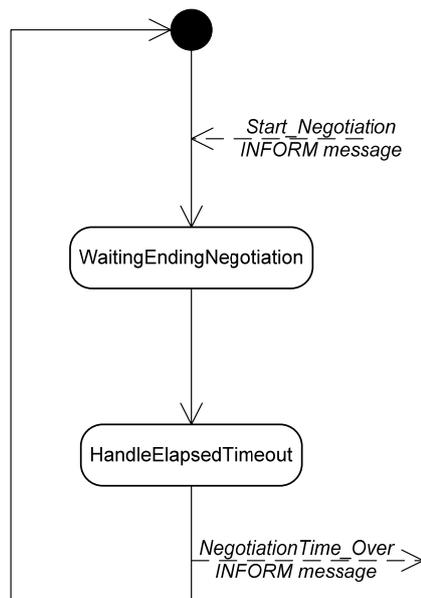


Figura 5.21 – Descrição do comportamento auxiliar *NegotiationTimeControl* do agente *JobOrder*

Este comportamento inicia assim que o agente *JobOrder* muda seu estado de negociação para *WaitForProposal* no comportamento principal *Negotiation*. Nesse instante é enviada uma mensagem *Start\_Negotiation*, cuja a origem e o destinatário é o próprio agente *JobOrder* informando ao comportamento auxiliar *NegotiationTimeControl* que deve mudar seu estado para *WaitingEndingNegotiation* (*WEN*), dando início à contagem do tempo para o término da negociação. Ao término deste tempo, o estado no comportamento auxiliar *NegotiationTimeControl* mudará para *HandleElapsedTimeout* e uma mensagem *NegotiationTime\_Over* é enviada ao comportamento *Negotiation* do agente *JobOrder*.

Uma exceção ocorre caso o ciclo de negociações esteja começando pela primeira vez no comportamento *Negotiation*. Isto significa que não haveria, naquele instante, nenhuma operação que compõe o plano de processo do agente *JobOrder* sendo processada. Neste caso

específico, o evento que iniciará o comportamento *NegotiationTimeControl* será a chegada de uma mensagem *Job\_Start* ao invés da mensagem *Start\_Negotiation*.

#### 5.8.4 Agente *JobOrder* – Comportamento auxiliar *WaitToNegotiate*

O comportamento auxiliar *WaitToNegotiate* é responsável por controlar o tempo que o agente *JobOrder* irá permanecer no estado *WaitForNegotiation*. Conforme citado anteriormente, o estado *WaitForNegotiation* do comportamento principal do agente *JobOrder* é necessário, pois a próxima negociação já está agendada para iniciar antes do término do processamento da operação atual pelo agente *Resource* contratado. A figura 5.22 abaixo apresenta o comportamento auxiliar *WaitToNegotiate* do agente *JobOrder*.

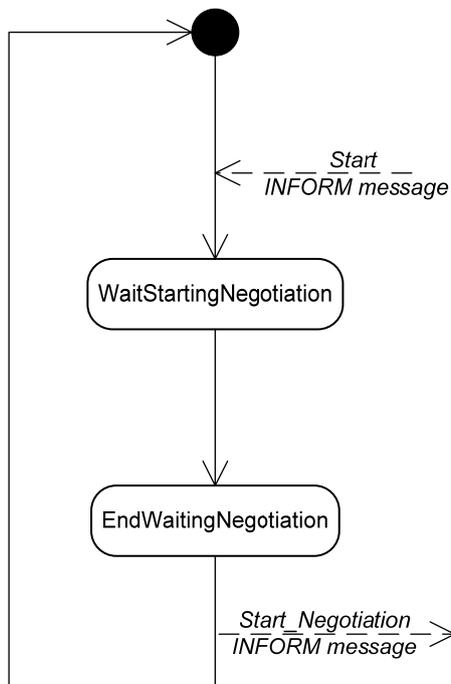


Figura 5.22 – Descrição do comportamento auxiliar *WaitToNegotiate* do agente *JobOrder*

Este comportamento inicia a partir da chegada de uma mensagem *Start* oriunda do agente *Resource* contratado pelo comportamento principal, *Negotiation*, do agente *JobOrder*, indicando o início da execução de uma ou mais operações que compõem o plano de processo da peça representada pelo agente.

Assim que a mensagem é recebida, o comportamento auxiliar *WaitToNegotiate* muda seu estado para *WaitStartingNegotiation* (*WSNeg*), dando início à contagem do tempo para a mudança do estado *WaitForNegotiation* para *CallForProposal* no comportamento principal

---

do agente *JobOrder*. Este intervalo é determinado baseando-se no tempo total de processamento da operação atual sendo executada pelo agente *JobOrder*, menos um valor pré-fixado, variando caso a caso.

Assim que o período de tempo termina, o estado do comportamento auxiliar *WaitToNegotiate* muda para *EndWaitingNegotiation* (*End\_WNeg*), enviando ao comportamento principal do agente *JobOrder* uma mensagem *StartNegotiation*, o que indica o início do ciclo de negociação do agente.

### 5.8.5 Agente Resource

Como citado anteriormente no item 5.2, cada *Resource* representa um dos recursos produtivos no chão-de-fábrica, tendo por principais responsabilidades: (a) recuperar os dados de capacidade de cada um dos recursos a partir de um banco de dados, bem como os tempos de processamento de cada uma das operações que compõem os planos de processos das peças; (b) controlar a fila do recurso; e (c) elaborar propostas de fabricação em resposta às mensagens *CFP* recebidas, baseando-se em sua disponibilidade durante o tempo de simulação.

O ciclo de vida deste agente é composto por 4 comportamentos distintos, *Negotiation*, *WaitToSend*, *RequestInfo* e *HandleQueue*, e por 2 estados diferentes *Idle* e *Producing*, dos quais somente os comportamentos *Negotiation* e *HandleQueue* serão explicados em detalhes, devido à sua complexidade. A figura 5.23 apresenta o comportamento *HandleQueue* do agente *Resource* durante seu ciclo de vida.

O comportamento *HandleQueue* é responsável pela criação e controle da fila de operações a serem processadas pelo agente *Resource*. Neste caso, cada operação é referente a uma parte do plano de processo da peça a ser executada, aqui representada pelo agente *JobOrder*.

Este comportamento inicia com o recebimento de uma mensagem *Simulation\_Start* enviada pelo agente *Server*, indicando o início da simulação. Neste instante, o método *StartResource* esvazia a fila de operações do agente *Resource* vazia (*JobOnQueue* = 0). A mensagem *Simulation\_Start* é respondida com o envio de uma mensagem *ResourceStart*, que indica ao agente *Server* que o recurso estará disponível naquela simulação. Neste instante, o agente *Resource* passa para o estado *IDLE* e fica aguardando o recebimento de uma nova mensagem, que pode ser: (a) *Simulation\_Stop*, enviada pelo agente *Server*; ou (b) *CFP*, enviada por um agente *JobOrder*.

---

Caso a mensagem recebida seja *Simulation\_Stop*, o agente *Resource* encerrará seu ciclo de vida. Porém, se a mensagem recebida for um *CFP*, será iniciada a negociação entre o agente *Resource* e o agente *JobOrder* de onde a mensagem se originou.

Neste ponto do trabalho, é interessante lembrar que, conforme apresentado nos itens 5.5.1 e 5.5.2, que tratam dos protocolos de negociação e renegociação desenvolvidos para o presente modelo de simulação, diferentes cenários podem ocorrer durante uma negociação. Estes cenários são definidos pela contratação ou não de um recurso, ou mesmo pelo cancelamento de uma ordem.

Então, para se descrever o comportamento *Negotiation*, será apresentado, deste ponto em diante, a problemática do tratamento da fila pelo agente *Resource*, partindo-se da premissa que este agente já foi contratado pelo agente *JobOrder* e que não haverá cancelamento desta ordem.

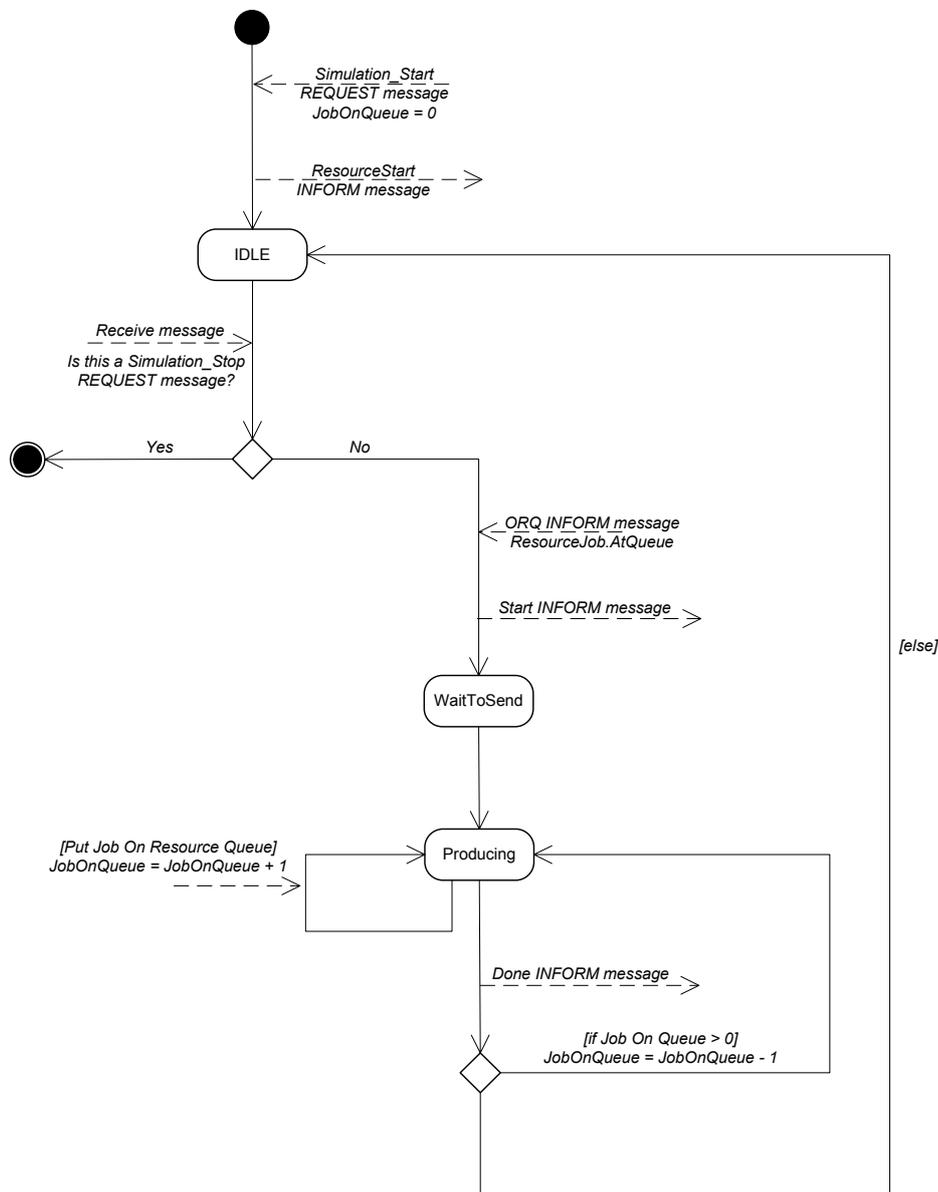


Figura 5.23 – Descrição do comportamento *HandleQueue* do agente *Resource* durante seu ciclo de vida

---

Após o término da negociação e a sua contratação, o agente *Resource* permanecerá no estado *IDLE* até o recebimento de uma mensagem *ORQ* enviada pelo agente *JobOrder* que o contratou. Esta mensagem indica a chegada desta ordem à fila de operações a serem processadas pelo recurso, onde, através do método *ResourceJob.aTQueue*, será adicionada no final da fila existente.

Tão logo o recurso seja liberado e a peça que estava na fila comece a ser processada, é enviada uma mensagem *Start* ao agente *JobOrder* contratante. Ao mesmo tempo o comportamento do agente *Resource* passa a ser *WaitToSend*. Este comportamento trabalha de forma análoga aos comportamentos *NegotiationTimeControl* e *WaitToNegotiate* já explicados anteriormente nos itens 5.8.3 e 5.8.4, havendo um contador interno que indica o fim do tempo de processamento da operação no recurso.

Assim que este tempo termina, o estado do agente *JobOrder* passa a ser *Producing* e é enviada uma mensagem *Done* ao agente *JobOrder* que o havia contratado, indicando assim o término da execução da ordem naquele recurso. Após o envio desta mensagem duas diferentes situações podem ocorrer:

- a) Caso a fila de operações do recurso não esteja vazia, ou seja, havendo mais operações a serem executadas enviadas por outros agentes *JobOrder*, o agente *Resource* permanecerá no estado *Producing* e irá subtrair a operação executada de sua fila de operações ( $JobOnQueue = JobOnQueue - 1$ ); ou
- b) Caso a operação executada tenha sido a última na fila de operações do agente *Resource*, o mesmo retornará ao estado *IDLE*, onde ficará aguardando por novas negociações.

É importante notar durante que no estado *Producing* o agente *Resource* poderá negociar com outros agentes *JobOrder*. Caso ocorra uma contratação, a nova operação será incluída na fila de operações do agente *Resource* contratado ( $JobOnQueue = JobOnQueue + 1$ ).

### 5.8.6 Agente *Resource* – comportamento *Negotiation*

O comportamento cíclico *Negotiation* do agente *Resource* tem como principal função o tratamento das mensagens recebidas pelo mesmo no decorrer do seu ciclo de negociação com os agentes *JobOrder*, conforme apresentado na figura 5.24.

Após o recebimento da mensagem *Simulation\_Start*, enviada pelo agente *Server*, indicando o início da simulação, o agente *Resource* inicia o método *HandleMessage*, ficando

---

---

no aguardo do recebimento de novas mensagens enviadas pelo agente *JobOrder* durante o ciclo de negociação.

A seguir serão descritos os dois tipos de mensagens possíveis de serem recebidas e tratadas pelo agente *Resource*, oriundas de um agente *JobOrder*, durante um ciclo de negociação.

- a) *ORQ (On Resource Queue)*– esta mensagem indica que a ordem chegou à fila de operações a serem processadas pelo recurso, sendo tratada pelo método *Treat\_ORQ*, que irá posicionar as operações negociadas com o agente *JobOrder* no final da fila de processamento do recurso através do método *ResourceJob.AtQueue*; e
- b) *CFP (Calling For Proposals)* – esta mensagem indica uma solicitação de proposta de processamento, sendo tratada pelo método *Treat\_CFP*, podendo resultar em dois tipos de ação: (a) caso o recurso tenha capacidade para executar alguma das operações presentes no plano de processo da ordem, será enviada uma proposta com o tempo previsto de processamento através de uma mensagem *PROPOSE*; ou (b) caso o recurso não tenha capacidade para executar alguma das operações presentes no plano de processo da ordem, ele irá declinar da ordem através de uma mensagem *REFUSE*.

As próximas mensagens a serem apresentadas podem utilizar como resposta um dos três estados:

- a) *Reject*, a qual indica que o agente *JobOrder* rejeitou a proposta enviada pelo agente *Resource*;
- b) *Contracted*, a qual indica que o agente *JobOrder* fechou contrato com o agente *Resource*, porém ainda não chegou na fila de processamento do mesmo; e
- c) *WaitingAnswer*, a qual indica que o agente *Resource* está no aguardo de uma resposta do agente *JobOrder* à sua proposta.

Os estados abaixo são necessários e referem-se ao modo com que um agente *Resource* considera e classifica internamente para seu controle, um agente *JobOrder*.

- a) *ACCEPT\_PROPOSAL* – tratada pelo método *Treat\_ACCEPT\_PROPOSAL*, muda o estado do agente *JobOrder* para *Contracted*;
  - b) *REJECT\_PROPOSAL* – tratada pelo método *Treat\_REJECT\_PROPOSAL*, muda o estado do agente *JobOrder* para *Reject*;
  - c) *DISCONFIRM* – tratada pelo método *Treat\_DISCONFIRM*, muda o estado do agente *JobOrder* para *Reject*; e
-

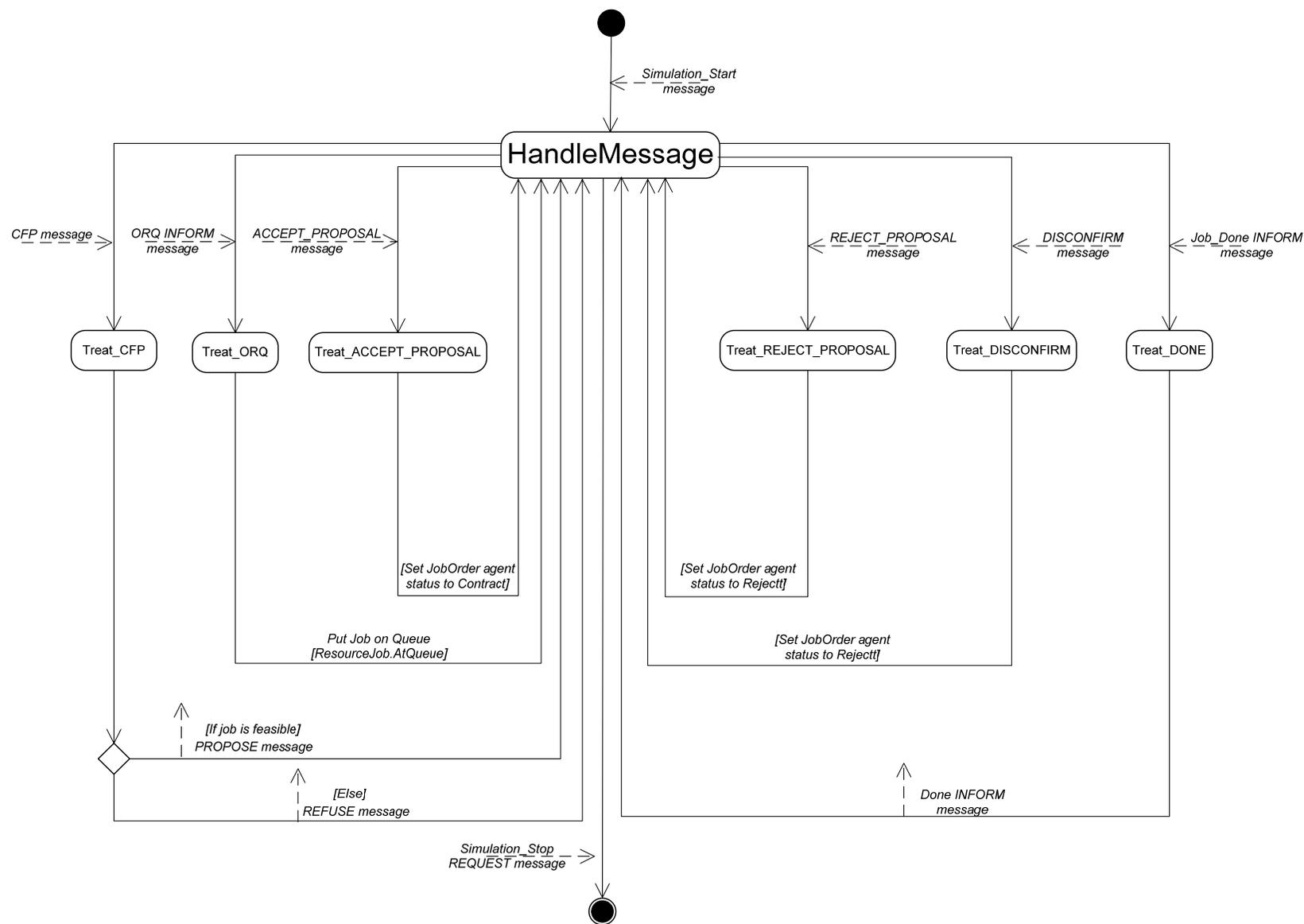


Figura 5.24 – Descrição do comportamento *Negotiation* adotado pelo agente *Resource* durante seu ciclo de vida

---

d) *JOB\_DONE* – mensagem interna do agente *Resource* para ele mesmo, indicando o término do processamento de uma ordem – tratada pelo método *Treat\_DONE*. Envia ao agente *JobOrder* uma mensagem *Done* referente à ordem finalizada, indicando o fim de seu processamento naquele recurso.

## 5.9 Conclusão do capítulo

Este capítulo tratou do desenvolvimento computacional do modelo de simulação proposto através das fases da metodologia PASSI, bem como de parte de sua verificação, com especial atenção aos processos sociais dos agentes e sua consistência como organização multiagente.

Para este fim, foram apresentadas as fases da metodologia PASSI que tratam da verificação do modelo, iniciando com a continuidade da fase de requisitos do sistema, iniciada no capítulo 4. Nesta etapa, os estereótipos de relações entre casos de uso mapeados no item 4.1.11 que trata dos requisitos do domínio, são convertidos em comunicação.

Em seguida foi apresentada a fase de sociedade de agentes, responsável por modelar a solução em função de seus papéis, interações, dependências e ontologia. Nesta fase foram descritos os protocolos de negociação e renegociação desenvolvidos especialmente para o modelo proposto. O protocolo de negociação tem como objetivo permitir a programação das ordens de produção entre os recursos alocados no chão-de-fábrica simulado, permitindo a negociação das ordens entre os agentes *JobOrder* e *Resource*.

O capítulo se encerra com o modelo de implementação, responsável por apresentar a arquitetura proposta em função de suas classes e métodos.

---

---

## CAPÍTULO 6 – FORMULAÇÃO DA PROPOSTA DE NEGOCIAÇÃO E DESCRIÇÃO DO CASO DE TESTE

Este capítulo inicia com a formulação da proposta de negociação, usada como critério de desempate e de agendamento entre as diversas ordens que estejam concorrendo concomitantemente aos recursos disponíveis durante uma simulação.

Será visto que a abordagem de negociação utilizada neste modelo é uma versão estendida daquela proposta por USHER (2003), à qual é acrescentado um mecanismo de agrupamento de operações. Os tempos que compõem uma negociação também serão tratados e discutidos neste capítulo.

Em seguida será apresentado o caso de teste desenvolvido por YEO et al. (2003), o qual será utilizado para a validação do modelo.

### 6.1 Formulação da proposta de negociação

Uma negociação e seus respectivos cenários não ocorrem aleatoriamente. Eles são governados por mecanismos que regem as regras de encontro (do inglês *rules of encounter*) entre os agentes (WOOLDRIDGE, 2002). Estas regras definem a estratégia que um agente, de forma individual, irá utilizar durante uma negociação a fim de alcançar seus objetivos, bem como os objetivos globais do sistema multiagente.

Durante a apresentação dos protocolos de negociação e renegociação, desenvolvidos e utilizados neste modelo de simulação (itens 5.5.1 e 5.5.2), o conceito de proposta foi citado algumas vezes, mas até este ponto ela não foi descrita, o que é feito formalmente abaixo:

Neste trabalho, uma proposta será vista como um critério de desempate, permitindo aos agentes chegar a um acordo mutuamente benéfico em assuntos de interesse comum (WOOLDRIDGE, 2002).

Definição 6.1  
Proposta

A formulação da proposta desenvolvida tem como base aquela sugerida por USHER (2003), e considera todos os tempos envolvidos em uma negociação desta natureza, que são os seguintes: tempos de máquina, tempos internos ao sistema e tempo de troca de mensagens entre agentes. Cada um destes tempos será discutido no decorrer do presente capítulo.

---

### 6.1.1 Duração do período de negociação

Normalmente, em sistemas baseados em multiagentes, a duração de uma negociação é determinada somente pelo tempo de troca de mensagens entre os agentes participantes, ou seja, a duração da negociação corresponde diretamente ao tempo que transcorre entre o envio e a resposta de uma mensagem (WONG, 2006a); (WONG, 2006b); ( SOUSA et al., 2004); (ZHOU et al., 2003); (WANG et al., 2003) e (LIM e ZHANG, 2004).

No modelo proposto por USHER (2003), o prazo final de uma negociação não é determinado diretamente pelo tempo de troca de mensagens. USHER (2003) propõe uma abordagem onde o período de negociação inicia e termina antes do final do processamento da operação corrente. Então, ao invés de se ter uma negociação simples, formada apenas pela troca de mensagens entre uma única ordem e um recurso, existe agora a possibilidade de ocorrerem *n* negociações, envolvendo uma ordem e inúmeros recursos simultaneamente. Estas negociações ocorrerão independentemente do seu tempo de troca de mensagens, dentro de em um determinado período de tempo, cujo intervalo é definido por valores percentuais fixos, subtraídos do tempo total de processamento, conforme apresentado na figura 6.1.

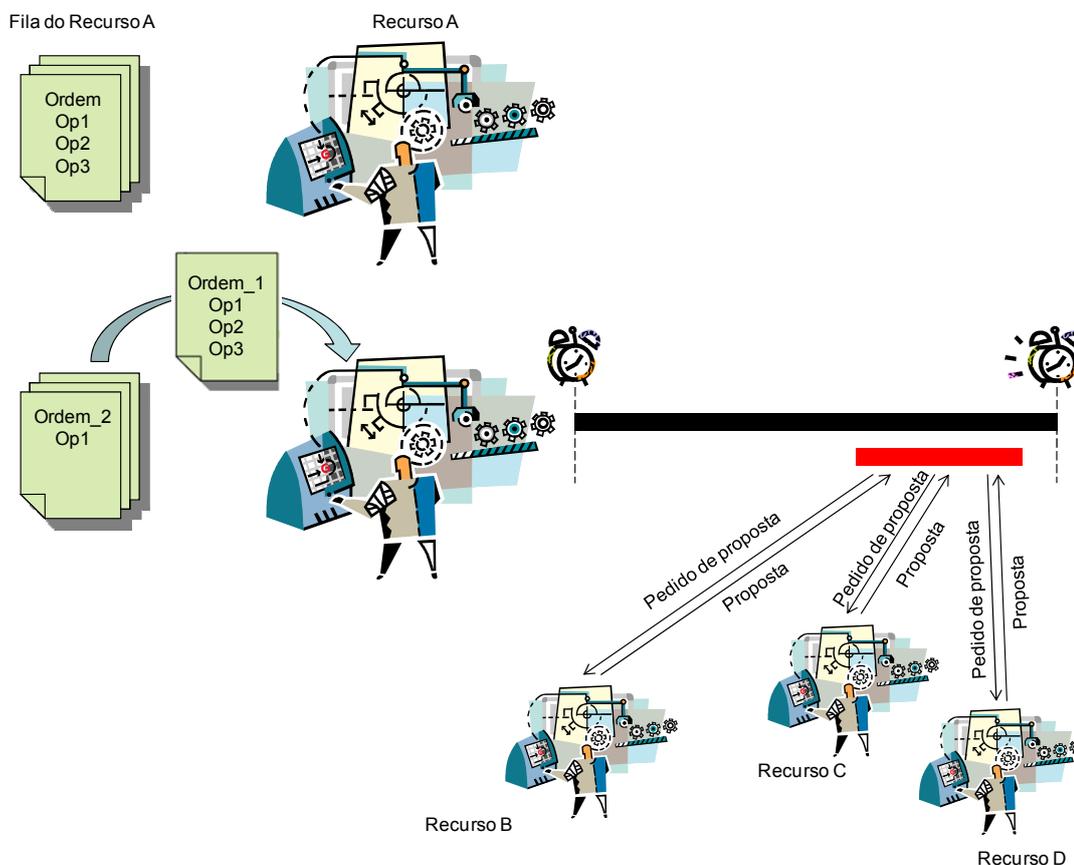


Figura 6.1 – Período de negociação, baseado em USHER (2003)

Este período de negociação indicará quando a ordem deverá iniciar o envio do pedido de propostas (CFP) aos recursos participantes daquela simulação, bem como quando deverá interromper o recebimento destas propostas e fechar um contrato com o recurso que melhor atender as suas necessidades.

No exemplo apresentado na figura 6.1 (a), observa-se que várias ordens de produção estão aguardando para iniciar o processamento de uma ou mais operações na fila do recurso A, lembrando que estas ordens negociaram anteriormente quais operações seriam executadas por este recurso.

Tão logo a ordem 1 saia da fila do recurso A e inicie seu processamento neste recurso, figura 6.1 (b), inicia-se também a contagem de seu tempo total de processamento, o qual já foi previsto pelo recurso e enviado na proposta de negociação, aqui representado pela barra de tempo na cor preta. Maiores detalhes sobre os tempos e a formulação da proposta de negociação serão tratados nos itens 6.1.3 e 6.1.4 do presente capítulo. A partir de certo ponto do início do tempo de processamento da ordem 1 no recurso A, inicia o período de negociação, representado pela barra de tempo na cor vermelha. O início da contagem do tempo do período de negociação é definido através de um valor percentual em relação ao tempo total de processamento, bem como o término deste período, que deve se dar antes do término do tempo de processamento da ordem 1. Ao forçar o término do tempo de negociação antes do término do processamento no recurso A tende-se a evitar a formação de filas que poderiam ser geradas entre a troca de recursos.

USHER (2003) explica que ao se considerar um intervalo de tempo definido, obtém-se algumas vantagens sobre o tempo convencional de negociação, especialmente sob o ponto de vista de uma ordem de produção. Isto acontece porque ao se permitir a negociação simultânea, sem a necessidade de uma finalização imediata, possibilita-se a comparação entre as propostas recebidas dentro do tempo limite de negociação, o que leva a uma melhor escolha dentro das condições momentâneas do chão-de-fábrica.

Sob o ponto de vista do recurso, a análise de propostas concomitantes permite a escolha do recurso baseado em sua ocupação, possibilitando uma melhor distribuição do carregamento de máquina e a redução de filas de processamento, além de evitar a criação de recursos gargalos<sup>11</sup>.

---

<sup>11</sup> SADEH (1991) define recurso gargalo como aquele cuja utilização é próxima ou maior do que sua capacidade disponível.

---

---

### 6.1.2 Agrupamento de operações

Embora o mecanismo de negociação desenvolvido por USHER (2003) apresente vantagens, no decorrer do desenvolvimento deste trabalho foi detectada uma importante limitação daquele mecanismo de negociação, gerando uma oportunidade de melhoria, a qual será discutida a seguir.

Em USHER (2003), a cada rodada de negociação a ordem cria e mantém uma lista de recursos a serem contatados. Esta lista é gerada a partir da capacidade tecnológica do recurso, ou seja, sua habilidade em executar, ou não, uma das operações requeridas no plano de processo da peça. O principal objetivo deste armazenamento de possíveis recursos para a execução de uma operação é a redução do número de mensagens a serem enviadas no transcorrer de uma negociação, uma vez que as mensagens serão enviadas somente para os recursos presentes na lista de contato da operação seguinte a ser negociada, em vez de serem enviadas a todos os recursos presentes no chão-de-fábrica.

Entretanto, foi observado que, apesar de considerar um intervalo de negociação pré-definido, o que permite a uma mesma ordem negociar de forma simultânea com vários recursos, e de se limitar o número de comunicações durante esta negociação, no mecanismo sugerido por USHER (2003) cada ordem pode negociar apenas uma operação por vez. Ou seja, não há nenhuma previsão para o agrupamento de operações pertencentes a um mesmo plano de processo.

As implicações decorrentes da falta de agrupamento são várias, sendo que no âmbito deste projeto as principais são:

Este tipo de abordagem equivale a considerar o tempo de *setup* de forma independente da seqüência de operações, o que pode vir a gerar perdas tanto no tempo de fluxo médio (*flow time*) de uma dada ordem quanto no tempo total de conclusão (*makespan*) de um grupo de ordens;

Ocorrem perdas provenientes dos tempos de *setup* de preparação das máquinas e de fixação das peças, visto que o modelo não se preocupa com as trocas desnecessárias de recursos entre operações; e

Não possibilita o uso, e posterior estudo, da flexibilidade de máquina, já descrita no item 4.1.10, o que seria prejudicial ao cumprimento dos objetivos estabelecidos no capítulo 1 do presente trabalho.

A figura 6.2 ilustra as diferenças entre as abordagens de negociação com e sem agrupamento de operações sob o ponto de vista do aumento no número de possibilidades de

---

processamento a serem negociadas. Nesta figura é representada a diferença entre as abordagens na primeira rodada de negociações entre a peça 4 e os recursos aqui denominados de CE1, CE2, FC1, FC2 e FR1.

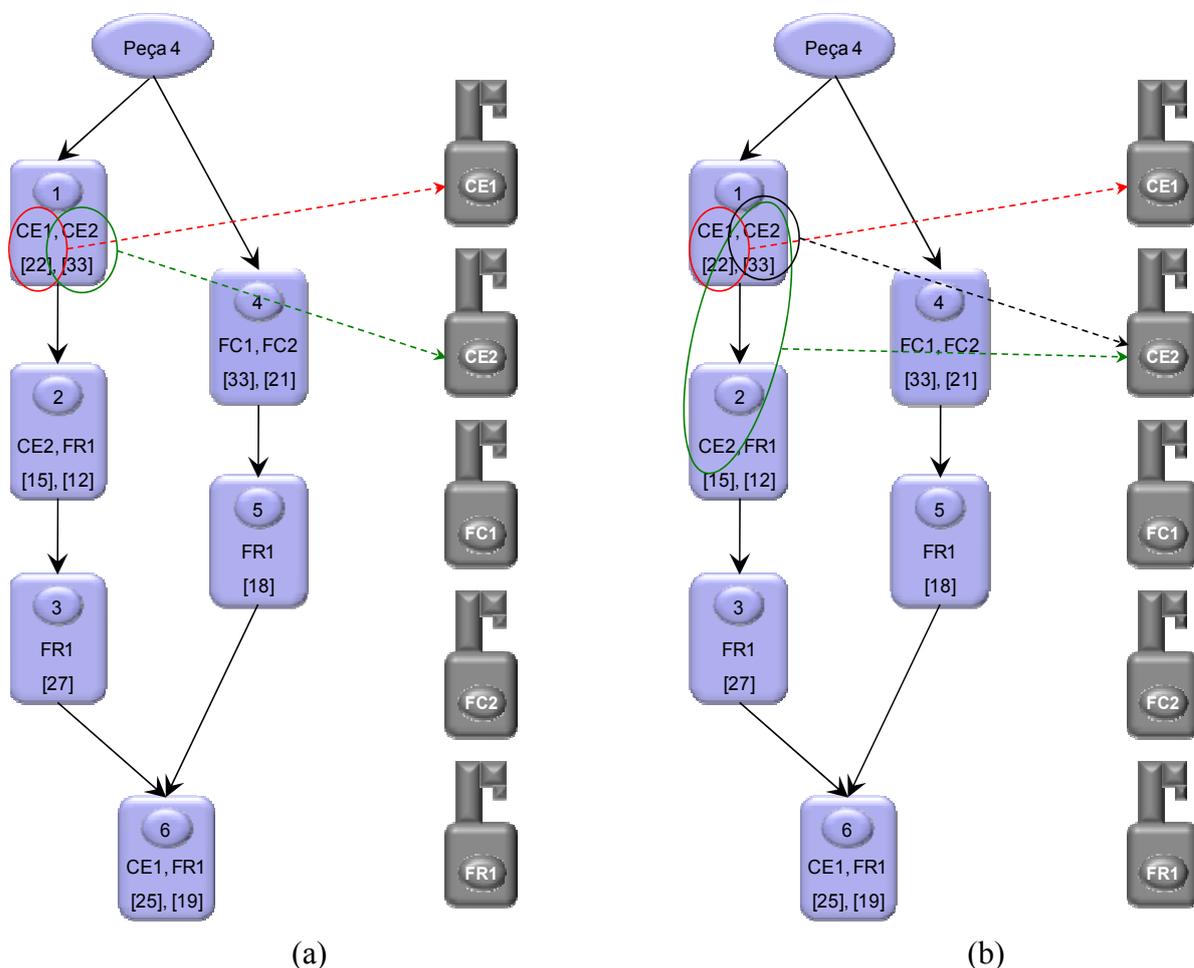


Figura 6.2 – (a) Negociação sem agrupamento de operações; (b) Negociação com agrupamento de operações

Ao ser utilizada a abordagem de negociação sem agrupamento (figura 7.1 a), somente a operação 1 inicia sua negociação com os recursos, possibilitando que a peça 4 venha a ser processada de apenas duas formas: executando-se a operação 1 no recurso CE1, ou no recurso CE2. Isto ocorre em função da existência de relações de precedência no plano de processo da peça, o que limita o número de operações a serem negociadas por vez.

Ao se observar novamente a rodada de negociações, agora sob o prisma da abordagem com agrupamento de operações, sugerida e adotada neste trabalho, é possível perceber que a mesma permite o rastreamento das operações que poderão ser agrupadas em um dado recurso. Para que este rastreamento seja válido, é necessário que sejam respeitadas as relações de precedência impostas no plano de processo da peça. Desta forma, é permitido à peça 4 continuar negociando a mesma operação 1 com os recursos CE1, CE2, FC1, FC2 e FR1, além

de adicionar a negociação da operação 2. Após o rastreamento e agrupamento das operações, condiciona-se que as operações 1 e 2 sejam executadas no mesmo recurso, e que a operação 2 somente seja processada após o término da operação 1.

Desta forma a peça 4 passa a ter a possibilidade de ser processada de três formas distintas: operação 1 no recurso CE1; operação 1 no recurso CE2; ou operação 1 e 2 no recurso CE2, conforme o melhor tempo de proposta recebido, a ser discutido a partir do item 7.1.3 do presente trabalho.

Sob o ponto de vista da contribuição no modelo desenvolvido neste trabalho, o agrupamento de operações maximiza o número de possibilidades de processamento por rodada de negociação, sempre que o plano de processo assim o permitir.

### 6.1.3 Cálculo do tempo de proposta e a formação da parcela de tempos internos ao sistema

Conforme citado no início deste capítulo, a formulação da proposta desenvolvida leva em consideração todos os tempos envolvidos em uma negociação. Estes tempos se dividem em tempos de máquina, que são os tempos de processamento e *setups*, e em tempos internos ao sistema, como o tempo de troca de mensagens entre agentes, por exemplo. A soma destes tempos irá resultar no tempo de proposta.

A nomenclatura utilizada para a representação das variáveis que compõem o tempo de proposta é uma adaptação de CONWAY et al. (2003), onde:

$i$  – número de ordens a serem processados;  $1 \leq i \leq n$ ;

$j$  – número de operações de uma ordem;  $1 \leq j \leq g_i$ ;

$g_i$  – número total de operações em um trabalho;

$p_{i,m}$  – quantidade de tempo requerida para o recurso  $m$  processar a ordem  $i$ ;

$s_{i,m}$  – tempo total de setup requerido pelo recurso  $m$  para processar a ordem  $i$ ;

$sp_{i,m}$  – tempo de setup de preparação requerido pelo recurso  $m$  para processar a ordem  $i$ . Este valor é considerado independente do tamanho do lote;

$sf_{i,m}$  – tempo de setup de fixação requerido pelo recurso  $m$  para processar a ordem  $i$ . Este valor é dependente do tamanho do lote a ser processado.

As equações (1) a (4), apresentadas a seguir, representam os tempos que compõem uma proposta, onde a primeira parcela ( $Tq_m$ ) da equação (1) corresponde à soma de todos os tempos internos ao sistema, que incluem os tempos de espera, contratação e fila do recurso, os

quais serão descritos na seqüência deste capítulo. Já a segunda parcela da equação (1) corresponde aos tempos de máquina, anteriormente citados.

$$Tempo\_de\_proposta = Tq_m + \left( \frac{\sum_{j=1}^{g_i} p_{i,m} + s_{i,m}}{g_i} \right) \quad (1)$$

$$Tq_m = \sum_{i=1}^n (q_i + c_i + w_i) \quad (2)$$

$$s_{i,m} = sp_{i,m} + sf_{i,m} * batch\_size \quad (3)$$

$$Tempo\_de\_proposta = \sum_{i=1}^n (q_i + c_i + w_i) + \left( \frac{\sum_{j=1}^{g_i} p_{i,m} + sp_{i,m} + sf_{i,m} * batch\_size}{g_i} \right) \quad (4)$$

Onde:

$w_i$  → tempo de espera (*waiting time*), que corresponde ao intervalo de tempo entre o envio da mensagem solicitando propostas (*CFP*) pelo agente iniciador (*JobOrder*) e sua resposta final contratando ou rejeitando o agente participante (*Resource*);

$c_i$  → tempo de contrato (*contract time*), que é o intervalo de tempo entre o aceite da proposta recebida pelo agente iniciador (*JobOrder*) até a chegada desta ordem na fila de processamento do recurso. Este tempo pode ser nulo ou não, dependendo se a ordem que firmou contrato com o recurso já está finalizada ou se ainda está em processamento no recurso anterior; e

$q_i$  → tempo de fila (*queue time*), que corresponde ao intervalo de tempo entre a chegada da ordem na fila do recurso contratado e o início do seu processamento pelo mesmo.

