

Universidade do Minho

Escola de Engenharia

Departamento de Informática



Agentes Inteligentes em Sistemas Holónicos de Produção

Paulo Alexandre Gandra de Sousa

Tese de Doutoramento

2000



Universidade do Minho

Escola de Engenharia

Departamento de Informática



Instituto Politécnico do Porto

Instituto Superior de Engenharia do Porto

Departamento de Engenharia Informática

Agentes Inteligentes em Sistemas Holónicos de Produção

Tese submetida à Universidade do Minho para obtenção do grau de Doutor em Informática,
elaborada sob a orientação do Doutor José Carlos Ferreira Maia Neves (Professor Catedrático da
Universidade do Minho) e do Doutor Carlos Fernando Silva Ramos (Professor Coordenador do
Instituto Superior de Engenharia do Porto/Instituto Politécnico do Porto)

Paulo Alexandre Gandra de Sousa



Dezembro, 2000

1998 – 2000
Paulo Sousa
Departamento de Engenharia Informática
Instituto Superior de Engenharia do Porto (ISEP/IPP)
Rua Dr. António Bernardino de Almeida, 431
4200-072 PORTO
Portugal
Tel. +351 228 340 500
Fax +351 228 325 219

Correio electrónico: Paulo.Sousa@dei.isep.ipp.pt
Correio electrónico (alternativo): psousa@altavista.net
URL: <http://www.dei.isep.ipp.pt/~psousa>
(1.50 – U:02:14/01)

Para a Teresa,
Para os meus Pais

Prefácio

O Mestre nunca lhe daria um desejo sem que lhe desse também o poder de o realizar. Você talvez tenha de trabalhar muito para o concretizar.

Richard Bach, “Ilusões: As Aventuras de um Messias Relutante”.

Esta dissertação apresenta o resultado do meu trabalho de investigação na área dos Sistemas Holónicos de Produção, realizado entre Setembro de 1997 e Dezembro de 2000, nas instalações do Centro CIM do Instituto Superior de Engenharia do Instituto Politécnico do Porto (ISEP/IPP), sob a co-orientação do Professor Maia Neves da Universidade do Minho (UM) e do Professor Carlos Ramos do ISEP/IPP.

Desde cedo no meu percurso profissional (anterior à via académica) lidei com sistemas distribuídos, tendo inclusive desenvolvido um projecto interessantíssimo para a área de seguros que implicou a integração multiplataforma e o desenvolvimento de infra-estruturas de comunicação e distribuição. Quando mais tarde iniciei a minha colaboração com o ISEP/IPP como assistente, tentei, naturalmente, manter as afinidades com os sistemas distribuídos e nesse sentido, tomei conhecimento dos sistemas multiagente. Ao mesmo tempo, o Centro de CIM do ISEP/IPP ganhava um certo fôlego e um novo ânimo devido a novos projectos de investigação então aprovados.

Foi no âmbito de um desses projectos, mais concretamente, o projecto Europeu n.º 21955 do programa ESPRIT – *Intelligent Manufacturing Systems Working Group* (IMS-WG), que me vi envolvido a convite do Professor Carlos Ramos. Nas reuniões desse grupo de trabalho pude conviver com investigadores de outros países e entrar em contacto com os seus trabalhos. Dessas reuniões saí também com conhecimentos mais consolidados sobre os Sistemas Inteligentes para Produção e mais especificamente sobre os Sistemas Holónicos de Produção. Estes conceitos apresentavam características de sistemas distribuídos e de sistemas inteligentes, pelo que (embora não tocassem directamente a outra menina dos meus olhos, isto é, a Computação Gráfica) me pareceram interessantes como tema a abordar num futuro trabalho de pós-graduação.

Iniciei então o meu Doutoramento em 1997, tendo frequentado a parte lectiva do Mestrado em Informática da UM onde pude aprofundar alguns conhecimentos teóricos que mais tarde me foram úteis na elaboração do trabalho. O ISEP/IPP beneficiou de uma acção de formação avançada ao abrigo do programa PRODEP, podendo por conseguinte dispensar-me de serviço

docente. Desenvolvi então, durante três anos, o presente trabalho, aproveitando para participar em conferências onde trocava ideias e tomava conhecimento de outros trabalhos.

Três anos é um período de tempo relativamente longo e no qual muita coisa acontece. Dando razão à máxima filosófica “*nenhum homem é uma ilha*”, também a Vida é mais que o trabalho, e como tal, tive que encontrar o equilíbrio entre o dever profissional e o dever familiar, conjugando ainda o meu círculo de amigos. Penso que consegui encontrar esse equilíbrio, dando atenção àqueles que me são mais queridos sempre que precisaram de mim, sem por isso descuidar o trabalho. Este documento escrito dentro do prazo é para mim prova disso mesmo.

É com prazer que agora, volvidos cerca de três anos e meio, observo o trabalho feito, o suor escorrido, os desânimos e as alegrias. Especialmente, sinto a alegria daqueles que me são chegados pela minha própria alegria de acabar. É um alívio que sinto no final do trabalho, embora tenha a sensibilidade de que ainda há muito para fazer, e espere vir a poder fazer uma parte desse muito que ainda falta.

Porto, 18 de Dezembro de 2000

Paulo Sousa

Agradecimentos

Tenho muito que agradecer a várias pessoas pela ajuda e disponibilidade que mostraram ao longo destes três anos de realização dos trabalhos de Doutoramento. Começo, obviamente, por agradecer aos meus orientadores, o professor Maia Neves e o professor Carlos Ramos.

Um agradecimento especial é devido ao Professor Carlos Ramos, que como Director do Centro de CIM do ISEP/IPP proporcionou as condições necessárias para a elaboração deste trabalho, com apoio material na cedência de equipamento e apoio monetário para as diversas deslocações efectuadas. Também na sua qualidade de Coordenador do Grupo de Disciplinas de Engenharia da Programação, em 1997, depositou confiança em mim tendo apresentado a minha candidatura ao Conselho Científico do ISEP/IPP para as acções de formação do programa PRODEP.

Aproveito para agradecer ao PRODEP pelo apoio concedido à instituição de forma a suportar o pagamento das propinas de Doutoramento e a dispensa de serviço docente. Gostaria também de aproveitar a ocasião para agradecer ao Conselho Científico do ISEP/IPP pelos subsídios concedidos para deslocações a conferências onde fui apresentando o trabalho elaborado. Também no âmbito financeiro agradeço à Fundação Luso-Americana para o Desenvolvimento, ao Programa ESPRIT da União Europeia, à Junta Nacional de Investigação Científica e à Fundação para a Ciência e a Tecnologia, pelo financiamento de projectos de investigação aos quais estive ligado e que suportaram parte das despesas de deslocações efectuadas.

Agradeço também aos colegas do Departamento de Engenharia Informática do ISEP/IPP e aos colegas do Centro CIM do ISEP/IPP, em especial àqueles que se encontravam na mesma altura em processo de Doutoramento, pela ajuda e intercâmbio de ideias e material para a elaboração do trabalho e deste documento. Um agradecimento especial ao João Rocha, ao Lino Ferreira e ao João Pinho pela cedência de algumas imagens, e ao Miguel Losa pelo apoio no acesso aos meios de impressão. Além dos colegas de trabalho, agradeço também ao Johan dos Santos e ao Jan Peeters, alunos ao abrigo do programa ERASMUS que contribuíram para este trabalho desenvolvendo alguns módulos do protótipo.

Aos meus amigos de longo data, sem nenhuma ordem em particular, o Alexandre, o Miranda, o Joaquim Miguel, o Rui, o Marcelo e o Paulo, que foram perguntando pelo trabalho e suportaram as minhas ausências, bem como as minhas presenças(!).

Não podia deixar de agradecer às pessoas mais importantes da minha vida. À minha família, pais, irmãs, irmão, cunhadas e cunhados, sobrinhas e sobrinhos, pelo apoio incondicional que sempre me deram. Sei que estão orgulhosos de mim por ter concluído mais esta fase e este trabalho é em parte para vós. Especialmente quero agradecer aos meus pais por terem suportado

os encargos dos meus estudos e pela confiança que me incutiram ao longo dos meus anos de vida, sei que é a vós que devo o facto de ser aquilo que sou hoje – *Muito Obrigado Pai e Mãe.*

Em especial tenho de agradecer à Teresa, companheira, esposa e amiga, que conforme prometido me apoiou nos bons e nos maus momentos, suportou as minhas faltas de atenção para ela e para a nossa casa e que me foi ajudando dentro do que lhe era possível. Foi também ela a minha conselheira gráfica. Por todo o amor e carinho; por toda a confiança em mim depositada; pela ajuda e motivação; pela companhia – *Obrigado Amor.*

Por último, gostaria de estender os meus agradecimentos a todos aqueles de uma forma ou de outra (fornecendo ideias e/ou criticando), foram ajudando anonimamente nas inúmeras discussões ao longo destes três anos.

A todos, os meus sinceros agradecimentos.



Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA E DA TECNOLOGIA

Resumo

A sociedade tem sofrido constantes mutações desde a sua génese, facto acentuado na última metade do século XX, onde a criação de uma sociedade de informação (com uma preponderância de empregos no sector de serviços), alterou o *status quo* originando também grandes alterações à orgânica das empresas do sector produtivo.

De um ponto de vista tecnológico, verificou-se que o conceito de Produção Integrada por Computador enferma de várias limitações, nomeadamente no que toca a problemas de excessiva rigidez e centralização, pelo que novas abordagens para a solução do problema têm sido equacionadas com vista a colmatar tais falhas e criar produtos mais adequados ao contexto socioeconómico actual e futuro. Essencialmente, defende-se que os sistemas produtivos devem possuir atributos tais como, a distribuição, a descentralização, a autonomia e a agilidade. Adicionalmente, a capacidade de representar a incompletude do conhecimento real nos sistemas artificiais foi também identificada como um dos atributos essenciais a serem exibidos pelos sistemas produtivos. Nesse contexto, os Sistemas Holónicos de Produção e os Sistemas de Produção Baseados em Agentes apresentam-se como alternativas promissoras.

Assim, e para ultrapassar estes desafios, equacionou-se uma solução baseada nos *Sistemas Holónicos de Produção*, recorrendo aos *Sistemas Multiagente* e à *Programação em Lógica Estendida*, onde as principais entidades intervenientes no processo produtivo são modeladas como holons, cada uma contribuindo com uma pequena parcela da funcionalidade geral do sistema. Para suportar os conceitos desenvolvidos com o presente trabalho foi desenvolvido um sistema denominado *Fabricare*. Este sistema é composto por diversos holons que cooperam entre si utilizando uma variante estendida do *Protocolo de Rede de Contrato*. A finalidade da cooperação entre holons tem a ver com o escalonamento dinâmico de tarefas industriais. O sistema proposto é comparado do ponto de vista estrutural com trabalhos anteriores e, quando possível, são comparados os resultados obtidos, para uma melhor avaliação do seu desempenho.

Este trabalho apresenta como principais contribuições (i) a definição e especificação de uma arquitectura holónica para produção; (ii) a representação e identificação de cenários de informação incompleta na produção; (iii) uma extensão ao protocolo de rede de contrato; (iv) um algoritmo distribuído para escalonamento; e (v) o desenvolvimento de um sistema experimental.

Palavras Chave: Sistemas Multiagente; Sistemas Holónicos; Produção; Sistemas Inteligentes para Produção; Escalonamento Dinâmico; Programação em Lógica Estendida; Informação Incompleta.

Abstract

Society has experienced evolving mutations since the dawn of ages, especially on the last half of the twentieth century. The birth of an information society with a relevant number of jobs in the services sector has changed the *status quo*, and demanded enormous changes to the manufacturing enterprises' organisation.

From a technological point of view, it was observed that Computer Integrated Manufacturing (CIM) poses several drawbacks, namely excessive rigidity and centralization. New approaches have been proposed in order to solve these problems, providing systems that are more adequate to today's and future social-economic context. Essentially, these new approaches advocate a set of attributes such as distribution, decentralisation, autonomy and agility. Moreover, the ability to represent the incompleteness of real world's knowledge in artificial worlds has also been identified as an essential attribute. To this point, Holonic Manufacturing Systems and Multi-Agent System seem as a promising approach.

In order to overcome these challenges, a solution based on *Holonic Manufacturing Systems* has been proposed using *Multi-Agent Systems* and *Extended Logic Programming*. In this solution the main entities in the manufacturing process are modelled as holons, each one contributing with a small parcel of the overall system's functionality. To demonstrate the previous concepts a prototype system has been developed, named *Fabricare*. The *Fabricare* system is composed of several holons that cooperate among themselves using an extension to the *Contract Net Protocol*. The goal of this cooperation is the dynamic scheduling of manufacturing orders. The system is compared from a structural point of view with previous research works.

The main contributions of this work are (i) the definition of a holonic architecture for manufacturing enterprises; (ii) the representation and identification of scenarios with incomplete information in manufacturing; (iii) an extension to the contract net protocol; (iv) a distributed scheduling algorithm; and (v) a prototype system.

Keywords: Multi-Agent Systems; Holonic Systems; Manufacturing; Intelligent Manufacturing Systems; Dynamic Scheduling; Extended Logic Programming; Incomplete Information.

Índice

Prefácio	vii
Agradecimentos.....	ix
Resumo.....	xi
Abstract	xiii
Índice.....	xv
Índice de Figuras.....	xxi
Índice de Tabelas.....	xxv
Notação e Terminologia.....	xxvii

DISSERTAÇÃO

CAPÍTULO 1 INTRODUÇÃO _____ **1**

1.1 Enquadramento	2
1.2 Questões e Hipótese do Trabalho.....	3
1.3 Justificação da Investigação.....	5
1.4 Paradigmas Utilizados.....	6
1.5 Objectivos, Delimitação e Pressupostos	9
1.6 Metodologia	11
1.7 Contribuições e Originalidades.....	12
1.8 Estrutura e Resumo da Dissertação.....	14

CAPÍTULO 2 SISTEMAS DE PRODUÇÃO _____ **19**

2.1 Introdução	20
2.2 Conceitos de Produção.....	21
2.3 Tipos de Produção.....	23
2.3.1 Função de Quantidade/Variedade.....	23
2.3.2 Relação com o Inventário	26
2.4 Funções da Actividade de Fabrico	27
2.4.1 Gestão da Produção	28
2.4.2 Projecto	28
2.4.3 Planeamento de Processos	29
2.4.4 Planeamento de Produção.....	31
2.4.5 Escalonamento e Balanceamento.....	32
2.4.6 Execução, Inspeção e Controlo de Qualidade.....	33

2.5	Produção Integrada por Computador.....	34
2.5.1	Introdução ao Conceito CIM.....	34
2.5.2	Suporte Tecnológico à Produção.....	35
2.5.2.1	Sistemas Flexíveis de Fabrico.....	37
2.5.2.2	Sistemas de Execução da Produção & Controlo.....	38
2.5.2.3	Apoio Computadorizado às Funções de Fabrico.....	39
2.5.3	Desafios à Implementação de CIM.....	41
2.6	Resumo do Capítulo.....	42

CAPÍTULO 3 A PRODUÇÃO E A ‘NOVA ECONOMIA’ _____ **45**

3.1	Introdução.....	46
3.2	Evolução dos Mercados e da Produção.....	48
3.2.1	Enquadramento Histórico.....	48
3.2.2	Tendências Observadas.....	50
3.3	O Contexto da ‘Nova Economia’.....	51
3.3.1	Força Laboral.....	53
3.3.2	Responsabilidade Social e Ambiental.....	54
3.3.3	O Conhecimento como Activo da Empresa.....	55
3.3.4	Dinâmica, Reactividade e Organização.....	56
3.3.4.1	Dinamismo, Globalização e Competição.....	56
3.3.4.2	Cadeia de Valor.....	57
3.3.4.3	Agilidade.....	59
3.3.5	Revolução nas Tecnologias de Informação.....	60
3.4	Produção: Presente e Futuro.....	61
3.4.1	Problemas com o ‘Presente’ da Produção.....	61
3.4.2	O ‘Futuro’ da Produção.....	63
3.4.3	Propriedades Desejáveis.....	65
3.5	Resumo do Capítulo.....	69

CAPÍTULO 4 SISTEMAS HOLÓNICOS DE PRODUÇÃO _____ **71**

4.1	Introdução.....	72
4.2	Sistemas Holónicos.....	73
4.2.1	As Origens.....	73
4.2.2	Características.....	76
4.3	A Produção e os Sistemas Holónicos.....	79
4.3.1	As Origens.....	79
4.3.2	Características de Sistemas Holónicos de Produção.....	79

4.3.3	Relacionamento com Requisitos Impostos pela ‘Nova Economia’	82
4.3.4	Metodologias Similares	83
4.3.4.1	Fábrica Fractal	83
4.3.4.2	Produção Biónica.....	84
4.3.4.3	O Que Há de Novo Nestas Abordagens	84
4.4	Mecanismos de Prototipagem	86
4.4.1	Sistemas Baseados em Agentes	86
4.4.1.1	Agentes	86
4.4.1.2	Sistemas Multiagente.....	89
4.4.1.3	Interacção.....	91
4.4.1.4	Redes de Contrato.....	93
4.4.1.5	Aplicabilidade.....	94
4.4.2	Programação em Lógica Estendida.....	95
4.4.2.1	Introdução	96
4.4.2.2	Extensões à Programação em Lógica	98
4.4.2.3	Aplicabilidade.....	101
4.5	Estado da Arte.....	102
4.5.1	Trabalhos de Referência	102
4.5.1.1	AARIA	102
4.5.1.2	HMS “ <i>Testbed</i> ”	103
4.5.1.3	PROSA	104
4.5.1.4	Gou & Luh.....	105
4.5.2	Outros Trabalhos.....	105
4.5.3	Sinopse.....	107
4.6	Resumo do Capítulo.....	109

CAPÍTULO 5 MODELO PROPOSTO: ESPECIFICAÇÃO _____ 111

5.1	Introdução	112
5.2	Arquitectura Proposta	115
5.2.1	Arquitectura Holónica do Sistema	115
5.2.2	Arquétipos de Holons e Holarquias	120
5.2.2.1	Holons.....	120
5.2.2.2	Holarquias.....	122
5.3	Especificação dos Holons	124
5.3.1	Serviço de Directório	124
5.3.2	Holon de Produto	125
5.3.3	Holon de Tarefa	127

5.3.4	Holon de Recurso	129
5.3.5	Holon de Compra	131
5.3.6	Holon de Fornecedor	133
5.3.7	Holon de Venda.....	134
5.3.8	Holon de Cliente.....	136
5.4	Tratamento de Informação Incompleta.....	137
5.4.1	Representação Utilizada	138
5.4.1.1	Informação Negativa Explícita	138
5.4.1.2	Nulos do Tipo Desconhecido	138
5.4.1.3	Nulos do Tipo Desconhecido de um Conjunto de Valores	139
5.4.1.4	Nulos do Tipo Não Permitido	139
5.4.1.5	Meta-Interpretador	140
5.4.2	Casos Identificados	141
5.4.2.1	Arquétipo dos Holon	141
5.4.2.2	Holon de Serviços de Directório	142
5.4.2.3	Holons de Produto.....	143
5.4.2.4	Holons de Tarefa	144
5.4.2.5	Holons de Recurso	145
5.4.2.6	Holons de Compra	146
5.4.2.7	Holons de Fornecedor	147
5.4.2.8	Holon de Gestão de Compras.....	148
5.4.2.9	Holons de Venda	149
5.4.2.10	Holons de Cliente.....	151
5.4.2.11	Holon de Gestão de Vendas	151
5.5	Resumo do Capítulo	152

CAPÍTULO 6 MODELO PROPOSTO: OPERAÇÃO 155

6.1	Introdução.....	156
6.2	Processo de Negociação	158
6.2.1	Descrição.....	159
6.2.1.1	Anúncio de Tarefa.....	159
6.2.1.2	Requisição de Serviços	161
6.2.1.3	Influência Directa.....	162
6.2.1.4	Influência Inversa e Proposta	163
6.2.1.5	Contratação	164
6.2.2	Tratamento de Conflitos.....	166
6.2.3	Tratamento de Excepções.....	168

6.2.4	Análise da Complexidade do Protocolo.....	169
6.3	Funcionamento dos Holons.....	178
6.3.1	Arquétipo de Holon	179
6.3.2	Serviço de Directório	181
6.3.3	Holon de Produto	183
6.3.4	Holon de Escalonamento	185
6.3.5	Holon de Tarefa	187
6.3.6	Holon de Recurso.....	195
6.3.7	Relação com o Método Original de Escalonamento.....	205
6.4	Sistema Desenvolvido.....	207
6.4.1	Introdução	208
6.4.2	Núcleo de Holons.....	209
6.4.3	Ferramentas de Exploração.....	213
6.4.4	Programas Adicionais.....	214
6.4.5	Comentários ao Desenvolvimento do Protótipo <i>Fabricare</i>	217
6.5	Comparação com Trabalhos Relacionados	218
6.5.1	Âmbito e Estrutura (Entidades Modeladas).....	220
6.5.2	Socialização	221
6.5.3	Tratamento de Informação Incompleta.....	222
6.5.4	Procedimento de Escalonamento	223
6.6	Resumo do Capítulo.....	224
CAPÍTULO 7 CONCLUSÕES		225
7.1	Síntese da Dissertação.....	226
7.2	Contribuições, Conclusões e Originalidades.....	231
7.3	Tese	239
7.4	Limitações e Trabalho Futuro	241
BIBLIOGRAFIA		247
APÊNDICES		
APÊNDICE A EXPERIÊNCIAS E RESULTADOS		271
A.1	Experiências do Método Original	271
A.1.1	Exemplo 1	272
A.1.2	Exemplo 2.....	273
A.1.3	Exemplo 3.....	274

A.1.4 Exemplo 4	275
A.2 Experiências <i>Fabricare</i>	277
A.2.1 Exemplo 1	278
A.2.2 Exemplo 2	279
A.2.3 Exemplo 3	280
A.2.4 Exemplo 4	281
A.2.5 Exemplo 5	282
 APÊNDICE B MODELO MATEMÁTICO PARA ANÁLISE DA COMPLEXIDADE DE PRCPR	285
B.1 Simplificações	285
B.2 Folha de Cálculo	286
 APÊNDICE C SÍTIOS WWW	291
C.1 Pessoas	291
C.2 Instituições e Grupos I&D	293
C.3 Projectos	295
C.4 Miscelânea	296
 ÍNDICE REMISSIVO	299

Índice de Figuras

<i>Figura 2.1 – Sistema de produção</i>	20
<i>Figura 2.2 – Job shop: (a) agrupamento por função; (b) ilhas multi-funcionais</i>	25
<i>Figura 2.3 – Relação entre os vários tipos de produção</i>	26
<i>Figura 2.4 – Funcionalidades de uma empresa de produção industrial</i>	27
<i>Figura 2.5 – Exemplo de produto: (a) desenho técnico; (b) imagem tridimensional</i>	29
<i>Figura 2.6 – Exemplo de produto: identificação das operações</i>	30
<i>Figura 2.7 – Exemplo de produto: (a) grafo de precedências; (b) plano gerado</i>	30
<i>Figura 2.8 – Ciclo do planeamento da produção</i>	31
<i>Figura 2.9 – Exemplo de produto: escalonamento das operações</i>	33
<i>Figura 2.10 – Componentes CIM</i>	37
<i>Figura 2.11 – Exemplo de um sistema FMS</i>	38
<i>Figura 2.12 – Exemplo de arquitectura de controlo de um FMS</i>	39
<i>Figura 3.1 – Agilidade, adaptabilidade e flexibilidade</i>	67
<i>Figura 3.2 – Sistemas adaptáveis vs. não adaptáveis</i>	68
<i>Figura 4.1 – Exemplos de sistemas holónicos: (a) militar; (b) biológico</i>	76
<i>Figura 4.2 – Exemplo de um sistema holónico</i>	78
<i>Figura 4.3 – Relação entre comportamentos</i>	92
<i>Figura 4.4 – Protocolo de rede de contrato</i>	94
<i>Figura 4.5 – Algoritmo do meta-interpretador para lógica estendida</i>	100
<i>Figura 4.6 – Arquitectura do sistema AARIA</i>	103
<i>Figura 4.7 – Arquitectura de referência PROSA</i>	104
<i>Figura 4.8 – Arquitectura holónica de Gou & Luh</i>	105
<i>Figura 5.1 – Principais áreas funcionais do sistema</i>	112
<i>Figura 5.2 – Caso de utilização para a função ‘Produção’</i>	113
<i>Figura 5.3 – Um possível cenário de exploração</i>	114
<i>Figura 5.4 – Entidades intervenientes no sistema e seus relacionamentos</i>	115
<i>Figura 5.5 – Arquitectura holónica do sistema de fabrico</i>	116
<i>Figura 5.6 – Arquitectura holónica do sistema comercial</i>	116
<i>Figura 5.7 – Arquétipo de um agente</i>	120
<i>Figura 5.8 – Holarquias</i>	123
<i>Figura 5.9 – Diagrama de classes para holons de produto</i>	126
<i>Figura 5.10 – Exemplo de representação de um plano</i>	127
<i>Figura 5.11 – Diagrama de classes para holons de tarefa</i>	128
<i>Figura 5.12 – Diagrama de classes para holons de recurso</i>	130

<i>Figura 5.13 – Diagrama de classes para holons de compras</i>	132
<i>Figura 5.14 – Diagrama de classes para holons de fornecedor</i>	134
<i>Figura 5.15 – Diagrama de classes para holons de vendas</i>	135
<i>Figura 5.16 – Diagrama de classes para holons de cliente</i>	137
<i>Figura 6.1 – Escalonamento</i>	156
<i>Figura 6.2 – Escalonamento com operações antecessoras de maior duração</i>	157
<i>Figura 6.3 – Escalonamento com recursos alternativos</i>	157
<i>Figura 6.4 – Escalonamento com operações paralelas</i>	158
<i>Figura 6.5 – Anúncio de tarefa</i>	159
<i>Figura 6.6 – Requisição de recursos</i>	161
<i>Figura 6.7 – Influência directa</i>	162
<i>Figura 6.8 – Influência inversa e ofertas</i>	163
<i>Figura 6.9 – Conclusão da negociação</i>	165
<i>Figura 6.10 – Alteração ao PRCPR: tratamento de conflitos</i>	166
<i>Figura 6.11 – Alterações ao PRCPR: (a) fim de negociação; (b) abandono de recursos</i>	167
<i>Figura 6.12 – Exemplo de avaria de um recurso</i>	169
<i>Figura 6.13 – Exemplos de planos para estudo da complexidade do protocolo RCPR</i>	172
<i>Figura 6.14 – Alterações na fase de influência directa, influência inversa e proposta</i>	174
<i>Figura 6.15 – Evolução do número de mensagens com o aumento de operações</i>	176
<i>Figura 6.16 – Tamanho total de mensagens</i>	178
<i>Figura 6.17 – Algoritmo de processamento de mensagens de gestão de holarquias</i>	179
<i>Figura 6.18 – Algoritmo de funcionamento do holon de serviços de directório</i>	181
<i>Figura 6.19 – Algoritmo de funcionamento de um holon de produto</i>	184
<i>Figura 6.20 – Algoritmo de funcionamento do holon de escalonamento</i>	186
<i>Figura 6.21 – Algoritmo de funcionamento do holon de escalonamento: libertar tarefas</i>	187
<i>Figura 6.22 – Diagrama de estado para holons de tarefa</i>	188
<i>Figura 6.23 – Algoritmo de funcionamento de um holon de tarefa</i>	188
<i>Figura 6.24 – Algoritmo de funcionamento de um holon de tarefa: inicialização</i>	189
<i>Figura 6.25 – Algoritmo de funcionamento de holon de tarefa: negociação</i>	191
<i>Figura 6.26 – Algoritmo de funcionamento de holon de tarefa: avaliação de propostas</i>	192
<i>Figura 6.27 – Algoritmo de funcionamento de um holon de tarefa: recombinar</i>	194
<i>Figura 6.28 – Diagrama de estado para holons de tarefa: acompanhamento</i>	195
<i>Figura 6.29 – Agenda de um recurso e agenda livre</i>	196
<i>Figura 6.30 – Limitação de agendas: (a) inferior; (b) superior</i>	196
<i>Figura 6.31 – Influência de agendas: (a) directa; (b) inversa</i>	197
<i>Figura 6.32 – Diagrama de estado para holons de recurso: planeamento</i>	197
<i>Figura 6.33 – Algoritmo de planeamento de holons de recurso</i>	198

<i>Figura 6.34 – Protocolo RCPR nos holons de recurso</i>	199
<i>Figura 6.35 – Diagrama de estado para holons de recurso: execução</i>	204
<i>Figura 6.36 – Comportamentos de escalonamento</i>	206
<i>Figura 6.37 – Vista geral do protótipo Fabricare</i>	208
<i>Figura 6.38 – Metodologia Fabricare</i>	209
<i>Figura 6.39 – Interface do protótipo Fabricare: holons de suporte</i>	210
<i>Figura 6.40 – Interface do protótipo Fabricare: holons de tarefa</i>	211
<i>Figura 6.41 – Interface do protótipo Fabricare: holons de recurso</i>	212
<i>Figura 6.42 – Interface do protótipo Fabricare: aplicação “Deploy”</i>	213
<i>Figura 6.43 – Interface do protótipo Fabricare: aplicação “Lançador de Tarefas”</i>	214
<i>Figura 6.44 – Interface do protótipo Fabricare: aplicação “Configuration Designer”</i>	214
<i>Figura 6.45 – Interface do protótipo Fabricare: aplicação “Control Panel”</i>	215
<i>Figura 6.46 – Interface do protótipo Fabricare: escalonamento de um Holon de Recurso</i>	216
<i>Figura 6.47 – Interface do protótipo Fabricare: escalonamento de um Holon deTarefa</i>	216
<i>Figura 6.48 – Interface do protótipo Fabricare: aplicação “Product Builder”</i>	217
<i>Figura 7.1 – Arquitectura holónica proposta (actividades de fabrico)</i>	231
<i>Figura 7.2 – Protocolo RCPR</i>	234
<i>Figura 7.3 – Vista geral do protótipo Fabricare</i>	237
<i>Figura 7.4 – Exemplos de meta-regras para selecção de recursos</i>	242
<i>Figura 7.5 – Reescalonamento para operação prioritária: (a) pedido; (b) re-arranjo</i>	244
<i>Figura A.1 – Plano utilizado nas experiências</i>	271
<i>Figura A.2 – Resultado da experiência original nº 1</i>	272
<i>Figura A.3 – Resultado da experiência original nº 2</i>	273
<i>Figura A.4 – Resultado da experiência original nº 3</i>	275
<i>Figura A.5 – Planos utilizado na experiência nº 4</i>	275
<i>Figura A.6 – Resultado da experiência original nº 4</i>	277
<i>Figura A.7 – Resultado da experiência Fabricare nº 1</i>	279
<i>Figura A.8 – Resultado da experiência Fabricare nº 2</i>	280
<i>Figura A.9 – Resultado da experiência Fabricare nº 3</i>	281
<i>Figura A.10 – Resultado da experiência Fabricare nº 4</i>	282
<i>Figura A.11 – Resultado da experiência Fabricare nº 5</i>	283
<i>Figura B.1 – Plano exemplo utilizado para análise de complexidade</i>	285
<i>Figura B.2 – Folha de cálculo utilizada para análise de complexidade do PRCPR</i>	288

Índice de Tabelas

<i>Tabela 3.1 – Comparação entre a ‘Velha’ e a ‘Nova Economia’</i>	46
<i>Tabela 3.2 – A indústria na ‘Velha’ e na ‘Nova Economia’</i>	47
<i>Tabela 3.3 – A força laboral na ‘Velha’ e na ‘Nova Economia’</i>	53
<i>Tabela 4.1 – Relacionamento entre propriedades desejadas e sistemas holónicos</i>	82
<i>Tabela 4.2 – Sinopse dos trabalhos relacionados</i>	107
<i>Tabela 5.1 – Legenda da arquitectura</i>	117
<i>Tabela 5.2 – Holons por holarquia</i>	117
<i>Tabela 5.3 – Características dos holons da arquitectura proposta</i>	118
<i>Tabela 5.4 – Constantes utilizadas na arquitectura proposta</i>	121
<i>Tabela 5.5 – Informação incompleta no arquétipo dos holons</i>	141
<i>Tabela 5.6 – Informação incompleta no holon de serviços de directório</i>	142
<i>Tabela 5.7 – Informação incompleta nos holons de produto</i>	143
<i>Tabela 5.8 – Informação incompleta nos holons de tarefa</i>	144
<i>Tabela 5.9 – Informação incompleta nos holons de recurso</i>	145
<i>Tabela 5.10 – Informação incompleta nos holons de compra</i>	146
<i>Tabela 5.11 – Informação incompleta nos holons de fornecedor</i>	147
<i>Tabela 5.12 – Informação incompleta no holon de gestão de compras</i>	148
<i>Tabela 5.13 – Informação incompleta nos holons de venda</i>	149
<i>Tabela 5.14 – Informação incompleta nos holons de cliente</i>	151
<i>Tabela 5.15 – Informação incompleta no holon de gestão de vendas</i>	152
<i>Tabela 6.1 – Valores de $\chi(\mathcal{P})$ e M_{prepr} para os planos exemplo</i>	172
<i>Tabela 6.2 – Valores de M_{prepr}^2 para planos exemplo</i>	175
<i>Tabela 6.3 – Transições na máquina de estados do PRCPR nos holons de recurso</i>	200
<i>Tabela 6.4 – Comparação entre o sistema Fabricare e trabalhos relacionados</i>	219
<i>Tabela A.1 – Condições de teste da experiência original nº 1</i>	272
<i>Tabela A.2 – Resultados da experiência original nº 1</i>	272
<i>Tabela A.3 – Condições de teste da experiência original nº 2</i>	273
<i>Tabela A.4 – Resultados da experiência original nº 2</i>	273
<i>Tabela A.5 – Condições de teste da experiência original nº 3</i>	274
<i>Tabela A.6 – Resultados da experiência original nº 3</i>	274
<i>Tabela A.7 – Condições de teste da experiência original nº 4</i>	275
<i>Tabela A.8 – Resultados da experiência original nº 4</i>	276
<i>Tabela A.9 – Condições de teste das experiências Fabricare</i>	278
<i>Tabela A.10 – Resultados da experiência Fabricare nº 1</i>	278

<i>Tabela A.11 – Resultados da experiência Fabricare n° 2.....</i>	<i>279</i>
<i>Tabela A.12 – Resultados da experiência Fabricare n° 3.....</i>	<i>280</i>
<i>Tabela A.13 – Resultados da experiência Fabricare n° 4.....</i>	<i>281</i>
<i>Tabela A.14 – Resultados da experiência Fabricare n° 5.....</i>	<i>282</i>
<i>Tabela B.1 – Fórmulas da folha de cálculo.....</i>	<i>286</i>
<i>Tabela C.1 – Legenda utilizada nos web sites.....</i>	<i>291</i>

Notação e Terminologia

Notação Geral

A notação utilizada ao longo do documento segue a convenção apresentada a seguir:

- *Texto em itálico* – para palavras em língua estrangeira (*i.e.*, Inglês ou Latim), exemplo: “[...] utilizando abordagens *ad hoc*”. Também utilizado para dar ênfase a um determinado termo; exemplo: “Uma fórmula *válida* é aquela [...]”;
- *Texto em negrito* – utilizado para realçar um conceito/palavra no meio de um parágrafo; exemplo: “Um **Recurso** tem várias [...]”;
- *Texto de cor clara* – para indicar uma referência bibliográfica; exemplos: “[Sousa e Ramos, 1997]”; “Bradshaw (1997) apresenta [...]”; Também utilizado para endereços *web*; exemplo: “<http://www.dei.isep.ipp.pt/>”
- *Parágrafo indentado em itálico* – citações bibliográficas;
- *Parágrafo com espaçamento simples em itálico* – para os extractos de código ou extensões de predicados lógicos; exemplo: “*cliente(301, ‘Silva’, ‘João’)*”;
- *Letras em caligrafia e letras do alfabeto Grego* – utilizadas para referenciar elementos (*e.g.*, conjuntos, tuplos) nas descrições formais; exemplo “seja \mathcal{A} o conjunto[...]”; exemplo: “O tuplo $\langle \beta, \delta \rangle$ representa [...]”.

Ao longo do texto são utilizadas várias letras do alfabeto grego em algumas fórmulas e descrições formais, pelo que em seguida se apresenta uma relação desses símbolos por forma a facilitar a sua leitura [Larouse, 1995].

Maiúscula	Minúscula	Nome	Equivalência
A	α	Alfa	a
B	β	Beta	b
Γ	γ	Gama	g
Δ	δ	Delta	d
E	ε	Épsilon	e
Z	ζ	Zeta	dz
H	η	Eta	e (longo)
Θ	θ	Téta	th

Maiúscula	Minúscula	Nome	Equivalência
I	ι	Iota	i
K	κ	Kapa	k
Λ	λ	Lambda	l
M	μ	Mi	m
N	ν	Ni	n
Ξ	ξ	Csi	ks ou x
O	ο	Omicron	o (breve)
Π	π	Pi	p
P	ρ	Ró	r
Σ	σ	Sigma	s
T	τ	Tau	t
Υ	υ	Ípsilon	u
Φ	φ ou ϕ	Fi	f
X	χ	Qui	kh
Ψ	ψ	Psi	ps
Ω	ω	Omega	o (longo)

Os diagramas apresentados ao longo da dissertação para descrever conceitos e funcionalidades seguem a notação UML (*Unified Modelling Language*) [Rumbaugh *et al.*, 1997], que é uma notação normalizada para o desenho de diagramas orientados a objectos. A notação UML surgiu da combinação da notação Booch [Booch, 1991] com a notação OMT (*Object Modelling Technique*) [Rumbaugh *et al.*, 1991] e a notação de Jacobson [Jacobson, 1992] (em particular no que se refere aos Casos de Utilização).

Terminologia

Ao longo do texto utilizou-se a terminologia Portuguesa; no entanto, e embora já exista algum consenso na passagem de alguns termos técnicos para Português, o seu uso ainda não é generalizado. Ao longo do texto optou-se, por conseguinte, por traduzir os termos e conceitos, mas manter os acrónimos em Inglês, pois são estes os mais conhecidos. Por exemplo, ‘Projecto Assistido por Computador’ é certamente reconhecido como ‘*Computer Aided Design*’, no entanto, o acrónimo PAC nem sempre é reconhecido como sinónimo de CAD.

Para facilitar a leitura, apresenta-se em seguida a tradução para a língua Portuguesa de alguns termos e expressões Anglo-Saxónicas.

Termo Anglo-Saxónico	Tradução para Português
<i>Art-to-Part</i>	Arte-até-Peça
<i>Bottleneck</i>	“Gargalo” de eficiência
<i>Holarchic</i>	Holárquico
<i>Holarchy</i>	Holarquia
<i>Knowledge worker</i>	Trabalhador Perito
<i>Mass Customisation</i>	Personalização em Massa
<i>Responsiveness</i>	Reactividade
<i>Stock</i>	Existência
<i>Template</i>	Modelo
<i>Time-to-market</i>	Tempo de colocação no mercado

Certos termos e expressões não foram traduzidos devido à dificuldade de instalação do termo ou expressão na língua de Camões. Essas excepções são apresentadas de seguida:

- *Batch* – (informática) aplicação de execução não interactiva;
- *Blackboard* – (informática) “quadro negro”, área de memória partilhada para escrita e leitura entre vários processos;
- *Buffer* – (produção) espaço de armazenamento à entrada de um recurso onde podem ser colocados material e componentes a utilizar numa operação;
- *Business-to-business* – comércio electrónico entre empresas (*e.g.*, fornecedores de componentes, subcontratados)
- *Business-to-consumer* – comércio electrónico entre empresas e consumidores finais;
- *Downsizing* – diminuição do número de efectivos de uma empresa;
- *Empowerment* – (gestão) aumento da autonomia e responsabilidade de decisão por parte dos trabalhadores;
- *Flow shop* – a melhor tradução seria “oficina de fluxo”;
- *Hardware* – não foi traduzido devido à utilização corrente do termo em Inglês;
- *Holon* – a melhor tradução seria “holão”;
- *Job shop* – a melhor tradução seria “oficina de processo”;

- *Just-in-time* – a melhor tradução seria “mesmo-a-tempo”;
- *Marketing* – não foi traduzido devido à utilização corrente do termo em Inglês;
- *Master/slave* – “mestre/escravo”, forma de organização em que um processo controla (comanda) o funcionamento de outro;
- *Offline* – “fora de linha”, normalmente utilizado para situações em que um determinado procedimento é efectuado sem ligação em tempo real ao sistema físico que representa;
- *Online* – “em linha”, normalmente utilizado para situações em que um determinado procedimento é efectuado com ligação em tempo real ao sistema físico que representa;
- *Outsourcing* – (gestão) contratação de determinadas capacidades a empresas terceiras;
- *Script* – (informática) um programa interpretado que permite a extensão de funcionalidades de uma aplicação ou então que define o controlo de funcionamento de uma aplicação;
- *Setup* – (produção) actividade de preparação de um recurso para a execução de determinada operação (*e.g.*, carregar as ferramentas necessárias);
- *Shell* – (informática) a consola de comandos do sistema operativo;
- *Sockets* – (informática) um mecanismo de comunicação entre processos;
- *Software* – não foi traduzido devido à utilização corrente do termo em Inglês;
- *Web site* – não foi traduzido devido à utilização corrente do termo em Inglês.

Predicados Primitivos e Auxiliares

<i>atribui(V, Expr)</i>	Calcula o valor de <i>Expr</i> e coloca o resultado em <i>V</i>
<i>comprimento(L, N)</i>	Determina o número de elementos (<i>N</i>) da lista <i>L</i>
<i>enviar_msg(Destino, Msg)</i>	Envia uma mensagem de conteúdo <i>Msg</i> ao holon <i>Destino</i>
<i>inserir(P)</i>	Aumenta a base de conhecimento do holon, acrescentando-lhe o termo <i>P</i>
<i>juntar(L1, L2, LR)</i>	Concatenação das listas <i>L1</i> e <i>L2</i> formando a lista <i>LR</i>
<i>membro(X, L)</i>	Determina se <i>X</i> é um dos elementos da lista <i>L</i>
<i>nivelar(Lista-de-Listas, Lista)</i>	Obtenção de uma lista de elementos a partir de uma lista de listas
<i>receber_msg(Emissor, Msg)</i>	Recebe uma mensagem com conteúdo <i>Msg</i> do holon <i>Emissor</i>

$retira_elemento(X, L, LR)$	Constrói a lista LR a partir da lista L retirando-lhe o elemento X
$str_term_cat(A1, A2, AR)$	Concatenação dos átomos $A1$ e $A2$ formando o átomo AR
$todas_as_soluções(T, G, R)$	Obtenção da lista R com membros do tipo T , contendo todas as soluções da questão G à base de conhecimento

Símbolos

\mathcal{L}	Linguagem lógica
KB	Base de conhecimento de um holon
$KB\epsilon$	Base de conhecimento mutuamente exclusivo
$KB\pi$	Base de conhecimento sobre situações proibidas
\mathcal{D}	Conjunto de identificadores de nulos desconhecidos
Q	Uma questão sobre um predicado $q(a_0, \dots, a_n)$
a_i	Um argumento de uma questão ou de um predicado
P	Um axioma da base de conhecimento na forma $p(a_0, \dots, a_n)$
ω	Valor nulo do tipo desconhecido
δ	Valor nulo do tipo não permitido
\mathcal{P}	Plano de produção de um produto
$\chi(\mathcal{P})$	Número de combinações existentes para um plano \mathcal{P} tendo em conta os recursos alternativos para cada operação do plano
M_{prcpr}	Número de mensagens trocadas numa execução do Protocolo de Rede de Contrato com Propagação de Restrições (PRCPR)
$\psi(\mathcal{P})$	Número de mensagens trocadas durante a fase influência directa ou inversa na execução do PRCPR
\emptyset	Conjunto vazio
M_{prcpr}^2	Número de mensagens trocadas numa execução da variante do protocolo RCPR
T_{inf}	Tamanho de uma mensagem de influência no PRCPR
T_{bid}	Tamanho de uma mensagem de proposta no PRCPR
$T_{inf}^2(i)$	Tamanho de uma mensagem de influência na variante do PRCPR
$T_{bid}^2(i)$	Tamanho de uma mensagem de proposta na variante do PRCPR

$T_{prcpr inf+bid }$	Tamanho total das mensagens de influência e proposta no PRCPR
$T_{prcpr inf+bid }^2$	Tamanho total das mensagens de influência e proposta na variante do PRCPR
$\frac{T_{prcpr inf+bid }^2}{T_{prcpr inf+bid }}$	Ganho relativo da variante do PRCPR em relação ao PRCPR original
γ	Identificação de um holon
\mathcal{M}	Uma mensagem recebida
Act	A identificação da operação a executar na lista de membros
η	Identificação de um holon
H	a identificação de uma holarquia
\mathcal{H}	Lista de habilidades de um holon
\hat{h}	Identificação de uma habilidade
\mathcal{V}	Valor de verdade (<i>verdadeiro, falso, desconhecido</i>) da prova de uma questão
$Crit$	Critério de optimização de um plano de produção
LR	O conjunto de recursos libertados
HE	Identificação do Holon de Escalonamento
\mathcal{R}	Conjunto de recursos necessários para a execução de um plano \mathcal{P}
Λ	Conjunto de recursos contactados para cada operação do plano \mathcal{P}
Θ	Conjunto de propostas recebidas
op	Uma operação
\mathcal{T}	Lista de contratos de operações de um holon de tarefa
Φ	Conjunto de propostas avaliadas
κ	Um critério de avaliação de propostas
Π	O conjunto de planos já utilizados
\mathcal{A}	Agenda de actividades de um holon

Acrónimos

AGV	<i>Automated Guided Vehicle</i> (Veículo Autónomo)
BC	Base de Conhecimento
BMS	<i>Bionic Manufacturing System</i> (Sistema de Produção Biónico)

CAD	<i>Computer Aided Design</i> (Projecto Assistido por Computador)
CAE	<i>Computer Aided Engineering</i> (Engenharia Assistida por Computador)
CAM	<i>Computer Aided Manufacturing</i> (Produção Assistida por Computador)
CAPP	<i>Computer Aided Process Planning</i> (Planeamento de Processos Assistido por Computador)
CAQC	<i>Computer Aided Quality Control</i> (Controlo da Qualidade Assistido por Computador)
CIM	<i>Computer Integrated Manufacturing</i> (Produção Integrada por Computador)
CNC	<i>Computer Numerical Control</i> (Controlo Numérico Computadorizado)
CRP	<i>Capacity Requirements Planning</i> (Planeamento de Necessidades de Capacidade)
CSCW	<i>Computer Supported Co-operative Work</i> (Trabalho Cooperativo Apoiado por Computador)
EDI	<i>Electronic Data Interchange</i> (Troca de Dados Electrónicos)
ERP	<i>Enterprise Resource Planning</i> (Planeamento de Recursos Empresariais)
EUA	Estados Unidos da América
EV	Empresa Virtual
FF	<i>Fractal Factory</i> (Fábrica Fractal)
FMS	<i>Flexible Manufacturing System</i> (Sistema Flexível de Fabrico)
HMS	<i>Holonic Manufacturing System</i> (Sistema Holónico de Produção)
IA	Inteligência Artificial
IAD	Inteligência Artificial Distribuída
IMS	<i>Intelligent Manufacturing System</i> (Sistema Inteligente de Produção)
ISO	<i>International Organization for Standardization</i> (Organização Internacional para a Padronização)
KQML	<i>Knowledge Query and Manipulation Language</i> (Linguagem de Interrogação e Manipulação de Conhecimento)
MES	<i>Manufacturing Execution System</i> (Sistema de Execução da Produção)
MRP1	<i>Material Requirements Planning</i> (Planeamento de Necessidades de Materiais)
MRP2	<i>Manufacturing Resources Planning</i> (Planeamento de Recursos de Produção)
n.a.	Não aplicável
NC	<i>Numerical Control</i> (Controlo Numérico)

PC	<i>Personal Computer</i> (Computador Pessoal)
PL	Programação em Lógica
PLE	Programação em Lógica Estendida
PMF	Pressuposto do Mundo Fechado
PMP	Plano Mestre de Produção
PNM	Plano de Necessidades de Material
PPC	<i>Production Planning and Control</i> (Planeamento e Controlo da Produção)
PRC	Protocolo de Rede de Contrato
RCPR	Rede de Contrato com Propagação de Restrições
RDP	Resolução Distribuída de Problemas
SMA	Sistema Multiagente
SP	SICStus PROLOG
SPBA	Sistema de Produção Baseado em Agentes
TI	Tecnologia(s) de Informação
UE	União Europeia
UML	<i>Unified Modelling Language</i> (Linguagem de Modelação Universal)
URSS	União das Repúblicas Socialistas Soviéticas
VB	Visual Basic
WWW	<i>World Wide Web</i>

DISSERTAÇÃO

CAPÍTULO 1

INTRODUÇÃO



Os mercados mudam, os gostos mudam, como tal, também as empresas e indivíduos que escolhem competir nesses mercados têm de mudar.

An Wang (1920 – 1990), ex-presidente da Wang Laboratories Inc., inventor do Dispositivo de Controlo de Transferência de Impulsos.

A sociedade está em constante mudança, tendo sido especialmente no século XX que se verificaram algumas das mais radicais transformações [Drucker, 1994]. Essa mudança tem tido implicações a vários níveis, forçando os cidadãos e empresas a adaptarem-se de forma a prosperarem. Ao longo da evolução da sociedade, e consequentemente da economia, de base Agrícola para base Industrial e mais tarde para base Informacional, os Sistemas Produtivos têm acompanhado as mudanças observadas em seu redor [RCBI, 1998]. Será no entanto possível continuar a acompanhar o ritmo vertiginoso destas transformações?

Como resultado da maior concorrência (devido ao excesso de capacidade de fabrico e à globalização) os produtores viram-se forçados a aumentar a diversidade de produtos e reduzir os custos, o que teve como consequência o aumento da complexidade dos processo de fabrico e tornou as empresas mais susceptíveis de serem afectadas por distúrbios vindos do exterior

[Albayrak e Bussmann, 1996b] [Bussmann, 1998]. Com todas as convulsões sociais ocorridas nos últimos tempos e tendo em conta os mais diversos avanços tecnológicos, novos desafios são colocados às empresas em geral e ao sector produtivo em particular.

Na senda de resolver tais desafios, apresenta-se com este trabalho o sistema experimental *Fabricare* baseado nos paradigmas dos sistemas holónicos, dos sistemas multiagente e da programação em lógica estendida.

Antes, porém, de descrever o sistema, convém enquadrar o trabalho aqui realizado, pelo que neste capítulo será apresentada uma introdução ao tema da dissertação, começando-se por um enquadramento do trabalho e pela questão central em termos de investigação, à qual se segue uma hipótese proposta de resolução. Após se elaborar um pouco sobre a investigação a realizar, serão apresentados e justificados os diversos paradigmas para a resolução de problemas a utilizar no trabalho, bem como os objectivos a atingir e a metodologia de execução utilizada. Em seguida apresentam-se as contribuições e originalidades do trabalho. Finalmente, será referida a estrutura da dissertação.

1.1 Enquadramento

Nos últimos anos têm sido observadas várias tendências na sociedade e nos sistemas de produção [Solberg e Kashyap, 1993] [Kusiak, 1990] [NGM, 1997], entre as quais: a globalização de mercados; o aumento da personalização de produtos; o aumento da complexidade tecnológica; o aumento da concorrência; a diminuição dos ciclos de vida de produtos; e o aumento dos requisitos de qualidade. Adicionalmente, é necessário prestar especial atenção a questões ambientais e ao custo e consumo de energia.

Das mudanças observadas surge uma Nova Economia caracterizada por adjectivos como *Digital, Global, Competitiva, e Focada no Indivíduo* [Schonfeld, 1998] [Kelly, 1998].

Os avanços nas Tecnologias de Informação (TI) e a popularidade da Internet tornaram possível a existência de uma *Economia Digital* com troca electrónica de documentação de negócio em grande escala (*e.g.*, facturas, notas de encomenda, descrições de processos) [Schonfeld, 1998] [Korper *et al.*, 1999]. Além disso, também tornaram possível a prestação de serviços de “forma digital”, não convencional (*e.g.*, bancos ou corretoras na Internet).

A abertura de novos mercados na Europa de Leste, bem com a criação de um mercado único na União Europeia (UE); a diminuição de entraves fiscais e medidas proteccionistas às importações/exportações; e (mais uma vez) a ubiquidade da Internet criaram uma verdadeira *Economia Global* [Kelly, 1998] [Schwartz, 1997]. Serviços podem ser prestados de forma digital em qualquer parte do globo com acesso à Internet e os produtos podem ser entregues graças às

grandes multinacionais de distribuição e a acordos/parcerias entre produtores e distribuidores. As redes mundiais de telecomunicações fomentam o comércio global, aproximando-nos cada vez mais da Aldeia Global [Tapscott e Caston, 1993], ao mesmo tempo que o maior acesso à informação favorece os consumidores, permitindo-lhes estar melhor informados, tendo conseqüentemente mais escolhas e como tal mais poder [Langer, 1999].

A globalização tornou possível o acesso a novos mercados, mas também tornou possível o acesso de “jogadores” exteriores aos mercados domésticos de cada empresa [Tapscott e Caston, 1993]. Por outro lado, as redes de distribuição e (novamente) a Internet encurtaram distâncias, pelo que uma empresa a milhares de quilómetros é agora concorrente de qualquer empresa que competia apenas a nível regional. Esta nova configuração criou uma *Economia Extremamente Competitiva*, mais do que alguma vez foi visto. Além disso, o estado saudável das economias ocidentais favorecem a criação de novas empresas [Atkinson e Court, 1998], aumentando ainda mais o número de competidores.

A *Sociedade de Consumo Personalizado* é uma consequência da tentativa de atingir os gostos individuais de cada cidadão, recorrendo à *Personalização em Massa*; i.e., ao “*desenvolvimento, produção, marketing e distribuição de produtos e serviços personalizados de forma maciça*” [Pine *et al.*, 1999]. Ou seja, por outras palavras, pretende-se produzir inúmeros produtos e inúmeras variações desses produtos, de acordo com requisitos específicos dos clientes, em qualquer quantidade, a preços e a velocidades próprios dos da produção em massa [Davis, 1987] [Rabon e Scheller, 1997] [Martin, 1997] [Cox e Alm, 1998]. Neste cenário a tecnologia desempenha um papel fundamental [Schonfeld, 1998].

Uma empresa de produção necessita então de lidar com o aumento de concorrência, consequência da globalização de mercados, e com o aumento da diversidade de produtos, consequência do crescente foco no indivíduo em vez dos mercados massificados. Para qualquer empresa é fundamental encontrar o seu lugar no triângulo custo-inovação-qualidade, pois estes três vectores condicionam a competitividade da empresa. Nesse sentido, é importante considerar as alterações na sociedade e nas tecnologias como uma força motriz e não como uma condicionante, tentando daí gerar mais valias (*e.g.*, a Internet permite novas oportunidades de negócio – possibilidade de alcançar novos mercados – bem como as tecnologias de trabalho em grupo permitem a geração de ideias e o aumento da produtividade).

1.2 Questões e Hipótese do Trabalho

Conforme foi apresentado na secção anterior, as enormes mudanças verificadas na sociedade impõem novos desafios às empresas de Produção. Esta ‘Nova Economia’ implica uma mudança

continua, crescente dependência no conhecimento e inovação, e maiores parcerias (local e globalmente [NGM, 1997]), obrigando as empresas a mudar os seus procedimentos internos e filosofias de trabalho.