A fim de facilitar a compreensão dos tempos internos ao sistema, é apresentada na figura 6.3 uma negociação entre os agentes *JobOrder* e *Resource*.

Conforme já descrito no item 5.5.1, que trata dos protocolos de negociação utilizados neste modelo, ao iniciar uma negociação o agente *JobOrder* envia uma mensagem (*CFP*) ao agente *Resource* solicitando propostas para o processamento de suas operações. Neste instante começa a ser incrementado o tempo de espera ( $w_i$ ), que será computado até a resposta final do agente *JobOrder*, contratando (*Accept*) ou rejeitando (*Reject*) o agente *Resource*. Se o agente *Resource* é contratado, o tempo de espera total é armazenado em uma variável temporária de mesmo nome. Caso contrário, este tempo de espera é descartado e a variável relativa a esta negociação zerada.

Assim que o recurso é contratado, cria-se uma nova variável de tempo. Para compreendê-la é importante lembrar que, conforme citado no início deste capítulo, uma negociação deve se encerrar antes do final do processamento da operação atual. Isto significa que entre o fechamento do contrato e a chegada da ordem ao novo recurso, seja para o início do seu processamento, seja para se posicionar na fila do mesmo, transcorrerá um período de tempo. Este intervalo é chamado de tempo de contratação ( $c_i$ ) e é armazenado em uma variável temporária de mesmo nome.

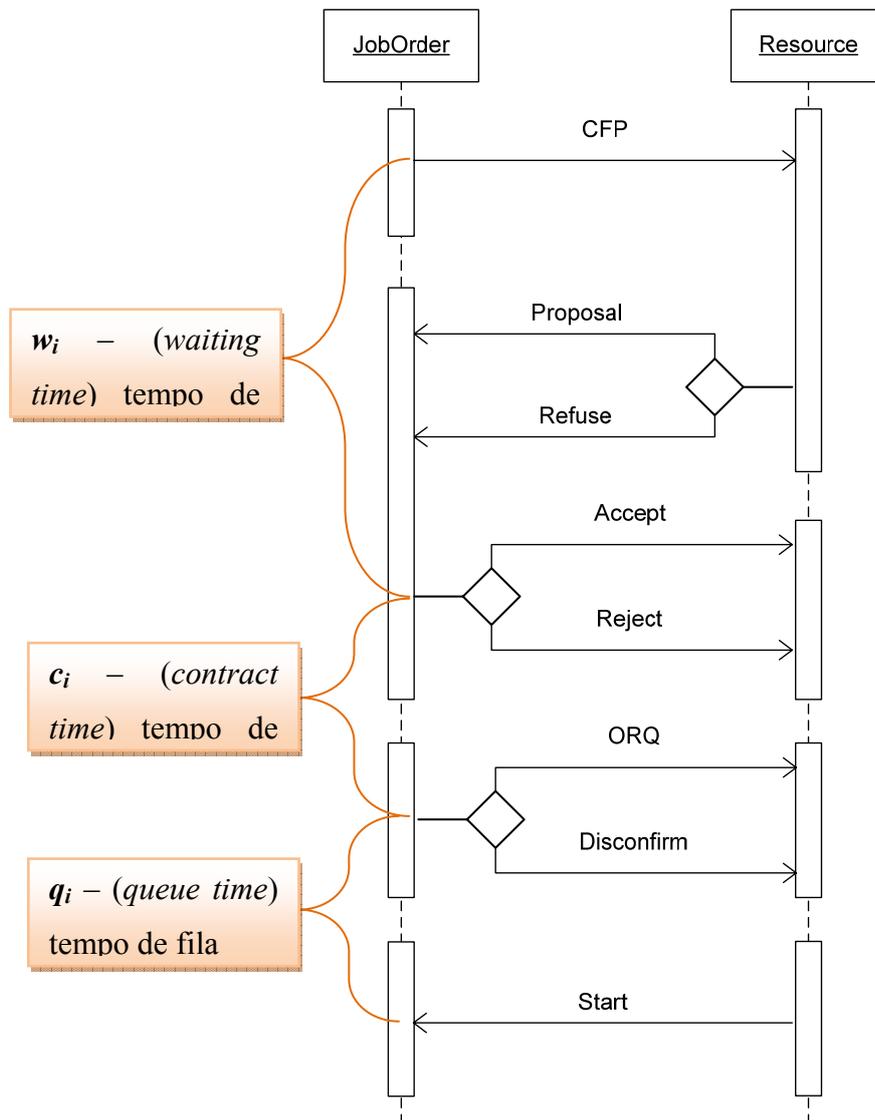


Figura 6.3 – Tempos internos ao sistema

Tão logo a ordem termine seu processamento no recurso atual, duas situações poderão ocorrer:

Se a ordem for encaminhada à fila do recurso contratado, uma nova variável de tempo é criada. Esta variável armazena o tempo entre a chegada da ordem na fila do recurso e o início de seu processamento no mesmo, e é chamada de tempo de fila ( $q_i$ ); ou

---

Caso a ordem inicie imediatamente seu processamento, a variável armazenada será o tempo de processamento ( $p_{i,m}$ ) anteriormente descrito, e que não faz parte dos tempos internos ao sistema.

A fim de evitar conflitos e contratações não cumpridas, o que poderia gerar resultados finais baseados em um modelo de simulação não confiável, uma ordem pode realizar um cancelamento de contratação (*disconfirm*) de um recurso em qualquer momento, porém somente até o início de seu processamento. Quando este cancelamento ocorrer, a variável de tempo que estiver sendo incrementada (referente àquela negociação) será automaticamente zerada.

É importante observar que se a ordem já estiver na fila do recurso contratado, um pedido de cancelamento (*disconfirm*) será penalizado com um aumento do seu tempo de processamento no novo recurso contratado.

#### 6.1.4 A formação da parcela de tempos de máquina

Conforme citado anteriormente, a segunda parcela que compõe o tempo de proposta é relativa aos tempos de máquina. Este tempo é formado pelas parcelas de processamento, *setup* de preparação e de fixação, sendo que cada parcela varia conforme a operação a ser realizada e o recurso escolhido para este processamento.

Devido ao mecanismo de agrupamento de operações proposto neste trabalho, foi necessário estabelecer um critério para a composição da parcela de tempo relativo ao *setup* de preparação ( $sp_{i,m}$ ). Isto ocorre porque apesar de diferentes operações exigirem uma preparação de máquina distinta, na maioria dos casos é possível executar as preparações em paralelo ou de forma complementar. Assim, quando do agrupamento de duas ou mais operações, optou-se por utilizar somente o maior tempo de *setup* de preparação envolvido. Já os tempos de processamento e de *setup* de fixação das operações serão resultado da soma direta dos valores relativos às operações agrupadas.

A fim de exemplificar o funcionamento deste mecanismo, é apresentada na tabela 6.1 a matriz de tempos da peça 4, que havia sido mostrada na figura 6.2. Nesta tabela pode-se observar que o tempo total de processamento da peça 4 é decomposto em três parcelas, sendo 20% para o tempo gasto no processamento da operação, 70% para a preparação do recurso e ferramentais, e finalmente 10% para a fixação da peça no mesmo. Os valores de 20%, 70% e 10% foram arbitrados a fim de ilustrar o exemplo.

---

Tabela 6.1 – Tabela de tempos da peça 4

Operação	Recurso	Tempo de processamento 20%	Setup de preparação 70%	Setup de fixação 10%	Tempo total de máquina
1	CE1	4,4	15,4	2,2	22
	CE2	6,6	23,1	3,3	33
2	CE2	3,0	10,5	1,5	15
	FR1	2,4	8,4	1,2	12
3	FR1	5,4	18,9	2,7	27
4	FC1	6,6	23,1	3,3	33
	FC2	4,2	14,7	2,1	21
5	FR1	3,6	12,6	1,8	18

Utilizando-se a abordagem sem agrupamento e considerando a matriz de tempos na tabela 7.1, será tomada como exemplo a negociação entre a peça 4 e os recursos CE1, CE2, FC1, FC2 e FR1, representados na figura 6.4. Dentro destes parâmetros, para um lote de peças igual a um o tempo de máquina<sup>12</sup>, na primeira negociação será de:

$$\text{Tempo de máquina } op_{1,ce1} = \left( \frac{\sum_{j=1}^1 p_{1,ce1} + sp_{1,ce1} + sf_{1,ce1} * 1}{1} \right)$$

$$\text{Tempo de máquina } op_{1,ce1} = \left( \frac{4,4 + 15,4 + 2,2 * 1}{1} \right)$$

$$\text{Tempo de máquina } op_{1,ce1} = 22$$

O mesmo raciocínio pode ser aplicado para a negociação da operação 2 com o recurso CE2:

$$\text{Tempo de máquina } op_{2,ce2} = \left( \frac{\sum_{j=1}^1 p_{2,ce2} + sp_{2,ce2} + sf_{2,ce2} * 1}{1} \right)$$

$$\text{Tempo de máquina } op_{2,ce2} = \left( \frac{3,0 + 10,5 + 1,5 * 1}{1} \right)$$

$$\text{Tempo de máquina } op_{2,ce2} = 15$$

Somando-se os tempos para o processamento das operações 1 e 2 sem agrupamento, o tempo total de máquina será igual a trinta e sete unidades de tempo. Deve-se ressaltar, neste

<sup>12</sup> A variável de tempo será medida em unidades de tempo não especificadas.

caso, que os recursos escolhidos são aqueles com os menores tempos de máquina entre as possíveis opções para o processamento das operações 1 e 2 individualmente.

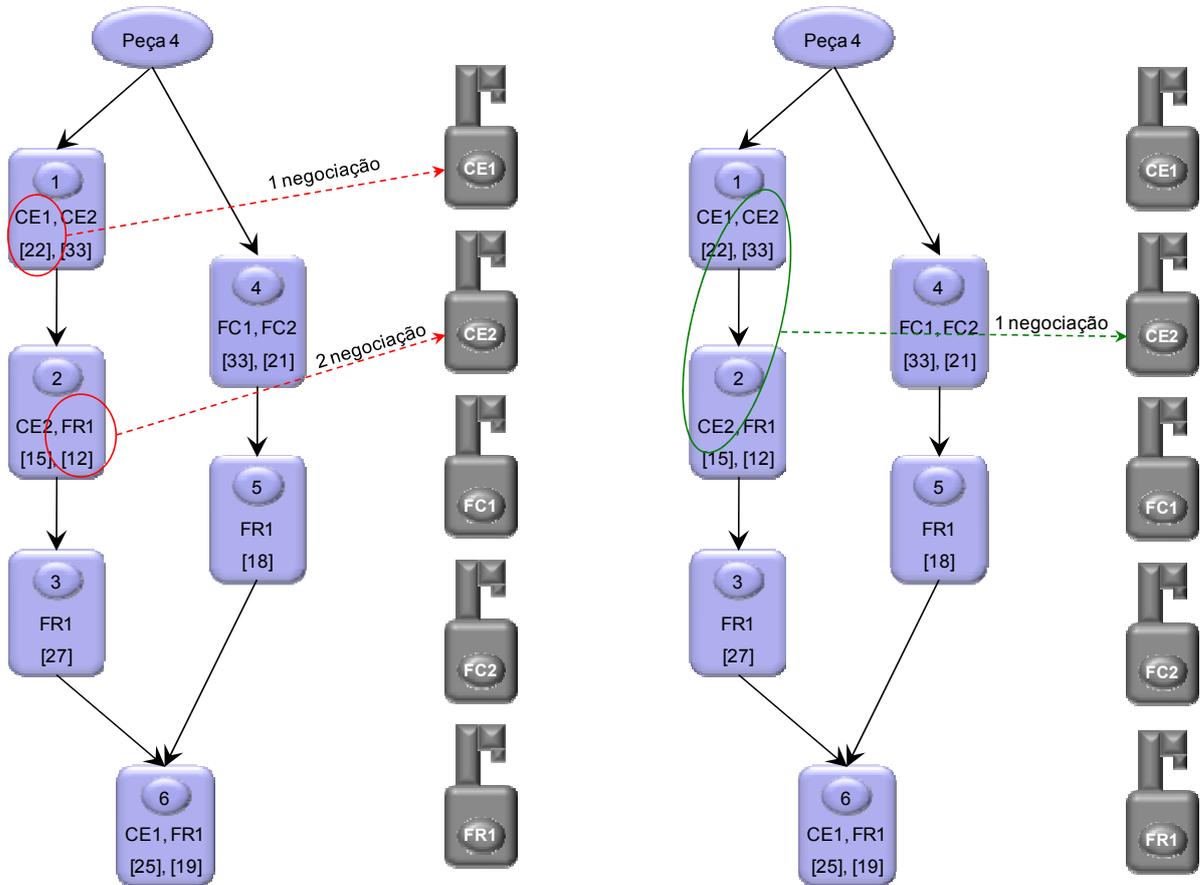


Figura 6.4 – Tempo de proposta nas abordagens sem agrupamento x com agrupamento de operações

Utilizando-se os mesmos parâmetros da negociação anterior na abordagem com agrupamento de operações, o tempo de máquina total para as operações 1 e 2 será de:

$$\text{Tempo de máquina total} = \left( \frac{\sum_{j=1}^2 (p_{1,ce2} + p_{2,ce2}) + sp_{2,ce2} + (sf_{1,ce2} + sf_{2,ce2}) * 1}{2} \right)$$

$$\text{Tempo de máquina total} = \left( \frac{(6,6 + 3,0) + 23,1 + (3,3 + 1,5) * 1}{2} \right)$$

$$\text{Tempo de máquina total} = 18,75$$

Como é possível processar as operações 1 e 2 em um mesmo recurso, ocorre o agrupamento em uma única rodada de negociação. Os tempos de processamento e *setup* de

---

fixação são somados diretamente, enquanto o tempo de *setup* de preparação utiliza somente o valor mais alto (tabela 6.1) entre as duas operações.

Com este exemplo simples observa-se que o agrupamento de operações, mesmo quando não utiliza os recursos mais velozes para o processamento das ordens, representa, usualmente, um ganho de tempo em relação ao não agrupamento das mesmas.

## 6.2 Caso de teste para avaliação do trabalho proposto

Conforme citado no item 3.13, que trata do conceito de validação e verificação, será apresentada uma comparação entre os dados obtidos pelo modelo proposto e aqueles obtidos por KIM et al. (2003) e POTTER (1997). Para que isto seja viável, todas as abordagens utilizaram o mesmo caso de teste desenvolvido por YEO et al. (2003). É necessário lembrar que este caso de teste teve que ser adaptado para atender a necessidade do modelo proposto, em particular no que diz respeito ao cálculo da flexibilidade máquina, inexistente nas demais abordagens. Outras explicações sobre a adaptação do caso de teste poderão ser encontradas no item 7.3.2 do presente trabalho.

O ambiente modelado é um sistema de manufatura por processo (*job shop*), estático, cuja caracterização foi apresentada no item 4.1.2 do presente trabalho. As métricas analisadas, já descritas no item 4.1.3, serão o tempo de fluxo de trabalho (*flow time*) e o tempo total de conclusão de um grupo de ordens (*makespan*). Também serão apresentados os tempos de fila por ordem além do desvio padrão em cada conjunto de simulações. O modelo simulado é composto por quinze recursos produtivos com capacidades variadas, capacidades estas representadas pelo número total de operações diferentes que o recurso é capaz de executar, e que podem variar entre quarenta e sete e sessenta, conforme mostrado na tabela 6.2.

O modelo também é formado por dezoito peças diferentes, sendo que cada peça é composta por um número variável de operações, entre oito e vinte e duas. Cada operação está relacionada a um ou mais recursos, com seu respectivo tempo de processo. A tabela 6.3 apresenta como exemplo de plano de processo a peça 1 do caso de testes, com seus respectivos tempos e recursos alternativos por operações.

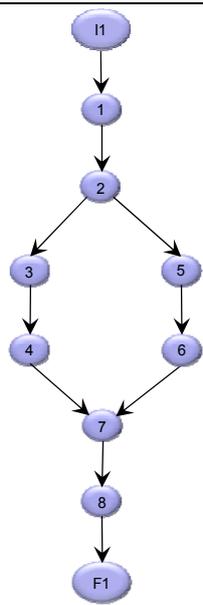
---

Tabela 6.2 – Número de operações diferentes por recurso

Recurso	Número de operações
01	56
02	56
03	54
04	59
05	60
06	60
07	52
08	53
09	52
10	56
11	47
12	56
13	58
14	49
15	48

Tabela 6.3 - Operações x Recursos para a peça 1 x Tempo de processamento

Operação	Recursos		
1	R9	R14	
	13	10	
2	R11	R15	
	24	18	
3	R15		
	43		
4	R12		
	43		
5	R13		
	30		
6	R4	R12	
	32	25	
7	R1	R5	R11
	40	49	39
8	R8		
	47		



Tempo de processamento

Um detalhamento referente à representação dos planos de processo no modelo desenvolvido pode ser encontrada no item 4.1.5 do presente trabalho.

A partir do plano de processo identifica-se o tempo mínimo para o processamento de cada uma das peças, que irá depender das operações e recursos selecionados. Para o cálculo deste valor de tempo mínimo é considerado aquele plano de processo, entre todas as alternativas possíveis para uma dada peça, que independentemente do número de operações, tem a menor somatória de tempos de processamento, ou seja, são considerados apenas os recursos mais rápidos em cada operação. A tabela 6.4 apresenta o tempo mínimo de máquina necessário para a execução de cada uma das dezoito peças representadas no modelo.

Tabela 6.4 – Tempo mínimo de processamento por peça

	Peça	Tempo Mínimo
Tempo de processamento	1	255
	2	304
	3	427
	4	306
	5	247
	6	345
	7	267
	8	318
	9	344
	10	221
	11	233
	12	372
	13	184
	14	218
	15	314
	16	194
	17	264
	18	320

Este tempo deve ser respeitado a fim de preservar a integridade do modelo. Por exemplo, um resultado de *flow time* menor que o tempo de processamento mínimo indicaria uma sobreposição (*overlapping*) ou uma divisão (*split*) de operações, o que já foi definido no item 4.1.2 como não permitido para o modelo proposto.

### 6.2.1 Representação dos níveis de flexibilidade no modelo desenvolvido

Conforme citado no item 4.1.6, que trata da definição de flexibilidade, o modelo desenvolvido permite a representação de quatro flexibilidades diferentes, sendo três destas relativas ao produto e uma relativa ao processo. Cada flexibilidade é representada por uma métrica distinta, e deve estar situada em um dos três níveis: alto (A), médio (M) ou baixo (B). Uma ampla discussão sobre os tipos de flexibilidade e as métricas escolhidas para sua representação pode ser encontrada nos itens 4.1.6 a 4.1.10 do presente trabalho. A tabela 6.5 relaciona cada um dos dezoito tipos de peça modelados, com seu número mínimo de operações, seu número total de possibilidades de operações, isto caso a peça possua a opção OU (“OR”) em seu plano de processo e seus diferentes níveis de flexibilidade.

Tabela 6.5 – Peças x níveis de flexibilidade

Peça	Número de operações	Número máximo de operações	FP	FS	FO	Níveis de flexibilidade
1	8	8	1	0,14	1,62	BBB
2	12	14	2	0,61	2,42	BMM
3	19	19	1	0,76	4,1	BAA
4	13	16	4	0,38	1,81	MBB
5	12	18	3	0,51	4,05	MMA
6	16	20	3	0,72	2,6	MAM
7	14	21	9	0,33	2,47	ABM
8	12	20	10	0,41	1,8	AMB
9	17	20	8	0,72	3,8	AAA
10	9	11	4	0,47	2,27	MMM
11	9	9	1	0,67	3,55	BMA
12	16	18	2	0,74	1,83	BAB
13	11	18	5	0,33	3,61	MBA
14	11	13	4	0,58	2,46	MMM
15	13	15	4	0,73	1,87	MAB
16	13	21	6	0,38	3,67	ABA
17	15	22	12	0,69	1,95	AMB
18	13	17	8	0,72	2,29	AAM

Tomando como exemplo a peça 10, observa-se que são necessárias no mínimo nove operações, entre as onze possíveis em seu plano de processos, a fim de finalizá-la, estes valores estão representados nas duas primeiras colunas da tabela 6.5. A peça 10, suas operações com os respectivos recursos e tempos alternativos estão representados na figura 6.4.

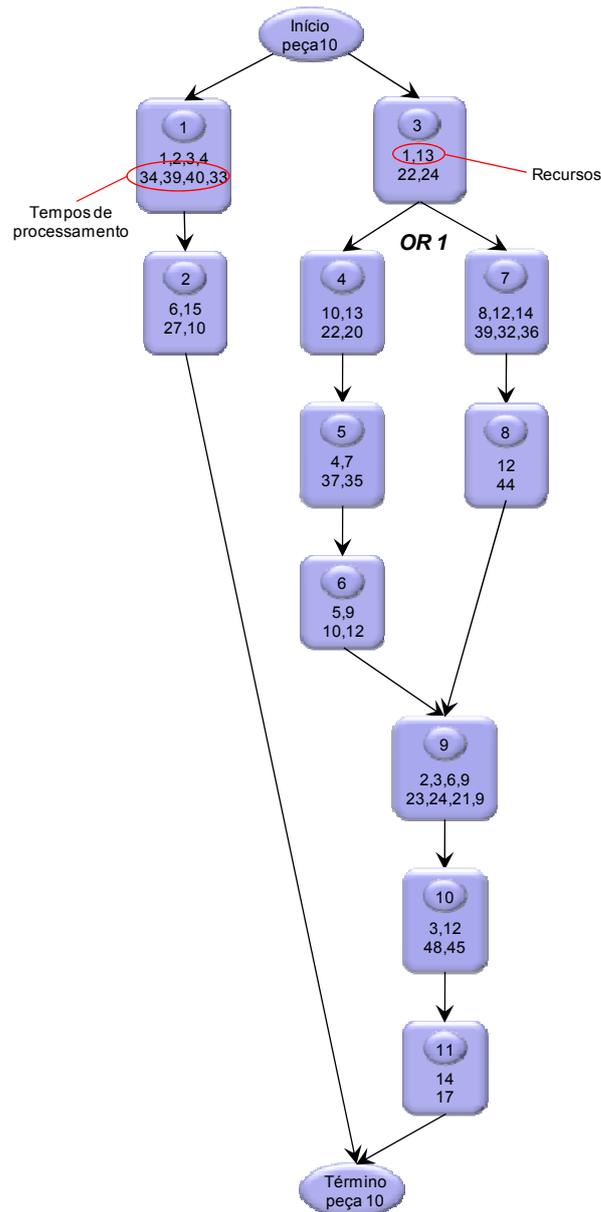


Figura 6.4 – Plano de processo com alternativas para a peça 10 do caso de testes

O cálculo da flexibilidade de processo da peça 10, conforme descrito no item 4.1.9, é dado pelo número de plano de processos alternativos existentes, que neste caso será igual a 4, sendo eles:

- a) Primeiro plano – composto pelas operações 1, 2, 3, 4, 5, 6, 9, 10 e 11;
- b) Segundo plano – composto pelas operações 1, 2, 3, 7, 8, 9, 10 e 11;
- c) Terceiro plano – composto pelas operações 3, 4, 5, 6, 9, 10, 11, 1 e 2; e
- d) Quarto plano – composto pelas operações 3, 7, 8, 9, 10, 11, 1 e 2.

A seguir é feito o cálculo relativo à flexibilidade de seqüência, que conforme definido e exemplificado no item 4.1.8 é representado pela divisão do número de seqüências possíveis pelo número total de operações.

$$\text{Flexibilidade\_de\_sequencia} = 1 - \frac{\text{soma\_máquinas\_alternativas}}{\left(\frac{n(n-1)}{2}\right)}$$

Seqüência escolhida – 3, 4, 5, 6, 9, 10, 11, 1 e 2;

Soma de máquinas alternativas nesta seqüência (exceto primeira operação) – 19;

Número de operações (n) – 9.

$$\text{Flexibilidade\_de\_sequencia} = 1 - \frac{19}{\left(\frac{9(9-1)}{2}\right)} = 0,47$$

Finalmente é feito o cálculo da flexibilidade de operações da peça 10, que conforme definido no item 4.1.7 é dado pela relação entre a média do número de máquinas alternativas disponíveis em cada uma das operações e o número total de operações que irá compor o plano de processo.

$$\text{Flexibilidade\_de\_operações} = \frac{\sum(\text{n\_de\_máquinas\_alternativas\_por\_operação})}{\text{n\_total\_de\_operações\_do\_plano\_de\_processo}}$$

$$\text{Flexibilidade\_de\_operações} = \frac{\sum(25)}{11} = 2,27$$

Ao se comparar estes valores às métricas estabelecidas anteriormente no capítulo 5, é possível enquadrar a peça 10 nos níveis de flexibilidade de processo média (M), flexibilidade de seqüência média (M e flexibilidade de operações média (M), o que resulta no valor MMM representado na última coluna.

Após enquadrar cada uma das peças representadas no modelo nos seus respectivos níveis de flexibilidade de produto, as mesmas são agrupadas em vinte e quatro diferentes grupos de simulação. Cada grupo privilegia um nível de flexibilidade dominante entre as peças. Por exemplo, na tabela 6.5 podem ser identificadas todas as peças que possuem um nível de flexibilidade de seqüência (FS) baixo (B) (peças 1, 4, 7, 10,13 e 16). Este processo se

repete até que todos os níveis de flexibilidade estejam agrupados, os quais são apresentados na tabela 6.6.

Observa-se que somente os grupos 1 a 21 são caracterizados por um nível de flexibilidade de produto predominante. Os grupos 22, 23 e 24, por sua vez, não possuem nenhuma característica de destaque. Os mesmos foram criados a fim de testar o comportamento do modelo e dos protocolos de negociação, a partir do aumento do número de peças simuladas de forma simultânea.

Tabela 6.6 – Agrupamento de peças por nível de flexibilidade dominante

Grupo	Peças	Nível de Flexibilidade Predominante
01	1, 2, 3, 10, 11, 12	Baixo FP
02	4, 5, 6, 13, 14, 15	Médio FP
03	7, 8, 9, 16, 17, 18	Alto FP
04	1, 4, 7, 10, 13, 16	Baixo FS
05	2, 5, 8, 11, 14, 17	Médio FS
06	3, 6, 9, 12, 15, 18	Alto FS
07	1, 4, 8, 12, 15, 17	Baixo FO
08	2, 6, 7, 10, 14, 18	Médio FO
09	3, 5, 9, 11, 13, 16	Alto FO
10	1, 2, 3, 5, 6, 10, 11, 12, 15	Baixo ou Médio FP
11	4, 7, 8, 9, 13, 14, 16, 17, 18	Médio ou Alto FP
12	1, 4, 5, 7, 8, 10, 13, 14, 16	Baixo ou Médio FS
13	2, 3, 6, 9, 11, 12, 15, 17, 18	Médio ou Alto FS
14	1, 2, 4, 7, 8, 12, 15, 17, 18	Baixo ou Médio FO
15	3, 5, 6, 9, 10, 11, 13, 14, 16	Médio ou Alto FO
16	1, 2, 3, 4, 5, 6, 10, 11, 12, 13, 14, 15	Baixo ou Médio PF
17	4, 5, 6, 7, 8, 9, 13, 14, 15, 16, 17, 18	Médio ou Alto PF
18	1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17	Baixo ou Médio FS
19	2, 3, 5, 6, 8, 9, 11, 12, 14, 15, 17, 18	Médio ou Alto FS
20	1, 2, 4, 6, 7, 8, 10, 12, 14, 15, 17, 18	Baixo ou Médio FO
21	2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 16, 18	Médio ou Alto FO
22	2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18	-----
23	1, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18	-----
24	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18	-----

Além do agrupamento das peças por similaridade no nível de flexibilidade predominante, as mesmas foram agrupadas em ordem crescente de número de peças por rodada de simulação, ou seja, os grupos de 01 a 09 são compostos por seis peças; de 10 a 15 por nove peças; de 16 a 21 por doze peças; 22 e 23 por quinze peças e 24 por dezoito peças.

### 6.2.2 Representação dos níveis de flexibilidade de máquina

O modelo aqui proposto e desenvolvido diverge daqueles sugeridos por KIM et al. (2003) e POTTER (1997), não somente pelas ferramentas escolhidas para a resolução do problema, mas também pela utilização e análise de uma flexibilidade adicional, relacionada ao processo, chamada de flexibilidade de máquina. Esta flexibilidade bem como o cálculo da métrica escolhida para sua representação, já foi descrita no item 4.1.10 do presente trabalho. A tabela 6.7 apresenta os níveis de flexibilidade de máquina para cada um dos quinze recursos modelados.

Tabela 6.7 – Flexibilidade de máquina por recurso

Recurso	Número de operações	Número total de operações agrupadas	Flexibilidade de máquina
1	56	12	0,21
2	56	10	0,18
3	54	15	0,28
4	59	16	0,27
5	60	17	0,28
6	60	20	0,33
7	52	18	0,35
8	53	12	0,23
9	52	9	0,17
10	56	18	0,32
11	47	8	0,17
12	56	8	0,14
13	58	26	0,45
14	49	16	0,33
15	48	11	0,23

O valor da flexibilidade de máquina corresponde ao quociente do número total de operações passíveis de agrupamento, por recurso pelo número total de operações realizáveis por recurso, considerando-se as dezoito peças que compõem o modelo.

### 6.3 Conclusão do capítulo

Este capítulo tratou da formulação da proposta de negociação, usada como critério de desempate e de agendamento entre as diversas ordens que estejam concorrendo concomitantemente aos recursos disponíveis durante uma simulação.

---

A importância de uma negociação reside no fato de que suas regras irão definir a estratégia que um agente irá utilizar a fim de alcançar seus objetivos, tanto individuais quanto globais.

Foi visto que a abordagem de negociação utilizada neste modelo é uma versão estendida daquela proposta por USHER (2003), à qual é acrescentado um mecanismo de agrupamento de operações. A necessidade de tal mecanismo foi observada, pois na abordagem sugerida por USHER (2003) cada ordem pode negociar apenas uma operação por vez, o que inviabiliza o agrupamento de operações e como consequência o estudo da flexibilidade de máquina, que faz parte do escopo dos objetivos do presente trabalho.

Os tempos que compõem uma negociação também foram tratados neste capítulo, estes tempos se dividem em tempos de máquina, que são os tempos de processamento e *setups*, e em tempos internos ao sistema, como o tempo de troca de mensagens entre agentes, por exemplo.

Finalmente foi apresentado o caso de teste desenvolvido por YEO et al. (2003), o qual foi utilizado para a validação do modelo. Um caso de testes é necessário, pois permitirá a comparação entre os dados obtidos pelo modelo proposto e aqueles obtidos por KIM et al. (2003) e POTTER (1997). Lembrando, porém, que este caso de teste teve que ser adaptado para atender a necessidade do modelo proposto, em particular no que diz respeito ao cálculo da flexibilidade máquina, inexistente nas demais abordagens.

---

---

## CAPÍTULO 7 – PARÂMETROS DE SIMULAÇÃO E ANÁLISE DE RESULTADOS

Este capítulo trata da análise dos resultados obtidos através do protótipo de simulação desenvolvido em laboratório para o presente trabalho.

Inicialmente são apresentados os parâmetros de simulação utilizados, tanto para a validação do modelo proposto, quanto para as demais condições estudadas. A seguir é feita uma breve discussão a respeito dos trabalhos de KIM et al. (2003) e POTTER (1997), utilizados para a validação e comparação de resultados obtidos neste trabalho.

Finalmente são apresentados e discutidos os valores das variáveis de resposta *makespan* e *flow time* obtidos através do modelo proposto, sendo então comparados àqueles obtidos por KIM et al. (2003) e POTTER (1997), valores estes que foram obtidos a partir do caso de testes sugerido por YEO et al. (2003).

### 7.1 Parâmetros de simulação

Dentro do contexto descrito até o momento neste trabalho e a fim de obter as variáveis de resposta *flow time* e *makespan*, foram modelados quatro diferentes fatores, na forma de tipos de flexibilidades, os quais foram: as flexibilidades de processo, seqüência, operações e máquinas, conforme apresentado na figura 7.1, sendo que cada um destes fatores pode assumir três diferentes níveis, alto (A), médio (M) ou baixo (B). Uma descrição detalhada dos tipos de flexibilidades e suas métricas foram apresentadas no capítulo 5 do presente trabalho.

As simulações foram divididas em quatro diferentes grupos, sendo que cada grupo considerou entre dois e quatro diferentes tipos de flexibilidade:

- a) Grupo A – flexibilidades de operação e seqüência;
- b) Grupo B – flexibilidades de operação, seqüência e máquina;
- c) Grupo C – flexibilidades de operação, seqüência e processo; e
- d) Grupo D – flexibilidades de operação, seqüência, processo e máquina.

Os resultados desta simulação serão apresentados considerando a média obtida das vinte replicações para cada um dos grupos de peças e a média das médias para os vinte e quatro grupos. O número de replicações foi determinado a partir dos experimentos realizados por KIM et al. (2003) e POTTER (1997) e considerado adequado ao projeto de experimentos em questão.

---

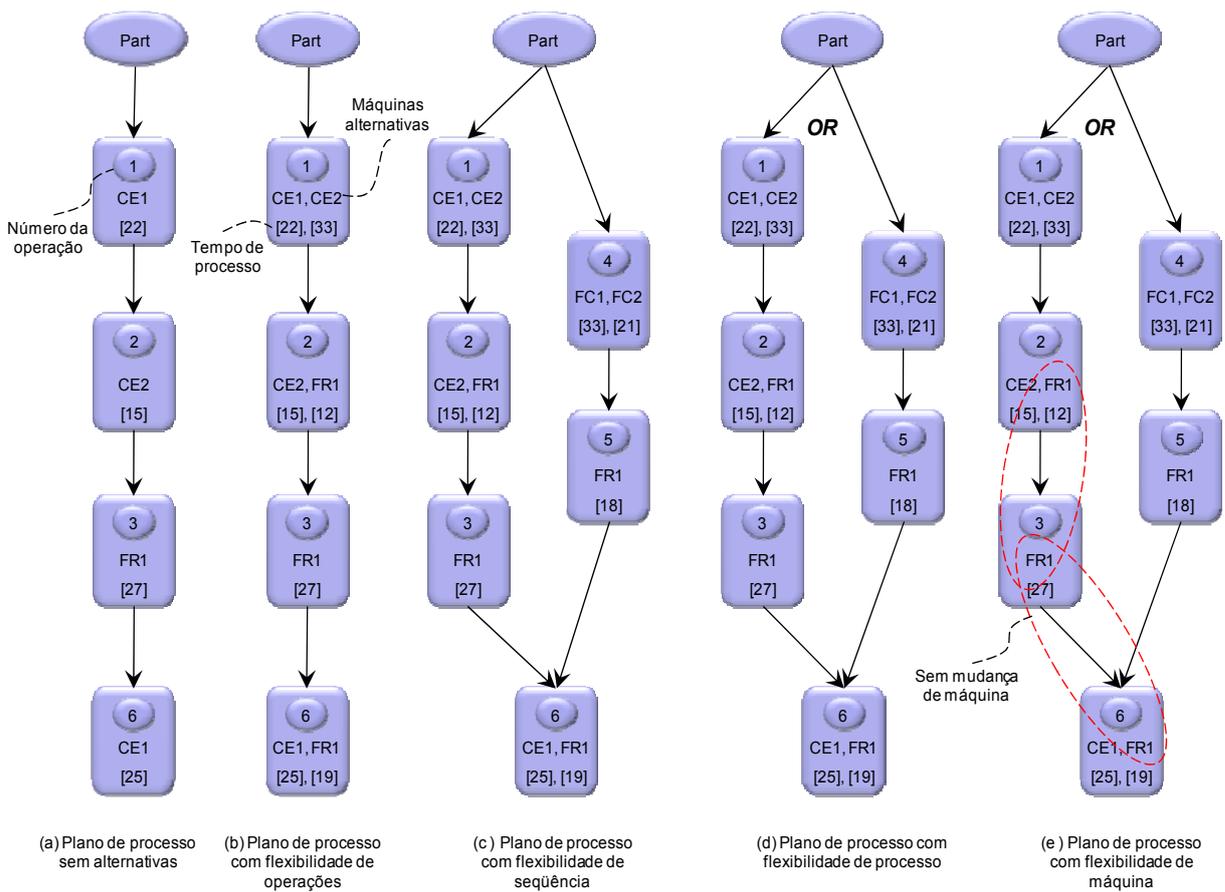


Figura 7.1 – Diferentes tipos de flexibilidades permitidas pelo modelo proposto

É importante ressaltar que a diferença entre as de replicações de um mesmo grupo de simulação, é devida a aleatoriedade inerente a velocidade de troca de mensagens entre os agentes *resource* e *jobOrder*, a qual irá resultar em diferentes agendamentos entre as ordens e os recursos disponíveis no chão-de-fábrica.

Ao se observar a figura 7.2 e comparar as replicações 1 e 2 do grupo de simulação 24, com *setups* de preparação e fixação iguais a zero, é possível notar os diferentes agendamentos para um mesmo dado recurso. Por exemplo, no lado esquerdo da figura, correspondente a primeira replicação, o recurso 8 (R8) foi agendado para inicialmente processar a operação 14 da peça 5 (5 op14), enquanto na replicação 2, lado direito da figura, o mesmo recurso foi agendado para processar a operação 1 da peça 7 (7 op1).

Com isto fica estabelecida a aleatoriedade exigida, a qual justifica um experimento de simulação com um número *n* de replicações.

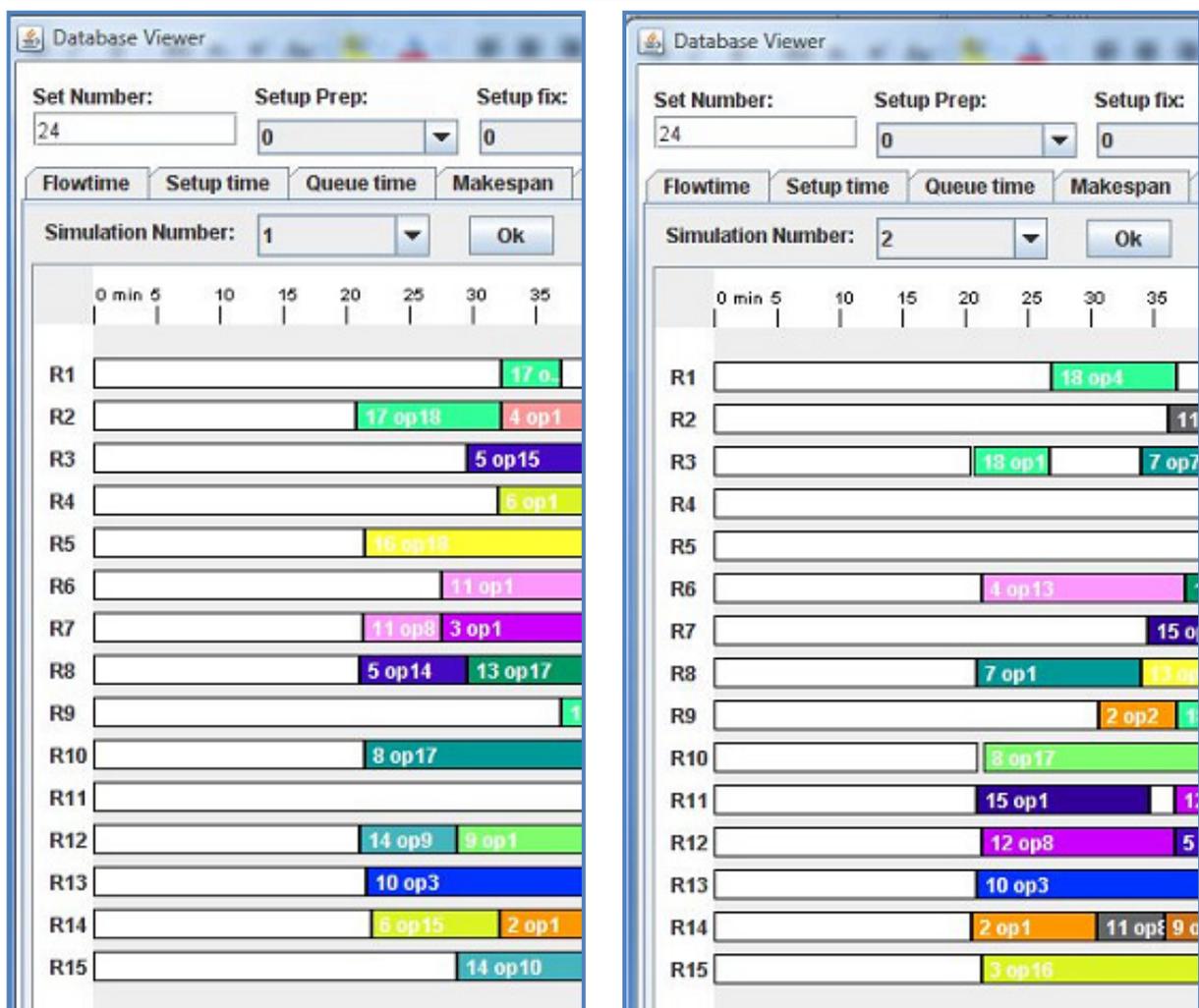


Figura 7.2 – Diferentes agendamentos para um mesmo grupo de simulação

Para cada um dos grupos acima mencionados foram simulados 24 grupos de peças apresentados na tabela 6.7, sendo que cada experimento foi replicado vinte vezes. Os resultados desta simulação serão apresentados considerando a média obtida das vinte replicações para cada um dos grupos de peças e a média das médias para os vinte e quatro grupos. O número de replicações foi determinado a partir dos experimentos realizados por KIM et al. (2003) e POTTER (1997), e esta quantidade de replicações foi considerada adequada ao projeto de experimentos em questão.