Para continuarem activas essas empresas necessitam adaptar-se e realçar as suas mais valias. Não só é necessário aumentar a *competitividade*, como também é necessário manter a *vitalidade*; *i.e.*, a capacidade de se adaptar a novos requisitos impostos pelos consumidores. Novas abordagens surgem, prometendo ajudar a empresa a ultrapassar os desafios que se lhe colocam, tendo como objectivo *a sobrevivência da empresa*.

Uma das principais questões colocadas por estas mudanças e pelo actual contexto socioeconómico é:

Questão – *Como será possível manter a vitalidade (e competitividade) de uma empresa de Produção na 'Sociedade de Consumo Personalizado' do século XXI?*

Obviamente, o âmbito desta questão ultrapassa em muito a área das Ciências da Computação (onde esta tese se insere) estando directamente relacionada com aspectos de gestão e organização de empresas. No entanto, de um ponto de vista mais tecnológico (especialmente, das tecnologias de informação), a questão pode ser limitada e reformulada para:

Questão – *Como será possível construir sistemas informáticos de apoio à produção capazes de responder às necessidades da empresa na 'Sociedade de Consumo Personalizado' do século XXI?*

Para tentar responder a estas questões sugere-se a seguinte hipótese cuja validade se irá equacionar ao longo do trabalho:

Hipótese – *A divisão dos Sistemas de Produção (mais concretamente dos sistemas informáticos de apoio à produção) em 'pequenas' unidades autónomas organizadas de acordo com a Teoria dos Sistemas Holónicos permite garantir a agilidade e vitalidade necessárias às empresas na nova sociedade.*

Essencialmente, defende-se que o novo contexto socioeconómico impõe novos requisitos e novos desafios aos sistemas de produção, e que os Sistemas Holónicos de Produção não só exibem tais características como permitem responder aos desafios impostos.

1.3 Justificação da Investigação

Embora nos foquemos e façamos algum abuso das tecnologias de informação e, mais concretamente, das tecnologias de Internet para suportar a chamada ‘economia de informação’, o planeta ainda necessita de agricultura e indústria para a criação de riqueza e para o fornecimento de necessidades humanas básicas [Tapscott e Caston, 1993].

Até agora, o conceito de Produção Integrada por Computador (CIM) tem sido promovido universalmente, no entanto, foram detectados alguns problemas no processo de implementação (*e.g.*, custo elevado) sendo opinião corrente que o CIM não é a resposta para os sistemas de produção do futuro, tendo sido, entre outros, identificados os seguintes problemas [Höpf, 1994] [Ueda, 1994] [Bongaerts *et al.*, 1995] [Parunak, 1996] [Tharumarajah *et al.*, 1996] [Gou e Luh, 1997] [Bussmann, 1998] [Kádár *et al.*, 1998] [Shen e Norrie, 1999]:

- *Inflexibilidade* – a arquitectura CIM é fixa, independentemente dos requisitos de produção, sendo difícil expandir ou reconfigurar um processo para a produção de novos produtos;
- *Falta de robustez* – a eficiência não é garantida fora da gama operatória pré-definida; além disso, a existência de um controlador central cria a existência de um único ponto de falha;
- *Falta de adaptabilidade* – no que se refere a avarias ou ao mau funcionamento dos recursos;
- *Dificuldade de manutenção* – devido à falta frequente de dados por parte das máquinas.

Um dos principais problemas das implementações CIM é a sua estrutura hierárquica com controlo centralizado que não se adequa à produção de lotes quase unitários em mercados altamente dinâmicos e em constante mudança [Kádár *et al.*, 1998].

Adicionalmente, a Produção mudou radicalmente no decurso dos últimos anos e novas alterações vão continuar a acontecer no futuro [Parker, 1997] [CVM, 1999] [Hunt, 1989] [Kidd, 2000], sendo factor crítico para o sucesso, a necessidade de ‘excelência’ na produção como resultado da criação de mercados globais [IMS, URL] [Barram, 1994]. Além disso, a Produção é actualmente (e continuará a ser) um dos principais geradores de riqueza, sendo de extrema

importância para manter esses mesmos níveis de riqueza [ISC, 1994] [IMS, URL] [CVM, 1999] [Wyns, 1999] e estabelecer uma base sólida para o crescimento económico no futuro [IMS, URL].

Todas estas razões criam a necessidade de construir sistemas de produção inovadores, capazes de tratar das mudanças, recuperar das perturbações e integrar-se no novo contexto socioeconómico de forma efectiva e eficiente.

1.4 Paradigmas Utilizados

Nesta secção são brevemente apresentados os paradigmas utilizados para a resolução de problemas, os quais serão posteriormente detalhados no Capítulo 4.

No que concerne à Produção, os problemas são extremamente complexos, com inúmeras variáveis, de dimensão considerável e, talvez o mais importante de tudo, extremamente dinâmicos; *i.e.*, as condições não são estáticas. Sendo que esta constante dinâmica influencia a resolução do problema, exigindo que se esteja frequentemente à procura de uma nova solução. Solução essa, que além de adequada deve ser encontrada em tempo útil.

Nas últimas décadas têm-se desenvolvido técnicas e sistemas para lidar com a crescente complexidade dos problemas, quer em termos da sua formulação, quer no que se refere à sua resolução. Tais sistemas, porém, não se apresentam sem algumas limitações, em que as de natureza computacional serão *primus inter pares*. Torna-se portanto necessário encontrar soluções técnicas/tecnológicas para o desenvolvimento de sistemas mais adequados a problemas com estas características. É neste contexto que se recorre à Computação Distribuída e à Inteligência Artificial.

Um sistema computacionalmente distribuído é composto por vários processos (possivelmente em localizações geográficas distintas), em que as tarefas estão divididas de forma funcional e/ou de acordo com o poder de computação de cada máquina. Os *Sistemas Distribuídos* oferecem várias vantagens na manutenção, pela facilidade de intercâmbio entre as diversas entidades do sistema. Os *Sistemas Inteligentes* (fruto da aplicação de técnicas de Inteligência Artificial) materializam-se através de entidades com valências tais como, cooperação, coordenação, racionalidade, e inteligência, oferecendo um grande grau de flexibilidade e de auto-adaptação. É pois de concluir que da fusão destas duas disciplinas (*i.e.*, Inteligência Artificial e Computação Distribuída) surja uma nova, capaz de oferecer, pelo menos teoricamente, melhores soluções para problemas complexos em domínios caracterizados por uma grande dinâmica de alteração e/ou adaptação.

A integração das técnicas da Inteligência Artificial com as da Computação Distribuída, deu corpo a uma nova disciplina, a *Inteligência Artificial Distribuída* (IAD). A IAD foi inicialmente

definida como uma disciplina de resolução de problemas em que os dados, o conhecimento, e a computação estão geograficamente separadas em termos lógicos e/ou físicos [Nilsson, 1981] [Davis, 1980]. A investigação em IAD deu mais tarde origem aos *Sistemas Baseados em Agentes* e aos *Sistemas Multiagente* (SMA). Embora seja um campo de estudo desde o início dos anos noventa, não existe uma definição consensual de agente. Normalmente, é aceite que um agente é uma entidade autónoma, com um certo grau de inteligência, que persegue os seus objectivos, muitas vezes em representação de um utilizador. Tais agentes têm que ser dotados de capacidade de socialização de modo a lidar com outros agentes. Exibem comportamentos pró-activos, guiados por objectivos e reactivos; e podem apresentar dotes para a aprendizagem automática.

Os Sistemas Baseados em Agentes são “*adequados para aplicações modulares, descentralizadas, dinâmicas, subespecificadas* [estrutura não completamente definida] *e complexas*” [Parunak, 1998a], que apresentem um “*grande número de interacções entre componentes*” [Kouiss *et al.*, 1997]. Ora, um Sistema de Produção é caracterizado [Sousa *et al.*, 2000b]:

- pelas suas funções (*e.g.*, Planeamento de Produção; Escalonamento; Gestão de Existências) e pelas suas entidades (*e.g.*, recursos, ordens de fabrico) que podem ser considerados *módulos*;
- como será apresentado na secção 3.4.1, a estrutura hierárquica e centralizada, tradicionalmente utilizada nos sistemas de Produção Integrada por Computador (CIM), não é satisfatória para os requisitos actuais, pelo que os novos sistemas de produção devem ser *descentralizados*;
- pelo *dinamismo* oriundo do funcionamento do sistema (*i.e.*, novas encomendas, novos produtos, mudanças de recursos, avarias);
- pela constante mudança do processo produtivo, onde consumidores e fornecedores podem variar, nunca se sabendo a estrutura exacta da cadeia de fornecimento, o que implica *subespecificação*;
- pelo elevado número de produtos, recursos e ordens de fabrico, bem como os diferentes planos alternativos de produção, que são factores contribuintes para a *complexidade* dos Sistemas Produtivos.

Por estes motivos, espera-se que os princípios nos quais se baseiam a Inteligência Artificial Distribuída e os Sistemas Multiagente sejam úteis na criação da nova geração de sistemas de produção [Sousa e Ramos, 1997]. Esta metodologia irá permitir modelar um sistema como um conjunto de elementos autónomos, inteligentes e cooperativos, de forma a obter arquitecturas flexíveis e reconfiguráveis [Sousa e Ramos, 1998].

Ora um sistema complexo, flexível e configurável é precisamente o alvo de atenção da teoria dos Sistemas Holónicos [Koestler, 1967]. Os *Sistemas Holónicos* baseiam-se na noção de ‘*holon*’ [Koestler, 1967] que significa “o todo” e “a parte”, permitindo esta dualidade que um holon seja ao mesmo tempo visto como um elemento de construção de unidades maiores e ele próprio uma dessas unidades. Koestler (1967) define duas propriedades fundamentais dos holons: *autonomia* (a capacidade de uma entidade criar e controlar a execução dos seus próprios planos e estratégias) e *cooperação* (um processo no qual um conjunto de entidades desenvolve e executa planos mutuamente aceites). A estrutura organizacional de um sistema holónico é denominada holarquia. Pela própria definição de holon, uma holarquia é um holon. O ponto forte dos sistemas holónicos reside no facto de permitir a construção de sistemas complexos que, não obstante a sua complexidade, são eficientes na utilização de recursos, altamente resilientes às perturbações (externas e internas) e adaptáveis às mudanças no ambiente no qual existem [Valckenaers *et al.*, 1997] [Langer, 1999].

Na área de Produção, os *Sistemas Holónicos de Produção* (HMS) [Valckenaers *et al.*, 1994a] [Valckenaers *et al.*, 1997] [van Brussel *et al.*, 1998] materializam o paradigma que melhor representa os conceitos da IAD e SMA [Sousa *et al.*, 2000b]. Em contrapartida aos Agentes Inteligentes existem Holons, que são entidades igualmente autónomas (de *hardware* e/ou de *software*), possivelmente inteligentes que cooperam com outros holons na realização de tarefas. Semelhante ao conceito de *Sistema de Produção Baseado em Agentes* (SPBA), surge o conceito de Sistema Holónico de Produção como um agrupamento de holons que possibilita a existência de um processo de fabrico dinâmico e descentralizado, onde as mudanças são efectuadas de forma dinâmica e contínua [Valckenaers *et al.*, 1994a]. Não obstante a sobreposição dos conceitos, eles complementam-se, nomeadamente, na concretização de Sistemas Holónicos através dos Sistemas Multiagente [Bongaerts, 1998] [Busmann, 1998] [Kirsch *et al.*, 1998] [Sousa *et al.*, 2000c] [Ulieru *et al.*, 2000].

Estes conceitos são inerentemente distribuídos e por isso torna-se necessário regulamentar as interacções entre as diversas entidades existentes no sistema. Ora, se existem alternativas para a prossecução de uma dada tarefa, e se esta for para ser conduzida por entes racionais, os agentes, então há que os agrupar (*i.e.*, socializar [Neves e Machado, 1997a]), dotá-los de capacidade de argumentação e formas de negociação (tímidas aproximações a este problema estão desde já documentadas, sendo de referir o trabalho desenvolvido sobre *Redes de Contrato* [Smith, 1980] [Davis e Smith, 1983]). Por outras palavras, um protocolo que regule o relacionamento entre contratante e contratado, é factor decisivo em todo este processo.

Para a representação de conhecimento nos computadores, recorre-se muitas vezes à utilização de uma *Linguagem Lógica* [Wooldridge, 1992] [Russell e Norvig, 1995]. As linguagens lógicas permitem uma descrição formal e não ambígua de factos, que podem ser verificados e validados

formalmente [Ligeza, 1997] [Pereira, 1997] [Santos *et al.*, 1999]. O paradigma da *Programação em Lógica* (PL) baseia-se na utilização da lógica como uma linguagem de programação. A PL é feita segundo uma abordagem declarativa, onde se descreve um problema em termos de axiomas lógicos. No entanto, cenários de Informação Incompleta ocorrem normalmente nas Bases de Conhecimento e nos processos de negociação [Neves, 1984] [Traylor e Gelfond, 1993] [Neves *et al.*, 1997b]. A PL é baseada nalguns pressupostos (*e.g.*, Mundo Fechado, que significa que todo o conhecimento de determinados factos é conhecido) que impõem limitações ao tipo de processamento necessário para tratar informação incompleta. Os sistemas reais podem no entanto, beneficiar largamente de abordagens que evitem estas limitações. Ao adicionar capacidade para representação e raciocínio sobre informação incompleta a um sistema, a sua base de conhecimento passa a poder descrever o mundo real de forma muito mais correcta. A *Programação em Lógica Estendida* (PLE) é uma extensão da PL, permitindo a inclusão de informação negativa explícita, sendo que, por defeito, todo o conhecimento é aberto, ou seja, é possível deduzir se determinado facto é verdadeiro, falso ou desconhecido.

Resumidamente, ao longo deste trabalho aplicar-se-ão então os paradigmas de Sistemas Multiagente, Sistemas de Produção Baseados em Agentes, Sistemas Holónicos, Sistemas Holónicos de Produção e Programação em Lógica Estendida.

1.5 Objectivos, Delimitação e Pressupostos

De uma maneira geral, com este trabalho procurar-se-ão desenvolver comunidades de agentes holónicos que ajudem na resolução de problemas em ambientes industriais (com especial ênfase no escalonamento dinâmico de ordens de fabrico – caso de teste seleccionado), com recurso à especialização e descentralização de tarefas.

Mais concretamente, os objectivos do presente trabalho serão enumerados a seguir:

1. *Estudo da área de aplicação (produção)* – obtenção de conhecimento na área de produção, nomeadamente no que toca a dificuldades e problemas existentes (Capítulo 2 e Capítulo 3);
2. *Estudo do conceito enunciado na hipótese* – obtenção de conhecimento acerca de Sistemas Holónicos de Produção e técnicas de implementação de tais sistemas (Capítulo 4);
3. *Especificação de uma arquitectura* – que permita antever soluções para os problemas encontrados (secção 5.2);
4. *Especificação de cada entidade constituinte da arquitectura* – em termos de conhecimento (secção 5.2) e programação (secção 6.2).

- conhecimento (secção 5.3) e operação (secção 6.3);
5. *Tratamento de informação incompleta* – representação (secção 5.4.1), identificação de casos (secção 5.4.2) e utilização de informação incompleta na base de conhecimento de cada entidade;
 6. *Especificação de um mecanismo de negociação* – para regulamentar a interacção entre os vários elementos constituintes da arquitectura (secção 6.2);
 7. *Desenvolvimento de um protótipo* – dando corpo à arquitectura conceptualizada anteriormente (secção 6.4);
 8. *Resposta às questões enunciadas* – aplicando as conclusões obtidas na elaboração do trabalho (secção 7.3);
 9. *Elaboração da tese escrita* – como veículo de transmissão do conhecimento científico alcançado durante a elaboração do trabalho.

Derivadas da questão inicialmente colocada em termos de investigação e de hipóteses porventura referenciadas, bem como dos objectivos apresentados, surgem algumas questões de carácter genérico ou particular que devem ser respondidas durante ou após a realização do trabalho:

- Que suporte tecnológico se deve usar para a realização de sistemas de acordo com a Teoria dos Sistemas Holónicos?
- Será possível utilizar a estrutura holónica noutras áreas que não a da Produção?
- Que holons e holarquias há que definir de forma a abordar as várias actividades da produção de forma integrada?
- Existirão cenários de falta de informação e conhecimento nas actividades de produção? Se sim, quais?
- De que forma deverão os vários elementos da arquitectura comunicar e interagir entre si?
- Como saberão os vários elementos da existência de outros elementos?
- Como assegurar a coerência das decisões tomadas por cada elemento?
- Como formular, descrever e decompor o problema?

Os vários capítulos irão apresentando respostas a estas perguntas que serão respondidas de forma explícita na secção 7.3 e abordadas nas conclusões.

Este trabalho (e mais concretamente, o protótipo) é delimitado à actividade de escalonamento de ordens de fabrico, assumindo-se para tal algumas simplificações, nomeadamente:

- *buffers* de capacidade infinita;
- operações de *setup* com tempo de concretização ou duração zero;
- transporte entre máquinas com duração zero e sempre disponível.

Na implementação do sistema são ignorados aspectos de segurança (tais como verificação da identidade) admitindo-se sempre que a informação enviada por um holon a outro é válida e de confiança – *pressuposto de veracidade*.

O termo ‘Fabrico’ é utilizado ao longo do texto para denotar as actividades de execução (fabricação) de produtos, ou seja, as actividades envolvidas no processo de obter um produto com existência física. Por outro lado, o termo ‘Produção’ é utilizado num sentido mais abrangente, englobando o fabrico e todas as restantes actividades económicas, financeiras e comerciais da empresa.

Ao longo do texto refere-se algumas vezes o termo ‘Nova Economia’, que não deve ser interpretado como referindo-se às indústrias de telecomunicações, entretenimento e conteúdos, mas devendo isso sim, ser interpretado como o contexto socioeconómico das últimas décadas do século XX, obviamente muito condicionado pelas alterações provocadas pelas indústrias referidas.

1.6 Metodologia

O método experimental que se utilizou para a realização deste trabalho teve em linha de conta os passos seguintes:

1. *Revisão de literatura existente sobre o domínio de aplicação; i.e., sobre Sistemas Produtivos* – serviu esta fase para identificar as tendências verificadas na sociedade em geral e as suas implicações para a Produção, bem como as visões de futuro propostas por alguns grupos de trabalho;
2. *Revisão de literatura existente sobre os paradigmas a utilizar (com especial realce para aplicações em Produção)* – nesta fase adquiriram-se conhecimentos sobre a utilização das técnicas e paradigmas, e analisaram-se trabalhos anteriores para identificar aspectos em aberto ainda não tratados;

3. *Concepção do sistema informático* – após uma análise profunda dos conhecimentos adquiridos nas fases anteriores, partiu-se para a concepção de um sistema informático capaz de apoiar as hipóteses em aberto na resolução do problema;
4. *Implementação do protótipo e teste* – esta fase pretendeu passar à prática as ideias conceptualizadas na fase anterior. Foi escolhida uma subárea de aplicação (escalonamento dinâmico de ordens de produção) e desenvolvido um protótipo com o qual se realizaram testes com vista a obter resultados que pudessem suportar todo o trabalho desenvolvido até então;
5. *Elaboração da Tese* – finalmente, passou-se à fase de escrita da dissertação, com base nas observações, experiências, e resultados obtidos nas fases anteriores.

1.7 Contribuições e Originalidades

O trabalho desenvolvido e apresentado nesta tese procura providenciar respostas às questões levantadas anteriormente e outras relacionadas, não tendo a pretensão de fornecer soluções definitivas, mas sim contribuir com uma pequena parcela de conhecimento que possibilite algum avanço científico nas áreas em estudo.

Deste trabalho obtiveram-se alguns resultados relacionados com os diversos objectivos previamente enunciados, sendo este documento e o protótipo desenvolvido a face mais visível desse esforço. Na sua prossecução foram abordados vários temas que vão desde os sistemas distribuídos, sistemas multiagente, sistemas holónicos, sistemas holónicos de produção e sistemas de produção baseados em agentes, até à programação em lógica e programação em lógica estendida, tendo esse conhecimento sido utilizado no desenvolvimento de um sistema de carácter experimental (*i.e.*, um protótipo) denominado *Fabricare*, que se aplica à resolução de problemas de escalonamento.

As principais contribuições e originalidades do presente trabalho são a seguir enunciadas:

- *Definição e especificação de uma arquitectura para sistemas holónicos de produção* – (objectivos 3 e 4). A secção 5.2.1 apresenta uma arquitectura baseada no paradigma dos Sistemas Holónicos para empresas de produção que abarca as áreas funcionais do fabrico (Projecto, Planeamento de Processos, Planeamento de Produção e Escalonamento), compras e vendas [Sousa e Ramos, 1997] [Sousa e Ramos, 1998] [Sousa *et al.*, 1999b]. Na secção 5.3 é especificada a base de conhecimento e na secção 6.3 o funcionamento de cada elemento da arquitectura. Sendo o escalonamento o caso escolhido para teste, é dada especial ênfase aos holons que possibilitam essa actividade, ou seja, os Holons de Tarefa e os Holons de Recurso. As originalidades desta

arquitectura centram-se nas entidades modeladas, na abordagem distribuída e descentralizada e no potencial da arquitectura para integração empresarial e integração inter-empresarial.

- *Notação para representação de informação incompleta* – (objectivo 5). A secção 5.4.1 apresenta uma notação para a representação de informação incompleta na forma de axiomas da base de conhecimento de cada holon [Sousa *et al.*, 2000b]. Com base em [Traylor e Gelfond, 1993] [Analide e Neves, 1996] [Neves *et al.*, 1997b] e utilizando a programação em lógica estendida (com representação explícita de informação negativa), essa notação permite a representação de nulos do tipo desconhecido, nulos desconhecidos de um conjunto finito de valores e nulos não permitidos. O facto de se abordar a problemática da informação incompleta nos sistemas de produção é original. A notação utilizada apresenta algumas alterações em relação à de Analide e Neves (1996), nomeadamente no que respeita à especificação de nulos não permitidos e à elaboração do meta-interpretador.
- *Identificação de situações de informação incompleta em sistemas de produção* – (objectivo 5). Na secção 5.4.2 são apresentados 46 casos identificados de situações onde pode existir falta de informação nas empresas de produção. Esses casos levam em linha de conta a base de conhecimento de cada holon identificado anteriormente e que consta da arquitectura proposta [Sousa *et al.*, 2000b]. Esta contribuição é original pois não se conhecem outros trabalhos que tenham efectuado um levantamento de situações de informação incompleta em sistemas de produção.
- *Extensão ao protocolo de rede de contrato com coordenação entre fornecedores de serviço* – (objectivo 6). A secção 5.4 apresenta o Protocolo de Rede de Contrato com Propagação de Restrições (PRCPR) [Sousa e Ramos, 1998] [Sousa *et al.*, 1999b] [Sousa e Ramos, 1999a] utilizado para regulamentar a interacção entre os Holons de Tarefa e os Holons de Recurso. Este protocolo é baseado no protocolo de rede de contrato, com duas novas funcionalidades, de tal modo que os vários holons fornecedores de serviços possam trocar informação entre si para coordenar as dependências temporais existentes entre as várias operações requisitadas pertencentes a uma mesma tarefa. Este protocolo apresenta um aspecto original, o de tratar explicitamente as fases de coordenação entre holons fornecedores de recursos. Contribui-se ainda com o estudo de complexidade do problema e do protocolo.
- *Concepção de um algoritmo distribuído para escalonamento* – como resultado colateral do protocolo RCPR obteve-se um algoritmo distribuído de escalonamento, baseado num esquema centralizado (*vide* secção 6.3.7) [Sousa *et al.*, 1999b]. Embora

este algoritmo se baseie no método descrito em [Almeida, 1995] e [Ramos *et al.*, 1995], a distribuição do algoritmo, as mudanças efectuadas ao nível da combinação de intervalos e a não utilização de comportamentos são originais.

- *Desenvolvimento de um protótipo* – (objectivo 7). Na secção 6.4 é apresentado o sistema *Fabricare* desenvolvido de acordo com a arquitectura dada (em termos de base de conhecimento e funcionamento) [Sousa *et al.*, 1999b] [Sousa *et al.*, 2000a]. Este protótipo é original pois corresponde à implementação da arquitectura que é proposta neste trabalho.

Nesta secção apresentou-se resumidamente o trabalho efectuado, tendo sido realçado em que é que contribuiu para o avanço do conhecimento bem como as suas originalidades. Nos capítulos subsequentes esse trabalho será melhor enquadrado e detalhadamente descrito.

1.8 Estrutura e Resumo da Dissertação

Esta secção apresenta a estrutura da dissertação, sendo apresentado um resumo do conteúdo de cada capítulo para fornecer uma noção geral do trabalho antes de entrar em detalhe na sua descrição.

A dissertação trata o tema das *abordagens inovadoras à actividade de produção*, e como tal numa **primeira parte** (capítulos 2, 3 e 4) torna-se necessário apresentar a produção e o fabrico propriamente dito, bem como o contexto socioeconómico da ‘Nova Economia’ e identificar problemas existentes nos sistemas de produção actuais. Na tentativa de solucionar esses problemas propõe-se então os *Sistemas Holónicos de Produção* e efectua-se um estudo dos paradigmas para a resolução de problemas que permitam implementar tais sistemas. Assim sendo, é abordado o tema genérico dos Sistemas Distribuídos Inteligentes, mais especificamente os *Sistemas Multiagente* e a problemática da interacção entre entidades computacionais autónomas. A *Programação em Lógica Estendida* é usada como ferramenta de representação de conhecimento e de implementação do protótipo. Na **segunda parte** (capítulos 5 e 6) é finalmente descrito o trabalho elaborado ao longo do Doutoramento, que pretende apresentar uma solução para os problemas detectados na primeira parte, utilizando os paradigmas descritos. Nestes capítulos vai então ser proposta uma arquitectura para sistemas de produção, arquitectura essa que será em seguida especificada em termos de conhecimento e de modos de operação.

O conteúdo de cada um dos sete capítulos em que a tese se encontra dividida é apresentado em seguida de forma mais detalhada.

O **Capítulo 1** enquadrou o trabalho realizado, apresentou a Questão central do trabalho e as hipóteses que apontam para uma solução do problema. Após a justificação do trabalho de

investigação foram apresentadas as diversas metodologias de trabalho, bem como os objectivos a atingir. Em seguida foram apresentadas as contribuições e originalidades do trabalho. Finalmente, é referida a organização estrutural e o conteúdo da dissertação.

O **Capítulo 2** introduz o campo de aplicação deste trabalho, isto é, os Sistemas de Produção, apresentando uma breve introdução do que é um sistema de produção, visto pela e na sociedade, quais os seus constituintes e quais as suas funções. Em seguida são apresentados conceitos fundamentais, tais como produto e processo. Torna-se necessário classificar diferentes tipos de Sistemas de Produção, pelo que se apresentam dois tipos de classificação possível, onde irão ser definidos conceitos como *job shop*, *flow shop*, produção por encomenda ou para existência. O fabrico é uma actividade complexa, na qual várias funções se inter-relacionam de forma a produzir os bens que a empresa fabrica. Assim sendo, são apresentadas as funções da actividade de fabrico: Gestão da Produção; Projecto; Planeamento de Processos; Planeamento da Produção; Escalonamento; e Execução, Inspeção e Controlo da Qualidade. Por fim é apresentada a Produção Integrada por Computador (CIM), como sendo uma metodologia para a competitividade e aumento da produtividade da empresa. Após uma breve introdução ao conceito CIM, são apresentados vários conceitos relacionados com o suporte tecnológico, tais como Sistemas Flexíveis de Fabrico (FMS), Projecto Assistido por Computador (CAD) ou Fabrico Assistido por Computador (CAM). Para terminar, são apresentados alguns aspectos das dificuldades de implementação do conceito CIM na prática.

No **Capítulo 3** faz-se uma apresentação das várias tendências observadas na sociedade que implicam novas maneiras de abordar os Sistemas de Produção. Este capítulo estende a secção 1.1 “Enquadramento”, apresentando de forma mais detalhada o contexto socioeconómico das últimas décadas do século XX. Esta época é chamada por muitos de época da ‘Nova Economia’ e por isso é importante apresentar definições, características e comparações com práticas correntes. Na secção 3.2.1 será apresentado um pequeno enquadramento histórico da produção e economia, bem como do contexto social e político desde o século XVIII até ao final do século XX. Esta secção serve de base à secção 3.2.2 onde se analisam as tendências de mercados e produção observados nos últimos 40 anos do século XX. Em seguida, após análise das mudanças e das novas características encontradas na sociedade e economia, será apresentado o contexto socioeconómico da ‘Nova Economia’, segundo as vertentes de Força Laboral; Responsabilidade Social e Ambiental; O Conhecimento como activo da Empresa; Dinamismo, Reactividade e Organização; e Revolução das Tecnologias de Informação. Desta apresentação importa realçar a importância cada vez maior de características como a *flexibilidade* e a *agilidade* e da importância crescente das tecnologias de informação como impulsionadoras de novas práticas de negócio. Apresentada a ‘Nova Economia’ interessa discutir aspectos relacionados com a produção neste novo contexto; para isso serão equacionados alguns dos problemas dos sistemas de produção actuais. Em seguida

irão ser apresentadas algumas visões de futuro para a produção, tendo como base as alterações ocorridas na sociedade e economia. Finalmente, elabora-se uma lista de características e requisitos a cumprir por uma nova geração de sistemas de produção para tentar combater as falhas actuais e responder aos desafios futuros.

O **Capítulo 4** apresenta uma abordagem inovadora aos sistemas de produção, os *Sistemas Holónicos de Produção* (HMS), que tenta colmatar as falhas identificadas nos sistemas de produção actuais, estando de acordo com a lista de características desejáveis nos sistemas de produção elaborada na secção 3.4.3 para responder aos requisitos actuais e futuros da sociedade e economia. Assim, começa-se por apresentar as origens filosóficas (que tentam explicar a evolução e organização de sistemas biológicos e sociais) dos sistemas holónicos e as suas características, com realce para a *distribuição*, *autonomia* e *cooperação*. Em seguida, são apresentadas as origens dos sistemas holónicos de produção no seio da comunidade internacional de investigadores, bem como as suas características. Os HMS não são a única abordagem inovadora aos sistemas de produção e por isso serão também sumariamente descritas outras abordagens, nomeadamente a *Fábrica Fractal* e a *Produção Biónica*. Na secção seguinte apresentam-se as metodologias a seguir para a concretização do sistema dando corpo ao conceito holónico: os *Sistemas Baseados em Agentes* e a *Programação em Lógica Estendida*. Assim, começa-se por apresentar o conceito de *agente* e as suas características, sendo em seguida apresentado o conceito de *sistema multiagente*. Na secção seguinte é apresentada a lógica como linguagem formal para especificação de agentes e a programação em lógica estendida (*i.e.*, com representação de negação explícita) para a codificação desses agentes. Finalmente, é apresentado um estado da arte na área dos Sistemas Holónicos de Produção e Sistemas de Produção Baseados em Agentes. Dos trabalhos escolhidos como referência será elaborada uma sinopse para evidenciar os pontos em que não há acordo com os objectivos enunciados nesta tese.

O **Capítulo 5** apresenta a arquitectura proposta tendo por base o conceito holónico descrito no Capítulo 4. Assim sendo, é efectuada uma análise funcional da produção, identificando os principais casos de utilização. Após a análise funcional, identificam-se quais as principais entidades do sistema e os seus relacionamentos básicos: *Fornecedor*, *Compra*, *Produto*, *Cliente*, *Venda*, *Tarefa* e *Recurso*. Na secção seguinte é então apresentada a arquitectura holónica proposta, constituída por holarquias para as funções de *Projecto*, *Planeamento de Processos*, *Planeamento da Produção*, *Escalonamento*, *Compras* e *Vendas*. Estas holarquias são constituídas por holons que representam os Produtos, Recursos, Tarefas, Fornecedores, Clientes, Compras e Vendas. Os vários holons constituintes de cada holarquia são especificados, na secção seguinte, em termos de base de conhecimento, ciclo de vida, objectivos e identificação. Na área da produção, existem várias situações onde toda a informação necessária pelo sistema não está disponível. Por esse motivo, após a apresentação de uma notação para a representação de

informação incompleta, a base de conhecimento de cada holon da arquitectura proposta será aumentada com as produções que lhe permitam representar essas situações.

No **Capítulo 6** será apresentado o protocolo de negociação que rege o processo de contratação de serviços entre Tarefas e Recursos, já que, devido à natureza distribuída da arquitectura proposta, torna-se necessário regulamentar a interacção entre os vários holons. Esse protocolo está preparado para o tratamento de excepções e principalmente para evitar o *Problema de Indecisão*. Em seguida será analisada a complexidade do protocolo no que se refere ao número de soluções encontradas, ao número de mensagens trocadas e ao tamanho total das mensagens trocadas. Após a descrição do protocolo será descrito o funcionamento de cada holon. Os principais algoritmos para a operação dos holons directamente relacionados com o escalonamento (*Holon de Tarefa*, *Holon de Recurso*) e outros holons auxiliares (*Holon de Serviço de Directórios*, *Holon de Produto* e *Holon de Escalonamento*) serão então apresentados e explicados. Em seguida, apresenta-se o protótipo de sistema informático implementado seguindo as descrições efectuadas no Capítulo 5 e no Capítulo 6. Este protótipo denominado *Fabricare* é composto por um conjunto de aplicações que incluem os Holons de Tarefa e de Recurso, bem como os holons auxiliares (Serviço de Directório, Planeamento de Processos, Escalonamento) e algumas ferramentas para a exploração do sistema, permitindo colocar em execução os Holons de Recurso e a criação de tarefas. Nesta secção serão incluídos alguns comentários às diferenças entre o método original de escalonamento e o método utilizado neste trabalho, bem como alguns comentários ao desenvolvimento do protótipo. Finalmente, será apresentada uma comparação entre o trabalho desenvolvido e os trabalhos de referência abordados anteriormente no “Estado da Arte”. Essa comparação é efectuada segundo quatro perspectivas de interesse, de acordo como os objectivos enunciados, nomeadamente a estrutura, a socialização das entidades, o tratamento de informação incompleta e o procedimento de escalonamento.

Finalmente, no **Capítulo 7** são apresentadas as conclusões. Assim, começa-se por fazer uma síntese do que foi dito nos capítulos de enquadramento referente à produção e aos seus problemas, bem como à sociedade actual e aos requisitos que coloca nos sistemas de produção. Em seguida serão detalhadas as contribuições e respectivas conclusões do trabalho que permitirão elaborar a *Tese* da dissertação e responder às questões colocadas na introdução. Finalmente, serão apresentadas as limitações do trabalho desenvolvido e as perspectivas de trabalho futuro, apontando as direcções que se pensa vir a trilhar.

No final do documento existem diversos **Apêndices** com informação adicional: o *Apêndice A* apresenta as experiências efectuadas e os resultados obtidos com o sistema *Fabricare*. No *Apêndice B* é apresentado o modelo matemático simplificado e respectiva folha de cálculo utilizada para a análise de complexidade do protocolo RCPR. O *Apêndice C* apresenta uma lista

de recursos existentes na Internet, tais como laboratórios, projectos de investigação e pessoas envolvidas nestas áreas ■

CAPÍTULO 2

SISTEMAS DE PRODUÇÃO



Todos somos produtores. Produzindo o bem, produzindo sarilhos, ou produzindo desculpas.

H. V. Adolt

Os automóveis que conduzimos, assim como as canetas que usamos, a roupa que vestimos, os computadores onde trabalhamos, os livros que lemos ou os sapatos que calçamos, são exemplos de artefactos, objectos que não se encontram na Natureza, mas que foram produzidos de alguma forma.

Neste capítulo começa-se por apresentar uma definição de ‘Sistema de Produção’ e de outros conceitos tais como *produto*, *plano de produção* e *produtividade*. Em seguida, são apresentadas duas formas de classificar os Sistemas de Produção e introduzidos novos conceitos como *produção tipo job shop*, *produção por projecto* e *produção para existência*. Serão também apresentadas as várias actividades directa ou indirectamente relacionadas com o fabrico de bens. Finalmente, é dedicada atenção à utilização de computadores na produção, introduzindo-se o conceito de Produção Integrada por Computador, assim como as várias tecnologias existentes (*e.g.*, CAD, FMS). São também discutidos alguns aspectos da dificuldade de implementação deste conceito.

2.1 Introdução

Um *sistema de produção* é constituído por um conjunto de *recursos* (e.g., máquinas) que efectuem uma ou mais *operações de transformação* de *matéria prima* ou associação de *componentes*, dando origem a *produtos* acabados [Riggs, 1987] [Kalpakjian, 1995]. Esta meta-operação desdobra-se em operações elementares tendo como objectivo a obtenção de um produto [Figueiredo, 1996].

O sistema de produção relaciona-se com o meio envolvente (*i.e.*, sociedade, economia e meio ambiente) com o qual efectua operações de troca nas quais estão presentes três tipos de fluxos [Courtois *et al.*, 1991]:

- *Fluxo de materiais* – objectos com existência física (e.g., máquinas, ferramentas, matéria prima, componentes, produtos, embalagens, combustíveis);
- *Fluxo de informação* – troca de informação entre o meio envolvente e o sistema (e.g., encomendas, facturas, descrições técnicas dos produtos, publicidade);
- *Fluxo financeiro* – entrada ou saída monetária (e.g., pagamento de bens ou serviços, recebimentos, salários, juros de empréstimos, impostos, taxas).

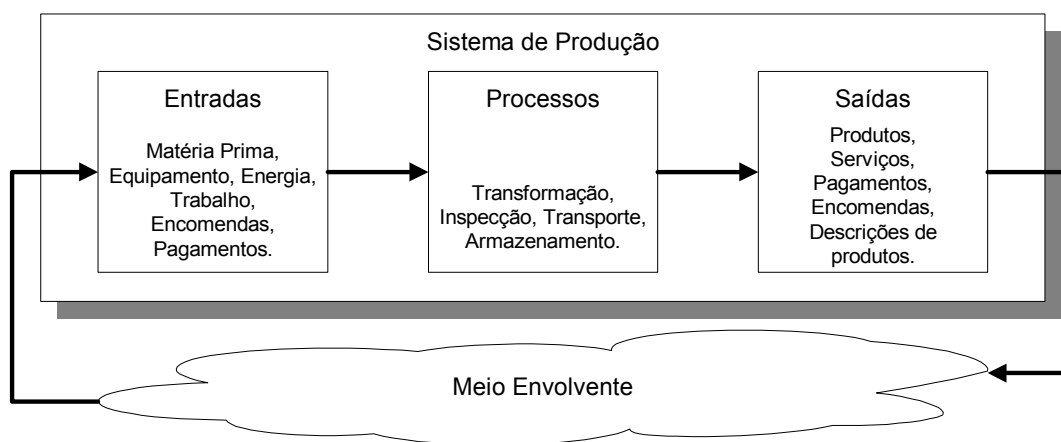


Figura 2.1 – Sistema de produção

Na Figura 2.1 apresenta-se um sistema de produção em interacção com o meio envolvente. O sistema recebe, por exemplo, matéria prima e equipamento (fluxo de material), mas também pedidos de encomenda (fluxo de informação) e pagamentos de entregas de produtos feitos a terceiros (fluxo financeiro). Adicionalmente, o sistema fornece ao meio envolvente os produtos para o mercado (fluxo de material), bem como o pagamento de fornecimentos (fluxo financeiro), assim como, descrições comerciais/técnicas dos produtos (fluxo de informação).

O fabrico de um produto é uma actividade complexa, atravessando horizontalmente várias áreas disciplinares, uma variedade de pessoas, maquinaria e automatismos [Kalpakjian, 1995] [Rocha, 1999], que se desenvolve desde a fase da encomenda, passando pelo projecto, produção e *marketing* [Sousa *et al.*, 2000c]. Devido à enorme quantidade de variáveis em jogo, às inúmeras interacções entre essas variáveis e à dificuldade de controlo sobre essas mesmas variáveis, não há dúvida que estamos na presença de um problema NP-completo¹ [Kalpakjian, 1995].

2.2 Conceitos de Produção

Como já foi referido anteriormente, um sistema de produção é composto por um conjunto de *recursos* que efectuem uma ou mais *operações de transformação de matéria prima* ou *associação de componentes*, dando origem a *produtos acabados*.

Um *produto* é o resultado de um processo de transformação sobre materiais ou outros produtos efectuado pela instalação fabril. O *modelo do produto* descreve o produto de forma completa e inequívoca, contendo informação de desenho, de composição do produto (quantidades de matéria prima/componentes) e de processo [PDMIC, URL]. Os produtos são normalmente agrupados em *famílias de produtos*, atendendo às suas similaridades de forma a facilitar a gestão do catálogo de produtos.

Um produto é normalmente composto por vários *componentes*; *i.e.*, peças já montadas por terceiros ou na própria fábrica (*e.g.*, parafusos, *chips* de memória). Além de componentes, o produto pode também necessitar de *matéria prima*, ou seja, bens que virão a sofrer um processo de transformação ou tratamento (*e.g.*, tecido utilizado no fabrico de peças de vestuário).

O *processo* de fabrico de um produto segue um *plano* composto por várias *operações* elementares (*e.g.*, furar, polir), que devem ser executadas segundo uma dada ordem [Kusiak, 1990]. Este processo gera mais valias ao transformar e/ou juntar componentes para formar novos produtos [Kalpakjian, 1995].

O processo de fabrico dá origem a produtos ditos *contínuos* ou *discretos*. *Produtos discretos* são itens individuais, tais como canetas, pregos ou latas de bebidas. Por outro lado, como *produtos contínuos* têm-se as mangueiras, fios eléctricos, tubos metálicos ou plásticos, que poderão ser cortados em peças, dando origem a produtos discretos [Kalpakjian, 1995].

¹ Um problema é considerado NP (polinomial não-determinista) se necessita de um tempo polinomialmente proporcional ao comprimento das entradas para encontrar uma solução [Garey e Johnson, 1979] [Parunak, 1991] [Skiena, 1997]. Um problema diz-se NP-completo se for NP e pelo menos tão difícil quanto todos os outros problemas NP. Um problema NP-duro é NP-completo ou mais difícil que um problema NP.

Um produto pode ficar inacabado, isto é, o processo de fabrico é interrompido nalgum ponto pré-definido, o que dá origem a *produtos semi-acabados*. Normalmente este tipo de produto é utilizado como componente para outros produtos, ou armazenado para ser acabado mais tarde. Durante a produção, um produto em fabrico que se encontra em circulação entre estações de trabalho, ou a sofrer uma operação numa estação de trabalho, é classificado como estando “*em curso*”.

Um produto não é fabricado de uma forma instantânea demorando um certo tempo a ser executado, denominado por *tempo de fabrico*. Infelizmente, o tempo de fabrico é algo complexo de determinar pois está associado a processos físicos sujeitos a acontecimentos imprevistos (*e.g.*, quebra de uma ferramenta). É no entanto possível fazer uma estimativa do tempo de fabrico em condições ideais – para o caso de planos de produção em que o conjunto de operações que levam desde a matéria prima até ao produto final são executadas de forma sequencial – traduzida na equação (2.1):

$$\text{tempo-de-fabrico} = \sum_{i=1}^n [\iota(o_i, \rho(o_i)) + Q \times \delta(o_i, \rho(o_i))] \quad (2.1)$$

sendo n o número de operações do processo de fabrico, o_i a operação de ordem i , $\rho(o)$ uma função que retorna o recurso associado à execução da operação o , $\iota(o, r)$ uma função que retorna o tempo de inicialização do recurso r para a execução da operação o , (*e.g.*, troca de ferramenta), Q é a quantidade de itens a produzir, e $\delta(o, r)$ uma função que retorna a duração da execução da operação o no recurso r .

Um *recurso* é uma entidade física da instalação fabril imprescindível à execução das *operações* de transformação ou transporte de *matéria prima*, *componentes* ou *produtos*. Um recurso é normalmente uma máquina (*e.g.*, um torno) ou conjunto de máquinas (*e.g.*, célula de maquinaria constituída por um torno, uma fresadora e um manipulador robótico), mas também pode ser uma ferramenta ou um(a) operário(a).

A *flexibilidade* de um sistema de produção refere-se à facilidade/capacidade de produzir diferentes produtos, com tempos de reconfiguração da(s) linha(s) de produção reduzidos e a custos relativamente baixos, o que passa pela utilização de recursos de ordem genérica, programáveis, capazes de executar uma grande variedade de operações.

A *produtividade* de um sistema de produção prende-se com a utilização óptima de todos os recursos da empresa: materiais, equipamento, energia, capital financeiro, capital humano e tecnologia [Kalpakjian, 1995]. A *produtividade* é definida como um quociente entre as saídas (produtos, serviços, ...) e as entradas (mão de obra, capital, matéria prima, energia, ...) [Besant e Lui, 1986].

$$produtividade = \frac{saídas}{entradas} \quad (2.2)$$

O *custo de fabrico de um produto* é um factor importantíssimo a ter em conta, principalmente num mercado competitivo. Existem vários métodos de cálculo de custos (que ultrapassam o âmbito deste trabalho), no entanto, pode-se afirmar que o custo de fabrico de um produto depende de quatro factores: (i) custos de materiais; (ii) custos de processamento; (iii) custos fixos; e (iv) custos de mão de obra.

$$custo-de-fabrico-de-um-produto = M + P + F + Td + Ti \quad (2.3)$$

Na equação anterior, M representa o custo de material (*i.e.*, matéria prima e componentes utilizados no fabrico do produto), P denota o custo de processamento (*e.g.*, custo de ferramentas, maquinaria, energia e transporte), F refere-se aos custos fixos (*i.e.*, custos que existem independentemente do fabrico ou não dos produtos, tais como os juros de empréstimos), Td representa os custos directos de mão de obra (*i.e.*, custos com operários no manuseamento e operação de máquinas), e Ti representa os custos indirectos de mão de obra (*i.e.*, custos com empregados em funções não directamente relacionadas com o fabrico, tais como os custos administrativos, com vendedores e/ou publicidade).

Um aspecto importante do processo produtivo prende-se com o *prazo de entrega* de uma encomenda a um cliente, que corresponde ao tempo passado entre o instante em que a encomenda é colocada e o instante em que os produtos são entregues, que é dado na forma:

$$prazo-de-entrega = PE + I + F + D \quad (2.4)$$

onde PE denota o tempo de processamento da encomenda, I o tempo de inicialização da instalação fabril para executar a encomenda, F o tempo de fabrico definido pela equação (2.1), e D o tempo necessário à distribuição do produto por revendedores e entrega no cliente.

2.3 Tipos de Produção

Embora a classificação dos tipos de produção seja função de uma miríade de parâmetros [Courtois *et al.*, 1991], serão aqui considerados apenas dois: (i) o binómio quantidade/variedade de produtos fabricados; e (ii) o relacionamento com as existências.

2.3.1 Função de Quantidade/Variedade

Este tipo de classificação analisa um sistema de produção de acordo com o tipo de produtos que são fabricados em associação com a quantidade produzida.

É possível dividir os sistemas de produção de acordo com a seguinte taxonomia [Groover e Zimmers, 1984] [Dilworth, 1992]:

- *Flow shop*
- Produção em lote
- *Job shop*
- Produção por projecto.

No caso dos sistemas do tipo *flow shop* fabricam-se poucos tipos de produtos, mas em grandes quantidades (*e.g.*, indústrias petroquímica, cimenteira, automóvel). As máquinas utilizadas são específicas para o processo de fabrico em equação e estão dispostas na instalação fabril de acordo com a sequência de operações a realizar [Figueiredo, 1996].

O tipo de produção anterior dá pelo nome de *produção em massa*, devido à grande quantidade de produtos fabricados [PDMIC, URL]. Um sistema de produção deste tipo possui pouca ou nenhuma flexibilidade, sendo caracterizado por um grande nível de automação dos processos de produção, bem como dos sistemas de manutenção, o que permite obter alta produtividade e baixos custos de produção [Courtois *et al.*, 1991].

A *produção por lote* é caracterizada pelo fabrico de conjuntos de produtos, onde os itens do conjunto apresentam similaridades entre si [PDMIC, URL]. Esta é uma etapa intermédia entre a produção contínua e a produção descontínua, uma vez que a quantidade a produzir não justifica uma linha de fabrico dedicada e, por outro lado, a variedade não é tanta que justifique a flexibilização total da instalação fabril [MESA, URL]. A utilização de lotes de produtos permite rentabilizar o tempo de preparação dos recursos, visto que apenas é feita uma preparação para o lote completo, e não para cada peça.

Um sistema de produção tipo *job shop* lida com quantidades relativamente pequenas e com uma grande variedade de produtos, que são produzidos no parque de máquinas existentes na fábrica [Courtois *et al.*, 1991]. No limite, os sistemas tipo *job shop* tratam lotes de cardinalidade um, uma vez que a penalização por preparação das máquinas e carregamento de programas é mínima, evitando-se assim a necessidade de agrupamento de ordens de fabrico em lotes, o que reduz o efeito de *setups* deveras demorados.

As máquinas utilizadas numa instalação fabril do tipo *job shop* necessitam de muita flexibilidade (*i.e.*, polivalência de operações), devido à grande variedade de produtos com que laboram. A polivalência das máquinas permite a utilização de máquinas diferentes para a execução de uma mesma operação, não havendo um fluxo pré-definido dos produtos entre estações de trabalho [Sousa *et al.*, 2000c]. Essas máquinas estão normalmente agrupadas (Figura 2.2) de acordo com a funcionalidade (*i.e.*, agrupamentos de máquinas iguais a executar operações

como tornear, furar), ou em ilhas multifuncionais (*i.e.*, agrupamentos de máquinas com funcionalidades diferentes).

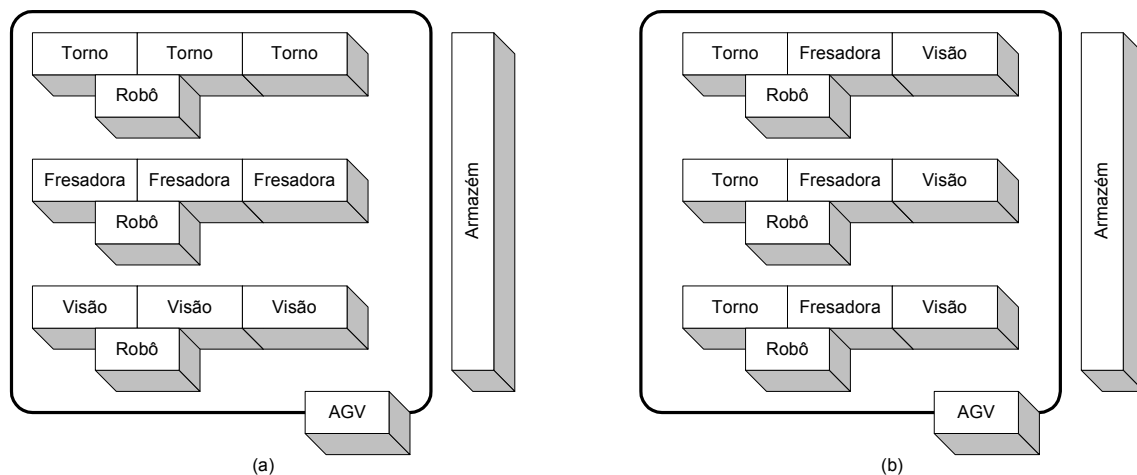


Figura 2.2 – Job shop: (a) agrupamento por função; (b) ilhas multi-funcionais

Na primeira hipótese as peças irão de ilha para ilha consoante necessitem de uma operação diferente, sendo servidas pela máquina que estiver livre nessa ilha. Na segunda hipótese, tenta-se minimizar os tempos de transporte entre operações, visto (que pode) não ser necessário uma mudança de ilha, normalmente mais demorada que as mudanças entre estações de trabalho na mesma ilha, podendo inclusive cada ilha ter o seu próprio armazém. A tecnologia utilizada em *job shop* pode também ser utilizada em empresas que sigam uma filosofia do tipo *personalização em massa*. A *personalização em massa* é um tipo de produção onde todos os produtos são desenvolvidos e fabricados de forma personalizada, de acordo com as especificações de cada cliente [Pine *et al.*, 1999]. Isto implica que componentes específicos, montagens específicas e programas de execução para as máquinas, terão que ser desenvolvidos de acordo com os requisitos fornecidos pelo cliente. Normalmente, estas empresas não possuem em carteira um leque interminável de produtos, mas oferecem um conjunto de produtos e componentes que podem ser combinados de inúmeras formas pelo cliente [Schonfeld, 1998]. Em situações deste tipo é necessário um projecto cuidadoso de cada componente do produto e/ou processo produtivo por forma a minimizar os tempos de fabrico e maximizar as interações entre componentes.

Na *produção por projecto* cada obra é única (*i.e.*, quantidade = 1), tratando produtos altamente complexos em que o tempo de fabrico é normalmente longo (*e.g.*, ponte rodoviária, barragem) [Ferreira, 1998]. A produção por projecto implica normalmente a agregação de recursos à volta do produto e/ou no local onde o produto irá ser colocado.

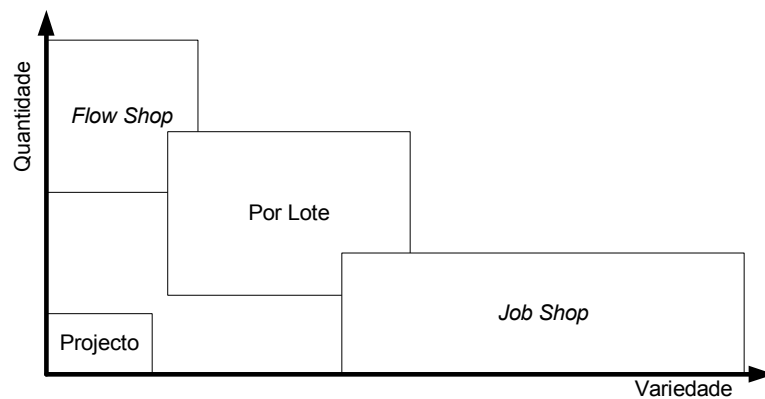


Figura 2.3 – Relação entre os vários tipos de produção

A Figura 2.3 expressa a relação entre a quantidade e variedade de produtos, para cada um dos tipos de fabrico apresentado anteriormente. O eixo ‘variedade’ também se relaciona com a flexibilidade da instalação fabril, pois há uma relação de proporcionalidade entre flexibilidade da instalação fabril e a variedade de produtos. Isto é, quanto maior a flexibilidade, maior a variedade de produtos que se podem fazer. Por outro lado, se há uma grande variedade de produtos a fabricar, maior é a flexibilidade necessária.

2.3.2 Relação com as Existências de Produtos

Esta classificação analisa um sistema de produção relacionando a produção com as existências de produtos finais, sendo possível funcionar segundo uma filosofia de *produzir para existência*, ou *produzir por encomenda*. No primeiro caso, o fabrico dos produtos baseia-se em previsões de procura, sendo os produtos armazenados e posteriormente encomendados pelos clientes. No segundo caso os produtos são fabricados de acordo com as encomendas dos clientes, que definem as características a que o produto deve obedecer e a data de entrega [PDMIC, URL].

O método de *produção para existência* permite reduzir os custos de produção ao aumentar a quantidade fabricada (*e.g.*, uma tiragem de vários milhares de exemplares de um livro). Este método é também particularmente útil quando o tempo de fabrico é superior ao prazo de entrega suportável pelo cliente (*e.g.*, no vestuário, em electrodomésticos) [Courtois *et al.*, 1991].

O método de *produção por encomenda* permite reduzir os custos financeiros do fabrico ao diminuir (ou tornar nulas) as quantidades em existência. No entanto, este método exige um compromisso por parte do cliente no que toca ao cumprimento da encomenda. Um outro aspecto a ter em conta prende-se com o prazo de entrega do produto ser ou não ser aceite por parte do cliente.