Os parâmetros de configuração do modelo são estabelecidos antes do início de cada uma das simulações ou grupo de simulações. Isto é realizado através do arquivo *Config.java* a ser apresentado na figura 7.3.

A configuração do modelo inicia com a escolha dos grupos de peças que irão participar da simulação. Esta escolha é feita entre os vinte e quatro possíveis grupos existentes, conforme apresentado na tabela 6.7 do presente trabalho.

---

```

public class Config {

    //Escolha os grupos de peças que irão participar da simulação (1 até
    24)
    //Escolha as peças que irão participar da simulação (opcional)
    //Escolha o tamanho do lote
    public static final int SET = 1,2,5,12,13,20;
    public static final int[] INITIAL_PARTS_ID = Sets.SETS[SET];
    public static final int BATCHSIZE = 1;

    //Escolha os recursos que irão participar da simulação (opcional)
    //Atencao: a opção false indica que todos os recursos irão participar
    da simulação
    public static final boolean SELECTABLE_RESOURCES = false;
    public static final String RESOURCES_NAMES = "R9";

    //Configuração das flexibilidades de processo e de máquina
    public static final boolean SIMULATION_WITH_OR = false;
    public static final boolean GROUPING_BY_OPERATIONS = false;

    // Defina o tempo de setup de preparação e fixação (valores %)
    //public static final double[] SETUP_PREP = {0, 0.1, 0.3, 0.5, 0.7};
    //public static final double[] SETUP_FIX = {0, 0.1};

    public static final double[] SETUP_PREP = { 0 };
    public static final double[] SETUP_FIX = { 0 };

    // Escolha o número de replicações
    public static final int SIMULATION_PARAMETER = 20;

    // Escolha a constante de tempo
    public static final int TIMECONSTANT = 500;

    // Configuração do banco de dados
    public static final String DATABASEHOST = "localhost";
    public static final String DATABASELOGIN = "isfa";
    public static final String DATABASEPASS = "mysql2006";

```

Figura 7.3 – Classe que permite a configuração dos parâmetros de simulação do modelo proposto

Caso o usuário opte por não utilizar os grupos de simulação já configurados, é possível a composição de novos grupos através da escolha das peças, esta escolha deverá ser feita entre as dezoito peças já modeladas. Na seqüência é escolhido o tamanho de lote a ser utilizado, e este valor será igual para todos os grupos inseridos em uma dada simulação.

O modelo permite ainda a opção de escolha dos recursos que irão participar da simulação. Como valor padrão (*default*), todos os quinze recursos apresentados na tabela 6.2 são considerados como disponíveis (*SELECTABLE\_RESOURCES = false*).

Dos quatro tipos de flexibilidade permitidos no modelo e já discutidos no capítulo 4, somente as flexibilidades de processo e de máquina são configuráveis através da classe *Config.java*, sendo que a flexibilidade de operações e seqüência são parte da estrutura do banco de dados.

---

A flexibilidade de processo é controlada através da opção *SIMULATION\_WITH\_OR*, que indica o uso ou não de planos de processo alternativos, conforme visto no item 5.1.9 do presente trabalho. Por valor padrão esta simulação não está habilitada (*false*).

O segundo tipo, flexibilidade de máquina, é configurado em duas etapas: primeiramente através da opção de agrupamento de operações (*GROUPING BY OPERATIONS*), que indica o uso ou não da opção de agrupamento de operações, já descrita no item 6.1.2 do presente trabalho. Por padrão esta flexibilidade também não está habilitada (*false*).

Na segunda etapa de configuração da flexibilidade de máquina é escolhida a combinação entre os tempos de *setup* de preparação e fixação a ser utilizada. Esta escolha é baseada nas opções apresentadas na tabela 7.1 e serão discutidas posteriormente no item 7.5 do presente capítulo. Por valor padrão os tempos de *setup* de preparação e fixação são iguais a zero.

Grupos de <i>Setup</i>	<i>Setup</i> de Preparação ( $sp_{i,m}$ )	<i>Setup</i> de fixação ( $sf_{i,m}$ )
1	0,10	0,10
2	0,30	0,10
3	0,50	0,10
4	0,70	0,10

Tabela 7.1 – Diferente níveis de *setup* de preparação e fixação utilizados no modelo

Finalmente, na última etapa de configuração do modelo, são escolhidos o número de replicações a serem feitas, a constante de tempo a ser utilizada, bem como a localização (endereço) do banco de dados escolhido, ressaltando que este banco de dados pode estar executando localmente ou em um ambiente distribuído.

É relevante realizar algumas considerações sobre a adoção da constante de tempo. A mesma foi utilizada como denominador nos tempos de máquina a fim de minimizar os tempos totais de simulação. Seu uso se mostrou necessário devido ao elevado número de replicações necessárias para a validação do modelo e o tempo total de processamento para executá-las.

A escolha desta constante foi baseada em testes que demonstraram que o valor de 500<sup>13</sup> é o máximo permitido sem causar interferência nos resultados obtidos. Esta interferência está relacionada à duração do período de negociação, já descrito no item 6.1.1 do presente trabalho.

Constatou-se durante a fase de testes, que uma constante de tempo elevada faz com que o intervalo de negociação fique reduzido em tal proporção que compromete o recebimento das

<sup>13</sup>A constante de tempo é dada em unidades de tempo não especificadas, de maneira análoga aos tempos de processamento.

---

propostas de execução por parte dos agentes de ordem. Ou seja, os agentes de recurso não conseguem elaborar uma proposta e retorná-la em tempo hábil aos agentes de ordem, dificultando, desta forma, a análise e a comparação das mesmas, o que conseqüentemente irá resultar em uma má escolha entre as propostas recebidas.

Para quantificar esta interferência foram comparados os números de propostas recebidas pelos agentes de ordem após o término do período de negociação para diferentes constantes de tempo, cujos valores se encontravam entre 100 e 1000, optando-se, então, pelo valor de 500.

## 7.2 Abordagens utilizadas para validação do modelo

O principal objetivo deste sub-item é possibilitar através de uma visão geral, a compreensão das abordagens utilizadas por KIM et al. (2003) e POTTER (1997) na solução do caso de testes sugerido por YEO et al. (2003). Para tanto é importante observar que não faz parte do escopo deste trabalho uma apresentação e discussão profunda sobre o uso de algoritmos genéticos e suas variantes. Uma ampla revisão sobre o assunto pode ser encontrada em REZENDE (2005) e LUGER (2004). Já os detalhes das implementações de algoritmos genéticos co-evolucionários cooperativos podem ser encontrados nos próprios trabalhos de KIM et al. (2003) e POTTER (1997). Finalmente, para uma discussão a respeito da adequabilidade de seu uso no escopo na programação de ambientes *job shop* através da subdivisão do espaço do problema, sugere-se a leitura de MORIARTY e MIIKKULAINEN (1997).

Tanto a abordagem de KIM et al. (2003) quanto a de POTTER (1997) fazem uso de algoritmos genéticos co-evolucionários. Algoritmos genéticos, de acordo com REZENDE (2005), são algoritmos de otimização global, baseados nos mecanismos de seleção natural e da genética, que empregam uma estratégia de busca paralela e estruturada, embora aleatória, direcionada à busca de pontos de “alta aptidão”, ou seja, pontos nos quais a função a ser minimizada ou maximizada tem valores relativamente altos ou baixos.

Ao contrário dos algoritmos genéticos tradicionais, em que indivíduos de uma espécie evoluem enquanto interage somente com indivíduos da mesma espécie, nos algoritmos co-evolucionários uma espécie influencia e é influenciada por outras espécies. Ou seja, enquanto uma das espécies estiver evoluindo, estará interagindo e se adaptando a outras espécies, imitando a co-evolução biológica através de uma série de mudanças recíprocas em duas ou mais espécies.

---

---

De acordo com KIM et al. (2003), algoritmos co-evolucionários são tipicamente classificados como cooperativos ou competitivos. Cooperativos são aqueles que operam através da imitação de uma relação de simbiose, originando daí o termo “algoritmo evolucionário simbiótico”. Competitivos que são aqueles que operam através da imitação de uma relação de parasitismo.

Conforme HOLLAND (1992), algoritmos genéticos têm se mostrado eficientes para a busca de soluções ótimas ou aproximadamente ótimas em uma grande variedade de problemas.

O uso de algoritmos genéticos simbióticos e evolucionários para a integração do planejamento do processo e programação da produção baseia-se na hipótese de que buscas paralelas em diferentes partes de um espaço de solução são mais eficientes que uma busca simples para toda a solução (MORIARTY and MIIKKULAINEN, 1997). Desta forma, um problema pode ser visto como a combinação de diversos subproblemas, cuja inter-relação é demasiadamente complexa.

A primeira abordagem utilizada para comparação da solução da programação de ordens de produção em um ambiente funcional foi desenvolvida por POTTER (1997), e foi chamada de CCGA – *Cooperative Coevolutionary Genetic Algorithm*. Já a segunda abordagem foi proposta por KIM et al. (2003), e foi chamada de algoritmo evolucionário simbiótico (SEA – *Symbiotic Evolutionary Algorithm*).

Desta forma, tanto CCGA quanto SEA baseiam-se na idéia que um sistema de buscas paralelas para partes diferentes de solução é mais eficiente que uma única busca para a solução inteira. Deste modo, o problema é decomposto em vários subproblemas, que são resolvidos pelo algoritmo de procura multilateral modificado.

A principal diferença entre as duas abordagens reside no fato de que no trabalho de POTTER (1997) a estratégia de evolução é baseada na população (*population-based*), ou seja, um indivíduo que apresente a aptidão mais alta em cada uma das outras populações se torna um sócio simbiótico. Já na abordagem proposta por KIM et al.(2003), esta evolução é baseada na vizinhança (*neighborhood-based*), o que significa que o sócio simbiótico é selecionado randomicamente em um bairro de cada das outras populações.

Além disto, a fim de alcançar a redução de *makespan* e *flow time*, KIM et al. (2003) adotaram estratégias de interações localizadas, reprodução em regime permanente (*steady-state*) e seleção simbiótica randômica de parceiros.

Apesar de utilizarem o mesmo caso de testes e serem os padrões para validação, comparação dos resultados e verificação da adequabilidade da abordagem sugerida nesta

---

---

pesquisa, os trabalhos de KIM et al. (2003) e POTTER (1997) diferenciam-se do aqui proposto nos seguintes itens:

- a) Na escolha da ferramenta utilizada para a construção do modelo, neste caso algoritmo genético ao invés de multiagentes;
- b) Nas flexibilidades permitidas nos planos de processo das peças, ou seja, somente as flexibilidades de operações, seqüência e processo, desconsiderando o uso de flexibilidade de máquina; e
- c) Na análise dos dados obtidos, observando somente os resultados das variáveis de resposta *makespan* e *flow time*, não havendo nenhum tipo de análise da influência ou não, de uma flexibilidade sobre as demais.

### 7.3 Análise dos resultados – *makespan*

Antes de iniciar a apresentação e análise dos resultados, é importante observar que o critério de comparação entre os resultados obtidos neste trabalho e aqueles obtidos nos trabalhos de KIM et al. (2003) e POTTER (1997), que é a comparação entre a média das médias para cada uma das condições replicadas, não é considerado o mais adequado ou o mais confiável, porém foi adotado visando replicar as condições oferecidas pelos citados trabalhos, permitindo desta forma uma comparação direta entre os resultados.

Apesar disto, nas tabelas a seguir, sempre que houver uma comparação direta com os valores obtidos por KIM et al. (2003) e POTTER (1997) e a abordagem proposta, ou seja nas tabelas 7.4 a 7.11 para análise de *makespan*, e 7.8 a 7.11 para análise de *flow time*, será possível encontrar os valores, mínimo e máximo, do intervalo de confiança de 95% baseado na média obtida pela abordagem sugerida. Também é importante observar que as médias obtidas por KIM et al. (2003) e POTTER (1997) se encontram dentro do intervalo de confiança calculado, o que a principio indicaria que não há uma variação significativa entre a média obtida pela abordagem sugerida e as demais abordagens.

#### 7.3.1 *Makespan* – flexibilidades de operação, seqüência, processos e máquina

Nesta primeira fase do projeto de experimentos são executadas pelo modelo proposto as condições exatas apresentadas pelas abordagens de KIM et al. (2003) e POTTER (1997) no que diz respeito às condições de flexibilidade.

---

---

Neste caso, são permitidas apenas as flexibilidades de operação, seqüência e processo, representados no grupo C, não havendo, portanto o agrupamento de operações, o que caracterizaria o uso da flexibilidade de máquina. A tabela 7.2 apresenta os resultados desta comparação para a variável de resposta *makespan*.

Primeiramente são analisados os resultados obtidos de forma global, através da média das médias da variável *makespan*. É possível observar que esta média (426,61) é menor que aquelas obtidas por de KIM et al. (2003) e POTTER (1997). Este resultado indica que os ganhos nos grupos de peças 10, 14, 17, 19, 20, 22, 23 e 24 foram significativos, em especial nos grupos formados por mais de 12 peças.

Ao analisar os resultados caso a caso, observa-se que as médias obtidas são em geral inferiores àquelas encontradas por POTTER (1997) e superiores às de KIM et al. (2003). Além disto, foi observada a influência de alguns fatores distintos:

- 1) Número de peças: existe uma discreta tendência de melhores resultados para a variável de resposta *makespan* em grupos formados por um maior número de peças, situado entre 12 e 18. Observa-se, ainda, que estes grupos apresentam os maiores ganhos percentuais, com valores acima de 4%, podendo chegar a valores superiores a 12% em alguns casos; e
- 2) Flexibilidades predominantes: observou-se que nos grupos em que a característica predominante é a flexibilidade de operações, com seu nível variando entre médio e alto, ou flexibilidade de seqüência, com seu nível variando entre baixo e médio, independente do número de peças que o compõe, apresentam os piores resultados para a variável *makespan*.

Em relação ao desvio-padrão, a abordagem sugerida apresenta um aumento nos valores médios bem como na média das médias, o que indica uma maior variabilidade em relação às abordagens de KIM et al. (2003) e POTTER (1997).

A seguir são comparados os valores obtidos por KIM et al. (2003) e POTTER (1997) com aqueles obtidos neste trabalho através da inclusão do mecanismo de agrupamento de operações, representados pelo grupo D. Com isto é introduzida a flexibilidade de máquina aos experimentos. Uma discussão a respeito do funcionamento do mecanismo de agrupamento de operações pode ser encontrada no item 6.1.2 do presente trabalho.

Uma observação importante deve ser feita neste ponto do trabalho a fim de facilitar a sua compreensão. Nesta comparação, os tempos de *setup* de preparação e fixação são considerados independentes e computados diretamente na parcela de tempo de processamento, ou seja, são considerados iguais a zero.

---

Tabela 7.2 – Comparação dos resultados obtidos pela abordagem proposta para a variável *makespan* para a mesma condição de flexibilidade de KIM et al. (2003) e POTTER (1997)

Grupos	Makespan - flexibilidades de operação, seqüência e processo									
	Grupo C				Potter (1997)		Kim et al. (2003)		Taxa de melhoria (%)	Nível de flexibilidade predominante
	média	dv	mínimo	máximo	média	dv	média	dv		
1	445,37	7,93	375,86	514,88	470,40	8,60	437,60	10,90	-1,74	Baixo FP
2	380,25	11,43	280,06	480,44	369,20	5,60	349,70	5,90	-8,03	Médio FP
3	373,13	11,75	270,14	476,12	382,00	9,10	355,20	7,40	-4,81	Alto FP
4	327,45	9,23	246,55	408,35	321,50	7,60	306,20	0,40	-6,49	Baixo FS
5	345,47	11,32	246,25	444,69	337,80	6,10	323,70	3,60	-6,30	Médio FS
6	445,35	8,05	374,79	515,91	485,60	6,10	443,80	5,00	-0,35	Alto FS
7	377,62	2,90	352,20	403,04	385,60	5,10	372,40	1,30	-1,38	Baixo FO
8	375,11	18,21	215,49	534,73	373,80	7,00	348,30	5,70	-7,15	Médio FO
9	446,35	7,25	382,80	509,90	474,50	8,90	434,90	9,80	-2,57	Alto FO
10	450,97	8,57	375,85	526,09	502,60	12,10	456,50	10,80	1,23	Baixo ou Médio FP
11	389,23	9,76	303,68	474,78	423,80	10,50	378,90	5,10	-2,65	Médio ou Alto FP
12	357,05	13,41	239,51	474,59	379,50	15,20	332,80	3,40	-6,79	Baixo ou Médio FS
13	470,93	15,89	331,65	610,21	511,10	9,30	469,00	10,70	-0,41	Médio ou Alto FS
14	396,83	12,28	289,19	504,47	433,40	12,10	402,40	10,60	1,40	Baixo ou Médio FO
15	459,35	9,95	372,13	546,57	493,90	11,00	445,20	11,00	-3,08	Médio ou Alto FO
16	483,61	15,60	346,87	620,35	549,70	18,20	478,80	12,00	-0,99	Baixo ou Médio FP
17	429,57	20,38	250,93	608,21	496,90	16,00	448,90	8,70	4,50	Médio ou Alto FP
18	403,21	22,18	208,79	597,63	419,80	17,10	389,60	7,50	-3,38	Baixo ou Médio FS
19	480,18	14,46	353,43	606,93	557,00	12,20	508,10	10,00	5,81	Médio ou Alto FS
20	434,46	14,99	303,07	565,85	482,70	15,60	453,80	5,20	4,45	Baixo ou Médio FO
21	485,39	19,38	315,52	655,26	534,00	18,10	483,20	6,80	-0,45	Médio ou Alto FO
22	494,74	19,05	327,76	661,72	587,50	12,80	548,30	6,90	10,83	-----
23	452,96	11,66	350,76	555,16	557,90	13,20	507,50	8,30	12,04	-----
24	534,14	16,57	388,90	679,38	633,30	19,30	602,20	7,10	12,74	-----
<b>Média</b>	<b>426,61</b>	<b>13,01</b>			<b>465,10</b>	<b>11,50</b>	<b>428,20</b>	<b>7,30</b>	<b>0,37</b>	

Na tabela 7.3 pode-se observar os resultados obtidos para a variável de resposta *makespan* ao compararmos as abordagens de KIM et al. (2003) e POTTER (1997), sem flexibilidade de máquina, e a abordagem com agrupamento de operações, a qual permite o uso da flexibilidade de máquina.

Na avaliação global observa-se que, ao inserir-se a flexibilidade de máquina, a média das médias dos valores de *makespan* (451,79) passa a ser superior àquela obtida por de KIM et al. (2003), porém inferior à média encontrada no trabalho de POTTER (1997).

Este resultado foi considerado adequado, tendo em vista que o agrupamento de operações foi utilizado em conjunto com tempos de *setup* independentes. Ou seja, quando as parcelas de tempo de *setup* de preparação e fixação estão inseridas na parcela de tempo de

processamento, elas são consideradas pelo modelo como iguais a zero. Neste caso específico o modelo proposto não irá priorizar o menor tempo total de conclusão, mas sim a menor quantidade de troca de recursos, que neste caso poderão ou não ser aqueles com o maior tempo de processamento.

Ao analisar os resultados caso a caso, observa-se que as médias obtidas são, em sua maioria, superiores àquelas encontradas por KIM et al. (2003) e POTTER (1997). Além disto, novamente foi observada a influência de alguns fatores distintos:

Tabela 7.3 – Comparação dos resultados de *makespan* com a inclusão da flexibilidade de máquina com os valores obtidos por KIM et al. (2003) e POTTER (1997)

Grupos	<i>Makespan</i> - flexibilidades de operação, seqüência, processo e máquina									
	Grupo D				Potter (1997)		Kim et al. (2003)		Taxa de melhoria (%)	Nível de flexibilidade predominante
	intervalo confiança 95%									
	média	dv	mínimo	máximo	média	dv	média	dv		
1	517,73	8,96	439,19	596,27	470,40	8,60	437,60	10,90	-15,48	Baixo FP
2	369,62	7,74	301,78	437,46	369,20	5,60	349,70	5,90	-5,39	Médio FP
3	366,72	7,35	302,29	431,15	382,00	9,10	355,20	7,40	-3,14	Alto FP
4	324,02	11,17	226,11	421,93	321,50	7,60	306,20	0,40	-5,50	Baixo FS
5	364,06	38,16	29,57	698,55	337,80	6,10	323,70	3,60	-11,09	Médio FS
6	504,01	7,06	442,13	565,89	485,60	6,10	443,80	5,00	-11,95	Alto FS
7	381,64	1,37	369,63	393,65	385,60	5,10	372,40	1,30	-2,42	Baixo FO
8	373,14	12,20	266,20	480,08	373,80	7,00	348,30	5,70	-6,66	Médio FO
9	520,26	10,54	427,87	612,65	474,50	8,90	434,90	9,80	-16,41	Alto FO
10	520,76	23,79	312,23	729,29	502,60	12,10	456,50	10,80	-12,34	Baixo ou Médio FP
11	379,41	9,74	294,04	464,78	423,80	10,50	378,90	5,10	-0,13	Médio ou Alto FP
12	367,90	18,91	202,15	533,65	379,50	15,20	332,80	3,40	-9,54	Baixo ou Médio FS
13	513,26	11,25	414,65	611,87	511,10	9,30	469,00	10,70	-8,62	Médio ou Alto FS
14	394,67	13,50	276,34	513,00	433,40	12,10	402,40	10,60	1,96	Baixo ou Médio FO
15	522,39	18,79	357,69	687,09	493,90	11,00	445,20	11,00	-14,78	Médio ou Alto FO
16	533,18	26,98	296,69	769,67	549,70	18,20	478,80	12,00	-10,20	Baixo ou Médio FP
17	432,11	21,69	241,99	622,23	496,90	16,00	448,90	8,70	3,89	Médio ou Alto FP
18	401,65	23,45	196,10	607,20	419,80	17,10	389,60	7,50	-3,00	Baixo ou Médio FS
19	521,02	23,12	318,36	723,68	557,00	12,20	508,10	10,00	-2,48	Médio ou Alto FS
20	432,56	16,23	290,30	574,82	482,70	15,60	453,80	5,20	4,91	Baixo ou Médio FO
21	523,48	19,27	354,57	692,39	534,00	18,10	483,20	6,80	-7,69	Médio ou Alto FO
22	547,76	23,82	338,97	756,55	587,50	12,80	548,30	6,90	0,10	-----
23	464,89	22,03	271,79	657,99	557,90	13,20	507,50	8,30	9,17	-----
24	566,64	33,39	273,96	859,32	633,30	19,30	602,20	7,10	6,28	-----
<b>Média</b>	<b>451,79</b>	<b>17,10</b>			<b>465,10</b>	<b>11,50</b>	<b>428,20</b>	<b>7,30</b>	<b>-5,22</b>	

- 1) Número de peças: existe uma discreta tendência de melhores resultados para a variável de resposta *makespan* em grupos formados por um maior número de peças, situado entre 12 e 18. Porém, em valores percentuais, estes ganhos se apresentam menos representativos do que àqueles encontrados no grupo C, apresentados na tabela 7.2; e
- 2) Flexibilidades predominantes: observou-se que nos grupos em que a característica predominante é a flexibilidade de operações, com seu nível variando entre médio e alto, ou flexibilidade de processo, com nível variando entre baixo e médio, ou ainda a flexibilidade de seqüência, em qualquer um de seus níveis (alto, médio ou baixo), independente do número de peças que os compõem, encontram-se os piores resultados para a variável *makespan*.

### 7.3.2 *Makespan* – flexibilidades de operação, seqüência e máquina

Nesta segunda fase do projeto de experimentos são comparados os resultados da variável de resposta *makespan*, obtidos pelas abordagens de KIM et al. (2003) e POTTER (1997), com aqueles resultantes da exclusão da flexibilidade de processo dos planos de processos das peças que compõem o modelo. Para isto são utilizados os grupos A e B anteriormente citados no item 7.1, que trata dos parâmetros de simulação utilizados neste trabalho.

Na seqüência são comparados os resultados obtidos para a variável *makespan* somente entre os grupos A, que permite flexibilidade de operações e seqüência, com o grupo B, que permite flexibilidade de operações, seqüência e máquina.

A escolha da exclusão da flexibilidade de processo dos grupos de simulação A e B baseia-se no fato de que, conforme discutido no item 4.1.2 que trata da caracterização do ambiente modelado, em um *layout job shop* (funcional) não existem restrições nas exigências associadas a cada uma das operações, o que significa que uma tarefa pode ser executada em um ou em vários recursos, sem obedecer a nenhum critério específico, ou seja, não há necessariamente restrições quanto:

- a) Ao uso de diferentes máquinas para o processamento de uma mesma operação, desde que haja capacidade tecnológica para isto (flexibilidade de operações);
  - b) À mudança nas seqüências de operações, desde que respeitadas às precedências entres as mesmas (flexibilidade de seqüência); e
  - c) À substituição de operações ou grupos de operações (caminhos), desde que a possibilidade tenha sido prevista no plano de processo a ser executado e que haja da capacidade tecnológica dos recursos (flexibilidade de processo).
-

---

Dentro do contexto acima, observou-se que apesar de não haver restrições quanto ao uso das diversas flexibilidades nos planos de processo em um ambiente funcional, a exclusão da flexibilidade de operações ou de seqüência descaracterizaria o problema de agendamento proposto. Entretanto, a exclusão da flexibilidade de processo não implicaria na mudança do *layout* ou na descaracterização do problema como *NP-hard* (*non deterministic polynomial-time hard*).

Outra observação relevante deve ser feita a fim de facilitar a compreensão da forma como os dados serão apresentados deste ponto em diante. Ao se excluir a flexibilidade de processo dos planos de processo das peças modeladas são alterados os tipos e níveis de flexibilidades predominantes nos grupos de simulação. Esta alteração é apresentada na tabela 7.4.

Ao recalcular os níveis de flexibilidade das peças que compõem os grupos diretamente influenciados, ou seja, aqueles que possuíam a flexibilidade de processo como característica predominante observa-se que os mesmos passam a não mais possuir uma característica definida. Isto ocorre com os grupos de simulação 1, 2, 3, 10, 11, 16 e 17.

---

Tabela 7.4 – Comparação dos tipos e níveis de flexibilidades predominantes com e sem o uso de flexibilidade de operações

Grupo	Peças	Nível de Flexibilidade Predominante com Flexibilidade de Operações	Nível de Flexibilidade Predominante sem Flexibilidade de Operações
01	1, 2, 3, 10, 11, 12	Baixo FP	-----
02	4, 5, 6, 13, 14, 15	Médio FP	-----
03	7, 8, 9, 16, 17, 18	Alto FP	-----
04	1, 4, 7, 10, 13, 16	Baixo FS	Baixo FS
05	2, 5, 8, 11, 14, 17	Médio FS	Médio FS
06	3, 6, 9, 12, 15, 18	Alto FS	Alto FS
07	1, 4, 8, 12, 15, 17	Baixo FO	Baixo FO
08	2, 6, 7, 10, 14, 18	Médio FO	Médio FO
09	3, 5, 9, 11, 13, 16	Alto FO	Alto FO
10	1, 2, 3, 5, 6, 10, 11, 12, 15	Baixo ou Médio FP	-----
11	4, 7, 8, 9, 13, 14, 16, 17, 18	Médio ou Alto FP	-----
12	1, 4, 5, 7, 8, 10, 13, 14, 16	Baixo ou Médio FS	Baixo ou Médio FS
13	2, 3, 6, 9, 11, 12, 15, 17, 18	Médio ou Alto FS	Médio ou Alto FS
14	1, 2, 4, 7, 8, 12, 15, 17, 18	Baixo ou Médio FO	Baixo ou Médio FO
15	3, 5, 6, 9, 10, 11, 13, 14, 16	Médio ou Alto FO	Médio ou Alto FO
16	1, 2, 3, 4, 5, 6, 10, 11, 12, 13, 14, 15	Baixo ou Médio FP	-----
17	4, 5, 6, 7, 8, 9, 13, 14, 15, 16, 17, 18	Médio ou Alto FP	-----
18	1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17	Baixo ou Médio FS	Baixo ou Médio FS
19	2, 3, 5, 6, 8, 9, 11, 12, 14, 15, 17, 18	Médio ou Alto FS	Médio ou Alto FS
20	1, 2, 4, 6, 7, 8, 10, 12, 14, 15, 17, 18	Baixo ou Médio FO	Baixo ou Médio FO
21	2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 16, 18	Médio ou Alto FO	Médio ou Alto FO
22	2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18	-----	-----
23	1, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18	-----	-----
24	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18	-----	-----

Os valores de *makespan* obtidos através do uso do grupo A, o qual permite somente as flexibilidades de operações e seqüência, ou seja, desconsidera o uso das flexibilidades de processo e máquina, são apresentados na tabela 7.5.

Tabela 7.5 – Comparação dos resultados de *makespan* somente com as flexibilidades de operação e seqüência com os valores obtidos por KIM et al. (2003) e POTTER (1997)

Grupos	Makespan - flexibilidades de operação, seqüência, processo e máquina									
	Grupo A				Potter (1997)		Kim et al. (2003)		Taxa de melhoria (%)	Nível de flexibilidade predominante
	intervalo confiança 95%				média	dv	média	dv		
	média	dv	mínimo	máximo						
1	451,03	7,21	387,83	514,23	470,40	8,60	437,60	10,90	-2,98	-----
2	360,09	10,97	263,93	456,25	369,20	5,60	349,70	5,90	-2,89	-----
3	370,18	21,26	183,83	556,53	382,00	9,10	355,20	7,40	-4,05	-----
4	317,03	8,50	242,52	391,54	321,50	7,60	306,20	0,40	-3,42	Baixo FS
5	353,47	19,14	185,70	521,24	337,80	6,10	323,70	3,60	-8,42	Médio FS
6	455,07	13,46	337,09	573,05	485,60	6,10	443,80	5,00	-2,48	Alto FS
7	384,04	7,39	319,26	448,82	385,60	5,10	372,40	1,30	-3,03	Baixo FO
8	360,40	9,51	277,04	443,76	373,80	7,00	348,30	5,70	-3,36	Médio FO
9	447,88	6,67	389,41	506,35	474,50	8,90	434,90	9,80	-2,90	Alto FO
10	460,76	14,27	335,68	585,84	502,60	12,10	456,50	10,80	-0,92	-----
11	375,49	10,78	281,00	469,98	423,80	10,50	378,90	5,10	0,91	-----
12	356,88	13,18	241,35	472,41	379,50	15,20	332,80	3,40	-6,75	Baixo ou Médio FS
13	465,68	20,04	290,02	641,34	511,10	9,30	469,00	10,70	0,71	Médio ou Alto FS
14	406,57	8,25	334,26	478,88	433,40	12,10	402,40	10,60	-1,03	Baixo ou Médio FO
15	464,55	9,59	380,49	548,61	493,90	11,00	445,20	11,00	-4,17	Médio ou Alto FO
16	477,23	17,78	321,38	633,08	549,70	18,20	478,80	12,00	0,33	-----
17	407,21	8,79	330,16	484,26	496,90	16,00	448,90	8,70	10,24	-----
18	395,42	9,42	312,85	477,99	419,80	17,10	389,60	7,50	-1,47	Baixo ou Médio FS
19	480,18	25,71	254,82	705,54	557,00	12,20	508,10	10,00	5,82	Médio ou Alto FS
20	439,07	17,02	289,88	588,26	482,70	15,60	453,80	5,20	3,36	Baixo ou Médio FO
21	471,39	12,77	359,46	583,32	534,00	18,10	483,20	6,80	2,51	Médio ou Alto FO
22	501,31	14,15	377,28	625,34	587,50	12,80	548,30	6,90	9,37	-----
23	458,21	16,37	314,72	601,70	557,90	13,20	507,50	8,30	10,76	-----
24	541,31	24,32	328,14	754,48	633,30	19,30	602,20	7,10	11,25	-----
<b>Média</b>	<b>425,02</b>	<b>13,61</b>			<b>465,10</b>	<b>11,50</b>	<b>428,20</b>	<b>7,30</b>	<b>0,75</b>	

Analisando os resultados obtidos de forma global, ou seja, através da média das médias da variável de reposta *makespan* (425,02), observa-se que esta média é inferior aquelas obtidas por KIM et al. (2003) e POTTER (1997). Já em relação à média das médias do desvio padrão, esta se apresenta superior àquelas obtidas por KIM et al. (2003) e POTTER (1997).

---

Ao observar os resultados obtidos em cada um dos vinte e quatro grupos de simulação, pode-se notar que a abordagem proposta neste trabalho resulta em um desempenho melhor que aquela apresentada por KIM et al. (2003) nos grupos 11, 13, 16, 17, 19, 20, 21, 22, 23 e 24, sendo porém, melhor que aqueles obtidos pela abordagem sugerida por POTTER (1997) em todos os vinte e quatro grupos de simulação. Também foi observada a influência de alguns fatores distintos:

- 1) Número de peças: persiste a tendência de melhores resultados para a variável de resposta *makespan* em grupos formados por um maior número de peças, em especial aqueles situados entre 12 e 18; e
- 2) Flexibilidades predominantes: observou-se que nos grupos em que a característica predominante é a flexibilidade de seqüência, com seu nível variando entre baixo e médio, independente do número de peças que os compõem, encontram-se os piores resultados para a variável *makespan*.

Os resultados obtidos através do uso do grupo B, o qual permite o uso das flexibilidades de operações, seqüência e máquina, desconsiderando o uso da flexibilidade de processo, são apresentados na tabela 7.6.

Ao comparar-se os resultados obtidos, observa-se que na avaliação global dos vinte e quatro grupos de simulação, que ao permitir-se a flexibilidade de máquina, a média das médias da variável *makespan* passa a ter um pior desempenho em relação ao método apresentado por KIM et al. (2003), porém melhor que a obtida por POTTER (1997). Novamente são encontrados ganhos significativos nos grupos 17, 20, 22, 23 e 24, notando-se que, em valores percentuais, estes ganhos se apresentam inferiores àqueles encontrados no grupo A. Também foi observada a influência de alguns fatores distintos:

- 1) Número de peças: persiste a tendência de melhores resultados para a variável de resposta *makespan* em grupos formados por um maior número de peças, situados entre 12 e 18; e
  - 2) Flexibilidades predominantes: observou-se que nos grupos em que a característica predominante é a flexibilidade de operações, com seu nível variando entre médio e alto, ou a flexibilidade de seqüência, em qualquer um de seus níveis (alto, médio ou baixo), independente do número de peças que o compõe, encontram-se os piores resultados para a variável *makespan*.
-

Em relação ao desvio padrão, os resultados obtidos pela abordagem proposta neste trabalho se mostram novamente mais elevados do que aqueles obtidos por KIM et al. (2003) e POTTER (1997).

Tabela 7.6 – Comparação dos resultados de *makespan* utilizando-se as flexibilidades de operação, seqüência e máquina, com os valores obtidos por KIM et al. (2003) e POTTER (1997)

Grupos	<i>Makespan</i> - flexibilidades de operação, seqüência, processo e máquina									
	Grupo B				Potter (1997)		Kim et al. (2003)		Taxa de melhoria (%)	Nível de flexibilidade predominante
	média	dv	mínimo	máximo	média	dv	média	dv		
1	520,53	13,41	402,99	638,07	470,40	8,60	437,60	10,90	-15,93	-----
2	374,58	7,80	306,21	442,95	369,20	5,60	349,70	5,90	-6,64	-----
3	372,72	7,95	303,04	442,40	382,00	9,10	355,20	7,40	-4,70	-----
4	329,27	2,53	307,09	351,45	321,50	7,60	306,20	0,40	-7,01	Baixo FS
5	362,24	26,00	134,34	590,14	337,80	6,10	323,70	3,60	-10,64	Médio FS
6	513,75	15,05	381,83	645,67	485,60	6,10	443,80	5,00	-13,62	Alto FS
7	386,65	5,10	341,95	431,35	385,60	5,10	372,40	1,30	-3,68	Baixo FO
8	375,73	10,27	285,71	465,75	373,80	7,00	348,30	5,70	-7,30	Médio FO
9	521,21	8,28	448,63	593,79	474,50	8,90	434,90	9,80	-16,56	Alto FO
10	518,38	23,21	314,94	721,82	502,60	12,10	456,50	10,80	-11,94	-----
11	387,15	24,40	173,27	601,03	423,80	10,50	378,90	5,10	-2,13	-----
12	391,73	32,46	107,21	676,25	379,50	15,20	332,80	3,40	-15,04	Baixo ou Médio FS
13	520,45	11,57	419,03	621,87	511,10	9,30	469,00	10,70	-9,89	Médio ou Alto FS
14	404,00	10,75	309,77	498,23	433,40	12,10	402,40	10,60	-0,40	Baixo ou Médio FO
15	529,41	11,65	427,29	631,53	493,90	11,00	445,20	11,00	-15,91	Médio ou Alto FO
16	533,26	27,89	288,79	777,73	549,70	18,20	478,80	12,00	-10,21	-----
17	424,25	21,17	238,69	609,81	496,90	16,00	448,90	8,70	5,81	-----
18	411,84	18,68	248,10	575,58	419,80	17,10	389,60	7,50	-5,40	Baixo ou Médio FS
19	532,05	25,01	312,83	751,27	557,00	12,20	508,10	10,00	-4,50	Médio ou Alto FS
20	441,95	33,51	148,22	735,68	482,70	15,60	453,80	5,20	2,68	Baixo ou Médio FO
21	523,98	16,61	378,39	669,57	534,00	18,10	483,20	6,80	-7,78	Médio ou Alto FO
22	541,23	17,00	392,22	690,24	587,50	12,80	548,30	6,90	1,31	-----
23	475,47	29,33	218,38	732,56	557,90	13,20	507,50	8,30	6,74	-----
24	563,72	19,65	391,48	735,96	633,30	19,30	602,20	7,10	6,83	-----
<b>Média</b>	<b>456,48</b>	<b>17,47</b>			<b>465,1</b>	<b>11,5</b>	<b>428,2</b>	<b>7,3</b>	<b>-6,2</b>	

### 7.3.3 Conclusões finais a respeito da variável de resposta *Makespan*

Finalizando a análise da variável de resposta *makespan*, são comparados os resultados obtidos para os grupos de flexibilidade A, B, C e D, conforme apresentado na tabela 7.7.

Deve-se lembrar que, conforme citado no item 7.1, que trata dos parâmetros de simulação, cada grupo permite o uso de diferentes flexibilidades nos planos de processo das peças simuladas.

Na tabela 7.7 são assinalados todos os resultados cujo valor da variável de resposta *makespan* encontrados foram inferiores àqueles obtidos pelas abordagens de KIM et al. (2003) e POTTER (1997).

Tabela 7.7 - Comparação dos resultados de *makespan* entre os grupos A, B, C e D

Grupos	<i>Makespan</i>							
	Grupo A		Grupo B		Grupo C		Grupo D	
	média	dv	média	dv	média	dv	média	dv
1	451,03	7,21	520,53	13,41	445,37	7,93	517,73	8,96
2	360,09	10,97	374,58	7,80	380,25	11,43	369,62	7,74
3	370,18	21,26	372,72	7,95	373,13	11,75	366,72	7,35
4	317,03	8,50	329,27	2,53	327,45	9,23	324,02	11,17
5	353,47	19,14	362,24	26,00	345,47	11,32	364,06	38,16
6	455,07	13,46	513,75	15,05	445,35	8,05	504,01	7,06
7	384,04	7,39	386,65	5,10	377,62	2,9	381,64	1,37
8	360,40	9,51	375,73	10,27	375,11	18,21	373,14	12,20
9	447,88	6,67	521,21	8,28	446,35	7,25	520,26	10,54
10	460,76	14,27	518,38	23,21	450,97	8,57	520,76	23,79
11	375,49	10,78	387,15	24,40	389,23	9,76	379,41	9,74
12	356,88	13,18	391,73	32,46	357,05	13,41	367,90	18,91
13	465,68	20,04	520,45	11,57	470,93	15,89	513,26	11,25
14	406,57	8,25	404,00	10,75	396,83	12,28	394,67	13,50
15	464,55	9,59	529,41	11,65	459,35	9,95	522,39	18,79
16	477,23	17,78	533,26	27,89	483,61	15,6	533,18	26,98
17	407,21	8,79	424,25	21,17	429,57	20,38	432,11	21,69
18	395,42	9,42	411,84	18,68	403,21	22,18	401,65	23,45
19	480,18	25,71	532,05	25,01	480,18	14,46	521,02	23,12
20	439,07	17,02	441,95	33,51	434,46	14,99	432,56	16,23
21	471,39	12,77	523,98	16,61	485,39	19,38	523,48	19,27
22	501,31	14,15	541,23	17,00	494,74	19,05	547,76	23,82
23	458,21	16,37	475,47	29,33	452,96	11,66	464,89	22,03
24	541,31	24,32	563,72	19,65	534,14	16,57	566,64	33,39
<b>Média</b>	<b>425,02</b>	<b>13,61</b>	<b>456,48</b>	<b>17,47</b>	<b>426,61</b>	<b>13,01</b>	<b>451,79</b>	<b>17,10</b>

São analisados, primeiramente, os valores globais da variável de resposta, relativos à média das médias do *makespan* obtido, bem como a média do desvio-padrão de cada um dos grupos analisados. Sob este prisma observa-se que:

- 1) Os maiores ganhos, percentuais e na média global, da variável de resposta *makespan*, foram encontrados no grupo A, que permite o menor número de flexibilidades (somente operação e seqüência);

- 
- 2) Ao se incluir a flexibilidade de processo (Grupo C), e comparando-se os resultados com aqueles obtidos pelo Grupo A, que permite apenas a flexibilidade de operação e seqüência, observa-se que os valores globais da variável *makespan* situam-se bastante próximos. Observa-se também que o desvio padrão encontrado em ambos os grupos são equivalentes, com o resultado obtido pelo grupo C sendo um pouco menor. Isto indica que, apesar do incremento no número de opções de plano de processo, com a inclusão da flexibilidade de processo, não há uma influência direta na variável de resposta *makespan*;
  - 3) Em relação à flexibilidade de máquina, observa-se que nos grupos em que a mesma foi incluída (grupos B e D) o desempenho global da variável *makespan* foi pior do que os demais grupos (A e C). Este resultado é explicado tendo-se em vista que o agrupamento de operações foi utilizado considerando-se os tempos de *setup* de forma independente. Ou seja, as parcelas de tempo de *setup* de preparação e fixação estão inseridas na parcela de tempo de processamento. Desta forma, a heurística que rege o tempo de proposta no modelo desenvolvido, discutida anteriormente no item 6.1.3, não prioriza o menor tempo total de conclusão, mas sim a menor quantidade de troca de recursos. Isto faz com que muitas vezes os recursos escolhidos por este critério sejam os mais lentos em tempo de processamento;
  - 4) Ainda sobre a análise do uso de flexibilidade de máquina, observou-se que a inclusão ou não de mais uma flexibilidade gerou pouco impacto nos resultados obtidos, o que pode ser observado nas médias globais da variável *makespan* nos grupos D e B;
  - 5) Observa-se, ainda, que os maiores ganhos se concentram nos grupos formados por um maior número de peças (a partir de nove), com destaque para os grupos com o número de peças variando entre doze e dezoito. Por exemplo, nos grupos 22 (15 peças), 23 (15 peças) e 24 (18 peças), o melhor desempenho da variável *makespan* mostra-se independente do tipo de combinação de flexibilidades.

Em relação à influência observada entre as flexibilidades predominantes e seus respectivos níveis, em cada um dos vinte e quatro grupos de simulação em relação à variável *makespan*, foi observado que:

- 1) Nos grupos onde foi permitido o uso de flexibilidade de máquina, através do agrupamento de operações (grupos B e D), detectou-se uma influência direta e negativa da flexibilidade de operações em seus níveis médio e alto, e da flexibilidade de seqüência em todos os seus níveis, baixo, médio e alto. Em ambos os casos foram obtidos os piores valores para a variável *makespan*;
-

- 
- 2) O uso da flexibilidade de seqüência em seus níveis baixo e médio, afeta de modo negativo os valores para a variável *makespan*. Isto ocorre independentemente da combinação de flexibilidade dos grupos de simulação, ou seja, nos grupos A, B, C e D; e
  - 3) A flexibilidade de operações, com seus níveis de predominância variando entre médio e alto, quando combinada com as flexibilidades de processo e máquina (grupos B, C e D), afetam de modo negativo os valores para a variável *makespan*.

As figuras 7.4 A 7.7 mostram as médias da variável de resposta *makespan* obtidas para cada um dos 24 grupos de simulação, divididos em relação ao número de peças que o compõem (entre 6 e 18 peças), independentemente da flexibilidade predominante em cada grupo.

Observa-se novamente nestas figuras que, ao aumentar-se o número de peças que compõem o grupo, menor é o valor da média da variável *makespan* obtida em relação aos valores apresentados por KIM et al. (2003) e POTTER (1997), em especial em grupos compostos a partir de 12 peças. Ou seja, ao se aumentar o número de peças que compõem um grupo de simulação, a abordagem baseada em agentes, sugerida neste trabalho, apresenta melhores resultados que as abordagens baseadas em algoritmos genéticos sugeridas por KIM et al. (2003) e POTTER (1997).

---

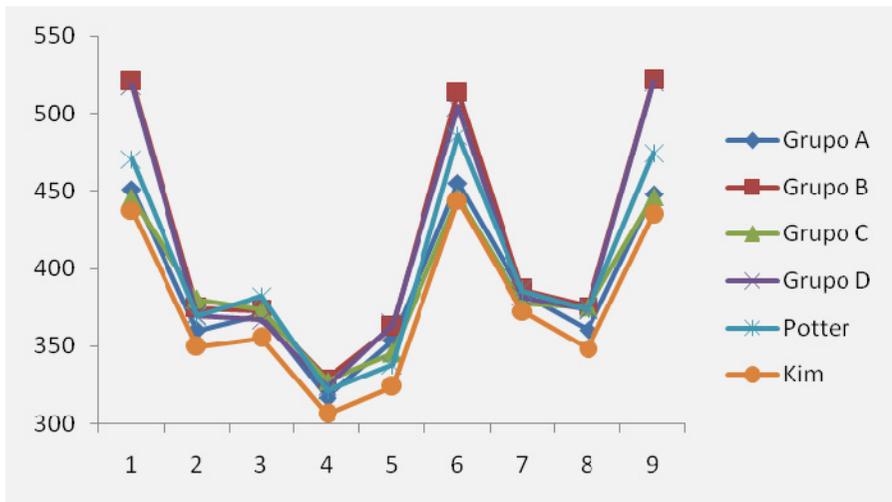


Figura 7.4 – Makespan dos grupos formados por 6 peças

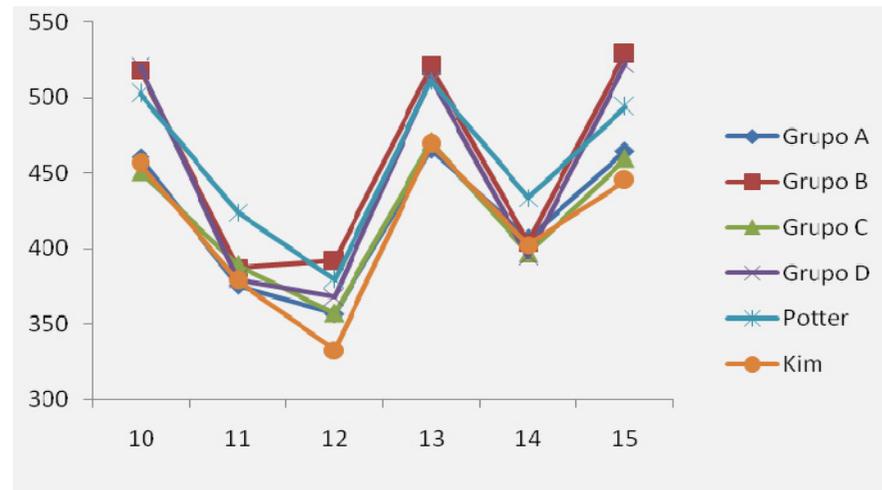


Figura 7.6 – Makespan dos grupos formados por 12 peças

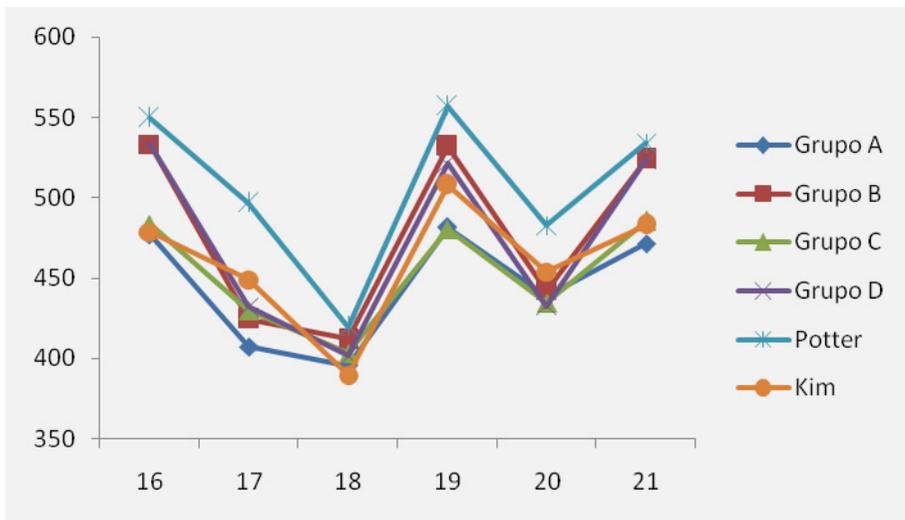


Figura 7.5 – Makespan dos grupos formados por 9 peças

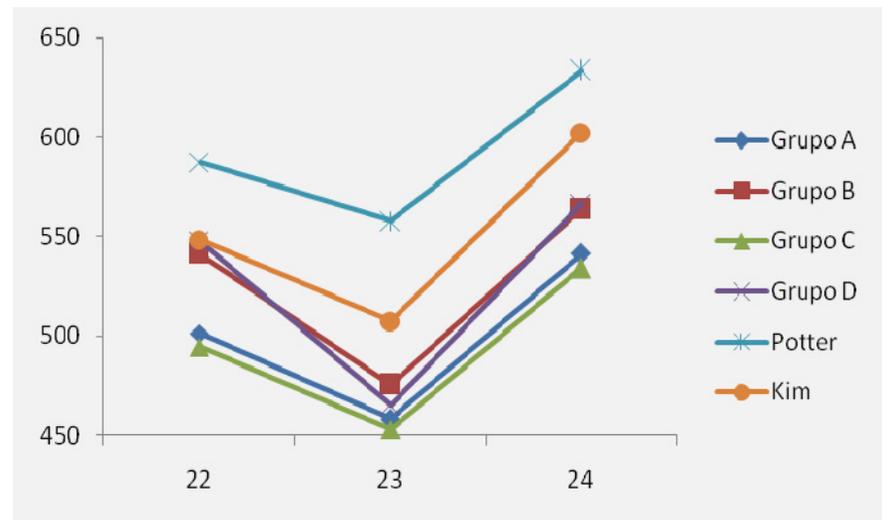


Figura 7.7 – Makespan dos grupos formados por 15 ou 18 peças

## 7.4 Análise dos resultados – *flow time*

### 7.4.1 *Flow time* – flexibilidades de operação, seqüência, processos e máquina

Novamente são replicadas pelo modelo proposto, as condições exatas apresentadas pelas abordagens de KIM et al. (2003) e POTTER (1997) no que diz respeito às condições de flexibilidade, sendo feita uma comparação direta entre os resultados obtidos. A tabela 7.8 apresenta os resultados desta comparação para a variável de resposta *flow time*.

Tabela 7.8 – Comparação dos resultados obtidos pela abordagem proposta para a variável *flow time* para a mesma condição de flexibilidade de KIM et al. (2003) e POTTER (1997)

Grupos	<i>Flow time</i> - flexibilidades de operação, seqüência e processo									
	Grupo C				Potter (1997)		Kim et al, (2003)		Taxa de melhoria (%)	Nível de flexibilidade predominante
	média	dv	intervalo confiança 95%		média	dv	média	dv		
1	318,24	4,15	281,90	354,58	347,70	4,50	318,90	3,70	0,21	Baixo FP
2	297,59	2,10	279,17	316,01	307,00	5,50	287,70	4,70	-3,32	Médio FP
3	311,89	2,77	287,64	336,13	334,70	4,20	304,80	4,30	-2,27	Alto FP
4	264,29	2,11	245,78	282,80	245,90	2,50	251,30	4,80	-4,92	Baixo FS
5	285,65	4,18	249,00	322,31	297,70	3,90	280,30	3,20	-1,87	Médio FS
6	374,33	5,15	329,15	419,50	415,60	7,20	384,70	5,70	2,77	Alto FS
7	322,49	2,90	297,08	347,90	331,50	5,50	314,10	2,60	-2,60	Baixo FO
8	309,36	5,92	257,48	361,25	318,10	6,10	295,20	5,00	-4,58	Médio FO
9	294,22	2,41	273,09	315,36	324,90	6,20	298,90	7,00	1,59	Alto FO
10	334,87	6,32	279,46	390,29	383,10	7,00	349,20	6,10	4,28	Baixo ou Médio FP
11	317,46	6,32	262,07	372,85	344,40	4,60	312,90	7,60	-1,44	Médio ou Alto FP
12	287,04	7,05	225,21	348,88	313,70	6,50	279,60	4,70	-2,59	Baixo ou Médio FS
13	364,82	7,07	302,85	426,78	417,00	7,50	387,00	7,10	6,08	Médio ou Alto FS
14	341,57	6,12	287,95	395,19	372,00	5,10	346,90	8,50	1,56	Baixo ou Médio FO
15	300,90	4,76	259,18	342,62	349,50	6,50	316,10	6,20	5,05	Médio ou Alto FO
16	342,46	7,37	277,86	407,06	413,40	8,70	359,70	4,30	5,03	Baixo ou Médio FP
17	345,54	6,07	292,35	398,72	401,60	7,90	364,70	4,70	5,55	Médio ou Alto FP
18	311,36	3,92	277,01	345,71	342,30	7,30	322,50	6,40	3,58	Baixo ou Médio FS
19	365,83	6,42	309,57	422,09	434,30	8,40	406,40	4,60	11,09	Médio ou Alto FS
20	358,36	4,62	317,84	398,88	394,50	7,90	372,00	5,70	3,81	Baixo ou Médio FO
21	335,76	6,54	278,46	393,06	389,60	7,80	365,40	8,20	8,83	Médio ou Alto FO
22	368,75	3,23	340,44	397,05	448,60	7,60	417,80	5,80	13,30	-----
23	359,92	6,87	299,73	420,12	434,70	8,30	404,70	5,10	12,44	-----
24	397,52	6,59	339,76	455,28	480,00	8,70	452,90	7,50	13,93	-----
<b>Média</b>	<b>329,59</b>	<b>5,04</b>			<b>368,41</b>	<b>6,48</b>	<b>341,40</b>	<b>5,56</b>	<b>3,15</b>	

---

Primeiramente são analisados os resultados obtidos de forma global, através da média das médias da variável *flow time*. É possível observar que esta média (329,59) é menor do que aquelas obtidas por KIM et al. (2003) e POTTER (1997).

Em relação ao desvio padrão, a abordagem sugerida também apresenta um valor menor do que os obtidos por KIM et al. (2003) e POTTER (1997), indicando uma menor variabilidade no processo em relação às demais abordagens.

Ao se realizar uma análise caso a caso, observa-se que a abordagem proposta apresenta um desempenho satisfatório em relação às demais abordagens, tendo em vista que este resultado é proveniente de uma heurística simples, ao contrário das abordagens de KIM et al. (2003) e POTTER (1997). Nesta análise foi observada a influência dos seguintes fatores:

- 1) Número de peças: da mesma maneira que o *makespan* há uma clara tendência de melhores resultados para a variável de resposta *flow time* em grupos formados por um maior número de peças, em especial aqueles situados entre 12 e 18 peças; e
- 2) Flexibilidades predominantes: os piores resultados para a variável de resposta *flow time* encontram-se nos grupos em que a característica predominante é a flexibilidade de processo, com seu nível variando entre médio e alto, ou a flexibilidade de seqüência, com seus níveis variando entre baixo e médio, especificamente em grupos compostos em até nove peças.

A seguir são comparados os valores obtidos por KIM et al. (2003) e POTTER (1997) com aqueles obtidos neste trabalho através da inclusão do mecanismo de agrupamento de operações, representados pelo grupo D. Na tabela 7.9 pode-se observar os resultados para a variável de resposta *flow time* ao comparar-se as abordagens de KIM et al. (2003) e POTTER (1997), sem flexibilidade de máquina, e a abordagem com agrupamento de operações e flexibilidade de máquina.

Ao serem analisados os resultados obtidos de forma global, através da média das médias da variável *flow time*, pode-se observar que esta média (332,21) novamente é menor do que aquelas obtidas por de KIM et al. (2003) e POTTER (1997), porém maior que a obtida pelo grupo de simulação C.

Em relação ao desvio padrão, este se apresenta levemente superior àqueles obtidos por KIM et al. (2003), e inferior aos obtidos por POTTER (1997), indicando novamente uma pequena variabilidade no processo.

Ao se realizar uma análise caso a caso, foi observada a influência dos seguintes fatores:

---

1) Número de peças: observa-se que, ao permitir-se o uso das quatro flexibilidades nos planos de processo, isto resulta em melhores resultados no caso dos grupos formados por um maior número de peças, especificamente a partir de nove; e

Tabela 7.9 – Comparação dos resultados de *flow time* com a inclusão da flexibilidade de máquina com os valores obtidos por KIM et al. (2003) e POTTER (1997)

Grupos	Flow time - flexibilidades de operação, seqüência e processo									
	Grupo D		Potter (1997)				Kim et al, (2003)		Taxa de melhoria (%)	Nível de flexibilidade predominante
	média	dv	intervalo confiança 95%		média	dv	média	dv		
1	337,74	4,50	298,33	377,15	347,70	4,50	318,90	3,70	-5,58	Baixo FP
2	292,28	2,08	274,04	310,51	307,00	5,50	287,70	4,70	-1,57	Médio FP
3	310,15	3,98	275,27	345,03	334,70	4,20	304,80	4,30	-1,73	Alto FP
4	255,56	3,68	223,32	287,80	245,90	2,50	251,30	4,80	-1,67	Baixo FS
5	292,03	8,50	217,49	366,57	297,70	3,90	280,30	3,20	-4,02	Médio FS
6	392,54	3,40	362,78	422,30	415,60	7,20	384,70	5,70	-2,00	Alto FS
7	333,82	4,62	293,29	374,35	331,50	5,50	314,10	2,60	-5,91	Baixo FO
8	305,68	10,05	217,55	393,80	318,10	6,10	295,20	5,00	-3,43	Médio FO
9	304,92	2,67	281,56	328,28	324,90	6,20	298,90	7,00	-1,98	Alto FO
10	342,96	5,60	293,91	392,02	383,10	7,00	349,20	6,10	1,82	Baixo ou Médio FP
11	304,79	5,53	256,32	353,26	344,40	4,60	312,90	7,60	2,66	Médio ou Alto FP
12	277,61	7,15	214,92	340,31	313,70	6,50	279,60	4,70	0,72	Baixo ou Médio FS
13	376,76	6,06	323,62	429,91	417,00	7,50	387,00	7,10	2,72	Médio ou Alto FS
14	345,11	6,69	286,48	403,75	372,00	5,10	346,90	8,50	0,52	Baixo ou Médio FO
15	306,78	4,93	263,54	350,03	349,50	6,50	316,10	6,20	3,04	Médio ou Alto FO
16	348,01	5,69	298,12	397,90	413,40	8,70	359,70	4,30	3,36	Baixo ou Médio FP
17	339,43	6,23	284,86	394,00	401,60	7,90	364,70	4,70	7,44	Médio ou Alto FP
18	308,40	8,10	237,37	379,43	342,30	7,30	322,50	6,40	4,57	Baixo ou Médio FS
19	370,71	6,70	311,99	429,43	434,30	8,40	406,40	4,60	9,63	Médio ou Alto FS
20	357,63	5,94	305,61	409,65	394,50	7,90	372,00	5,70	4,02	Baixo ou Médio FO
21	335,96	5,50	287,73	384,20	389,60	7,80	365,40	8,20	8,76	Médio ou Alto FO
22	370,54	6,10	317,06	424,02	448,60	7,60	417,80	5,80	12,75	-----
23	364,66	7,03	303,00	426,32	434,70	8,30	404,70	5,10	10,98	-----
24	399,06	6,67	340,58	457,55	480,00	8,70	452,90	7,50	13,49	-----
<b>Média</b>	<b>332,21</b>	<b>5,73</b>			<b>368,41</b>	<b>6,48</b>	<b>341,40</b>	<b>5,56</b>		

2) Flexibilidades predominantes: neste caso não foi observada uma influência específica relacionada a um único tipo ou nível de flexibilidade nos resultados obtidos.

7.4.2 *Flow time* – flexibilidades de operação, seqüência e máquina

Nesta fase do projeto de experimentos são comparados os resultados da variável de resposta *flow time* obtidos pelas abordagens de KIM et al. (2003) e POTTER (1997) com aqueles resultantes da exclusão da flexibilidade de processo dos planos de processos das peças que compõem o modelo, compostos pelos grupos A e B.

Primeiramente são discutidos os valores de *flow time* obtidos através do uso do grupo A, o qual permite somente as flexibilidades de operações e seqüência. Estes valores são apresentados na tabela 7.10.

Tabela 7.10 – Comparação dos resultados de *makespan* somente com as flexibilidades de operação e seqüência com os valores obtidos por KIM et al. (2003) e POTTER (1997)

Grupos	Flow time - flexibilidades de operação e seqüência									
	Grupo A				Potter (1997)		Kim et al, (2003)		Taxa de melhoria (%)	Nível de flexibilidade predominante
	média	dv	intervalo confiança 95%		média	dv	média	dv		
1	318,01	4,25	280,78	355,25	347,70	4,50	318,90	3,70	0,28	-----
2	288,42	6,38	232,53	344,32	307,00	5,50	287,70	4,70	-0,25	-----
3	307,43	3,51	276,68	338,17	334,70	4,20	304,80	4,30	-0,85	-----
4	249,51	2,59	226,82	272,20	245,90	2,50	251,30	4,80	0,72	Baixo FS
5	286,17	6,30	230,97	341,37	297,70	3,90	280,30	3,20	-2,05	Médio FS
6	375,20	6,37	319,34	431,06	415,60	7,20	384,70	5,70	2,53	Alto FS
7	333,15	5,90	281,47	384,83	331,50	5,50	314,10	2,60	-5,72	Baixo FO
8	297,88	4,74	256,38	339,39	318,10	6,10	295,20	5,00	-0,90	Médio FO
9	287,64	2,82	262,89	312,39	324,90	6,20	298,90	7,00	3,92	Alto FO
10	334,18	5,78	283,50	384,86	383,10	7,00	349,20	6,10	4,49	-----
11	302,53	4,57	262,44	342,62	344,40	4,60	312,90	7,60	3,43	-----
12	272,62	3,68	240,39	304,85	313,70	6,50	279,60	4,70	2,56	Baixo ou Médio FS
13	362,77	6,60	304,89	420,65	417,00	7,50	387,00	7,10	6,68	Médio ou Alto FS
14	339,45	3,71	306,89	372,01	372,00	5,10	346,90	8,50	2,19	Baixo ou Médio FO
15	300,66	5,91	248,83	352,50	349,50	6,50	316,10	6,20	5,13	Médio ou Alto FO
16	334,75	4,65	294,03	375,48	413,40	8,70	359,70	4,30	7,45	-----
17	332,99	5,64	283,53	382,45	401,60	7,90	364,70	4,70	9,52	-----
18	300,54	5,91	248,71	352,37	342,30	7,30	322,50	6,40	7,31	Baixo ou Médio FS
19	362,95	5,13	318,01	407,89	434,30	8,40	406,40	4,60	11,97	Médio ou Alto FS
20	354,08	10,58	261,30	446,85	394,50	7,90	372,00	5,70	5,06	Baixo ou Médio FO
21	322,54	4,42	283,84	361,24	389,60	7,80	365,40	8,20	13,29	Médio ou Alto FO
22	361,30	4,38	322,91	399,69	448,60	7,60	417,80	5,80	15,64	-----
23	354,83	5,73	304,60	405,05	434,70	8,30	404,70	5,10	14,06	-----
24	390,52	6,49	333,61	447,43	480,00	8,70	452,90	7,50	15,97	-----
<b>Média</b>	<b>323,76</b>	<b>5,25</b>			<b>368,41</b>	<b>6,48</b>	<b>341,40</b>	<b>5,56</b>		

---

Ao analisarem-se os resultados obtidos de forma global, ou seja, através da média das médias da variável de resposta *flow time*, observa-se que esta média (323,76) é menor do que aquelas obtidas por KIM et al. (2003) e POTTER (1997).

Em relação ao desvio-padrão, a abordagem sugerida também apresenta um valor inferior a KIM et al. (2003) e POTTER (1997), indicando uma menor variabilidade no processo em relação às demais abordagens.

Ao se realizar uma análise caso a caso, foi observada a influência dos seguintes fatores:

- 1) Número de peças: os maiores ganhos não se concentram somente nos grupos formados por um grande número de peças. Ao contrário do grupo C, onde os melhores resultados se concentravam a partir de 12 peças, no grupo A estes resultados iniciam a partir de 6 peças, embora sejam mais significativos nos grupos formados por 9 peças ou mais; e
- 2) Flexibilidades predominantes: neste caso não foi observada nenhuma influência relacionada a um único tipo ou nível de flexibilidade predominante nos resultados obtidos.

A seguir são analisados os resultados obtidos através do uso do grupo B, o qual permite o uso das flexibilidades de operações, seqüência e máquina, desconsiderando o uso da flexibilidade de processo. Estes valores são apresentados na tabela 7.11.

Na avaliação global dos vinte e quatro grupos de simulação, observa-se que ao permitir-se a flexibilidade de máquina, a média das médias da variável – *flow time* continua apresentando um desempenho melhor do que aquela obtida por KIM et al. (2003) e POTTER (1997). Em relação ao desvio padrão, os valores obtidos pela abordagem proposta se mostram novamente maiores do que aqueles obtidos por KIM et al. (2003), porém menores do que os obtidos por POTTER (1997).

Ao se realizar uma análise caso a caso, foi observada a influência dos seguintes fatores:

- 1) Número de peças: nota-se claramente que os ganhos se concentram quase que exclusivamente nos grupos formados por um número elevado de peças, entre doze e dezoito; e
  - 2) Flexibilidades predominantes: há uma tendência para um baixo desempenho da variável de resposta *flow time* quando associado aos níveis de flexibilidade de seqüência variando entre baixo e médio e a todos os níveis de flexibilidade de operações (baixo, médio e alto), para grupos compostos por até nove peças.
-

Tabela 7.11 – Comparação dos resultados de *flow time* utilizando-se as flexibilidades de operação, seqüência e máquina, com os valores obtidos por KIM et al. (2003) e POTTER (1997)

Grupos	Flow time - flexibilidades de operação e seqüência									
	Grupo B				Potter (1997)		Kim et al, (2003)		Taxa de melhoria (%)	Nível de flexibilidade predominante
	intervalo confiança 95%				média	dv	média	dv		
média	dv	mínimo	máximo	média	dv	média	dv			
1	338,58	4,01	303,40	373,76	347,70	4,50	318,90	3,70	-5,81	-----
2	305,75	4,86	263,19	348,32	307,00	5,50	287,70	4,70	-5,90	-----
3	320,16	5,75	269,75	370,57	334,70	4,20	304,80	4,30	-4,80	-----
4	271,23	1,24	260,39	282,07	245,90	2,50	251,30	4,80	-7,35	Baixo FS
5	292,16	5,43	244,60	339,71	297,70	3,90	280,30	3,20	-4,06	Médio FS
6	394,52	5,69	344,66	444,37	415,60	7,20	384,70	5,70	-2,49	Alto FS
7	336,94	7,56	270,65	403,22	331,50	5,50	314,10	2,60	-6,78	Baixo FO
8	310,34	5,37	263,26	357,41	318,10	6,10	295,20	5,00	-4,88	Médio FO
9	317,68	3,41	287,76	347,59	324,90	6,20	298,90	7,00	-5,91	Alto FO
10	350,08	7,93	280,56	419,59	383,10	7,00	349,20	6,10	-0,25	-----
11	319,97	6,95	259,05	380,90	344,40	4,60	312,90	7,60	-2,21	-----
12	296,58	6,50	239,63	353,54	313,70	6,50	279,60	4,70	-5,73	Baixo ou Médio FS
13	370,82	5,35	323,96	417,68	417,00	7,50	387,00	7,10	4,36	Médio ou Alto FS
14	349,40	6,79	289,86	408,95	372,00	5,10	346,90	8,50	-0,72	Baixo ou Médio FO
15	321,17	8,29	248,47	393,88	349,50	6,50	316,10	6,20	-1,58	Médio ou Alto FO
16	349,83	7,85	281,06	418,59	413,40	8,70	359,70	4,30	2,82	-----
17	345,17	6,03	292,29	398,06	401,60	7,90	364,70	4,70	5,66	-----
18	315,75	6,70	257,04	374,46	342,30	7,30	322,50	6,40	2,14	Baixo ou Médio FS
19	369,63	5,35	322,73	416,52	434,30	8,40	406,40	4,60	9,95	Médio ou Alto FS
20	356,47	3,89	322,35	390,59	394,50	7,90	372,00	5,70	4,36	Baixo ou Médio FO
21	344,24	5,12	299,35	389,14	389,60	7,80	365,40	8,20	6,15	Médio ou Alto FO
22	378,74	5,98	326,31	431,16	448,60	7,60	417,80	5,80	10,31	-----
23	371,15	6,67	312,68	429,62	434,70	8,30	404,70	5,10	9,04	-----
24	411,63	5,17	366,34	456,92	480,00	8,70	452,90	7,50	10,03	-----
<b>Média</b>	<b>339,08</b>	<b>5,75</b>			<b>368,41</b>	<b>6,48</b>	<b>341,40</b>	<b>5,56</b>		

#### 7.4.3 Conclusões finais a respeito da variável de resposta *flow time*

Finalizando a análise da variável de resposta *flow time*, são comparados os resultados obtidos para os grupos de flexibilidade A, B, C e D, conforme apresentado na tabela 7.12, destacando-se que, conforme citado no item 7.1, que trata dos parâmetros de simulação, cada grupo permite o uso de diferentes flexibilidades nos planos de processo das peças simuladas.

---

Nesta tabela são realçados todos os resultados encontrados cujo valor da variável de resposta *flow time* são menores do que aqueles obtidos pelas abordagens de KIM et al. (2003) e POTTER (1997).

Primeiramente são analisados os valores globais da variável de resposta, relativos à média das médias do *makespan* obtido, bem como a média do desvio padrão de cada um dos grupos analisados. Sob este prisma observa-se que:

- 1) Os maiores ganhos da variável de resposta *flow time*, tanto percentuais como em relação à média global, foram encontrados no grupo A, que permite o menor número de flexibilidades (somente operação e seqüência);
  - 2) Ao se incluir a flexibilidade de processo (Grupo C), e comparando-se os resultados com aqueles obtidos pelo Grupo A, que permite apenas a flexibilidade de operação e seqüência, os valores globais da variável *flow time* situam-se próximos. O desvio padrão encontrado em ambos os grupos são equivalentes, com o resultado obtido pelo grupo C sendo um pouco menor. Este valor indica que, apesar do incremento no número de opções de plano de processo, com a inclusão da flexibilidade de processo, isto não resulta em uma influência direta na variável de resposta *flow time*.
  - 3) Nos grupos em que a flexibilidade de máquina foi incluída (isto é, grupos B e D), o desempenho global da variável *flow time* foi pior em relação aos grupos A e C. Este resultado é explicado tendo-se em vista que o agrupamento de operações foi utilizado considerando-se os tempos de setup de forma independente. Ou seja, conforme já explicado anteriormente, as parcelas de tempo de setup de preparação e fixação estão inseridas na parcela de tempo de processamento. Desta forma, a heurística que rege o tempo de proposta no modelo desenvolvido, anteriormente discutida no item 7.1.3, não prioriza o menor tempo total de conclusão, mas sim o menor número de troca de recursos. Isto faz com que muitas vezes os recursos escolhidos por este critério sejam os mais lentos em tempo de processamento;
  - 4) Os maiores ganhos se concentram nos grupos formados por um maior número de peças, a partir de nove, com destaque para os grupos com o número de peças variando entre doze e dezoito. Nos grupos 16 a 24, formados por peças que variam entre doze e dezoito peças, encontra-se o melhor desempenho da variável *flow time*, independente do tipo de combinação de flexibilidades.
-

Tabela 7.12 - Comparação dos resultados de flow time entre os grupos A, B, C e D

Grupos	<i>Flow time</i>							
	Grupo A		Grupo B		Grupo C		Grupo D	
	média	dv	média	dv	média	dv	média	dv
1	318,01	4,25	338,58	4,01	318,24	4,15	337,74	4,50
2	288,42	6,38	305,75	4,86	297,59	2,10	292,28	2,08
3	307,43	3,51	320,16	5,75	311,89	2,77	310,15	3,98
4	249,51	2,59	271,23	1,24	264,29	2,11	255,56	3,68
5	286,17	6,30	292,16	5,43	285,65	4,18	292,03	8,50
6	375,20	6,37	394,52	5,69	374,33	5,15	392,54	3,40
7	333,15	5,90	336,94	7,56	322,49	2,90	333,82	4,62
8	297,88	4,74	310,34	5,37	309,36	5,92	305,68	10,05
9	287,64	2,82	317,68	3,41	294,22	2,41	304,92	2,67
10	334,18	5,78	350,08	7,93	334,87	6,32	342,96	5,60
11	302,53	4,57	319,97	6,95	317,46	6,32	304,79	5,53
12	272,62	3,68	296,58	6,50	287,04	7,05	277,61	7,15
13	362,77	6,60	370,82	5,35	364,82	7,07	376,76	6,06
14	339,45	3,71	349,40	6,79	341,57	6,12	345,11	6,69
15	300,66	5,91	321,17	8,29	300,90	4,76	306,78	4,93
16	334,75	4,65	349,83	7,85	342,46	7,37	348,01	5,69
17	332,99	5,64	345,17	6,03	345,54	6,07	339,43	6,23
18	300,54	5,91	315,75	6,70	311,36	3,92	308,40	8,10
19	362,95	5,13	369,63	5,35	365,83	6,42	370,71	6,70
20	354,08	10,58	356,47	3,89	358,36	4,62	357,63	5,94
21	322,54	4,42	344,24	5,12	335,76	6,54	335,96	5,50
22	361,30	4,38	378,74	5,98	368,75	3,23	370,54	6,10
23	354,83	5,73	371,15	6,67	359,92	6,87	364,66	7,03
24	390,52	6,49	411,63	5,17	397,52	6,59	399,06	6,67
<b>Média</b>	<b>323,76</b>	<b>5,25</b>	<b>339,08</b>	<b>5,75</b>	<b>329,59</b>	<b>5,04</b>	<b>332,21</b>	<b>5,73</b>

Em relação à influência das flexibilidades predominantes e seus respectivos níveis, em cada um dos vinte e quatro grupos de simulação, sobre a variável de resposta *flow time*, foi observado que:

- 1) Nos grupos onde não foi permitido o uso de flexibilidade de máquina, através do agrupamento de operações (grupos C e A), detectou-se uma influência direta e negativa da flexibilidade de seqüência nos seus níveis baixo e médio, destacando-se que esta influência foi percebida somente nos grupos formados por 6 ou 9 peças;
- 2) Além da flexibilidade de seqüência nos grupos C e A, não foi observada nenhuma outra influência direta entre as flexibilidades predominantes e seus respectivos níveis, em cada um dos vinte e quatro grupos de simulação, sobre a variável *flow time*.

---

As figuras 7.8 a 7.11 apresentam as médias da variável de resposta *flow time* obtidas para cada um dos 24 grupos de simulação, divididos em relação ao número de peças que os compõem (entre 6 e 18 peças), independentemente da flexibilidade predominante em cada grupo.

Observa-se novamente nestas figuras, que ao aumentar-se o número de peças que compõem o grupo, menor é o valor da média da variável *flow time* obtida em relação aos resultados apresentados por KIM et al. (2003) e POTTER (1997), em especial em grupos compostos a partir de 9 peças. Ou seja, ao se aumentar o número de peças que compõem um grupo de simulação, a abordagem baseada em agentes, sugerida neste trabalho, apresenta melhores resultados que as abordagens baseadas em algoritmos genéticos sugeridas por KIM et al. (2003) e POTTER (1997).

---

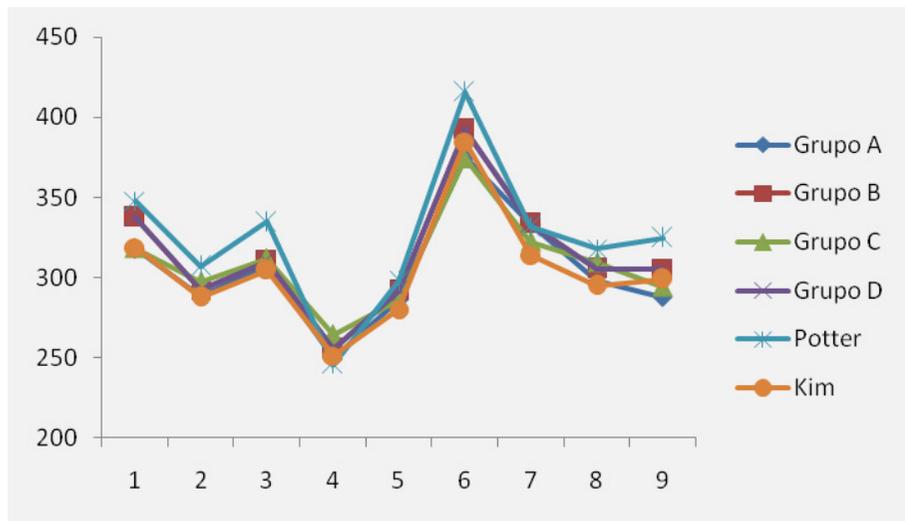


Figura 7.8 – Flow time dos grupos formados por 6 peças

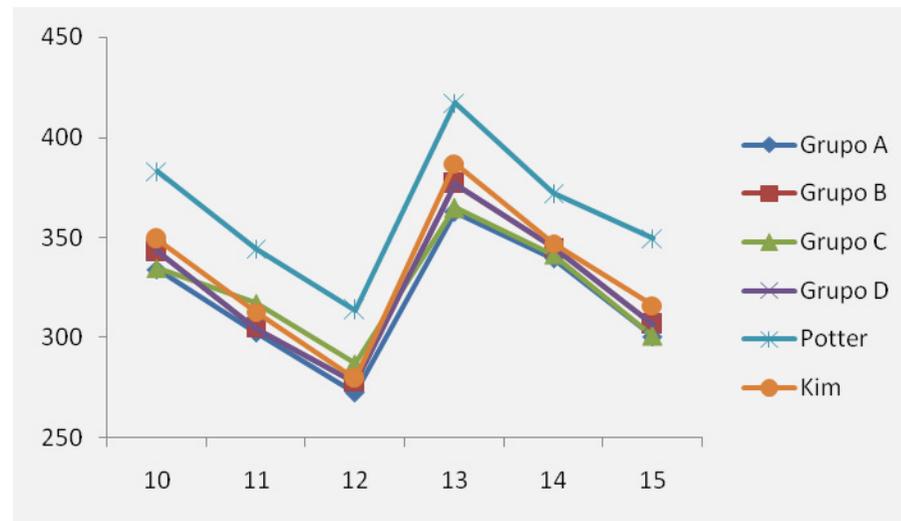


Figura 7.10 – Flow time dos grupos formados por 12 peças

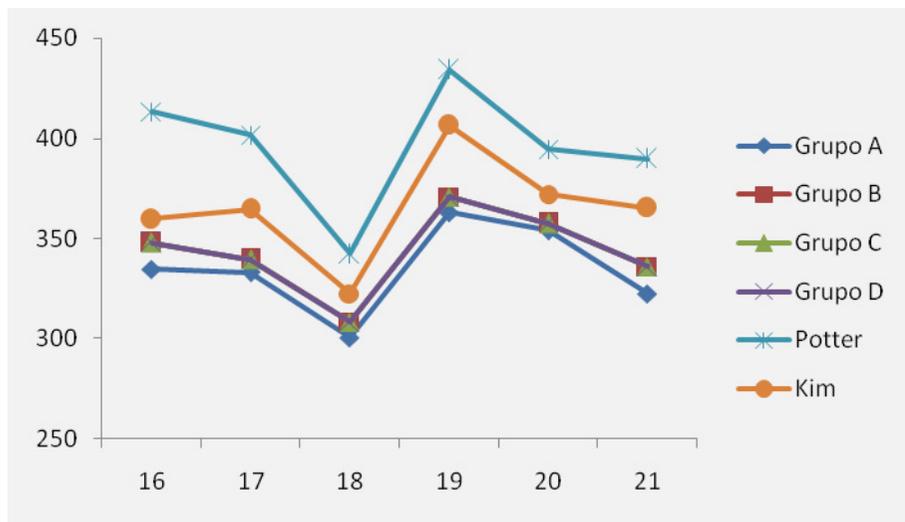


Figura 7.9 – Flow time dos grupos formados por 9 peças

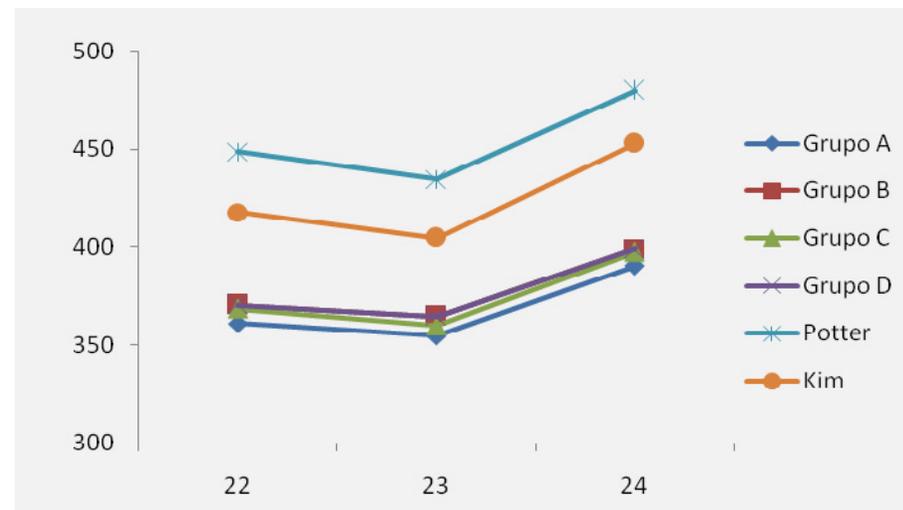


Figura 7.11 – Flow time dos grupos formados por 15 ou 18 peças

### 7.5 Análise dos resultados para as variáveis de resposta *makespan* e *flow time* utilizando seqüências dependentes do setup

Segundo FREITAS (2001) uma comparação de diferentes rodadas de simulação pode ser usada para avaliar o efeito de mudanças nas variáveis de controle, sendo que desta forma os efeitos podem ser medidos diretamente sobre as variáveis de resposta.