Normalmente, um sistema de produção funcionará num modo híbrido, dado que alguma quantidade de produtos em existência permite suprir falhas e avarias nas máquinas, bem como atrasos no fabrico e distribuição. Isto permite criar uma margem de segurança, evitando falhas na entrega. Também é comum manter-se alguma existência de produtos semi-acabados, que serão finalizados de acordo com as encomendas. Este método permite diminuir o prazo de entrega do produto final ao cliente, visto que se reduz o tempo de fabrico (apenas há que considerar as operações de acabamento do produto semi-acabado) entre o processamento da encomenda e a entrega do produto final.

2.4 Funções da Actividade de Fabrico

Uma empresa de produção compreende várias áreas funcionais (Figura 2.4), desde as que estão presentes em todas as empresas (tais como os sectores administrativos, de *marketing*, de compras, ou de vendas), a outras mais específicas, que se relacionam com o fabrico propriamente dito. No fabrico é possível identificar cinco fases: (i) projecto; (ii) planeamento de processos; (iii) planeamento da produção; (iv) escalonamento; e (v) execução, inspecção e controlo de qualidade. Além destas actividades, existe uma outra, igualmente importante, denominada de Gestão da Produção, responsável pela gestão do processo produtivo como um todo.

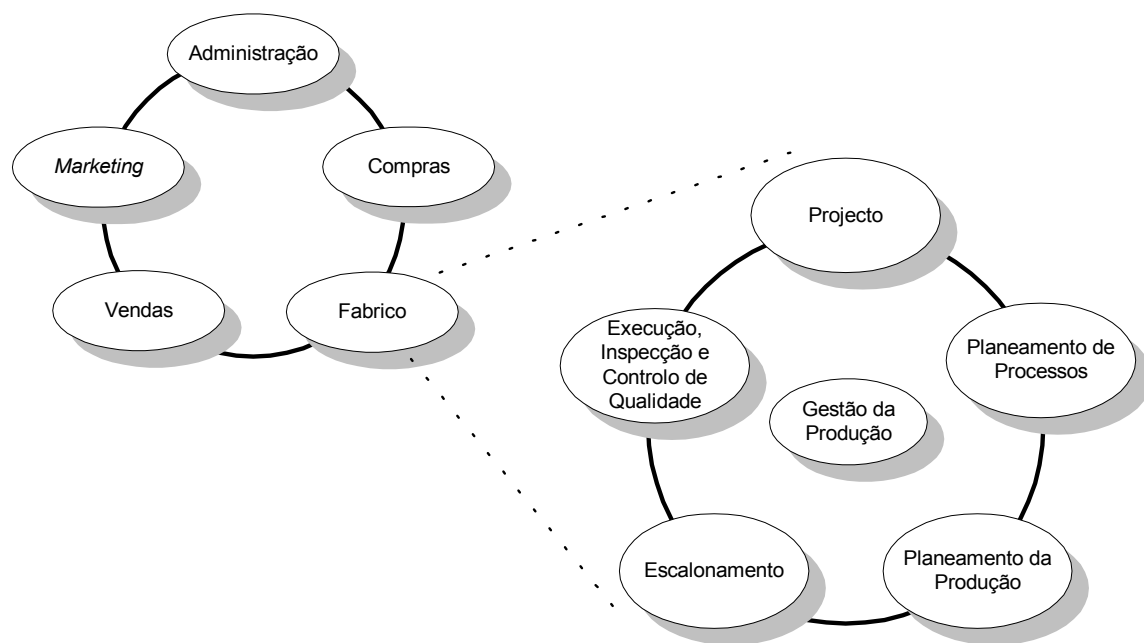


Figura 2.4 – Funcionalidades de uma empresa de produção industrial

2.4.1 Gestão da Produção

O papel fundamental de uma *Gestão da Produção* traduz-se na gestão dos fluxos de materiais e dos fluxos de informação que levem ao fabrico de acordo com os objectivos definidos pela direcção da empresa, de uma ou mais classes de produtos [Courtois *et al.*, 1991].

Segundo Laverty e Demeestère (1993) “*gerir a produção significa, descrever as diferentes fases do processo de fabrico de um produto acabado e as soluções de substituição em caso de falhas não sistemáticas, e também determinar os modos de operação a pôr em prática para fabricar um produto acabado de acordo com as suas especificações e ao menor custo*”.

A gestão da produção é uma actividade que atravessa horizontalmente a empresa, na medida em que se relaciona com todas as outras actividades da mesma, o que pode levar a conflitos [Courtois *et al.*, 1991]. De um modo geral pode dizer-se que o objectivo da gestão da produção é rentabilizar o sistema produtivo. Essa rentabilização passa pela diminuição dos prazos de entrega, pelo aumento da fiabilidade, pela diminuição de custos, pela motivação do pessoal, e pelo desenvolvimento de uma cultura empresarial [Courtois *et al.*, 1991].

A eficácia da gestão da produção apoia-se sobre um *domínio correcto dos fluxos de informação* que acompanham a elaboração do produto, bem como num *planeamento adequado da produção*. Assim, para gerir a produção é necessário recolher, juntar e compilar toda a informação referente ao sistema de produção, tendo em vista facilitar a tomada de decisão de forma correcta e no momento certo. Essa informação deve ser o mais completa possível, exacta e selectiva, de forma a informar criteriosamente, mas sem exagero, quem por ela se possa interessar [Courtois *et al.*, 1991]. O planeamento deve ter em linha de conta as capacidades reais do sistema de produção, avançando com soluções de substituição e analisando as causas de estrangulamentos. O planeamento baseia-se em algumas previsões, mas, realizar previsões com precisão dentro de um mercado dinâmico é uma tarefa difícil. Assim sendo, o planeamento efectuado pode/deve ser refinado à medida que informação mais precisa (isto é, factos e não previsões) esteja disponível [Courtois *et al.*, 1991].

2.4.2 Projecto

A fase de *Projecto* prende-se com a concepção do produto. Esta fase consiste na determinação da forma, dimensões e tolerâncias de cada um dos componentes do produto [Figueiredo, 1996]. O processo de projecto de um produto requer uma clara compreensão das funcionalidades e do desempenho esperado desse produto [Kalpakjian, 1995].

O engenheiro de projecto necessita de um conhecimento profundo dos processos de fabrico, das ferramentas e dos materiais a utilizar, de forma a certificar-se que cada componente pode ser

criado de acordo com a especificação [Figueiredo, 1996]. O mercado para um produto deve ser definido claramente com o auxílio do pessoal de vendas e apoiado em estudos de mercado. O projecto é uma actividade crítica, pois está estimado que 70% a 80% do custo de desenvolvimento e produção é determinado nas fases iniciais de projecto [Kalpakjian, 1995].

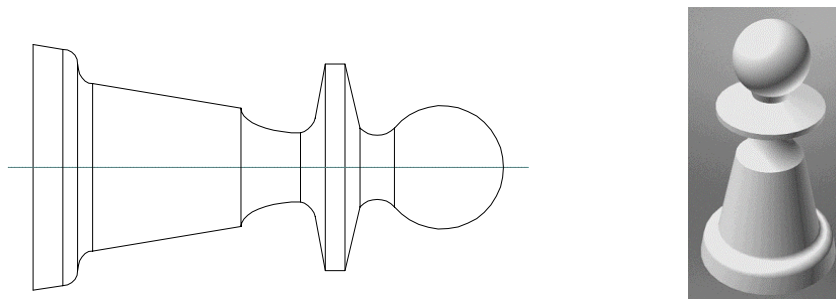


Figura 2.5 – Exemplo de produto: (a) desenho técnico; (b) imagem tridimensional

Tome-se como exemplo o fabrico de um peão de xadrez. Como resultado da fase de projecto tem-se o desenho técnico do produto (Figura 2.5a), com o respectivo dimensionamento e, eventuais representações tridimensionais (Figura 2.5b, fonte: [Rocha, 1999]).

2.4.3 Planeamento de Processos

O *Planeamento de Processos* passa pela elaboração da sequência de operações a realizar para o fabrico de um dado produto, e pela caracterização das máquinas e ferramentas necessárias à operação [Chang e Wysk, 1985] [Kusiak, 1990] [Rocha, 1999]. Com base nas especificações do projecto é elaborado um plano que atende a um ou mais factores de optimização (*e.g.*, tempo de processamento, custo de material, maximização da utilização dos recursos).

Esta tarefa é tradicionalmente efectuada por peritos, exigindo um enorme esforço e baseando-se de forma significativa na experiência destes [Rocha, 1999]. Se o engenheiro de projecto necessita de profundo conhecimento dos processos de fabrico, o responsável pelo Planeamento de Processos necessita ainda de mais, pois é dele a responsabilidade de escolher de entre os recursos existentes na fábrica, os mais adequados à execução de cada uma das operações do projecto, e em que sequência essas operações devem ser feitas. Essa escolha atende a factores como as capacidades dos recursos (*i.e.*, as operações que estes são capazes de executar), critérios de optimização como os referidos anteriormente, e requisitos/restrições impostas pelos recursos/operações (*e.g.*, em certas situações, a seguir a uma operação de soldadura, não se podem efectuar imediatamente operações de dobragem sob pena de destruir a soldadura anterior).

Continuando com o exemplo do peão de xadrez, o responsável pelo planeamento de processos deve pegar no desenho técnico do produto e identificar as operações a efectuar, de

acordo com os recursos e características da instalação fabril em questão. O caso apresentado é uma peça de revolução pelo que poderá ser efectuada num torno, partindo de um cilindro do material desejado (*e.g.*, alumínio). As operações de remoção de material escolhidas estão identificadas pelas áreas com numeração romana (Figura 2.6, *fonte*: [Rocha, 1999]).

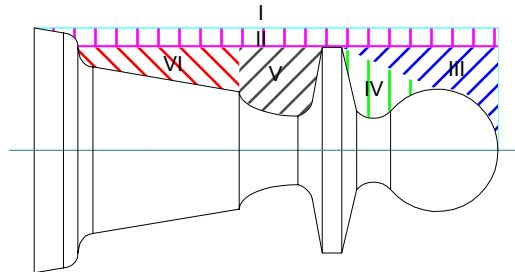


Figura 2.6 – Exemplo de produto: identificação das operações

Dependendo das características da instalação fabril poderiam ter sido escolhidas outras operações. Por exemplo, a remoção das áreas V e VI pode ser feita de uma só vez caso o torno a usar possua uma ferramenta de corte bidireccional. Contudo, se o torno em questão apenas possuir ferramentas de corte unidireccional é então necessário remover a área V com uma ferramenta e a área VI com outra. Após a identificação das operações é necessário determinar a sua ordem de execução. A ordem de execução das operações tem que obedecer a determinadas restrições do ponto de vista físico (*e.g.*, é impossível remover a área IV antes de remover a área II) e das características do material e/ou recurso a utilizar.

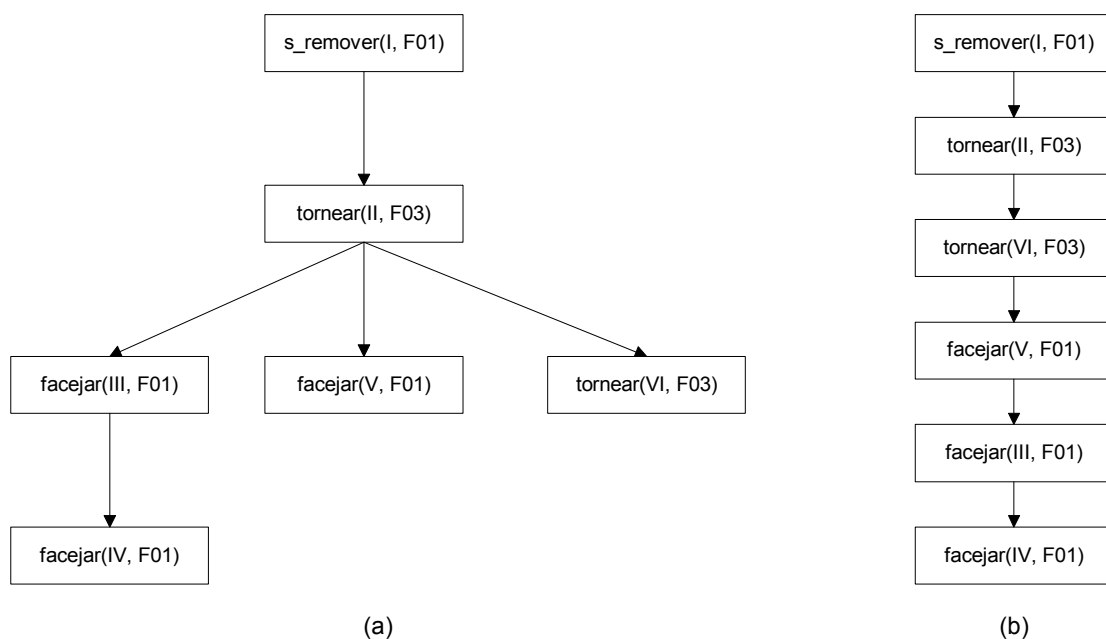


Figura 2.7 – Exemplo de produto: (a) grafo de precedências; (b) plano gerado

Na Figura 2.7a (fonte: [Rocha, 1999]) está representado o grafo de precedências entre operações, em que cada quadrado denota uma operação (segundo o modelo *operação(Área, Ferramenta)*), e as linhas entre quadrados as restrições de precedência. Na Figura 2.7b (fonte: [Rocha, 1999]) pode ver-se o plano gerado que minimiza o tempo de execução do produto [Rocha, 1999].

2.4.4 Planeamento de Produção

O *Planeamento da Produção* passa pelo estabelecimento de níveis de produção ao longo de um certo período de tempo (futuro) [Groover e Zimmers, 1984], a partir de previsões de vendas e disponibilidade de pessoal, maquinaria e/ou materiais [PDMIC, URL].

O planeamento da produção recorre normalmente a um Plano Mestre de Produção (PMP) e à aplicação de técnicas como o Planeamento de Necessidades de Materiais (MRP1) e o Planeamento de Necessidades de Capacidade (CRP).

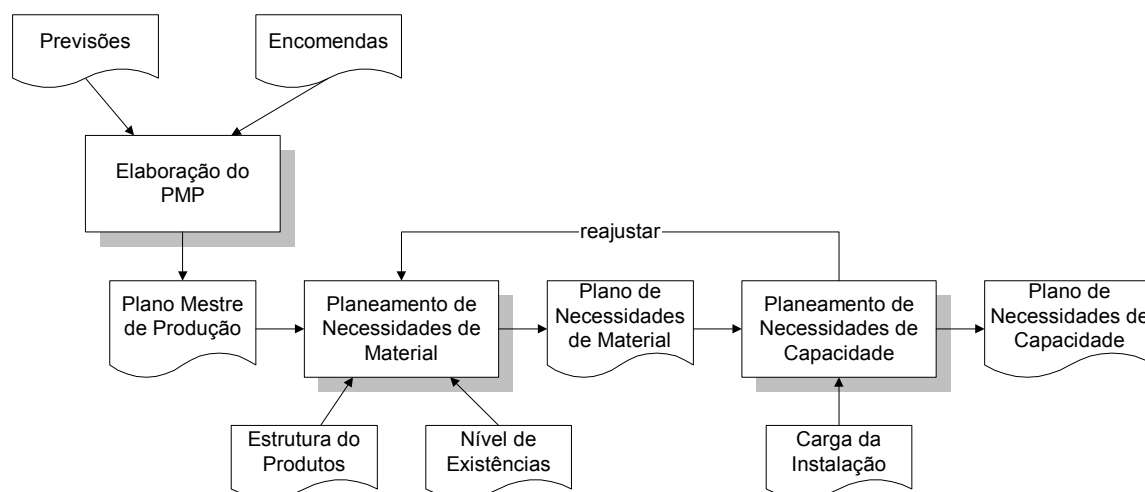


Figura 2.8 – Ciclo do planeamento da produção

O *Plano Mestre de Produção* define de forma precisa a calendarização das quantidades a produzir para cada produto acabado, considerando previsões de vendas, existências disponíveis, capacidade da instalação fabril, etc. [PDMIC, URL]. O *Planeamento de Necessidades de Material* é um sistema de controlo de existências que determina as ordens de compra e fabrico em resposta a um Plano Mestre de Produção, através da explosão das árvores de produto, ajustando-se às actuais existências de produtos e matérias primas (contabilizando também, as encomendas de matéria prima que ainda não foram entregues), e efectuando as deslocações temporais necessárias, de forma a colocar as ordens no instante correcto em que serão necessárias [FOLDOC, URL]. O *Planeamento de Necessidades de Capacidade* é uma técnica para determinar quantas pessoas e

recursos de equipamento são necessários para cumprir os objectivos de produção incorporados no Plano Mestre de Produção e no Plano de Necessidades de Material (PNM).

A Figura 2.8 representa de forma simplificada o ciclo de Planeamento da Produção. Do planeamento da produção resultam duas saídas essenciais: (i) o *Plano de Necessidades de Material*, que permite identificar as ordens de compra de componentes ou matéria prima, bem como as ordens de fabrico a lançar na instalação fabril; (ii) o *Plano de Necessidades de Capacidade* que permite identificar se existe capacidade de produção na instalação fabril para o definido no PMP e no PNM.

2.4.5 Escalonamento e Balanceamento

Através de um processo de *Escalonamento* tem-se como objectivo distribuir as operações e tarefas pelos recursos dos sistemas produtivo ao longo do tempo. A atribuição de operações/tarefas a recursos tem em consideração o estado da instalação fabril (*e.g.*, avarias, disponibilidade dos recursos) e os objectivos de produção (*e.g.*, reduzir o número de trabalhos em progresso, reduzir o atraso dos trabalhos) [Kouiss *et al.*, 1997]. Um conceito importante associado ao escalonamento é o de *Balanceamento*, que visa a distribuição equilibrada de operações pelos recursos evitando sobrecarga de determinados recursos e subcarga de outros.

O escalonamento pode ser feito segundo duas filosofias: *escalonamento de capacidade finita*, ou *escalonamento de capacidade infinita*. No primeiro caso, a carga de tarefas no sistema tem em linha de conta a capacidade máxima de produção da instalação fabril. Na segunda situação, o escalonamento é feito sem respeitar os limites físicos das máquinas, o que permite identificar as sobrecargas e estrangulamentos de produção [PDMIC, URL]. O escalonamento é normalmente considerado uma actividade de optimização da atribuição de tarefas a recursos, efectuada *a priori* [Bongaerts, 1998]. No entanto, dada a natureza estocástica (*e.g.*, quebra de ferramentas) e dinâmica (*e.g.*, entrada de novas ordens de fabrico) da produção, o escalonamento deve ser feito de forma dinâmica, em tempo real, considerando a situação efectiva da instalação fabril [Bongaerts, 1998].

Um dos maiores problemas do escalonamento é o grande esforço computacional necessário para determinar a solução óptima. À medida que o tamanho do problema aumenta (número de tarefas, número de máquinas), o tempo necessário para solucionar o problema rapidamente ultrapassa os limites razoáveis² [Bongaerts, 1998] visto que os problemas de escalonamento são de complexidade não polinomial (NP-duro ou NP-completo).

² Por limite de tempo razoável entenda-se conseguir determinar a solução em tempo útil, ou seja, enquanto ainda é válida. Não adianta de nada conseguir determinar o escalonamento óptimo se quando se acabar de o calcular as condições se modificaram.

A solução de um problema de escalonamento é normalmente representada de forma gráfica recorrendo a um *Gráfico de Gantt* [Gantt, 1919]. Um gráfico de Gantt é um gráfico bidimensional que apresenta no eixo horizontal o tempo, e no eixo vertical os recursos. Cada rectângulo no gráfico denota uma operação associada a uma dada tarefa, o que permite visualizar os intervalos de tempo atribuídos a cada operação em cada recurso.

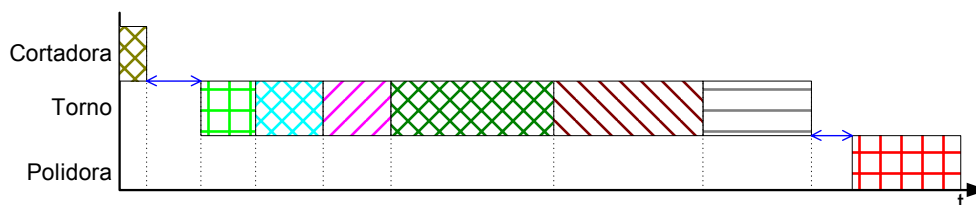


Figura 2.9 – Exemplo de produto: escalonamento das operações

A Figura 2.9 mostra o gráfico de Gantt para uma execução do peão de xadrez que se tem vindo a usar como exemplo nesta secção.

2.4.6 Execução, Inspeção e Controlo de Qualidade

A actividade de *Execução* prende-se com o fabrico real da peça através dos recursos existentes na unidade fabril. Para tal, é necessário converter as várias operações do plano de processo do produto em código executável pelos recursos programáveis, ou em descrições textuais para os operadores, no caso de operações manuais.

O caso mais frequente de recursos programáveis são as máquinas de Controlo Numérico Computadorizado (CNC), que permitem o controlo de movimentos de forma exacta e programável numa unidade NC³ através do uso de um computador dedicado [PDMIC, URL].

O *Controlo da Produção* refere-se ao ciclo de execução e acompanhamento do fabrico. Isto é, o controlo da produção compreende: (i) carregar o programa NC para a máquina-ferramenta; (ii) executar o programa; (iii) verificar se não ocorreram erros. O Controlo da Produção deve ser feito de uma forma global, vendo o sistema como um todo, e não recurso a recurso.

O objectivo da função de *Controlo de Qualidade* é o fabrico de produtos em conformidade com os requisitos e especificações de projecto [Figueiredo, 1996]. Em termos do senso comum, diz-se que um bem tem elevada qualidade se funciona de forma correcta durante um longo período de tempo. A qualidade não deve, por conseguinte, ser relegada para segundo plano, nem abordada em termos de “nível aceitável de qualidade”, pois isso implicaria que *existe* um nível

³ Controlo Numérico (*Numerical Control*): uma técnica de operação de máquinas-ferramenta em que o movimento é gerado em resposta a comandos codificados e ordenados numericamente [PDMIC, URL].

aceitável de *má qualidade* (denominado de forma eufemística de “produto fora de especificação”) [Hayes *et al.*, 1988].

A *Inspecção* de componentes e produtos é feita com o objectivo de determinar possíveis defeitos resultantes das operações de fabrico (*e.g.*, transporte, montagem, transformação) [Figueiredo, 1996]. Esta função é tradicionalmente efectuada após o fabrico das peças [Kalpakjian, 1995], podendo porém ser realizada em fases intermédias do processo produtivo (*e.g.*, verificar se o chassis de um carro está bem construído do ponto de vista “métrico”). Actualmente a inspecção já não se faz após o fabrico; *i.e.*, a “*qualidade deve ser construída no produto*” [Kalpakjian, 1995]. Quer isto dizer que os princípios de qualidade devem estar presentes em todo o processo de fabrico, do projecto à execução, e não apenas após o fabrico. Uma implicação directa desta filosofia é a de que o *controlo está nos processos* e não nos produtos; aliás, as normas de qualidade ISO 9000 são disso exemplo, pois focam-se nos processos e não nos produtos.

2.5 Produção Integrada por Computador

Desde a década de 1970 nota-se nas empresas de produção uma tendência no sentido da utilização de sistemas automáticos para a execução de muitas das tarefas relacionadas com as actividades de projecto e fabrico. Estas tecnologias denominam-se de *Projecto e Fabrico Assistido por Computador* (CAD/CAM) [Britannica, URLa].

É no entanto amplamente reconhecido que o âmbito de aplicação de computadores nas actividades de produção deve ultrapassar o projecto e o fabrico, e passar a incluir as ligações ao mercado por parte da empresa. O nome dado a esta utilização mais abrangente é o de *Produção Integrada por Computador* (CIM) [Britannica, URLa].

2.5.1 Introdução ao Conceito CIM

A *Produção Integrada por Computador* (CIM) define que o processo de fabrico é automaticamente controlado e executado através de um sistema tecnológico computadorizado [MESA, URL]. O conceito CIM refere-se à crescente integração das funções de gestão (*e.g.*, gestão comercial, pessoal, contabilidade) e de fabrico (*e.g.*, projecto, planeamento, programação), através da utilização de Tecnologias de Informação (TI), e do controlo de fabrico usando uma hierarquia de computadores [Smith *et al.*, 1996] [Britannica, URLa] [PDMIC, URL].

O CIM representa o grau mais elevado de automação na produção [Britannica, URLa]. Este conceito refere a completa automatização da fábrica, na qual todos os processos e actividades são controlados por computador, existindo em permanente circulação uma grande quantidade de

informação proveniente de todos os sectores desta. Sendo o processamento dessa informação efectuado de forma integrada entre as várias unidades funcionais da empresa, terá que haver uma total integração das bases de dados de maneira a evitar redundância e garantir a coerência dos dados [Figueiredo, 1996]. A base de dados de um sistema CIM contém informação actualizada, detalhada e correcta, acerca dos produtos, projectos, máquinas, processos, materiais, finanças, compras, vendas, etc. [Kalpakjian, 1995].

São vários os benefícios decorrentes da utilização da Produção Integrada por Computador [Rembold, 1994] [Kalpakjian, 1995]:

- reactividade a mudanças bruscas na procura e ciclos de vida muito curtos dos produtos;
- reactividade à modificação de produtos ou inclusão de novos produtos;
- aumento da produtividade na ordem dos 40% a 70%;
- melhor qualidade na especificação e concepção do produto;
- produtos de qualidade a preços baixos;
- redução de existências;
- redução dos custos de concepção e projecto em 15% a 30%;
- utilização eficiente de materiais, maquinaria, pessoal (com redução do tempo total de fabrico entre 30% a 60%; redução dos desperdícios de 20% a 50%);
- melhor controlo da produção e gestão.

Por estes motivos, a Produção Integrada por Computador é vista como uma estratégia única para melhorar a capacidade de reacção e qualidade dos Sistemas de Produção [Smith *et al.*, 1996], sendo fundamental para empresas a competir no mercado global [Chen e Su, 1996]. A Produção Integrada por Computador é uma metodologia e um objectivo, e não uma amálgama de equipamentos e computadores [Kalpakjian, 1995].

2.5.2 Suporte Tecnológico à Produção

Na secção 2.4 “Funções da Actividade de Fabrico” foram apresentadas várias unidades funcionais da empresa associadas às tarefas de produção. No que toca aos Sistemas de Produção Integrada, Kusiak (1990) identificou as seguintes áreas:

- *Projecto e especificação do produto* – determinação da forma, dimensões e tolerâncias de cada componente do produto;

- *Projecto de ferramentas e meios auxiliares* – com base na informação gerada a partir do projecto de cada componente, são projectadas as ferramentas necessárias para a execução de cada componente do produto;
- *Planeamento do processo* – elaboração do plano que define o percurso, as operações, as máquinas e as ferramentas necessárias para a execução do produto;
- *Programação* – programação das máquinas de controlo numérico (elaboração dos programas com as instruções de maquinação das operações do processo) e dos sistemas de manipulação e transporte;
- *Planeamento da produção* – envolve o estabelecimento de níveis de produção ao longo de um período de tempo, sendo dividida em planeamento de recursos e materiais, escalonamento e sequenciamento;
- *Maquinação* – operações de transformação do produto ou componentes do produto, envolvendo normalmente remoção de material ou alteração de forma (*e.g.*, fresar, cortar, dobrar, pintar, polir);
- *Montagem* – junção de componentes do produto de forma a obter componentes intermédios ou o produto final (*e.g.*, soldadura, colagem, inserção de componentes em ranhuras de outros componentes, aparafusar);
- *Manutenção* – prevenção, diagnóstico e correcção de falhas que possam ocorrer em qualquer um dos recursos do sistema;
- *Controlo de qualidade* – verificação da conformidade dos produtos finais com os requisitos exigidos no projecto. Conforme já foi referido, a qualidade deve estar sempre presente ao longo do processo de fabrico;
- *Inspecção e teste* – determinação de possíveis defeitos resultantes das operações de fabrico. Esta função está intimamente ligada com o controlo de qualidade, permitindo a identificação e remoção de eventuais erros de processo existentes em qualquer fase do fabrico;
- *Armazenamento e transporte* – armazenamento (e inventário) automático de materiais, componentes e produtos, e transporte entre as estações de trabalho e os armazéns.

De forma a cumprir os objectivos do CIM, existe um vasto conjunto de ferramentas computacionais (Figura 2.10) para integrar as várias funções do fabrico, sendo cada um dos módulos responsável por uma ou mais das funções acabadas de enumerar.

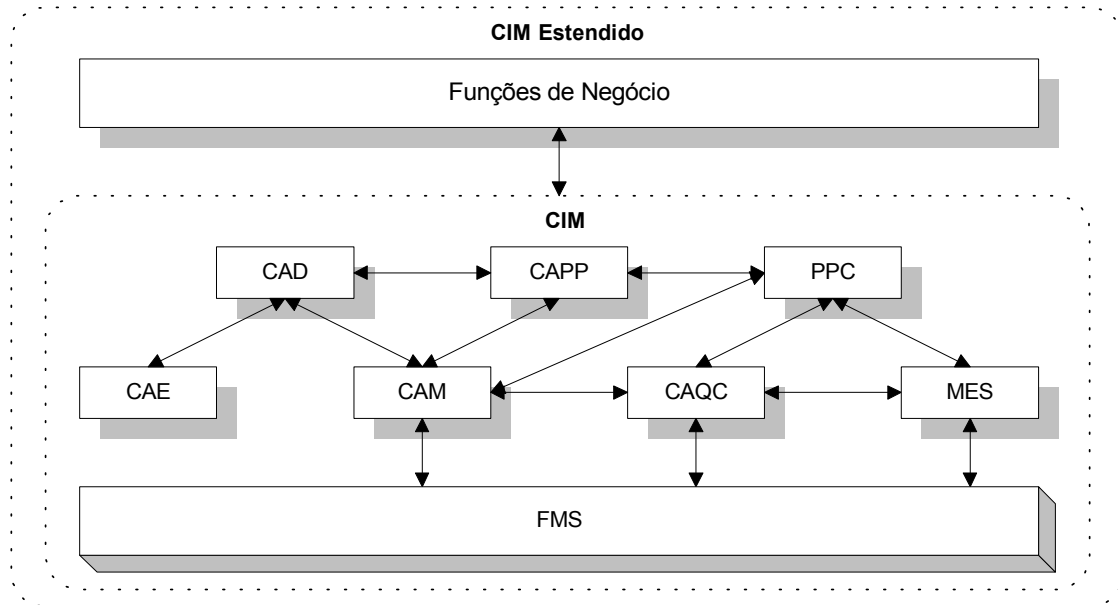


Figura 2.10 – Componentes CIM

2.5.2.1 Sistemas Flexíveis de Fabrico

Um *Sistema Flexível de Fabrico* (FMS) é uma forma de automação industrial na qual várias máquinas-ferramenta se encontram ligadas por um sistema de transporte e armazenamento de materiais/componentes, sendo todos os aspectos do sistema controlados por computador [Britannica, URLa].

Um FMS tem a capacidade de produzir uma grande variedade de produtos e componentes, com uma resposta rápida às solicitações do mercado, podendo enfrentar com êxito mudanças dos produtos e produção e na escala de produção, à medida que mudam os padrões de procura para os diferentes produtos [Kalpakjian, 1995] [PDMIC, URL] [Britannica, URLa]. A flexibilidade de um sistema prende-se com a capacidade deste em processar um número de peças diferentes em simultâneo, de forma automática e em qualquer sequência [Kalpakjian, 1995] [PDMIC, URL]. A flexibilidade de um FMS é atribuída ao facto de: (1) cada máquina estar equipada com um porta-ferramentas (dado que uma troca eficiente de ferramentas aumenta o desempenho do sistema); (2) cada operação poder ser executada em diferentes máquinas (por existir duplicação de máquinas e funcionalidades).

Os componentes de um FMS são [Kalpakjian, 1995] [Britannica, URLa]: (1) estações de trabalho, normalmente máquinas-ferramenta CNC, que executam operações de transformação, ou estações de inspecção; (2) um sistema de armazenamento e transporte de materiais/componentes que permite a entrega de peças em qualquer máquina do FMS; e (3) um computador central responsável pelo envio dos programas NC para cada máquina, bem como pela coordenação de

actividades entre máquinas. Adicionalmente, o trabalho humano é considerado o quarto componente de um FMS. Apesar do alto grau de automação de um sistema flexível de fabrico, ainda são necessários operadores humanos para gerir, reparar e manter o sistema.

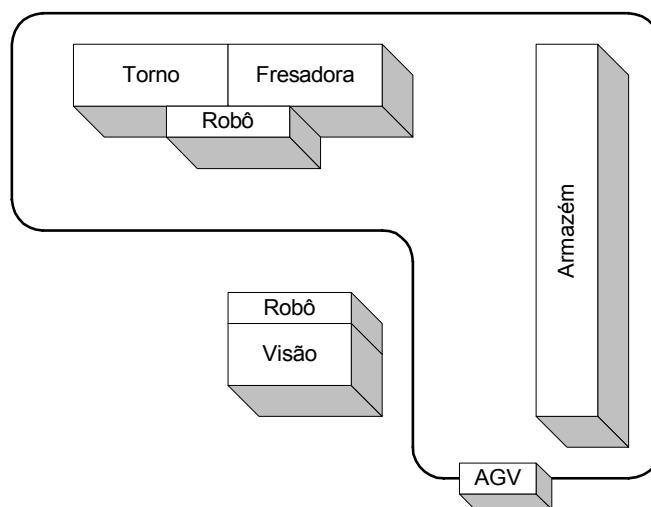


Figura 2.11 – Exemplo de um sistema FMS

A Figura 2.11 apresenta o exemplo de um FMS composto por duas células e dois sistemas auxiliares: *célula de maquinação* (torno, fresadora, e um robô); *célula de controlo de qualidade* (sistema de visão, e um segundo robô); *sistema de armazenamento* (um armazém automático com leitor de código de barras); *sistema de transporte* (um veículo autónomo).

A característica principal de um FMS é a habilidade que apresenta de mudar de um processo de fabrico para outro (*i.e.*, mudança do tipo de produto a fabricar) sem interrupções excessivas, já que a coordenação computadorizada do FMS permite que os produtos sejam fabricados a baixo custo, mesmo quando se produzem pequenas quantidades [Britannica, URLc]. A flexibilidade de um FMS permite que novos tipos de produtos sejam introduzidos no sistema, desde que as operações necessárias estejam dentro dos limites dos recursos existentes [Britannica, URLa]. Os FMS são adequados para situações onde a procura é baixa ou média e com grandes variações [Britannica, URLa]. Por este motivo, são adequados para o fabrico nos dias de hoje, e principalmente nos de amanhã, caracterizados por pequenas quantidades de produtos, com um ciclo de vida curto, e grande variedade de produtos.

2.5.2.2 Sistemas de Execução da Produção & Controlo

Os Módulos PPC (*Planeamento e Controlo da Produção*) e MES (*Sistemas de Execução da Produção*) são responsáveis por funções de Planeamento da Produção e de Controlo da Produção (Figura 2.10).

O módulo de *Planeamento e Controlo da Produção* possui uma dupla faceta. Por um lado, tem funcionalidades de Planeamento de Necessidades de Material, de Planeamento de Necessidades de Capacidade e de escalonamento *offline*, estabelecendo antecipadamente o que a empresa deve produzir, de que recursos deve dispor, quer sejam matérias-primas, componentes, pessoas, etc. Por outro lado, acompanha e controla o processo de fabrico, corrigindo eventuais erros ou desvios que possam surgir [Figueiredo, 1996].

Os *Sistemas de Execução da Produção* são sistemas de gestão ao nível da instalação fabril, que recebem ordens de produção e programas NC, e retornam dados sobre a produção, eliminando dessa forma o fosso entre os níveis de negócio e a saída de produtos da instalação fabril [PDMIC, URL]. Estes sistemas ao invés de se centrarem na utilização de materiais ou no controlo do processo produtivo, focam-se no “*produto à medida que ele se move na instalação fabril*” [MESA, URL].

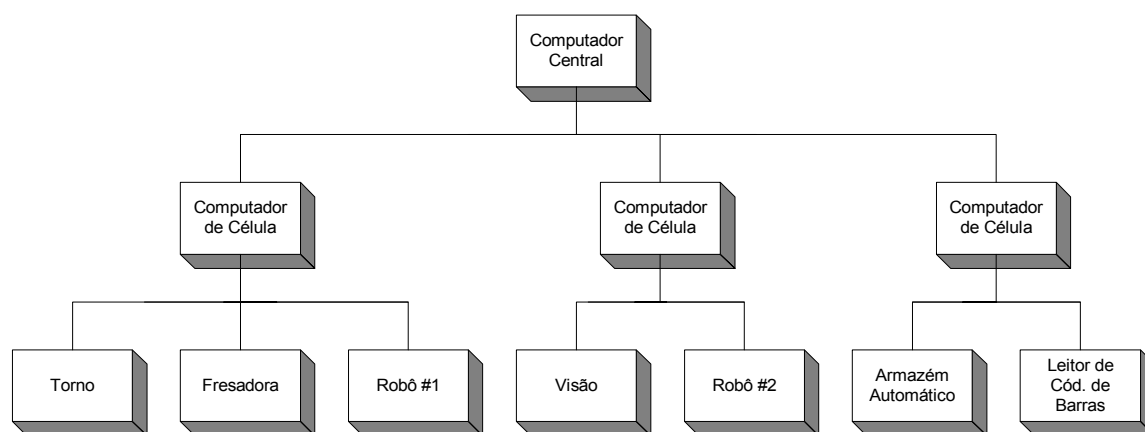


Figura 2.12 – Exemplo de arquitectura de controlo de um FMS

Continuando com o exemplo da Figura 2.11, o controlo do sistema é feito por via hierárquica, a partir de um computador central e três computadores subordinados que controlam cada uma das células de fabrico e o sistema de armazenamento (Figura 2.12, fonte: [Figueiredo, 1996]).

2.5.2.3 Apoio Computadorizado às Funções de Fabrico

São vários os módulos de apoio às funções de fabrico por computador (CAxx), cada um deles ajudando numa ou noutra área, e apresentando uma ou mais funcionalidades.

É hoje possível realizar o processo de seis passos “arte-até-peça” recorrendo a computadores [Encarta, URLb]. Os dois primeiros passos neste processo passam pela utilização de *software* de esboço para capturar as ideias iniciais do projecto e produzir desenhos técnicos de qualidade. O terceiro passo consiste na síntese de uma imagem da peça. Em seguida, é feita uma análise

estrutural da peça. O quinto passo refere-se à produção de um protótipo. Finalmente, utilizam-se programas para computador para controlar as máquinas que fabricam a peça.

Projecto Assistido por Computador

Projecto Assistido por Computador (CAD) é um termo dirigido para a utilização de ferramentas computadorizadas no projecto de um produto, desde a criação e manipulação da geometria, à produção de desenhos impressos [Kalpakjian, 1995] [PDMIC, URL]. O Projecto Assistido por Computador é um guarda-chuva para as actividades de projecto e especificação do produto, projecto de ferramentas e projecto de meios auxiliares de produção.

A utilização de um sistema CAD com visualização gráfica de alta resolução permite criar a geometria de superfícies, guardar essas geometrias na base de dados e, modificá-las posteriormente, aumentando assim a produtividade em relação ao tradicional estirador [Britannica, URLa] [MESA, URL].

Engenharia Assistida por Computador

A *Engenharia Assistida por Computador* (CAE) refere-se ao uso de computadores na ajuda às fases de engenharia (*i.e.*, análise de erros ou optimização para fabrico de um desenho/projecto) [PDMIC, URL] [FOLDOC, URLb].

As ferramentas CAE são adequadas para o estudo funcional do desempenho do produto, recorrendo a técnicas como a análise de elementos finitos, análise de forças e deformações, deflexões e distribuição de temperaturas [Kalpakjian, 1995]. As ferramentas CAE integram-se normalmente com as ferramentas CAD para formar um único sistema, permitindo ao projectista invocar as funcionalidades CAE dentro da ferramenta CAD, para ir efectuando um refinamento do projecto dentro de um único ambiente [Britannica, URLa].

Planeamento de Processos Assistido por Computador

O *Planeamento de Processos Assistido por Computador* (CAPP), como o nome indica, refere-se a uma ferramenta (ou conjunto de ferramentas) computadorizada que assiste na execução da tarefa de planeamento de processo [PDMIC, URL]. As ferramentas CAPP são peças essenciais de um sistema CIM, requerendo uma boa coordenação com os sistemas CAD/CAM (*i.e.*, a transferência de informação deve ser mútua, dinâmica, concorrente, automática e eficiente) [Kalpakjian, 1995] [Zohoor, 1998].

Em relação ao planeamento manual de processos, a utilização de sistemas CAPP oferece vantagens do tipo: redução do tempo de planeamento do processo; desenvolvimento de planos de fabrico óptimos e consistentes; redução dos custos de fabrico e planeamento; aumento da produtividade; facilidade de integração CAD/CAM; redução da dependência de peritos humanos; redução do nível de perícia necessária para elaborar um plano de fabrico; e redução do

conhecimento necessário para lidar com os problemas de planeamento [Rembold, 1994] [Rocha, 1999].

Controlo de Qualidade Assistido por Computador

O *Controlo de Qualidade Assistido por Computador* (CAQC) é responsável pelas funções de teste e inspecção de peças, bem como do controlo de qualidade da produção. As ferramentas CAQC podem ajudar ao nível dos recursos, controlando o estado das máquinas e ferramentas; ao nível dos produtos, inspeccionando e testando; ou ao nível do processo, verificando a conformidade com as especificações [Figueiredo, 1996].

A utilização de CAQC implica a existência de mecanismos de controlo de qualidade na instalação fabril (e.g., sistemas de visão para inspecção, sistemas de medição) com realimentação de valores de parâmetros de fabrico, de forma a efectuar as correcções oportunas no processo.

Fabrico Assistido por Computador

O *Fabrico Assistido por Computador* (CAM) envolve o uso de computadores para ajudar nas tarefas de planeamento, controlo e gestão das operações de produção [Groover, 1987] [Britannica, URLa]. As ferramentas CAM são responsáveis pelas funções de Programação, Maquinação, Montagem e Manutenção.

O CAM efectua o *controlo directo* do processo de fabrico através da utilização de um computador que controla o equipamento de fabrico e de manipulação de material, ou o *controlo indirecto* das operações de fabrico em curso (i.e., planeamento) [Dilworth, 1992]. O controlo directo do fabrico tem dois aspectos importantes: (i) o *controlo computadorizado de processos* que envolve o uso de computadores para executar acções de controlo que operam a instalação fabril de forma automática [Britannica, URLa]; (ii) a *monitorização computadorizada de processos* que envolve a recolha de dados na fábrica, análise e comunicação do desempenho de cada processo produtivo à gestão da instalação fabril [Britannica, URLa].

A partir de meados da década de 80, as ferramentas CAD e CAM surgem-nos com maiores índices de integração visto que, algum *software* CAM funciona “dentro” do *software* CAD e não através de base de dados partilhadas [Encarta, URLb]. No entanto, a integração destas aplicações não é tarefa fácil [Besant e Lui, 1986]. O CAD/CAM refere-se não apenas à automação das operações de fabrico, mas também à automação de elementos no ciclo de projecto-fabrico [Britannica, URLa].

2.5.3 Desafios à Implementação de CIM

Provavelmente, o maior desafio à implementação do conceito CIM consiste em definir “implementação” [Riley e Cox, 1998]. Em sentido lato, a *Produção Integrada por Computador*

inclui várias tecnologias e equipamentos que melhoram a eficiência operacional de uma dada instalação. Em sentido mais restrito, a adopção do CIM deve endereçar necessidades organizacionais tais como, reduzir o tempo de colocação no mercado; diminuir custos de mão de obra; ou aumentar a produtividade [Riley e Cox, 1998].

A tecnologia para implementar o conceito CIM (*e.g.*, CApp, FMS, CNC) é actualmente bem compreendida e dominada. No entanto, o conceito é passível de ser estendido a todas as áreas funcionais da empresa, necessitando de uma base de dados extensa e um enorme esforço de integração [Kalpakjian, 1995]. O resultado efectivo do CIM depende largamente da existência de uma infra-estrutura de comunicações integrada que ligue os computadores, as máquinas e os controladores. Problemas graves têm surgido na interligação entre equipamentos de diferentes vendedores. Actualmente a tendência é para a utilização de normas de comunicação [Kalpakjian, 1995]. Inerente à implementação CIM existem determinadas métricas que permitem verificar se as necessidades organizacionais estão ou não a ser colmatadas. A Produção Integrada por Computador não deve ser adoptada por moda, ou para seguir a concorrência, mas sim justificada racionalmente [Riley e Cox, 1998].

O conceito CIM tenta integrar e otimizar as actividades da empresa, por esse motivo, os riscos de grandes investimentos são elevados, bem como os riscos de criação de sistemas rígidos devido a problemas de escala e centralização. Além disso, a rigidez e a ênfase na automação podem levar a uma perda de flexibilidade e adaptabilidade a mudanças no ambiente [Bohez e Limsombutan, 1999]. Devido ao investimento considerável e ao risco associado à implementação das ferramentas e métodos do conceito CIM, os custos podem ser proibitivos (especialmente para pequenas e médias empresas) se a instalação for feita de uma só vez [Samadhi *et al.*, 1995] [Kalpakjian, 1995]. Um estudo efectuado em 1995 [Seidman, 1995] sugere que o conceito CIM foi sobreavaliado e não cumpriu as expectativas de retorno do investimento, facto que é devido à percepção de que os gestores de topo das empresas não estão convencidos dos benefícios do CIM. Sem o apoio dos principais decisores é natural que os cofres das empresas continuem fechados ao CIM [Riley e Cox, 1998]. Um outro factor que condiciona a evolução do CIM tem a ver com os montantes financeiros envolvidos no desenvolvimento, manutenção e integração dos sistemas computacionais [Merchant, 1988] [Ayres, 1989].

2.6 Resumo do Capítulo

Este capítulo introduziu o campo de aplicação do trabalho da tese (*i.e.*, os Sistemas de Produção). Começou-se por fazer uma breve introdução do que é um sistema do fabrico, visto pela e na sociedade, quais os seus constituintes e quais as suas funcionalidades.

Conceitos fundamentais, tais como, produto ou processo, foram apresentados de seguida. Tornou-se necessário, por conseguinte, classificar diferentes tipos de Sistemas de Produção, pelo que se apresentaram dois tipos de classificação possível, onde foram definidos conceitos como *job shop*, *flow shop*, produção por encomenda ou para existência.

Conforme foi demonstrado, o fabrico é uma actividade complexa na qual várias variáveis se inter-relacionam de forma a produzir os bens que a empresa fabrica. Assim sendo, foram apresentadas as funções da actividade de fabrico: Gestão da Produção; Projecto; Planeamento de Processos; Planeamento da Produção; Escalonamento; Execução, Inspeção e Controlo da Qualidade.

Finalmente, apresentou-se a Produção Integrada por Computador (CIM), como sendo uma metodologia para a competitividade e aumento da produtividade da empresa. Após uma breve introdução ao conceito CIM, foram apresentados vários temas relacionados com o suporte tecnológico, tais como Sistemas Flexíveis de Fabrico (FMS), Projecto Assistido por Computador (CAD) ou Fabrico Assistido por Computador (CAM). Para terminar, focaram-se alguns aspectos das dificuldades de implementação prática do conceito CIM ■

CAPÍTULO 3

A PRODUÇÃO E A ‘NOVA ECONOMIA’



As empresas com sucesso serão aquelas com coragem para rapidamente transformarem o seu negócio para capitalizar na mudança

John Scully, ex-presidente da Apple Computer.

A economia global que se pode observar actualmente é impulsionada por inúmeros e variados factores, dos quais os mais evidentes são a taxa crescente de inovação tecnológica e a crescente exigência por parte dos clientes (*i.e.*, consumidores finais ou empresas) [NGM, 1997]. Neste contexto, novos desafios são colocados ao sector produtivo para se manter uma fonte de riqueza para a sociedade.

As mudanças ocorridas na sociedade e nos mercados no final do século XX marcaram o aparecimento de um novo ambiente de negócio, em geral, e de produção em particular, caracterizado pela luta pela sobrevivência das empresas. O número de jogadores em campo, os factores de diferenciação, o elevado nível de vida do mundo ocidental, entre outros, são factores a ter em conta e que condicionam e impulsionam a economia de hoje. As velhas condições de estabilidade, certeza e previsibilidade estão a dar lugar à mudança, à incerteza e à imprevisibilidade [Kidd, 2000].

Quais as forças por trás destes acontecimentos? Que mudanças se têm observado na sociedade, comércio e indústria? E, mais particularmente, do ponto de vista do sector produtivo, que novas abordagens se devem empregar por forma a que se ajuste aos requisitos impostos pela ‘Nova Economia’?

Neste capítulo será feito uma breve apresentação da chamada ‘Nova Economia’ a partir de um paralelo como as velhas práticas da produção em massa. Em seguida é dedicado algum espaço à descrição do contexto socioeconómico em que se insere a ‘Nova Economia’, de acordo com os vectores: força laboral, aspectos ambientais, tecnologias de informação, globalização, conhecimento e adaptação. A problemática da produção é abordada neste capítulo tentando-se estabelecer um caminho entre as práticas correntes, as expectativas futuras e os requisitos para atingir essas expectativas.

3.1 Introdução

‘Nova Economia’ é um conceito difícil de definir, muitas vezes também denominada ‘Economia do Conhecimento’, ou ‘Economia Digital’, ou até ‘Economia da Atenção’. É um termo normalmente utilizado para designar a actividade económica do final do século XX, caracterizada por maneiras de pensar e fazer negócio muito distintas das existentes no período pós-guerra.

Segundo Atkinson e Court (1998) “[o termo] ‘Nova Economia’ refere-se a um conjunto de mudanças qualitativas e quantitativas que nos últimos quinze anos têm transformado a estrutura, funcionamento e regras da economia. A nova economia é uma economia de conhecimento e ideias onde a chave para a criação de empregos e melhores padrões de vida são ideias inovadoras e tecnologia embebida em serviços e produtos fabricados. É uma economia onde o risco, a incerteza e a mudança constante são a regra e não a excepção”.

A Tabela 3.1 e a Tabela 3.2 dão a conhecer alguns aspectos da ‘Nova Economia’ em oposição aos velhos dogmas de economia de produção em massa [Atkinson e Court, 1998].

Tabela 3.1 – Comparação entre a ‘Velha’ e a ‘Nova Economia’

	‘Velha Economia’	‘Nova Economia’
Mercados	Estáveis	Dinâmicos
Âmbito da competição	Nacional	Global
Estrutura organizacional	Hierárquica, burocrática	Em rede
Rel. c/ outras empresas	Isolamento	Alianças e colaborações

A dinâmica do mundo de hoje, com as mudanças políticas, com a globalização da economia, com a crescente utilização de tecnologias informáticas, e com consumidores exigentes, impõe novos padrões de conduta, ou mesmo a reescrita de velhos dogmas. O colapso das barreiras físicas, políticas, económicas e sociais, e as mudanças na ordem política e económica mundial afectam qualquer país, desafiando as tradicionais maneiras de pensar e colocando desafios nunca antes imaginados [Tapscott e Caston, 1993]. Os mercados domésticos, que antigamente eram dominados por empresas nacionais, são agora “assaltados” por um número cada vez maior de concorrentes estrangeiros com produtos de alta qualidade e baixo preço [Hunt, 1989]. Além disso, empresas recém criadas e em franco crescimento vendem desde logo para mercados globais, forçando as velhas instituições a reinventar os seus processos de forma a manterem-se competitivas [Atkinson e Court, 1998].

É prática corrente das empresas quererem controlar directamente todas as fases do processo de negócio internamente, apenas com o seu pessoal [Tapscott e Caston, 1993] [Sousa *et al.*, 2000c]. Por esse motivo, as estruturas empresariais do passado eram rígidas e “pesadas”. No entanto, uma empresa já não consegue dominar sozinha um negócio necessitando de parceiros para colmatar as suas falhas e, por sua vez, colmatar ela própria as falhas dos parceiros. A dinâmica da sociedade, e particularmente da economia, irá de forma crescente obedecer à lógica das redes, sendo estas “o arquétipo de todos os circuitos, toda a inteligência, todas as interdependências, todos os elementos económicos, sociais, ou ecológicos, toda a democracia, todas as famílias, todos os grandes sistemas, quase tudo o que é considerado interessante e importante” [Kelly, 1998].

Tabela 3.2 – A indústria na ‘Velha’ e na ‘Nova Economia’

	‘Velha Economia’	‘Nova Economia’
Organização da produção	Produção em massa	Fabrico flexível
Fonte de crescimento	Capital/trabalho	Inovação/conhecimento
Importância da investigação e inovação	Baixa a moderada	Alta
Fonte de vantagem competitiva	Diminuição de custos através de economia de escala	Inovação, qualidade, tempo de colocação no mercado e custo
Tecnologia	Mecanização	“Digitalização”

Enquanto a indústria da ‘Velha Economia’ era organizada à volta da mecanização e produção em massa, verifica-se agora uma viragem para o fabrico flexível e personalizado de produtos, bem

como a prestação de serviços por via digital e de forma personalizada. Devido ao ambiente altamente competitivo, a inovação de produtos e os nichos de mercado deixaram de ser negligenciáveis. Na ‘Velha Economia’ o capital e o trabalho eram as principais fontes de crescimento e vantagem competitiva, no entanto, à medida que os mercados se fragmentam, a tecnologia evolui e a competição cresce (oriunda por vezes de locais e intervenientes inesperados), a criatividade, adaptação e a aprendizagem, estão a tornar-se os principais factores de vantagem competitiva da ‘Nova Economia’.

A ‘Nova Economia’ realça o conceito de *Eficiência Adaptável*, isto é, a habilidade das instituições para inovar, aprender continuamente e mudar produtivamente [Atkinson e Court, 1998]. Uma *Empresa Adaptável* é aquela que possui a capacidade de se adaptar, promover a mudança e lidar com a incerteza e imprevisibilidade de forma a prosperar num mundo multimodal [Kidd, 2000].

3.2 Evolução dos Mercados e da Produção

É indiscutível que grandes mudanças ocorreram na maneira de fazer negócio na última metade do século XX. Essas mudanças puderam ser observadas gradualmente ao longo de vários anos e são o resultado da evolução da sociedade, do comércio e da indústria.

3.2.1 Enquadramento Histórico

Em meados do século XVIII, no início da Revolução Industrial, o ciclo de vida de um produto e a duração do ciclo de produção eram incomparavelmente maiores que hoje em dia. Por outro lado, a complexidade e a variedade dos produtos existentes era muito menor. De um ponto de vista social, o poder de compra e o acesso à informação eram escassos na população em geral, traduzindo-se num baixo nível de vida.

Existem dois grandes efeitos catalisadores que transformaram a produção e a sociedade nos séculos XVIII e XIX. Por um lado, o *aumento da mecanização e automatização* – a invenção e utilização de máquinas permitiu gerar grandes quantidades de energia para utilizar em tarefas árduas, tornando os processos produtivos mais expeditos, melhorando a produtividade e qualidade do trabalho, bem como, promovendo a mobilidade ao serem utilizada em novos meios de transporte (e.g., barcos e comboios a vapor); por outro lado, o *aumento da comunicabilidade* – que se refere à facilidade de comunicação de informação noticiosa e comercial a longa distância em tempos cada vez mais curtos, primeiro com o telégrafo eléctrico e mais tarde com o telefone.