Dentro deste contexto, e a fim de verificar a consistência dos resultados obtidos pelo modelo proposto, em especial no que diz respeito à influência dos níveis de flexibilidades predominantes sobre os diversos grupos de simulação, foram criados 96 novos grupos a partir do caso de teste proposto por YEO et al. (2003).

Desta forma, foram realizadas 20 simulações para cada um dos 24 grupos de peças já apresentados na tabela 6.7 do presente trabalho, em cada um dos quatro níveis de *setup* de fixação e preparação, conforme apresentado na tabela 7.1.

Os níveis de *setup* 3 e 4, nos quais os *setups* de preparação são de 50% e 70% respectivamente, foram escolhidos a partir da observação informal dos valores de *setup* utilizados na indústria. Os demais níveis foram criados somente para possibilitar a conferência da confiabilidade dos dados através do uso de uma linha de tendência com quatro pontos.

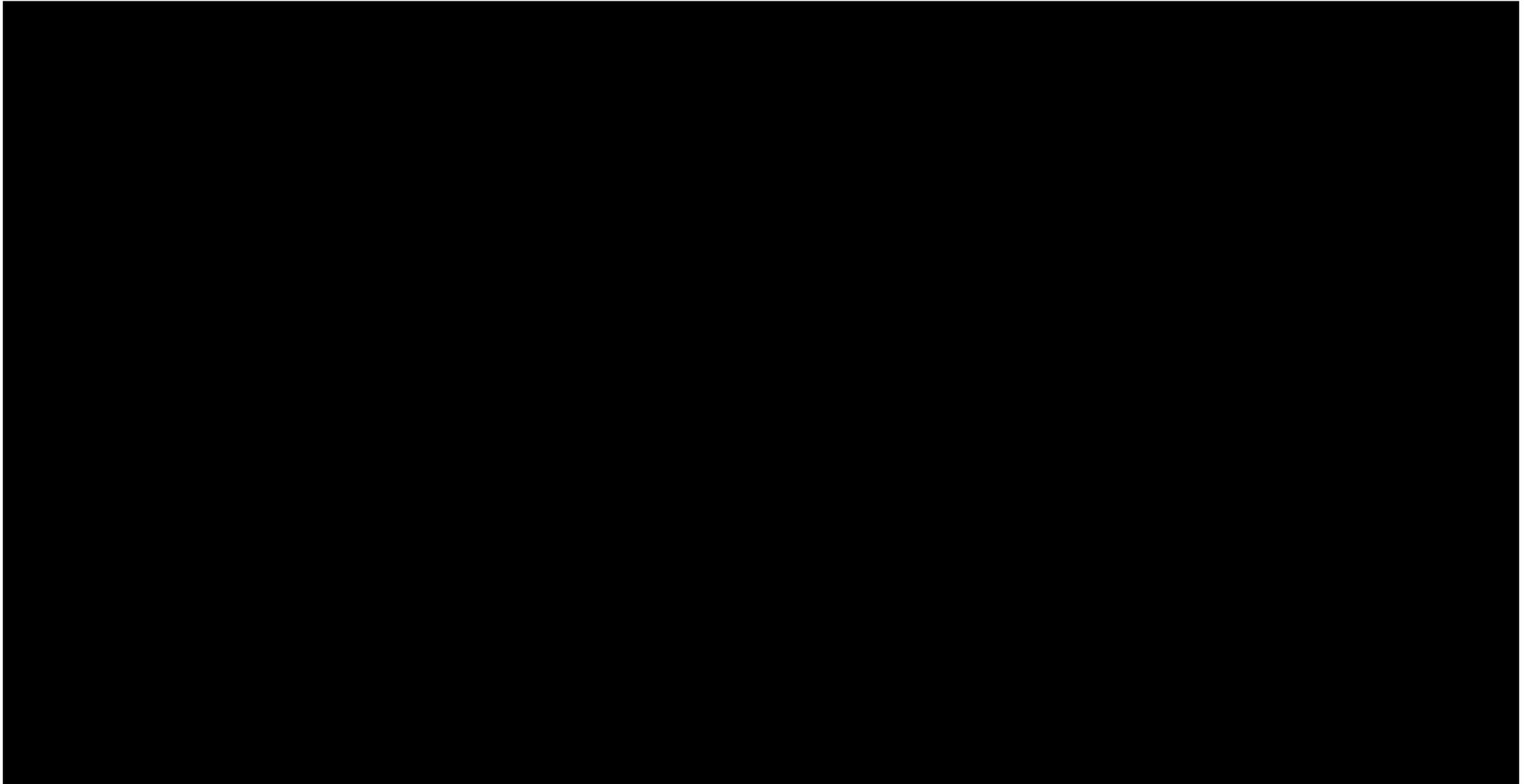
A tabela 7.13 apresenta a média e a média das médias, obtidas para a variável de resposta *makespan* nos grupos de flexibilidade C e D para as diferentes combinações de *setup* citadas acima.

Na análise dos resultados obtidos para o grupo C, que permite as flexibilidades de operação, seqüência e máquina, pode-se notar que as tendências observadas anteriormente para o grupo C com *setup* independente, ou seja, com valor de *setup* de preparação e fixação igual a zero, permanecem válidas para todas as combinações de *setup* estudadas, isto é:

- 1) Número de peças: permanece uma discreta tendência de melhores resultados para a variável de resposta *makespan* em grupos formados por um maior número de peças, situado entre 12 e 18, embora seja possível perceber que os grupos formados por 9 peças também começam a apresentar bons resultados; e
  - 2) Flexibilidades predominantes: a principal característica predominante é a flexibilidade de seqüência, com seu nível variando entre baixo e médio, independentemente da combinação de *setups* que é utilizada em sua composição. Estes grupos apresentam os piores resultados para a variável *makespan*. Observa-se, ainda, que a flexibilidade de operações com seu nível variando entre médio e alto continua influenciando negativamente os valores para a variável *makespan*, em especial os grupos formados por até 9 peças.
-

---

Tabela 7.13 - Comparação dos resultados de *makespan* entre os grupos C, D, Potter e Kim et. al, para os diversos *setups* simulados



---

Ao se analisar os resultados obtidos para o grupo D, que permite as flexibilidades de operação, seqüência, processo e máquina, é possível notar que as tendências observadas anteriormente para o grupo D com *setup* independente, ou seja, com valor de *setup* de preparação e fixação igual a zero, também permanecem válidas para todas as combinações de *setup* estudadas, isto é:

- 1) Número de peças: existe uma discreta tendência de melhores resultados para a variável de resposta *makespan* em grupos formados por um maior número de peças, situado entre 12 e 18; e
- 2) Flexibilidades predominantes: a principal característica observada é a influência da flexibilidade de seqüência em seus níveis baixo e médio. Nestes grupos, se encontram os piores resultados para a variável *makespan*. Menos perceptível, mas ainda significativa, é a influência da flexibilidade de operações com seu nível variando entre médio e alto, e da flexibilidade de processo com nível variando entre baixo e médio.

A tabela 7.14 apresenta a média das médias obtidas para as variáveis de resposta *makespan* nos grupos de flexibilidade A e B para as diferentes combinações de *setup* citadas anteriormente.

Na análise dos valores de *makespan* obtidos através do uso do grupo A, o qual permite somente as flexibilidades de operações e seqüência, ou seja, desconsidera o uso das flexibilidades de processo e máquina, pode-se observar que:

- 1) Número de peças: persiste a tendência de melhores resultados para a variável de resposta *makespan* em grupos formados por um maior número de peças, em especial aqueles situados entre 12 e 18; e
- 2) Flexibilidades predominantes: permanece a tendência dos grupos em que a característica predominante é a flexibilidade de seqüência com seu nível variando entre baixo e médio apresentarem os piores resultados para a variável *makespan* independentemente da combinação de *setups* utilizada em sua composição.

Na comparação dos resultados obtidos através do uso do grupo B, o qual permite o uso das flexibilidades de operações, seqüência e máquina, desconsiderando-se o uso da flexibilidade de processo, nota-se que:

---

---

Tabela 7.14 - Comparação dos resultados de *makespan* entre os grupos A, B, Potter e Kim et. al, para os diversos *setups* simulados

A large black rectangular area representing a redacted table. The table content is completely obscured by a solid black fill.

- 1) Número de peças: persiste a tendência de melhores resultados para a variável de resposta *makespan* em grupos formados por um maior número de peças, situados entre 12 e 18; e
- 2) Flexibilidades predominantes: nos grupos em que a característica predominante é a flexibilidade de seqüência nos níveis baixo e médio encontram-se os piores resultados para a variável *makespan*, independentemente da combinação de *setups* utilizada em sua composição.

A tabela 7.15 mostra a média das médias obtidas para a variável de resposta *flow time* nos grupos de flexibilidade C e D para as diferentes combinações de *setup* citadas anteriormente. Observa-se nesta tabela que os resultados obtidos para o grupo C, que permite as flexibilidades de operação, seqüência e máquina, indicam as seguintes conclusões:

- 1) Número de peças: uma clara tendência de melhores resultados para a variável de resposta *flow time* em grupos formados por um maior número de peças. Porém, diferentemente dos resultados obtidos com o *setup* igual a zero, observa-se que os ganhos na variável *flow time* já são significativos em grupos a partir de 9 peças;
- 2) Flexibilidades predominantes: os piores resultados para a variável de resposta *flow time* encontram-se nos grupos em que a característica predominante é a flexibilidade de processo, com seu nível variando entre médio e alto, ou a flexibilidade de seqüência, com seus níveis variando entre baixo e médio, especificamente em grupos compostos por até nove peças. Também foi observada a influência negativa da flexibilidade de operações nos níveis baixo e médio em grupos de até nove peças.

Ainda analisando os resultados da tabela 7.15, porém aqueles referentes ao grupo D, que permite as flexibilidades de operação, seqüência, processo e máquina, observa-se que:

- 1) Número de peças: fica evidente que, ao permitir-se o uso das quatro flexibilidades nos planos de processo, os grupos formados por um maior número de peças, especificamente a partir de nove apresentam os melhores resultados; e
- 2) Flexibilidades predominantes: ao contrário dos resultados obtidos com o uso do *setup* igual a zero, onde nenhuma influência relativa aos níveis de flexibilidade foi encontrada, os piores resultados para a variável de resposta *flow time* encontram-se nos grupos em que a característica predominante é a flexibilidade de processo, com seu nível variando entre médio e alto, ou a flexibilidade de seqüência, com seus níveis variando entre baixo e médio, especificamente em grupos compostos em até nove peças. Também foi observada a influência negativa da flexibilidade de operações nos níveis baixo e médio em grupos de até nove peças.

Tabela 7.15 - Comparação dos resultados de *flow time* entre os grupos C, D, Potter e Kim et. al, para os diversos *setups* simulados

Grupos	Grupo C		Grupo D		Potter	dv	Kim et. al	dv	Nível de flexibilidade predominante												
	0.1_0.1	0.1_0.1	0.1_0.1	0.1_0.1	0.3_0.1	0.3_0.1	0.3_0.1	0.3_0.1	0.5_0.1	0.5_0.1	0.5_0.1	0.5_0.1	0.7_0.1	0.7_0.1	0.7_0.1	0.7_0.1	(1997)		(2003)		
1	318,15	5,09	323,94	4,20	317,65	4,16	312,85	3,90	316,70	5,51	301,36	4,04	318,65	4,95	286,17	2,67	470,40	4,50	437,60	10,90	Baixo FP
2	286,70	1,79	298,54	1,54	289,49	5,50	277,41	1,50	288,71	6,24	263,60	5,77	286,23	3,72	249,60	2,83	369,20	5,50	349,70	5,90	Médio FP
3	309,62	7,09	306,01	6,13	307,68	4,01	295,26	2,84	307,92	4,10	283,64	3,86	306,74	1,97	271,96	4,88	382,00	4,20	355,20	7,40	Alto FP
4	249,14	2,21	266,47	3,17	249,22	1,69	258,49	4,61	249,41	3,07	251,21	5,79	249,47	3,19	250,91	9,07	321,50	2,50	306,20	0,40	Baixo FS
5	285,05	4,03	283,69	5,62	286,51	3,71	271,62	4,24	282,72	4,47	265,25	3,82	284,84	6,66	250,05	9,79	337,80	3,90	323,70	3,60	Médio FS
6	372,51	5,78	371,20	4,68	374,59	4,20	351,12	4,74	374,31	4,92	334,03	5,93	371,89	4,13	309,95	4,00	485,60	7,20	443,80	5,00	Alto FS
7	330,15	6,51	328,83	4,80	329,49	5,84	323,38	7,93	329,98	4,04	315,54	6,57	327,28	5,41	299,38	8,71	385,60	5,50	372,40	1,30	Baixo FO
8	297,17	5,54	304,44	4,50	301,74	5,56	293,69	8,07	301,80	6,75	273,94	5,30	300,58	7,14	261,33	7,18	373,80	6,10	348,30	5,70	Médio FO
9	287,12	2,26	297,99	2,16	285,79	2,50	280,82	5,79	287,06	1,76	260,43	2,94	286,67	4,42	242,20	7,16	474,50	6,20	434,90	9,80	Alto FO
10	332,50	4,76	335,87	6,98	335,60	6,87	320,87	3,87	333,11	3,77	311,67	4,44	331,37	4,83	298,70	9,44	502,60	7,00	456,50	10,80	Baixo ou Médio FP
11	305,48	5,64	307,32	5,46	305,72	6,14	299,18	7,81	304,29	4,14	289,81	10,68	303,10	8,18	275,84	5,43	423,80	4,60	378,90	5,10	Médio ou Alto FP
12	273,07	4,51	283,11	6,83	273,32	4,99	277,60	7,44	273,05	3,51	265,60	5,64	274,72	4,53	254,73	7,17	379,50	6,50	332,80	3,40	Baixo ou Médio FS
13	366,06	5,92	355,74	5,88	363,15	7,47	343,70	5,30	360,80	5,19	333,92	5,70	364,85	8,61	308,02	8,04	511,10	7,50	469,00	10,70	Médio ou Alto FS
14	342,71	5,13	339,63	7,36	341,45	6,66	333,37	6,96	343,99	7,06	323,90	10,46	339,10	4,83	306,53	11,66	433,40	5,10	402,40	10,60	Baixo ou Médio FO
15	296,09	3,34	304,62	6,39	299,72	6,61	284,96	4,31	298,73	4,99	267,35	5,67	295,32	5,03	254,09	10,25	493,90	6,50	445,20	11,00	Médio ou Alto FO
16	333,98	5,04	339,40	4,16	333,84	5,75	327,27	11,09	334,33	4,22	314,32	6,81	337,52	5,70	299,96	9,57	549,70	8,70	478,80	12,00	Baixo ou Médio FP
17	331,38	5,27	335,87	9,14	330,33	4,15	319,77	7,16	332,80	6,23	304,27	5,92	331,11	5,78	293,17	5,60	496,90	7,90	448,90	8,70	Médio ou Alto FP
18	303,56	5,26	308,91	6,43	303,70	4,79	301,22	4,58	302,37	4,93	288,82	7,53	302,58	4,41	277,26	8,09	419,80	7,30	389,60	7,50	Baixo ou Médio FS
19	361,55	6,20	357,84	6,43	367,87	4,83	342,08	5,66	367,13	5,52	332,84	7,96	370,19	6,40	309,04	8,12	557,00	8,40	508,10	10,00	Médio ou Alto FS
20	349,53	6,62	348,88	6,43	351,35	3,32	339,25	8,42	352,74	6,72	326,91	6,61	349,19	5,87	310,04	5,18	482,70	7,90	453,80	5,20	Baixo ou Médio FO
21	322,74	3,45	328,40	6,86	325,28	7,46	316,67	4,01	324,16	4,46	299,19	8,57	325,31	4,57	284,21	6,85	534,00	7,80	483,20	6,80	Médio ou Alto FO
22	357,34	3,84	364,54	7,00	361,65	6,47	351,11	7,03	364,47	4,42	339,82	6,28	363,84	6,48	324,61	7,40	587,50	7,60	548,30	6,90	-----
23	357,75	5,29	355,60	5,54	356,17	8,01	343,62	10,44	358,68	6,49	332,49	7,98	351,76	5,68	317,36	7,33	557,90	8,30	507,50	8,30	-----
24	393,79	8,48	393,78	7,20	393,68	5,39	378,21	6,02	392,14	3,21	372,78	8,43	393,80	4,20	351,94	14,17	633,30	8,70	602,20	7,10	-----
<b>Média</b>	<b>323,46</b>	<b>4,96</b>	<b>326,69</b>	<b>5,62</b>	<b>324,37</b>	<b>5,25</b>	<b>314,31</b>	<b>5,99</b>	<b>324,22</b>	<b>4,82</b>	<b>302,20</b>	<b>6,36</b>	<b>323,59</b>	<b>5,28</b>	<b>286,96</b>	<b>7,52</b>	<b>465,10</b>	<b>6,48</b>	<b>428,21</b>	<b>7,25</b>	

---

A tabela 7.16 apresenta a média das médias obtidas para as variáveis de resposta *flow time* nos grupos de flexibilidade A e B para as diferentes combinações de *setup* citadas anteriormente. Quando se analisa os valores de *makespan* obtidos através do uso do grupo A, o qual permite somente as flexibilidades de operações e seqüência, ou seja, desconsidera o uso das flexibilidades de processo e máquina, nota-se que:

- 1) Número de peças: os maiores ganhos não se concentram somente nos grupos formados por um grande número de peças, mas também nos grupos a partir de 6 peças, embora sejam mais significativos nos grupos formados por 9 peças ou mais; e
- 2) Flexibilidades predominantes: os piores resultados para a variável de resposta *flow time* encontram-se nos grupos formados por até 9 peças, em que a característica predominante é a flexibilidade de seqüência com seu nível variando entre baixo e médio.

E finalmente, ao serem analisados os resultados obtidos através do uso do grupo B da tabela 7.16, o qual permite o uso das flexibilidades de operações, seqüência e máquina, desconsiderando o uso da flexibilidade de processo, observa-se que:

- 1) Número de peças: os maiores ganhos não se concentram somente nos grupos formados por um grande número de peças, mas também nos grupos a partir de 6 peças, embora esses ganhos sejam mais significativos nos grupos formados por 9 peças ou mais; e
  - 2) Flexibilidades predominantes: não há a influência predominante de nenhum tipo ou nível de flexibilidade.
-

Tabela 7.16 - Comparação dos resultados de *flow time* entre os grupos A, B, Potter e Kim et. al, para os diversos setups simulados

Grupos	Grupo A		dv		Grupo B		dv		Grupo A		dv		Grupo B		dv		Potter	dv	Kim et. al	dv	Nível de flexibilidade predominante
	0.1_0.1	0.1_0.1	0.1_0.1	0.1_0.1	0.3_0.1	0.3_0.1	0.3_0.1	0.3_0.1	0.5_0.1	0.5_0.1	0.5_0.1	0.5_0.1	0.7_0.1	0.7_0.1	0.7_0.1	0.7_0.1	(1997)		(2003)		
1	317,84	5,62	320,17	5,06	320,33	5,55	309,72	5,38	317,83	4,62	303,03	7,24	317,03	4,61	283,35	4,51	470,40	4,50	437,60	10,90	-----
2	298,53	3,12	284,64	2,40	300,43	5,84	271,35	3,30	298,58	5,53	258,35	7,12	297,98	4,51	246,79	3,93	369,20	5,50	349,70	5,90	-----
3	310,12	4,24	306,73	6,21	310,68	5,22	298,43	5,33	309,74	3,93	286,80	3,54	310,78	3,38	278,15	5,91	382,00	4,20	355,20	7,40	-----
4	265,68	3,04	250,41	1,23	263,93	1,29	242,89	1,40	265,33	1,84	237,73	0,99	262,81	2,83	230,28	1,86	321,50	2,50	306,20	0,40	Baixo FS
5	286,88	5,48	280,42	4,92	288,43	4,89	267,44	4,54	284,71	3,89	264,42	4,52	285,08	3,09	254,28	5,11	337,80	3,90	323,70	3,60	Médio FS
6	376,37	5,41	371,89	6,20	376,09	6,10	354,22	6,84	376,84	6,70	332,35	3,95	379,04	5,98	314,09	5,29	485,60	7,20	443,80	5,00	Alto FS
7	327,89	5,32	331,47	6,68	328,49	6,15	323,10	9,09	322,81	4,36	318,42	10,69	326,31	5,91	311,27	9,08	385,60	5,50	372,40	1,30	Baixo FO
8	310,89	3,95	298,37	6,28	307,08	5,20	287,87	4,44	307,76	5,15	271,78	5,31	307,60	4,25	260,28	5,47	373,80	6,10	348,30	5,70	Médio FO
9	297,41	3,60	291,72	2,83	297,64	3,71	277,28	5,38	297,66	3,41	259,94	3,89	296,18	3,69	242,83	4,73	474,50	6,20	434,90	9,80	Alto FO
10	338,66	4,98	331,89	7,17	338,59	7,26	317,32	4,76	340,04	6,31	307,50	4,73	334,94	4,85	291,33	6,50	502,60	7,00	456,50	10,80	-----
11	316,71	6,25	305,60	8,37	316,04	10,79	293,65	6,87	319,03	6,82	288,20	6,98	317,60	3,67	277,49	6,15	423,80	4,60	378,90	5,10	-----
12	286,06	4,90	274,13	3,50	288,51	5,78	260,25	6,27	285,13	6,25	258,02	7,17	282,73	6,32	246,94	8,52	379,50	6,50	332,80	3,40	Baixo ou Médio FS
13	367,37	3,49	360,08	6,93	363,49	4,11	346,39	4,43	365,21	5,98	327,33	4,36	365,59	7,40	315,72	7,68	511,10	7,50	469,00	10,70	Médio ou Alto FS
14	343,11	7,83	341,93	10,83	345,26	7,77	333,51	6,23	343,14	7,38	327,78	5,73	341,24	5,13	319,36	8,48	433,40	5,10	402,40	10,60	Baixo ou Médio FO
15	307,30	4,04	297,63	3,91	306,15	5,93	279,76	5,23	304,10	2,85	267,54	4,68	309,26	3,37	251,38	4,50	493,90	6,50	445,20	11,00	Médio ou Alto FO
16	341,34	7,45	331,75	4,65	339,47	5,35	321,98	3,30	342,29	4,11	309,66	4,80	343,25	4,86	299,46	9,15	549,70	8,70	478,80	12,00	-----
17	340,25	5,63	324,34	9,72	340,52	3,84	318,36	5,25	340,47	4,25	313,81	6,64	342,48	4,52	297,07	8,72	496,90	7,90	448,90	8,70	-----
18	308,41	5,46	297,45	7,24	308,95	4,66	294,71	8,58	312,32	5,84	286,72	7,37	313,57	7,64	272,62	5,83	419,80	7,30	389,60	7,50	Baixo ou Médio FS
19	370,43	7,58	359,75	8,40	368,75	10,34	344,57	5,85	369,95	7,34	336,33	5,41	369,96	7,67	317,28	8,83	557,00	8,40	508,10	10,00	Médio ou Alto FS
20	356,80	8,54	349,59	8,10	356,80	5,34	339,83	10,05	354,75	9,56	331,59	6,93	350,64	5,27	319,43	10,65	482,70	7,90	453,80	5,20	Baixo ou Médio FO
21	332,98	5,06	324,30	4,54	332,54	6,24	311,39	6,93	332,22	4,08	295,37	5,68	334,67	5,80	284,19	7,28	534,00	7,80	483,20	6,80	Médio ou Alto FO
22	370,29	5,32	361,60	7,70	369,38	3,11	349,42	8,78	370,37	6,11	334,47	4,47	372,55	7,54	323,86	7,28	587,50	7,60	548,30	6,90	-----
23	367,15	2,40	352,92	4,56	359,48	8,32	345,74	4,34	361,13	8,98	337,65	8,13	365,67	7,75	323,17	9,71	557,90	8,30	507,50	8,30	-----
24	399,26	4,97	392,02	7,42	399,67	3,19	381,55	5,85	401,99	7,28	371,19	10,43	399,24	4,72	357,23	5,96	633,30	8,70	602,20	7,10	-----
<b>Média</b>	<b>330,74</b>	<b>5,15</b>	<b>322,53</b>	<b>6,04</b>	<b>330,28</b>	<b>5,67</b>	<b>311,28</b>	<b>5,77</b>	<b>330,14</b>	<b>5,52</b>	<b>301,08</b>	<b>5,87</b>	<b>330,26</b>	<b>5,20</b>	<b>288,24</b>	<b>6,71</b>	<b>465,10</b>	<b>6,48</b>	<b>428,21</b>	<b>7,25</b>	

### 7.6 Conclusão do capítulo

De acordo com CONWAY et al. (2003) a média da variável *flow time*, em um ambiente estático<sup>14</sup>, é diretamente influenciada pelo número de trabalhos no chão-de-fábrica, conforme demonstrado pela figura 7.12 e pelas equações 7.1 a 7.3, não sendo esta mesma relação válida para a variável de resposta *makespan*.

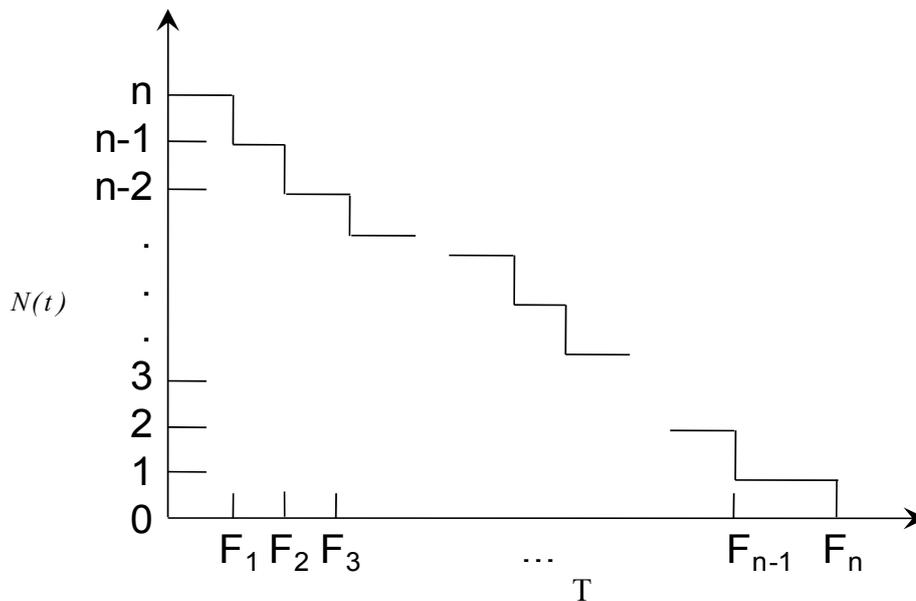


Figura 7.12 – Número de trabalhos pelo tempo em um problema envolvendo chegadas simultâneas (CONWAY, 1967)

$$\int_0^{F_n} N(t)dt = nF_1 + (n-1)(F_2 - F_1) + (n-2)(F_3 - F_2) + \dots + (n-(n-1))(F_n - F_{n-1}) = \sum_{i=1}^n F_i, \tag{7.1}$$

$$\bar{N}(0, F_{máx}) = \frac{\sum_{i=1}^n F_i}{F_{máx}} = n \frac{\bar{F}}{F_{máx}}, \tag{7.2}$$

$$\frac{\bar{N}(0, F_{máx})}{n} = \frac{\bar{F}}{F_{máx}} \tag{7.3}$$

Onde:

---

<sup>14</sup> CONWAY et al. (2003) define ambiente estático aquele onde todos os trabalhos são considerados disponíveis para agendamento simultaneamente no início do período de tempo considerado.

---

$\bar{N}(t)$  → número de trabalhos simultâneos no chão-de-fábrica<sup>15</sup>;

$n$  → número de trabalhos;

$\bar{F}$  → flow time médio

$F_{\text{máx}}$  → flow time máximo

$F_n$  → *flow time* relativo ao e-ésimo trabalho.

Ao observar o resultado obtido na equação 7.3, CONWAY et al. (2003) concluíram que a relação entre a média do número de trabalhos e o número máximo de trabalhos simultâneos em um dado chão-de-fábrica é igual à relação entre a média do *flow time* e seu valor máximo. Ou seja, para um dado intervalo de tempo, a média do número de trabalhos simultâneos no chão-de-fábrica é diretamente proporcional à média obtida para o *flow time* destes trabalhos.

Esta relação é válida para qualquer grupo de trabalhos para  $n \geq 1$  que iniciem em um dado instante, não sendo dependente do(a):

- a) Número, tipo ou arranjo dos recursos;
- b) Natureza dos trabalhos, suas rotas, tempos de processo, ou conhecimento prévio a respeito destes tempos;
- c) Atividades de processamento dos recursos; e
- d) Procedimento de agendamento.

Desta forma, a diferença pouco significativa entre a média das médias da variável de resposta *flow time* obtida pelo modelo proposto neste trabalho e por KIM et al. (2003) e POTTER(1997), para os grupos cujo tempo de *setup* foi considerado independente, ou seja igual a zero, pode ser explicada a partir do fato de que o número de trabalhos no chão de fábrica, por grupo de simulação, manteve-se o mesmo nas três diferentes abordagens. Estes resultados validam novamente os resultados já apresentados e discutidos no item 7.4 do presente trabalho.

Dando continuidade à discussão dos resultados apresentados para a variável de resposta *flow time*, serão discutidos agora aqueles dados obtidos através da utilização de seqüências dependentes do *setup*, dados que foram apresentados no item 7.5 do presente trabalho.

CONWAY et al. (2003) explicam que a redução do máximo tempo de fluxo (*flow time*) está associada à redução da soma dos  $n$  tempos de *setup*, conforme já citado no item 4.1.3 do presente trabalho. Assim, ao se comparar os dados relativos ao *flow time* com tempo de *setup*

---

<sup>15</sup> Notação utilizada para o inventário de trabalhos em processamento (*work in process inventory*) (CONWAY, 1967).

---

---

igual a zero àqueles obtidos para diferentes níveis de *setup*, é possível observar que, conforme previsto, os resultados para as seqüências dependentes do *setup* se apresentam menores do que aqueles onde o *setup* não foi considerado, com ganhos significativos nos grupos B e D. As tabelas 7.17 a 7.20 apresentam os resultados para a variável de resposta *flow time* para cada um dos quatro grupos de simulação, comparando os valores resultantes do uso ou não de seqüências dependentes do *setup*.

Estes resultados se repetem para todos os grupos de simulação, à exceção do grupo de simulação A, que considera somente as flexibilidades de operação e seqüência. A principal causa para este comportamento reside no fato de que este grupo não permite o uso de flexibilidade de máquina, ou seja, não utiliza o agrupamento de operações que poderiam gerar ganhos nos tempos de *setup* de preparação e fixação.

O mesmo raciocínio quando aplicado ao grupo C, que permite somente o uso das flexibilidades de operação, seqüência e processo, resulta em uma ligeira melhora na média das médias obtida para a variável *flow time* quando do uso de seqüências dependentes do *setup*, ao contrário do previsto na literatura, que as médias se mantivessem as mesmas pois o número de trabalhos no chão-de-fábrica permaneceu igual.

Em relação à variável *makespan*, PINEDO (1995) explica que para problemas de agendamento em uma única máquina, em que todas as datas de chegada ( $r_i$ ) sejam iguais a zero e sem seqüências dependentes do *setup*, o *makespan* será igual à soma dos tempos de processamento ( $\sum p_{i,m}$ ). Por outro lado, quando existe a seqüência dependente do *setup*, o mesmo *makespan* passa a ser dependente da forma de agendamento adotada.

Falar no uso de seqüências dependentes do *setup* para a programação de uma única máquina pode parecer inadequado em um modelo que considera múltiplos recursos trabalhando de forma concomitante. PINEDO (1995) explica que os resultados obtidos para um modelo de máquina única não provêm conhecimento apenas para ambientes desta natureza, mas também resultam em heurísticas para ambientes com combinações de recursos mais complexas.

Na prática, ainda conforme PINEDO (1995), os problemas de agendamento em ambientes complexos são freqüentemente decompostos em sub-problemas, na forma de problemas de agendamento em máquinas únicas.

Desta forma ao se comparar os dados relativos ao *makespan* com tempo de *setup* igual a zero, àqueles obtidos para diferentes níveis de *setup*, é possível observar que conforme previsto, os resultados para as seqüências dependentes do *setup* se apresentam melhores do que aqueles onde o *setup* não foi considerado, com ganhos significativos em todos os grupos.

---

Tabela 7.17 – Comparação do *flow time* para o grupo A sem e com seqüência dependente do setup

Set	Grupo A									
	0_0	dv	0.1_0.1	dv	0.3_0.1	dv	0.5_0.1	dv	0.7_0.1	dv
1	318,01	4,25	317,84	5,62	320,33	5,55	317,83	4,62	317,03	4,61
2	288,42	6,38	298,53	3,12	300,43	5,84	298,58	5,53	297,98	4,51
3	307,43	3,51	310,12	4,24	310,68	5,22	309,74	3,93	310,78	3,38
4	249,51	2,59	265,68	3,04	263,93	1,29	265,33	1,84	262,81	2,83
5	286,17	6,30	286,88	5,48	288,43	4,89	284,71	3,89	285,08	3,09
6	375,20	6,37	376,37	5,41	376,09	6,10	376,84	6,70	379,04	5,98
7	333,15	5,90	327,89	5,32	328,49	6,15	322,81	4,36	326,31	5,91
8	297,88	4,74	310,89	3,95	307,08	5,20	307,76	5,15	307,60	4,25
9	287,64	2,82	297,41	3,60	297,64	3,71	297,66	3,41	296,18	3,69
10	334,18	5,78	338,66	4,98	338,59	7,26	340,04	6,31	334,94	4,85
11	302,53	4,57	316,71	6,25	316,04	10,79	319,03	6,82	317,60	3,67
12	272,62	3,68	286,06	4,90	288,51	5,78	285,13	6,25	282,73	6,32
13	362,77	6,60	367,37	3,49	363,49	4,11	365,21	5,98	365,59	7,40
14	339,45	3,71	343,11	7,83	345,26	7,77	343,14	7,38	341,24	5,13
15	300,66	5,91	307,30	4,04	306,15	5,93	304,10	2,85	309,26	3,37
16	334,75	4,65	341,34	7,45	339,47	5,35	342,29	4,11	343,25	4,86
17	332,99	5,64	340,25	5,63	340,52	3,84	340,47	4,25	342,48	4,52
18	300,54	5,91	308,41	5,46	308,95	4,66	312,32	5,84	313,57	7,64
19	362,95	5,13	370,43	7,58	368,75	10,34	369,95	7,34	369,96	7,67
20	354,08	10,58	356,80	8,54	356,80	5,34	354,75	9,56	350,64	5,27
21	322,54	4,42	332,98	5,06	332,54	6,24	332,22	4,08	334,67	5,80
22	361,30	4,38	370,29	5,32	369,38	3,11	370,37	6,11	372,55	7,54
23	354,83	5,73	367,15	2,40	359,48	8,32	361,13	8,98	365,67	7,75
24	390,52	6,49	399,26	4,97	399,67	3,19	401,99	7,28	399,24	4,72
<b>Média</b>	<b>323,76</b>	<b>5,25</b>	<b>330,74</b>	<b>5,15</b>	<b>330,28</b>	<b>5,67</b>	<b>330,14</b>	<b>5,52</b>	<b>330,26</b>	<b>5,20</b>

Tabela 7.18 – Comparação do *flow time* para o grupo B sem e com seqüência dependente do setup

Set	Grupo B									
	0_0	dv	0.1_0.1	dv	0.3_0.1	dv	0.5_0.1	dv	0.7_0.1	dv
1	338,58	4,01	320,17	5,06	309,72	5,38	303,03	7,24	283,35	4,51
2	305,75	4,86	284,64	2,40	271,35	3,30	258,35	7,12	246,79	3,93
3	320,16	5,75	306,73	6,21	298,43	5,33	286,80	3,54	278,15	5,91
4	271,23	1,24	250,41	1,23	242,89	1,40	237,73	0,99	230,28	1,86
5	292,16	5,43	280,42	4,92	267,44	4,54	264,42	4,52	254,28	5,11
6	394,52	5,69	371,89	6,20	354,22	6,84	332,35	3,95	314,09	5,29
7	336,94	7,56	331,47	6,68	323,10	9,09	318,42	10,69	311,27	9,08
8	310,34	5,37	298,37	6,28	287,87	4,44	271,78	5,31	260,28	5,47
9	317,68	3,41	291,72	2,83	277,28	5,38	259,94	3,89	242,83	4,73
10	350,08	7,93	331,89	7,17	317,32	4,76	307,50	4,73	291,33	6,50
11	319,97	6,95	305,60	8,37	293,65	6,87	288,20	6,98	277,49	6,15
12	296,58	6,50	274,13	3,50	260,25	6,27	258,02	7,17	246,94	8,52
13	370,82	5,35	360,08	6,93	346,39	4,43	327,33	4,36	315,72	7,68
14	349,40	6,79	341,93	10,83	333,51	6,23	327,78	5,73	319,36	8,48
15	321,17	8,29	297,63	3,91	279,76	5,23	267,54	4,68	251,38	4,50
16	349,83	7,85	331,75	4,65	321,98	3,30	309,66	4,80	299,46	9,15
17	345,17	6,03	324,34	9,72	318,36	5,25	313,81	6,64	297,07	8,72
18	315,75	6,70	297,45	7,24	294,71	8,58	286,72	7,37	272,62	5,83
19	369,63	5,35	359,75	8,40	344,57	5,85	336,33	5,41	317,28	8,83
20	356,47	3,89	349,59	8,10	339,83	10,05	331,59	6,93	319,43	10,65
21	344,24	5,12	324,30	4,54	311,39	6,93	295,37	5,68	284,19	7,28
22	378,74	5,98	361,60	7,70	349,42	8,78	334,47	4,47	323,86	7,28
23	371,15	6,67	352,92	4,56	345,74	4,34	337,65	8,13	323,17	9,71
24	411,63	5,17	392,02	7,42	381,55	5,85	371,19	10,43	357,23	5,96
<b>Média</b>	<b>339,08</b>	<b>5,75</b>	<b>322,53</b>	<b>6,04</b>	<b>311,28</b>	<b>5,77</b>	<b>301,08</b>	<b>5,87</b>	<b>288,24</b>	<b>6,71</b>

Tabela 7.19 – Comparação do *flow time* para o grupo C sem e com seqüência dependente do setup

Set	Grupo C									
	0_0	dv	0.1_0.1	dv	0.3_0.1	dv	0.5_0.1	dv	0.7_0.1	dv
1	318,24	4,15	318,15	5,09	317,65	4,16	316,70	5,51	318,65	4,95
2	297,59	2,10	286,70	1,79	289,49	5,50	288,71	6,24	286,23	3,72
3	311,89	2,77	309,62	7,09	307,68	4,01	307,92	4,10	306,74	1,97
4	264,29	2,11	249,14	2,21	249,22	1,69	249,41	3,07	249,47	3,19
5	285,65	4,18	285,05	4,03	286,51	3,71	282,72	4,47	284,84	6,66
6	374,33	5,15	372,51	5,78	374,59	4,20	374,31	4,92	371,89	4,13
7	322,49	2,90	330,15	6,51	329,49	5,84	329,98	4,04	327,28	5,41
8	309,36	5,92	297,17	5,54	301,74	5,56	301,80	6,75	300,58	7,14
9	294,22	2,41	287,12	2,26	285,79	2,50	287,06	1,76	286,67	4,42
10	334,87	6,32	332,50	4,76	335,60	6,87	333,11	3,77	331,37	4,83
11	317,46	6,32	305,48	5,64	305,72	6,14	304,29	4,14	303,10	8,18
12	287,04	7,05	273,07	4,51	273,32	4,99	273,05	3,51	274,72	4,53
13	364,82	7,07	366,06	5,92	363,15	7,47	360,80	5,19	364,85	8,61
14	341,57	6,12	342,71	5,13	341,45	6,66	343,99	7,06	339,10	4,83
15	300,90	4,76	296,09	3,34	299,72	6,61	298,73	4,99	295,32	5,03
16	342,46	7,37	333,98	5,04	333,84	5,75	334,33	4,22	337,52	5,70
17	345,54	6,07	331,38	5,27	330,33	4,15	332,80	6,23	331,11	5,78
18	311,36	3,92	303,56	5,26	303,70	4,79	302,37	4,93	302,58	4,41
19	365,83	6,42	361,55	6,20	367,87	4,83	367,13	5,52	370,19	6,40
20	358,36	4,62	349,53	6,62	351,35	3,32	352,74	6,72	349,19	5,87
21	335,76	6,54	322,74	3,45	325,28	7,46	324,16	4,46	325,31	4,57
22	368,75	3,23	357,34	3,84	361,65	6,47	364,47	4,42	363,84	6,48
23	359,92	6,87	357,75	5,29	356,17	8,01	358,68	6,49	351,76	5,68
24	397,52	6,59	393,79	8,48	393,68	5,39	392,14	3,21	393,80	4,20
<b>Média</b>	<b>329,59</b>	<b>5,04</b>	<b>323,46</b>	<b>4,96</b>	<b>324,37</b>	<b>5,25</b>	<b>324,22</b>	<b>4,82</b>	<b>323,59</b>	<b>5,28</b>

Tabela 7.20 – Comparação do *flow time* para o grupo D sem e com seqüência dependente do setup

Set	Grupo D									
	0_0	dv	0.1_0.1	dv	0.3_0.1	dv	0.5_0.1	dv	0.7_0.1	dv
1	337,74	4,50	323,94	4,20	312,85	3,90	301,36	4,04	286,17	2,67
2	292,28	2,08	298,54	1,54	277,41	1,50	263,60	5,77	249,60	2,83
3	310,15	3,98	306,01	6,13	295,26	2,84	283,64	3,86	271,96	4,88
4	255,56	3,68	266,47	3,17	258,49	4,61	251,21	5,79	250,91	9,07
5	292,03	8,50	283,69	5,62	271,62	4,24	265,25	3,82	250,05	9,79
6	392,54	3,40	371,20	4,68	351,12	4,74	334,03	5,93	309,95	4,00
7	333,82	4,62	328,83	4,80	323,38	7,93	315,54	6,57	299,38	8,71
8	305,68	10,05	304,44	4,50	293,69	8,07	273,94	5,30	261,33	7,18
9	304,92	2,67	297,99	2,16	280,82	5,79	260,43	2,94	242,20	7,16
10	342,96	5,60	335,87	6,98	320,87	3,87	311,67	4,44	298,70	9,44
11	304,79	5,53	307,32	5,46	299,18	7,81	289,81	10,68	275,84	5,43
12	277,61	7,15	283,11	6,83	277,60	7,44	265,60	5,64	254,73	7,17
13	376,76	6,06	355,74	5,88	343,70	5,30	333,92	5,70	308,02	8,04
14	345,11	6,69	339,63	7,36	333,37	6,96	323,90	10,46	306,53	11,66
15	306,78	4,93	304,62	6,39	284,96	4,31	267,35	5,67	254,09	10,25
16	348,01	5,69	339,40	4,16	327,27	11,09	314,32	6,81	299,96	9,57
17	339,43	6,23	335,87	9,14	319,77	7,16	304,27	5,92	293,17	5,60
18	308,40	8,10	308,91	6,43	301,22	4,58	288,82	7,53	277,26	8,09
19	370,71	6,70	357,84	6,43	342,08	5,66	332,84	7,96	309,04	8,12
20	357,63	5,94	348,88	6,43	339,25	8,42	326,91	6,61	310,04	5,18
21	335,96	5,50	328,40	6,86	316,67	4,01	299,19	8,57	284,21	6,85
22	370,54	6,10	364,54	7,00	351,11	7,03	339,82	6,28	324,61	7,40
23	364,66	7,03	355,60	5,54	343,62	10,44	332,49	7,98	317,36	7,33
24	399,06	6,67	393,78	7,20	378,21	6,02	372,78	8,43	351,94	14,17
<b>Média</b>	<b>332,21</b>	<b>5,73</b>	<b>326,69</b>	<b>5,62</b>	<b>314,31</b>	<b>5,99</b>	<b>302,20</b>	<b>6,36</b>	<b>286,96</b>	<b>7,52</b>

---

As tabelas 7.21 a 7.24 apresentam os resultados para a variável de resposta *makespan* para cada um dos quatro grupos de simulação, comparando os valores resultantes do uso ou não de seqüências dependentes do *setup*. Novamente, os resultados obtidos para o grupo de simulação A, que considera somente as flexibilidades de operação e seqüência, apontam que não houve necessariamente uma redução dos valores do *makespan* com a alteração dos valores referentes ao tempo de *setup*.

---

Tabela 7.21 – Comparação do *makespan* para o grupo A sem e com seqüência dependente do setup

Set	Grupo A									
	0_0	dv	0.1_0.1	dv	0.3_0.1	dv	0.5_0.1	dv	0.7_0.1	dv
1	451,03	7,21	448,76	9,07	445,64	3,63	448,37	6,35	443,53	9,55
2	360,09	10,97	366,56	15,62	368,80	14,41	366,40	14,37	361,68	9,47
3	370,18	21,26	369,78	13,09	371,63	17,87	366,93	11,25	367,63	13,84
4	317,03	8,50	331,00	11,39	326,09	6,66	325,83	9,32	328,18	10,51
5	353,47	19,14	352,51	20,73	356,72	16,41	354,72	18,97	353,69	19,40
6	455,07	13,46	449,56	3,39	448,80	9,38	455,81	11,18	451,96	14,16
7	384,04	7,39	383,18	5,84	384,52	6,18	380,30	5,63	381,46	5,75
8	360,40	9,51	370,05	18,23	365,93	11,58	367,39	10,99	367,17	12,27
9	447,88	6,67	452,03	5,36	449,43	7,26	451,35	8,81	450,69	8,64
10	460,76	14,27	463,79	10,88	462,71	17,17	463,80	12,67	457,35	12,29
11	375,49	10,78	401,55	16,62	385,68	29,96	384,21	7,35	392,66	24,09
12	356,88	13,18	367,34	11,27	361,91	17,10	368,51	21,76	368,49	18,08
13	465,68	20,04	466,47	14,67	459,73	10,26	467,49	15,80	464,53	8,23
14	406,57	8,25	403,00	13,26	408,33	11,80	399,65	9,90	395,25	7,94
15	464,55	9,59	464,04	6,12	458,14	9,00	459,15	4,56	458,36	9,29
16	477,23	17,78	483,67	20,40	483,92	19,73	491,21	18,72	492,03	19,96
17	407,21	8,79	422,25	12,57	419,95	15,58	417,42	12,46	414,46	10,16
18	395,42	9,42	410,26	25,24	400,62	14,11	398,79	20,10	409,38	24,37
19	481,59	25,71	493,43	9,39	489,57	17,54	482,66	16,57	494,61	13,06
20	439,07	17,02	433,14	14,25	437,94	20,86	432,82	28,76	427,10	10,36
21	471,39	12,77	473,25	14,15	489,81	16,72	479,86	15,28	485,46	17,04
22	501,31	14,15	508,74	18,28	502,07	13,71	510,13	13,55	509,41	12,91
23	458,21	16,37	468,07	15,87	473,25	9,82	461,48	21,63	460,21	20,33
24	541,31	24,32	536,33	20,74	542,50	23,49	537,37	19,05	534,51	23,60
<b>Média</b>	<b>425,08</b>	<b>13,61</b>	<b>429,95</b>	<b>13,60</b>	<b>428,90</b>	<b>14,18</b>	<b>427,99</b>	<b>13,96</b>	<b>427,91</b>	<b>13,97</b>

Tabela 7.22 – Comparação do *makespan* para o grupo B sem e com seqüência dependente do setup

Set	Grupo B									
	0_0	dv	0.1_0.1	dv	0.3_0.1	dv	0.5_0.1	dv	0.7_0.1	dv
1	520,53	13,41	476,96	13,17	429,57	17,79	396,12	19,26	355,96	3,86
2	374,58	7,80	363,62	9,10	348,82	10,72	340,46	16,63	332,91	24,79
3	372,72	7,95	352,83	8,66	335,92	9,13	338,25	9,70	330,24	7,81
4	329,27	2,53	339,50	1,97	331,33	0,98	324,19	1,98	315,12	3,10
5	362,24	26,00	341,56	16,65	330,98	10,85	334,73	12,96	328,58	9,19
6	513,75	15,05	466,47	11,71	429,19	12,29	385,29	9,65	360,51	10,86
7	386,65	5,10	376,79	5,20	367,20	5,83	359,51	5,47	361,10	7,95
8	375,73	10,27	348,59	9,81	333,49	8,92	323,55	12,02	314,00	4,63
9	521,21	8,28	475,47	9,45	431,80	13,18	390,69	11,57	360,81	7,31
10	518,38	23,21	482,50	15,32	438,95	15,07	406,32	15,30	374,68	14,64
11	387,15	24,40	390,60	42,02	377,02	21,35	376,55	28,36	384,34	33,50
12	391,73	32,46	365,60	11,12	357,41	34,93	352,03	23,67	355,03	19,66
13	520,45	11,57	475,63	11,96	445,95	18,14	398,23	16,08	381,58	13,63
14	404,00	10,75	394,35	17,09	381,92	10,58	384,84	12,60	371,93	17,25
15	529,41	11,65	483,89	13,14	441,90	12,84	399,61	19,26	373,11	15,09
16	533,26	27,89	492,98	31,84	454,08	19,79	420,46	10,93	400,58	13,90
17	424,25	21,17	394,82	21,79	412,07	33,29	418,41	32,18	403,67	40,74
18	411,84	18,68	401,59	42,53	390,53	26,01	388,86	23,11	361,63	17,64
19	532,05	25,01	494,43	24,12	446,04	13,99	432,36	9,79	413,32	23,95
20	441,95	33,51	426,08	12,08	413,73	15,11	419,12	25,72	397,92	17,97
21	523,98	16,61	501,46	26,74	452,83	21,31	424,82	25,66	370,55	14,45
22	541,23	17,00	521,14	26,04	467,70	14,73	440,88	16,72	437,72	23,90
23	475,47	29,33	467,42	14,98	468,27	30,57	446,65	34,13	427,05	17,04
24	563,72	19,65	538,60	31,02	500,25	14,04	498,11	23,31	482,64	23,49
<b>Média</b>	<b>456,48</b>	<b>17,47</b>	<b>432,20</b>	<b>17,81</b>	<b>407,79</b>	<b>16,31</b>	<b>391,67</b>	<b>17,34</b>	<b>374,79</b>	<b>16,10</b>

Tabela 7.23 – Comparação do *makespan* para o grupo C sem e com seqüência dependente do setup

Set	Grupo C									
	0_0	dv	0.1_0.1	dv	0.3_0.1	dv	0.5_0.1	dv	0.7_0.1	dv
1	445,37	7,93	449,21	9,42	451,07	7,45	445,76	7,90	447,81	8,29
2	380,25	11,43	363,13	7,51	364,73	8,26	366,12	8,53	363,47	7,16
3	373,13	11,75	363,09	14,18	364,63	16,48	361,96	20,13	363,34	15,65
4	327,45	9,23	319,08	5,62	320,60	7,35	317,31	6,67	321,19	6,02
5	345,47	11,32	340,70	10,78	347,60	12,74	343,75	16,45	347,38	14,28
6	445,35	8,05	449,90	8,30	447,08	8,60	448,90	11,15	448,13	6,11
7	377,62	2,90	387,27	8,08	385,42	8,56	384,15	8,55	383,67	5,29
8	375,11	18,21	361,06	11,01	370,06	12,27	368,92	14,37	364,23	16,16
9	446,35	7,25	448,81	6,96	444,97	6,63	447,96	4,62	447,95	8,12
10	450,97	8,57	458,50	10,33	462,21	12,36	454,53	14,87	460,30	13,68
11	389,23	9,76	382,84	13,10	383,43	11,09	374,39	12,27	373,67	14,67
12	357,05	13,41	361,28	16,29	364,47	18,58	355,47	12,74	368,17	12,56
13	470,93	15,89	475,33	21,40	465,11	12,45	468,63	21,43	463,74	22,75
14	396,83	12,28	399,65	5,67	408,15	9,69	404,82	12,59	402,11	11,83
15	459,35	9,95	464,15	12,79	461,73	15,28	457,08	13,14	460,70	13,64
16	483,61	15,60	491,65	31,12	483,62	21,15	485,53	17,83	468,49	11,85
17	429,57	20,38	409,32	22,91	405,02	18,01	422,44	20,82	408,18	13,53
18	403,21	22,18	398,85	17,64	403,05	22,33	399,74	19,91	400,11	23,12
19	480,18	14,46	478,12	19,02	490,42	18,62	491,21	16,34	486,06	16,03
20	434,46	14,99	422,66	9,88	427,97	11,68	434,03	14,99	429,03	15,90
21	485,39	19,38	485,75	18,18	480,67	12,82	479,78	35,28	482,34	17,77
22	494,74	19,05	494,42	14,24	502,81	12,66	498,48	13,22	500,47	16,04
23	452,96	11,66	472,19	23,00	464,64	28,13	466,52	18,25	451,04	10,21
24	534,14	16,57	527,40	17,74	527,52	15,40	531,68	19,84	533,18	20,28
<b>Média</b>	<b>426,61</b>	<b>13,01</b>	<b>425,18</b>	<b>13,97</b>	<b>426,12</b>	<b>13,69</b>	<b>425,38</b>	<b>15,08</b>	<b>423,95</b>	<b>13,37</b>

Tabela 7.24 – Comparação do *makespan* para o grupo C sem e com seqüência dependente do setup

Set	Grupo D									
	0_0	dv	0.1_0.1	dv	0.3_0.1	dv	0.5_0.1	dv	0.7_0.1	dv
1	517,73	8,96	479,12	11,94	436,02	12,98	394,51	14,95	359,12	8,53
2	369,62	7,74	365,70	15,35	337,22	8,50	334,43	7,17	326,30	13,69
3	366,72	7,35	342,94	4,70	334,68	7,32	328,81	7,59	324,00	14,41
4	324,02	11,17	325,79	6,29	314,33	12,88	318,14	5,45	318,53	6,57
5	364,06	38,16	349,55	14,94	335,24	7,97	355,14	23,94	342,67	19,82
6	504,01	7,06	466,57	8,99	425,97	11,78	395,00	10,26	366,42	15,26
7	381,64	1,37	373,34	9,19	369,04	8,85	365,19	10,68	357,18	7,93
8	373,14	12,20	352,35	11,72	339,48	13,48	327,40	8,71	321,82	10,19
9	520,26	10,54	474,16	6,78	433,08	14,85	395,31	11,17	344,02	19,17
10	520,76	23,79	476,86	13,54	426,79	17,75	400,51	15,16	380,99	13,57
11	379,41	9,74	364,90	16,88	379,09	39,50	378,94	34,18	370,95	41,57
12	367,90	18,91	363,73	19,58	359,48	15,79	371,55	21,96	357,88	22,60
13	513,26	11,25	473,87	10,22	446,08	19,77	409,05	12,93	382,25	13,99
14	394,67	13,50	385,34	10,95	391,18	15,19	376,87	18,31	368,36	19,70
15	522,39	18,79	484,57	13,50	441,36	11,80	396,23	13,82	366,79	19,65
16	533,18	26,98	487,25	21,29	441,41	13,44	420,39	18,43	401,40	14,74
17	432,11	21,69	416,68	21,68	424,91	31,21	405,75	40,47	397,82	24,56
18	401,65	23,45	417,45	40,87	393,44	16,21	380,77	20,25	373,32	22,92
19	521,02	23,12	490,15	10,22	462,22	20,34	424,61	24,01	408,49	16,05
20	432,56	16,23	423,61	12,02	421,35	17,94	414,76	19,40	395,14	18,05
21	523,48	19,27	504,39	15,39	457,60	20,33	417,10	21,95	390,34	15,52
22	547,76	23,82	504,38	15,31	473,73	19,70	458,53	26,47	438,00	14,48
23	464,89	22,03	459,42	15,94	457,29	33,72	429,78	19,38	423,79	21,76
24	566,64	33,39	531,66	14,11	501,56	22,60	500,11	27,45	471,54	19,32
<b>Média</b>	<b>451,79</b>	<b>17,10</b>	<b>429,74</b>	<b>14,22</b>	<b>408,44</b>	<b>17,25</b>	<b>391,62</b>	<b>18,09</b>	<b>374,46</b>	<b>17,25</b>

---

## CAPÍTULO 8 – O USO DE SISTEMAS ESPECIALISTAS ASSOCIADOS A AGENTES

Este capítulo tem por finalidade apresentar a estrutura do banco de dados utilizado no modelo proposto, o qual foi desenvolvido em MySQL (2006), que é um sistema de gerenciamento de banco de dados que utiliza a linguagem de consulta estruturada SQL (*Structured Query Language*), a qual permite o desenvolvimento de bancos de dados relacionais.

Na seqüência é demonstrado como os dados gerados pelas simulações, já apresentados e discutidos no capítulo 7, e que foram armazenados neste banco de dados, são utilizados como fonte de conhecimento para a comparação entre novas simulações e casos pré-existentes através do uso de sistemas especialistas.

Dentro deste contexto, é feita uma breve revisão a respeito do uso de sistemas especialistas associados a agentes, bem como da escolha do *Shell* empregado.

Finalmente é descrita a implementação das novas funcionalidades no modelo proposto e discutidos os resultados obtidos.

### 8.1 Estrutura de dados

Neste trabalho a estrutura de armazenamento de dados possui dois objetivos distintos e complementares:

- a) Disponibilizar uma série de dados do modelo conceitual e arquitetura, capítulos 4 e 5 do presente trabalho, antes da instanciação dos agentes, a fim de permitir que os mesmos cumpram seus objetivos; e
- b) Armazenar os dados gerados pelos agentes durante o cumprimento destes objetivos, ou seja, os dados resultantes das simulações, disponibilizando-os para posterior análise, conforme apresentado e discutido no capítulo 7 do presente trabalho.

A estruturação de banco de dados de forma relacional foi desenvolvida por CODD (1970), também sendo conhecida como modelo de dados entidade-relacionamento. Segundo KORTH e SILBERSCHATZ (1995) baseia-se na percepção do mundo real, que neste caso, consiste em um conjunto de objetos denominados entidades e nos seus respectivos relacionamentos.

---

A escolha pela utilização do sistema de gerenciamento de banco de dados MySQL baseou-se principalmente no fato do mesmo ser um software *open source*, tendo sido utilizado e validado por milhares de usuários e amplamente difundido devido a fatores como: robustez, multi-tarefa, multi-usuário, consistência, alto desempenho, confiabilidade, compatibilidade, portabilidade e facilidade de uso (MySQL, 2006).

O banco de dados desenvolvido para o presente trabalho é composto por seis tabelas inter-relacionadas: *parts*, *resources*, *jobs*, *job\_stats*, *resource\_stats* e *simulation*, sendo que as três primeiras são responsáveis por disponibilizar aos agentes os dados definidos no modelo conceitual e arquitetura, capítulos 4 e 5 respectivamente; as demais tabelas, *job\_stats*, *resource\_stats* e *simulation*, por sua vez, são responsáveis por armazenar e disponibilizar os dados gerados pelo modelo de simulação. Cada uma destas tabelas será descrita a seguir:

a) Tabela *parts* – é responsável por armazenar a estrutura que compõe uma peça<sup>16</sup> e seu plano de processo<sup>17</sup>, sendo formada por um(a):

- a. Campo *type*, que identifica se a peça é do tipo prismática ou rotacional (no momento esta distinção não está sendo utilizada nas simulações);
- b. Campo número de identificação, chamado de *part\_id*, que deve ser único;
- c. Coluna de *precedences*, onde é armazenado o vetor *prec*, responsável por armazenar todas as operações que possuam precedências no plano de processo da peça;
- d. Coluna de operações OR, que descreve a possibilidade de operações alternativas, já descrito no item 4.1.9 do presente trabalho; e
- e. Coluna com o campo operação<sup>18</sup>, onde será explicitada cada uma das operações que compõem o plano de processo da peça, chamada de *operations*.

A figura 8.1 apresenta a estrutura da tabela *parts* representada a partir do exemplo da peça 9 utilizada no caso de testes, descrito anteriormente no capítulo 7 do presente trabalho. É importante lembrar, que conforme definido no capítulo 4, as relações de precedências indicam a seqüência exata na qual as operações que compõem o plano de processo de uma peça devem ser executadas, enquanto as alternativas (“OR”) indicam os caminhos alternativos para a manufatura de uma peça.

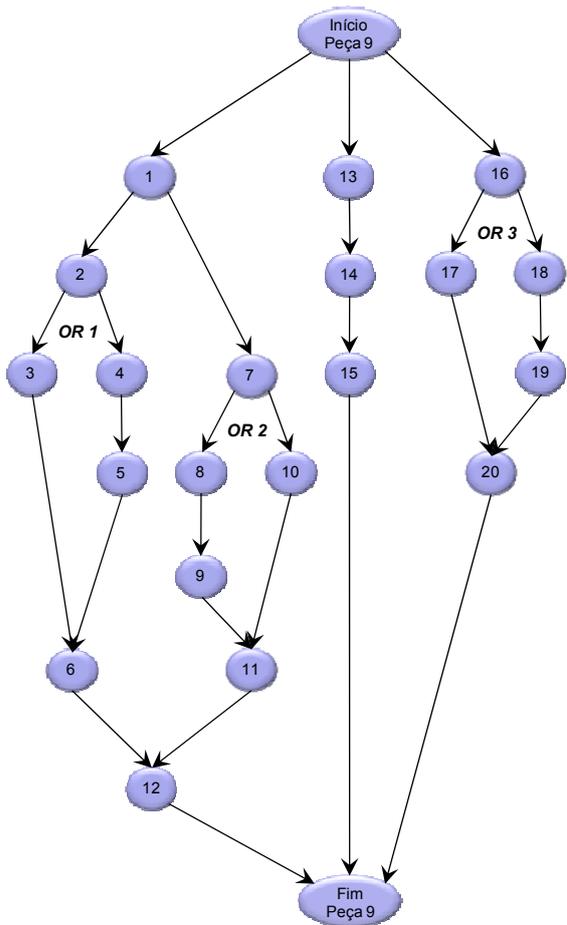
---

<sup>16</sup> Peça – elemento definido pelos atributos tipo (rotacional ou prismática); identificação, que deve ser única; e as operações que compõem seu plano de processo.

<sup>17</sup> Plano de processo – na visão do modelo é o conjunto de operações necessárias, com suas respectivas restrições e alternativas, para executar uma peça

<sup>18</sup> Operação – corresponde a uma unidade do plano de processo.

---



ID	9
Type	1
Prec	op2_9(op1_9) op3_9(op2_9) op4_9(op2_9) op5_9(op4_9) op6_9(op3_9*op5_9) op7_9(op1_9) op8_9(op7_9) op9_9(op8_9) op10_9(op7_9) op11_9(op9_9*op10_9) op12_9(op6_9,op11_9) op14_9(op13_9) op15_9(op14_9) op17_9(op16_9) op18_9(op16_9) op19_9(op18_9) op20_9(op17_9*op19_9)
Or	(op3_9,op4_9) (op8_9,op10_9) (op17_9,op18_9)
Operations	Op1_9 Op2_9 Op3_9 Op4_9 Op5_9 Op6_9 Op7_9 Op8_9 Op9_9 Op10_9 Op11_9 Op12_9 Op13_9 Op14_9 Op15_9 Op16_9 Op17_9 Op18_9 Op19_9 Op20_9

Figura 8.1 – Plano de processo da peça 9 com suas precedências e alternativas

b) Tabela *resources* – é responsável por armazenar dados sobre a capacidade dos recursos, sendo formada por um:

- a. Campo de identificação, chamado de *resource\_id*, que deve ser único;
- b. Campo de identificação, chamado de *name*, que deve ser único;
- c. Campo de custo, *cost* que possibilita o custo de processamento por unidade de tempo. É importante observar que este campo não está sendo utilizado neste trabalho, porém sua implementação esta completa para utilizações futuras;
- d. Campo de tempo de operações, *operation\_time*, que possui duas finalidades distintas: identificar a capacidade do recurso em termos de operações; e identificar o tempo de cada uma das operações associadas aquele recurso.

c) Tabela *jobs* – é responsável por criar um registro para cada ordem de produção<sup>19</sup> que entra no sistema, sendo formada por um:

<sup>19</sup> Ordem ou ordem de produção – é a menor unidade de um pedido, sendo que cada ordem corresponde a uma única peça.

- 
- a. Campo de identificação *job\_id*, que identifica cada ordem de produção e que deve ser único;
  - b. Um campo *part\_id*, que aponta para o *id* da tabela *parts*, e relaciona a peça correspondente a cada pedido;
  - c. Campo de tamanho de lote, *batchsize*, que identifica o tamanho de lote para cada ordem de produção; e
  - d. Campo *email*, que possibilita a identificação da pessoa que enviou a ordem de produção.
- d) Tabela *job\_stats* – é responsável por criar um registro para cada ordem de produção que sai do sistema, identificando os valores de saída da mesma e sendo formada por um:
- a. Um campo *job\_id* que aponta para o *id* da tabela *jobs*;
  - b. Campo *total\_time*, que corresponde ao tempo total em que a ordem de produção esteve em processamento, ou seja, é computada a diferença entre o tempo de entrada e saída da ordem no sistema (corresponde ao *flow time* de uma peça);
  - c. Campo *total\_machining\_time*, tempo total de processamento (nos recursos) da ordem de produção;
  - d. Campo *total\_queue\_time*, tempo total em que a ordem de produção esteve no aguardo do início de processamento na fila de um recurso;
  - e. Campo *total\_setup\_time*, tempo total de setup;
  - f. Campo *total\_cost*, custo total da ordem, computado a partir da soma dos custos de cada operação por recurso.

Observando que os campos de *total\_queue\_time* e *total\_cost*, apesar de implementados, não fazem parte do escopo de análise do presente trabalho.

- e) Tabela *resource\_stats* – é responsável por armazenar as informações relativas a uma ordem de produção sob o prisma da utilização de cada um dos recursos, apresentando um detalhamento dos dados estatísticos totais de uma dada ordem, sendo formada por um:
- a. Um campo *job\_id* que aponta para o *id* da tabela *jobs*;
  - b. Um campo *resource\_id* que aponta para o *id* da tabela *resource*;
  - c. Campo *contracted\_sequence*, representa todas as operações, de uma dada ordem, que fecharam contrato para serem executadas por recurso durante uma rodada de negociação. Detalhes sobre a forma de negociação e agrupamento de operações podem ser encontrados no capítulo 6 do presente trabalho;
-

- 
- d. Campo *done\_sequence*, representa as operações de uma dada ordem que efetivamente foram executadas por recurso em uma dada simulação.

O campo *done\_sequence* apesar de implementado e funcional, não faz parte do escopo de análise deste trabalho. Sua implementação visa no futuro, auxiliar no estudo de quebras de máquinas, sendo que na atual etapa de implementação do modelo os valores dos campos *contracted\_sequence* e *done\_sequence* são iguais.

- e. Campo *arrived\_at*, armazena o tempo de chegada de cada ordem de produção na fila de um dado recurso;
- f. Campo *started\_machining\_at*, armazena o instante do início do processamento de cada ordem de produção em um dado recurso;
- g. Campo *left\_resource\_at*, armazena o instante do término do processamento de cada ordem de produção em um dado recurso;
- h. Campo *total\_time*, armazena a diferença entre os tempos *started\_machining\_at* e *left\_resource\_at*;
- i. Campo *setup\_time*, armazena o tempo de *setup* de cada ordem de produção em um dado recurso.

Ainda estão implementados nesta tabela os campos *queue\_time*, que armazena o tempo de fila de cada ordem de produção em um dado recurso, e *cost*, responsável por armazenar o custo relativo ao processamento da ordem naquele recurso. Estes últimos dois campos não fazem parte do escopo deste trabalho, e foram implementados pensando em utilizações futuras.

A figura 8.1 apresenta as tabelas que compõem o banco de dados desenvolvido para o modelo proposto, bem como suas correlações, grifadas nas cores vermelha, verde e amarela.

- f) Tabela *simulation* – é responsável por armazenar um conjunto de registros que identificam cada uma das simulações com seus respectivos pedidos<sup>20</sup>, sendo formada por um:
- a. Campo *simulation\_id* que identifica cada uma das simulações e que deve ser único;
- b. Campo *job\_id* que aponta para o *id* da tabela *jobs*. Este campo relaciona as ordens que fazem parte de uma dada simulação;

---

<sup>20</sup> Pedido - corresponde a um grupo de ordens.

---

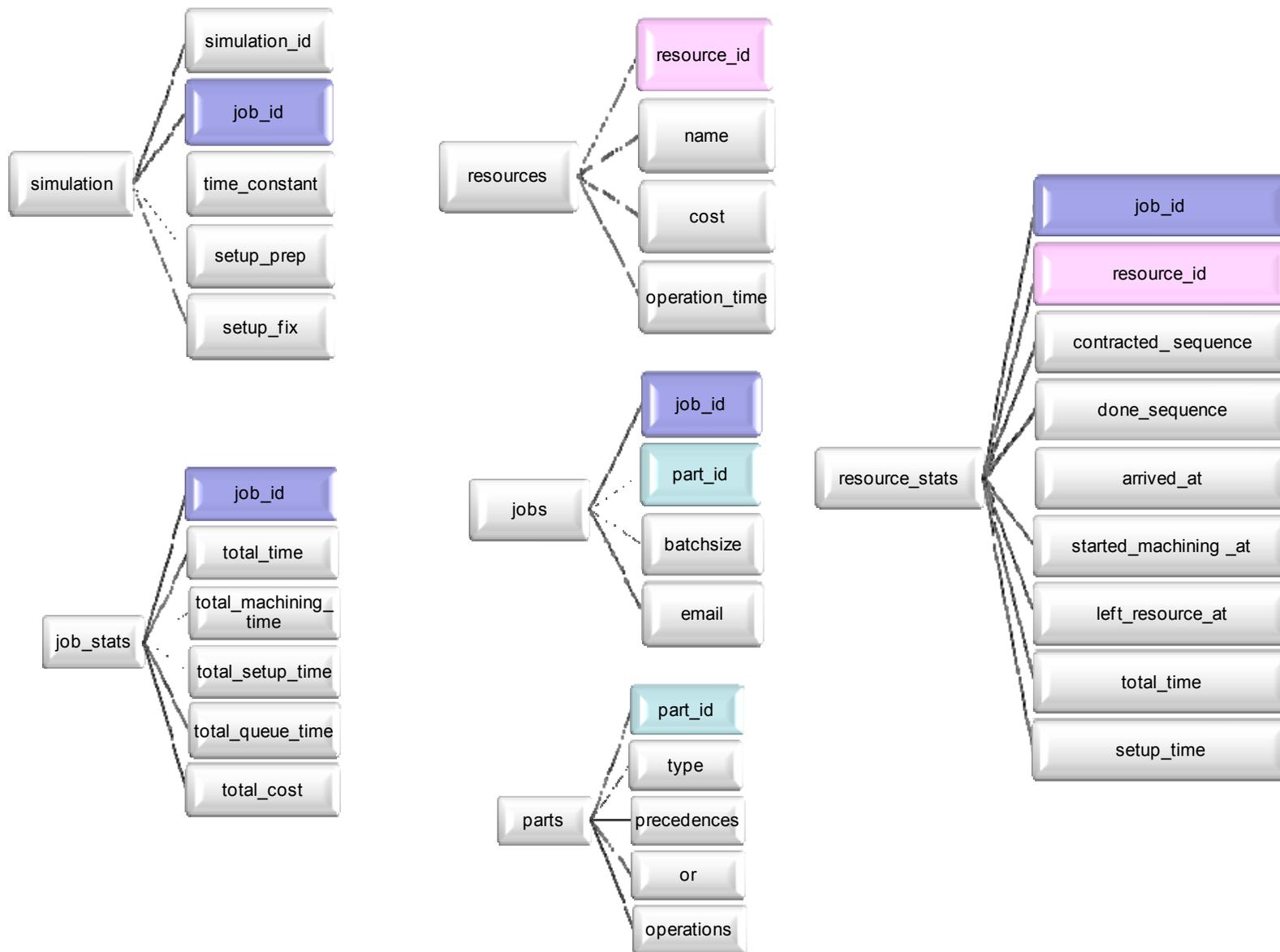


Figura 8.2 – Estrutura de dados desenvolvida para o modelo proposto

- a. Campo *time\_constant*, representa a constante de tempo utilizada para a simulação, maiores detalhes sobre o uso da constante de tempo e sua especificação, podem ser encontrados no item 7.1 que trata dos parâmetros de simulação utilizados no presente trabalho;
- b. Campos de *setup\_prep* e *setup\_fix*, responsáveis por identificar o percentual de cada um dos tempos de *setup* utilizados nas simulações, conforme já discutido no item 7.5 do presente trabalho.

Além dos objetivos anteriormente citados, a estrutura de armazenamento de dados desenvolvida para o modelo proposto também pode ser utilizada como base de dados, indicando aos futuros pedidos qual a melhor rota a ser seguida durante as simulações, a fim de minimizar uma das variáveis de resposta, *makespan* ou *flow time*. Para que isto ocorra foi adotado o uso de um sistema especialista baseado em regras de produção e integrado ao protótipo já desenvolvido, conforme será descrito a seguir.

## 8.2 O uso de sistemas especialistas combinados a agentes

De acordo com AYTUG et al., (1994) a adequação do uso de sistemas baseados em conhecimento<sup>21</sup> em qualquer que seja o domínio, reside no fato de que o conhecimento sobre este domínio poderá ser codificado em estruturas, como regras, lógicas, *scripts*, semânticas, e outras, sendo manipulado através de um mecanismo de inferência a fim de resolver um problema.

A associação entre agentes e sistemas especialistas pode ser encontrada nos mais diversos domínios, entre os quais: sistemas tutoriais (CRUCES e ARRIAGA, 2000); representação do conhecimento (QIAN, LI, JIANG e WEN, 2003); simulação industrial (ALDEA et al., 2004); cadeia de suprimentos (GJERDRUM et al., 2001).

Esta associação, conforme REZENDE (2005) faz uso da principal vantagem decorrente da união de objetos a regras, que reside no fato de que:

A orientação a objetos apesar de oferecer uma representação estrutural concisa de relações estáticas não oferece facilidades para descrever diretamente de forma declarativa como o conhecimento armazenado deve ser utilizado, o que pode ser tratado através de regras de produção.

---

<sup>21</sup> Sistemas baseados em conhecimento e sistemas especialistas são comumente tratados na literatura como sinônimos (REZENDE, 2005; AYTUG et al., 1994).

REZENDE (2005) ainda esclarece que:

Enquanto a orientação a objetos oferece uma forma rica, simples e natural para expressar os objetos do domínio, suas relações e a forma de comportamento, as regras de produção oferecem um meio simples e natural de expressar o raciocínio do sistema.

Para a integração com o protótipo desenvolvido na plataforma JADE optou-se pelo uso do JESS - *Java Expert System Shell* (JESS, 2007). O JESS é um ambiente de desenvolvimento para sistemas especialistas e uma máquina de inferência, tendo sido desenvolvido no final dos anos 90 em linguagem JAVA por Ernest J. Friedman-Hill do *Sandia National Laboratories* (FRIEDMAN-HILL, 2003). Com o avanço da utilização da linguagem JAVA e de sistemas baseados em agentes, o JESS passou a ser utilizado também como um componente de decisão para agentes implementados de forma declarativa (CARDOSO, 2007), conforme representado na figura 8.3.

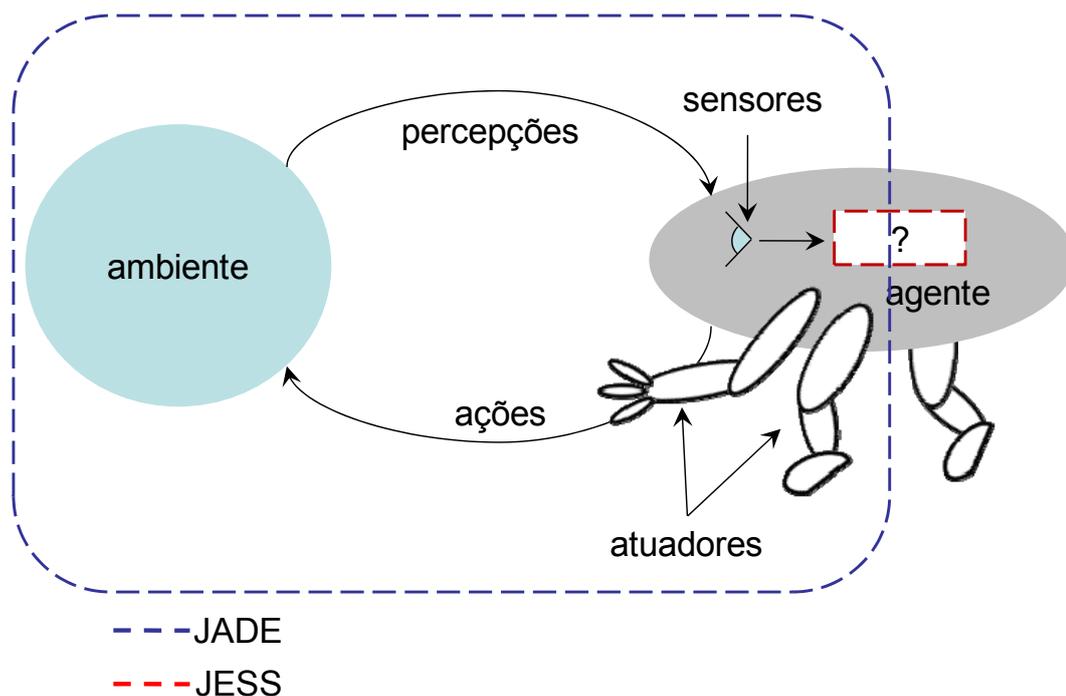


Figura 8.3 – Integração entre o *framework* JADE e o JESS (RUSSEL e NORVIG, 1995)

Nesta figura RUSSEL e NORVIG (1995) descrevem a estrutura genérica de um agente, nela é possível perceber que o agente possui uma visão limitada de seu ambiente através de suas percepções, e que são estas percepções que fazem com que o mesmo tome uma ou mais ações, através de seus atuadores, a fim de alcançar seus objetivos.

Neste contexto, o JESS age como um auxiliar na tomada de decisões, permitindo no caso do protótipo desenvolvido, que o agente mesmo sem ter conhecimento do cenário de

---

atuação, tenha capacidade de escolher a partir de uma experiência prévia semelhante, através da consulta a base de simulações já existente.

A associação do protótipo desenvolvido no *framework* JADE ao JESS foi desenvolvida de modo que a estrutura de inserção e retirada de informações do código seja feita através de funções estáticas do JAVA. Já as regras a serem aplicadas e a lógica utilizada pelo sistema especialista, foram desenvolvidas em código JESS, aproveitando-se desta forma as vantagens de ambas as linguagens.