De um ponto de vista de processos de fabrico, o período entre o final do século XIX e o início do século XX, é marcado pelo conceito de *produção em massa* (cujo exemplo pioneiro é dado por

Henry Ford⁴ ao criar a primeira linha de montagem em série para o Ford modelo T). A *produção em massa* traduz-se pela aplicação dos princípios de especialização e divisão do trabalho, bem como pela normalização de componentes, com ganhos de produtividade e diminuição de custos, sendo o custo por produto inversamente proporcional ao volume de produção [Britannica, URLe]. Os princípios utilizados por Ford surgiram no trabalho de Frederick Taylor⁵ que, por volta de 1890, criou a base da *Gestão Científica*. Baseando-se em estudos de tempos e métodos, cada tarefa era subdividida em tarefas mais simples, analisada e temporizada, para determinar os procedimentos a seguir na execução de cada tarefa, eliminando-se pontas mortas, e consequentemente aumentando-se a produtividade [Britannica, URLf] [Britannica, URLg].

O início do século XX é sem dúvida uma época de grandes convulsões sociais, políticas e económicas, marcado pela perda de vidas humanas, destruição material e pela reconfiguração política resultante da *1ª Guerra Mundial* (1914 – 1919) e da *2ª Guerra Mundial* (1939 – 1945) que termina na divisão da Alemanha e na criação de um mundo bipolar entre os EUA e a URSS, com a “perda” de importância das potências Europeias de outrora (França e Reino Unido) [Verbo, URLb]. De um ponto de vista económico, o início do século é marcado pela *Grande Depressão* económica (1929 – 1936) que criou barreiras proteccionistas às importações, aumentou o desemprego, e provocou a falência de inúmeros bancos e outras empresas [Britannica, URLd]. Com a reconstrução terminada e o renascimento das economias debilitadas no pós-guerra, assiste-se a uma época de relativa prosperidade. Em 1957 é assinado o Tratado de Roma que constituiu a formação da *Comunidade Económica Europeia* numa tentativa de recuperar o fosso económico da Europa relativamente aos EUA e Japão. A criação de um Mercado Único sem barreiras à circulação de pessoas e mercadorias transformou a noção de “mercado doméstico” e a maneira de gerir os negócios. Com a Queda do Muro de Berlim (em 9 de Novembro de 1989) e a consequente desintegração da URSS em 1991, a Europa de Leste abre-se ao Ocidente, criando novas oportunidades de negócio nunca antes pensadas.

A segunda metade do século é também marcada pelas telecomunicações, acesso a informação e influência dos computadores na sociedade. Por um lado as comunicações evoluíram imenso desde os tempos do telégrafo eléctrico e do telefone, surgindo agora verdadeiras redes globais de comunicações por fibra óptica, rádio, micro ondas e satélite, dando origem à grande aldeia global em que se tornou este planeta, permitindo não só a comunicação oral, como o envio de dados e

⁴ Henry Ford (1863 – 1947) Pioneiro da Indústria Automobilística, Inventor do sistema de produção em série, iniciado com o Modelo T. Fundou a Ford Automotor em 1903. Pioneiro na política de participação dos membros da empresa nos seus benefícios e inovador com o sistema de venda a crédito.

⁵ Frederick W. Taylor (1856 – 1915) Inventor e Engenheiro Americano, conhecido como o pai da Gestão Científica, um sistema de Gestão Industrial no qual as funções de gestão, planeamento e coordenação foram elevadas a uma posição principal no processo produtivo, tendo influenciado o desenvolvimento industrial de virtualmente todos os países [Britannica, URLb].

prestação de serviços. O acesso à informação e a quantidade de informação é cada vez maior, em parte devido à facilidade e rapidez de comunicação. Os computadores permitiram efectuar cálculos complexos para além das potencialidades humanas em tempos cada vez menores. O aparecimento dos computadores pessoais e de redes de computadores (principalmente a Internet), veio dar uma nova dimensão à computação, acabando com o isolamento: milhões de computadores ligados em rede e fornecendo acesso a informação e serviços de forma rápida, oferecendo novas oportunidades de negócio, quebrando barreiras e criando também toda uma série de problemas nunca antes surgidos.

3.2.2 Tendências Observadas

Na última metade do século XX verificaram-se várias alterações a nível económico, sendo possível identificar três fases distintas (com diferentes encadeamentos no tempo conforme o sector de actividade) no que se refere às condições de competitividade [Courtois *et al.*, 1991].

Segundo esses autores, a primeira fase caracteriza-se por uma oferta inferior à procura e a existência de margens confortáveis nos negócios, o que teve como consequência um forte crescimento empresarial. A frase de ordem era *produzir e depois vender*, tendo a produção como características principais o fabrico em série, prazos fixados pelo ciclo de produção, existência de armazenamento temporário entre os postos de trabalho, volume de produção e gestão manual. A segunda fase é marcada pelo equilíbrio entre oferta e procura, passando o cliente a escolher o fornecedor. Nesta fase há que *produzir o que será vendido*, pelo que é necessário fazer previsões comerciais, organizar os aprovisionamentos, equilibrar as existências e fixar as datas de entrega. A oferta excedentária face à procura fez surgir a terceira fase onde o cliente exigente cria uma concorrência aguerrida entre as empresas. A estratégia é agora *produzir o que está vendido*, sendo imperativo um controlo preciso da gestão e da produção, um controlo perfeito dos custos, alta qualidade (sem defeitos), prazos de entrega curtos, pequenas séries de produtos altamente personalizados, renovação de produtos, evolução e adaptação.

Paralelamente, a partir de 1960, novas tendências em termos gerais de mercados e também na área de produção vieram a lume [Hunt, 1989] [Kusiak, 1990] [Solberg e Kashyap, 1993]⁶ [ISC, 1994] [Kalpakjian, 1995]:

- *Produtos* – redução do ciclo de vida dos produtos; aumento significativo da variedade de produtos; aumento da complexidade dos produtos; aumento dos requisitos de qualidade; redução da dimensão de encomendas;

⁶ Solberg e Kashyap (1993) analisaram os resultados de alguns estudos independentes sobre as mudanças na Produção: [Dertouzos *et al.*, 1989] [Compton, 1988] [Hayes *et al.*, 1988] [NRC, 1986] [NRC, 1990] [USC, 1990].

- *Processos* – incremento de opções em termos de materiais e processos; avanços tecnológicos mais rápidos; aumento da diversidade tecnológica; aumento do número de instalações produtivas a nível mundial; aumento dos requisitos de especialização dos operadores; redução do tempo de colocação no mercado; aparecimento de ilhas isoladas de automação; sistematização insuficiente da tecnologia actual; necessidade de preservação de recursos naturais; restrições ambientais;
- *Práticas de gestão e factores humanos* – novas estruturas organizacionais; novas filosofias de trabalho (e.g., teletrabalho); incremento de parcerias; falta de mão de obra especializada; relutância dos jovens engenheiros em trabalhar na produção; mudanças nas filosofias de gestão;
- *Mercados* – crescente globalização de mercados; maior concorrência a nível local e global; concorrência mais activa; as condições de mercado flutuam de forma significativa; procura diversificada; mercados complexos e altamente competitivos; alargamento e globalização das cadeias de fornecimento;
- *Clientes* – maior informação por parte dos clientes; menor fidelidade dos clientes; pretendem participação mais significativa na concepção do produto; pretendem produtos de elevada qualidade e custo reduzido; pretendem entregas imediatas; pretendem personalização de produtos para necessidades específicas.

Estas tendências estão inter-relacionadas e os seus efeitos já se fazem sentir em todos os países industrializados; além disso, a competitividade futura das empresas de produção a nível global dependerá da resposta dada a estas tendências [Hunt, 1989] [ISC, 1994]. É pois necessário um novo paradigma, uma nova abordagem e maneira de pensar para perceber e lidar com a nova realidade, a situação económica mundial e a natureza dos negócios [Tapscott e Caston, 1993].

3.3 O Contexto da 'Nova Economia'

Conforme foi apresentado nas secções anteriores, a sociedade e consequentemente a economia, está muito diferente e os velhos dogmas e maneiras de pensar já não são válidos. Várias empresas estão na linha da frente desta revolução, reinventando a sua própria identidade para se tornarem mais competitivas e fortalecidas, no entanto muitas outras continuam agarradas a velhas tecnologias, processos de trabalho e filosofias de gestão [Good, 1994].

Tapscott e Caston (1993) identificaram quatro vectores com um impacto significativo na maneira de fazer negócios:

- nas últimas décadas tem-se assistido a um desenvolvimento e disseminação de enorme amplitude das *Tecnologias de Informação* (TI), que estão actualmente no núcleo de praticamente todos os negócios; a computação distribuída, as redes de computadores e principalmente a Internet facilitam o acesso à informação, a tomada de decisão e a delegação de poderes;
- a época de *estabilidade e competição* limitada do período pós-guerra acabou. Os mercados e as economias nacionais estão em transformação, as barreiras proteccionistas vão ruindo, novos intervenientes aparecem diariamente, enquanto outros desaparecem de igual forma;
- não há dúvidas que a *ordem política mundial* se modificou com o fim da Guerra Fria e consequente criação de um mundo multipolar, aberto e volátil, onde os poderes de outrora já não governam e novos governos e nações surgem na corrida à liderança e posição de destaque em termos políticos, económicos e sociais;
- de igual forma, também *as organizações estão a mudar*, dotando-se de uma estrutura mais aberta e dinâmica, com maior facilidade de reacção e adaptação a novas condições de mercado; uma estrutura mais flexível, menos hierárquica e mais orientada à equipa/projecto; a nova empresa é baseada em compromissos e não em controlo.

São vários os vectores no novo ambiente de negócio (materializando-se em “forças” que condicionam o contexto competitivo do século XXI, e existem independentemente de acções isoladas por parte de indivíduos, empresas ou nações) [Tapscott e Caston, 1993] [NGM, 1997]:

- disponibilidade e distribuição ubíqua de informação;
- qualidade e aumento das expectativas dos clientes;
- produtividade de trabalhadores peritos e de serviços;
- movimentações globais de índices salariais e perícias laborais;
- reactividade;
- globalização de mercados e competição;
- parcerias e *outsourcing*;
- responsabilidade social e ambiental, bem como escassez de recursos;
- ritmo acelerado de mudança na tecnologia;

- acesso à tecnologia em rápida expansão.

Estas forças inter-relacionam-se, sendo normalmente causa e efeito umas das outras, provocando uma miríade de eventos, que por sua vez são novamente causa e efeito de outros eventos. Devido à enorme inter-relação e à dualidade causa/efeito destas forças, é difícil identificar e resumir as forças impulsionadoras que condicionam o actual contexto socioeconómico. No entanto, identificaram-se cinco áreas de grande importância, que serão apresentados resumidamente nas subsecções seguintes. São elas: (i) Força Laboral; (ii) Responsabilidade Social e Ambiental; (iii) O Conhecimento como Activo da Empresa; (iv) Dinamismo, Reactividade e Organização; e (v) Revolução das Tecnologias de Informação.

3.3.1 Força Laboral

O aparecimento de novas indústrias levou ao aparecimento de novos empregos, ao mesmo tempo que novas tecnologias e novas formas de organização modificaram muitos dos empregos existentes.

A Tabela 3.3 apresenta alguns dos atributos que marcam a força laboral na 'Velha' e na 'Nova Economia' [Atkinson e Court, 1998].

Tabela 3.3 – A força laboral na 'Velha' e na 'Nova Economia'

	'Velha Economia'	'Nova Economia'
Objectivo	Garantia de emprego	Salários mais altos
Perícia	Específico de uma tarefa	Perícias e formação alargada
Educação	Perícia ou grau académico	Aprendizagem ao longo da vida
Rel. administração-empregados	Adversas	Colaborativas
Natureza	Estável	Risco e oportunidades

Espera-se que no futuro as empresas e os empregados abandonem o conceito de *emprego para toda a vida* e adoptem o conceito de *empregável durante toda a vida*, ao mesmo tempo que adoptam um novo contrato social onde os indivíduos são remunerados em função do trabalho e do conhecimento. As pessoas terão que seguir processos de certificação de competências, em vez do processo convencional de recrutamento e formação à medida, com as empresas a exigirem produtividade instantânea do trabalhador em resposta a oportunidades momentâneas de negócio [NGM, 1997].

À medida que se abandonou o conceito de produção em massa e se verificou um aumento de produtividade na produção, o número de empregos na produção diminuiu, aumentando o número de empregados de escritório⁷, sendo nos escritórios que é gerado a maior parte do valor acrescentado dos produtos. [Atkinson e Court, 1998]. Verifica-se que o número de empregos baseados em conhecimento aumentou percentualmente a sua quota no número total de empregos. Por outro lado, embora o número de empregos com altos salários e altos requisitos de perícia tenham aumentado, o mesmo se passou com os empregos de salários baixos e baixos requisitos [Atkinson e Court, 1998]. Ao contrário do que se possa julgar, é nos empregos de baixos salários (*e.g.*, operadores de caixa, porteiros) que se espera o maior incremento de novos empregos a criar nos próximos anos.

A produtividade dos trabalhadores peritos é actualmente um dos mais importantes aspectos a ter em linha de conta pelo gestor [Tapscott e Caston, 1993]. Nas décadas de 1980/1990 as empresas começaram por se reestruturar, recorrendo principalmente ao *downsizing*; *i.e.*, ao ‘emagrecimento’ das estruturas intermédias de gestão (vários observadores estimam que entre 1981 e o final de 1986, foram eliminadas pelo menos 10% das posições intermédias de gestão na indústria Americana [Hayes *et al.*, 1988]). Esta atitude provocou mais efeitos negativos do que os esperados [Kelley, 1999] (*e.g.*, excesso de trabalho para os empregados que não foram dispensados). Com o abandono das estruturas hierárquicas pesadas, e o recurso a uma força laboral especializada, tornou-se prática corrente o *empowerment* dos trabalhadores, consistindo no aumento de responsabilidade e autoridade para a tomada de decisões, com o objectivo de melhorar a produtividade da empresa.

3.3.2 Responsabilidade Social e Ambiental

A produção em massa está directamente ligada ao consumo em massa, o que implica, em muitos sentidos, desperdício em massa. É importante reconhecer que uma economia baseada no consumo em massa é insustentável a longo prazo, e que muito provavelmente provocará um colapso global do meio ambiente [Kidd, 2000].

O actual contexto socioeconómico exige que as empresas tenham uma atitude “verde”, dando prioridade a técnicas do tipo [Kidd, 2000]:

- *Reciclagem* – reutilização de materiais em novos produtos e embalagens;
- *Minimização* – redução ao mínimo do consumo de energia e materiais;

⁷ Nos EUA, 80% dos trabalhadores não é do sector produtivo e o número de empregos não relacionados directamente com o fabrico aumentou de 32% em 1976 para 40% em 1998 [Atkinson e Court, 1998].

- *Gestão do consumo* – oferta de bens e serviços de que os clientes realmente necessitam, e apenas pelo tempo necessário.

As responsabilidades das empresas não são apenas para com o ambiente, sendo de igual importância as relações com a sociedade (*e.g.*, com os trabalhadores, outras empresas e instituições comunitárias). Um caso de sucesso é o da empresa “The Body Shop”, que se apresentou perante os consumidores como uma empresa “verde”, criadora de produtos saudáveis e amigos do ambiente, sem testes em animais, vindo a patrocinar várias causas comunitárias com os lucros obtidos [Tapscott e Caston, 1993].

No sector de produção (tradicionalmente com grande impacto no meio ambiente), e em particular nas últimas décadas, tem-se vindo a criar uma “consciência ambiental” por forma a diminuir os estragos da sua actividade no Planeta e na qualidade de vida humana. Muito tem sido feito, por exemplo, no que toca à redução de desperdícios, redução da utilização de produtos nocivos, tratamento de resíduos tóxicos, aumento da reciclagem [Kalpakjian, 1995]. É neste momento possível dar exemplos como o do BMW série 3, no qual 81% do material usado na construção é material reciclado.

3.3.3 O Conhecimento como Activo da Empresa

O *Conhecimento* é considerado o activo mais importante na nova empresa, aquele que juntamente com a perícia dos trabalhadores lhe pode garantir vantagem competitiva [NCMS, 1996] [Thurrow, 1997] [Atkinson e Court, 1998].

As empresas estão a constatar o facto de que existe documentação, conhecimento e capital humano, que quando aproveitado e estimulado, pode formar a base para a sua expansão futura [AIIM, URL]. Por esse motivo, conceitos como a *Gestão do Conhecimento*, a *Gestão do Capital Intelectual*, ou a *Gestão dos Recursos de Conhecimento* são cada vez mais referidos. Pretende-se um esforço no sentido de capturar, organizar e partilhar a perícia dos trabalhadores (conhecimento tácito), tornando esse conhecimento acessível a quem dele necessitar, em qualquer lugar, a qualquer momento [Stuart, 1996]. A exploração de informação e conhecimento deve ser vista como uma fonte de riqueza e um meio de desenvolver vantagens competitivas sustentáveis [Kidd, 2000]. O próprio valor de mercado das empresas está cada vez mais relacionados com bens intangíveis (Investigação e Desenvolvimento, marcas registadas, talento e conhecimentos dos trabalhadores), que se tornaram tão ou mais importantes que os bens tangíveis (*e.g.*, imobiliário, capital) [Atkinson e Court, 1998] [Kidd, 2000].

Grandes repositórios de dados/informação foram criados face à evolução das tecnologias de informação (*e.g.*, aumento da capacidade de armazenamento) conjugada com o aumento de consumo e com a democratização do acesso à informação (*e.g.*, Internet, WWW). De um ponto de

vista de gestão, o que é interessante é o conhecimento que se pode extrair desses dados. Uma utilização efectiva da Gestão de Conhecimento (rotulada como uma ferramenta crítica da empresa do século XXI) garante menos erros, menos redundância, mais rapidez na resolução de problemas, melhor tomada de decisão, redução de custos de investigação e desenvolvimento, menor dependência dos trabalhadores, relações privilegiadas com os clientes, e melhores produtos e serviços – tudo somado para manter a empresa uns passos à frente da concorrência [Stuart, 1996].

3.3.4 Dinâmica, Reactividade e Organização

Estas três vertentes do contexto socioeconómico (*i.e.*, dinâmica, reactividade e organização) são analisadas conjuntamente pois existem grandes interligações entre elas. Essencialmente, a dinâmica da sociedade implica alterações na estrutura organizacional da empresa e um maior nível de reacção por parte da empresa, conforme se poderá constatar nas subsecções seguintes.

3.3.4.1 Dinamismo, Globalização e Competição

A sociedade assume-se como uma entidade pujante de vida [Atkinson e Court, 1998], conforme se pode observar pelo número de novas empresas que são criadas, gerando novos empregos, bem como o número de empresas que acabam, provocando movimentações no tecido laboral. A competição é cada vez mais feroz impulsionada por variadíssimos factores, tais como, o aparecimento de um mercado global, o aumento do número de empresas e novas tecnologias, o que leva as empresas a repensar as suas estruturas, produtos e estratégias. Demonstrativo disso é o número de escolhas possíveis existentes para os consumidores (*e.g.*, vários tipos de leite). Um outro exemplo é o das seguradoras, outrora uma indústria estável, que se viu a braços com a concorrência dos bancos, dos correios e de novas seguradoras com culturas de empresa diferentes (*e.g.*, inexistência de canais de distribuição).

A globalização colocará inevitavelmente novos requisitos na empresa que levarão ao desenvolvimento de novas tecnologias, novas estruturas e novas práticas de negócio [Kidd, 2000]. Um exemplo disso é o equilíbrio entre a necessidade de desenvolvimento de produtos adaptados às culturas “locais” de cada mercado e a obtenção de economias de escala. Segundo Fraser e Oppenheim (1997), *“estamos à beira de uma importante transformação da economia mundial no longo prazo, de uma série de indústrias locais presas a mercados nacionais a passar para um sistema global integrado de mercados disputados por intervenientes globais”*. Esta transformação já pode ser observada nos mercados de capitais, na diminuição das barreiras económicas e comerciais entre países e no aumento do investimento directo estrangeiro (*e.g.*, implantação de subsidiárias e compra de capital de empresas locais), pois as empresas sabem que um dos

requisitos actuais para se manterem competitivas é investirem globalmente para aceder a novos mercados, tecnologia e competências (talentos) [Tapscott e Caston, 1993] [Atkinson e Court, 1998].

A forte competição beneficia os consumidores, pois as empresas estão constantemente a desenvolver novos produtos e serviços de forma a ganhar novos mercados, eventualmente cobrindo as necessidades específicas de um número cada vez maior de consumidores. Além disso, a forte competição aliada aos avanços tecnológicos provocou uma diminuição do ciclo de vida dos produtos e serviços, reduzindo assim o tempo entre a sua colocação no mercado e a sua substituição. A capacidade de inovar e colocar no mercado produtos no momento certo é um factor determinante da vantagem competitiva de uma empresa [Atkinson e Court, 1998].

3.3.4.2 Cadeia de Valor

No passado as empresas tentavam ser auto-suficientes através de uma integração vertical de serviços dentro da empresa; actualmente existe uma grande tendência para o *outsourcing* de certos aspectos da actividade empresarial, o que permite focar os recursos da empresa nas áreas chave de valor acrescentado, sem deixar de prestar atenção a outras áreas não fundamentais [Tapscott e Caston, 1993].

A nova maneira de pensar a cadeia de valor acrescentado de um produto ou serviço pode ser encontrada nas seguintes citações:

Eu vejo o nosso futuro como sendo de parcerias com os nossos melhores clientes. À medida que nos aliarmos com clientes inovadores e com mentalidade de crescimento e os servirmos bem, prosperaremos e cresceremos juntamente.

Robert J. Darnall, Presidente da Inland Steel Corp.

Nós não proclamamos ter todas as respostas, apenas as parcerias para as encontrar.

Slogan publicitário da Compaq

Apenas aqueles processos onde temos excelência serão efectuados 'dentro de portas', os outros virão da empresa estendida.

Departamento de Defesa dos EUA, "Plano da Área Tecnológica para 1997"

Estas citações são bons exemplos do que se está a passar em termos de mudanças na cadeia de valor de um produto ou serviço. De uma situação de auto-suficiência e "fecho" da empresa

sobre si própria, passa-se para uma *extensão* para os clientes, para os fornecedores, para parceiros e mesmo para competidores, formando uma verdadeira rede que liga um extremo ao outro, em que estão presentes todos os participantes no processo. Os clientes envolvem-se no processo definindo os requisitos e as especificações do produto ou serviço, os fornecedores de componentes participam no projecto, também os distribuidores, e mesmo outras empresas normalmente concorrentes podem participar, desenvolvendo certas partes do produto que a empresa, *per si*, não teria capacidade de desenvolver.

Deixa-se de falar em *Empresa* para se falar em *Empresa Estendida* ou *Empresa Virtual*. Por um lado tem-se o conceito clássico, convencional, em que a *Empresa* é vista como uma entidade geradora de lucros, com controlo sobre as suas próprias acções, e com um enquadramento legal muito bem definido. Por outro lado, uma *Empresa Estendida* ou *Empresa Virtual*⁸ é um agrupamento temporário de instituições tendo em vista aproveitar uma oportunidade momentânea de negócio, que partilhando conhecimentos e recursos colaboram na criação de um produto ou serviço, corporizando as suas sinergias, ao mesmo tempo que vêm realizados os seus objectivos estratégicos [Tapscott e Caston, 1993] [McHugh *et al.*, 1995] [NGM, 1997] [Camarinha-Matos *et al.*, 1997].

É cada vez mais frequente constatar-se a geração de ideias inovadoras e de valor acrescentado em redes dinâmicas e colaborativas (*e.g.*, parcerias, alianças) [Atkinson e Court, 1998]. Bradenburger e Nalebuff (1997) pegam nas palavras de Gore Vidal⁹ e Bernard Baruch¹⁰, respectivamente, para descrever a passagem gradual de uma situação onde “*não basta ter sucesso, os outros têm que falhar*”, para uma outra onde “*não é preciso apagar a chama da vela do outro sujeito para que a tua brilhe*”. A velha máxima “negócio é guerra” está a pouco e pouco a ser abandonada, e a ser substituída por outra, onde se ouvem os clientes, trabalha-se com os fornecedores, e estabelecem-se parcerias (muitas vezes com competidores). Um novo termo é criado para descrever esta maneira de pensar: ‘coopetição’, formado pela junção de ‘cooperação’ com ‘competição’. O significado deste termo é traduzido de forma alegórica em “*negócio é cooperação para fazer a tarte e competição quando toca a dividi-la*” [Bradenburger e Nalebuff, 1997].

⁸ Camarinha-Matos *et al.* (1997) distinguem *Empresa Estendida* de *Empresa Virtual*, sendo a principal diferença a existência ou não, respectivamente, de uma empresa dominante na estrutura. Além disso, a empresa estendida tem um carácter mais duradouro que a empresa virtual. Por outro lado, McHugh *et al.* (1995) falam de Empresa Virtual como um arranjo temporário de empresas de um conjunto pré-definido ao qual eles denominam de *Rede Holónica*.

⁹ Gore Vidal (1925 –) Escritor, ensaísta, dramaturgo Americano. Conhecido pelos seus romances sobre pessoas poderosas e maquinações de grandes instituições [Encarta, URLc].

¹⁰ Bernard Baruch (1870 – 1965) Investidor, economista Americano. Conselheiro de vários presidentes dos EUA e conselheiro especial na mobilização para a 2ª Guerra Mundial [Encarta, URLa].

O novo contexto socioeconómico representa mudanças radicais na cadeia de valor, com a inclusão de novos e improváveis parceiros com um papel activo na definição e gestão de um produto, com o único objectivo de sobreviver e prosperar.

3.3.4.3 Agilidade

Vários observadores notaram que o ritmo crescente de mudança no contexto de negócio ultrapassava a capacidade de reacção das empresas. Para descrever as características em falta nestas organizações foi criado o conceito de 'Agilidade'.

A definição original de agilidade é dada por Nagel e Dove (1991):

[agilidade é] *a habilidade de prosperar e florescer num ambiente competitivo, em mudança contínua e imprevista, [e a capacidade de] responder rapidamente a mercados em mudança rápida provocada pela valorização que os clientes fazem dos produtos.*

O conceito de agilidade abarca várias áreas e técnicas que devem ser adoptadas pelas empresas, e tenta endereçar novas maneiras de as gerir de forma a garantir uma vantagem competitiva [Gould, 1997]. Um sistema ágil tem a capacidade de adaptar-se às mudanças e aproveitar as oportunidades inerentes a um ambiente dinâmico [Langer, 1999].

A *Agilidade* combina as noções de velocidade e flexibilidade; isto é, uma organização ágil é capaz de responder rapidamente a novas oportunidades e efectuar as mudanças necessárias em tempo útil e sem necessidade de interromper a sua actividade (*i.e.*, sobre o acontecimento). O conceito de *Agilidade* prende-se com a flexibilidade total e rapidez na reconfigurabilidade das cadeias de valor, ao aproveitar as oportunidades de negócio num meio em rápida mudança, tornando assim menos relevantes as previsões [EAF, URLa].

A capacidade e o tempo de reacção são pontos chave na elaboração de estratégias para fazer com que a empresa seja mais orientada para o mercado e mais oportunista, já que nos mercados globais há a necessidade de eliminar, ou pelo menos reduzir, as dependências espaciais e temporais [Tapscott e Caston, 1993]. O velho ditado “*mais vale tarde do que nunca*” foi transformado em “*mais vale nunca do que tarde*”, pois é preferível nunca ter começado o desenvolvimento de um produto, do que colocá-lo no mercado após um concorrente, ou, pior ainda, após uma mudança no mercado [Tapscott e Caston, 1993].

A agilidade necessita de cooperação, pois “*para tapar o fosso entre o visível e o realizável é necessário desenvolver um pensamento e trabalho cooperativo. A imprevisibilidade tem que ser limitada através de consultoria mútua, cenários mutuamente acordados, senso comum e uma linguagem comum*” [EAF, URLb]. Um dos primeiros desafios a ultrapassar é pois, colocar todos

os intervenientes a falar uma mesma linguagem, sendo esse um dos grandes desafios para a agilidade [CMAF, URL].

3.3.5 Revolução nas Tecnologias de Informação

Conforme se disse anteriormente, não só a situação geopolítica mundial se modificou, provocando alterações nos sistemas internacionais de negócio, mas também a estrutura organizacional da empresa teve que se adaptar a essa nova realidade política, económica e social; *i.e.*, da mesma forma que as estruturas organizacionais, o ambiente de negócio e a ordem política mundial foram dramaticamente alteradas, também as Tecnologias de Informação (TI) passaram por grandes revoluções.

É actualmente possível verificar que os custos de computação e de transmissão de dados estão em queda brusca sendo possível ‘comprar’ maior poder de computação e de largura de banda. Além disso, verifica-se a utilização de microprocessadores não exclusivamente nos sistemas computacionais, mas de uma forma quase ubíqua, nos telefones, nos automóveis, nos electrodomésticos, nos brinquedos, e nos equipamentos médicos [Atkinson e Court, 1998], sendo que um dos maiores consumidores de microprocessadores são os telecomandos de televisão, rádio, vídeo, DVD [Grehan, 1994].

Após vários anos a serem consideradas como meios para reduzir custos, as TI começaram a ser encaradas como “ferramentas” estratégicas para aumentar a produtividade, criatividade, e garantir a vantagem competitiva das empresas. Tapscott e Caston (1993) identificaram, entre outras, as seguintes grandes mudanças que ocorreram nas TI e nas suas aplicações:

- da computação pessoal para a computação em grupo;
- de ilhas computacionais para sistemas integrados;
- de normas proprietárias para normas gerais;
- da computação intra-empresa para a computação inter-empresas.

O abandono dos grandes computadores centrais para redes de Computadores Pessoais (PC), é notório. As ferramentas de Trabalho Cooperativo Apoiado por Computador (CSCW) ajudam na criação e operação de unidades organizacionais, apresentando-se como o vector essencial para garantir a sua competitividade, ultrapassando os limites do PC enquanto máquina isolada, e transformando-o numa ferramenta impulsionadora e potenciadora do trabalho de grupo (não só partilhando informação mas também permitindo a interacção dos vários elementos).

Inicialmente as TIs eram utilizadas para reduzir custos e gerir recursos físicos, financeiros e humanos, mas de forma não integrada devido à novidade da tecnologia e do conhecimento para

explorar totalmente o seu potencial. Esta situação levou à criação de sistemas isolados, fragmentados, com duplicação de conteúdo, que eram difíceis e onerosos de manter. Actualmente, é possível implementar uma arquitectura informática empresarial que funcione como espinha dorsal dos vários sistemas de informação necessários ao funcionamento da empresa. Isto é especialmente facilitado pela existência de normas gerais adoptadas pela indústria, em detrimento de normas proprietárias de vendedores específicos. A maturidade das TI e a existência e adopção de normas gerais contribuíram também para a *extensão* da computação, permitindo a ligação entre os vários parceiros, transformando a cadeia de valor manual numa cadeia de valor electrónica, através, entre outras, de acesso Internet e transferência electrónica de documentos (*e.g.*, via EDI). As tecnologias de informação tornaram-se mais que uma ferramenta de negócio, transformando-se num veículo impulsionador de novas práticas de negócio.

3.4 Produção: Presente e Futuro

No Capítulo 2 foi feita uma introdução à produção, tendo sido apresentados alguns conceitos relacionados com o apoio computadorizado à produção (*e.g.*, arquitecturas CIM). Esta secção toma por base esse capítulo, apresentando alguns problemas encontrados nos sistemas de produção actuais e, após apresentar visões sobre o futuro da produção, elabora uma lista de requisitos para os sistemas de produção do futuro.

3.4.1 Problemas com o 'Presente' da Produção

Na secção 2.5 “Produção Integrada por Computador” foram apresentados alguns temas relacionados com o apoio computadorizado às operações de produção, com especial ênfase na Produção Integrada por Computador (CIM). Nessa mesma secção foram também dissecados alguns dos problemas inerentes à implementação desse conceito.

Tudo o que foi dito na secção anterior permite constatar que a era da produção em massa está rapidamente a dar lugar a novas filosofias baseadas em fabrico flexível, personalização, inovação e qualidade, e que esta maneira de pensar tem produzido melhorias dramáticas na qualidade, custo e no tempo de colocação no mercado dos produtos [Good, 1994].

O conceito CIM pretende ser uma metodologia de trabalho, perfeitamente normalizada, tendo como objectivo a integração das funções projecto, fabrico e comercial, alicerçando-se em infra-estruturas comuns de comunicações, controlo centralizado, intervenção humana reduzida e hierarquia rígida. Em princípio, os sistemas CIM são muito centralizados, com uma base de dados e um computador central para controlar a instalação fabril [Parunak, 1996]. Ora, embora o conceito CIM tenha vindo a ser promovido universalmente, alguns problemas têm vindo a ocorrer

com a sua colocação no terreno (*vide* secção 2.5.3), e vários investigadores (*e.g.*, [Höpf, 1994], [Ueda, 1994], [Bongaerts *et al.*, 1995], [Parunak, 1996], [Tharumarajah *et al.*, 1996], [Gou e Luh, 1997], [Bussmann, 1998], [Kádár *et al.*, 1998], [Shen e Norrie, 1999]) têm vindo a aperceber-se de que o CIM não é a resposta para os sistemas de produção do futuro.

Entre outros, foram identificados os seguintes aspectos negativos:

- *Inflexibilidade* – a arquitectura CIM é fixa, independentemente dos requisitos de produção, sendo difícil expandir ou reconfigurar um processo para novos produtos;
- *Falta de robustez* – a eficiência não é garantida fora da gama operativo pré-definida; além disso, a existência de um controlador central cria a existência de um único ponto de falha;
- *Falta de adaptabilidade* – no que se refere a avarias ou ao mau funcionamento dos recursos;
- *Dificuldade de manutenção* – devido à falta frequente de dados por parte das máquinas.

As estruturas de controlo da maioria dos sistemas de produção actuais (*e.g.*, arquitecturas CIM) são definidas de acordo com uma hierarquia multinível rígida com um computador central [GM, WP] [Höpf, 1994] [van Leeuwen e Norrie, 1997]. Embora o controlo hierárquico do CIM seja adequado à produção por lotes em mercado estáveis ou em mudança lenta [GM, WP], não é adequado para a produção de lotes quase unitários em mercados altamente dinâmicos em constante mudança como os de hoje em dia [Kádár *et al.*, 1998]. Gou e Luh (1997) citam vários estudos que indicam que as estruturas hierárquicas adoptadas pelas implementações de FMS e CIM, colocam graves limitações à expansão, reconfiguração e fiabilidade dos sistemas de produção, devido à forte interdependência entre os módulos de decisão.

Por outro lado, Hayes *et al.* (1988) consideram que muitos dos problemas da indústria se devem a uma gestão inadequada. A atitude dos gestores pode ser sumariada numa conversa, onde ao lhes ser perguntado como ia o negócio, um gestor respondeu “*o negócio iria ótimo se ao menos não tivéssemos de fazer os produtos*”. Os autores entendem esta atitude como demonstrativa da visão da maioria dos gestores, que consideram os recursos de produção (maquinaria, instalações e operários) como restrições, impedimentos, devido à dificuldade de gerir a instalação fabril e garantir que tudo corre como planeado. Por esse motivo, os recursos não eram vistos como activos da empresa e sim como passivos. Como resultado, a gestão era feita “em torno” dos recursos e não “através” dos recursos, ou seja, os recursos de produção eram colocados em segundo lugar no que tocava a investimentos, o que provocou a desactualização e degradação de equipamento, edifícios e relações laborais.

O problema fundamental dos sistemas de produção é a falta de flexibilidade da estrutura produtiva e de procedimentos necessários a uma fácil adaptação a mudanças no ambiente envolvente [Tharumarajah *et al.*, 1996], já que a tecnologia de produção actual só trabalha bem dentro de um conjunto rígido bem definido de condições que, caso não sejam cumpridas, fazem com que o desempenho do sistema diminua de forma drástica e abrupta [Valckenaers *et al.*, 1997].

3.4.2 O 'Futuro' da Produção

Na secção 3.2.2 “Tendências Observadas” foram apresentadas algumas das sensibilidades que condicionam a evolução da presente organização social; na presente secção, pretende-se extrapolar por forma a prever qual o contexto que prevalecerá num futuro próximo de forma a melhor definir quais os requisitos a cumprir pelos sistemas de produção.

Obviamente, tentar adivinhar o futuro é algo perigoso e normalmente pouco científico, levando geralmente à criação de cenários irrealistas. Nesta secção serão pois apresentadas algumas perspectivas sobre a evolução da produção, tendo por base dois estudos: (i) o relatório do Comité Visionário para Produção (CVM) “Produção em 2020” [CVM, 1999]; e (ii) o relatório do projecto “Produção de Próxima Geração” (NGM) [NGM, 1997].

Em primeiro lugar importa salientar que a produção para além do momento presente (pelo menos nas próximas duas décadas) continuará a ser baseada em empresas compostas por pessoas e recursos, possivelmente distribuídos, que convertem sonhos em realidade, partindo de matérias primas e/ou componentes amigos do ambiente [CVM, 1999].

A empresa do futuro utilizará processos inteligentes e equipamento flexível e modular para alcançar novos níveis de flexibilidade e reactividade; potenciará o poder de decisão dos seus gestores com sistemas baseados em conhecimento, apoiados num “armazém” rico em conhecimento, que fornece a informação certa e necessária no momento certo [NGM, 1997]. Estes sistemas baseados em conhecimento serão garantia da preservação e evolução de competências, estruturas organizacionais, práticas e culturas de negócio. Com isto, a empresa terá de gerir não só a produtividade mas também a inovação e a criatividade [NGM, 1997].

No futuro, o conceito de produto será diferente daquele usado na produção em massa, passando do fornecimento de soluções parciais, para a contemplação das necessidades dos consumidores, para uma situação onde a cada cliente individual é oferecida uma solução personalizada [NGM, 1997] [Kidd, 2000]; ou seja, não fornecer o que é encomendado mas sim o que o cliente realmente necessita; ultrapassar os requisitos actuais para cumprir requisitos em constante evolução. Além disso, os clientes também exigirão que os fornecedores de produtos e

serviços maximizem as relações entre qualidade, serviço e preço, tornando-se necessário que a empresa encontre a sua posição no triângulo “melhor – mais rápido – mais barato” [CVM, 1999].

As empresas de produção colocarão no mercado novos produtos de uma forma expedita, uma vez que todas as actividades da empresa serão feitas de forma integrada e concorrente (*i.e.*, engenharia concorrente aplicada a todas as funções da empresa, desde o planeamento, desenho e gestão de produtos, processos, investigação, cadeia de fornecimento, *marketing*, passando pelo apoio a clientes, aspectos financeiros, e gestão de pessoal), funcionando como uma ligação entre clientes e fabricantes de produtos inovadores [CVM, 1999]. A utilização de ferramentas dirigidas para a simulação, avaliação e optimização das operações de produção permitirá a redução de custos e riscos, incrementará a reactividade e facilitará a inovação [NGM, 1997].

A necessidade de satisfazer clientes cada vez mais exigentes (procurando soluções abrangentes para os seus problemas, em vez de soluções pontuais “a pacote”), e a crescente complexidade tecnológica, facilmente ultrapassam a capacidade de uma única empresa conseguir satisfazer tais requisitos [NGM, 1997]. O trabalho em equipa é a resposta, tendo as empresas que aprender a controlar as suas competências nucleares, mesmo que essas actividades sejam transferidas para um parceiro [NGM, 1997]. Por esse motivo, a forma e identidade das empresas será radicalmente diferente da de hoje em dia. As empresas serão agregações de pessoas e recursos ligados entre si por acordos de confiança e uma infra-estrutura tecnológica que permita a rápida e eficiente criação de “estruturas virtuais” em resposta à dinâmica de mercado, quando surge uma nova oportunidade de negócio, e que da mesma forma serão desagregadas quando essa oportunidade desaparecer [CVM, 1999]. Ao longo deste processo a empresa definirá as suas competências nucleares na cadeia de valor, consideradas como recursos que podem ser combinados e adaptados em resposta às mudanças do mercado.

Segundo CVM (1999), a natureza das empresas de produção vai evoluir como resposta a mudanças no “clima” tecnológico, político e económico. Assim sendo, os factores de importância primordial para o desenvolvimento do sistema produtivo poderão ser enunciados na forma:

- o *clima competitivo*, impulsionado pelas comunicações e partilha de conhecimento, vai requerer respostas rápidas às forças de mercado;
- *clientes sofisticados*, em países desenvolvidos, vão passar a exigir produtos personalizados e de qualidade;
- a *base da competição* será a criatividade e inovação em todos os aspectos da empresa;
- o *desenvolvimento de tecnologia inovadora* para processos de fabrico irá modificar o âmbito e a escala da produção;

- a *protecção do ambiente* será essencial à medida que o ecossistema global é exaurido de recursos, devido ao aumento demográfico e ao aparecimento de novas economias assentes em alta tecnologia;
- a *informação e o conhecimento* acerca de todos os aspectos condicionantes do funcionamento da empresa e do Mercado estarão disponíveis numa forma que facilite a tomada de decisão;
- a *distribuição* dos recursos de produção (incluindo a força laboral especializada) será um factor crítico na organização das empresas de produção.

Os autores concluem que estes factores apontam para a *flexibilidade e reactividade* como características críticas dos sistemas de produção de futuro. Ora recordando o que foi dito na secção 3.3.4.3 sobre agilidade, pode concluir-se que os autores se referem a que no futuro uma empresa de produção terá que ser uma *empresa ágil no domínio da produção*.

Uma empresa ágil no domínio da produção evolui de forma contínua, adapta-se e persegue parcerias estratégicas de forma a prosperar numa economia extremamente dinâmica e exigente [Sousa *et al.*, 2000c]. Enquanto estratégia, a produção ágil preocupa-se com objectivos, estruturas, processos e recursos e não com soluções pontuais, tecnologias ou métodos, em particular. Actualmente a flexibilidade na produção está directamente relacionada com o que pode ser controlado; no entanto, a agilidade prende-se com aquilo que não pode ser controlado [Langer, 1999].

3.4.3 Propriedades Desejáveis

Nas secções anteriores foram apresentados alguns dos problemas dos sistemas de produção actuais bem como aquilo que se pensa venha a ser o futuro da produção. Nesta secção será identificado um conjunto de propriedades desejáveis nos sistemas de produção do futuro.

O consórcio NGM identificou um conjunto de atributos que um sistema de produção deve subscrever [NGM, 1997]:

- *Sensibilidade para com o cliente* – a empresa deve trabalhar com os clientes e em antecipação aos clientes, para lhes fornecer produtos e serviços que poderão evoluir de acordo com as suas necessidades;
- *Sensibilidade para com os recursos físicos* – a empresa deve empregar o conhecimento disponível acerca da problemática da produção para olhar a processos, equipamentos e instalações que sejam reconfiguráveis, escaláveis, efectivos em termos de custos, passíveis de se adaptarem às necessidades específicas de produção;

- *Sensibilidade para com os recursos humanos* – a força laboral deverá ser altamente habilitada e motivada, capaz de trabalhar num ambiente flexível e com autonomia de decisão;
- *Sensibilidade para com o mercado global* – as estratégias deverão ser desenvolvidas em antecipação a um mercado global em constante mudança com operações e infra-estruturas dirigidas aos requisitos locais;
- *Trabalho em equipa como competência nuclear* – a empresa deve formar equipas intra e inter-muros para obter e focar conhecimento e capacidades necessárias para o desenvolvimento, distribuição e suporte dos seus produtos e serviços;
- *Prática e cultura de reactividade* – a empresa deve desenvolver continuamente as suas competências nucleares, a sua estrutura organizacional, cultura e práticas de negócio, de forma a antecipar e responder em tempo útil às expectativas expressas pelos seus clientes.

Para cumprir estes requisitos as empresas necessitarão de uma rede de fábricas, fornecedores, distribuidores e centros de serviço distribuídos a nível planetário [NGM, 1997]. As empresas no futuro têm que atender a descontinuidades em termos dos processos produtivos, de modo a responder a essas descontinuidades, aproveitando-as para ganhar vantagens competitivas [Kidd, 2000].

Conforme observado por Valcknaers *et al.* (1994), é de esperar que os sistemas de produção rígidos, estáticos e hierárquicos dêem lugar a sistemas mais adaptáveis a mudanças, já que as estruturas de controlo tradicionais dos sistemas CIM não são adequadas à reconfigurabilidade e adaptabilidade a distúrbios [Bongaerts *et al.*, 1995]. Além disso, os problemas dos sistemas de produção actuais, bem como as tendências de mercado observadas, apontam para o advento de sistemas de produção distribuídos, onde entidades de produção autónomas e flexíveis cooperarão de forma coerente e coordenada [Solberg e Kashyap, 1993] [Tharumarajah *et al.*, 1996] [Sousa *et al.*, 2000c].

A nova geração de sistemas de produção deve pois caracterizar-se pelas seguintes propriedades [Sousa *et al.*, 2000c]:

- *Distribuição* – o sistema deixa de ser monolítico e passa a ser constituído por várias entidades;
- *Descentralização* – as funcionalidades estão repartidas por várias entidades do sistema;
- *Autonomia* – cada entidade do sistema possui capacidade de decisão e controlo sobre a execução das suas próprias acções;

- *Dinamismo* – o estado do sistema não é estático, alterando-se no que se refere à estrutura, comportamentos e relações entre entidades;
- *Reactividade* – a selecção das acções a executar é feita de acordo com a percepção do sistema acerca do meio envolvente;
- *Flexibilidade* – capacidade dos recursos instalados poderem atender de uma forma expedita a uma mudança de processos (definidos *a priori*), centrados no produto a fabricar;
- *Adaptabilidade* – capacidade que o sistema exibe de continuar em funcionamento perante mudanças e/ou perturbações que possam ocorrer nas linhas de produção;
- *Agilidade* – capacidade de evolução contínua e de aproveitamento de oportunidades de negócio, através da realização de alianças estratégicas.

Adicionalmente, por forma a melhor se aproximar da realidade, os sistemas de produção devem estar apetrechados de mecanismos que lhes permitam a representação e manipulação de cenários com informação incompleta [Hatvani, 1983] [Sousa *et al.*, 2000b].

Das cinco primeiras propriedades não há muito mais a dizer, no entanto é importante esclarecer um pouco mais alguns aspectos acerca da *flexibilidade*, *adaptabilidade* e *agilidade*, e principalmente da sua interligação [Sousa *et al.*, 2000c] (Figura 3.1).

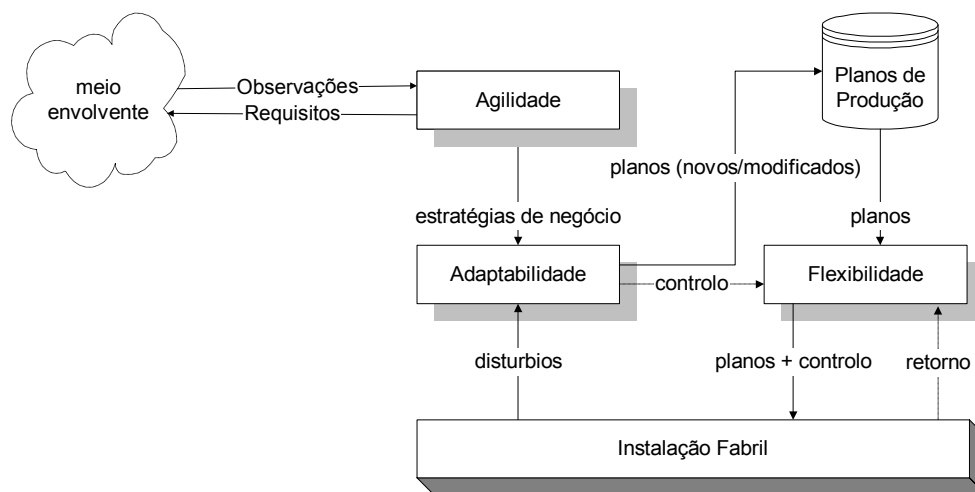


Figura 3.1 – Agilidade, adaptabilidade e flexibilidade

A *flexibilidade* relaciona-se directamente com a instalação fabril, permitindo que esta se adapte/reaja a um conjunto pré-definido de situações por forma a que seja possível ultrapassar distúrbios básicos. O retorno da instalação refere-se ao lote corrente, permitindo que se faça a descarga e execução dos programas correctos. Por outro lado, a *adaptabilidade* baseia-se na subespecificação, dando assim ao sistema a possibilidade de tratar situações não previstas da

forma menos onerosa possível. A adaptabilidade traduz-se na interpretação dos distúrbios e geração de planos alternativos, para a produção dos produtos de acordo com o estado da instalação fabril. A adaptabilidade é uma característica muito importante num sistema de produção, especialmente ao considerar o tempo de inoperância (unidades de produção paradas, por avaria ou falta de material, componentes, ferramentas, ...), que na indústria automóvel Americana, por exemplo, acarreta custos de cerca de 5.000 dólares por minuto [Parunak, 1999a]. Como já foi explicado na subsecção 3.3.4.3, a *agilidade* prende-se com opções estratégicas e aproveitamento de oportunidades. Para isso a empresa deve compreender o ambiente, o negócio e os clientes, por forma a evoluir de forma contínua. As estratégias de negócio geradas em consequência da agilidade devem ser “interpretadas pela adaptabilidade” para gerar novos produtos e planos de produção, bem como alterações a produtos e planos de produção.

Espera-se que os sistemas adaptáveis tenham um comportamento mais eficiente em termos de custo que os sistemas não adaptáveis [Sousa *et al.*, 2000c] (Figura 3.2).

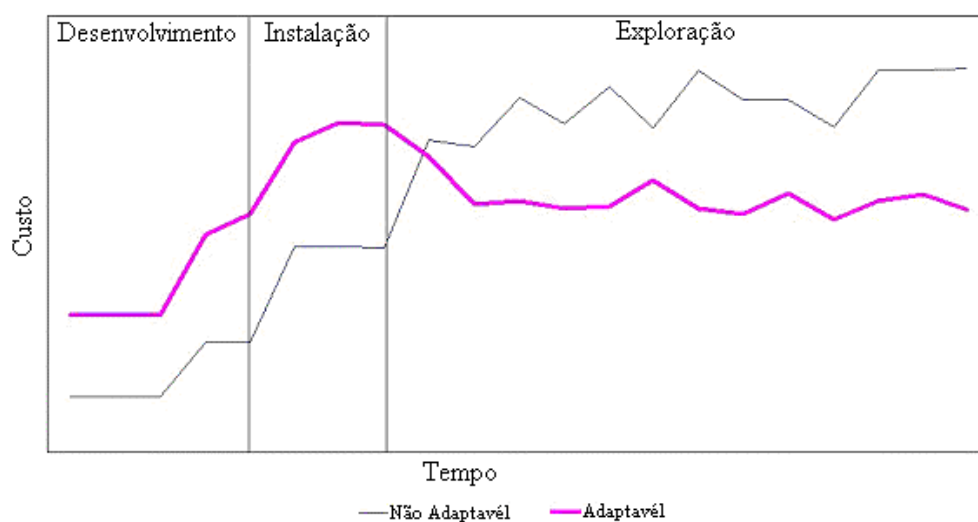


Figura 3.2 – Sistemas adaptáveis vs. não adaptáveis

Entenda-se custo em sentido lato, em que se tomam em consideração factores como custo de desenvolvimento, custo de operação, custo de não operação (*e.g.*, em caso de avaria ou falta de trabalho), etc. No primeiro sector do gráfico está representada a fase de desenvolvimento. Os custos de desenvolvimento de um sistema adaptável são mais elevados do que os de um sistema convencional, pois exige-se um esforço extra para a codificação de tolerância a falhas, reconfiguração, etc. No segundo sector (instalação) os custos continuam superiores aos dos sistemas convencionais pois há o esforço de configuração dos inúmeros componentes do sistema, isto sem falar no necessário para o retorno do *hardware*. As vantagens deste tipo de sistema virão na terceira fase, exploração, onde é esperado que os sistemas adaptáveis tenham menos custos ao

tratar automaticamente os distúrbios, com pequenas ou nenhuma paragens de produção, evitando grandes atrasos ou filas de espera. Como esta fase é a mais longa, um sistema destes deve funcionar durante anos, os custos iniciais mais altos serão atenuados pelos ganhos de exploração.

Nesta secção apresentaram-se as propriedades desejáveis para a nova geração de sistemas de produção, no entanto importa colocar ainda algumas questões: como implementar este tipo de sistemas? Qual ou quais os paradigmas a utilizar? No Capítulo 4 será apresentada uma tentativa de solução para estas questões.

3.5 Resumo do Capítulo

Este capítulo estendeu a secção 1.1 “Enquadramento”, apresentando de forma mais detalhada o contexto socioeconómico das últimas décadas do século XX. Esta época é chamada por muitos de época da ‘Nova Economia’. Na secção 3.2.1 foi apresentado um pequeno enquadramento histórico da produção e economia, bem como do contexto social e político desde o século XVIII até ao final do século XX. Esta secção serviu de base à secção 3.2.2 onde se analisaram as tendências de mercado e de produção nos últimos 40 anos do século XX.

Em seguida, após análise das mudanças e das novas características encontradas na sociedade e economia, foi apresentado o contexto socioeconómico da ‘Nova Economia’, segundo as vertentes de Força Laboral; Responsabilidade Social e Ambiental; Conhecimento como activo da Empresa; Dinamismo, Reactividade e Organização; e Revolução das Tecnologias de Informação. Importa ainda realçar a importância para a empresa de vectores como a flexibilidade, a agilidade e da importância crescente das tecnologias de informação como impulsionadoras de novas práticas de negócio.

Apresentada a ‘Nova Economia’ interessava discutir aspectos relacionados com a produção neste novo contexto, pelo que foram apresentados alguns problemas dos sistemas de produção actuais. Em seguida perspectivou-se um futuro para a produção, tendo como base as alterações ocorridas na sociedade e na economia. Finalmente, elaborou-se uma lista de características e requisitos a cumprir por uma nova geração de sistemas de produção para tentar combater as falhas actuais e responder aos desafios futuros ■

CAPÍTULO 4

SISTEMAS HOLÓNICOS DE PRODUÇÃO



Sem procedimentos holónicos, a Natureza não conseguiria produzir uma anémoma no tempo de vida da Terra.

Arthur Koestler (1905 – 1983), *Ghost in the Machine*.
Hutchinson & Co: London, 1967.

Os actuais sistemas de produção não têm respondido adequadamente às imposições e mudanças do mundo neste virar de milénio [Kádár *et al.*, 1998]. As estruturas rígidas e hierárquicas pensadas para os processos produtivos, à base de sistemas monolíticos e fechados, não se adequam a uma economia dinâmica onde a optimização não é praticável e a flexibilidade e as parcerias são imperativas.

Na secção 3.3.5 foram apresentados alguns dos problemas existentes nos sistemas de produção actuais. Nessa mesma secção identificaram-se algumas das propriedades desejáveis nos novos sistemas de produção para responderem às necessidades futuras. Mas, como implementar estes novos sistemas de produção? Como responder a todos estes requisitos? Será que já existem técnicas/ferramentas capazes de responderem a estas questões?

Neste capítulo será apresentada uma abordagem que se pretende inovadora para os sistemas de produção, denominada *Sistemas Holónicos de Produção* (HMS). Assim sendo, começa-se o

capítulo por uma referência às origens e características dos Sistemas Holónicos. Em seguida, passa-se para os Sistemas Holónicos de Produção, que como o nome indica, aplicam os conceitos holónicos à produção. O conceito HMS é depois relacionado com o contexto socioeconómico vigente e com as propriedades enunciadas na secção 3.4.3. Na secção seguinte são sumariamente apresentados outros sistemas similares. Na secção 4.4 serão apresentadas duas metodologias para a implementação de sistemas holónicos de produção: *Sistemas Multiagente* e *Programação em Lógica Estendida*. Finalmente, é apresentado o estado da arte de sistemas holónicos de produção e sistemas de produção baseados em agentes.

4.1 Introdução

As alterações que se têm vindo a registar nas relações sociais repercutiram-se necessariamente em novas maneiras de gerir e organizar os negócios, não sendo a componente da produção a excepção que confirma a regra. Da mesma forma que a agilidade é necessária nos negócios característicos da ‘Nova Economia’ (e.g., indústrias de conteúdo, vendas via Internet), também as empresas de produção necessitam de alterar os seus métodos ancestrais.

Por outro lado, as empresas de produção necessitam também de “combater” o abandono da economia baseada no sector primário e secundário, em direcção a uma economia cada vez mais baseada no sector terciário (i.e., serviços). Conforme foi dito na secção 1.3 “Justificação da Investigação”, a produção continua a ser um importante veículo de geração de riqueza numa sociedade, sendo por isso necessário encontrar novas maneiras de gerir e pensar a produção para fazer face a novos requisitos, novas culturas, e novas tendências. Analisando todos estes factores, verifica-se uma grande necessidade de rápida e eficiente adaptação dos processos produtivos de forma a acompanhar a evolução e as convulsões da sociedade contemporânea. Os sistemas convencionais, monolíticos e inflexíveis, não são a abordagem apropriada a estes problemas, sendo necessário um novo conceito.

Os *Sistemas Inteligentes de Produção* (IMS) dão uma visão futurista da produção, com capacidades inatas para responder imediata e correctamente a estímulos, diferindo dos sistemas tradicionais pela sua capacidade de adaptação a um mundo em constante mutação, sem necessidade de intervenções externas [Van Brussel, 1995]:

O Sistema Inteligente de Produção fará uso de um juízo prudencial, do bom senso, para melhor harmonizar os recursos disponíveis, humanos ou materiais, consolidando-os, indo do marketing à concepção do produto, da produção à distribuição, de forma flexível, para aumentar a produtividade e a qualidade.

Monostori (1997a) refere que os Sistemas Inteligentes de Produção são um termo criado por Hatvani e Nemes (1978) delineando a próxima geração de sistemas de produção. Tais sistemas, utilizando os resultados da investigação em Inteligência Artificial (IA), seriam capazes de resolver, dentro de certos limites, problemas sem precedentes e imprevistos, mesmo na presença de informação incompleta e imprecisa [Hatvani, 1983].

Têm entretanto surgindo metodologias para tentar dar resposta às necessidades de resolução de problemas no domínio da produção; algumas baseadas em processos de evolução das espécies, outras com base matemática, e outras mesmo com bases filosóficas. Tais sistemas (Sistemas de Produção Biónicos, Fábrica Fractal, e Sistemas Holónicos de Produção) têm aspectos em comum, realçando-se o facto de serem por natureza sistemas distribuídos, que dão especial ênfase a questões de autonomia e cooperação entre as suas entidades constituintes. O âmbito desta tese incide sobre os Sistemas Holónicos de Produção, pelo que as outras alternativas para a resolução do mesmo problema serão apenas objecto de uma consideração deveras sumária na secção 4.3.3.

Os *Sistemas Holónicos de Produção* (HMS) são um dos resultados do programa internacional de investigação “Sistemas Inteligentes de Produção”. Os Sistemas Holónicos de Produção tentam tirar partido nas actividades de produção, dos mesmos benefícios que a organização holónica fornece aos organismos vivos e às sociedades; *i.e.*, estabilidade face a perturbações, flexibilidade na mudança, e utilização eficiente de recursos [Valcknaers *et al.*, 1994] [Valckenaers *et al.*, 1997].

4.2 Sistemas Holónicos

O conceito de *Sistema Holónico* foi criado na década de 1960 pelo filósofo Húngaro Arthur Koestler¹¹ para explicar as estruturas sociais e biológicas dos seres vivos, assim como a evolução da própria vida. Desse seu trabalho surgiu o que então foi apelidado de *Holon* e de *Sistema Holónico*.

4.2.1 As Origens

Os *Sistemas Holónicos* baseiam-se na noção de *holon* [Koestler, 1967], um neologismo que, de um ponto de visto etimológico, resulta da combinação da palavra Grega *holos* que significa “todo” com o sufixo *-on* da Língua Inglesa, que sugere “parte” tal como em “*proton*” ou

¹¹ Arthur Koestler (1905 – 1983) Filósofo, Jornalista, Escritor e Crítico Britânico de origem Húngara. O seu trabalho mais conhecido é “*Darkness at Noon*” (1940), um manifesto político anticomunista. De 1950 em diante, Koestler publicou livros científicos e filosóficos [Kaupunginkirjasto, 1999]. O seu livro “*The Ghost in the Machine*” (1967) é o último de uma trilogia – “*The Sleepwalkers*” (1959) e “*The Act of Creation*” (1964) – onde Koestler desenvolve o conceito de Sistemas Hierárquicos Abertos como uma estrutura geral de organização na natureza [Höpf, 1994]. Neste seu livro Koestler também examina a noção de que as partes do cérebro humano responsáveis pela razão e emoção não estão totalmente coordenadas.

“*neutron*”¹². Um *holon* tem assim um duplo significado: “todo” e “parte de um todo”, implicando isto, que um *holon* tem uma estrutura de natureza recorrente, podendo ser constituído por outros *holons* e ser constituinte de outros *holons*.

Partindo do desejo de explicar as estruturas observadas na natureza e na sociedade Koestler (1967) introduz o conceito holónico¹³:

Nenhum Homem é uma ilha – ele é um holon. Uma entidade Janus [com duas faces] que, olhando para o seu interior, se vê a ela própria como um todo autocontido e único, [e] olhando para o exterior como uma parte dependente. A sua tendência auto-assertiva é a manifestação dinâmica da sua totalidade única, da sua autonomia e independência como um holon. A [relação] antagónica, igualmente universal, [é] a sua tendência integradora, [que] expressa a sua dependência no todo maior a que pertence. [...] A tendência auto-assertiva é a expressão dinâmica da totalidade do holon, a tendência integradora, a expressão dinâmica da sua parcialidade.

[...] onde quer que seja que encontremos sistemas ordenados [não caóticos] e estáveis na Natureza, observamos que eles são estruturados hierarquicamente, pela simples razão de que sem essa estruturação de sistemas complexos em subsistemas, não poderia haver ordem nem estabilidade – excepto a ordem de um universo morto cheio de um gás uniformemente distribuído.

Kostler propõe a palavra *holon* após analisar a “Parábola dos Dois Relojoeiros” [Simon, 1962] [Simon, 1969] de Herbert Simon¹⁴:

Num certo sítio existiam dois relojoeiros, Tempus e Hora, que fabricavam excelentes relógios. Os telefones nas oficinas de cada um tocavam frequentemente com novos pedidos. No entanto, Hora prosperava e Tempus empobrecia cada vez mais, até que perdeu o negócio. Qual a razão para tal?

Cada relógio consistia de 1000 partes. Os relógios feitos por Tempus eram desenhados de tal forma que de cada vez que ele tinha que pousar um relógio

¹² Por analogia, a tradução de “Holon” para a Língua Portuguesa seria “Holão”.

¹³ A palavra “Holónico” é criada à semelhança de “radiofónico” pela junção da palavra “holon” com o sufixo -ico que significa presença ou semelhança, assim sendo, holónico significa “que diz respeito a Holon”.

¹⁴ Herbert Simon (1916 –) Cientista Social Americano; Professor na Universidade Carnegie Mellon. O Doutor Simon foi laureado com o prémio Nobel da Economia em 1978 pelo seu trabalho sobre racionalidade limitada. É um dos fundadores da actual Inteligência Artificial. No seu livro “*The Sciences of the Artificial*” (1969), o Doutor Simon explora economia, gestão, informática, psicologia e filosofia para perceber os seres humanos e seus artefactos.

parcialmente montado, por exemplo para atender o telefone, automaticamente o relógio desmontava-se e tinha que ser montado de novo.

Hora tinha desenhado os seus relógios de tal forma que pudessem ser montadas partes com cerca de 10 componentes cada. 10 destas partes podiam ser montadas num bloco maior e finalmente 10 desses blocos eram montados para obter o relógio final. Cada parte podia ser facilmente pousada sem se desfazer.

Simon concluiu que os sistemas complexos evoluem de sistemas simples muito mais rapidamente se existirem *formas intermédias estáveis*. Além disso, a habilidade para construir sistemas complexos de grande dimensão também depende do *ambiente*.

- *Formas intermédias estáveis* – partes que resistam a distúrbios, podendo ser facilmente substituídas por novos elementos;
- *Ambiente* – com que frequência e com que gravidade surgem distúrbios no ambiente (*e.g.*, evolução tecnológica, introdução de novos produtos, encomendas urgentes, flutuações na procura de produtos ou serviços).

Uma segunda constatação feita por Koestler ao analisar as estruturas das hierarquias e/ou formas intermédias estáveis de organização nos organismos vivos e complexos sociais, foi que apesar de ser fácil identificar “todos” e “partes”, estes não existem em sentido absoluto [Höpf, 1994] [Valckenaers *et al.*, 1997] [Bongaerts, 1998]. Cada unidade de organização possível de identificar, tal como uma célula num animal ou uma família na sociedade, inclui unidades mais básicas (plasma e núcleos, pais e filhos), ao mesmo tempo que formam parte de uma unidade maior (um músculo ou uma comunidade). Daí a razão de ser da palavra *holon*, que descreve a natureza híbrida dos “todos” e “partes” em sistemas reais – os holons são simultaneamente todos autocontidos para as suas partes constituintes, e partes subordinadas quando vistos no sentido inverso.

No contexto da Teoria dos Sistemas Holónicos, a palavra “hierarquia” tem um significado mais amplo do que aquele que é normalmente atribuído pelo senso comum ou encontrado no dicionário (“*relação de serviço em que o superior tem o poder de direcção, e o subalterno o dever de obediência*” [Porto Editora, URLb]). Aqui, hierarquia representa um sistema ordenado (*i.e.*, não caótico) de partes interrelacionadas, sendo cada parte por sua vez um novo sistema ordenado de partes interrelacionadas e assim por diante.

Koestler (1967) estabelece a ligação entre o conceito de holon e a Parábola dos Dois Relojoeiros visto que, um *holon* é uma parte autónoma, estável e cooperativa de um todo maior. *Autónoma* porque tem poder de decisão no que toca às suas acções sem pedir autorização a superiores; *estável* porque reage de forma adequada quando confrontada com adversidades ou

perturbações; e *cooperativa* porque trabalha em conjunto com outros holons em projectos ou objectivos comuns [Valckenaers *et al.*, 1997] [Adam *et al.*, 1997].

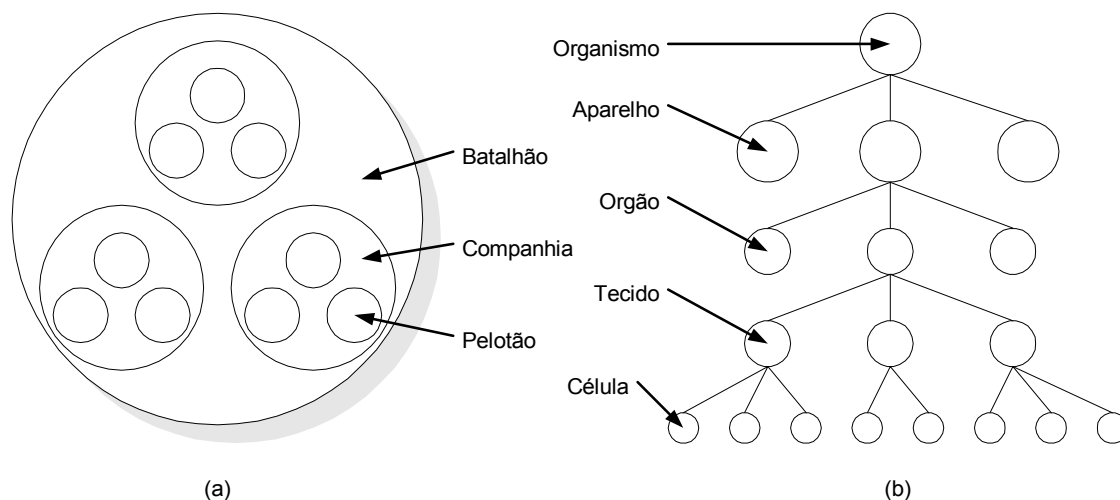


Figura 4.1 – Exemplos de sistemas holónicos: (a) militar; (b) biológico

Koestler (1967) apresenta, entre outros, exemplos de sistemas holónicos na esfera militar (Figura 4.1a) e na biologia (Figura 4.1b). Qualquer um dos holons (representados na Figura 4.1 por circunferências) pode ser considerado como um “todo”, no entanto todos eles são compostos por outros holons e por sua vez fazem parte de um todo maior. Por exemplo, um pelotão é composto por soldados (alguns praças, um alferes, sargentos, vários furriéis), e faz parte de uma companhia. O batalhão, o todo maior, por sua vez faz parte de um holon ainda maior, que são as forças armadas. Outros exemplos de holons sociais são os indivíduos, as famílias, as tribos e as nações.

O ponto forte dos sistemas holónicos reside no facto de permitir a construção de sistemas complexos que, não obstante a sua complexidade, são eficientes na utilização de recursos, altamente resilientes às perturbações (externas e internas), e adaptáveis às mudanças no ambiente em que existem [Valckenaers *et al.*, 1997] [Langer, 1999]. Todas estas características podem ser observadas em sistemas biológicos e/ou sociais.

4.2.2 Características

Um sistema holónico é capaz de se auto-organizar e evoluir de forma dinâmica, tornar-se eficiente, aumentando a sua flexibilidade, adaptabilidade e capacidade de sobrevivência.

Um holon é um componente, uma parte estável, de um todo maior, tendo autonomia nas suas decisões, cooperando com outros holons para projectos comuns, controlando parcialmente outros holons que o constituem e sendo parcialmente controlado por outros holons dos quais é

constituente. Os holons apresentam-se aqui como um paradoxo, pois se por um lado são autónomos, por outro estão simultaneamente sujeitos ao controlo de (múltiplas) entidades definidas a níveis superiores da hierarquia [Valckenaers *et al.*, 1997].

Koestler (1967) apresenta duas das propriedades fundamentais dos holons:

- *Autonomia* – a capacidade de uma entidade criar e controlar a execução dos seus próprios planos e estratégias;
- *Cooperação* – um processo no qual um conjunto de entidades desenvolve e executa planos mutuamente aceites.

Como alternativa às estruturas hierárquicas surge o conceito de *holarquia*¹⁵ para definir um sistema de holons que podem cooperar na prossecução de um objectivo ou projecto. A holarquia define as regras de cooperação entre holons, limitando assim a sua autonomia. Pela própria definição de holon, uma holarquia é um holon [Koestler, 1967]. Define-se uma holarquia como uma estrutura auto-organizada de holons que funcionam:

- como um todo autónomo em *sobreordinação* às suas “partes” constituintes;
- como partes dependentes em *subordinação* ao controlo efectuado pelo “todo”; e
- em *coordenação* com o ambiente local.

O conceito de holarquia combina as melhores características dos sistemas hierárquicos e dos sistemas distribuídos [Bongaerts, 1998]. Por um lado os holons são autónomos, o que lhes permite adaptarem-se a distúrbios e tratar os seus problemas ao nível da holarquia em que foram definidos, assegurando assim a estabilidade do sistema. Por outro lado, estão sujeitos ao controlo de níveis superiores na holarquia (com uma visão mais global do problema), o que os torna partes intermédias e subordinadas de um todo garantindo a funcionalidade de um “todo maior” [Valckenaers *et al.*, 1997].

Uma outra característica das holarquias é o seu carácter de auto-organização, permitindo criar associações dinâmicas de holons conforme os objectivos a atingir e o estado do meio envolvente [Höpf, 1994] [van Brussel *et al.*, 1998] [Langer, 1999]. Podem existir agregações de holons noutros holons definidas *a priori* na concepção do sistema, no entanto, as mais interessantes, são as resultantes das diversas interacções entre holons, cooperando num projecto comum e adaptando-se às mudanças no ambiente.

Os holons são governados por um conjunto de regras fixas (o *cânone* do sistema) e estratégias flexíveis [Tharumarajah *et al.*, 1996] [Wyns, 1999]. As regras definem os invariantes

¹⁵ A palavra “Holarquia” não existe em Português, sendo criada à semelhança de “Hierarquia” e “Anarquia” representando a “ordem, governo ou estrutura de Sistemas Holónicos”.

do sistema, ou seja, a sua configuração estrutural e/ou o seu padrão de funcionamento. As estratégias permitem escolher de entre as acções permitidas (pelo cânone), qual a acção mais adequada de acordo com as contingências do ambiente e os objectivos do holon.

Uma característica interessante dos holons é o facto de poderem pertencer a várias holarquias em simultâneo [Höpf, 1994] [Wyns, 1999], ou seja, um holon pode ser utilizado como componente de vários outros holons num mesmo instante de tempo (Figura 4.2).

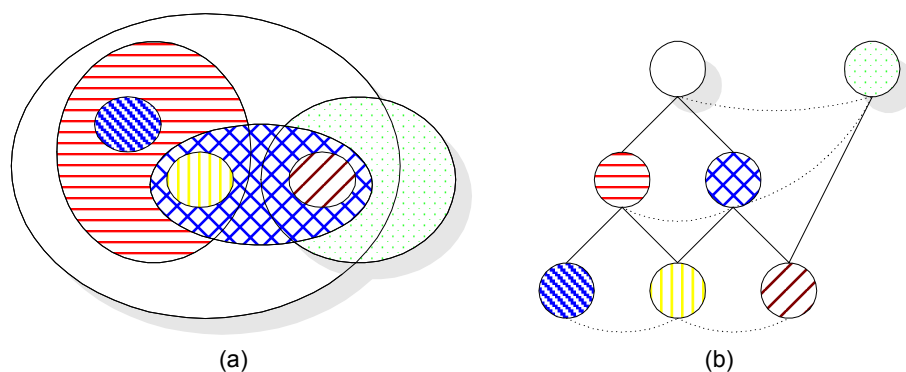


Figura 4.2 – Exemplo de um sistema holónico

A Figura 4.2 representa um sistema holónico, sendo cada holon representado por uma elipse. Conforme se pode ver nas figuras um holon pode pertencer a vários outros holons em simultâneo. A representação da Figura 4.2a permite identificar facilmente os constituintes de cada holon como sendo as elipses mais pequenas dentro das elipses maiores. Por outro lado, na Figura 4.2b é fácil identificar as relações de controlo (linhas contínuas) e de cooperação (linhas tracejadas) existentes entre os holons.

Continuando com o exemplo da Figura 4.1a, um tenente, comandante de companhia, pertencerá ao mesmo tempo à holarquia da companhia, mas também à holarquia dos oficiais. Por outro lado, como indivíduo ele pertencerá a uma família (outra holarquia) e talvez ainda a outras holarquias (por exemplo, clube de xadrez).

Wyns (1999) apresenta um outro exemplo sobre múltiplas hierarquias, relacionado com a razão e memória humana: ao recordar uma canção é possível que nos lembremos da letra mas não da música. Koestler (1967) presume que as memórias são construções abstractas e estão “armazenadas” em holarquias separadas, que interagem de forma a criar uma “imagem” global.

4.3 A Produção e os Sistemas Holónicos

O princípio base dos Sistemas Holónicos de Produção está em criar um processo de fabrico dinâmico e descentralizado, no qual os Humanos são efectivamente integrados, de forma a que as mudanças possam ser feitas de forma dinâmica e contínua [Valckenaers *et al.*, 1994a].

4.3.1 As Origens

Conforme foi apresentado nas secções anteriores, o cenário de negócio está em constante mutação, sendo necessário o sector produtivo acompanhar essa mudança para manter a prosperidade das empresas, garantindo a sua operação de forma eficiente e global, e atendendo a uma base de clientes cada vez mais exigentes.

Em Outubro de 1989 vários especialistas do meio industrial e académico propuseram um programa de investigação denominado “Sistemas Inteligentes de Produção” (IMS) [Hayashi, 1993] [Yoshikawa, 1993] com o objectivo de criar uma ciência de produção capaz de satisfazer os requisitos do século XXI, e fornecer soluções a problemas de produção independentemente do país ou organização [ISC, 1994]. O projecto traça uma agenda para alcançar e manter a excelência no sector produtivo nesta era de economia e competição global [Barram, 1994].

Para sedimentar e testar o novo conceito foram propostos seis casos de teste para estudo de viabilidade do projecto IMS. O caso de teste n.º 5 denominava-se “Sistemas Holónicos de Produção” e tinha por principal missão traduzir o conceito de Sistema Holónico para a área de produção [Valckenaers *et al.*, 1997], focando-se nas arquitecturas e tecnologias para sistemas abertos, distribuídos, inteligentes e autónomos (*i.e.*, holónicos) com aplicação em Sistemas Inteligentes de Produção [ISC, 1994]. Adicionalmente, o estudo devia examinar os requisitos para a Produção no século XXI e alcançar um melhor entendimento desses requisitos através de colaboração internacional [Höpf, 1994].

A partir deste estudo de viabilidade, decidiu-se continuar com um projecto de investigação a longo prazo que aprofundasse em detalhe os conceitos de Sistemas Holónicos de Produção. O projecto visa avançar o estado-da-arte em produção contínua, discreta e por lote, através da integração de unidades altamente flexíveis, reutilizáveis e modulares (*i.e.*, os holons) [van Leeuwen e Norrie, 1997].

4.3.2 Características de Sistemas Holónicos de Produção

Um *Sistema Holónico de Produção* é uma holarquia que abarca a totalidade das actividades de produção desde o projecto até à venda de um produto, passando pelo fabrico, *marketing* e

recepção de encomendas, para alcançar a empresa ágil de produção [Valckenaers *et al.*, 1997] [Bongaerts, 1998].

Um *holon num HMS* é uma entidade inteligente, autónoma e cooperativa, composta por um sistema de controlo inteligente, um subsistema de processamento de material e/ou informação e um ou mais seres humanos que podem ou não estar presentes a todo o momento [ISC, 1994].

De forma a sintetizar as características dos Sistemas Holónicos de Produção, Valckenaers *et al.* (1997) definem um glossário de termos holónicos¹⁶:

- *Holon* – uma entidade autónoma e cooperativa do sistema de produção, consistindo numa parte de processamento de informação e eventualmente uma parte física. Os *holons* podem ser agrupados formando outros *holons*;
- *Autonomia* – a capacidade de uma entidade criar e controlar a execução dos seus próprios planos e estratégias;
- *Cooperação* – um processo no qual um conjunto de entidades desenvolve e executa planos mutuamente aceites;
- *Holarquia* – a organização estrutural dos *holons*, em que se define a sua forma de cooperação, tendo em vista objectivos pré-definidos, e limitando-se assim a sua autonomia;
- *Atributos holónicos* – os atributos de uma entidade que a definem como *holon*. O conjunto mínimo é a autonomia e capacidade de cooperação.

Os Sistemas Holónicos de Produção (HMS) tratam os elementos de um sistema de produção, tais como máquinas e operadores humanos, e inclusive os produtos e seus componentes, como *holons*. Por natureza, os *holons* são estruturalmente recorrentes, o que permite dizer que a instalação fabril é um *holon*, e que por sua vez, a fábrica é um *holon*, e também a empresa é um *holon*, e assim por diante. Esta característica tem implicações interessantes no desenho de controladores para as máquinas e também para a organização dos processos e empresas de produção [ISC, 1994]. Por exemplo, Höpf (1994) descreve a visão holónica para o equipamento de fabrico do futuro, como sendo constituído por módulos reutilizáveis, permutáveis e incrementais, funcionando como blocos altamente normalizados, capazes de permitir qualquer tipo de automação, desde uma máquina isolada até uma linha de produção, onde estes elementos (*i.e.*, *holons*) formam uma rede autoconfigurável e adaptável para o funcionamento eficiente de unidades de produção complexas.

¹⁶ Também definido nas referências [Valckenaers *et al.*, 1994a], [van Leeuwen e Norrie, 1997], [Sousa e Ramos, 1997], [van Brussel *et al.*, 1998] e [Bongaerts, 1998].

Uma das principais vantagens dos HMS está na sua aptidão para aplicação a problemas de automação a 80% [Höpf, 1994], o que passa por uma melhor interacção entre trabalhadores e máquinas, de forma a criar sinergias, permitindo a utilização das perícias de cada trabalhador.

As raízes sociais e biológicas do conceito holónico colocam um pesado fardo nos Sistemas Holónicos de Produção, que são ao mesmo tempo uma das suas características mais interessantes e singulares. Ao derivar do modelo proposto por Koestler para a Vida, a evolução e organização de organismos e sociedades, os HMS consideram os seres humanos como entidades complementares no sistema, podendo executar o papel de um holon ou ser parte de um holon, dependendo das suas capacidades. Desta forma, os Sistemas Holónicos de Produção podem ser considerados como o conceito mais antropocêntrico na produção [Höpf, 1994].

As características que distinguem os HMS de outros sistemas similares podem agora ser enunciadas [HMSC, URL]:

- elementos autónomos, cooperativos, reutilizáveis e autoconfiguráveis;
- estrutura recursiva dos seus elementos constituintes;
- reconhecimento do papel do indivíduo pessoas para o sucesso do sistema;
- inexistência de controlo central na instalação fabril;
- integração de trabalho humano nas células de fabrico.

Adicionalmente, os HMS introduzem um novo factor de organização social ao basearem-se em hierarquias temporárias entre entidades, para satisfazer objectivos comuns e de natureza global [Bongaerts *et al.*, 1996]. Além disso, os HMS implicam a criação de canais de comunicação horizontais entre as unidades de produção para permitir a tomada de decisão de forma autónoma e a negociação de estratégias (*i.e.*, cooperação) [Höpf, 1994]. Os HMS combinam o elevado rendimento/eficiência e previsibilidade dos sistemas hierárquicos com a robustez e agilidade dos sistemas heterárquicos, tendo-se em consideração que para tal objectivo, um HMS deve possuir o poder de autoconfiguração, aprendizagem e auto-organização: os holons devem, de forma autónoma, seleccionar o melhor referencial (*i.e.*, os parâmetros apropriados), escolher o seu comportamento e determinar a sua própria estrutura [Bongaerts, 1998].

Os HMS visam responder a requisitos impostos pelo fabrico de pequenas séries de produtos e a uma grande diversidade dos mesmos [Valckenaers e van Brussel, URL] [Langer, 1999], o que se julga vir a ser uma característica dominante no futuro [van Leeuwen e Norrie, 1997]. Espera-se que os HMS tragam benefícios [Höpf, 1994] no que toca à redução dos custos originados por mudanças, maior capacidade de recuperação automática de paragens não planeadas do processo produtivo, bem como uma melhor utilização das capacidades de produção instaladas e maior

satisfação dos operadores humanos, assim como uma mais eficaz reutilização de equipamento de automação.

São inúmeros os exemplos de holons na Produção: ferramentas, máquinas, controladores PLC, matéria prima, produtos, componentes, operários, encomendas, ordens de fabrico. Estes holons “básicos” podem ainda ser combinados; por exemplo, um holon de escalonamento dinâmico pode ser composto por holons tarefa (ordens de fabrico), holons recurso e holons produto [Sousa e Ramos, 1997] [Sousa *et al.*, 1999b].

4.3.3 Relacionamento com Requisitos Impostos pela ‘Nova Economia’

Na secção 4.1 sugeriu-se o conceito de Sistemas Holónicos de Produção como solução para os requisitos de produção na ‘Nova Economia’. De facto, alguns desses requisitos são características intrínsecas dos HMS (*e.g.*, distribuição, autonomia), no entanto, o que dizer sobre as outras propriedades enunciadas e algumas das características da ‘Nova Economia’?

Tabela 4.1 – Relacionamento entre propriedades desejadas e sistemas holónicos

Propriedade desejada / Característica da Nova Economia	Sistemas Holónicos
Distribuição	Os holons são por definição distribuídos
Descentralização	Autonomia
Autonomia	Os holons são por definição autónomos
Dinamismo	Holarquias dinâmicas
Reactividade	Implementada na concepção do sistema
Flexibilidade	Autonomia dos holons e holarquias dinâmicas
Adaptabilidade	Autonomia dos holons e holarquias dinâmicas
Agilidade	Holarquias dinâmicas e cooperação entre holons
<i>Empowerment</i>	Autonomia
Empresas Virtuais	Holarquias dinâmicas e cooperação entre holons
<i>Outsourcing</i>	Cooperação com outros holons

A Tabela 4.1 apresenta a relação entre as propriedades enunciadas para os sistemas de produção do futuro e os sistemas holónicos. Da análise desta tabela verifica-se que a autonomia e cooperação intrínsecas dos holons, aliadas às holarquias dinâmicas, permitem garantir grande parte dos requisitos impostos à produção pela ‘Nova Economia’. Claro está que muitas destas

propriedades têm que ser consideradas aquando da implementação do sistema, mas do ponto de vista teórico, o conceito holónico oferece os mecanismos e as abstrações necessárias à conceptualização de sistemas para cumprir os requisitos identificados na secção 3.4.3.

4.3.4 Metodologias Similares

Os Sistemas Holónicos de Produção não são a única abordagem, porventura inovadora, aos sistemas de produção. Existem várias alternativas, com muitos factores em comum, cada uma com as suas particularidades.

Esta secção pretende apresentar, de forma resumida, duas outras aproximações aos sistemas de produção distribuídos e inteligentes, para responder aos novos requisitos da produção. Na secção 4.3.4.2 é apresentada uma sinopse destas diferentes abordagens e pontos de vista para a solução de um mesmo problema.

4.3.4.1 Fábrica Fractal

O termo ‘Fractal’ foi introduzido por Mandelbrot¹⁷ ao referir-se à geometria complexa da natureza. Warnecke (1993) segue um raciocínio semelhante aplicado-o agora à física e geometria da empresa definindo um fractal como uma unidade autónoma, a partir de pressupostos de auto-similaridade, auto-organização, auto-optimização, orientado a objectivos e grande capacidade de evolução individual. Warnecke (1993) sugere que os sistemas de produção do futuro devem ser vistos como abrangendo o “todo” da realidade usando para tal uma abordagem holística.

O termo fractal é utilizado devido à sua natureza, que de acordo com Warnecke (1993) deve ser imitada nos sistemas de produção para a criação da *Fábrica Fractal* (FF). Um fractal é uma entidade independente dentro de uma estrutura empresarial, cujos objectivos e funcionalidades podem ser descritos de forma precisa. Cada fractal na empresa age de forma independente e com os seus próprios objectivos. O objectivo global do sistema é consistente devido a um mecanismo de formação de objectivos suportado por herança. A auto-organização e a estruturação dinâmica permite à Fábrica Fractal [Warnecke, 1993] [Glistau, 1996] [Sihn, 1997], adaptar-se a mudanças.

Um fractal possui *vitalidade* pois adapta-se e reage a influências no seu ambiente [Tharumarajah *et al.*, 1996]. A evolução dos agrupamentos de fractais é conseguida à custa de reestruturação dinâmica, no interesse de servir a empresa como um todo.

¹⁷ Benoît Mandelbrot (1924 –) Matemático Francês de origem Polaca que desenvolveu a geometria fractal como um campo separado da matemática [Encarta, URLd].

4.3.4.2 Produção Biónica

A estrutura e funcionamento da Vida exhibe comportamentos autónomos e espontâneos, bem como harmonia social dentro de certas relações hierárquicas [Tharumarajah *et al.*, 1996]. A *Produção Biónica* [Okino, 1989] [Ueda, 1992] [Okino, 1993] [Ueda, 1994] observa as estruturas de sistemas biológicos e propõe formas de resolução de problemas (na produção) baseados nestas observações.

Os *Sistemas de Produção Biónica* (BMS) têm como objectivo tratar as mudanças de natureza dinâmica no ambiente interno e externo do ciclo de vida de um produto, desde o projecto até ao seu desmantelamento, baseando-se em conceitos de inspiração biológica, tais como autodesenvolvimento, auto-organização, adaptação e evolução [Ueda e Vaario, 1997]. O objectivo principal da produção biónica é modelar um sistema de produção imitando a simbiose e metamorfose biológica. Ueda (1994) acredita que esta visão levará à criação de sistemas de produção mais efectivos e inteligentes onde todos os elementos funcionam de uma maneira mais harmoniosa e estável do que nos sistemas actuais.

O conceito primário da produção biónica é o *modelon* [Tharumarajah *et al.*, 1996]. Um *modelon* é uma hierarquia composta por *modelons* de nível inferior, operadores e uma memória de trabalho para a troca de informação entre *modelons*. Esta estrutura é utilizada para a realização de relações parte-todo, autodecisão, integração e harmonia entre entidades autónomas.

4.3.4.3 O Que Há de Novo Nestas Abordagens

Tharumarajah *et al.* (1996) analisaram e compararam as abordagens Biónica, Holónica e Fractal para sistemas de produção. Também Butala e Sluga (1998) e Sousa *et al.* (2000c) apresentam um breve comparação entre os três conceitos (HMS, FF e BMS). Bohez e Limsombutan (1999) comparam a abordagem fractal com a holónica, e Glistau (1996) apresenta uma comparação entre estes dois conceitos e os *Sistemas de Produção Baseados em Agentes*¹⁸ (SPBA). HMS (URL) apresenta uma lista de características distintas dos HMS e compara esta abordagem com o conceito CIM. Parunak (1996) compara a abordagem dos Sistemas de Produção Baseados em Agentes com as abordagens convencionais.

No essencial, o que estas abordagens apresentam de inovador em relação ao conceito CIM é:

- *Distribuição* – o sistema é composto por várias entidades em vez de um único bloco monolítico, sendo a funcionalidade do sistema resultante da coordenação/cooperação entre essas entidades;

- *Descentralização* – as entidades, além de distribuídas, têm também competências descentralizadas, não havendo uma única entidade para cada funcionalidade, mas estando essa funcionalidade distribuída por várias entidades;
- *Autonomia* – as entidades do sistema são independentes até certo ponto, não estando subordinadas ao controlo de uma entidade superior;
- *Flexibilidade* – o sistema possui uma estrutura flexível, não rígida, que pode ser modificada rápida e eficazmente (*e.g.*, hierarquias temporárias);
- *Resiliência* – o sistema como um todo apresenta um elevado grau de resistência (*i.e.*, o oposto de fragilidade) a situações imprevistas, continuando em operação, normalmente através de adaptação da sua estrutura interna e das ligações entre entidades.

O conceito CIM apresenta, no entanto, duas características que não são possíveis de encontrar nestes sistemas: *ótimo teórico* e *previsibilidade*. Devido à distribuição, descentralização, concorrência de actividades e não determinismo destas abordagens, a solução óptima teórica e a previsibilidade do sistema é muito difícil (senão impossível) de obter em casos reais [Gasser *et al.*, 1987]. No entanto, a Produção Holónica pode ser considerada como uma tecnologia complementar ao CIM, e como uma implementação avançada do conceito CIM [Höpf, 1994], no caso de se entender CIM como “Integração por Computador” e não pelo prisma de arquitectura correntemente utilizada. Além disso, a adopção de uma arquitectura distribuída diminui a rigidez e centralização de um sistema, ao mesmo tempo que aumenta a sua flexibilidade [Kádár *et al.*, 1998].

Na Produção Holónica, entidades individuais juntam-se em hierarquias temporárias para alcançar um objectivo comum [Bongaerts *et al.*, 1996]. Adicionalmente, ao invés de uma estrutura rígida tipo comando/obediência como no CIM, um holon possui responsabilidade e autonomia de decisão [Höpf, 1994]. O mesmo pode ser dito para os fractais e para os *modelons* da Fábrica Fractal e da Produção Biónica, respectivamente.

Ao contrário da Produção Biónica, a Produção Holónica não é baseada em factos funcionais da natureza, mas sim em explicações filosóficas da evolução [Höpf, 1994]. De um ponto de vista operacional, os fractais efectuem formação iterativa e concorrente de objectivos, ao passo que os holons empreendem um planeamento conjunto através de negociação e cooperação com outros holons [Bohez e Limsombutan, 1999].

Uma novidade introduzida nos HMS, FF e BMS é a estrutura recorrente das suas entidades constituintes, isto é, holons feitos de holons, auto-semelhança fractal a diferentes escalas, e

¹⁸ O conceito de *Agente* e *Sistema Multiagente* é abordado na secção 4.4.1. Na secção 4.4.1.3 é discutida a aplicabilidade dos agentes e sistemas multiagente à produção, *i.e.*, os Sistemas de Produção Baseados em Agentes.

modelons feitos de *modelons*. Além disso, todas as técnicas apresentam orientação a oportunidades de negócio e orientação a projecto, garantindo maior flexibilidade que o conceito CIM.

4.4 Mecanismos de Prototipagem

Na secção 4.2 foi apresentado o conceito holónico, e na secção 4.3 apresentou-se o conceito de Sistema Holónico de Produção. Estas apresentações foram feitas descrevendo e enumerando as características de tais sistemas, mas nada foi dito sobre a sua implementação.

Nesta secção apresentam-se duas vias para a realização de sistemas holónicos de produção, nomeadamente os *Sistemas Baseados em Agentes* e a *Programação em Lógica Estendida*, que facilitam a rápida prototipagem de um sistema que concretize o conceito holónico.

4.4.1 Sistemas Baseados em Agentes

Os *Sistemas Baseados em Agentes* apresentam-se como uma técnica recente no campo das Ciências da Computação, sendo considerados como “o próximo avanço significativo no desenvolvimento de software” [Sargent, 1992]. Por sistemas baseados em agentes, entenda-se um sistema no qual a principal abstracção utilizada é o *agente* [Wooldridge, 1997], à semelhança dos objectos na análise de um problema orientada pela especificação dos seus objectos constituintes.

4.4.1.1 Agentes

Pode dizer-se que o trabalho em *Sistemas Baseados em Agentes* começou aquando da investigação em *Inteligência Artificial* (IA)¹⁹, pois desde sempre o objectivo da IA foi a criação de entidades computacionais inteligentes e racionais. O termo *agente* surge mais tarde na terminologia da IA e ultimamente tem sido usado para caracterizar diversos tipos de sistemas, sendo no entanto normalmente associado à comunidade de investigação em IA.

De um ponto de vista não computacional, a palavra “agente” é definida de forma muito semelhante em vários dicionários [Verbo, URLa] [Porto Editora, URLa] [Merrian-Webster, URLa], podendo ser resumida como *algo que age, capaz de produzir um efeito*. Efectivamente, a etimologia Latina da palavra agente é “*agere*”, que significa *agir*.

¹⁹ A *Inteligência Artificial* é o ramo da informática que procura fazer os computadores exibirem comportamentos similares aos dos seres humanos (*i.e.*, o conceito de que os computadores podem ser programados de forma a assumir capacidades tais como aprendizagem, raciocínio, adaptação e autocorreção) [PDMIC, URL].

A comunidade científica dá várias definições para o termo “Agente”, não existindo nenhuma definição consensual. No entanto, analisando várias definições existentes na literatura ([Ferber, 1993] [Durfee e Rosenschein, 1994] [Genesereth e Ketchpel, 1994] [Maes, 1994] [White, 1994] [Coelho, 1995] [Russel e Norvig, 1995] [Hayes-Roth, 1995] [Nwana, 1996a] [Bradshaw, 1997a] [Gentia, 1997] [Franklin e Gasser, 1997] [Green *et al.*, 1997] [Jennings *et al.*, 1998] [Wooldridge, 1999] [Chiariglione, URL] [FOLDOC, URLa]) é possível sumariar os pontos mais importantes de cada uma nas seguintes definições:

Um agente é uma entidade real ou virtual, persistente, que habita um determinado ambiente (complexo), sentindo esse ambiente através de sensores e, agindo de modo autónomo e flexível através de actuadores para a execução de tarefas.

Um agente social é um agente que comunica com os utilizadores através de uma interface possivelmente multimédia, e com os seus pares através de uma linguagem expressiva de comunicação entre agentes, para efectuar uma tarefa colectiva em benefício próprio e/ou dos seus utilizadores.

Um agente inteligente é um agente que possui ‘crenças, obrigações, objectivos’ e raciocina para interpretar as suas percepções, resolver problemas, extrair conclusões e determinar acções, sendo capaz de aprender com o resultado das suas decisões passadas.

Um agente móvel é um agente com a capacidade de se transferir entre os nós de uma rede electrónica (heterogénea). O agente escolhe quando e para onde migrar, suspendendo a sua execução num ponto arbitrário, movendo-se para outra máquina e continuando a perseguir os seus objectivos a partir do ponto em que os deixara.

Das definições dadas em epígrafe e do que se pode encontrar na literatura, é possível identificar um conjunto de atributos que um agente pode possuir [Wooldridge e Jennings, 1995] [Franklin e Gasser, 1997] [Foner, 1997] [Chiariglione, URL]:

- *Autonomia* – os agentes operam sem intervenção directa de seres humanos ou outros, e possuem algum tipo de controlo sobre as suas acções e estado interno; *i.e.*, a capacidade para criar e controlar a execução dos seus próprios planos e/ou estratégias;
- *Capacidade social* – os agentes interagem uns com os outros (e possivelmente com humanos) através de uma (ou mais) linguagens de comunicação;

- *Reactividade* – os agentes têm noção do ambiente (que tanto pode ser um mundo físico ou lógico), e respondem adequada e atempadamente às mudanças ocorridas nesse ambiente;
- *Proactividade* – os agentes não se limitam a reagir às mudanças no seu ambiente, mas são capazes de tomar a iniciativa exibindo comportamento orientado por objectivos;
- *Mobilidade* – a capacidade de um agente se movimentar entre os nós de uma rede electrónica;
- *Veracidade* – o pressuposto de que um agente não fornecerá propositadamente informação falsa;
- *Benevolência* – o pressuposto de que os agentes não têm objectivos conflituosos, e que portanto cada agente tentará sempre fazer o que lhe é pedido;
- *Racionalidade* – o pressuposto de que um agente age de forma a atingir os seus objectivos, e não irá agir de tal modo que evite a realização dos mesmos;
- *Persistência* – os agentes possuem continuidade temporal, sendo processos em execução contínua e não uma forma de computação isolada;
- *Aprendizagem* – capacidade de um agente de ao longo da sua “vida” reconhecer padrões de comportamentos e actualizar a sua base de conhecimento para melhorar o seu desempenho futuro;
- *Adaptabilidade* – adaptam-se automaticamente a mudanças no ambiente que “habitam”, degradando a sua eficiência da forma mais graciosa possível em caso de imprevistos e/ou falha de recursos (e.g., conhecimento insuficiente, erros de comunicação).

Os *Agentes Inteligentes* serão talvez os mais directamente ligados à IA e surgem na sequência de várias tendências na disciplina, sendo no entanto de notar que a definição de “inteligência” para o campo da IA e dos agentes é ligeiramente diferente [Wooldridge, 1997]. A IA procura atingir inteligência próxima da humana, enquanto que nos agentes o termo é relaxado para “*tomar atempadamente uma decisão aceitável acerca de uma acção a executar*” [Wooldridge, 1997].

De forma a poder ser útil, um agente contém uma representação (normalmente parcial) do seu universo de discurso (eventualmente, o agente pode ir construindo essa representação). O seu comportamento é função das suas percepções, do seu conhecimento e eventualmente das interacções com outros agentes sendo essencial que o agente possua conhecimento específico no seu domínio de actuação [Coelho, 1995] [Foner, 1997].

4.4.1.2 Sistemas Multiagente

A complexidade crescente dos problemas, que se nos deparam no dia-a-dia, o aumento do poder computacional dos equipamentos disponíveis, e o aparecimento das redes de computadores, levou à criação de *Sistemas Distribuídos* de computação como alternativa aos sistemas de natureza monolítica. Os sistemas distribuídos tornaram-se uma promessa de maior fiabilidade, maior robustez e maior rapidez computacional. Começaram então a aparecer no mercado aplicações cliente/servidor, que distribuíam a carga computacional entre a máquina do utilizador e a máquina servidora de aplicações. Apareceram também aplicações paralelas que dividiam a carga computacional entre vários nós, para assim reduzir o tempo de cálculo.

A comunidade científica da IA apercebeu-se que os fundamentos dos sistemas distribuídos podiam ser vantajosos para o desenvolvimento de sistemas *inteligentes* e, mais importante ainda, com possibilidade de *socialização*. Da junção dos Sistemas Distribuídos com a IA surgiu então a *Inteligência Artificial Distribuída* (IAD). A IAD foi inicialmente definida como uma disciplina que visava a resolução de problemas em que os dados (conhecimento) e a computação (procedimentos) estavam geograficamente separadas, isto em termos lógicos e/ou físicos [Nilsson, 1981]. Davis (1980) considera a IAD apropriada para aquele tipo de problemas onde um único programa de resolução de problemas, uma única máquina ou um único *locus* de computação seja inadequado.

Bond e Gasser (1988) dão uma lista mais extensa de motivações para a utilização das técnicas de Inteligência Artificial Distribuída:

- *Adaptabilidade* – os sistemas de IAD são mais apropriados para lidar com problemas distribuídos em termos espaciais, lógicos, temporais ou semânticos;
- *Custo* – um grande número de pequenas unidades computacionais pode ser mais interessante que uma solução centralizada, principalmente quando os custos com comunicações não são relevantes;
- *Desenvolvimento e gestão* – a inerente modularidade do sistema permite o desenvolvimento separado das diversas partes;
- *Eficiência e velocidade* – a concorrência e distribuição de processos em diferentes computadores pode aumentar a velocidade de computação;
- *Autonomia* – o controlo de processos locais de partes de sistemas isolados ou separados pode ser encarado como uma maneira de protecção ou segurança necessária;
- *Naturalidade* – alguns problemas são melhor descritos em termos distribuídos;

- *Fiabilidade* – os sistemas distribuídos podem exibir um maior grau de fiabilidade/segurança do que sistemas centralizados, pois podem prover, por exemplo, redundância, múltiplas verificações, “triangulação de resultados”;
- *Limitação de recursos* – agentes computacionais individuais ligados a recursos escassos podem cooperar em busca da resolução de problemas complexos;
- *Especialização* – o conhecimento representado e as acções executadas podem ser estritamente relacionadas com o domínio do agente.

As metodologias da IAD deram corpo a duas abordagens à distribuição de sistemas inteligentes: *Resolução Distribuída de Problemas* (RDP); e *Sistemas Multiagente* (SMA). Na RDP existe um conjunto de módulos, ou nós, que cooperam na resolução de um problema específico. O conhecimento acerca do problema bem como a sua solução está dividida pelos nós do sistema. Nos SMA a distinção entre resolução de problemas e cooperação é muito mais distinta, sendo dada especial atenção ao processo de cooperação entre os agentes.

Um Sistema Multiagente pode ser definido como “*uma rede fracamente acoplada de agentes que trabalham em conjunto para solucionar um problema que está para além das suas capacidades individuais*” [Durfée *et al.*, 1989]. Os sistemas baseados em agentes dão corpo a estratégias adequadas para a distribuição não hierárquica de actividades; a solução consiste em criar uma comunidade de Agentes que cooperam entre si na resolução de problemas, o que pressupõe uma certa forma de organização social [Analide e Neves, 1997]. Um *Sistema Multiagente* (SMA) é constituído por uma população de agentes que exibam comportamento social e sejam capazes de interagir de forma cooperativa [Durfée e Rosenschein, 1994], ao mesmo tempo que perseguem os seus objectivos individuais [Ferber, 1993]. Esta interacção pressupõe uma linguagem de comunicação entre agentes (*e.g.*, KQML [Finn *et al.*, 1997]), bem como a percepção do ambiente envolvente.

Os agentes dum SMA podem não ser perfeitamente altruístas, podendo ser egoístas e/ou representar pontos de vistas antagónicos. Por exemplo, o sistema *Persuader* [Sycara, 1990] faz a modelação de negociações laborais através da troca de propostas e contrapropostas, utilizando agentes para representar o(s) sindicato(s), a(s) companhia(s) e um mediador.

Adicionalmente, os agentes que compõem um SMA não têm necessariamente de ser homogéneos. Num SMA podem existir agentes de diversos tipos ou de origens diferentes; *i.e.*, cuja implementação é distinta quer seja do ponto de vista de funcionalidade ou do ponto de vista de desenvolvimento de *software*.

Alguns dos principais problemas no desenvolvimento de sistemas multiagente são a distribuição de funcionalidades (acções) pelos agentes, a cooperação entre os agentes, a

sincronização das acções, o estabelecimento de grupos de agentes apropriados e a especificação de métodos descentralizados para a resolução de conflitos [Hahndel *et al.*, 1994]. Sempre que existam agentes diferentes capazes de executar a mesma acção, é de vital importância a existência de um protocolo de negociação que estabeleça as relações entre contratantes e contratados (*e.g.*, o Protocolo de Rede de Contrato [Smith, 1980] [Davis e Smith, 1983] apresentado na secção 4.4.1.4).

4.4.1.3 Interacção

Uma “componente” fundamental de um sistema multiagente prende-se com a interacção entre as diversas entidades (*i.e.*, os agentes) que constituem o sistema. Os agentes terão de interagir entre si em diversas situações tendo em conta os seus objectivos e garantindo o(s) objectivo(s) e a coerência do sistema. A *coerência* é entendida como uma medida de quão bem um conjunto de entidades se comporta como um “todo” [Sycara, 1989]. Um sistema coerente minimizará ou evitará conflitos e esforços redundantes [Nwana *et al.*, 1996b].

Para atingir um estado coerente, os agentes de um sistema devem relacionar-se na forma:

- *Cooperação* – um processo através do qual um conjunto de entidades desenvolve planos mutuamente aceites e os executa [Valckenaers *et al.*, 1994a]. Estas entidades concordam explicitamente em tentar atingir um (ou vários) objectivo(s) com a contribuição parcial de cada interveniente [Haddadi, 1995]. Cada participante não tem necessariamente o mesmo objectivo que os outros, mas espera beneficiar do processo de cooperação;
- *Coordenação* – é o processo de gerir as interdependências entre actividades [Malone e Crowston, 1994]. Isto é, “*o funcionamento harmonioso das partes para [obter] resultados efectivos*” [Merrian-Webster, URLc];
- *Competição* – é um processo onde várias entidades tentam, independentemente, alcançar um mesmo objectivo (com ou sem o conhecimento dos outros participantes, ou seja, “*lutar consciente ou inconscientemente por um objectivo*” [Merrian-Webster, URLb];
- *Comando ou Controlo* – um processo onde uma entidade governa as acções de outra(s) (limitando assim a autonomia dessas entidades). Segundo Bradshaw (1997b) controlo consiste na definição de *quem* (um agente) faz *o quê* (tarefa/acção), *onde* (localização espacial), *quando* (localização temporal) e *com quem* (outros agentes).

Durante o funcionamento do sistema é natural observar vários destes processos, por exemplo, num SMA que efectue análise de elementos finitos (*e.g.*, previsão climática), cada agente opera

numa pequena fracção dos dados, necessitando de coordenar a sua actividade com os agentes “vizinhos” de forma a fornecer entradas válidas a cada um. O processo global resulta da cooperação de todos os agentes. A Teoria do Institucionalismo Neoliberal [Axelrod, 1984] advoga que a cooperação é melhor que a competição em qualquer cenário, partindo do princípio que o objectivo principal de cada entidade pode ser representado pelo seu ganho individual em termos absolutos e não relativamente a outros agentes. Ou seja, desde que obtenha um ganho positivo um agente não se importa que outro agente obtenha um maior ganho (implicando, para esse, um ganho relativo negativo) [Pontrandolfo e Okogbaa, 1999].

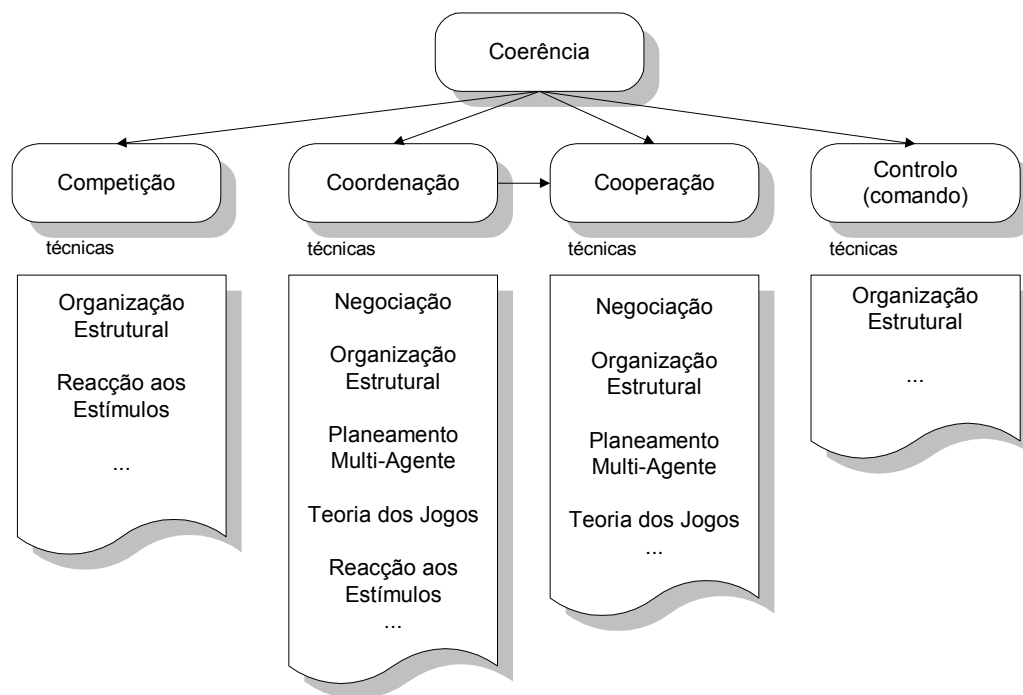


Figura 4.3 – Relação entre comportamentos

A Figura 4.3 (fonte: [Sousa *et al.*, 2000c]) mostra a relação entre os comportamentos dos agentes mencionados anteriormente. Nesta figura a coerência é identificada como o objectivo de mais alto nível (*i.e.*, o resultado de qualquer acção efectuada deve sempre deixar o sistema num estado coerente), e como tal impõe limitações à competição, coordenação e cooperação. A figura também mostra a dependência existente entre coordenação e cooperação: a coordenação pode necessitar de cooperação, mas também pode existir *per se*.

Existem várias técnicas de interacção que dão corpo aos processos descritos anteriormente, sendo apresentadas na Figura 4.3 apenas algumas, que são brevemente explicadas em seguida:

- *Organização estrutural* – explora a organização intrínseca do sistema pois esta define implicitamente as responsabilidades, capacidades, ligações e fluxo de controlo de cada agente [Nwana *et al.*, 1996b] (*e.g.*, cliente/servidor);

- *Reacção a estímulos* – permite a um agente modificar os seus planos e acções de acordo com as suas percepções do ambiente e outros agentes, não havendo comunicação directa;
- *Negociação* – é o esforço efectuado por duas ou mais entidades para alcançar um acordo mutuamente aceite, e que as beneficie [Bussmann e Müller, 1992]. A negociação resulta numa menor dependência entre as entidades que o modelo tradicional de pedido/resposta [Rannanjärvi e Heikkilä, 1998];
- *Planeamento multiagente* – evita acções inconsistentes e conflituosas através da construção de um plano detalhando todas as acções futuras de todos os agentes para atingirem os seus objectivos;
- *Teoria de jogos* – parte do princípio que os agentes são racionais nas suas decisões e perseguem objectivos bem definidos, tomando em linha de conta o seu conhecimento ou previsões acerca das decisões de outros agentes [Osborne e Rubinstein, 1994].

Uma descrição detalhada de aspectos relacionados com coordenação e cooperação pode ser encontrada em [Nwana *et al.*, 1996b]. Como foi dito no final da secção 4.4.1.2, um protocolo de negociação é uma parte importante de um sistema multiagente. Na secção seguinte é apresentado o protocolo mais conhecido e mais frequentemente utilizado.