### 8.3 Implementação

Conforme citado anteriormente no item 5.1, o agente *Server* tem as seguintes responsabilidades: (a) disponibilizar os planos de processos ao agente *InputOrder* quando solicitado; (b) receber os novos pedidos enviados pelo usuário através do agente *InputOrder*; (c) preparar os agentes *Resources* para o posterior início da simulação; (d) manter uma lista de todas as ordens em processamento na simulação para organizar e controlar o andamento da simulação; (e) instanciar os agentes *JobOrders*, que representam as ordens de fabricação por peça; e (f) instanciar o agente *Sync*, o qual é responsável por sincronizar os agentes *JobOrders* no chão-de-fábrica.

Além destas responsabilidades, duas novas foram incluídas a fim de permitir que o modelo proposto possa ser utilizado tanto para a criação de novas bases de consulta, o que poderá ser feito através de novas simulações; quanto como fonte de conhecimento para a comparação de casos pré-existentes, utilizando para tal um sistema especialista.

Dentro deste contexto também serão responsabilidades do agente *Server*: (g) detectar a disponibilidade, ou não, de um agente *Expert* na plataforma de simulação e (h) tomar a decisão cabível para cada um dos casos anteriores, de detecção ou não do agente *Expert*, antes de instanciar os agentes *JobOrders*. Ambas as funcionalidades serão descritas a seguir.

Para adequar-se as estas novas responsabilidades, foi criada uma alteração no estado *StartingSimulation* do agente *Server*, a qual é apresentada na figura 8.4.

Inicialmente, conforme já descrito no item 5.8.1 do presente trabalho, ao ser instanciado o agente *Server* assume o estado *WaitingOrders*, iniciando o comportamento *WaitForOrders*. Este comportamento é cíclico, ou seja, o agente *Server* permanece apto a receber novos pedidos do agente *InputOrder* durante todo o seu ciclo de vida. O comportamento *WaitForOrders* é interrompido quando: (a) for solicitado o resgate dos planos de processo das peças pelo agente *InputOrder*, através de uma mensagem *AllParts*; ou (b) receber uma

---

solicitação de novos pedidos, também gerada pelo agente *InputOrder*, através de uma mensagem *New\_Order*, sendo que esta solicitação permanecerá armazenada até que ocorra o início da simulação.

SERVER AGENT

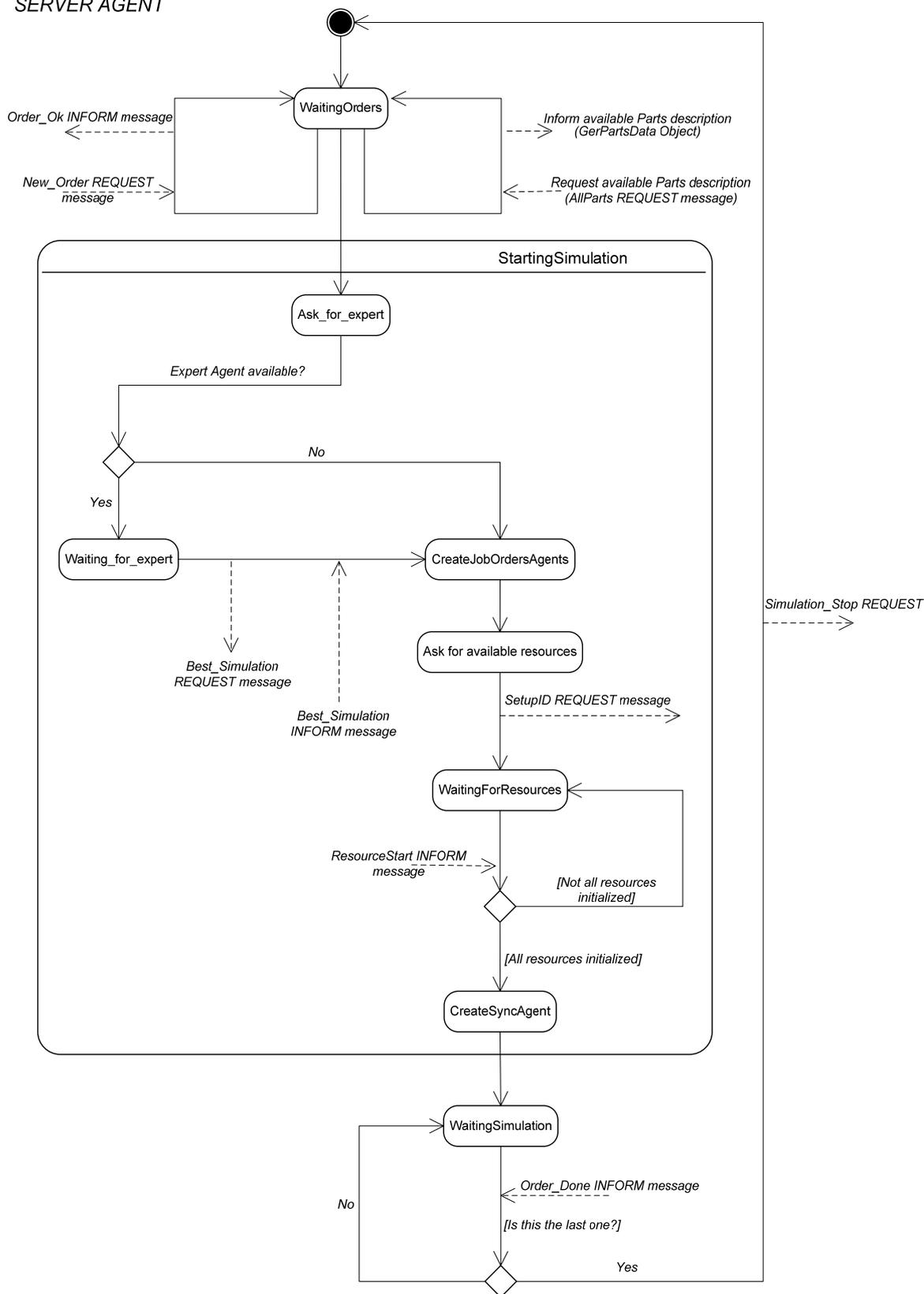


Figura 8.4 – Estrutura de dados desenvolvida para o modelo proposto

---

Para que a integração com o agente *Expert* ocorra, é necessário que a condição (b), acima citada, já tenha ocorrido, ou seja, que o agente *Server* já tenha recebido um pedido contendo ordens para uma nova simulação, neste instante o agente *Server* passa do estado *WaitingOrders* para o estado *StartingSimulation*, alterando também seu comportamento para *StartSimulation*.

Neste estado, o agente *Server* deveria transformar o pedido feito pelo usuário, o qual foi transmitido ao agente *Server* através da mensagem *New\_Order*, em diversas ordens de serviço, uma para cada peça que compõe o pedido original, utilizando para isto, o método *createJobOrderAgents*, porém, neste ponto da estrutura do agente *Server* foram inseridos dois novos métodos *ask\_for\_Expert* e *waiting\_for\_Expert* a serem descritos a seguir.

### 8.3.1 Métodos *ask\_for\_Expert* e *waiting\_for\_Expert*

O método *ask\_for\_Expert* é responsável por verificar a disponibilidade ou não de um agente *Expert* na plataforma de agentes. É importante lembrar que um agente pode ser visto por outro agente como um serviço a ser requisitado, o qual pode ser localizado através do recurso de páginas amarelas do diretório facilitador (DF) da plataforma JADE<sup>22</sup>. Quando o método *ask\_for\_Expert* é invocado duas possibilidades poderão ocorrer como retorno, a primeira indicando que o modelo estará sendo utilizado somente para simulação e conseqüente geração de base de dados, e a segunda indicando que o modelo estará sendo utilizado como fonte de consulta, (base de casos similares) para o sistema especialista:

- a) O registro do agente *Expert* não é localizado e o agente *Server* dá continuidade ao seu funcionamento através do método *createJobOrdersAgents*, conforme já descrito no item 5.8.1; ou
- b) O registro do agente *Expert* é localizado, fazendo com que o agente *Server* passe a executar o método estático *waiting\_for\_Expert*.

O método *waiting\_for\_Expert* é responsável por enviar uma mensagem *Best\_Simulation* ao agente *Expert* solicitando a melhor sugestão de roteamento para o pedido atual. Para tal é necessário enviar os dados relativos à simulação para a qual se deseja consultar a base de dados já existente, a fim de verificar as possíveis sugestões de roteamento. Esta simulação

---

<sup>22</sup> A plataforma JADE, bem como o recurso de diretório facilitador de sua estrutura já apresentados no capítulo 2 do presente trabalho.

---

---

atual corresponde ao pedido feito pelo agente *InputOrder* através da mensagem *New\_Order*, cujo funcionamento já foi descrito no item 5.8.1 do presente trabalho.

Para que o sistema especialista possa comparar a simulação atual às simulações em sua base de dados, é necessário que o método *waiting\_for\_Expert* envie os campos que a identificam, que são: *setup\_prep*, *setup\_fix*, *part\_id* e *batch\_size*, além de indicar qual critério deverá ser utilizado para a comparação entre os casos, que poderá ser o *makespan* de um grupo de peças ou o *flow time* de uma única peça. Estes dados são enviados através da mensagem *Best\_Simulation*.

O agente *Server* irá permanecer no estado *waiting\_for\_Expert* até que receba uma resposta para a mensagem *Best\_Simulation*, que poderá conter ou não a sugestão de roteamento solicitada, desta forma duas são as possíveis respostas para esta mensagem:

- a) Uma mensagem *Best\_Simulation* poderá ter um retorno nulo caso nenhuma simulação com características idênticas à solicitada tenha sido encontrada, neste caso as ordens geradas através do método *createJobOrdersAgents* irão utilizar os planos de processo originais para cada uma das peças que compõe o pedido feito através do agente *InputOrder*; ou
- b) A mensagem *Best\_Simulation* retorna ao agente *Server* uma sugestão de roteamento, baseada no critério de comparação solicitado. Neste caso as ordens geradas através do método *createJobOrdersAgents* não irão utilizar os planos de processo originais para cada uma das peças que compõe o pedido feito através do agente *InputOrder* e sim aqueles sugeridos pela base de casos do sistema especialista.

Um exemplo da diferença de como as ordens de produção podem ser geradas com ou sem uma sugestão de um roteamento é dado na figura 8.5. Inicialmente o agente *Server* recebe um pedido através do agente *InputOrder* composto por três peças distintas, a peça 3, 8 e 12. Este pedido tem como características o tamanho de lote igual a um e os tempos de *setup* de preparação e fixação iguais a 0,1. A forma de entrada dos parâmetros de uma dada simulação bem como uma discussão a respeito dos mesmos pode ser encontrada no capítulo 7 do presente trabalho.

Ao receber este pedido o agente *Server* passa para o estado *StartingSimulation*, alterando também seu comportamento para *StartSimulation* e invocando o método *ask\_for\_Expert*, anteriormente descrito. Caso o método *ask\_for\_Expert* tenha um retorno nulo, ou seja, não encontre o registro de um agente *Expert*, o agente *Server* dará continuidade ao seu funcionamento através do método *createJobOrdersAgents*.

---

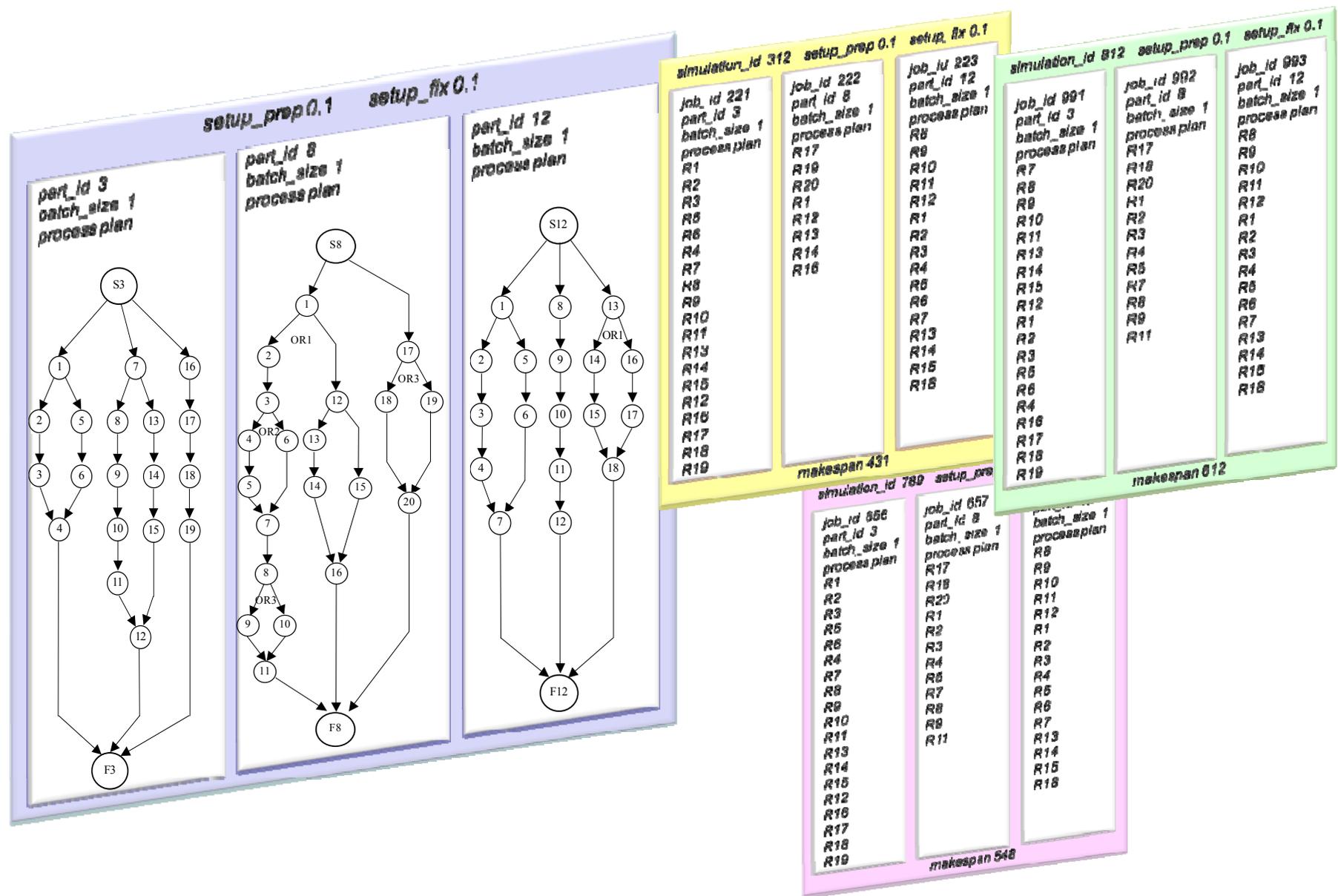


Figura 8.5 – Sugestões de roteamento obtidas a partir da variável *makespan*

---

Serão criadas três ordens, uma para cada peça que compõe o pedido, sendo que nestas ordens o roteamento das peças entre os recursos disponíveis no chão-de-fábrica não estará definido. Isto significa que na primeira rodada de negociação entre as ordens e os recursos, todas as alternativas dos planos de processo serão consideradas, desta forma a peça 3 negociará as operações 1, 7, 16; a peça 8 negociará as operações 1 e 17; e finalmente a peça 12 as operações 1, 8 e 13, entre todos os recursos disponíveis para aquela simulação e com capacidade para executar as mesmas.

Porém, na hipótese do método *ask\_for\_Expert* encontrar o registro de um agente *Expert*, a mensagem *Best\_Simulation* retornará ao agente *Server* uma sugestão de roteamento baseada no critério de comparação solicitado, no caso do exemplo da figura 8.3 este critério foi o menor valor encontrado para a variável de resposta *makespan* do pedido de manufatura das peças 3, 8 e 12. Observa-se que por este critério três simulações, cujas características *setup\_prep*, *setup\_fix*, *part\_id* e *batch\_size* atendem às exigidas pelo pedido, foram encontradas, sendo elas: as simulações 312, 812 e 789, cujos valores de *makespan* são respectivamente 431, 612 e 548.

A seguir estas três simulações são ranqueadas por melhor solução baseada no critério de solução selecionado, assim tem-se por ordem crescente, ou seja, da melhor para a pior, as simulações 312, 789 e 812.

Finalmente, obedecendo-se a este ranqueamento, a primeira rodada de negociação entre as ordens e os recursos seguirá a sugestão de roteamento estabelecida na simulação 312, onde a peça 3 negociará a operação 1; a peça 8 negociará a operação 17; e finalmente a peça 12 a operação 8.

### 8.3.2 O agente Expert e sua interface com o JESS - Java Expert System Shell

O agente *Expert* tem por principal função criar uma sugestão de roteamento entre as ordens de produção e os recursos disponibilizados no chão-de-fábrica, sendo composto por um único comportamento cíclico chamado *Communication*. Esta sugestão de roteamento, conforme já descrito no exemplo da figura 8.5, será utilizada na primeira rodada de negociações do modelo proposto. Para este fim três são as responsabilidades que o agente *Expert* terá de cumprir a partir do recebimento da mensagem *Best\_Simulation*, a qual contém os dados da simulação, bem como o critério que deverá ser utilizado para a comparação entre os casos: (a) criar uma base de fatos a fim de recuperar os casos similares a partir do banco de dados; (b) a partir da base de fatos gerar um ranqueamento das simulações baseado nos

---

resultados apresentados e no critério de solução selecionado; e (c) retornar ao agente *Server* a melhor sugestão de roteamento dentro do critério selecionado. O funcionamento da interface entre o agente *Server*, o agente *Expert* e o JESS é apresentado na figura 8.6 e descrito a seguir.

Inicialmente o comportamento *Communication* do agente *Expert* se encontra bloqueado, assim permanecendo até que o agente receba uma mensagem *Best\_Simulation* por parte do agente *Server*.

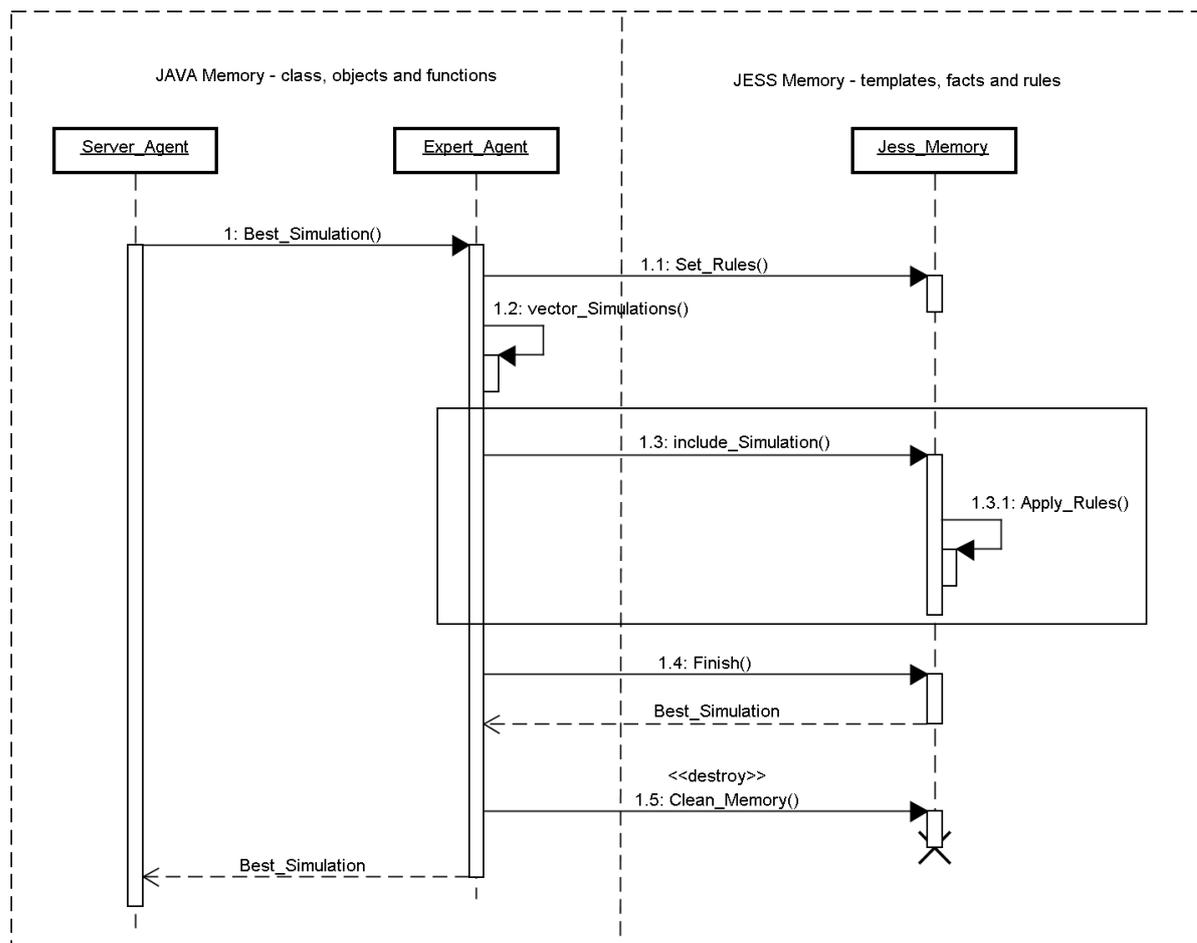


Figura 8.6 – Interface entre os agentes e o JESS

A partir do recebimento desta mensagem o agente *Expert* invoca o método *setRules* o qual é responsável por carregar a regra selecionada, a partir do critério recebido na mensagem *Best\_Simulation*, na memória *JESS*. Até o presente momento três regras foram desenvolvidas no protótipo proposto, todas baseadas em uma variável de resposta que possa ser utilizada como comparação entre os casos, sendo elas: menor *makespan*, menor *flow time* e menor tempo de fila (*queue time*). É importante observar que para que uma regra seja desenvolvida, é necessário instanciar classes JAVA como fatos da base de conhecimento do agente *Expert*,

---

além de instanciar as classes JAVA a partir de fatos da base de conhecimento deste agente. Esta dupla instanciação permite uma comunicação de duas vias entre a memória em JAVA e a memória em JESS.

Na seqüência, o agente *Expert* invoca o método *includeSimulation* responsável por iniciar no banco de dados uma busca por todas as simulações similares àquela que se deseja controlar. Isto é feito dentro de um *loop* de programação responsável por selecionar as simulações que possuam as características semelhantes aquelas pré-estabelecidas no pedido. Estas simulações são então adicionadas uma a uma na memória JESS sendo que para isto devem ser instanciadas como fatos. Isto é necessário, pois ao se instanciar uma classe como um fato na memória do agente *Expert*, o objeto e o fato continuam sendo duas entidades diferentes, porém com possibilidade de associação entre elas. De forma análoga, ao instanciar um fato da memória JESS a partir de um objeto JAVA, cria-se um ponteiro a partir do qual é possível a recuperação do mesmo. Ainda durante o *loop* é feita uma varredura na base de fatos que esta sendo criada, aplicando-se a regra ou o conjunto de regras previamente habilitadas através do método *applyRules* a fim de selecionar a melhor simulação sob o ponto de vista do critério de comparação selecionado, que é disponibilizada da memória JESS para o agente *Expert* através do objeto *BestSimulation*. Finalmente o agente *Expert* invoca o método *BestSimulation* cujo retorno é uma referência a classe *BestSimulation* anteriormente instanciada pelo agente *Server*, enviando uma mensagem com a melhor simulação ao agente *Server* e limpando a memória JESS, preparando-a desta forma para uma próxima busca.

#### 8.4 Resultados encontrados e conclusão do capítulo

A partir da implementação das novas funcionalidades, já descritas no presente capítulo, foram feitos testes a fim de validar sua aplicabilidade no protótipo desenvolvido. Procurando-se evitar um grande volume de tabelas e informações repetidas, optou-se por apresentar a seguir somente os valores obtidos para a variável de controle *makespan* em três grupos de controle, *sets* 01, 12 e 24. Os *sets* escolhidos têm por finalidade representar grupos formados por diferentes números de peças, 6, 9 e 18 respectivamente, sendo que sua estrutura e as peças que os compõe já foram apresentados no capítulo 6 do presente trabalho.

As tabelas 8.1 a 8.3 apresentam os resultados obtidos para cada um dos grupos de simulação: grupo A que permite as flexibilidades de operação e seqüência; grupo B que permite as flexibilidades de operação, seqüência e máquina; grupo C que permite as flexibilidades de operação, seqüência e processo; e finalmente grupo D que permite as

---

---

flexibilidades de operação, seqüência, processo e máquina. Ao observar estes resultados é possível concluir que:

- a) Existe uma clara tendência de redução do desvio padrão, o que indica uma menor variabilidade, ou seja, a solução proposta apresenta uma boa repetibilidade;
- b) Também é possível observar que os grupos que apresentam um desvio padrão acima de 3%, têm este valor em decorrência da melhoria dos resultados, o que é observado através da redução da média da variável *makespan* obtida. Isto ocorre em virtude da entrada de novos casos na base de dados após cada simulação;
- c) Existe uma tendência de estabilização nos valores obtidos;
- d) Os resultados que apresentam um desvio padrão abaixo de 3% são decorrentes da variabilidade inerente ao tempo de troca de mensagens, comum a sistema de simulação baseados em agentes.

Finalmente é importante observar que o termo sugestão de roteamento foi utilizado, pois não há o impedimento da geração das ordens de serviço no caso da primeira rodada de negociações não ser cumprida devido a algum imprevisto, como a indisponibilidade de um determinado recurso, por exemplo. Neste caso as ordens serão geradas normalmente através do método *createJobOrdersAgents* utilizando todas as alternativas contidas nos planos de processo na primeira rodada de negociação entre as operações e os recursos disponíveis no chão-de-fábrica, conforme já citado anteriormente.

---

Tabela 8. 1 – Valores da variável *flow time* obtidos com o sistema especialista para os grupos A, B, C e D nos sets 01, 12 e 24

Set 01 - Grupo A						Set 12 - Grupo A						Set 24 - Grupo A					
setup 0_0	setup0.1_0.1	setup0.3_0.1	setup0.5_0.1	setup0.7_0.1		setup 0_0	setup0.1_0.1	setup0.3_0.1	setup0.5_0.1	setup0.7_0.1		setup 0_0	setup0.1_0.1	setup0.3_0.1	setup0.5_0.1	setup0.7_0.1	
Makespan	Makespan	Makespan	Makespan	Makespan		Makespan	Makespan	Makespan	Makespan	Makespan		Makespan	Makespan	Makespan	Makespan	Makespan	
501,41	461,25	449,25	371,84	328,59		386,28	374,25	340,47	361,47	346,41		576,53	553,84	510,25	451,69	399,94	
504,25	461,22	449,38	371,84	328,59		386,34	373,81	342,34	354,63	346,44		558,50	569,00	512,54	468,00	399,84	
504,28	461,28	449,34	371,81	328,62		386,28	373,62	339,56	355,50	343,63		560,56	559,41	495,47	452,34	395,03	
504,22	461,25	449,12	371,81	328,59		386,28	373,72	343,37	350,44	342,53		563,44	555,81	512,44	464,12	395,16	
504,25	461,22	449,28	372,13	328,59		378,44	373,69	346,47	356,59	341,09		558,44	550,47	501,97	451,63	389,37	
<b>Desvio padrão</b>	<b>1,27</b>	<b>0,03</b>	<b>0,10</b>	<b>0,13</b>	<b>0,01</b>	<b>Desvio padrão</b>	<b>3,51</b>	<b>0,25</b>	<b>2,71</b>	<b>3,97</b>	<b>2,37</b>	<b>Desvio padrão</b>	<b>7,57</b>	<b>7,09</b>	<b>7,55</b>	<b>7,89</b>	<b>4,35</b>
Set 01 - Grupo B						Set 12 - Grupo B						Set 24 - Grupo B					
setup 0_0	setup0.1_0.1	setup0.3_0.1	setup0.5_0.1	setup0.7_0.1		setup 0_0	setup0.1_0.1	setup0.3_0.1	setup0.5_0.1	setup0.7_0.1		setup 0_0	setup0.1_0.1	setup0.3_0.1	setup0.5_0.1	setup0.7_0.1	
Makespan	Makespan	Makespan	Makespan	Makespan		Makespan	Makespan	Makespan	Makespan	Makespan		Makespan	Makespan	Makespan	Makespan	Makespan	
504,41	461,25	449,25	371,84	328,59		361,50	358,97	359,44	388,25	365,22		544,62	489,69	512,34	406,59	407,38	
504,25	461,22	449,38	371,84	328,59		361,34	358,69	359,28	402,56	365,34		544,34	504,56	512,66	416,78	393,41	
504,28	461,28	449,34	371,81	328,62		362,41	357,69	359,38	388,25	365,25		532,38	499,97	480,03	416,63	393,59	
504,22	461,25	449,31	371,81	328,59		337,38	358,56	359,28	388,16	372,44		535,16	505,28	483,31	412,44	406,41	
504,25	461,22	449,28	372,13	328,59		337,34	358,19	359,31	387,94	371,12		535,47	502,94	483,81	412,84	407,50	
<b>Desvio padrão</b>	<b>0,07</b>	<b>0,03</b>	<b>0,05</b>	<b>0,13</b>	<b>0,01</b>	<b>Desvio padrão</b>	<b>13,37</b>	<b>0,50</b>	<b>0,07</b>	<b>6,45</b>	<b>3,60</b>	<b>Desvio padrão</b>	<b>5,69</b>	<b>6,37</b>	<b>16,56</b>	<b>4,15</b>	<b>7,46</b>
Set 01 - Grupo C						Set 12 - Grupo C						Set 24 - Grupo C					
setup 0_0	setup0.1_0.1	setup0.3_0.1	setup0.5_0.1	setup0.7_0.1		setup 0_0	setup0.1_0.1	setup0.3_0.1	setup0.5_0.1	setup0.7_0.1		setup 0_0	setup0.1_0.1	setup0.3_0.1	setup0.5_0.1	setup0.7_0.1	
Makespan	Makespan	Makespan	Makespan	Makespan		Makespan	Makespan	Makespan	Makespan	Makespan		Makespan	Makespan	Makespan	Makespan	Makespan	
452,63	447,59	447,47	440,63	445,25		343,59	367,50	377,41	346,25	350,16		537,59	553,56	496,50	523,62	612,25	
452,56	447,59	447,50	439,50	450,28		342,50	369,56	377,22	349,37	354,22		538,75	555,53	499,37	514,47	612,53	
447,59	447,62	454,37	439,50	450,25		348,44	369,53	336,34	343,53	352,19		540,72	547,66	509,38	522,69	612,31	
447,62	452,56	447,47	439,53	444,25		343,44	364,41	336,38	338,41	350,19		537,66	554,59	508,44	525,69	584,16	
452,66	447,59	454,41	439,50	446,31		344,47	364,44	336,37	338,50	346,16		537,59	550,07	505,50	527,66	580,97	
<b>Desvio padrão</b>	<b>2,74</b>	<b>2,22</b>	<b>3,79</b>	<b>0,50</b>	<b>2,83</b>	<b>Desvio padrão</b>	<b>2,32</b>	<b>2,57</b>	<b>22,43</b>	<b>4,81</b>	<b>2,99</b>	<b>Desvio padrão</b>	<b>1,35</b>	<b>3,31</b>	<b>5,66</b>	<b>5,05</b>	<b>16,36</b>
Set 01 - Grupo D						Set 12 - Grupo D						Set 24 - Grupo D					
setup 0_0	setup0.1_0.1	setup0.3_0.1	setup0.5_0.1	setup0.7_0.1		setup 0_0	setup0.1_0.1	setup0.3_0.1	setup0.5_0.1	setup0.7_0.1		setup 0_0	setup0.1_0.1	setup0.3_0.1	setup0.5_0.1	setup0.7_0.1	
Makespan	Makespan	Makespan	Makespan	Makespan		Makespan	Makespan	Makespan	Makespan	Makespan		Makespan	Makespan	Makespan	Makespan	Makespan	
457,59	447,59	447,44	440,56	446,19		340,41	343,47	344,31	359,44	354,13		561,31	499,66	521,41	558,69	536,13	
453,66	447,59	454,34	436,56	439,19		340,31	343,44	340,25	360,41	331,34		560,94	514,78	535,50	559,69	533,12	
454,59	454,69	454,44	444,56	439,16		333,38	340,09	339,87	359,44	354,31		559,75	508,62	534,53	554,75	537,28	
453,69	447,75	447,44	444,53	439,12		338,37	343,50	340,09	359,37	333,16		563,91	510,59	533,44	556,72	510,22	
454,59	454,63	454,37	444,53	439,16		346,47	340,53	333,28	359,66	340,13		552,62	510,56	537,50	558,66	510,31	
<b>Desvio padrão</b>	<b>1,62</b>	<b>3,84</b>	<b>3,81</b>	<b>3,57</b>	<b>3,14</b>	<b>Desvio padrão</b>	<b>4,70</b>	<b>1,74</b>	<b>3,96</b>	<b>0,43</b>	<b>11,09</b>	<b>Desvio padrão</b>	<b>4,24</b>	<b>5,61</b>	<b>6,37</b>	<b>1,97</b>	<b>13,91</b>

---

## CAPÍTULO 9 – CONCLUSÕES E RECOMENDAÇÕES

Este capítulo trata da discussão e apresentação dos resultados obtidos durante o desenvolvimento da presente tese.

Inicialmente é apresentada uma discussão relacionando o objetivo geral do trabalho com os resultados obtidos, também é apresentado o desenvolvimento do protótipo do *software* de simulação proposto.

A seguir são apresentados e discutidos, em forma de conclusões, os dados gerados a partir das simulações do modelo proposto.

Na continuidade do capítulo são descritas as contribuições da presente tese, estas contribuições são discutidas e apresentadas sob três diferentes aspectos: contribuições científicas e técnicas; contribuições de natureza geral e publicações.

Finalmente são apresentadas as recomendações para trabalhos futuros.

### 9.1 Discussão

Buscando responder a pergunta de pesquisa que norteia este trabalho: *Uma solução baseada em agentes é adequada para o estudo da correlação entre planos de processo alternativos, representados sob a forma de diferentes flexibilidades, e a programação da produção em um ambiente de layout funcional?* Foi implementado um modelo de simulação baseado em multiagentes a partir de um modelo conceitual da organização multiagente e de sua respectiva representação em termos de uma especificação de projeto, a qual foi desenvolvida a partir da metodologia PASSI. O modelo consiste dos agentes de recurso (*Resource*), agentes de ordem (*JobOrder*), de interface (*Server*, *InputOrder* e *Sync*) e de sistema especialista (*Expert*).

Cada agente foi considerado sob a ótica de dois diferentes aspectos: como entidade autônoma, em função de seus papéis e conhecimento; e como unidade de software. Cada papel é visto como uma coleção de tarefas a serem realizadas pelo agente e a partir dos papéis são derivados os diversos comportamentos que compõem cada um dos agentes.

Neste protótipo duas abordagens de encapsulamento de funções ou tarefas foram utilizadas: a decomposição física para os agentes de recurso (*Resource*); e a decomposição funcional para os demais agentes (*Server*, *JobOrder*, *InputOrder*, *Sync* e *Expert*).

Também foram desenvolvidos dois diferentes protocolos para esta aplicação, os protocolos de negociação e renegociação. Ambos os protocolos tem como objetivo tratar da

---

---

comunicação e resolução de conflitos entre os agentes de ordem e de recurso, permitindo o roteamento das ordens entre os recursos disponíveis a cada rodada de simulação.

Finalmente, como forma de representar a modularidade do protótipo, foi desenvolvida uma interface com um sistema especialista através da inclusão de um agente especialista (*Expert*). Esta interface age como um auxiliar na tomada de decisões, permitindo, no caso do protótipo desenvolvido, que o agente mesmo sem ter conhecimento do cenário de atuação, tenha capacidade de escolher a partir de uma experiência prévia semelhante, através da consulta a base de simulações já existente.

É importante observar que, tendo em vista a modularidade, escalabilidade e flexibilidade das organizações multiagentes, novas instâncias poderão ser incluídas ou excluídas a qualquer momento, sem prejuízo da autonomia ou desempenho do modelo aqui apresentado e desenvolvido.

Com o propósito de auxiliar na validação e verificação, foi feita uma comparação entre os dados obtidos pelo modelo proposto e aqueles obtidos por KIM et al. (2003) e POTTER (1997), sendo que todas as abordagens utilizaram o mesmo caso de testes, sugerido por YEO et al. (2003).

Os resultados experimentais obtidos demonstraram que a autonomia e capacidade de cooperação de uma sociedade heterárquica baseada em agentes são capazes de apresentar bons resultados em problemas onde o número de soluções possíveis aumenta exponencialmente (*NP-Hard*). Foi observado que, ao aumentar-se o número de peças que compõem um *set* de simulação, menor é o valor da média da variável *makespan* obtida em relação aos valores apresentados por KIM et al. (2003) e POTTER (1997), em especial em grupos compostos a partir de 12 peças. Ou seja, ao se aumentar o número de peças que compõem um grupo de simulação, a abordagem baseada em agentes, sugerida neste trabalho, apresenta melhores resultados que as abordagens baseadas em algoritmos genéticos sugeridas por KIM et al. (2003) e POTTER (1997).

Dentro deste contexto, o modelo proposto foi considerado como uma abordagem viável e promissora para o estudo da correlação entre planos de processo alternativos, representados sob a forma de diferentes flexibilidades, e o seqüenciamento da produção em um ambiente de *layout* funcional.

### 9.1.1 Desenvolvimento do protótipo de *software* de simulação

---

---

Um protótipo de *software* de simulação baseado em agentes, cujo principal objetivo é o estudo da correlação entre planos de processo alternativos, representados sob a forma de diferentes flexibilidades, e a programação da produção, foi desenvolvido e validado no presente trabalho de pesquisa.

O protótipo foi desenvolvido em linguagem JAVA, utilizando a plataforma JADE - *Java Agent DEvelopment framework*, em conformidade às especificações da FIPA – *Foundation for Intelligent Physical Agents*.

A fim de permitir o uso de regras de produção no protótipo desenvolvido, utilizou-se um ambiente de desenvolvimento para sistemas especialistas e uma máquina de inferência, denominada JESS - *Java Expert System Shell*, criado no final dos anos 90 em linguagem JAVA por Ernest J. Friedman-Hill do *Sandia National Laboratories*.

A validação do modelo proposto foi realizada através da demonstração da viabilidade de seu modelo conceitual, em especial na representação dos componentes de simulação, agentes e protocolos de negociação específicos. Já a sua verificação foi realizada sob uma ótica quantitativa, através da comparação dos valores obtidos pelo método proposto e aqueles obtidos por KIM et al. (2003) e POTTER (1997).

Finalmente, a maior dificuldade encontrada residiu na falta de casos de testes padronizados para o escopo do presente trabalho. Embora seja relativamente simples encontrar bases de dados comparativas para a programação da produção, nas suas mais diversas formas e regras, estas bases não trazem, em geral, o plano de processo das peças utilizadas, ou mesmo as flexibilidades permitidas para o roteamento das operações nos diversos recursos.

## 9.2 Conclusões

A partir dos dados obtidos através do modelo de simulação proposto, diversas conclusões sobre a correlação entre planos de processo alternativos, representados sob a forma de diferentes flexibilidades, e o seqüenciamento da produção, representado a partir das métricas de *makespan* e *flow time*, foram observadas. As conclusões consideradas relevantes serão apresentadas a seguir.

---

---

### 9.2.1 Conclusões sobre os grupos de simulação A, B, C e D em relação à variável de resposta *makespan* para tempos de *setup* iguais a zero

Conforme citado no capítulo 7, as simulações foram separadas em quatro grupos distintos por tipos de flexibilidades de planos de processo permitidas. O grupo A foi composto pelas flexibilidades de operação e seqüência; o grupo B pelas flexibilidades de operação, seqüência e máquina; o grupo C pelas flexibilidades de operação, seqüência e processo; e, finalmente, o grupo D pelas flexibilidades de operação, seqüência, processo e máquina.