4.4.1.4 Redes de Contrato

O *Protocolo de Rede de Contrato* (PRC) [Smith, 1980] [Davis e Smith, 1983] é um modelo correntemente utilizado para cooperação em Sistemas Multiagente. Este protocolo baseia-se em dois tipos de agentes (*consumidores* e *fornecedores de serviços*), podendo cada agente desempenhar qualquer um desses papéis em momentos diferentes, e mesmo em simultâneo, para tarefas diferentes.

Essencialmente o protocolo decorre em três fases (Figura 4.4): (i) *pedido*; (ii) *oferta*; e (iii) *contrato*. Na primeira fase, o agente consumidor que necessita de um serviço entra em contacto com todos os agentes fornecedores daquele serviço. Na segunda fase, cada agente fornecedor contactado, vai preparar uma proposta (com custo, janela temporal, etc.) que enviará ao consumidor – nesta fase existe competição entre os agentes fornecedores. Finalmente, na terceira fase, o consumidor selecciona a proposta que melhor satisfaz os seus requisitos e restrições (*e.g.*, financeiras, temporais) e contrata o serviço a esse agente fornecedor.

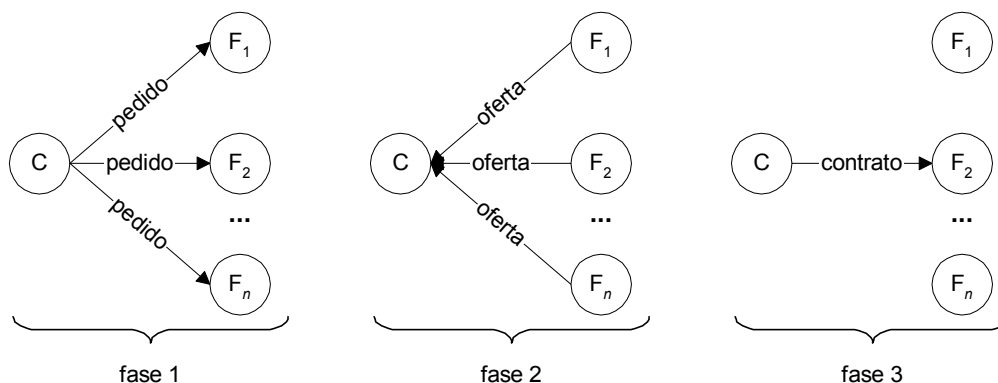


Figura 4.4 – Protocolo de rede de contrato

Este é o algoritmo base do PRC, não contemplando situações onde cada fornecedor prepara zero ou n propostas. Adicionalmente, extensões ao PRC contemplam a troca de propostas e contrapropostas para negociar o melhor acordo entre os agentes [Sycara, 1990]; ou permitem a quebra unilateral dos contratos [Sandholm e Lesser, 1996]; ou ainda o Protocolo de Rede de Contrato com Propagação de Restrições (PRCPR) [Sousa e Ramos, 1999a], onde os vários agentes envolvidos nas subtarefas coordenam as suas acções.

4.4.1.5 Aplicabilidade

Conforme foi apresentado nas secções anteriores, um *Agente* é uma entidade *autónoma*, *inteligente*, capaz de *sentir* e *actuar* sobre o ambiente em que se encontra, perseguindo os seus próprios objectivos; e um *Sistema Multiagente* (SMA) é constituído por uma população de agentes que exibem comportamento *social* e são capazes de interagir de forma *cooperativa*, ao mesmo tempo que perseguem os seus próprios objectivos.

A breve descrição de Agentes e Sistemas Multiagente do parágrafo anterior apresenta muitas semelhanças com o que foi dito sobre Holons e Sistemas Holónicos. Para alguns autores [Bongaerts, 1998] [Bussmann, 1998] [Kirsch *et al.*, 1998] [Sousa *et al.*, 2000c] [Ulieru *et al.*, 2000], os sistemas baseados em agentes são um processo tecnológico que permite a concretização de sistemas holónicos. Outros autores apresentam uma outra visão algo semelhante, onde denominam de holon a agregação de vários agentes, formando assim um *Sistema Multiagente Holónico* [Bürckert *et al.*, 1998] [Fischer, 1999] [Gerber *et al.*, 1999].

Segundo Parunak (1998) os sistemas baseados em agentes são adequados a problemas que exibam características de modularidade, descentralização, dinamismo, subespecificação e complexidade. O autor argumenta que na produção as tarefas de projecto, simulação e escalonamento apresentam muitas destas características, em particular considerando o impacto das tendências recentes, sendo por isso natural utilizar agentes para a solução dos problemas de

produção. Um *Sistema de Produção Baseado em Agentes* (SPBA) [Parunak, 1996] [Parunak, 1998a] [GM, WP] é pois um sistema multiagente para a execução de tarefas relacionadas com os sistemas de produção (*e.g.*, projecto, escalonamento). Os SPBA adoptaram as ideias do CIM, introduzindo-lhe maior flexibilidade através de coordenação ao longo de hierarquias ou heterarquias [Bussmann, 1998].

É notório que os conceitos de HMS e SPBA são muito semelhantes. Ambos apresentam uma visão semelhante sobre o futuro da produção baseada em sistemas distribuídos, com entidades autónomas e cooperativas. A principal diferença entre agentes e holons, é que um holon é uma entidade inerentemente constituída por outros holons e integra seres humanos e maquinaria. Apesar de os SPBA utilizarem estes conceitos até certo ponto, a sua utilização resulta da *praxis* individual de cada projecto e não de uma teoria (semi-)formalmente definida como no caso dos HMS. Até certo ponto é possível afirmar que um holon é um agente com capacidade de agregação temporária e possivelmente intermediário com seres Humanos [Sousa *et al.*, 1999b].

As deficiências dos sistemas de produção actuais foram a motivação para os sistemas holónicos de produção, que como tal, tentaram estabelecer uma visão de como os sistemas de produção devem ser organizados para responder aos requisitos futuros. Por outro lado, os sistemas multiagente são uma tecnologia genérica motivada essencialmente por aspectos de investigação em autonomia, formação de equipas, etc., focando-se no que pode ser feito e como pode ser feito.

Embora semelhantes, os HMS e os SPBA são abordagens complementares com diferentes focos [Bussmann, 1998], sendo que o conceito holónico abarca a estrutura organizacional da produção e a integração de equipamento, controlo e operários, enquanto que os sistemas multiagente se concentram no desenho do processamento de informação e sua implementação.

Em suma, o conceito holónico oferece a visão e a estrutura organizacional ao passo que o conceito de agência oferece a tecnologia que possibilita a implementação de tais sistemas.

4.4.2 Programação em Lógica Estendida

Nesta secção introduz-se o tema da Programação em Lógica Estendida e a sua viabilidade para o desenvolvimento de agentes e representação do conhecimento de cada agente. De um ponto de vista de representação é dada especial ênfase à problemática da informação incompleta (*i.e.*, de conhecimento parcial) identificada como uma das características dos sistemas de produção do futuro.

4.4.2.1 Introdução

De um ponto de vista computacional, a ambiguidade semântica, juntamente com a complexidade gramatical e léxica das linguagens naturais (*e.g.*, Português), tornam difícil a sua utilização para a descrição formal de conhecimento.

Para a representação de conhecimento em sistemas computacionais recorre-se não menos vezes à utilização de linguagens lógicas [Coelho, 1995]. De uma maneira geral uma *Linguagem Lógica* é composta por [Wooldridge, 1992] [Russell e Norvig, 1995]:

- uma *sintaxe*, que define os objectos de discurso;
- uma *semântica*, que atribui um significado a cada objecto sintáctico; e
- uma *teoria de prova*, que define as manipulações sobre os objectos sintácticos.

Das várias linguagens lógicas existentes destacam-se a *Lógica Proposicional* e a *Lógica de Predicados de 1ª Ordem*, como mecanismos de representação de conhecimento acerca do mundo real. As linguagens lógicas permitem uma descrição formal e não ambígua de factos, que podem ser verificados e validados formalmente [Ligeza, 1997] [Pereira, 1997] [Santos *et al.*, 1999]. A lógica proposicional tem um poder de expressão relativamente simples, permitindo apenas trabalhar com factos do mundo real. Por outro lado, a lógica de predicados de 1ª ordem já permite trabalhar com factos e relações entre objectos do mundo real [Coelho, 1995].

A *Lógica Proposicional* formaliza a estrutura lógica mais elementar do discurso matemático, sendo uma linguagem que permite o tratamento de expressões lógicas simples sobre a sua forma abstracta [Machado e Neves, 1992]. O alfabeto \mathcal{L} de uma linguagem proposicional consiste num conjunto de operadores (\neg , \equiv , \rightarrow , \wedge , \vee), os parêntesis e um conjunto de símbolos proposicionais (*e.g.*, A, B, ...) que representam factos do mundo real. As fórmulas, ou proposições, assumem um dos dois valores de verdade: *verdadeiro* ou *falso*. A lógica proposicional é extremamente simples apresentando algumas limitações, principalmente de um ponto de vista ontológico, representando o mundo real através de *factos* que podem ser *verdadeiros* ou *falsos*. Uma outra linguagem lógica de mais poder expressivo é a *Lógica de Predicados*, que permite representar o mundo através de *objectos* (que possuem *propriedades*), *relações* entre objectos e *funções* que manipulam esses mesmos objectos.

A *Lógica Predicativa* ou *Cálculo dos Predicados* pode ser caracterizada como uma sistema formal apropriado à definição de teorias do universo de discurso da Matemática. A lógica predicativa deve ser vista como uma extensão da lógica proposicional [Machado e Neves, 1992], onde são introduzidos três novos conceitos lógicos: (i) funções; (ii) predicados; e (iii) quantificadores. Em relação à lógica proposicional, a lógica predicativa apresenta a possibilidade

de trabalhar com variáveis, e não apenas com constantes, o que permite não ter de enumerar todos os elementos do universo de discurso.

Para deduzir novos factos numa linguagem lógica é necessário aplicar *regras de inferência*. O conceito de *derivabilidade* deve ser entendido como “pode deduzir-se que”, ou seja, partindo de um conjunto de fórmulas e aplicando uma ou mais regras, deduz-se uma nova fórmula.

$$H \vdash_R T \quad (4.1)$$

A expressão (4.1) indica que T (teorema) pode ser deduzido de H (conjunto das hipóteses) aplicando uma sequência de regras de R (regras de inferência).

O paradigma da *Programação em Lógica* (PL) baseia-se na utilização da lógica como uma linguagem de programação. A PL é feita segundo uma abordagem descritiva na qual um programa consiste na descrição de um problema em termos de axiomas lógicos, em contraste com a abordagem imperativa das linguagens tradicionais (*e.g.*, C, Pascal) na qual um programa descreve uma sequência de passos a ser executada. De facto, ao invés de colocar o foco de interesse nos procedimentos e em que procedimentos desenvolver, os predicados lógicos especificam um relacionamento possível entre objectos computacionais. Na programação em lógica a execução de um programa corresponde à dedução controlada de consequentes com base numa teoria lógica, normalmente expressa como um conjunto de cláusulas de Horn²⁰. Esta dedução é efectuada pelo mecanismo de inferência intrínseco da linguagem de PL (*e.g.*, PROLOG [Clocksin e Mellish, 1981] [Coelho e Cotta, 1988]).

A PL em PROLOG não permite a representação de literais negados, ou seja, uma questão Q colocada a uma base de conhecimento KB pode ser provada ou não, sendo consequentemente *verdadeira* ou *falsa* (4.2):

$$((KB \vdash Q) \rightarrow \text{verdadeiro}) \vee ((KB \not\vdash Q) \rightarrow \text{falso}) \quad (4.2)$$

Isto é, caso exista prova de Q em KB então Q é *verdadeiro*, caso não exista prova de Q na base de conhecimento, Q é considerado *falso*. Este facto é denominado *Pressuposto do Mundo Fechado* (PMF) e permite simplificar o cálculo lógico computacional.

Um outro pressuposto da PL é o do *Domínio Fechado*, onde se considera que todos os objectos no universo de discurso estão representados na base de conhecimento. Estes pressupostos, se por um lado simplificam o desenvolvimento de sistemas inteligentes, por outro

²⁰ Uma *cláusula* é uma classe especial de fórmulas, composta exclusivamente por disjunção de literais, em que um literal é uma fórmula atómica ou a sua negação (*e.g.*, $A \vee B \vee \neg C \vee \neg D \vee \neg E$). Expressões lógicas deste tipo podem, sem perda de equivalência, ser reescritas em formato “se conjunção de literais então disjunção de literais” (*e.g.*, $C \wedge D \wedge E \rightarrow A \vee B$). As *Cláusulas de Horn* são um tipo especial de cláusulas onde apenas existe um literal positivo (*i.e.*, $Q \wedge R \wedge S \wedge T \rightarrow V$).

limitam o campo de aplicação, pois na maioria dos casos o conhecimento total do domínio é impossível de obter.

Na secção seguinte será apresentada uma extensão à programação em lógica que permite a representação de conhecimento negativo, contornando assim o PMF, com o objectivo de permitir uma melhor representação da realidade nas bases de conhecimento de entidades computacionais.

4.4.2.2 Extensões à Programação em Lógica

É com alguma frequência que se depara com cenários de Informação Incompleta nas Bases de Conhecimento (BC) e nos processos de negociação [Neves, 1984] [Traylor e Gelfond, 1993] [Neves *et al.*, 1997b]. A programação em lógica clássica é baseada nalguns pressupostos (*e.g.*, os pressupostos do Mundo Fechado e do Domínio Fechado) que impõem limitações ao tipo de processamento necessário para representar e manipular informação incompleta. Os sistemas reais podem beneficiar largamente de abordagens que evitem estas limitações [Sousa *et al.*, 2000b]. Ao adicionar capacidade para representação e raciocínio sobre informação incompleta a um sistema, a sua base de conhecimento passa a poder descrever o mundo real de forma muito mais realística.

Um *Programa em Lógica Estendida* é um conjunto possivelmente infinito de fórmulas segundo o modelo (4.3), onde cada um dos literais L_i é uma fórmula atómica de primeira ordem ou a sua negação explícita (*i.e.*, A ou $\neg A$) e ‘*não*’ representa a negação por falha [Brewka, 1996] [Schroeder *et al.*, 1997] [Li e Moniz Pereira, 1997].

$$L_0 \leftarrow L_1 \wedge \dots \wedge L_m \wedge \text{não } L_{m+1} \wedge \dots \wedge \text{não } L_n \quad (0 \leq m \leq n) \quad (4.3)$$

Informalmente, ‘*não L*’ lê-se como “*não há prova de que L é o caso*”, enquanto que ‘ $\neg L$ ’ é interpretado como “*L definitivamente não é o caso*”.

A diferença entre a negação fraca (operador ‘*não*’) e a negação forte (operador ‘ \neg ’), compreende-se melhor com um exemplo [Analide e Neves, 1996]:

$$\text{atravessar} \leftarrow \text{não comboio} \quad (4.4)$$

$$\text{atravessar} \leftarrow \neg \text{comboio} \quad (4.5)$$

O conhecimento representado na equação (4.4) significa que se atravessa se não existir uma prova de que há um comboio, enquanto que em (4.5) atravessa-se se existir uma prova que não há um comboio a aproximar-se.

Os dois operadores podem ser utilizados em conjunto como se mostra em seguida, que se traduz por “*não atravessar se não há certeza de que não há um comboio a aproximar-se*”.

$$\neg \text{atravessar} \leftarrow \text{não } \neg \text{comboio} \quad (4.6)$$

A linguagem proposta é de natureza lógica; *i.e.*, contém uma conectiva unidireccional com duas posições para formação de regras com base em literais, que abrange dois tipos de negação: *fraca* (também chamada *negação por falha*) e *forte* (também chamada *negação explícita*).

Por outras palavras, em vez de utilizar o pressuposto do mundo fechado (*i.e.*, tudo o que não é conhecido é falso), o conhecimento de algo falso tem de ser explicitamente representado na Base de Conhecimento (BC). Assim sendo, a BC tem duas partes: o que é sabido como *verdadeiro*, e o que é sabido como *falso*. Tudo o resto é *desconhecido* [Sousa *et al.*, 2000b]. Ou seja, se \mathcal{R} for o conjunto de respostas de um programa em lógica, $\neg q$ é verdadeiro se $\neg q \in \mathcal{R}$ e *não* q é verdadeiro se $q \notin \mathcal{R}$.

Há no entanto casos onde o Pressuposto do Mundo Fechado (PMF) faz sentido existir; por exemplo, um sistema de gestão de encomendas pode assumir que uma encomenda não representada no sistema, de facto não existe (*i.e.*, é *falsa* e não *desconhecida*). De uma forma geral, o fecho de um axioma P é representado por:

$$\neg P \leftarrow \text{não } P \quad (4.7)$$

o que se traduz em linguagem corrente por “ P é falso se não for possível prova-lo”; *i.e.*, negação por falha da Programação em Lógica.

A representação de informação negativa explícita, contornando o PMF, permite distinguir entre o que se sabe, o que não se sabe e o que se desconhece. No entanto, existem outras situações interessantes a representar na base de conhecimento de forma a melhor aproximar essa base de conhecimento da realidade, nomeadamente no que toca a informação incompleta.

Seguindo o trabalho de Traylor e Gelfond (1993) as extensões à programação em lógica para representação de informação incompleta em bases de conhecimento de agentes serão de quatro tipos [Analide e Neves, 1996] [Neves *et al.*, 1997b] [Sousa *et al.*, 2000b]:

- *Informação negativa* – pretende-se representar informação negativa de forma explícita e não por defeito como acontece na Programação em Lógica via PMF;
- *Nulos desconhecidos* – esta extensão permite representar situações onde nem toda a informação é conhecida; *i.e.*, nem todos os argumentos de um predicado são conhecidos;
- *Informação mutuamente exclusiva* – Esta extensão permite representar informação para a qual não se tem a certeza, mas sabe-se o conjunto finito de opções mutuamente exclusivas; *i.e.*, implementa o operador lógico “ou exclusivo”, \oplus ;

- *Informação não permitida* – este género de nulos denota situações não admitidas na base de conhecimento. Obviamente, para evitar incoerência na base de conhecimento, qualquer asserção deve ser primeiramente validada.

O interpretador para esta linguagem lógica é então dado por uma função que recebendo um programa em PLE escrito numa linguagem lógica \mathcal{L} e uma questão Q escrita na linguagem lógica \mathcal{L} , devolva um dos três valores lógicos possíveis – *vide* (4.8)

$$demo : KB \subseteq \mathcal{L} \times Q \in \mathcal{L} \mapsto \{\text{verdadeiro}, \text{falso}, \text{desconhecido}\} \quad (4.8)$$

A Figura 4.5 descreve o meta-interpretador desta lógica estendida, sendo que ‘ \neg ’ representa negação explícita e ‘ \perp ’ representa a cláusula vazia [Sousa *et al.*, 2000b].

```

demo( $KB \cup KB\epsilon \cup KB\pi \cup \mathcal{D}, Q$ )  $\stackrel{\text{def}}{=}$ 
  seja  $KB$  uma base de conhecimento escrita na linguagem lógica  $\mathcal{L}$ 
  seja  $KB\epsilon$  a base de conhecimento mutuamente exclusivo
  seja  $KB\pi$  a base de conhecimento sobre situações proibidas
  seja  $\mathcal{D}$  o conjunto de identificadores de nulos desconhecidos
  seja  $Q$  uma questão sobre um predicado  $q(a_0, \dots, a_n)$ 
  seja  $a_i$  um dos argumentos da questão  $Q$  (um átomo ou uma variável)
  início
    se  $KB \vdash Q \wedge (\forall a_i) a_i \notin \mathcal{D}$  então  $demo \mapsto \text{verdadeiro}$ 
    se  $KB \vdash \neg Q$  então  $demo \mapsto \text{falso}$ 
    se  $KB \not\vdash Q \wedge KB \not\vdash \neg Q$  então  $demo \mapsto \text{desconhecido}$ 
    se  $KB \vdash Q \wedge (\exists a_i) a_i \in \mathcal{D}$  então  $demo \mapsto \text{desconhecido}$ 
    se  $KB\epsilon \vdash Q$  então  $demo \mapsto \text{desconhecido}$ 
    se  $KB\pi \vdash Q$  então  $demo \mapsto \perp$ 
  fim
    
```

Figura 4.5 – Algoritmo do meta-interpretador para lógica estendida

O algoritmo da Figura 4.5 traduz-se informalmente em “dada uma base de conhecimento e uma interrogação Q a essa base de conhecimento, tem-se que:

- Q é *verdadeiro*, se for possível encontrar uma prova de Q e nenhum dos argumentos de Q instanciar com um dos nulos desconhecidos;
- Q é *falso*, se for possível provar $\neg Q$, ou seja, prova-se que Q é falso;
- Q é *desconhecido*:
 - se não for possível provar Q nem $\neg Q$; ou
 - se for possível provar Q mas pelo menos um dos argumentos de Q instancia com um dos nulos desconhecidos; ou então

- se Q for um nulo mutuamente exclusivo;
- Q é *insatisfazível*, se não respeita os invariantes não permitidos.”

Este algoritmo permite assim determinar o valor de verdade de qualquer programa em lógica estendida utilizado como base de conhecimento de um agente, permitindo a representação e manuseamento de conhecimento incompleto.

4.4.2.3 Aplicabilidade

A Lógica pode ser usada com sucesso como uma linguagem formal para a especificação de agentes [Wooldridge, 1992] [Lomuscio e Colombetti, 1997] [Schroeder *et al.*, 1997] [Santos, 1999] pois é uma linguagem expressiva e não ambígua que permite a especificação dos comportamentos dos agentes, a representação de características (tais como crenças, desejos, e intenções), permitindo, além disso, o estudo, formalização e raciocínio acerca das propriedades dos agentes. Adicionalmente, a utilização de uma linguagem formal permite a verificação e validação automática do conhecimento [Ligeza, 1997] [Pereira, 1997] [Santos *et al.*, 1999], evitando inconsistências e redundâncias.

Noções como *inferência*, *dedução* e *processamento simbólico* da programação em lógica facilitam a criação de sistemas racionais que manuseiam itens de conhecimento; *i.e.*, os agentes. Os *Agentes Baseados em Lógica* utilizam uma linguagem lógica como forma de especificação e uma forma de programação em lógica (estendida) para a sua implementação, representando o seu conhecimento como um conjunto de axiomas lógicos (KB) sobre o qual se deduzem teoremas (T), eventualmente gerando novos axiomas ($KB' = KB \cup T$).

A programação em lógica utiliza o PMF atribuindo um valor de *falso* a todos as questões que não possam ser provadas à luz da base de conhecimento do agente. No entanto, em termos práticos, é mais frequente conhecer-se a verdade ou falsidade de um número limitado de factos, e desconhecer-se por completo tudo o resto [Lamma *et al.*, 1998]. Ora um agente, escolhe as suas acções com base no conhecimento que possui, e saber que uma determinada acção produz um efeito negativo é diferente de não saber o resultado dessa acção. Isto permite ao agente não escolher as acções com efeitos negativos e tentar acções com efeitos desconhecidos quando não tem outra alternativa de forma a aumentar o seu conhecimento.

A utilização da programação em lógica estendida com o meta-interpretador apresentado na Figura 4.5 permite a representação e manipulação de nulos, nomeadamente, *nulos desconhecidos*, *nulos mutuamente exclusivos* e *nulos não permitidos*, na base de conhecimento de um agente. Por exemplo, um agente pode desconhecer quais as capacidades (*i.e.*, conjunto de acções que esse agente pode executar) de um outro agente que habita o mesmo ambiente; ou então desconhecer

qual de dois agentes específicos consegue executar determinada acção; o agente pode também ter presente na sua base de conhecimento um invariante que represente o facto de não ser permitido ao agente ter objectivos que entrem em conflito.

4.5 Estado da Arte

Esta secção apresenta alguns trabalhos relacionados com o tema/trabalho desta tese, no que toca a Sistemas Holónicos de Produção (HMS) e Sistemas de Produção Baseados em Agentes (SPBA). Conforme foi referido os SPBA são, em princípio, baseados em conceitos muito próximos aos que regem os HMS, existindo, por conseguinte, um número significativo (e representativo) de trabalhos que não devem ser ignorados.

Dos trabalhos que se apresentam a seguir será dada especial ênfase aos trabalhos de escalonamento de tarefas em ambientes industriais, que tem sido a escolha de eleição para apresentação de conceitos relacionados com SPBA e HMS. A área de escalonamento apresenta vários pontos de interesse na investigação de sistemas distribuídos inteligentes, já que o problema de escalonamento é um problema complexo e inerentemente distribuído (*i.e.*, várias tarefas, várias máquinas).

4.5.1 Trabalhos de Referência

Nesta primeira subsecção são apresentados com algum detalhe os trabalhos considerados mais representativos e mais directamente relacionados com esta tese. Na subsecção 4.5.2 serão sumariamente apresentados outros trabalhos.

4.5.1.1 AARIA

O sistema AARIA [Parunak *et al.*, 1997] foi construído com a visão de demonstrar que ligando um grupo de agentes representando “habilidades” de produção é possível criar uma empresa de produção cujo desempenho e funcionalidades suplantem a dos sistema actuais. O sistema tem funcionalidade ERP (Planeamento de Recursos Empresariais) e MES (Sistema de Execução da Produção); *e.g.*, recepção de encomendas, compras, gestão de existências, gestão de recursos, gestão de pessoal, contabilidade, escalonamento de capacidade finita e simulação.

Existem no sistema agentes separados (com igual grau de inteligência e responsabilidade) que representam *Partes*, *Recursos*, e *Processos Unitários* (operações) (Figura 4.6, fonte: [Parunak *et al.*, 1997]), não havendo controlo centralizado. As partes movem-se de *Processo unitário* (PU) em *Processo unitário* através de *Buffers*. Cada PU tem como entrada um ou mais componentes

(vindos de um ou mais *buffers*) e produz como saída, um ou mais componentes (colocando-os num *buffer*), usando para tal os recursos seleccionados. Cada agente efectua escalonamento local, e a política de escalonamento é baseada em janelas temporais dinâmicas e não em instantes fixos; *i.e.*, é estabelecido um acordo entre cliente e fornecedor com base numa função custo/data-de-entrega [Sauter e Parunak, 1999].

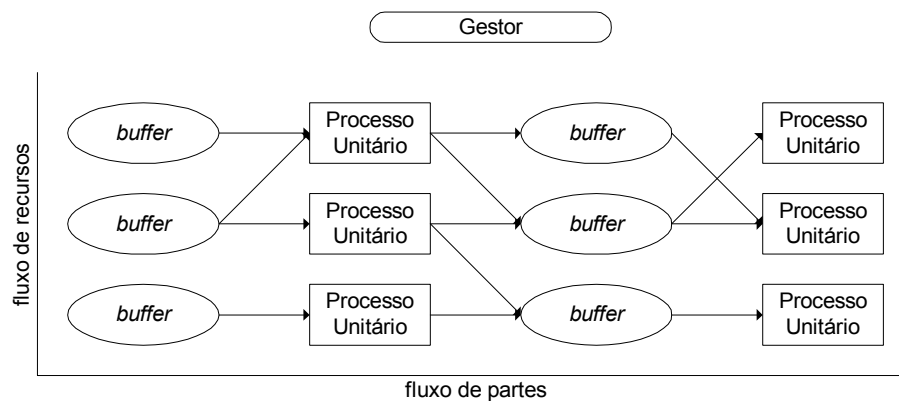


Figura 4.6 – Arquitectura do sistema AARIA

O sistema inicial foi mais tarde estendido por forma a cobrir a cadeia de fornecimento ligando clientes, fornecedores e (eventualmente) instalações fabris distintas [Baker *et al.*, 1997] [Sauter e Parunak, 1999].

4.5.1.2 HMS “Testbed”

A Universidade Católica de Leuven, Bélgica, responsável pelo caso de teste n.º 5 do projecto IMS, desenvolveu um protótipo de uma estação de montagem holónica para o estudo de viabilidade do conceito HMS [Valckenaers *et al.*, 1994a] [Valckenaers *et al.*, 1994b] [Bongaerts *et al.*, 1995]. Os recursos físicos do sistema consistem em várias estações de trabalho para maquinaria e montagem, bem como um sistema de transporte (cada um representado por um holon). Além deste holons (*hardware* + *software*) existem outros três holons (apenas de *software*) para escalonamento, planeamento e controlo.

Neste trabalho o escalonamento não era elaborado por vários agentes mas sim por um só, sendo o objectivo do trabalho analisar a cooperação entre o elemento de controlo e o de escalonamento. O holon de controlo assumia a escala de tarefas enviada pelo holon de escalonamento como uma proposta e tentava cumpri-la enquanto fosse possível. Em situações não previstas o holon de controlo decidia autonomamente o que fazer e enviava um pedido para o holon de escalonamento gerar uma nova escala.

4.5.1.3 PROSA

O trabalho de investigação em Sistemas Holónicos de Produção difere, principalmente no que respeita às entidades escolhidas como “blocos de construção”. Bongaerts *et al.* (1996) identificaram três tipos de holons necessários e suficientes, pois, “*para uma implementação minimalista de um sistema de produção é suficiente ter uma holarquia constituída por holons de produtos, recursos e ordens*”.

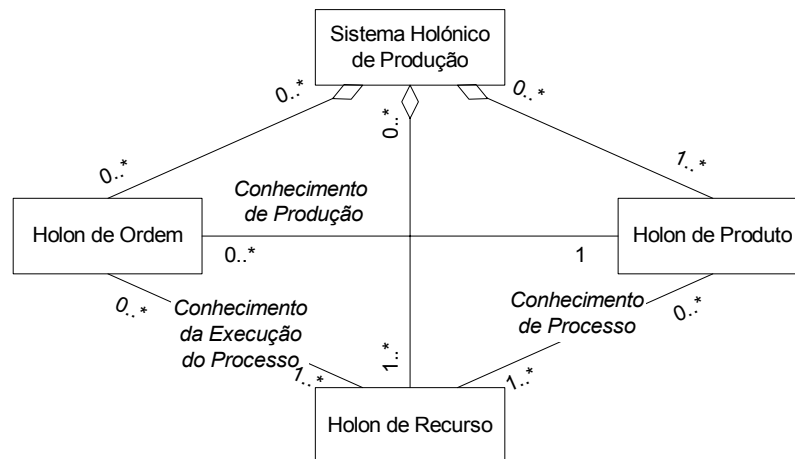


Figura 4.7 – Arquitectura de referência PROSA

Desta afirmação surgiu a taxonomia PROSA [van Brussel *et al.*, 1998] [Wyns, 1999] para a implementação de HMS. A taxonomia PROSA define quatro arquétipos de holons (Figura 4.7, fonte: [van Brussel *et al.*, 1998]):

- *Produtos* – encapsulam o Modelo do Produto; *i.e.*, informação de projecto e processo de um tipo de produtos;
- *Recursos* – encapsulam os recursos físicos da instalação fabril e contêm a componente de controlo do equipamento;
- *Ordens* – encapsulam as tarefas a executar no sistema, fornecem funcionalidade de escalonamento;
- *Auxiliares* – encapsulam funcionalidades para auxiliar os outros holons funcionando normalmente como conselheiros (*e.g.*, holon que implementa um algoritmo óptimo de escalonamento centralizado).

PROSA não é um sistema, mas sim uma arquitectura de referência para a implementação de HMS. Esta arquitectura de referência foi usada em [Bongaerts, 1998] e HoMUCS [Langer, 1999] para a implementação de sistemas específicos.

4.5.1.4 Gou & Luh

A arquitectura holónica de Gou e Luh (1997) é bastante modular (Figura 4.8, fonte: [Gou e Luh, 1997]) sendo composta por oito tipos de holons: (i) *Produto*; (ii) *Parte*; (iii) *Tipo-de-Máquina*; (iv) *Máquina*; (v) *Coordenador de Célula*; (vi) *Célula*; (vii) *Coordenador de Fábrica*; e (viii) *Fábrica*.

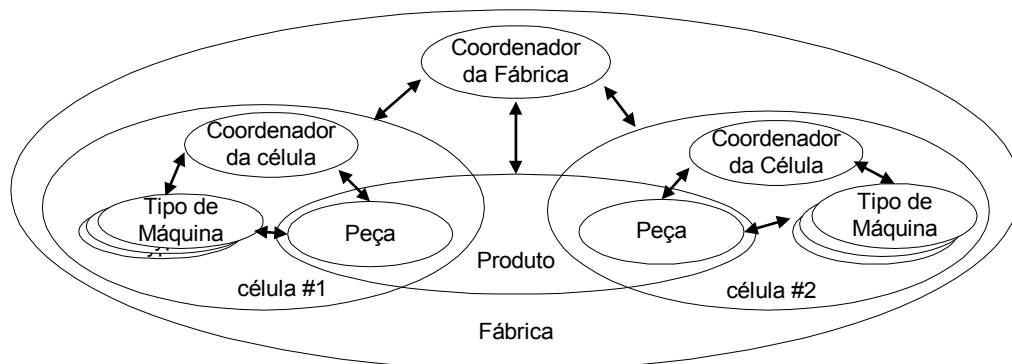


Figura 4.8 – Arquitectura holónica de Gou & Luh

Os holons *Produto* correspondem a holarquias representando as ordens de fabrico, sendo composto por holons *Parte*, que representam um estágio do processo de fabrico. Um holon *Célula* é uma holarquia composta por *Coordenador de Célula*, *Máquina*, *Tipo-de-Máquina* e *Parte*. Uma *Fábrica* é composta por múltiplas Células cada uma com múltiplas Máquinas. Os Produtos devem movimentar-se pelas Máquinas (eventualmente em Células diferentes).

O escalonamento assume capacidade infinita e é efectuado através da decomposição do problema em subproblemas, que são resolvidos usando o método de relaxação Lagrangiana. Ao nível das Células, os holons *Parte* trocam mensagens com os holons *Tipo-de-Máquina* para calcular o custo de utilização da máquina e determinar o início da operação. Ao nível da Fábrica, os holons *Célula* cooperam com os holons *Produto* para determinar as rotas de fabrico.

4.5.2 Outros Trabalhos

Nesta secção são resumidamente apresentados mais alguns trabalhos relacionados com o tema da tese. Para obter informação sobre um número mais extenso de trabalhos consultar, por exemplo, [Shen e Norrie, 1999], [Sousa *et al.*, 2000c] e [Weng e Ren, URL]. Um grande número de trabalhos pode também ser encontrado nas actas de conferências especializadas em Produção Inteligente (e.g., [Albayrak e Bussmann, 1996a] [Monostori, 1997b] [IMS98, 1998] [Gini e Boddy, 1998] [van Brussel e Valckenaers, 1999] [Kopacek, 1999]).

Os HMS e os SPBA têm sido utilizados, em primeiro lugar, nas seguintes áreas:

- Planeamento de produção, Escalonamento e Controlo;
- Planeamento do processo;
- Projecto;
- Integração Empresarial;
- Gestão da cadeia de fornecimento e Empresa Virtual.

No **Planeamento da Produção** os agentes/holons elaboram um plano do que deve ser produzido (*e.g.*, na indústria automóvel em que cada agente representa uma unidade de produção na instalação fabril [Baumgartel *et al.*, 1997], ou uma decomposição de tarefas por departamentos para a produção de televisores [Hazdra e Pechoucek, 1999]). No **Escalonamento e Controlo**, os agentes/holons elaboram o escalonamento de tarefas e controlam a sua execução. O trabalho seminal de Parunak (1987), YAMS, utiliza uma hierarquia fixa de agentes (Fábrica, Célula, Máquina) que representam os recursos e negociam as ordens de fabrico aplicando o *Protocolo de Rede de Contrato* (PRC). Outra abordagem consiste na utilização de vários agentes “peritos” em escalonamento (decomposição por função), (*e.g.*, CORTES [Sycara *et al.*, 1991], MICROBOSS [Sadeh, 1994]), onde os agentes normalmente representam diferentes perspectivas de escalonamento e utilizam memórias partilhadas. A utilização do PRC é comum nos trabalhos de escalonamento, no entanto, o mesmo já não se pode dizer quanto aos agentes/holons existentes no sistema. Vários trabalhos seguem a filosofia de YAMS usando agentes representando os recursos (não necessariamente numa hierarquia) e eventualmente agentes/holons auxiliares (por exemplo, [Maturana e Norrie, 1996], [Kouiss *et al.*, 1997], [Kádár *et al.*, 1998], [Váncza e Márcus, 1998] e [Heikkilä *et al.*, 1999]). Outros recorrem à modelação de Recursos e Ordens de Fabrico ou Partes (*e.g.*, [Maley, 1988], [Duffie *et al.*, 1988] e FLAVORS [Morley e Schelberg, 1993]), ou de Recursos e Produtos (*e.g.*, IFCF [Lin e Solberg, 1992] e [Kanchanasevee *et al.*, 1997]). Outros ainda, além dos Recursos e das Ordens de Fabrico, modelam os Produtos (*e.g.*, MASCADA [Kollingbaum *et al.*, 2000] e HoMUCS [Langer, 1999]). Uma arquitectura diferente é usada no sistema LMS [Fordyce *et al.*, 1992] que tem agentes representando objectivos. Exemplos de trabalhos referentes ao controlo em indústrias de processo usando uma abordagem holónica são [Agré *et al.*, 1994] e [McFarlane *et al.*, 1995].

No **Planeamento do Processo** os agentes/holons são normalmente utilizados para representar conhecimento pericial numa determinada função (*e.g.*, MAPP [Hayes, 1998]). No **Projecto** os agentes/holons ajudam no processo de desenvolvimento de um novo produto, por exemplo, cada agente é responsável por uma “característica” do desenho (*e.g.*, RAPPID [Parunak *et al.*, 1999b]), ou pela integração de vários projectistas (*e.g.*, CONCENSUS [Cooper e Taleb-Bendiab, 1997]). A

Integração Empresarial corresponde a sistemas de partilha e integração de dados para as várias funções da empresa e onde cada parte tem conhecimento do seu lugar no todo da empresa sendo capaz de avaliar o impacto das suas decisões nesta (e.g., ABCDE [Balasubramanian e Norrie, 1995] e MetaMorph [Shen e Norrie, 1998]). A **Gestão da Cadeia de Fornecimento e as Empresas Virtuais** correspondem à integração de várias empresas, produtores, revendedores, distribuidores, etc. (e.g., HOLOS [Rabelo e Camarinha-Matos, 1994], [Fischer *et al.*, 1996], DASch [Parunak, 1998b], XCITIC [Azevedo e Sousa, 1997] e [Camarinha-Matos *et al.*, 1997]).

4.5.3 Sinopse

Nesta secção é elaborada uma sinopse (Tabela 4.2) dos trabalhos de referência apresentados na secção 4.5.1, tendo em conta as questões de investigação e os objectivos enumerados no Capítulo 1 para este trabalho.

Tabela 4.2 – Sinopse dos trabalhos relacionados

	Âmbito	Entidades modeladas	Socialização	Informação Incompleta
AARIA	Escalonamento e gestão da cadeia de fornecimento	Partes, Recursos e Processos Unitários	Protocolo de rede de contrato	n.a.
HMS “testbed”	Escalonamento e Controlo	Recursos, Escalonador	Holarquia; pedido-resposta	n.a.
Gou & Luh	Escalonamento	Máquinas, Células, Peças, Produtos, Coordenadores	Holarquia; propagação de resultados parciais	n.a.
PROSA	Integração empresarial (arquitectura de referência)	Produtos, Recursos, Ordens, Auxiliares	Holarquia; favorece a troca directa de mensagens (pedido-resposta)	n.a.

Os dois primeiros factores (âmbito do trabalho e identificação de entidades) relacionam-se com as arquitecturas de cada trabalho (objectivo 3). O terceiro factor prende-se com os mecanismos de interacção entre as entidades constituintes da arquitectura (objectivo 6). Finalmente, o quarto factor, tratamento de informação incompleta, prende-se com o objectivo 5.

Pode-se constatar que o escalonamento é uma (senão a única) das actividades de fabrico coberta pelos trabalhos, havendo no entanto dois trabalhos (AARIA e PROSA) que abarcam um pouco mais da globalidade da actividade empresarial.

A selecção de entidades segue normalmente uma decomposição por entidades (físicas ou lógicas) e não por funcionalidade. No entanto, PROSA favorece a existência de holons estritamente funcionais (*e.g.*, escalonamento) enquanto que, por exemplo, AARIA descentraliza a função escalonamento, sendo o escalonamento gerado de forma emergente através da negociação entre agentes. Todos os trabalhos modelam os recursos físicos e encaixam parcialmente na taxonomia PROSA.

A interacção e socialização dos vários agentes/holons é efectuada através de negociação no caso AARIA, enquanto que os outros trabalhos defendem a criação de holarquias e a troca directa de mensagens entre holons normalmente com uma filosofia pedido-resposta, nada sendo especificado quanto à possível negociação.

Nenhum dos trabalhos analisados aborda a questão da informação incompleta na base de conhecimento das suas entidades constituintes.

Analisando os pontos fracos e fortes de cada um dos trabalhos considerados de referência, pretende-se então uma arquitectura com um âmbito abrangente semelhante a PROSA, favorecendo no entanto uma abordagem descentralizada à semelhança de AARIA, ou seja, os elementos da arquitectura devem modelar entidades e não funções. Quanto à interacção entre entidades, entende-se que um mecanismo de negociação é mais favorável que um mecanismo de pedido-resposta (tipo cliente/servidor); aliás por esse motivo, o Protocolo de Rede de Contrato (PRC) é tão utilizado nos SPBA. No entanto, considera-se importante que seja levada em consideração a natureza específica da produção, nomeadamente no que toca ao escalonamento, enfatizando a cooperação entre os vários fornecedores de serviços ao contrário do PRC que evidencia a competição. Adicionalmente, entende-se que a inexistência de tratamento de informação incompleta é uma grave lacuna nos trabalhos existentes, devendo por isso ser colmatada.

Nos próximos capítulos será apresentada a arquitectura proposta (identificando as entidades modeladas), o mecanismo de negociação utilizado por essas entidades bem como o funcionamento dessas entidades e o protótipo desenvolvido. No final do Capítulo 6 será feita uma comparação do trabalho realizado com estes trabalhos de referência.

4.6 Resumo do Capítulo

No Capítulo 3 foram apresentadas as características do novo ambiente socioeconómico e as suas implicações para o sector produtivo e na secção 3.4.3 apresentou-se uma lista das características desejáveis nos sistemas de produção para responderem aos requisitos actuais e futuros da sociedade e economia.

No presente capítulo apresentou-se uma abordagem, que se pretende inovadora, aos sistemas de produção, os *Sistemas Holónicos de Produção* (HMS), cujas origens filosóficas tentam mimetizar a evolução e organização dos sistemas biológicos e de organização social. Assim, começou-se por descrever as origens dos sistemas holónicos e as suas características, com realce para a *distribuição, autonomia e cooperação*. Em seguida, foram descritas as origens dos sistemas holónicos de produção no seio da comunidade internacional de investigadores, bem como as suas características. Em seguida foi estabelecido um paralelo entre este conceito e as propriedades enunciadas na secção 3.4.3.

Os HMS não são a única abordagem inovadora conhecida aos sistemas de produção e, por isso, foram também sumariamente descritas outras abordagens, nomeadamente, a *Fábrica Fractal* e a *Produção Biónica*. Estas abordagens foram comparadas com os HMS e com o conceito de Produção Integrada por Computador. Na secção seguinte foram apresentadas duas soluções para a concretização de sistemas dando corpo ao conceito holónico: os *Sistemas Baseados em Agentes* e a *Programação em Lógica Estendida*. Assim, começou-se por apresentar o conceito de *agente* e as suas características, sendo em seguida apresentado o conceito de *sistema multiagente*. Na secção seguinte apresentou-se a lógica como linguagem formal para especificação de agentes e a programação em lógica estendida (*i.e.*, com representação de negação explícita) para a codificação desses agentes.

Finalmente, foi apresentado o estado da arte na área dos Sistemas Holónicos de Produção e dos Sistemas de Produção Baseados em Agentes, que tal como foi referido na secção 4.4.1.3 constituem um paradigma muito semelhante aos HMS. Dos trabalhos escolhidos como referência foi elaborada uma sinopse para evidenciar os pontos em falta de acordo com os objectivos enunciados neste trabalho ■

CAPÍTULO 5

MODELO PROPOSTO: ESPECIFICAÇÃO



O que quer que seja que tu faças será insignificante, mas é importante que o faças

“Mahatma” Gandhi (1869 – 1948), líder político e espiritual Indiano, adepto e mentor da não violência.

No capítulo 2 foi definido o que era um *Sistema de Produção* e apresentado o conceito de *Produção Integrada por Computador* (CIM), tendo sido referidas algumas das dificuldades de implementação desse conceito. Já no capítulo 3 caracterizou-se o contexto socioeconómico das últimas décadas do século XX, identificaram-se problemas nos sistemas de produção convencionais e, extrapolou-se sobre o futuro da sociedade e dos sistemas produtivos no sentido de se enumerarem as características necessárias à nova geração de sistemas de produção.

No capítulo 4 foi então apresentado o conceito de Sistema Holónico de Produção como solução e arquétipo a seguir no desenvolvimento de novos sistemas de produção. Para a sua implementação sugeriu-se recorrer aos *Sistemas Multiagente* e à *Programação em Lógica Estendida*.

Neste capítulo será efectuada uma análise funcional da produção, identificando-se os seus principais cenários e, em seguida olhar-se-á às principais entidades do sistema e aos seus

relacionamentos básicos. Seguidamente é apresentada a arquitectura holónica proposta e os vários holons constituintes de cada holarquia são descritos em termos de base de conhecimento, ciclo de vida, objectivos e identificação. Essa descrição é em seguida complementada com a identificação de casos de informação incompleta existentes na base de conhecimento de cada holon.

5.1 Introdução

De um ponto de vista económico, um sistema de produção deve oferecer três grandes funções [Sousa *et al.*, 2000a]: *Produção*; *Compras*; e *Vendas*. Todas elas tendo implicações ao nível das existências da empresa. A Figura 5.1 (*adaptada de*: [Sousa *et al.*, 2000a]) dá uma visão geral dos casos de utilização do sistema.

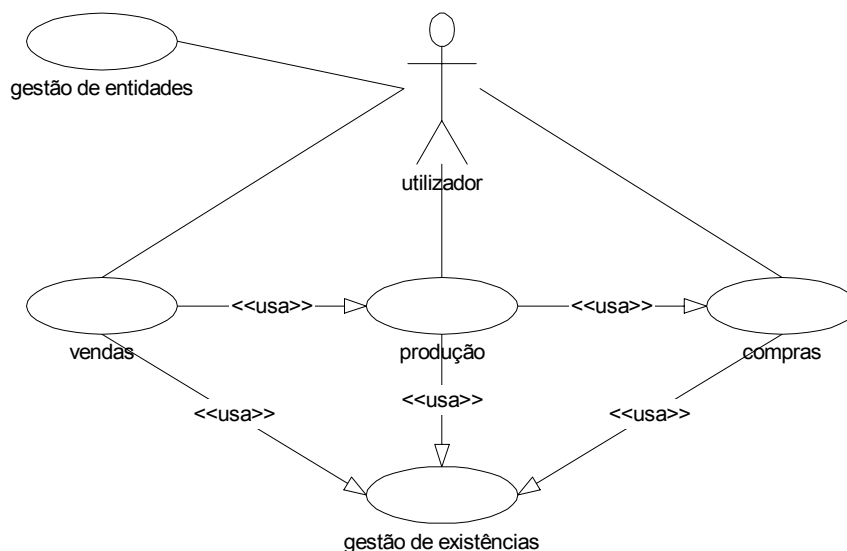


Figura 5.1 – Principais áreas funcionais do sistema

Cada uma das funções referidas é apresentada em seguida:

- *Produção* – corresponde à criação de algo. É um valor acrescentado da empresa para a sociedade em que está inserida;
- *Compras* – a participação da empresa na economia como consumidora de produtos de terceiros, a utilizar na produção;
- *Vendas* – um dos objectivos principais da empresa, que lhe permite escoar as existências, e produzir riqueza para si própria, eventualmente produzindo alterações na sociedade;
- *Gestão de existências* – trata de manter actualizada a informação sobre as existências de produtos e componentes nos diversos armazéns da empresa;

- *Gestão de entidades* – consiste em definir e manter actualizada a informação sobre as várias entidades (e.g., clientes, fornecedores) com as quais a empresa se relaciona.

A actividade de Produção, além de ser a principal no contexto desta análise, é deveras complexa. A Figura 5.2 (adaptada de: [Sousa *et al.*, 2000a]) pretende detalhar um pouco mais essa função.

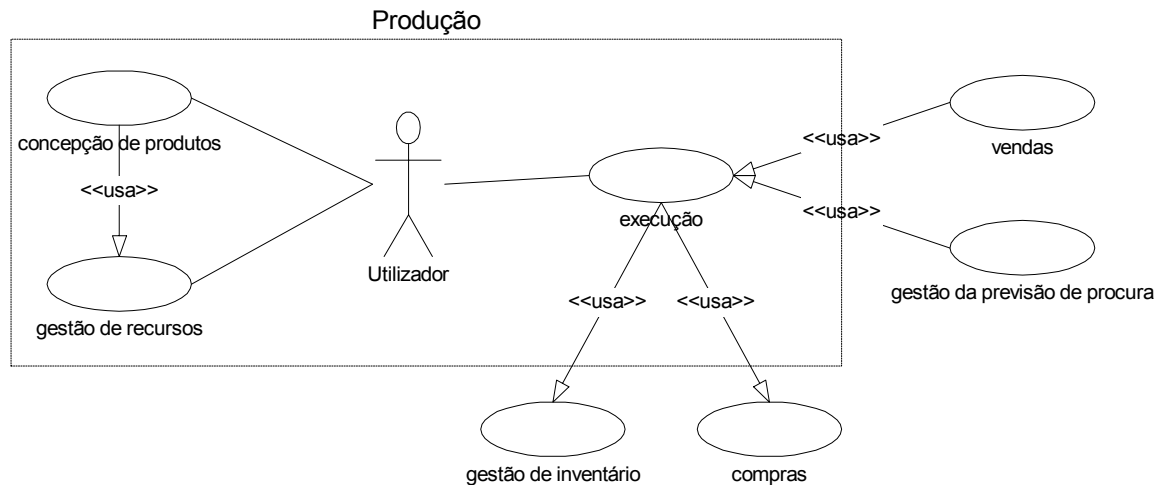


Figura 5.2 – Caso de utilização para a função ‘Produção’

As novas funções apresentadas neste diagrama são dadas a seguir [Sousa *et al.*, 2000a]:

- *Gestão de recursos* – consiste na definição de informação relativa à maquinaria existente na instalação fabril, bem como aos processos;
- *Concepção de produtos* – esta é uma verdadeira função de engenharia, visto envolver a concepção dos produtos da empresa e a especificação dos respectivos processos produtivos;
- *Execução* – função responsável pela efectiva criação de produtos (i.e., geração de existências);
- *Gestão da previsão de procura* – é uma funcionalidade do sistema importante para empresas que não trabalhem apenas em regime de projecto ou encomenda médio/longo prazo, na medida em que tenta prever as requisições futuras de produtos.

Conforme foi apresentado na secção 3.4.2 “O ‘Futuro’ da Produção” os Sistemas de Produção terão tendência a serem distribuídos com grande autonomia por parte de cada entidade constituinte. A Figura 5.3 representa um possível cenário de uma instalação fabril onde se podem ver os vários controladores dos dispositivos, bem como as várias consolas (postos de utilização) disponíveis para os utilizadores.

Na figura pode observar-se que é possível existirem vários controladores de *hardware* e esses controladores não são necessariamente homogéneos. Adicionalmente, cada controlador pode estar ligado a vários dispositivos, podendo existir vários dispositivos do mesmo tipo (ligados ou não a um mesmo controlador). Podem existir vários tipos de posto de utilização (consola) que não estão necessariamente na rede física da empresa e os postos de utilização podem não ser para funções directamente relacionadas com a produção (*e.g.*, Contabilidade).

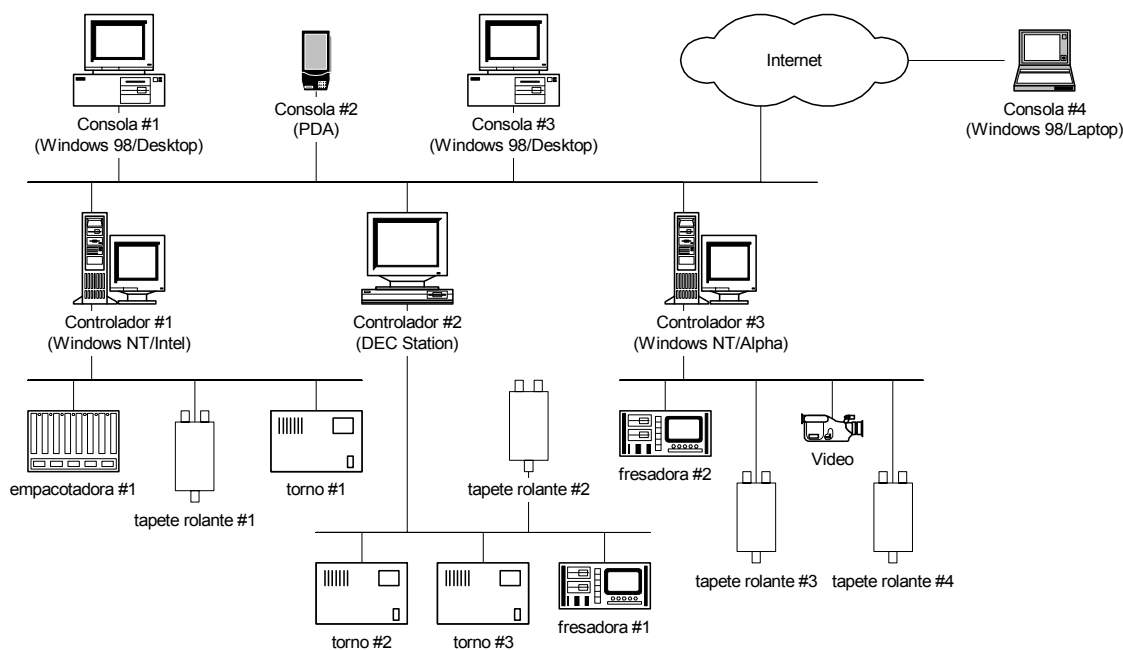


Figura 5.3 – Um possível cenário de exploração

Do ponto de vista da localização geográfica de cada equipamento, a maquinaria fabril, juntamente com os seus controladores, estarão colocados na oficina, enquanto que as consolas estarão espalhadas pela empresa. Assim, a título de exemplo, a “consola #1” poderia ser do departamento de vendas, onde seriam introduzidas as encomendas dos clientes; a “consola #2” seria do Gestor de Produção, onde teria uma cópia do plano de produção diário, que em qualquer altura poderia ser actualizado; a “consola #3” seria do departamento de concepção onde novos produtos são projectados; enquanto que a “consola #4” seria de um vendedor que acede ao sistema a partir das instalações dos clientes para colocar directamente as encomendas ou consultar o estado das encomendas anteriores.

Por outro lado, não é só na instalação fabril que se verificarão alterações, pois a própria empresa terá tendência a ser distribuída, possuindo possivelmente várias instalações fabris espalhadas pelo planeta, necessitando de uma gestão global de recursos (mantendo a autonomia de cada instalação). Além disso, a evolução para empresas virtuais colocará novos desafios perante a integração de funções de empresas separadas, pois provavelmente cada empresa

contribuirá com funcionalidades específicas para o fabrico do produto. Por exemplo, uma empresa pode fabricar os componentes, outra efectuar a montagem e outra os acabamentos, sendo necessário integrar o escalonamento de tarefas das três empresas como se fossem uma só (*e.g.*, após o fabrico dos componentes, colocá-los na “estação de montagem”, mesmo que essa estação de montagem esteja numa empresa separada).

5.2 Arquitectura Proposta

Nesta secção será apresentada a arquitectura proposta para um sistema de produção de nova geração usando uma abordagem holónica.

5.2.1 Arquitectura Holónica do Sistema

Após a análise dos casos de utilização identificaram-se algumas entidades/conceitos que desempenham papeis de destaque no sistema e identificaram-se também os seus principais relacionamentos (Figura 5.4, *adaptada de*: [Sousa *et al.*, 2000a]). Estas entidades são óptimas candidatas a serem representadas no sistema através de holons.

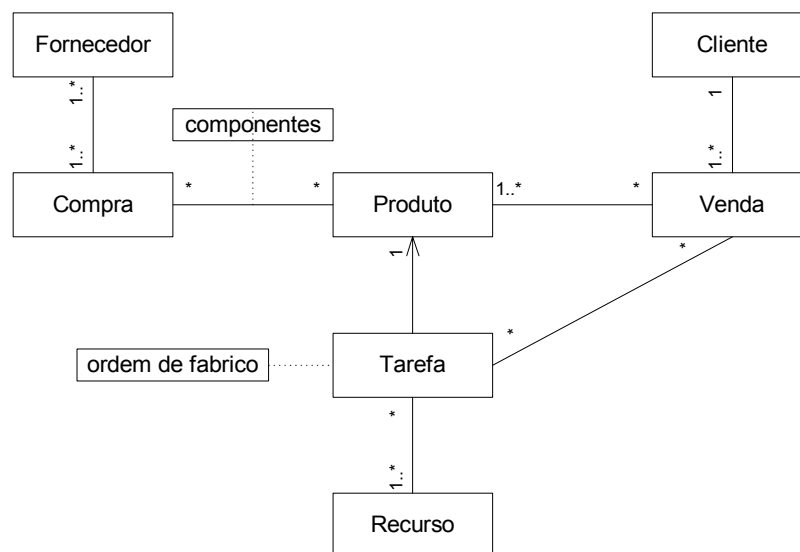


Figura 5.4 – Entidades intervenientes no sistema e seus relacionamentos

Os **Produtos** são entidades que representam algo que a instalação fabril é capaz de produzir. Um produto será normalmente composto por vários outros produtos ou *componentes*, o que é especificado pela *árvore do produto*. Alguns componentes são obtidos através da **Compra** a terceiros (**Fornecedores**). Os produtos da empresa são colocados no mundo exterior quando requisitados pelos **Clientes**. A **Venda** de produtos pode originar ordens de execução (**Tarefas**) a serem enviadas à instalação fabril para a produção de n unidades de um dado produto para uma

determinada data. A instalação fabril tem disponíveis vários **Recursos** (e.g., máquinas) capazes de *executar operações* de transformação ou transporte em/de matéria prima, componentes ou produtos. A obtenção de um produto real é o resultado final do processo produtivo efectuado na instalação fabril, existindo para isso o(s) *plano(s) de produção* de cada produto que especificam como produzi-lo. Cada **Tarefa** utilizará as potencialidades (*habilidades*) dos vários **Recursos** necessários à execução do produto, seguindo um *plano de execução* baseado num dos planos de produção do produto.

Com base nas entidades identificadas na Figura 5.4, é proposta uma arquitectura holónica para um sistema de produção (Figura 5.5 e Figura 5.6).

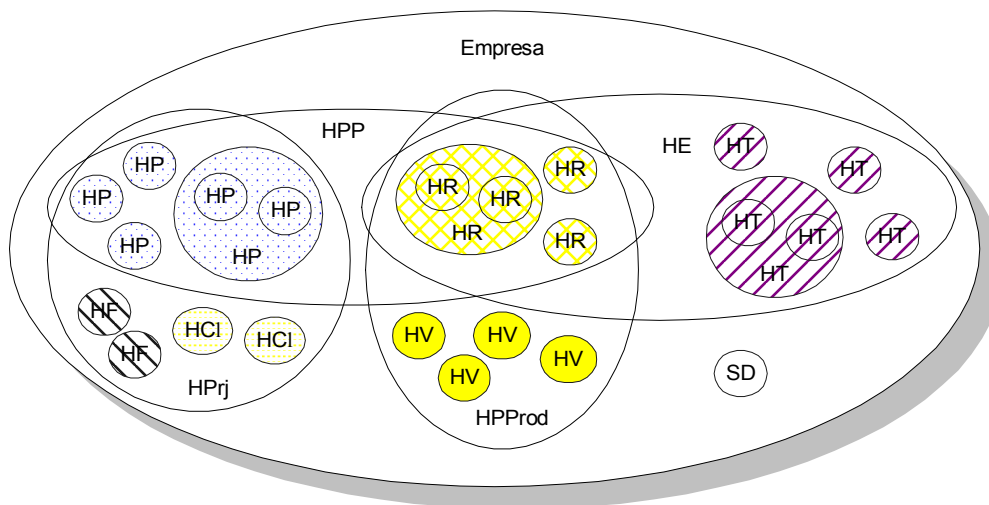


Figura 5.5 – Arquitectura holónica do sistema de fabrico

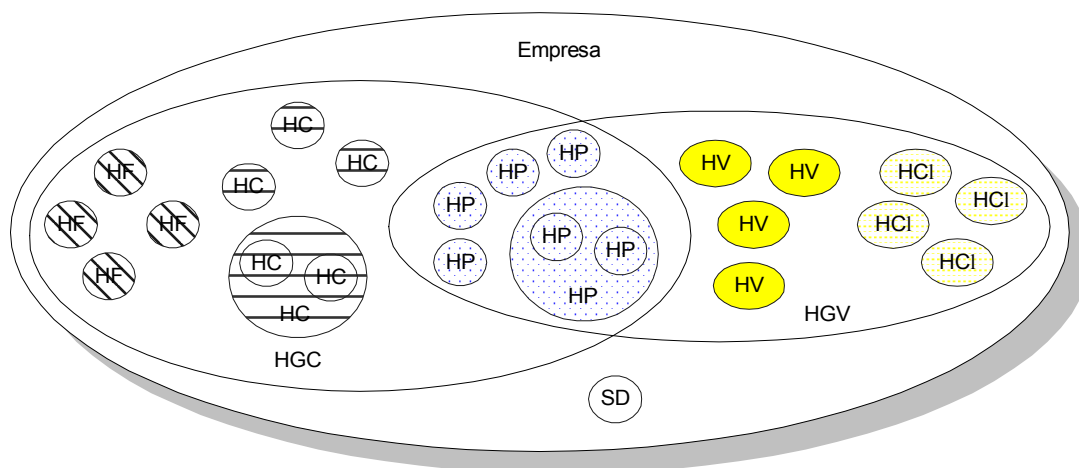


Figura 5.6 – Arquitectura holónica do sistema comercial

Na Figura 5.5 (adaptada de: [Sousa e Ramos, 1998], [Sousa et al., 1999b]) estão representadas apenas as funções da empresa mais directamente relacionadas com o fabrico, e na

Figura 5.6 estão representadas as funções comerciais. Cada elipse na Figura 5.5 e Figura 5.6 representa um holon, sendo o significado das abreviaturas apresentado na Tabela 5.1.

Tabela 5.1 – Legenda da arquitectura

Abreviatura	Significado	Abreviatura	Significado
HF	Holon de Fornecedor	HCI	Holon de Cliente
HC	Holon de Compra	HV	Holon de Venda
HP	Holon de Produto	HR	Holon de Recurso
HT	Holon de Tarefa	SD	Serviço de Directório
HPProd	Holon de Planeamento da Produção	HE	Holon de Escalonamento
HPP	Holon de Planeamento de Processos	HGC	Holon de Gestão de Compras
HGV	Holon de Gestão de Vendas	HProj	Holon de Projecto

Nesta arquitectura podem ser considerados dois níveis hierárquicos, ou mais correctamente dois níveis holárquicos, as funções empresariais e as entidades principais: as funções empresariais (*e.g.*, planeamento da produção) são representadas por holarquias pré-definidas; essas holarquias são constituídas por holons que representam as várias entidades identificadas na Figura 5.4. A Tabela 5.2 apresenta os vários tipos de holons existentes em cada holarquia pré-definida.

Tabela 5.2 – Holons por holarquia

	HGC	HGV	HPrj	HPP	HPProd	HE
HF	√		√			
HC	√					
HP	√	√	√	√		
HV		√			√	
HCI		√	√			
HR				√	√	√
HT						√

As holarquias pré-definidas do sistema correspondem às várias funções da empresa. Assim sendo, o *Holon de Projecto* agrega os Holons de Produto, Holons de Clientes e os Holons de Fornecedores para a actividade de projecto, onde os clientes especificam requisitos dos produtos e os fornecedores cooperam com a empresa na interligação de componentes e criação de novos componentes. O *Holon de Planeamento de Processos* agrega os Holons de Recursos e os Holons de Produtos, sendo responsável pela geração de planos de produção para cada produto, levando em linha de conta os recursos existentes e a especificação do produto (e.g., oriunda do Holon de Projecto, possivelmente suportada por um sistema CAD). O *Holon de Planeamento da Produção* agrega os Holons de Recursos e os Holons de Vendas para efectuar o planeamento de necessidades materiais e o planeamento de capacidades, com vista a cumprir o plano director de produção. O *Holon de Escalonamento e Execução* agrega os recursos físicos da instalação fabril, representados por Holons de Recurso, juntamente com os Holons de Tarefa, que representam as ordens de fabrico, com a finalidade de efectuar o escalonamento dinâmico das tarefas e o controlo (i.e., a execução) do fabrico. O *Holon de Gestão de Compras* está em contacto com o exterior (os fornecedores) tratando dos aspectos relacionados com as compras de componentes e matéria prima, e agregando os Holons de Produtos, Holons de Compra e Holons de Fornecedor. O *Holon de Gestão de Vendas* está também em contacto com o exterior, mais propriamente com os clientes, sendo responsável pela venda e tratamento de encomendas. Este holon agrega os Holons de Cliente, Holons de Vendas e Holons de Produto.

A Tabela 5.3 apresenta algumas características dos vários tipos de holons e holarquias existentes na arquitectura proposta, nomeadamente a cardinalidade de cada tipo de holons e o seu ciclo de vida.

Tabela 5.3 – Características dos holons da arquitectura proposta

	Cardinalidade	Carácter Temporal (Ciclo de Vida)
HGC	Um (1)	Permanente
HGV	Um (1)	Permanente
HPrj	Um (1)	Permanente
HPP	Um (1)	Permanente
HE	Um (1)	Permanente
HProd	Um (1)	Permanente

	Cardinalidade	Carácter Temporal (Ciclo de Vida)
HF	Um por cada fornecedor “activo”	Criado quando se inicia o relacionamento com o fornecedor e destruído quando já não se efectua negócio com o fornecedor em questão há um certo tempo
HC	Um por cada compra em aberto	Criado quando se inicia o processo de compra e destruído quando a compra está completa
HP	Um por cada produto do catálogo da empresa	Criado quando se inicia o desenvolvimento de um novo produto e destruído quando se deixa de dar suporte àquele produto
HV	Um por cada venda em aberto	Criado quando se inicia o processo de venda e destruído quando a venda está completa
HCI	Um por cada cliente em carteira	Criado quando se inicia o relacionamento com o cliente e destruído quando já não se efectua negócio com o cliente em questão há um certo tempo
HT	Um por cada tarefa em execução	Criado quando uma ordem de fabrico é dada à instalação fabril e destruído quando a tarefa for executada ou cancelada
HR	Um por cada recurso da instalação fabril	Criado quando um novo recurso é colocado na instalação fabril e destruído quando o recurso é removido
SD	Um (1)	Permanente

A destruição dos holons referida no ciclo de vida corresponde à cessação do processo computacional dos referidos holons e a colocação do seu estado interno numa base de dados de histórico.

Embora o que foi apresentado até agora nesta secção se refira a uma empresa, os mesmos conceitos podem ser aplicados a uma escala macroscópica para a empresa virtual ou estendida, principalmente os conceitos de Tarefa e Recurso para a integração de várias instalações fabris. Conforme foi apresentado, um holon é uma holarquia e vice-versa. Assim sendo, a holarquia *Empresa* pode ser considerada um holon constituinte de uma outra holarquia superior (e.g., *Empresa Virtual*). Mais propriamente, a holarquia ‘Empresa’ é considerada um *Holon de Recurso* na holarquia ‘Empresa Virtual’. Os *Holons de Tarefa* representariam tarefas de alto nível a efectuar pela Empresa Virtual e que seriam detalhados para cada Empresa constituinte.

5.2.2 Arquétipos de Holons e Holarquias

Na secção anterior foi proposta uma arquitectura para a organização do sistema de produção, nesta secção é apresentado um modelo para a arquitectura interna de cada holon do sistema.

5.2.2.1 Holons

O arquétipo proposto para a implementação de cada holon é representado na Figura 5.7 sendo um modelo abstracto que tenta incluir de forma genérica as características enunciadas para os sistemas baseados em agentes na secção 4.4.1.

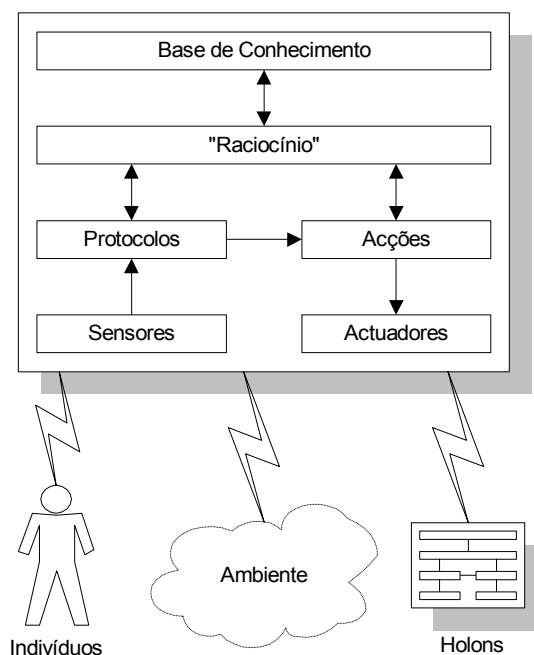


Figura 5.7 – Arquétipo de um agente

Os blocos *Sensores* e *Actuadores* representam a interface do sistema, possibilitando a interacção com seres humanos, o ambiente e outros holons (*e.g.*, a interface com o utilizador, a biblioteca de sistemas de comunicação). O bloco *Protocolos* representa o mecanismo de interpretação da informação recolhida pelos sensores (*i.e.*, percepções), podendo ser identificados com um máquina de estados finitos de um protocolo de comunicação (*e.g.*, protocolo de rede de contrato) ou da interacção Homem-Máquina. Este bloco permite a execução directa de *acções* e/ou o tratamento de conhecimento através do bloco “*Raciocínio*”. O “*raciocínio*” do holon permite-lhe compor resultados tendo em conta o seu conhecimento (*Base de Conhecimento*) e a informação que lhe chega via sensores. Este bloco faz parte da natureza de cada holon, definindo como o holon deve actuar de acordo com o seu estado mental e os objectivos que tem para atingir.

O conhecimento do holon pode ter diversas origens: inerente à concepção do holon; aprendizagem por experiência ou observação; informação de outro holon.

A base de conhecimento de cada holon deve conter um conjunto de axiomas dados em termos da extensão de predicados com o mesmo nome, e que obedecem ao formato [Sousa *et al.*, 2000b]:

nome(Id-Holon, Descrição)
tipo(Classe-Holon)
relação(Id-Outro-Holon, Id-Relação, Valor, Parâmetros)
holarquia(Id-Holon, Lista-de-Holons-Constituintes)

Para além destes axiomas de carácter geral, cada holon contém outro tipo de informação que lhe é específico (*e.g.*, um holon de Recurso possuirá uma agenda com as tarefas escalonadas). O predicado *nome* dá a identificação do holon, existindo apenas uma instância deste predicado na Base de Conhecimento (a Tabela 5.4 apresenta as constantes utilizadas para identificar as holarquias pré-definidas, que fazem parte do conhecimento intrínseco de cada holon do sistema). Cada holon pode pertencer a uma ou mais classes (*i.e.*, um *tipo*, que define as suas capacidades básicas através de um mecanismo de herança [Analide e Neves, 1997] [Sousa *et al.*, 2000b]). O tipo pode também ser visto como o papel a ser representado pelo holon (*e.g.*, recurso, tarefa). Um holon relaciona-se com outros holons (*e.g.*, um holon tarefa necessita de holons recursos para operar) pelo que possui uma *relação* com esse(s) holons. Finalmente, o predicado *holarquia* identifica as holarquias das quais este holon é parte e todo.

Tabela 5.4 – Constantes utilizadas na arquitectura proposta

Constante	Significado
<i>id_srv_dir</i>	Identificação do Holon de Serviços de Directório
<i>id_gestao_compras</i>	Identificação da holarquia de Gestão de Compras
<i>id_gestao_vendas</i>	Identificação da holarquia de Gestão de Vendas
<i>id_plan_processos</i>	Identificação da holarquia de Planeamento de Processos
<i>id_escalonamento</i>	Identificação da holarquia de Escalonamento
<i>id_fornecedor_</i>	Prefixo de identificação para holons de fornecedor
<i>id_compra_</i>	Prefixo de identificação para holons de compras
<i>id_produto_</i>	Prefixo de identificação para holons de produtos
<i>id_venda_</i>	Prefixo de identificação para holons de vendas
<i>id_cliente_</i>	Prefixo de identificação para holons de clientes
<i>id_tarefa_</i>	Prefixo de identificação para holons de tarefas
<i>id_recurso_</i>	Prefixo de identificação para holons de recursos

Além dos axiomas apresentados anteriormente, estão presentes em todos os holons as produções:

$$\begin{aligned}
 nome(Id) &\leftarrow \\
 &\quad nome(Id, _) \\
 todo(H) &\leftarrow \\
 &\quad holarquia(H, L) \wedge \\
 &\quad nome(H) \\
 parte_de(H) &\leftarrow \\
 &\quad holarquia(H, L) \wedge \\
 &\quad nome(N) \wedge \\
 &\quad N \in L \\
 demo_α(P) &\leftarrow \\
 &\quad P \\
 demo_α(P) &\leftarrow \\
 &\quad tipo(MeuTipo) \wedge \\
 &\quad demo_τ(MeuTipo, P) \\
 demo_τ(T, P) &\leftarrow \\
 &\quad tipo(T, T2) \wedge \\
 &\quad demo_τ(T2, P)
 \end{aligned}$$

O predicado *nome/1* denota um método auxiliar para obter apenas a identificação interna do holon. Um holon é uma holarquia (um *todo*) se existe uma holarquia com a sua identificação (recordar que por definição uma holarquia é um holon e vice-versa). Por outro lado, um holon é *parte_de* uma holarquia se é um dos membros da holarquia. O predicado *demo_α* é o responsável pela interpretação das questões colocadas à base de conhecimento dentro dos limites do holon ou então no âmbito do comportamento geral do holon definido pela hierarquia de tipos a que pertence [Sousa *et al.*, 2000b].

5.2.2.2 Holarquias

Conforme foi referido, um holon é uma holarquia e vice-versa, tendo sido advogado que a sua estrutura é uma das principais vantagens dos sistemas holónicos. De um ponto de vista teórico, as holarquias correspondem a domínios de cooperação que implementam mecanismos que facilitam essa cooperação, tais como, espaços de memória partilhada e acessível a todos os holons da holarquia, canais de comunicação dedicados (mais rápidos e com conexão permanente) entre os holons, visão global do problema em questão. A Figura 5.8 demonstra esta interpretação de holarquias, onde se podem observar alguns holons com os respectivos canais de comunicações para interagirem com outros holons e as zonas de memória partilhada e canais de comunicações dedicados às holarquias a que pertencem.

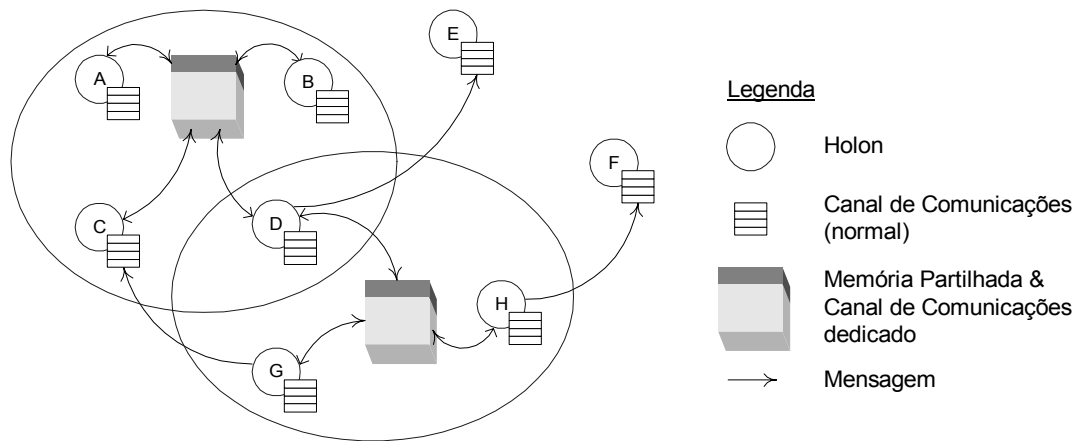


Figura 5.8 – Holarquias

Adicionalmente, a holarquia permite impor algumas regras e controlar em parte o processo de decisão dos holons constituintes, limitando-lhes a autonomia, de forma a garantir objectivos “globais” (à holarquia e não ao sistema). Desse ponto de vista, uma holarquia pode ser comparada a um agente supervisor/coordenador existente nalgumas arquitecturas de sistemas multiagente (*e.g.*, [Rabelo e Camarinha-Matos, 1994] [Maturana e Norrie, 1996]).

As holarquias podem existir *a priori*, definidas durante a concepção do sistema (*e.g.*, a holarquia de escalonamento da arquitectura proposta) ou podem ser criadas dinamicamente no decorrer da execução do sistema. No caso das holarquias dinâmicas é necessário definir regras para a formação de holarquias, bem como definir o processo de adesão e remoção. Ou seja, é necessário definir um protocolo de gestão de holarquias (o termo protocolo é aqui utilizado como um conjunto de regras que definem os procedimentos a executar pelos vários holons com vista à gestão das holarquias – criação, adesão, remoção, destruição – e não apenas como o conjunto de mensagens a serem trocadas). Para as holarquias pré-definidas são conhecidos quais os holons (ou tipos de holons) que fazem parte de cada holarquia, sendo por isso responsabilidade desses mesmos holons aderirem a essas holarquias.

A formação de novas holarquias (cuja existência temporal será em princípio reduzida) deve ser iniciada de acordo com as próprias condições dinâmicas do sistema, por exemplo, a criação de um novo produto. Os holons existentes no sistema devem reagir a estes eventos reagrupando-se numa nova holarquia que possibilite dar resposta correcta e atempada ao problema colocado. A criação de uma nova holarquia não pressupõe a destruição de algumas das existentes. As holarquias podem ser “desfeitas” quando o problema que pretendiam resolver for solucionado; por exemplo, supondo a criação de uma holarquia por cada tarefa a executar (composta pelo subconjunto de Holons de Recurso escolhidos para fabricar o produto em questão) essa holarquia será destruída quando o produto for fabricado e colocado em armazém.

O protocolo de gestão de holarquias deve então definir os eventos que iniciam o processo de formação de novas holarquias e quais os holons candidatos a formarem essa holarquia, sendo depois necessário definir as regras de selecção desses candidatos (*e.g.*, maximização de uma função de utilidade). Este protocolo deve também definir em que condições uma holarquia deve ser destruída e quais os mecanismos para os holons aderirem e abandonarem as holarquias.

Esta criação de holarquias, como reacção aos eventos do sistema, corresponde às características de dinamismo, adaptabilidade e agilidade enunciadas como requisitos para os sistemas de produção de nova geração.

No decorrer deste trabalho, o protocolo de gestão de holarquias definido cingiu-se às holarquias pré-definidas, simplificando-se o processo de adesão e remoção ao assumir a veracidade e validade de cada holon, ou seja, assume-se que apenas os holons que realmente têm direito de pertencer a uma holarquia executam o pedido de adesão.

5.3 Especificação dos Holons

Nesta secção é apresentado cada um dos holons existentes na arquitectura descrita na secção anterior. Cada holon é especificado do ponto de vista do conhecimento que possui, objectivos e ciclo de vida.

5.3.1 Serviço de Directório

Este é um holon auxiliar, funcionando como uma base de dados central de identificação de holons e anúncio de funcionalidades [Sousa *et al.*, 1999b], já que cada holon regista no serviço de directório quais as suas funcionalidades (acções que o holon é capaz de executar, serviços que pode fornecer).

A identificação deste holon é a constante do sistema “*id_srv_dir*” (*vide* Tabela 5.4). Este holon tem por objectivo servir os outros holons do sistema fornecendo-lhes a informação referente às potencialidades de cada um, funcionando assim como um serviço de páginas amarelas.

A base de conhecimento deste holon é composta por predicados do tipo enunciado a seguir, que permitem guardar informação sobre as potencialidades de cada holon.

serviços(Id-Holon, Lista-de-Habilidades)

O holon de serviço de directório contém também as seguintes produções:

$$\begin{aligned} & \text{tipo}(\text{tipo_srv_dir}) \\ & \text{fornece}(\text{Id-Holon}, \text{Id-Habilidade}) \leftarrow \\ & \quad \text{serviços}(\text{Id-Holon}, L) \wedge \end{aligned}$$

$$\begin{aligned}
& \text{membro}(\text{Id-Habilidade}, L) \\
& \text{fornecedores}(\text{Id-Habilidade}, \text{Lista-de-Holons}) \leftarrow \\
& \quad \text{todas_as_soluções}(H, \text{fornece}(H, \text{Id-Habilidade}), \text{Lista-de-Holons})
\end{aligned}$$

O predicado *fornece* definido em termos de serviços indica se um determinado holon executa um determinado serviço, enquanto que o predicado *fornecedor* permite cruzar informação sobre quais os holons capazes de executar uma dada tarefa.

5.3.2 Holon de Produto

Um *Holon de Produto* representa um item do catálogo de produtos fabricados pela empresa, sendo responsável pelos aspectos relacionados com a fabricação e logística de gestão das existências [Sousa *et al.*, 1999b]. Este holon contém informação actualizada sobre a classe do produto; *i.e.*, o modelo de produtos daquele tipo (*e.g.*, lista de componentes, processos de fabricação, informação do ciclo de vida, requisitos de utilização) e não de um produto em concreto.

Uma das responsabilidades deste holon é fornecer informação sobre os planos de produção do produto considerando planos alternativos para diferentes critérios (*e.g.*, custo de fabrico, tempo de execução) e eventualmente indicar alternativas pontuais em caso de acontecimentos imprevistos (*e.g.*, avaria de recursos). Assim sendo, os Holons de Produto possuem funcionalidades normalmente associadas às funções de Projecto e Planeamento de Processos.

Cada holon deste tipo tem por objectivo servir os outros holons do sistema quanto à informação referente ao modelo de produto. O ciclo de vida de um holon de produto começa ao iniciar-se o processo de desenvolvimento de um novo produto, acompanhando o desenvolvimento do produto desde as fases iniciais de análise de requisitos e projecto. Após o desenvolvimento do produto (incluindo o planeamento de processos) o holon de produto mantém informação actualizada sobre o ciclo de vida e o modelo do produto, juntamente com informação logística sobre as existências e os fornecedores de componentes. O holon de produto é destruído quando a empresa deixa de dar assistência ao produto, passando a sua informação para o histórico.

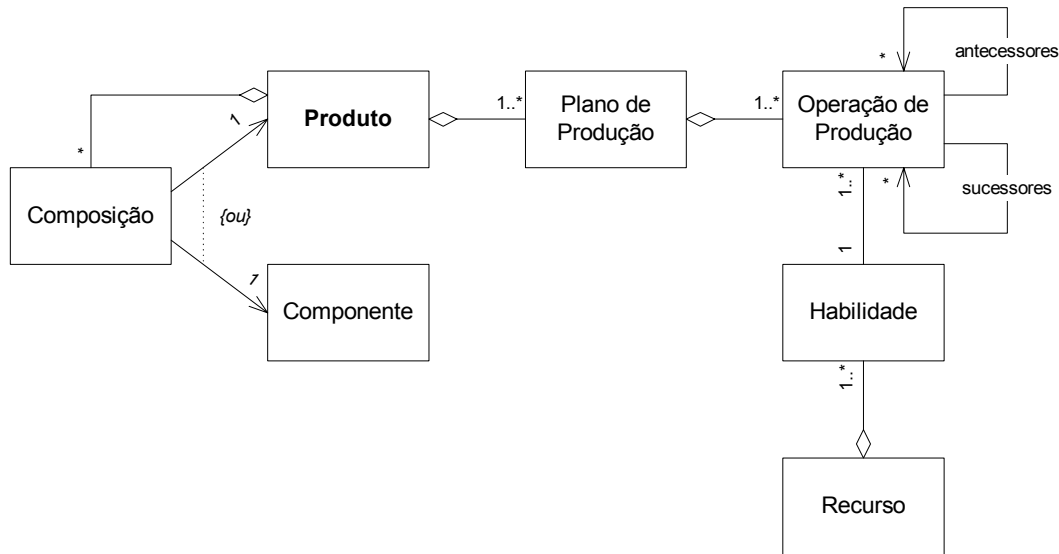


Figura 5.9 – Diagrama de classes para holons de produto

A Figura 5.9 apresenta o diagrama de classes para um Holon de Produto, que é representado na sua base de conhecimento pelas extensões dos predicados dados a seguir:

produto(*Id*, *Descrição*, *QtStock*)
atributo(*Parametro*, *Valor*)
plano(*Id*, *Id-Critério*, *Atributos-Plano*, *Operações*)
composição(*Id-Sub-Produto*, *Id-Componente*, *Qt*)

O predicado *produto* denota alguma da informação de carácter geral acerca do produto. O predicado *atributo* refere-se aos vários atributos do produto (e.g., classificação ABC, custo médio de produção). Cada instância do predicado *plano* descreve um plano de produção do produto otimizado segundo um dado critério (e.g., minimização do tempo de execução). O termo *Atributos-Plano* corresponde ao conjunto de atributos daquele plano (e.g., lote ideal, custo de produção, custo de material), sendo uma lista de pares (*Parâmetro*, *Valor*). O termo *Operações* corresponde ao grafo de operações necessárias à execução do produto de acordo com o plano a que se refere. Para cada operação é(são) indicado(s) a(s) operação(ões) sucessora(s) e a(s) antecessora(s), assim como qual o recurso ou tipo de recurso a utilizar. O atributo *Operações* traduz-se numa lista de termos dados na forma:

nodo(*Id*, *Id-Habilidade*, *EspecRecurso*, *Duração*, *Pred*, *Succ*)

em que *Pred* e *Succ* são listas de identificadores de nós no grafo de operações, representando as operações antecessoras e sucessoras, respectivamente. O atributo *EspecRecurso* indica se a operação em causa deve ser executada num recursos específico – *rec(Id_Holon_Recurso)* – ou em qualquer recurso capaz de executar aquela operação – *oper*.

A Figura 5.10 apresenta um plano de produção e a sua representação no Holon de Produto.

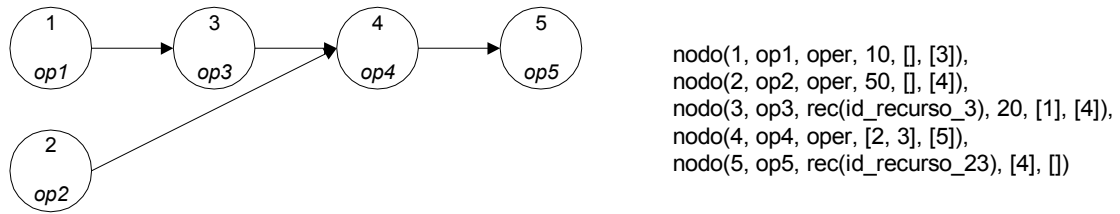


Figura 5.10 – Exemplo de representação de um plano

Um dado produto é constituído por componentes fornecidos por terceiros (*i.e.*, fornecedores) ou por outros produtos fabricados dentro da empresa. O predicado *composição* denota a constituição do produto (*i.e.*, a sua árvore de produto), sendo que em cada instância deste predicado o atributo *Id-Sub-Produto* faz referência a um produto ou está inicializado a zero caso se trate de um componente. Da mesma forma, o atributo *Id-Componente* faz referência a um componente ou está inicializado a zero caso se trate de um produto. A informação sobre componentes deve ser obtida da base de dados empresarial de suporte ao sistema, para evitar duplicação e inconsistência de informação.

A base de conhecimento dos holons de produtos contém também termos do seguinte tipo:

tipo(tipo_produto)
operacoes-plano(PlanId, LOp) ←
plano(PlanId, _, _, LOp)

A identificação deste tipo de holon é o prefixo “*id_produto_*” (uma das constante do sistema – *vide* Tabela 5.4) concatenado com o código de produto constante do catálogo da empresa. Os holons deste tipo registam-se no serviço de directório do sistema através de uma mensagem com o seguinte conteúdo:

registar(Id, [produto(PId)])

onde *Id* é obtido através de *nome(Id)* e *PId* é obtido através de *produto(PId, _, _, _, _)*.

5.3.3 Holon de Tarefa

Os *Holons de Tarefa* representam as ordens de fabrico enviadas à instalação fabril para a execução de *n* itens de um determinado produto, sendo responsáveis por garantir a sua execução correcta e atempada [Sousa e Ramos, 1998] [Sousa *et al.*, 1999b]. Um Holon de Tarefa gere informação relativa ao estado do produto; *i.e.*, a informação sobre o estado do processo de fabrico daquele(s) produto(s) em concreto. Um Holon de Tarefa contempla funcionalidades associadas aos processos de Escalonamento e Controlo.

Os holons deste tipo têm por objectivo efectuar o escalonamento da ordem de fabrico e monitorizar a sua execução. Um Holon de Tarefa tem um “nascimento” e uma “morte”. O seu ciclo de vida inicia-se quando a tarefa dá entrada no sistema (encomenda de um cliente ou balanço de existências). O holon “morre” quando a tarefa foi executada ou cancelada (*e.g.*, o cliente cancelou a encomenda). Os Holons de Tarefa são criados dinamicamente por um componente do Holon de Escalonamento denominado *Lançador de Tarefas*. Durante a sua existência cada Holon de Tarefa entra em negociação com os holons de recurso para a execução de cada operação do plano de produção de forma a dar cumprimento à tarefa, e monitoriza a sua execução. Após o término ou cancelamento da ordem de fabrico, o Holon de Tarefa “morre” e a sua informação é guardada numa base de dados histórica.

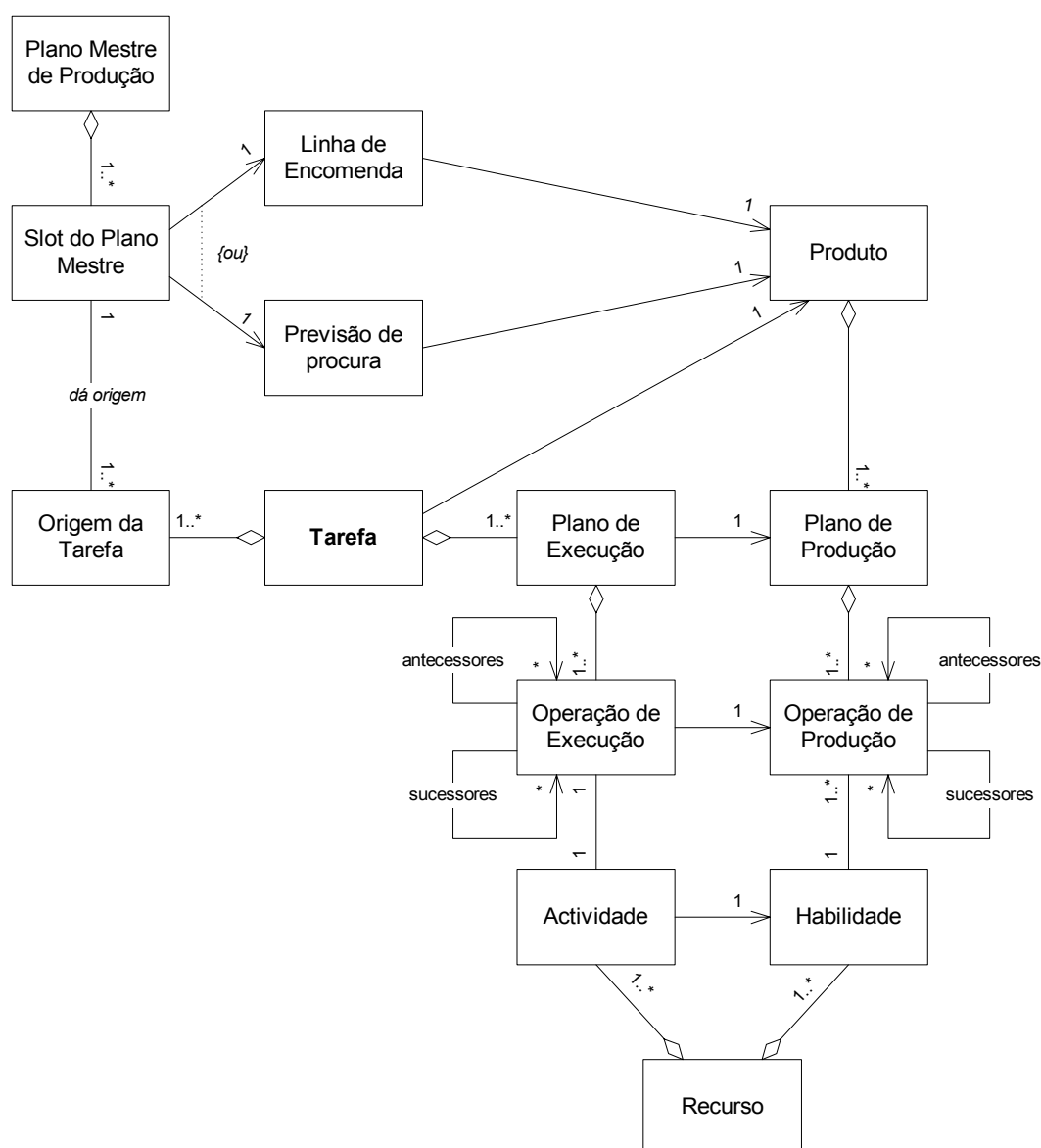


Figura 5.11 – Diagrama de classes para holons de tarefa

A Figura 5.11 apresenta o diagrama de classes para um Holon de Tarefa, que é representado na sua base de conhecimento pelas extensões dos predicados:

tarefa(*TId*, *NOF*, *Id-Produto*, *Quantidade*, *DataLimite*)
atributo(*Parâmetro*, *Valor*)
origem(*Id-Slot-PM*, *Qt*)
plano(*Id*, *Critério*, *Atributos*, *Operações*)

O predicado *tarefa* representa os dados da ordem de fabrico (*NOF*) associada a este holon de tarefa, indicando qual a *Quantidade* de produto a fabricar até que data (*DataLimite*). Outros atributos de tarefa (*e.g.*, data de criação, cliente a que se destina, precisão) surgem como atributos do predicado *atributo*. As instâncias do predicado *origem* representam as várias linhas do plano mestre de produção que deram origem a esta ordem de fabrico. O predicado *plano* identifica qual o plano de produção seleccionado para esta tarefa. A selecção do plano de produção de entre os planos alternativos de um produto é da responsabilidade do Holon de Planeamento de Processos.

A base de conhecimento dos holons de produto contém também os seguintes axiomas, que permitem derivar conhecimento a partir dos axiomas base:

tipo(*tipo_produto*)
holon(*Nome*, *Desc*) \leftarrow
 tarefa(*TId*, $_$, $_$, $_$, $_$) \wedge
 str_term_cat(*id_tarefa*, *TId*, *Nome*) \wedge
 str_term_cat("Tarefa Dinâmica N^o", *TId*, *Desc*)
operacoes-plano(*LOp*) \leftarrow
 plano($_$, $_$, $_$, *LOp*)

A identificação deste tipo de holon é o prefixo "*id_tarefa*" (constante do sistema – *vide* Tabela 5.4) concatenado com o código de tarefa atribuído pelo sistema. Os holons deste tipo registam-se no serviço de directório do sistema através de uma mensagem com o seguinte conteúdo:

registar(*Id*, [*tarefa*(*TId*)])

onde *Id* é obtido através de *nome*(*Id*) e *TId* é obtido através de *tarefa*(*TId*, $_$, $_$, $_$, $_$).

5.3.4 Holon de Recurso

Um *Holon de Recurso* representa o estado actual de um recurso físico da instalação fabril; *e.g.*, estado de funcionamento, actividades a efectuar [Sousa e Ramos, 1998] [Sousa *et al.*, 1999b]. A utilização dos recursos é representada numa *agenda* – a sequência de operações a serem executadas. As funcionalidades deste holon denotam as operações de transporte ou maquinação dos recursos físicos (*e.g.*, furar). Um Holon de Recurso possui funcionalidades associadas ao Planeamento de Processos, Planeamento de Produção, Escalonamento e Execução de tarefas.

Um Holon de Recurso pode representar um recurso básico (*e.g.*, uma fresadora) ou uma célula de trabalho composta por vários recursos (*e.g.*, fresadora, sistema de visão e robô). Neste último caso, o holon agrega numa holarquia os vários holons de recurso para cada um dos equipamentos. Estes holons serão normalmente compostos por duas partes, uma de *hardware* e outra de *software*. O *hardware* corresponde ao equipamento real da fábrica, enquanto que a parte de *software* permite a ligação entre o equipamento e o sistema informático. É no entanto possível existirem Holons de Recurso apenas de *software*, para simulação de equipamento ou para representação de macro-recursos com existência lógica (*e.g.*, uma instalação fabril).

Este holon tem por objectivo controlar o equipamento físico da instalação fabril, fornecer informação aos outros holons sobre as habilidades do recurso e gerir as actividades atribuídas ao recurso. O ciclo de vida de um holon de recurso é bastante longo. Um Holon de Recurso existe durante toda a existência do recurso físico na fábrica, sendo criado quando o recurso é colocado nas instalações e destruído quando o recurso é desactivado. Durante a sua existência o Holon de Recurso executa os comandos enviados pelo controlador (*e.g.*, carregar programa NC) e negocia operações com Holons de Tarefa para o cumprimento das ordens de fabrico.

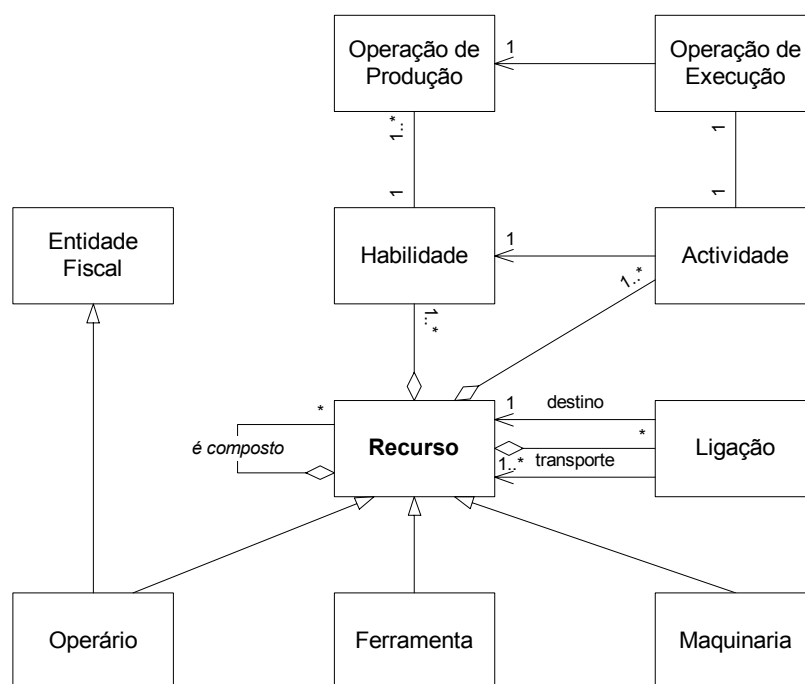


Figura 5.12 – Diagrama de classes para holons de recurso

A Figura 5.12 apresenta o diagrama de classes para um Holon de Recurso, que é representado na sua base de conhecimento por produções dos seguintes modelos:

recurso(*RI*, *Descrição*, *Data-Criação*, *Atributos*)
habilidade(*Id-Habilidade*, *Descrição*, *Duração*, *Custo*, *CNC*, *Argumentos*)
actividade(*TId*, *OpId*, *Qt*, *Inicio*, *Fim*, *DataLimite*, *Estado*, *Pred*, *Succ*)

O predicado *recurso* representa alguma informação de carácter geral do recurso. Os axiomas *habilidade* enumeram as várias operações de maquinação e respectivo código CNC, onde o termo *Duração* corresponde à duração da execução do programa CNC para um produto (quantidade = 1) e *Custo* é uma função que devolve o custo de operação (por item) desta habilidade tendo em conta os parâmetros de afinação da máquina (e.g., a velocidade da operação e a ferramenta utilizada). Os predicados *actividade* denotam uma actividade que este recurso tem contratada com uma tarefa, em que *TId* é a identificação da tarefa contratante, *OpId* é a identificação da operação contratada, *Qt* é a quantidade de itens a processar, *Inicio* é o instante de tempo em que está previsto iniciar-se a actividade, *Fim* é o instante de tempo previsto para a conclusão da tarefa, *DataLimite* é a data de finalização da tarefa, *Estado* representa o estado da actividade (e.g., em execução), *Pred* e *Succ* são listas de recursos contactados para as operações antecessoras e sucessoras respectivamente. A classe “ligação” do diagrama da Figura 5.12 é representada nos predicados tipo *relação* e denota os custos dos transportes de e para este recurso (e.g., passadeiras rolantes, robôs, AGVs). A relação “é composto” é dada pela extensão do predicado *holarquia*.

A base de conhecimento destes holons, para além das extensões dos predicados referidos em epígrafe, também contém as seguintes produções:

```

tipo(tipo_recurso)
holon(Nome, Desc) ←
    recurso(RId, Desc, _, _) ∧
    str_term_cat(id_recurso_, RId, Nome)
agenda(LActv) ←
    todas_as_soluções(act(T, O, Q, I, F, Dt, E, P, S), actividade(T, O, Q, I, F,
Dt, E, P, S), LActTemp) ∧
    ordenar_intervalos(LActTemp, LActv)
agenda-intervalos(LActv) ←
    todas_as_soluções(tw(I, F), actividade(_, _, _, I, F, _, _, _), LActTemp) ∧
    ordenar_intervalos(LActTemp, LActv)

```

A identificação deste tipo de holon é o prefixo “*id_recurso_*” (constante do sistema – vide Tabela 5.4) concatenado com o código de recurso. Os holons deste tipo registam-se no serviço de directório do sistema através de uma mensagem com o seguinte conteúdo:

```
registra(Id, [recurso(RId) | Hab])
```

onde *Id* é obtido através de *nome(Id)*, *RId* é obtido através de *recurso(RId, _, _, _)* e *Hab* é obtido através de *todas_as_soluções(operacao(I), habilidade(I, _, _, _), Hab)*.

5.3.5 Holon de Compra

Um *Holon de Compra* refere-se aos processos associados às operações de compra de componentes e matéria prima [Sousa *et al.*, 2000a]. Este tipo de holons definem-se em termos dos

contactos que se estabelecem com um ou mais fornecedores com vista ao fornecimento de um material e/ou componente. Os holons deste tipo implementam funcionalidades do Departamento de Compras.

Este holon tem por objectivo representar o processo de compra. O seu ciclo de vida inicia-se quando se pedem orçamentos a vários fornecedores ou então quando é feita uma encomenda a um dado fornecedor. Após a criação, o holon representará as várias fases do processo de compra, nomeadamente a escolha dos fornecedores e eventualmente quais os componentes a encomendar a cada um. O holon termina quando a compra é cancelada ou é completamente satisfeita; *i.e.*, todos as encomendas são entregues e pagas, passando nessa altura o estado interno do Holon de Compra para histórico.

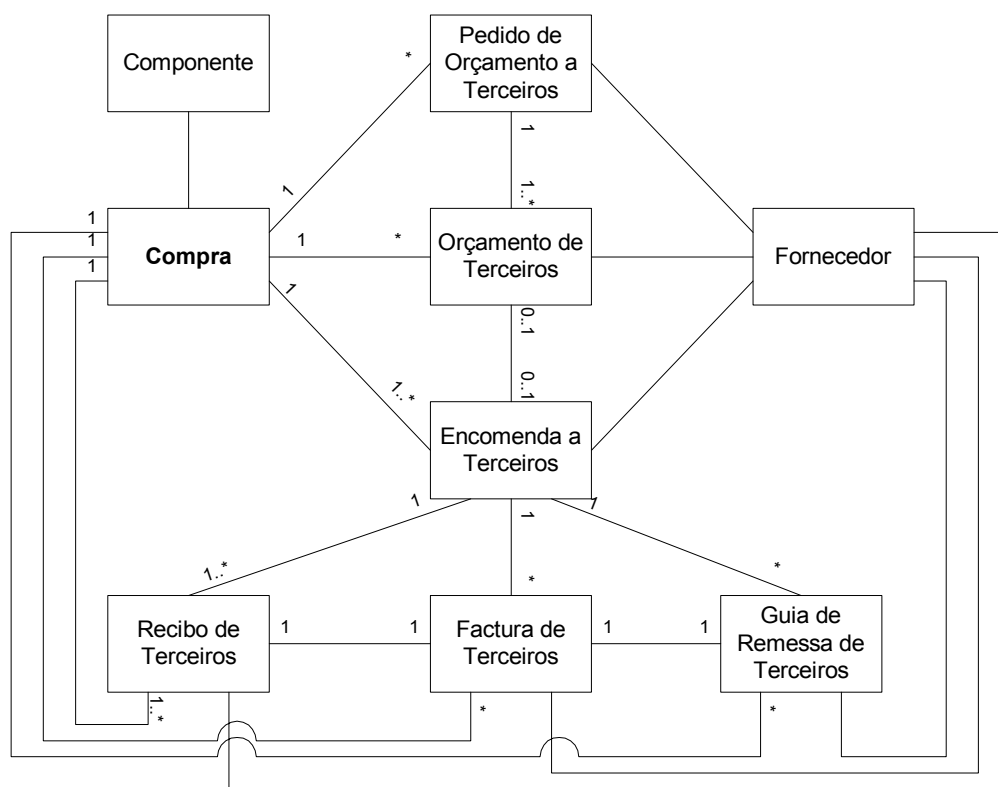


Figura 5.13 – Diagrama de classes para holons de compras

A Figura 5.13 apresenta o diagrama de classes para um Holon de Compra, que é representado na sua base de conhecimento pelas extensões dos predicados dados a seguir:

compra(CId, Data-Criação, Atributos)
pedido-orçamento(NDI, FId, NDoc, Data, Pedido, Condições, Atributos)
orçamento(NDI, FId, NDoc, Data, Proposta, Condições, Atributos)
encomenda(NDI, FId, NDoc, Data, DueDate, Pedido, Condições, Atributos)
guia-remessa(NDI, FId, NDoc, Data, Produtos, Condições, Atributos)
factura(NDI, FId, NDoc, Data, Produtos, Condições, Atributos)
recibo(NDI, FId, NDoc, Data, NDoc-Factura, Condições, Atributos)

Os atributos *Atributos* são listas de pares (*Atributo, Valor*) contendo atributos específicos presentes em cada predicado (e.g., anotações, local de descarga). Os atributos do tipo *Pedido* denotam uma lista de tuplos na forma (*Id-Produto, Quantidade, Condições*) especificando quais os produtos para que se requisita orçamento. Os termos *Condições* são listas de tuplos na forma (*Parâmetro, Relação, Valor*) que especificam as condições desejadas ou impostas pelo cliente/empresa para um item específico ou o seu conjunto (e.g., qualidade, limite de preço). Os termos do tipo *Proposta* correspondem a uma lista de termos na forma (*Id-Produto, Quantidade, Preço, Condições*) especificando a proposta de orçamento e as condições em que o preço de cada item é válido. Os atributos *Produtos* denotam listas na forma (*Id-Produto, Quantidade, Preço*) especificando os itens enviados ao cliente. O termo *NDoc* corresponde ao número do documento do fornecedor enquanto o termo *NDI* corresponde ao número de documento atribuído internamente. O termo *FID* corresponde à identificação de um fornecedor.

A base de conhecimento destes holons contém ainda a extensão do predicado:

tipo(tipo_compra)

A identificação deste tipo de holon é dada pelo prefixo “*id_compra_*” (constante do sistema – vide Tabela 5.4) concatenado com o código de compra. Os holons deste tipo registam-se no serviço de directório do sistema com uma mensagem com o seguinte conteúdo:

registar(Id, [compra(CId)])

onde *Id* é obtido através de *nome(Id)* e *CId* é obtido através de *compra(CId, _, _, _)*.

5.3.6 Holon de Fornecedor

Um *Holon de Fornecedor* representa a visão que a empresa tem de um dos seus fornecedores [Sousa *et al.*, 2000a]. Idealmente, holons deste tipo deveriam comunicar com holons representantes do fornecedor, estendendo assim a cadeia de negócio para o exterior da empresa. Estes holons são responsáveis por toda a informação do fornecedor (e.g., classificação ABC, total de vendas, que tipo de componentes fornecem). Adicionalmente, estes holons dão uma medida da satisfação da empresa para com os serviços prestados pelo fornecedor.

Este holon tem por objectivo servir os outros holons do sistema com informação relativa a um dado fornecedor. Um Holon de Fornecedor inicia o seu ciclo de vida quando se efectua o primeiro contacto com o fornecedor (e.g., para pedir orçamento ou encomendar material), existindo até ao momento em que se desiste de ter negócio com aquele fornecedor (e.g., por violação de acordo, por questões políticas/morais). Durante a sua existência este holon gere a informação relativa ao fornecedor.

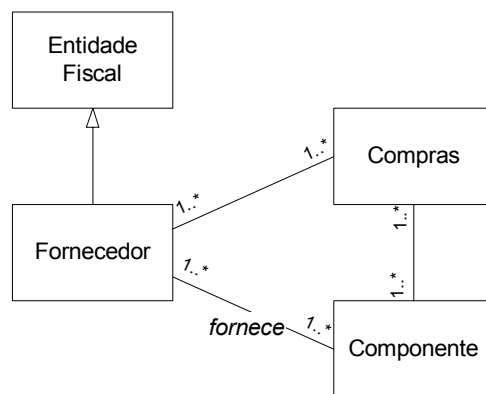


Figura 5.14 – Diagrama de classes para holons de fornecedor

A Figura 5.14 apresenta o diagrama de classes para um Holon de Fornecedor, que se materializa na base de conhecimento pelas extensões dos seguintes predicados:

entidade-fiscal(NIF, Morada, Designação, Contactos)
fornecedor(FId, Data-Criação, Data-Ultimo-Compra)
atributo(Parâmetro, Valor)
fornece(Id-Componente, Atributos)

O predicado *entidade-fiscal* corporiza a informação que tipifica uma entidade (firma) em termos fiscais, no qual o atributo *Contactos* é uma lista de contactos para aquela entidade fiscal (e.g., telefone, correio electrónico). Os predicados *atributo* descrevem os atributos do fornecedor (e.g., total de compras, satisfação). Cada instância do predicado *fornece* descreve informação sobre um componente que este fornecedor pode fornecer, indicando alguns parâmetros desse fornecimento no atributo *Atributos* (e.g., preço da última compra, disponibilidade, satisfação).