A seguir são apresentadas as conclusões consideradas mais relevantes após as rodadas de simulação e análise de dados do modelo proposto:

1. Os maiores ganhos, percentuais e na média global, da variável de resposta *makespan*, foram encontrados no grupo que permite o menor número de flexibilidades (grupo A);
  2. Ao se incluir a flexibilidade de processo (Grupo C), observou-se que os valores globais da variável *makespan* situaram-se bastante próximos aos encontrados no Grupo A, o mesmo ocorrendo com o desvio padrão. Isto indica que, apesar do incremento no número de opções de plano de processo, com a inclusão da flexibilidade de processo, não houve uma influência direta na variável de resposta *makespan*;
  3. Em relação à flexibilidade de máquina, observou-se que nos grupos em que a mesma foi incluída (grupos B e D) o desempenho global da variável *makespan* foi pior do que os demais grupos (A e C). Este resultado é explicado tendo-se em vista que o agrupamento de operações foi utilizado considerando-se os tempos de *setup* de forma independente;
  4. Ainda sobre a análise do uso de flexibilidade de máquina, observou-se que a inclusão ou não de mais uma flexibilidade gerou pouco impacto nos resultados obtidos, o que pode ser observado nas médias globais da variável *makespan* nos grupos D e B;
  5. Observou-se que a flexibilidade de operações, em especial nos seus níveis médio e alto, tem uma influência direta e negativa sobre o valor da variável *makespan*, quando associada às flexibilidades de processo e máquina, o mesmo não ocorrendo quando associada à flexibilidade de seqüência;
  6. Em relação à flexibilidade de seqüência, foi observado que a mesma, em seus níveis baixo e médio, afeta negativamente os valores da variável *makespan* em todos os grupos de simulação;
  7. Independentemente dos grupos de simulação, A, B, C ou D, os maiores ganhos se concentram nos grupos formados por um maior número de peças (a partir de nove), com destaque para os grupos com o número de peças variando entre doze e dezoito.
-

---

### 9.2.2 Conclusões sobre os grupos de simulação A, B, C e D em relação à variável de resposta *flow time* para tempos de *setup* iguais a zero

1. Os maiores ganhos da variável de resposta *flow time*, tanto percentuais como em relação à média global, foram encontrados no grupo A;
2. Ao se incluir a flexibilidade de processo (Grupo C), os valores globais situaram-se próximos daqueles encontrados no Grupo A, o mesmo ocorrendo com o desvio padrão. Isto indica que o incremento no número de opções de plano de processo não resulta em uma influência direta na variável de resposta *flow time*;
3. Nos grupos em que a flexibilidade de máquina foi incluída (grupos B e D), o desempenho global da variável *flow time* foi pior em relação aos grupos A e C. Este resultado é explicado tendo-se em vista que o agrupamento de operações foi utilizado considerando-se os tempos de *setup* de forma independente;
4. Os maiores ganhos se concentraram nos grupos formados por um maior número de peças, a partir de nove, com destaque para os grupos com o número de peças variando entre doze e dezoito, independente do tipo de combinação de flexibilidades; e
5. Não foi observada nenhuma outra influência direta entre as flexibilidades predominantes e seus respectivos níveis, em cada um dos vinte e quatro grupos de simulação, sobre a variável *flow time*, à exceção dos grupos C e A, onde houve uma pequena influência da flexibilidade de seqüência no seu nível baixo e médio sobre os grupos formados por 6 ou 9 peças.

### 9.2.3 Conclusões sobre os grupos de simulação A, B, C e D em relação às variáveis de resposta *makespan* para tempos de *setup* diferentes de zero

Observou-se que os resultados relativos à variável *makespan* e aqueles relacionados ao *flow time*, para todos os grupos de simulação A, B, C e D, com tempos de *setup* diferentes de zero, seguiram as mesmas tendências já apresentadas para os grupos com tempos de *setup* iguais a zero.

1. Novamente observou-se que um aumento no ganho percentual nos valores das médias de *makespan* relacionava-se a um maior número de peças por *set* de simulação; e
  2. Foi confirmada a tendência de que a flexibilidade de seqüência, em seus níveis baixo e médio, afeta negativamente os valores da variável *makespan* em todos os grupos de simulação.
-

---

#### 9.2.4 Conclusões sobre os grupos de simulação A, B, C e D em relação às variáveis de resposta *flow time* para tempos de *setup* diferentes de zero.

1. Novamente observou-se que um aumento no ganho percentual nos valores das médias de *makespan* está relacionado a um maior número de peças por *set* de simulação; e
2. Foi confirmada a tendência de que a flexibilidade de seqüência, em seus níveis baixo e médio, afeta negativamente os valores da variável *flow time* em todos os grupos de simulação.

### 9.3 Contribuições

O trabalho aqui apresentado buscou alcançar uma série de resultados, os quais serão sumarizados nas seguintes categorias: principais contribuições científicas e técnicas; contribuições de natureza geral; e publicações.

Como principais contribuições científicas e técnicas do presente trabalho, podem-se citar:

1. O modelo conceitual e a especificação de projeto para a arquitetura e implementação do protótipo de simulação baseado em agentes, para o estudo da correlação entre planos de processo alternativos, representados sob a forma de diferentes flexibilidades, e o seqüenciamento da produção. Tendo-se em vista que os mesmos foram desenvolvidos a partir dos objetivos geral e específicos estabelecidos.
2. O desenvolvimento do protótipo a partir das especificações de projeto do modelo e sua posterior verificação e validação, cujos detalhes formais da implementação têm por objetivo permitir a sua futura repetibilidade ou a construção de novas aplicações.
3. Os resultados obtidos e sua posterior análise dentro do contexto especificado a partir dos objetivos geral e específico estabelecidos, considerando-se a complexidade combinatorial do ambiente proposto (um *layout funcional*) e o número de alternativas consideradas por plano de processo.
4. Aplicação de tecnologia multiagente para a solução de um problema não determinístico.

Como contribuições de natureza geral, considerando o desenvolvimento de um trabalho de pesquisa padronizado que utiliza o paradigma de multiagentes, podem-se citar:

---

1. O desenvolvimento de um protótipo totalmente aderente às especificações FIPA e desenvolvido em linguagem FIPA ACL;
2. A extensão do protocolo de negociação padrão – *contract net protocol*;
3. A integração de diferentes paradigmas de *software*, entre os quais agentes e sistemas especialistas;
4. A utilização de uma metodologia para o desenvolvimento de uma sociedade baseada em agente, evitando assim uma solução *ad hoc*, cuja replicação seria difícil ou mesmo inviável.

Diversas publicações foram geradas até o momento com o desenvolvimento do presente trabalho, especialmente em relação aos capítulos 6, 7 e 8. Estas publicações foram apresentadas em jornais internacionais, livros e anais de conferências, cuja lista é apresentada a seguir:

Artigos completos publicados em periódicos

1. ZATTAR, Izabel Cristina; FERREIRA, João Carlos Espíndola; GRANADO, João Gabriel G.; SOUSA, Carlos Humberto Barreto de. **Integration between Process Planning and Scheduling Using Feature-Based Time-Extended Negotiation Protocols in a Multi-Agent System**. International Journal of Services Operations and Informatics (IJSOI), v. 3, p. 71-89, 2008.
2. FERREIRA, João Carlos Espíndola; ZATTAR, Izabel Cristina; GRANADO, João Gabriel G.; SOUSA, Carlos Humberto Barreto de. **Features de Usinagem Auxiliam Produção de Peças Compostas**. Máquinas e Metais, v. 43, p. 166-189, 2007.

Capítulos de livros publicados

1. ZATTAR, Izabel Cristina; FERREIRA, João Carlos Espíndola; GRANADO, João Gabriel G.; SOUSA, Carlos Humberto Barreto de. **Integrating Manufacturing Process Planning with Scheduling via Operation-Based Time-Extended Negotiation Protocols**. In: Geilson Loureiro; Richard Curran. (Org.). Complex Systems Concurrent Engineering. Londres: Springer, 2007, v. 1, p. 329-336.

Trabalhos completos publicados em anais de congressos

1. ZATTAR, Izabel Cristina; FERREIRA, João Carlos Espíndola; BOTURA, Paulo E de A. **Reducing the Standard Deviation When Integrating Process Planning and Production Scheduling Through the Use of Expert Systems in an**

---

**Agent-based Environment.** In: 15th ISPE International Conference on Concurrent Engineering, 2008, Belfast. 15th ISPE International Conference on Concurrent Engineering, 2008.

2. ZATTAR, Izabel Cristina ; FERREIRA, João Carlos Espíndola ; GRANADO, João Gabriel G. ; MORASCO, Rodrigo Dal Bello . **Fabricação de Peças Compostas por Features de Usinagem utilizando um Modelo de Agrupamento: Uma abordagem baseada em Sistemas Multitagentes.** In: Congresso Brasileiro de Engenharia de Fabricação, 2007, Estância de São Pedro. 4o COBEF - Congresso Brasileiro de Engenharia de Fabricação, 2007.

3. FERREIRA, João Carlos Espíndola; ZATTAR, Izabel Cristina; GRANADO, João Gabriel G.; SOUSA, Carlos Humberto Barreto de. **Integration between Process Planning and Scheduling Using Feature-Based Time-Extendend Negotiations Protocols in a Multi-agent System.** In: FAIM 2007 - Flexible Automation and Intelligent Manufacturing, 2007, Philadelphia. FAIM 2007 - Flexible Automation and Intelligent Manufacturing, 2007.

4. ZATTAR, Izabel Cristina; FERREIRA, João Carlos Espíndola ; GRANADO, João Gabriel G. ; SOUSA, Carlos Humberto Barreto de . **Integrating Manufacturing Process Planning with Scheduling via Operation-Based Time-Extended Negotiation Protocols.** In: 14th ISPE - International Conference on Concurrent Engineering, 2007, São José dos Campos. 14th ISPE - International Conference on Concurrent Engineering, 2007.

5. ZATTAR, Izabel Cristina; FERREIRA, João Carlos Espíndola; GRANADO, João Gabriel G.; SOUSA, Carlos Humberto Barreto de. **An Heterarchical Multi-Agent System for the Integration of Process Planning and Scheduling Using Operation-Based Time-Extended Negotiation Protocols.** In: 4th International CIRP-Sponsored Conference on Digital Enterprise Technology (DET2007), 2007, Bath, Inglaterra. 4th International CIRP-Sponsored Conference on Digital Enterprise Technology (DET2007). Paris, França, 2007.

6. ZATTAR, Izabel Cristina; FERREIRA, João Carlos Espíndola; GRANADO, João Gabriel G.; SOUSA, Carlos Humberto Barreto de. **Planning and Scheduling Integration Using Operation-Based Time Extended Negotiation Protocols.** In: 19th International Conference on Production Research, 2007, Valparaiso. ICPR-19, 2007.

---

---

## 9.4 Recomendações para trabalhos futuros

Ainda que esta tese apresente conceitos novos e interessantes para o estudo da correlação entre planos de processo alternativos, representados sob a forma de diferentes flexibilidades, e o seqüenciamento da produção em um ambiente de *layout* funcional, e ainda que os resultados alcançados encorajem o uso de sistemas de simulação baseados em agentes para este fim, há alguns aspectos que demandam novas pesquisas e desenvolvimentos. Alguns destes aspectos são identificados a seguir:

**1. Investigar os efeitos relacionados à consideração de novos eventos em tempo real.**

Neste trabalho foram considerados alguns eventos relacionados ao tempo real de manufatura, em geral aqueles relativos à negociação entre ordens e recursos fabris. Um exemplo é a ocupação de cada recurso, a qual resulta no tempo de espera para processamento, considerado no tempo de proposta. Porém outros aspectos relacionados ao tempo real se mostram igualmente interessantes de serem abordados, como os distúrbios no chão-de-fábrica decorrentes de quebras de máquinas ou entrada de ordens urgentes, entre outros. Estes novos eventos exigiriam estudos relacionados, por exemplo, a adoção de estratégias de re-agendamento.

**2. Estudar a possibilidade de interligar os agentes do modelo desenvolvido a softwares de auxílio à tomada de decisão, como MRPs (Materials Requirements Planning) e ERPs (Enterprise Resources Planning).**

De acordo com MONOSTORI et al. (2006) o sucesso da integração em toda uma cadeia de produção se deve principalmente a disponibilidade e acuracidade das informações que podem ser compartilhadas pela mesma. Neste contexto, torna-se vital a preocupação da integração entre o modelo proposto com sistemas proprietários de auxílio à tomada de decisões. Somente a partir desta integração feita e validada é que se pode considerar ou não, sua aplicabilidade em um ambiente fabril real.

**3. Explorar e desenvolver novas combinações entre os agentes de ordem e recurso visando estender o modelo conceitual proposto a novos layouts fabris.**

**4. Investigar a possibilidade de utilização dos resultados encontrados para possíveis benchmarks.**

---

---

De acordo com BALCI (1994) a credibilidade de um modelo é julgada tendo em vista a definição do modelo e os objetivos do estudo. Já a credibilidade dos resultados é avaliada tendo em vista os resultados existentes, por exemplo, do sistema real modelado. Neste contexto, sugere-se a replicação dos resultados obtidos neste estudo da correlação entre planos de processo alternativos, representados sob a forma de diferentes flexibilidades, e o seqüenciamento da produção.

---

---

**REFERÊNCIAS**

AHMAD, N.; HAQUE, Al; HASIN A. A. Current Trend in Computer Aided Process Planning. In 7th Annual Paper Meet and 2th International Conference. **Proceedings...** 2001. p. 81-92.

ALDEA, A.; BANARES, A.; JIMENEZ, R.; MORENO, L.; MARTINEZ, J.; RIANO, D. The scope of application of multi-agent systems in the process industry: three case studies. **Expert Systems with Applications**, v.26, p.39-47, 2004.

AYTUG, H.; BHATTACHARY, S.; KOEHLER, G. J.; SNOWDON, J. A Review of Machine Learning in Scheduling. **IEEE Transactions on Engineering Management**, v.41, n.2, 1994.

BAKER, A. D. A Survey of Factory Control Algorithms which can be implemented in a Multi-Agent Heterarchy: Dispatching, Scheduling, and Pull. **Journal of Manufacturing Systems**, n.17, p.297-320, 1998.

BALCI, O. Validation, verification, and testing techniques throughout the life cycle of a simulation study. In: Winter Simulation Conference, 1994. **Proceedings...** 1994, Annals of Operations Research, p.121-173.

BALCI, O. Verification, validation, and certification of modeling and simulation applications. In: Winter Simulation Conference, New Orleans, Louisiana, USA, 2003. **Proceedings...** 2003, Annals of Operations Research, p.150-158.

BANKS J; GERSTEIN S; SEARLES SP. Modeling processes, validation, and verification of complex simulations: A survey. In: Balci O (ed) Methodology and validation. **Proceedings...** of the conference on methodology and validation, Orlando Fl, San Diego, CA, 1987, Society for Computer Simulation, pp. 13–18.

BAUER, A.; BOWDEN, R.; BROWNE, J.; DUGGAN, J.; LYONS, G. **Shop Floor Control Systems: from design to implementation**. New York: Chapman & Hall, 1991.

BEACH, R.; MUHLEMANN, A. P.; PRICE, D. H. R.; PATERSON, A.; SHARP, J. A. A review of manufacturing flexibility. **European Journal of Operational Research**, v.122, p.41-57, 2000.

---

---

BELLIFEMINE, Fabio; CAIRE, Giovanni; TRUCCO, Tiziana; RIMASSA, Giovanni; MUNGENAST, Roland. **JADE Administrators guide** – last update: 10 November 2006. JADE 3.4.1 Disponível em: <<http://jade.tilab.com/>>. Acesso em: 21 maio 2006.

BENGTSSON, J. Manufacturing flexibility and real options: **A review. International Journal Production Economics**, v.74, p.213-224, 2001.

BENJAAFAR, S.; RAMAKRISHNAN, R. Modeling, measurement and evaluation of sequencing flexibility in manufacturing systems. **International Journal of Production Research**, v.34, n.5, p.1195-1220, 1996.

BOND, A.H.; GASSER, L. **An Analysis of Problem and Research in DAI**, San Mateo, California: Morgan Kaufmann Publishers, 1988.

BRESCIANI, P.; PERINI, A.; GIORGINI, P.; GIUNCHIGLIA, F.; MYLOPOULOS, John. Tropos: An Agent-Oriented Software Development Methodology. **Autonomous Agents and Multi-Agent Systems**, v.8, n.3, p.203-236, 2004.

BROWNE, J.; DUBOIS, D.; RATHMILL, K.; SETHI, S. P.; STECKE, K. E. Classification of flexible manufacturing systems. **The FMS Magazine**, p.114-117, 1984.

BRUCCOLERI, MANFREDI; AMICO, Michele; PERRONE, Giovanni. Distributed intelligent control of exceptions in reconfigurable manufacturing systems. **International Journal of Production Research**, v.41, n.7, p.1393-1412, 2003.

BUTLER, J.; OHTSUBO, H. ADDYMS: Architecture for Distributed Dynamic Manufacturing Scheduling. **Artificial Intelligence Applications in Manufacturing**, p. 199-213, 1992.

BURKE, P.; PROSSER, P. The Distributed Asynchronous Scheduler. **Intelligent Scheduling**, M. Zweben and M.S. Fox, eds., Morgan Kaufmann, San Francisco, p. 309-339, 1994.

BUSSMANN, S.; JENNINGS, N. R.; WOOLDRIDGE, M. **Multiagent Systems for Manufacturing Control: a Design Methodology**. London: Springer, 2004. 288 p.

CAIRE, G.; LEAL, F.; RODRIGUES, **MESSAGE - Methodology for Engineering Systems of Software Agents**, 2001, 27 p. (Technical Report, Project P907, EUROSCOM, 2001).

---

---

CARDOSO, Henrique Lopes. **Integrating JADE and Jess**. Disponível em: [http://jade.tilab.com/doc/tutorials/jade-jess/jade\\_jess.html](http://jade.tilab.com/doc/tutorials/jade-jess/jade_jess.html)>. Acesso em: 18 dezembro 2007.

CARSON II, John S. Introduction to modeling and simulation. In: Winter Simulation Conference, Orlando, FL, USA ,2005. **Proceedings...** 2005, p.16-23.

CARTER, M. F. Designing Flexibility into Automated Manufacturing Systems. In: Second ORSA/TIMS Conference on Flexible Manufacturing Systems, 1986. **Proceedings...** Ann, Arbor; STECKE, K. E.; SURI, R. (Eds.), Elsevier, Amsterdam, The Netherlands, p.107-118, 1986.

CHATTERJEE, A.; COHEN, M. A.; MAXWELL, W. L. A Planning Framework for Flexible Manufacturing Systems. **WP#87-07-04**, University of Pennsylvania, Philadelphia, 1987.

CHEN, R.J.; CALANTONE, I.J.; CHUNG, C. H. The marketing-manufacturing interface and manufacturing flexibility. **Omega**, v.20, n.4, (1992), p.431-443, 1992.

CHUNG, C. H.; CHEN, I. J. A systematic assessment of the value of flexibility for an FMS. In: Conference on Flexible Manufacturing Systems, Cambridge, 1989. **Proceedings...** K. E. STECKE, S.; SURI, R. (Eds.), Elsevier Science Publishers, B. V., Amsterdam, The Netherlands, p.27-34, 1989.

CODD, E. F. A relational model of data for large shared data banks. **Communications of the ACM**, v.13, n.6, p.377-387, 1970.

CONWAY, R. W.; MAXWELL, W. L. L.; MILLER W. **Theory of scheduling**. New York: Dover, 2003, 294 p.

COSENTINO, M. From Requirements to Code with the PASSI Methodology. **Agent-Oriented Methodologies**, B. Henderson-Sellers and P. Giorgini (Editors). Pennsylvania: Idea Group Publishing, 2005, 413 p.

COSENTINO, M.; SABATUCCI, CHELLA A. Patterns reuse in the PASSI methodology. In:Fourth International Workshop Engineering Societies in the Agents World (ESAW'03) London, UK, 2003. **Proceedings...** London, UK, 2003.

COSENTINO, M.; SABATUCCI, L.; CHELLA. A Possible Approach to the Development of Robotic Multi-Agent Systems. In: IAT'03 - IEEE/WIC Conference on Intelligent Agent Technology. Halifax , Canada. **Proceedings...** 2003.

---

---

COSENTINO, M.; SABATUCCI, L.; CHELLA. **Patterns Reuse in the PASSI Methodology**. Engineering Societies in the Agents World, v.3071, p.294-310, 2004.

COSENTINO, Massimo; CHELLA, Antonio; Lo FASO, Umberto. Designing agent-based systems with UML. In: International Symposium on Robotics and Automation (ISRA'2000), Monterrey, MEXICO, 2000. **Proceedings...** 2000.

CRUCES, A. L. L.; ARRIAGA, F. D. Reactive Agent Design for Intelligent Tutoring Systems. **Cybernetics and Systems: An International Journal**, v.31, p.1-47, 2000.

DAVIDSSON, Paul. Multi Agent Based Simulation: Beyond Social Simulation. **Lecture Notes in Computer Science**, v.1979, p.141-155, 2001.

DAVIDSSON, Paul; PERSON, J. A.; HOLMGREN, J. On the Integration of Agent-Based and Mathematical Optimization Techniques. **Agent and Multi-Agent Systems: Technologies and Applications**, v.4496, p.1-10, 2007.

DE MELLO, Homem L. S.; SANDERSON, A.C. AND/OR graph representation of assembly plans. **Robotics and Automation, IEEE Transactions**, v.6, n.2, p.188-199, 1990.

DE OLIVEIRA, J. A. B. **Coalition based approach for shop floor agility**. Ph.D. dissertation, Universidade Nova de Lisboa, PORTUGAL, 2005.

DEITEL, H. M.; DEITEL, P. J. **Java Como Programar**. 4 ed. Porto Alegre: Bookman, 2005. 1386 p.

DENKENA, B., BATTINO, A., WOELK, P.O. Intelligent Software Agents as a Basis for Collaborative Manufacturing Systems. In: First IPROMS Virtual Conference, Elsevier Ltd, Oxford, UK, 2005. **Proceedings...** 2005, p.17-22.

DOMINIC, P. D. D.; KALIYAMOORTHY, S.; MURUGAN, R. A conflict-based priority dispatching rule and operation-based approaches to job shops. **International Journal of Advanced Manufacturing Technology**, v.24, p.76-80, 2004.

DUFFIE, N. A. Synthesis of Heterarchical Manufacturing Systems. **Computers in Industry**, v. 14, n. 1-3, p.167-174, 1990.

DUFFIE, N. A.; CHITTURI, R.; MOU, J. Fault-Tolerant Heterarchical Control of Heterogeneous Manufacturing System Entities. **Journal of Manufacturing Systems**, v.7, n.4, p.315-327, 1988.

---

---

EVERSHEIM, W.; MICHALAS, N. **Integration of process planning and production scheduling:** benefits and difficulties. Disponível em: <<http://www.3s.hmg.inpg.fr/ci/project/GAMA/01Eversheim.pdf>> Acesso em: 2 mai. 2008.

FERBER, J. Multi-agent systems: An introduction to Distributed Artificial Intelligence. London: Addison-Wesley, 1999.

FIPA. FOUNDATION FOR PHYSICAL AGENTS. **SC00001L:** Abstract Architecture Specification. Geneva, Switzerland, 2002. 75 p. Disponível em: <http://www.fipa.org/specs/fipa00001/SC00001L.pdf>>. Acesso em: 13 abril 2006.

FIPA. FOUNDATION FOR PHYSICAL AGENTS. **SC00037J:** Communicative Act Library Specification. Geneva, Switzerland, 2002. 45 p. Disponível em: <http://www.fipa.org/specs/fipa00037/SC00037J.pdf>>. Acesso em: 13 abril 2006.

FIPA. FOUNDATION FOR PHYSICAL AGENTS. **SC00061G:** Message Structure Specification. Geneva, Switzerland, 2002. 11 p. Disponível em: <http://www.fipa.org/specs/fipa00061/SC00061G.pdf>>. Acesso em: 8 março 2006.

FIPA. FOUNDATION FOR PHYSICAL AGENTS. **Working Draft:** Modeling Interaction Diagram. Geneva, Switzerland, 2003. 45 p. Disponível em: <http://www.auml.org/auml/documents/ID-03-07-02.pdf>>. Acesso em: 5 maio 2006.

FREITAS, Paulo J. de. **Introdução à modelagem e simulação de sistemas:** com aplicações em Arena. Florianópolis: Visual Books, 2001. 322 p.

FRIEDMAN-HILL, E. **Jess in Action: Rule-based Systems in Java.** CT, USA: Manning Publications, 2003.

GAO, Jie; GEN, Mitsuo; SUN, Linyan; ZHAO, Xiaohui. A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems. **Computers & Industrial Engineering**, v.53, p.149-162, 2007.

GENESERETH, Michael R.; KETCHPEL, Steven P. Software agents. **Communications of the ACM**, v.37, n.7, p.48-54, 1994.

GIL, Antonio C. **Como elaborar projetos de pesquisa.** 4. ed. São Paulo: Atlas, 2002. 175 p.

---

---

GJERDRUM, J.; SHAH, N.; PAPAGEORGIOU, L. G. A combined optimization and agent-based approach to supply chain modeling and performance assessment. **Production Planning and Control**, n.12, p.81-88, 2001.

GRAVES, Stephen C. A review of production scheduling. **Operations Research**, v.29, n.4, p.646-675, 1981.

GUPTA, Mahesh C.; GUPTA, Yash P.; KUMAR, Anup. Minimizing flow time variance in a single machine system using genetic algorithms. **European Journal of Operational Research**, v.70, n.3, p. 289-303, 1993.

HATVANY, J. Intelligence and Cooperation in Heterarchic Manufacturing systems. **International Journal on Robotics and Computer Integrated Manufacturing**, v.2, n.2, p.101-104, 1985.

HENDERSON-SELLERS, B.; GIORGINI, P. **Agent-Oriented Methodologies**. Pennsylvania: Idea Group Publishing, 2005, 413 p.

HOLLAND, John H. **Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence**. Massasuchetts: MIT Press Cambridge, 1992.

HORLING, B.; LESSER, V. A Survey of Multi-Agent Organizational Paradigms. *The Knowledge Engineering Review*, 2005

HUANG, S. H.; ZHANG, Hong-Chao; SMITH, Milton L. A progressive approach for the integration of process planning and scheduling. **IIE Transactions**, v. 27, n. 4, p. 444-455, 1995.

HUHNS, M. N.; STEPHENS, L. M. Multiagent Systems and Societies of Agents. In G. Weiss (Ed.), **Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence**, p. 79-120,1999.

HUTCHINSON, G. K. Flexibility Is Key to Economic Feasibility of Automating Small Batch Manufacturing. **Industrial Engineering**, v.16, n.6, p.77-86, 1984.

IGLESIAS, C. Á. **Definición de una Metodología para el Desarrollo de Sistemas Multiagente**. Tese de Doutorado, Universidade Politécnica de Madri. ESPANHA, 1998.

---

---

IMBERTI, L.; TOLIO, T. Manufacturing planner agents in network enterprises. **International Journal of Automotive Technology and Management**, v.3, n.3/4, p.315-327, 2003.

JAIKUMAR, R. Post-Industrial Manufacturing. **Harvard Business Review**, p.66-76, 1986.

JENNINGS, Nicholas R. An agent-based approach for building complex software systems, **Communications of the Acm**, v. 44, n 4, p.35-41, 2001.

**JESS the Rule Engine for the Java™ Platform.** Disponível em: <<http://herzberg.ca.sandia.gov/>>. Acesso em: 5 novembro 2007.

JONES, A. T., MCLEAN, C. R. A Proposed Hierarchical Control Model for Automated Manufacturing Systems. **Journal of Manufacturing Systems**, v.5, p.15-25, 1986.

KIM YK, PARK K, KO J. A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling, **Computers & Operations Research**, v.30, p.1151-1171, 2003.

KORTH, H. F.; SILBERSCHATZ, A. **Sistema de Banco de Dados**. 2. ed. São Paulo: Makron Books do Brasil , 754 p.

KOSTE, L. L., MALHOTRA, M. K. Trade-offs among the elements of flexibility: a comparison from the automotive industry. **Omega**, v. 28, n. 6, p. 693-710, 2000.

LAM, A. **Measurement of a Flexible Manufacturing System**. Team Project, University of Toronto, Toronto, Canada, 1988.

LENDERINK A.; KALS H. J. J. The integration of process planning and machine loading in small batch part manufacturing. **Robotics and computer-integrated manufacturing**, v. 10, n. 1-2, p. 89-98, 1993.

LEONG, G.K.; SNYDER, D.L.; WARD, P.T. Research in the process and content of manufacturing strategy. **Omega**, v.18, n.2, p. 109-122, 1990.

LIM, M.K. Lim, ZHANG, D.Z. An integrated agent-based approach for responsive control of manufacturing resources. **Computers and Industrial Engineering**, v. 46, n. 2, p.221-232, 2004.

---

---

LIN, G. Yuh-Jiun; SOLBERG J. J. Integrated Shop Floor Control Using Autonomous Agents. **IIE Transactions**, v. 24, n.3, p. 57 – 71, 1992.

LOACH, S. A. De; WOOD, Mark. Developing Multiagent Systems with agentTool. **Intelligent Agents VII Agent Theories Architectures and Languages**, v.1986, p.30-41, 2001.

LUGER, GEORGE F. **Inteligência Artificial**. 4 ed. São Paulo: Bookman, 2004. 776 p.

MACAL, Charles M.; NORTH, Michael J. Tutorial on agent-based modeling and simulation. In Winter Simulation Conference, Orlando, FL, USA ,2005. **Proceedings...** 2005, p.4-15.

MCGUIRE J. G.; KUOKKA D. R.; WEBER, Jay C.; TENENBAUM J. M.; GRUBER T. R.; OLSEN G. R. SHADE: Technology for Knowledge-based Collaborative Engineering. **Concurrent Engineering**, v.1, n.3, p.137-146, 1993.

MAGLICA, R. **On Programming and Control of Robots in Flexible Manufacturing Cells**. PhD Thesis, Chalmers University of Technology, SWEDEN, 1996.

MATURANA, F.; SHEN, W.; NORRIE, D. H. MetaMorph: An Adaptive Agent-Based Architecture for Intelligent Manufacturing. **International Journal of Production Research**, v.1, n. 3, 159-168, 1999.

MARIK, V.; McFARLANE, D. Industrial Adoption of Agent-Based Technologies. **IEEE Intelligent Systems**, v.20, n.1, p.27 - 35, 2005.

MISER, Hugh J. A foundational concept of science appropriate for validation in operational research. **European Journal of Operational Research**, v.66, n.2, p.204-215, 1993.

MONOSTORI, L.; VÁNCZA, J.; KUMARA, S. R. T. Agent-Based Systems for Manufacturing. **In: CIRP Annals - Manufacturing Technology**, v.55, n.2, p. 697-720, 2006.

MOON, Chiung; SEO, Yoonho. Evolutionary algorithm for advanced process planning and scheduling in a multi-plant. **Computers & Industrial Engineering**, v. 48, n. 2, p. 311-325, 2005.

MORIARTY, David E.; MIIKKULAINEN, Risto. Forming Neural Networks Through Efficient and Adaptive Coevolution. **Evolutionary Computation**, v.5, n.4, p.373-399, 1997.

---

---

MySQL. **About MySQL AB**. Disponível em: <<http://www.mysql.com/company/>>. Acesso em: 10 março 2006.

NWANA, H. S., LEE, L. C.; JENNINGS, N. R. Coordination in software agent systems. **The British Telecom Technical Journal**, v.14, n.4, p.79-88, 1996.

OLIVEIRA, E., FISCHER, K.; STEPANKOVA, O. Multi-agent systems: which research for which applications. **Robotics and Autonomous Systems**, v. 7, n. 1-2, p.91-106, 1998.

ONORI, M. **The Robot Motion Module: A Task-Oriented Robot Programming System for FAA Cells**. Unpublished PhD thesis, The Royal Institute of Technology, STOCKHOLM, 1996.

ONORI, M.; LANGBECK, B.; Grondahl, P. The Mark III Flexible Automatic Assembly Cell. **International Journal on Robotics and Computer Integrated Manufacturing**, v.13, n. 3, p.193-202, 1997.

OUELHADJ, D., HANACHI, C.; BOUZOUIA, B. Multi-agent system for dynamic scheduling and control in manufacturing cells. In: IEEE International Conference on Robotics and Automation, Belgium, 1998. **Proceedings...** p. 1256-1262, Belgium, 1998.

OUELHADJ, D. A multi-agent system for the integrated dynamic scheduling of steel production. P.hD. Thesis, University of Nottingham, ENGLAND, 2003.

PACE, Dale K. Ideas About Simulation Conceptual Model Development. **Johns Hopkins Apl Technical Digest**, v.21, n.3, p.327-336, 2000.

PADGHAM, Lin; WINIKOFF, Michael. Prometheus: A Methodology for Developing Intelligent Agents, **Agent-Oriented Software Engineering III**, v.2585, p.174-185, 2003.

PADGET, J.; SHEHORY, O.; PARKES, D.; SADEH, N.; WALSH, W. E. (Eds.). **Agent-Mediated Electronic Commerce IV: Designing Mechanisms and Systems**, v.LNAI 231, Heidelberg: Springer-Verlag, 2002.

PAOLUCCI, M.; SACILE R. **Agent-Based Manufacturing and Control Systems**. London: CRC Press, 2005, 269 p.

PARKER, R. P., WIRTH, A. Manufacturing flexibility: measures and relationships. **European journal of operational research**, v. 118, n. 3, p. 429-449, 1994.

---

---

PARUNAK, Van Dyke; BAKER, Albert D.; CLARK, Steven J. The AARIA agent architecture: From manufacturing requirements to agent-based system design. **Integrated Computer-Aided Engineering**, v.8, n.1, p.45-58, 1998.

PECHOUCEK, M.; HODIK, J.; BECVAR, P.; VOKRINEK, J.; POSPISIL, J. ExPlanTech and ExtraPlant: multi-agent technology for production planning, simulation and extra-enterprise collaboration. **International Journal of Computer Systems Science & Engineering**, v.5, p. 357-367, 2005.

PENG, Y.; FININ, T.; LABROU, Y.; CHU, B.; LONG, J. A Multi-Agent System for Enterprise Integration. **International Journal of Agile Manufacturing**, 1998.

PINEDO, M. L. Planning and scheduling in manufacturing and services. New York: Springer, 2005, 506 p.

PINEDO, M. L. **Scheduling: Theory, algorithms and systems**. Englewood Cliffs, New Jersey: Prentice-Hall, 1995, 378 p.

POTTER, Mitchel A. **The design and analysis of a computational model of cooperative coevolution**. Ph.D. dissertation, George Mason University, USA, 1997.

QIAN, Y.; LI, X.; JIANG, Y.; WEN, Y. An expert system for real-time fault diagnosis of complex chemical process. **Expert Systems with Applications**, v.24, p.425-432, 2003.

RANTA, J. Interactive Analysis of FMS Productivity and Flexibility. **Proceedings...** W.P.-88-098, IIASA, A-2361 Laxenburg, Austria, 1988.

RASHID, Yahya; UEDA, Kanji; HATONO, Itsuo. From static to sensor scheduling: distributed environment. International, **Journal of Manufacturing Technology and Management**, v.3, n.6, p.586- 599, 2001.

REZENDE, S. O. (org). **Sistemas inteligentes: fundamentos e aplicações**. Barueri: Manole, 2005. 525 p.

RUSSEL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 2. ed. Englewood: Prentice Hall, 2003, 1057 p.

SADEH, Norman. **Look-Ahead Techniques For Micro-Opportunistic Job Shop Scheduling**. Ph.D. dissertation, Carnegie Mellon University, USA, 1991.

---

---

SEARLE, J. R. (1969). **Speech Acts - An Essay in the Philosophy of Language**. Cambridge: Cambridge University Press, 1999.

SETHI, A. K.; SETHI, S. P. Flexibility in Manufacturing: A Survey. **The International Journal of Flexible Manufacturing Systems**, v.2, p.289-328, 1990.

SHAFAEI R.; BRUNN P. Workshop scheduling using practical (inaccurate) data Part 3: A framework to integrate job releasing, routing and scheduling functions to create a robust predictive schedule. **International Journal of Production Research**, v.38, n. 1, pp.85-99, 2000.

SHAH, J. J.; MÄNTIYLA, M. **Parametric and Featured-Based CAD/CAM: concepts, techniques and application**. New York: John Willey, 1995, 619p.

SHEN W, WANG L, HAO Q. “Agent-Based Distributed Manufacturing Process Planning and Scheduling: A State-of-Art Survey”, **IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews** 2006; 36: 563-571.

SHEN, W. Distributed manufacturing scheduling using intelligent agents. **IEEE Expert/Intelligent Systems**, v. 17, n. 1, p. 88-94, 2002.

SHEN, W.; NORRIE, D. H.; BARTHES, J. P. **Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing**. London, U.K.: Taylor & Francis, 2001.

SHEN, W.; NORRIE, D. H. Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey. **Knowledge and Information Systems, an International Journal**, v.1, n.2, p.129-156, 1999.

SHEWCHUK, J. P.; MOODIE, C. L. Definition and Classification of Manufacturing Flexibility Types and Measures. **The International Journal of Flexible Manufacturing Systems**, v. 10, p.325-349, 1998.

SMITH, R. G. The Contract Net Protocol - High-Level Communication and Control in a Distributed Problem Solver. **IEEE Transactions on Computers**, v.29, n.12, p.1104-1113, 1980.

SILVA, Edna Lúcia da; MENEZES, Estera Muszkat. **Metodologia da pesquisa e elaboração de dissertação**. Florianópolis: UFSC/PPGEP/LED, 2000, 118 p.

---

---

SON, Y. K.; PARK, C. S. Economic Measure of Productivity, Quality and Flexibility in Advanced Manufacturing Systems. **Journal of Manufacturing Systems**, v.6, n.3, p.193-207, 1987.

SOUSA P.; RAMOS, C. A Dynamic Scheduling Holon for Manufacturing Orders. **Journal of Intelligent Manufacturing**, v.9, n.2, p.107-112, 1997.

TAN, W.; KHOSHNEVIS, B. Integration of process planning and scheduling – a review. **Journal of Intelligent Manufacturing**, v.11, n.1, p.51-63, 2000. London.

TSUBONE, H.; HORIKAWA, M. A comparison between machine flexibility and routing flexibility. **The International Journal of Manufacturing Systems**, v.11, p.83-101, 1999.

UML. **Unified Modeling Language**. Disponível em: <<http://uml.org>>. Acesso em: 12 de junho de 2006.

USHER, J. M. Evaluating the impact of alternative plans on manufacturing performance. **Computers & Industrial Engineering**, v. 45, p. 585-596, 2003.

WANG, C.; SHEN, W.; GHENNIWA, H. An adaptive negotiation framework for agent-based dynamic manufacturing scheduling, In: IEEE SMC 2003, v. 2, Washington DC, USA, 2003. **Proceedings...** 2003, p. 1211–1216.

WAZLAWICK, Raul S. **Análise e projeto de sistemas de informação orientados a objetos**. 2. Ed. Rio de Janeiro: Elsevier, 2004. 298 p.

WEISS, G. Multi-agent systems: **A modern approach to distributed artificial intelligence**. London, England: The MIT Press, 1999, 654 p.

WONG, T.N.; LEUNG, C.W.; MAK, K.L; FUNG. R.Y.K. Dynamic shopfloor scheduling in multi-agent manufacturing systems. **Expert Systems with Applications**, v.31, p.486-494, 2006a.

WONG, T.N.; LEUNG, C.W.; MAK, K.L; FUNG. R.Y.K. An agent-based negotiation approach to integrate process planning and scheduling. **International Journal of Production Research**, v. 44, n. 7, p.1331-1351, 2006b.

WOOLDRIDGE, M. **An Introduction to Multiagent Systems**. West Sussex, England: John Willey, 2002, 348 p.

---

---

WOOLDRIDGE, M.; JENNINGS, N.; KINNY, D. A methodology for agent-oriented analysis and design. In: International Conference on Autonomous Agents, 3th, 1999, Seattle, WA, USA. **Proceedings...** Autonomous Agents Online Proceedings, p. 69-76. 1999. Disponível em: <<http://sigart.acm.org/proceedings/agents97/>>. Acesso em: 12 setembro 2007.

WOOLDRIDGE, M.; JENNINGS. Agent theories, architectures, and languages: A survey. **Intelligent Agents**, v.890, p.1-39, 1995.

YEO, Keun K. **A set of data for the integration of process planning and job shop scheduling**, 2003. Disponível em: <<http://syslab.chonnam.ac.kr/links/data-pp&s.doc>>. Acesso em: 13 de setembro de 2006.

YILMAZ, L. Validation and verification of social process within agent-based computational organization models. **Computational & Mathematical Organization Theory**, v.12, p.283-312, 2006.

YILMAZ, L.; MACKENZIE, Garth R.; SCHULMEYER, G. Gordon. Verification Technology Potential with Different Modeling and Simulation Development and Implementation Paradigms. In: Foundations for V&V in the 21st Century Workshop, The Johns Hopkins University Applied Physics Laboratory Laurel, Maryland, USA, 2002. **Proceedings...** 2002, p.A1-A40.

ZHANG, Q.; VONDEREMBSE, Mark A. Manufacturing flexibility: defining and analyzing relationships among competence, capability, and customer satisfaction. **Journal of Operations Management**, v.21, p.173-191, 2003.

---