A base de conhecimento destes holons também contém o seguinte predicado:

tipo(tipo_fornecedor)

A identificação deste tipo de holon é o prefixo “*id_fornecedor_*” (constante do sistema – vide Tabela 5.4) concatenado com o código de fornecedor. Os holons deste tipo registam-se no serviço de directório do sistema com uma mensagem com o seguinte conteúdo:

registar(Id, [fornecedor(FId) | Srv])

onde *Id* é obtido através de *nome(Id)*, *FId* é obtido através de *fornecedor(FId, _, _)* e *Srv* é obtido através de *todas_as_soluções(fornece(I), fornece(I, _), Srv)*.

5.3.7 Holon de Venda

Um *Holon de Venda* corporiza um acto específico, o de uma venda, e os processos e documentação a si associados [Sousa *et al.*, 2000a]. Este tipo de holons representam contactos

para vendas sob a forma de encomendas *dos* clientes ou propostas de fornecimento *ao* cliente. Caso a venda se concretize, este holon materializará a encomenda do cliente, com a correspondente ordem à produção se os produtos encomendados não existirem em armazém. A venda processa-se de acordo com as condições acordados entre o cliente e a empresa (*e.g.*, qualidade, condições de pagamento). Os holons deste tipo implementam funcionalidades do Departamento de Vendas.

Este tipo de holon tem por objectivo corporizar o processo de venda. O ciclo de vida de um Holon de Vendas tem a sua génese numa de três situações possíveis: (1) o cliente pede um orçamento para o fornecimento de um ou mais produtos; (2) o cliente coloca uma encomenda; (3) a empresa envia ao cliente um orçamento para o fornecimento de produtos sem ter havido um pedido explícito do cliente. O último cenário aponta para um concurso público ou venda directa. Após a sua criação, o holon atravessará horizontalmente as várias fases do processo de venda até à sua conclusão devido à não concretização do negócio, ao cancelamento da encomenda ou à entrega da mercadoria (e respectivo pagamento) ao cliente. Após a venda, o holon “morre” e passa o seu estado interno para a base de dados históricos.

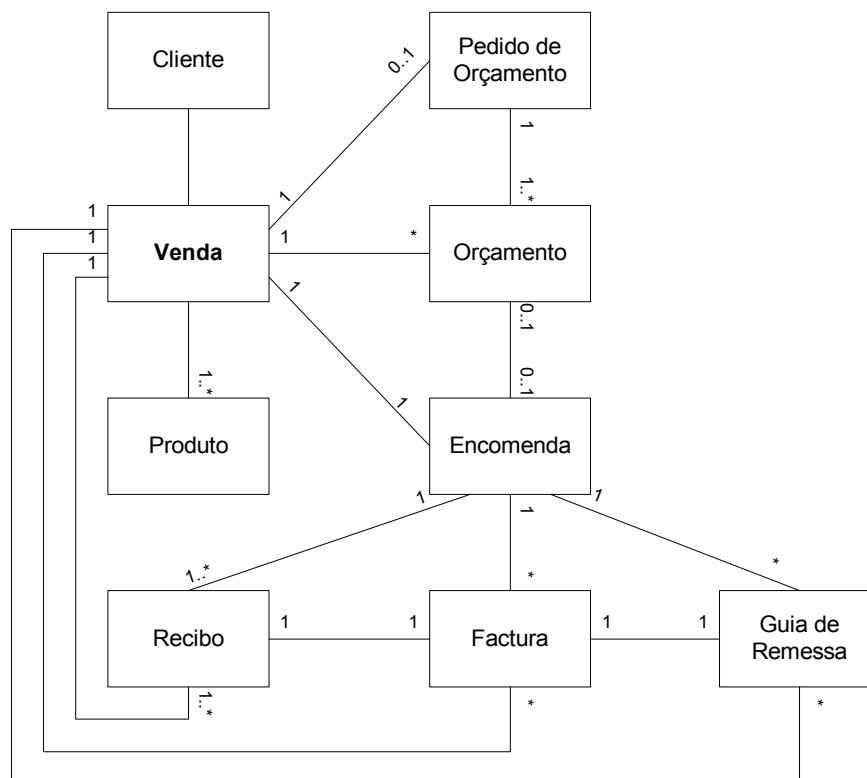


Figura 5.15 – Diagrama de classes para holons de vendas

A Figura 5.15 apresenta o diagrama de classes para um Holon de Vendas, que é representado na sua base de conhecimento pelas extensões dos seguintes predicados:

venda(VId, Id-Cliente, Data-Criação, Atributos)
pedido-orçamento(NDoc, Data, Pedido, Condições, Atributos)
orçamento(NDoc, Data, Proposta, Condições, Atributos)
encomenda(NDoc, Data, DataLimite, Pedido, Condições, Atributos)
guia-remessa(NDoc, Data, Produtos, Condições, Atributos)
factura(NDoc, Data, Produtos, Condições, Atributos)
recibo(NDoc, Data, NDoc-Factura, Condições, Atributos)

Os atributos *Atributos*, *Pedido*, *Condições*, *Proposta* e *Produtos* têm o mesmo significado anteriormente apresentado para os Holons de Compras. O atributo *NDoc* corresponde ao número do oficial do documento.

A base de conhecimento destes holons também contém o seguinte predicado:

tipo(tipo_venda)

A identificação deste tipo de holon é o prefixo “*id_venda_*” (constante do sistema – vide Tabela 5.4) concatenado com o código de venda. Os holons deste tipo registam-se no serviço de directório do sistema com uma mensagem com o seguinte conteúdo:

registar(Id, [venda(VId)])

onde *Id* é obtido através de *nome(Id)* e *VId* é obtido através de *venda(VId, _, _, _)*.

5.3.8 Holon de Cliente

Um *Holon de Cliente* representa a visão que a empresa tem de um seu cliente [Sousa *et al.*, 2000a]. Este holon responde pelos padrões do cliente em termos de exigências e gostos, possibilitando acções de *marketing* dirigido. Idealmente, holons deste tipo comunicariam com holons representantes do cliente, estendendo assim a cadeia de negócio para o exterior da empresa. Estes holons gerem toda a informação do cliente (*e.g.*, classificação ABC, compras efectuadas).

Este holon tem por objectivo servir os outros holons do sistema com informação relativa a um cliente. Um Holon de Cliente é criado (dando início ao seu ciclo de vida) quando se efectua o primeiro contacto com o cliente (aceitando uma encomenda ou pedido de orçamento, ou enviando uma proposta de fornecimento), existindo até ao momento em que se desiste de ter negócio com aquele cliente (*e.g.*, por violação de acordo) ou o cliente deixou de efectuar negócios com a empresa. Durante a sua existência este holon gere a informação relativa ao cliente.

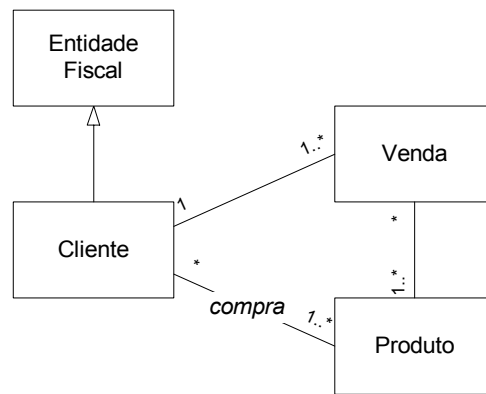


Figura 5.16 – Diagrama de classes para holons de cliente

A Figura 5.16 apresenta o diagrama de classes para um Holon de Cliente, que é representado na sua base de conhecimento pelas extensões dos seguintes predicados:

entidade-fiscal(NIF, Designação, Morada, Contactos)
cliente(CId, Data-Criação, Data-Ultima-Venda)
atributo(Parâmetro, Valor)

O predicado *entidade-fiscal* tem um significado idêntico ao já descrito para o Holon de Fornecedor. Os predicados *atributo* indicam o valor dos vários atributos do cliente (e.g., montante em dívida, limite de crédito).

A base de conhecimento destes holons também contém o seguinte predicado:

tipo(tipo_cliente)

A identificação deste tipo de holon é dada pelo prefixo “*id_cliente_*” (constante do sistema – vide Tabela 5.4) concatenado com o código de cliente. Os holons deste tipo registam-se no serviço de directório do sistema com uma mensagem com o conteúdo:

registar(Id, [*cliente*(CId)])

onde *Id* é obtido através de *nome*(Id) e *CId* é obtido através de *cliente*(CId, _, _).

5.4 Tratamento de Informação Incompleta

Na área da produção existem várias situações onde a informação necessária pelo sistema não está disponível na sua plenitude (e.g., uma encomenda de um cliente que não especifica completamente os atributos dos produtos, como por exemplo, a cor). Em vez de considerar esta informação como incorrecta e ignorá-la, o sistema pode utilizá-la para guiar as suas decisões dado o facto da informação ser de confiança (i.e., o cliente vai completar a encomenda mais tarde no tempo).

Embora esta informação não esteja completamente definida, tornando assim impossível a sua utilização, o facto de ela existir no sistema é mais útil do que a sua não existência. Por exemplo, num cenário de escalonamento de produção (situação em que embora não seja possível atribuir os recursos para a realização de uma encomenda em particular), é possível efectuar uma estimativa baseada em informação estatística do histórico do cliente e dos produtos.

5.4.1 Representação Utilizada

Nesta secção alude-se à representação utilizada para a manipulação de informação incompleta, de acordo com os tipos identificados na secção 4.4.2.2, nomeadamente: Nulos; Informação mutuamente exclusiva; e Informação não permitida. Esta representação utiliza como linguagem computacional a linguagem de programação em lógica PROLOG.

5.4.1.1 Informação Negativa Explícita

Pretende-se contornar o Pressuposto do Mundo Fechado (PMF) e assim permitir a inclusão de nulos do tipo “desconhecido” nos resultados do processo de inferência [Traylor e Gelfond, 1993] [Analide e Neves, 1996] [Sousa *et al.*, 2000b]. Por exemplo, quando é sabido peremptoriamente que um determinado holon não contempla um certo número de funcionalidades.

De um ponto de vista abstracto pode-se indicar explicitamente a falsidade de um qualquer predicado P da seguinte forma:

$$\neg P$$

Em termos de implementação optou-se pela utilização de um meta-predicado, *negacao*, cujo único argumento é o termo p a negar. Ou seja:

$$negacao(p(a_0, \dots, a_{n-1}))$$

em que a_i ($0 \leq i < n$, em que n denota o número de argumentos de p) responde pelo atributo de ordem i de p .

5.4.1.2 Nulos do Tipo Desconhecido

Um *nulo do tipo desconhecido* representa a falta de um ou mais itens de informação [Traylor e Gelfond, 1993] [Analide e Neves, 1996] [Sousa *et al.*, 2000b]. Por exemplo, a existência de uma encomenda que não foi totalmente especificada (*e.g.*, não se sabe quem encomendou 20 camisolas referência 304).

De um ponto de vista abstracto pode-se acrescentar informação incompleta à extensão de um qualquer predicado P da seguinte forma:

$$\begin{aligned} & p(a_0, \dots, \omega, \dots, a_{n-1}) \\ & \text{nulo}(\omega) \end{aligned} \tag{5.1}$$

$$\text{excepção_nulo}(p(A_0, \dots, _, \dots, A_{n-1})) \leftarrow p(A_0, \dots, W, \dots, A_{n-1}) \wedge \text{nulo}(W) \tag{5.2}$$

em que a_i ($0 \leq i < n$, em que n denota o número de argumentos de p) responde pelo atributo de ordem i de p , ω denota um valor desconhecido, sendo A_i e W variáveis que falam, respectivamente, por informação completa e incompleta.

A notação anterior pode ser simplificada, no caso da existência de apenas um argumento do tipo desconhecido, caso em que a axiomática pode ser abreviada, substituindo (5.1) e (5.2) por (5.3):

$$\text{excepção_nulo}(p(A_0, \dots, _, \dots, A_{n-1})) \leftarrow p(A_0, \dots, \omega, \dots, A_{n-1}) \tag{5.3}$$

5.4.1.3 Nulos do Tipo Desconhecido de um Conjunto de Valores

Os *nulos do tipo desconhecido de um conjunto de valores* falam por informação para descrever situações para as quais se posicionam diversas alternativas (mutuamente exclusivas) [Traylor e Gelfond, 1993] [Analide e Neves, 1996] [Sousa *et al.*, 2000b]. Por exemplo, uma encomenda de um cliente que ainda não se decidiu sobre qual de dois tipos de acabamento a utilizar.

De um ponto de vista abstracto, na extensão de um predicado P , esta situação pode ser dada na forma:

$$\begin{aligned} & \text{excepção_set}(p(a_{0,0}, a_{0,1}, \dots, a_{0,n-2}, a_{0,n-1})) \\ & \text{excepção_set}(p(a_{1,0}, a_{1,1}, \dots, a_{1,n-2}, a_{1,n-1})) \\ & \dots \\ & \text{excepção_set}(p(a_{k-2,0}, a_{k-2,1}, \dots, a_{k-2,n-2}, a_{k-2,n-1})) \\ & \text{excepção_set}(p(a_{k-1,0}, a_{k-1,1}, \dots, a_{k-1,n-2}, a_{k-1,n-1})) \end{aligned}$$

em que $a_{j,i}$ ($0 \leq i < n$, $0 \leq j < k$, sendo n o número de argumentos do predicado e k o número de opções da concretização do predicado), representa um argumento conhecido para uma das opções de concretização do predicado.

5.4.1.4 Nulos do Tipo Não Permitido

Este tipo de nulos representam situações não permitidas na Base de Conhecimento do holon [Traylor e Gelfond, 1993] [Analide e Neves, 1996] [Sousa *et al.*, 2000b] (*e.g.*, a impossibilidade de contratar pessoas com idade inferior ao limite legal).

De um ponto de vista abstracto a extensão de um predicado P pode denotar um nulo deste tipo utilizando uma das seguintes formas:

$$nulo_n_permitido(p(A_0, ..., \delta, ..., A_{n-1}))$$

$$nulo_n_permitido(p(A_0, ..., A_{n-1})) \leftarrow C_0 \wedge ... \wedge C_{m-1}$$

em que A_i ($0 \leq i < n$, sendo n o número de argumentos do predicado P) é uma variável para o argumento de ordem i de P e δ representa o valor não permitido para um dado argumento. Os predicados C_j ($0 \leq j < m$, sendo m o número de predicados de condição) representam condições necessárias para a ‘não permissão’.

5.4.1.5 Meta-Interpretador

Tendo como base o algoritmo para o meta-interpretador apresentado na Figura 4.5 e de acordo com a notação descrita nas secções anteriores, o meta-interpretador para tratar informação incompleta é descrito pelas seguintes produções [Sousa *et al.*, 2000b].

$$\begin{aligned} demo(P, verdadeiro) &\leftarrow \\ &\quad P \wedge \\ &\quad \text{não excepção_nulo}(P) \\ demo(P, desconhecido) &\leftarrow \\ &\quad \text{excepção_nulo}(P) \\ demo(P, desconhecido) &\leftarrow \\ &\quad \text{excepção_set}(P) \\ demo(P, nao_permitido) &\leftarrow \\ &\quad nulo_n_permitido(P) \\ demo(P, falso) &\leftarrow \\ &\quad \neg P \wedge \\ &\quad \text{não excepção_set}(P) \\ demo(P, V) &\leftarrow \\ &\quad \text{tipo}(T) \wedge \\ &\quad demo(T, P, V) \\ demo(_, desconhecido) & \end{aligned}$$

Para garantir a consistência da base de conhecimento é necessário que ao inserir novos termos se verifique a regra (5.4).

$$pode_inserir(P) \leftarrow demo(P, T) \wedge T \neq nao_permitido \quad (5.4)$$

Este conjunto de predicados (meta-interpretador e regra de consistência) passa a fazer parte do arquétipo dos holons, permitindo assim a todos os holons inferir sobre informação incompleta na sua base de conhecimento.

5.4.2 Casos Identificados

Esta secção apresenta os vários casos identificados para cada holon ou tipo de holon onde podem existir situações de informação incompleta. Os casos são enumerados e descritos, sendo também apresentados os predicados a acrescentar à base de conhecimento de cada holon para o tratamento de cada caso. Sempre que possível, cada caso é apresentado com exemplos práticos.

Nas tabelas das subsecções seguintes a coluna “Tipo de Nulo” pode conter um de seis valores:

- NE – negação explícita;
- PMF – pressuposto do mundo fechado (via negação explícita);
- MA – mundo aberto, *i.e.*, não utilização do PMF sendo desconhecido aquilo que não se conseguir provar;
- ND – nulo do tipo desconhecido;
- NDCV – nulo do tipo desconhecido de um conjunto de valores;
- NP – nulo do tipo não permitido.

5.4.2.1 Arquétipo dos Holon

A Tabela 5.5 apresenta os casos identificados de informação incompleta para os predicados correspondentes à base de conhecimento de todos os holons.

Tabela 5.5 – Informação incompleta no arquétipo dos holons

Caso	Descrição	Tipo de Nulo	Predicado
1	Cada holon só tem um nome	PMF e NP	<i>nome</i>
2	Cada holon conhece todos os tipos a que pertence	PMF	<i>tipo</i>
3	Cada holon conhece todas as relações que mantém com outros holons	PMF	<i>relação</i>
4	Cada holon conhece todas as holarquias a que pertence	PMF	<i>holarquia</i>

A base de conhecimento de todos os holons deve então ser aumentada com as seguintes produções:

$$\neg \text{nome}(I, D) \leftarrow \text{não nome}(I, D)$$

$$\begin{aligned}
 \text{nulo_n_permitido}(\text{nome}(_, _)) \leftarrow & \\
 & \text{todas_as_soluções}(\text{nome}(I, D), \text{nome}(I, D), L) \wedge \\
 & \text{comprimento}(L, N) \wedge \\
 & N > 1 \\
 \neg \text{tipo}(T) \leftarrow & \\
 & \text{não tipo}(T) \\
 \neg \text{relação}(H, R, V, P) \leftarrow & \\
 & \text{não relação}(H, R, V, P) \\
 \neg \text{holarquia}(H, L) \leftarrow & \\
 & \text{não holarquia}(H, L) \\
 \neg \text{todo}(H) \leftarrow & \\
 & \text{não todo}(H) \\
 \neg \text{parte_de}(H) \leftarrow & \\
 & \text{não parte_de}(H)
 \end{aligned} \tag{5.5}$$

Cada holon tem conhecimento completo de si próprio, daí o PMF aplicado aos predicados *nome*, *relação* e *holarquia* (e consequentemente também aos predicados *todo* e *parte_de*). A identificação de um holon representa a sua identidade única no sistema, pelo que um holon só deve ter uma identificação. Na regra (5.5) o termo $N > 1$ indica que o holon pode não ter identificação ou apenas uma identificação. A situação de um holon sem identificação é permitida pois os holons criados dinamicamente podem obter o seu “nome” apenas após a criação (*vide* o caso dos Holons de Tarefa).

5.4.2.2 Holon de Serviços de Directório

A Tabela 5.6 apresenta os casos identificados de informação incompleta para o holon de serviços de directório.

Tabela 5.6 – Informação incompleta no holon de serviços de directório

Caso	Descrição	Tipo de Nulo	Predicado
5	Pode desconhecer-se quem faz determinada habilidade	MA	<i>serviços</i>
6	Pode não se conhecer todas as habilidades de um holon	MA	<i>serviços</i>
7	Pode conhecer-se que determinado holon não faz uma determinada habilidade	NE	<i>fornece</i>

A base de conhecimento deste holon pode então ser aumentada com a seguinte produção:

$$\neg \text{fornece}(\text{Holon}, \text{Habilidade})$$

Embora por definição/especificação todos os holons se registem no serviço de directório, a base de conhecimento deste holon não é fechada (*i.e.*, não se aplica o PMF aos axiomas *serviços*,

fornece e *fornecedores*) pois um holon pode ainda não ter actualizado todas as suas habilidades no directório e a base de conhecimento do holon de directório estar desactualizada.

Supondo a seguinte base de conhecimento para o Holon de Serviços de Directório:

serviços(id_recurso_1, [fresar, furar_Z])
serviços(id_recurso_3, [facejar])
 \neg *fornece(id_recurso_2, furar_Z)*

O valor de verdade da questão *fornece(id_recurso_1, furar_angulo)* seria *desconhecido*, enquanto que o de *fornece(id_recurso_1, furar_Z)* seria *verdadeiro*. Por outro lado, *fornece(id_recurso_2, furar_Z)* seria *falso*, mas *fornece(id_recurso_2, facejar)* seria *desconhecido*.

5.4.2.3 Holons de Produto

A Tabela 5.7 apresenta os casos identificados de informação incompleta para os holons de produto.

Tabela 5.7 – Informação incompleta nos holons de produto

Caso	Descrição	Tipo de Nulo	Predicado
8	Todos os planos do produto são conhecidos	PMF	<i>plano</i>
9	Toda a composição do produto é conhecida	PMF	<i>composição</i>
10	A quantidade em existência pode ser desconhecida	ND	<i>produto</i>
11	O valor de um atributo pode ser desconhecido	MA	<i>atributo</i>
12	A duração de uma operação pode ser desconhecida	ND	<i>plano</i>

A base de conhecimento deste tipo de holons deve então ser aumentada com as seguintes produções:

\neg *plano(I, C, A, O)* \leftarrow
 não plano(I, C, A, O)
 \neg *composição(P, C, Q)* \leftarrow
 não composição(P, C, Q)
excepção_nulo(produto(I, D, _)) \leftarrow
 produto(I, D, alguma_quantidade)
excepção_nulo(plano(I, C, A, Op1)) \leftarrow
 plano(I, C, A, Op2) \wedge
 nulo_plano(Op1, Op2)
nulo(algum_tempo)
nulo(alguns_segundos)

```

nulo(alguns_minutos)
nulo(algumas_horas)
nulo_plano( $[X \mid T], [X \mid T2]$ )  $\leftarrow$ 
    nulo_plano( $T, T2$ )
nulo_plano( $[nodo(I, H, E, \_, P, S) \mid T], [nodo(I, H, E, Dur, P, S) \mid T2]$ )  $\leftarrow$ 
    nulo(Dur)
nulo_plano( $\_, \_$ )  $\leftarrow$ 
    falso
    
```

Como a informação do produto é de natureza técnica, os planos de produção e a sua composição estão devidamente especificados, pelo que se utiliza o PMF para esses axiomas. A quantidade em existência nem sempre é conhecida (ou muitas vezes está desactualizada) devido a falhas (ou inexistência) do sistema de controlo de existências (manual ou automático) pelo que se representa um nulo do tipo desconhecido (*alguma_quantidade*) para o axioma *produto*. Determinados atributos não técnicos do produto (*e.g.*, classificação ABC) poderão ser desconhecidos pelo que não se aplica o PMF a estes axiomas. Embora o plano contenha informação técnica, é usual desconhecer a duração exacta de uma operação, devido à existência de vários recursos capazes de efectuar essa operação, pelo que a duração de uma operação pode ser representada com valores nulos (*algum_tempo*). Optou-se pela utilização de quatro nulos que permitem de alguma forma quantificar a duração (*alguns_segundos*, *alguns_minutos*, *algumas_horas*), dessa forma, o nulo contém alguma informação que pode ser útil para o utilizador.

5.4.2.4 Holons de Tarefa

A Tabela 5.8 apresenta os casos identificados de informação incompleta para os holons de tarefa.

Tabela 5.8 – Informação incompleta nos holons de tarefa

Caso	Descrição	Tipo de Nulo	Predicado
13	Todas as origens estão especificadas	PMF	<i>origem</i>
14	O plano está especificado	PMF	<i>plano</i>
15	A duração de uma operação do plano pode ser desconhecida	ND	<i>plano</i>
16	O valor de um atributo pode ser desconhecido	MA	<i>atributo</i>

A base de conhecimento deste tipo de holons deve então ser aumentada com as seguintes produções:

```

¬origem(I, Q) ←
    não origem(I, Q)
¬plano(I, C, A, O) ←
    não plano(I, C, A, O)
nulo(algum_tempo)
nulo(alguns_segundos)
nulo(alguns_minutos)
nulo(algumas_horas)
excepção_nulo(plano(I, C, A, Op1)) ←
    plano(I, C, A, Op2) ∧
    nulo_plano(Op1, Op2)
nulo_plano([X | T], [X | T2]) ←
    nulo_plano(T, T2)
nulo_plano([nodo(I, H, E, _, P, S) | T], [nodo(I, H, E, Dur, P, S) | T2]) ←
    nulo(Dur)
nulo_plano(_, _) ←
    falso

```

O holon de tarefa é gerado para cumprimento de uma ordem de fabrico, originada por encomendas de clientes ou produção para existência, além disso, o holon de tarefa instancia um plano de produção para o cumprimento dessa ordem de fabrico, pelo que se aplica o PMF aos predicados *origem* e *plano*. Não se aplica o PMF aos predicados *atributo* porque é possível desconhecer o valor de alguns atributos da tarefa (e.g., duração).

5.4.2.5 Holons de Recurso

A Tabela 5.9 apresenta os casos identificados de informação incompleta para os holons de recurso.

Tabela 5.9 – Informação incompleta nos holons de recurso

Caso	Descrição	Tipo de Nulo	Predicado
17	Todas as habilidades estão especificadas	PMF	<i>habilidade</i>
18	Todas as actividades estão especificadas	PMF	<i>actividade</i>
19	O estado de uma actividade pode ser desconhecido	ND	<i>actividade</i>
20	Pode não se saber quem são os recursos antecessores	ND	<i>actividade</i>
21	Pode não se saber quem são os recursos sucessores	ND	<i>actividade</i>

A base de conhecimento deste tipo de holons deve então ser aumentada com as seguintes produções:

```

    ¬habilidade(I, D, P, A) ←
        não_habilidade(I, D, P, A)
    ¬actividade(T, O, Q, I, D, Dt, E, P, S) ←
        não_actividade(T, O, Q, I, D, Dt, E, P, S)
    excepção_nulo(actividade(T, O, Q, I, D, Dt, _, P, S)) ←
        actividade(T, O, Q, I, D, Dt, algum_estado, P, S)
    excepção_nulo(actividade(T, O, Q, I, D, Dt, E, Pr1, S)) ←
        actividade(T, O, Q, I, D, Dt, E, Pr2, S) ∧
        nulo_pred_succ(Pr1, Pr2)
    excepção_nulo(actividade(T, O, Q, I, D, Dt, E, P, Su1)) ←
        actividade(T, O, Q, I, D, Dt, E, P, Su2) ∧
        nulo_pred_succ(Su1, Su2)
    excepção_nulo(actividade(T, O, Q, I, D, Dt, E, Pr1, Su1)) ←
        actividade(T, O, Q, I, D, Dt, E, Pr2, Su2) ∧
        nulo_pred_succ(Pr1, Pr2) ∧
        nulo_pred_succ(Su1, Su2)
    nulo_pred_succ([X | T], [X | T2]) ←
        nulo_pred_succ(T, T2)
    nulo_pred_succ([_ | T], [algum_holon | T2])
    nulo_pred_succ(_, _) ←
        falso
    
```

O holon tem total conhecimento de si próprio, daí a necessidade de utilizar o PMF para os predicados *habilidade* e *actividade*. É no entanto possível desconhecer algumas características das actividades do holon, nomeadamente, o estado de uma actividade e algum dos recursos contactados para operações antecessoras e sucessoras. Nessas situações utilizam-se os dois nulos *algum_estado* e *algum_holon* respectivamente.

5.4.2.6 Holons de Compra

A Tabela 5.10 apresenta os casos identificados de informação incompleta para os Holons de Compra.

Tabela 5.10 – Informação incompleta nos holons de compra

Caso	Descrição	Tipo de Nulo	Predicado
22	Todos os pedidos de orçamento estão representadas no sistema	PMF	<i>pedido-orçamento</i>
23	Todos os orçamentos estão representadas no sistema	PMF	<i>orçamento</i>

Caso	Descrição	Tipo de Nulo	Predicado
24	Todas as encomendas estão representadas no sistema	PMF	<i>encomenda</i>
25	Todas as guias de remessa estão representadas no sistema	PMF	<i>guia-remessa</i>
26	Todas as facturas estão representadas no sistema	PMF	<i>factura</i>
27	Todos os recibos estão representadas no sistema	PMF	<i>recibo</i>

A base de conhecimento deste tipo de holons deve então ser aumentada com as seguintes produções:

$\neg \text{pedido-orçamento}(NI, F, N, Dt, P, C, A) \leftarrow$
 $\quad \text{não pedido-orçamento}(NI, F, N, Dt, P, C, A)$
 $\neg \text{orçamento}(NI, F, N, D, P, C, A) \leftarrow$
 $\quad \text{não orçamento}(NI, F, N, D, P, C, A)$
 $\neg \text{encomenda}(NI, F, N, D1, D2, P, C, A) \leftarrow$
 $\quad \text{não encomenda}(NI, F, N, D1, D2, P, C, A)$
 $\neg \text{guia-remessa}(NI, F, N, D, P, C, A) \leftarrow$
 $\quad \text{não guia-remessa}(NI, F, N, D, P, C, A)$
 $\neg \text{factura}(NI, F, N, D, P, C, A) \leftarrow$
 $\quad \text{não factura}(NI, F, N, D, P, C, A)$
 $\neg \text{recibo}(NI, F, N, D, NDF, C, A) \leftarrow$
 $\quad \text{não recibo}(NI, F, N, D, NDF, C, A)$

Os vários predicados da base de conhecimento dos holons de compra representam os documentos de negócio associados ao processo de compras, sendo esse conhecimento fechado, ou seja, se um documento não está descrito no sistema é porque não existe.

5.4.2.7 Holons de Fornecedor

A Tabela 5.11 apresenta os casos identificados de informação incompleta para os Holons de Fornecedor.

Tabela 5.11 – Informação incompleta nos holons de fornecedor

Caso	Descrição	Tipo de Nulo	Predicado
28	A data do primeiro contacto com o fornecedor pode ser desconhecida	ND	<i>fornecedor</i>

Caso	Descrição	Tipo de Nulo	Predicado
29	A data do último contacto com o fornecedor pode ser desconhecida	ND	<i>fornecedor</i>
30	Os atributos de um fornecedor podem não estar todos especificados	MA	<i>atributo</i>
31	O fornecedor pode fornecer componentes dos quais (ainda) não há conhecimento	MA	<i>fornece</i>

A base de conhecimento deste tipo de holons deve então ser aumentada com as seguintes produções:

$$\begin{aligned} \text{excepção_nulo}(\text{fornecedor}(I, _, DU, A)) &\leftarrow \\ &\quad \text{fornecedor}(I, \text{alguma_data}, DU, A) \\ \text{excepção_nulo}(\text{fornecedor}(I, DI, _, A)) &\leftarrow \\ &\quad \text{fornecedor}(I, DI, \text{alguma_data}, A) \end{aligned}$$

É possível que os atributos de um fornecedor não estejam todos especificados, pelo que não se fecha os predicados *atributo*. Os predicados *fornece* não são fechados pois é possível que o fornecedor ainda não tenha enviado o catálogo actualizado dos seus produtos.

5.4.2.8 Holon de Gestão de Compras

A Tabela 5.12 apresenta os casos identificados de informação incompleta para o Holon de Gestão de Compras.

Tabela 5.12 – Informação incompleta no holon de gestão de compras

Caso	Descrição	Tipo de Nulo	Predicado
32	Não se efectuam transacções comerciais com fornecedores oriundos de países com os quais não há relações diplomáticas	NP	<i>fornecedor</i>
33	Não se efectuam transacções comerciais com fornecedores que não sejam cumpridores	NP	—

A base de conhecimento deste holon deve então ser aumentada com as seguintes produções:

$$\text{nulo_n_permitido}(\text{entidade-fiscal}(N, \text{morada}(R, L, CP, Pais), Dsg, C)) \leftarrow \text{embargado}(Pais)$$

$$\text{nulo_n_permitido}(\text{entidade-fiscal}(N, M, Dsg, C)) \leftarrow \text{lista_negra}(Dsg, R, Dt)$$

Para o correcto funcionamento dos axiomas anteriores, pressupõem-se que na Base de Conhecimento do Holon de Gestão de Compras existem produções dos seguintes modelos:

$$\text{embargado}(P) \\ \text{lista_negra}(Dsg, Razão, Data)$$

Onde, *embargado* especifica um país com o qual existe um embargo comercial ditado pelo governo do país onde a empresa tem a sua sede social e *lista_negra* especifica os fornecedores considerados *persona non grata* pela empresa, *e.g.*, fornecedores que anteriormente não cumpriram os prazos de entrega provocando assim prejuízos avultados à empresa.

Por exemplo, uma empresa dos EUA poderia ter na base de conhecimento do seu Holon de Gestão de Compras os seguintes axiomas:

$$\text{embargado}(\text{cuba}) \\ \text{lista_negra}(\text{'Smith \& Sons, Inc.'}, \text{incumprimento_de_prazo}, \text{data}(1997, 1, 27)) \\ \text{lista_negra}(\text{'McGregor \& McAlister'}, \text{qualidade_insuficiente}, \text{data}(1999, 8, 12))$$

Devido ao embargo a Cuba, as transações comerciais com fornecedores desse país não são permitidas. Por outro lado, determinados fornecedores com os quais a empresa efectuou transações no passado pertencem agora à “lista negra” por critérios que a empresa definiu.

5.4.2.9 Holons de Venda

A Tabela 5.13 apresenta os casos identificados de informação incompleta para os Holons de Venda.

Tabela 5.13 – Informação incompleta nos holons de venda

Caso	Descrição	Tipo de Nulo	Predicado
34	Todas os pedidos de orçamento estão representadas no sistema	PMF	<i>pedido-orçamento</i>
35	Todas os orçamentos estão representadas no sistema	PMF	<i>orçamento</i>
36	Todas as encomendas estão representadas no sistema	PMF	<i>encomenda</i>
37	Todas as guias de remessa estão representadas no sistema	PMF	<i>guia-remessa</i>

Caso	Descrição	Tipo de Nulo	Predicado
38	Todas as facturas estão representadas no sistema	PMF	<i>factura</i>
39	Todas os recibos estão representadas no sistema	PMF	<i>recibo</i>
40	Desconhece-se o cliente que efectuou a encomenda	ND e NDCV	<i>venda</i>
41	O cliente ainda não decidiu entre vários produtos do catálogo qual vai encomendar	NDCV	<i>encomenda</i>

A base de conhecimento deste tipo de holons deve então ser aumentada com as seguintes produções:

```

¬pedido-orçamento(N, D1, P, C, A) ←
    não pedido-orçamento(N, D1, P, C, A)
¬orçamento(N, D, P1, P2, C, A) ←
    não orçamento(N, D, P1, P2, C, A)
¬encomenda(N, D1, D2, P, C, A) ←
    não encomenda(N, D1, D2, P, C, A)
¬guia-remessa(N, D, P, C, A) ←
    não guia-remessa(N, D, P, C, A)
¬factura(N, D, P, C, A) ←
    não factura(N, D, P, C, A)
¬recibo(N, D, NDF, C, A) ←
    não recibo(N, D, NDF, C, A)
excepção_nulo(venda(I, _, D, A) ←
    venda(I, alguém, D, A)

```

Na base de conhecimento dos Holons de Venda, os vários predicados representam os documentos de negócio associados ao processo de venda, sendo esse conhecimento fechado, ou seja, se um documento não está descrito no sistema é porque não existe.

O caso 40 é mais comum do que se supõe à primeira vista, principalmente em pequenas e médias empresas onde as encomendas são muitas vezes efectuadas verbalmente ao funcionário responsável pelo armazém. É possível que o funcionário que recebeu a encomenda se lembre do que lhe foi pedido mas não sabe ao certo quem lhe pediu:

```

venda(3345, alguém, data(2000, 8, 20), [])
encomenda(2003, data(2000, 8, 20), data(2000, 9, 15), [(25, 300, [])], [], [])

```

Ou então esteja em dúvida entre dois clientes:

```

excepção_set(venda(3234, id_cliente_450, data(2000, 1, 23)))
excepção_set(venda(3234, id_cliente_789, data(2000, 1, 23)))
encomenda(2079, data(2000, 1, 23), data(2000, 2, 30), [(7786, 100, [])], [], [])

```

O caso 41 aplica-se normalmente à escolha dos acabamentos dos produtos, por exemplo, o cliente sabe que deseja 1000 camisolas de gola alta mas não se decidiu quanto à cores e padrões a encomendar.

5.4.2.10 Holons de Cliente

A Tabela 5.14 apresenta os casos identificados de informação incompleta para os Holons de Cliente.

Tabela 5.14 – Informação incompleta nos holons de cliente

Caso	Descrição	Tipo de Nulo	Predicado
42	A data do primeiro contacto com o cliente pode ser desconhecida	ND	<i>cliente</i>
43	A data do último contacto com o cliente pode ser desconhecida	ND	<i>cliente</i>
44	Os atributos de um cliente podem não estar todos especificados	MA	<i>atributo</i>

A base de conhecimento deste tipo de holons deve então ser aumentada com as seguintes produções:

$$\begin{aligned} \text{excepção_nulo}(\text{cliente}(I, _, DU, A)) &\leftarrow \\ &\quad \text{cliente}(I, \text{alguma_data}, DU, A) \\ \text{excepção_nulo}(\text{cliente}(I, DI, _, A)) &\leftarrow \\ &\quad \text{cliente}(I, DI, \text{alguma_data}, A) \end{aligned}$$

Quer o caso 42, quer o caso 43 acontecem normalmente em situações de transição de um sistema para outro. Por exemplo, o cliente é um cliente já antigo e no sistema informático anterior não existia informação sobre a data de início de contacto, não existindo ninguém na empresa que saiba ao certo responder a essa pergunta.

5.4.2.11 Holon de Gestão de Vendas

A Tabela 5.15 apresenta os casos identificados de informação incompleta para o Holon de Gestão de Vendas.

Tabela 5.15 – Informação incompleta no holon de gestão de vendas

Caso	Descrição	Tipo de Nulo	Predicado
45	Não se efectuam transacções comerciais com clientes de países com os quais não existem relações diplomáticas	NP	<i>cliente</i>
46	Não se efectuam transacções comerciais com clientes que estejam na “lista negra”	NP	–

A base de conhecimento deste holon deve então ser aumentada com as seguintes produções:

$$\begin{aligned} & nulo_n_permitido(entidade-fiscal(N, morada(R, L, CP, P), D, C)) \leftarrow \\ & \quad embargado(P) \\ & nulo_n_permitido(entidade-fiscal(N, M, Dsg, C)) \leftarrow \\ & \quad lista_negra(Dsg, R, Dt) \end{aligned}$$

À semelhança do que foi dito para o Holon de Gestão de Compras, a Base de Conhecimento do Holon de Gestão de Vendas deve conter produções dos seguintes modelos:

$$\begin{aligned} & embargado(P) \\ & lista_negra(Dsg, Razão, Data) \end{aligned}$$

Onde *embargado* e *lista_negra* têm o significado anteriormente apresentado no Holon de Gestão de Compras.

5.5 Resumo do Capítulo

Nesta secção foi apresentada a arquitectura proposta, tendo por base o conceito holónico descrito no Capítulo 4. Assim, começou-se por apresentar um cenário de utilização com base no que foi descrito sobre as tendências de produção na secção 3.4.2 “O ‘Futuro’ da Produção”.

Em seguida foi efectuada uma análise funcional da produção, identificando os principais casos de utilização, que foram posteriormente detalhados. Após a análise funcional, identificaram-se quais as principais entidades do sistema e os seus relacionamentos básicos: *Fornecedor*, *Compra*, *Produto*, *Cliente*, *Venda*, *Tarefa* e *Recurso*.

Na secção 5.2.1 foi então apresentada a arquitectura holónica proposta, constituída por holarquias para as funções de *Projecto*, *Planeamento de Processos*, *Planeamento da Produção*, *Escalonamento*, *Compras* e *Vendas*. Estas holarquias são constituídas por holons que representam os Produtos, Recursos, Tarefas, Fornecedores, Clientes, Compras e Vendas. Na secção seguinte, os vários holons constituintes de cada holarquia foram especificados em termos de base de conhecimento, ciclo de vida, objectivos e identificação.

Na área da produção, existem várias situações onde toda a informação necessária pelo sistema não está disponível. Por esse motivo, após a apresentação de uma notação para representação de informação incompleta, a especificação da base de conhecimento de cada holon da arquitectura proposta foi aumentada com axiomas que permitem representar essas situações ■

CAPÍTULO 6

MODELO PROPOSTO: OPERAÇÃO



Quanto mais quero uma coisa feita, menos a considero como trabalho

Richard Bach, “Ilusões: As aventuras de um Messias relutante”.

No capítulo anterior foi proposta uma arquitectura holónica para sistemas de produção, de acordo com o que foi estipulado no Capítulo 3 sobre os sistemas de produção do futuro. Deste modo, partindo do conceito holónico e das ferramentas de prototipagem apresentadas no Capítulo 4, cada elemento (*i.e.*, holon) da arquitectura foi caracterizado em termos de objectivos, ciclo de vida e base de conhecimento. Neste capítulo esses holons continuarão a ser objecto de estudo mas agora sob um ponto de vista operacional.

Devido à natureza distribuída da arquitectura apresentada na secção 5.2 torna-se necessário regulamentar a interacção entre os vários holons, especialmente os Holons de Tarefa e de Recursos, na actividade de escalonamento (escolhida como caso de teste para o modelo proposto e respectiva arquitectura). Por esse motivo passar-se-á à especificação de um protocolo de negociação que está preparado para o tratamento de excepções e principalmente para evitar o *Problema de Indecisão*. É também efectuado um estudo da complexidade do protocolo. Em seguida serão apresentados os principais algoritmos de funcionamento dos holons relacionados com o processo de escalonamento de tarefas. Completada a especificação do sistema, será então

apresentado o protótipo *Fabricare*. Seguidamente é efectuada uma comparação entre o trabalho desenvolvido e os trabalhos de referência apresentados no estado da arte.

6.1 Introdução

De acordo com o que foi referido nos objectivos deste trabalho, o caso de teste escolhido para validar o modelo computacional e implementar o protótipo foi o de Escalonamento de Tarefas Industriais. Desse modo, convém abordar algumas questões de escalonamento, completando assim o que foi dito na secção 2.4.5 “Escalonamento e Balanceamento” sobre esta actividade de produção.

A Figura 6.1 apresenta um gráfico de Gantt respeitante ao escalonamento de uma ordem de fabrico com quatro operações sequenciais com o intuito de se produzirem três itens de um produto (aqui referenciados por rectângulos com linhas obliquas). Na figura são também visíveis algumas operações já escalonadas em cada recurso (aqui referenciadas por rectângulos ponteados) e a utilização de um mesmo recurso para mais que uma operação da tarefa (M1).

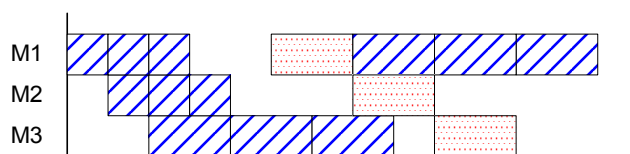


Figura 6.1 – Escalonamento

Este exemplo, e os seguintes, denotam uma situação por ventura simplificada, ao não se considerarem as operações de transporte entre máquinas (ou seja, assume-se que existe sempre transporte disponível de uma máquina para outra, estando a duração desse transporte reduzida a zero), não se considerando também o tempo para o setup da máquina para cada operação a executar (ou seja, assume-se que o setup tem duração zero).

Por outro lado, cada operação é iniciada logo a seguir ao fim da execução da operação antecessora; ou seja, no caso referido em epígrafe, não se espera que se complete o fabrico de cada um dos três produtos para executar a operação seguinte sendo, pelo contrário, cada produto tratado individualmente. Neste caso é então necessário tomar atenção às situações em que a operação que se segue tem uma duração menor que a operação antecessora (Figura 6.2). Conforme se pode observar na Figura 6.2a não é possível iniciar a execução das operações em M3 quando em M2 se terminar de processar um produto. Nestas situações é necessário deixar “buracos” no escalonamento correspondentes à diferença entre as durações das operações em presença, ou, atrasar o início das operações em M3, garantindo dessa forma que todos os produtos estão disponíveis quando M3 iniciar a laboração (Figura 6.2b).

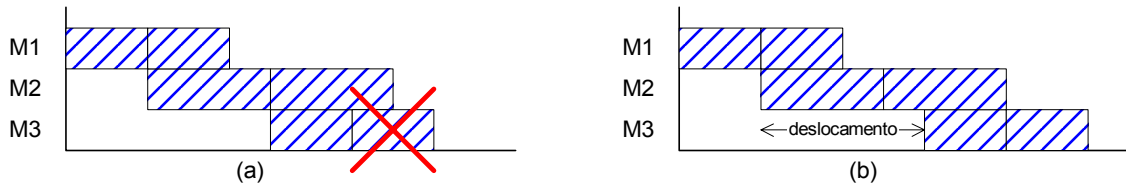


Figura 6.2 – Escalonamento com operações antecessoras de maior duração

O atraso (ou deslocamento) é calculado pela equação (6.1), onde $\delta(i)$ é uma função que retorna a duração da operação i .

$$\text{deslocamento} = \delta(i-1) + (\text{quantidade} - 1) \times (\delta(i-1) - \delta(i)) \quad (6.1)$$

Ao atrasar-se as operações em M3, surge então a necessidade da existência de *buffers* nas máquinas. Os *buffers* são espaços de armazenamento à entrada e/ou saída das máquinas onde é possível colocar o material ou produto a processar até que seja utilizado pelo recurso. Estes *buffers* têm obviamente uma limitação física quanto ao espaço de armazenamento mas, não raras vezes, para simplificar a compreensão do processo assumem-se *buffers* de tamanho infinito.

Um outro caso que pode surgir no processo de escalonamento é o da existência de recursos alternativos para a execução de uma mesma operação. Nesta situação, existirão várias combinações possíveis de recursos que podem ser utilizadas na execução da tarefa. A Figura 6.3 mostra uma situação onde uma operação pode ser feita em dois recursos diferentes (M1a e M1b) existindo, neste caso, apenas duas combinações possíveis (M1a – M2 – M3 e M1b – M2 – M3) gerando-se obviamente, escalonamentos diferentes para a tarefa (Figura 6.3a e Figura 6.3b respectivamente).

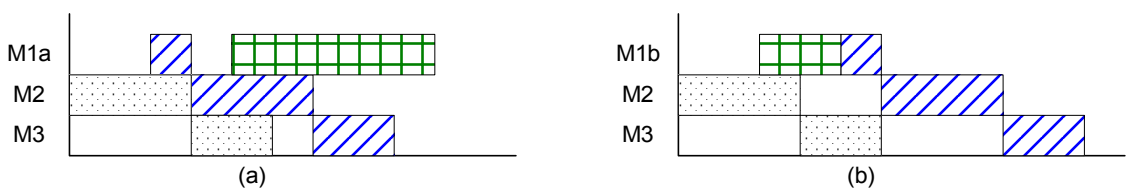


Figura 6.3 – Escalonamento com recursos alternativos

O último caso a considerar é o de poderem ocorrer operações paralelas num plano de produção. Conforme se pode ver na Figura 6.4, a operação *op2* pode ser escalonada em qualquer ponto daquele intervalo de tempo desde que termine antes do início da operação que a sucede; *i.e.*, *op4*. Este tipo de operações é vulgar em tarefas de montagem.

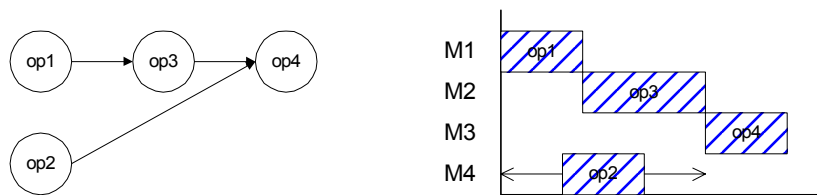


Figura 6.4 – Escalonamento com operações paralelas

6.2 Processo de Negociação

Esta secção descreve o processo utilizado para o escalonamento dinâmico de tarefas industriais, através de um protocolo de negociação entre os Holons de Tarefa e de Recurso, baseado no Protocolo de Rede de Contrato [Smith, 1980] [Davis e Smith, 1983]. Este protocolo inclui também uma fase de renegociação para tratamento de excepções e de alteração dinâmica de condições.

Os intervenientes no protocolo, denominado Protocolo de Rede de Contrato com Propagação de Restrições (PRCPR) [Sousa e Ramos, 1998] [Sousa e Ramos, 1999a] [Sousa *et al.*, 1999b], são os Holons de Tarefa e os Holons de Recurso, podendo no entanto ser entendidos como duas abstrações *Recurso* e *Tarefa* à semelhança dos papéis executados pelos agentes no protocolo de rede de contrato [Smith, 1980] [Davis e Smith, 1983]. Uma *Tarefa* representa um trabalho/projecto a ser executado podendo ser dividido em operações concorrentes ou sequenciais. Um *Recurso* é um dispositivo, uma pessoa ou entidade capaz de executar essas operações. Usando este modelo, o mesmo protocolo pode ser usado para escalonamento de ordens de fabrico numa instalação fabril, escalonamento de actividades de um projecto numa empresa virtual, ou gestão de uma cadeia de fornecimento de produtos sendo os fornecedores recursos e as tarefas as compras de produtos.

Para efectuar o escalonamento, o Holon de Tarefa vai negociar com os Holons de Recurso estabelecendo um contrato para a execução de cada operação do plano de produção do produto a que a tarefa se refere. O algoritmo de escalonamento propriamente dito é adaptado de um método centralizado, que considera agendas de recursos e comportamentos de escalonamento [Almeida, 1995] [Ramos *et al.*, 1995]. Cada holon de recurso possui uma agenda de actividades já contratadas e tem que escalonar as novas operações nos espaços livres, de acordo com a janela temporal definida pela tarefa. Devido à existência de restrições temporais de uma operação com as suas operações antecessoras e as suas sucessoras, os holons de recurso irão propagar as suas restrições de forma a eliminar intervalos de tempo que não respeitem a relação de precedência entre operações.

6.2.1 Descrição

O protocolo é definido pelos estados de [Sousa e Ramos, 1998] [Sousa e Ramos, 1999a] [Sousa *et al.*, 1999b]: (i) anúncio; (ii) requisição de serviços (iii) influência directa; (iv) influência inversa & oferta; e (v) contratação.

Em relação ao protocolo de rede de contrato original, foram acrescentadas as fases de influência directa e inversa. Estas novas fases correspondem à coordenação de intervalos temporais entre os vários recursos necessários à execução de cada operação da tarefa.

6.2.1.1 Anúncio de Tarefa

A Figura 6.5 (*adaptada de*: [Sousa *et al.*, 1999b]) apresenta as mensagens que são trocadas entre os holons na fase inicial do processo de negociação. Por uma questão de melhor visualização o componente Lançador de Tarefas do Holon de Escalonamento é apresentado em separado.

Em primeiro lugar, o componente *Lançador de Tarefas* do Holon de Escalonamento cria um novo *Holon de Tarefa* e comunica-lhe o que tem que fazer (mensagem *anúncio*). Em seguida, o Holon de Tarefa pede ao *Holon de Planeamento de Processos* um plano de produção para o item a fabricar. Por último, o Holon de Planeamento de Processos responde com o plano seleccionado.

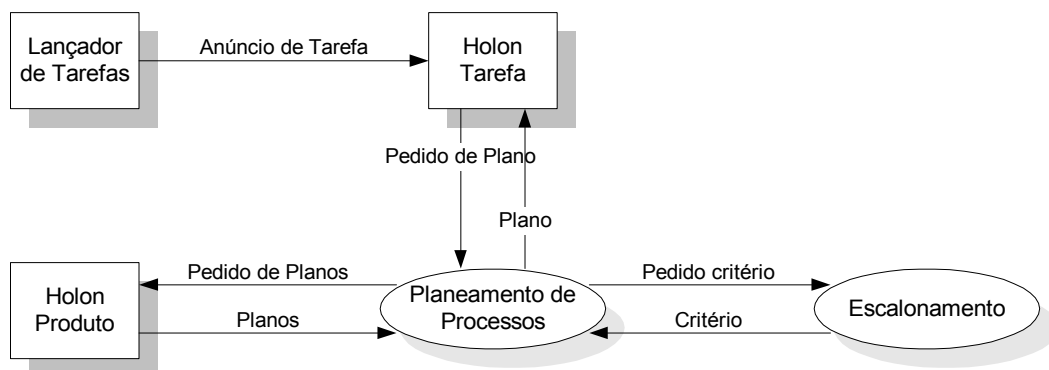


Figura 6.5 – Anúncio de tarefa

O *Holon de Planeamento de Processos* seleccionará de entre os vários planos que foram construídos para o produto, aquele que deve ser utilizado, baseando-se para isso na informação obtida do *Holon de Escalonamento* sobre as condições da instalação fabril (*e.g.*, o melhor plano é aquele que se adequa à minimização do tempo de fabrico, dado que a carga de trabalho está elevada).

Mensagem de Anúncio

O Lançador de Tarefas anuncia ao Holon de Tarefa o que fazer, com uma mensagem do seguinte teor:

anuncio(TId, NOF, PId, Quant, DataLimite, JanelaTemporal, Attr)

onde *TId*, *NOF*, *PId*, *Quant*, *DataLimite*, *Attr* e *JanelaTemporal* denotam respectivamente, a identificação atribuída ao holon de tarefa, o número de ordem de fabrico associada àquela tarefa, a identificação do produto a fabricar, a quantidade de produtos a fabricar, a data especificada para conclusão da tarefa, a lista de atributos da tarefa e, o intervalo de tempo em que o escalonamento faz sentido (o limite superior é atribuído com base na data limite da tarefa).

Mensagem de Pedido de Plano

O Holon de Tarefa requisita um plano ao Holon de Planeamento de Processos, enviando uma mensagem com o seguinte teor:

obter_plano(PId, Attr)

onde *PId* e *Attr* denotam respectivamente, a identificação do produto para o qual se requisita um plano e, a lista de atributos/restrições impostas a esse pedido. O termo *Attr* é usado para indicar ao Holon de Planeamento de Processos as características desejáveis (ou indesejáveis) no plano, permitindo assim que o Holon de Planeamento de Processos seleccione um plano de entre os planos alternativos, tendo em conta as necessidades impostas pela tarefa, (e.g., um plano que não utilize determinados recursos que de antemão se sabe estarem completamente ocupados). Algumas das instanciações possíveis para *Attr* podem tomar a forma: *nao_contem_operacoes(LOp)* que indica que se pretende um plano que não contempla as operações indicadas por *LOp*; *nao_contem_recursos(LRec)* que indica que se pretende um plano que não utilize os recursos indicados por *LRec*; *plano_usado(PlanId)* que indica que determinado plano já foi usado; e *usar_criterio(CritId)* que indica qual o critério de optimização que se pretende utilizar.

Mensagem de Resposta ao Pedido de Plano

O Holon de Planeamento de Processos responde ao Holon de Tarefa, enviando uma mensagem do seguinte teor:

r_obter_plano(PId, Attr, plano(Id, Id-Critério, Atributos-Plano, Operações))

onde *PId* e *Attr* denotam os valores passados na mensagem *obter_plano* e, o tuplo *plano/4* refere-se ao plano seleccionado para o fabrico do produto de acordo com a especificação dada.

6.2.1.2 Requisição de Serviços

Após o anúncio da tarefa, o *Holon de Tarefa* constrói a lista de recursos a contactar para cada operação do plano de fabrico e entra em contacto com esses *Holons de Recurso*, enviando-lhe uma mensagem *pedido* com informação sobre os recursos contactados para as operações a realizar imediatamente a montante e a jusante.

A Figura 6.6 (*adaptada de: [Sousa et al., 1999b]*) apresenta as mensagens trocadas numa situação em que se tem um produto que para ser fabricado necessita de três operações (*op1*, *op2* e *op3*) sequenciais. Neste exemplo considera-se que existem dois recursos (*A1* e *A2*) passíveis de efectuar a operação *op1*.

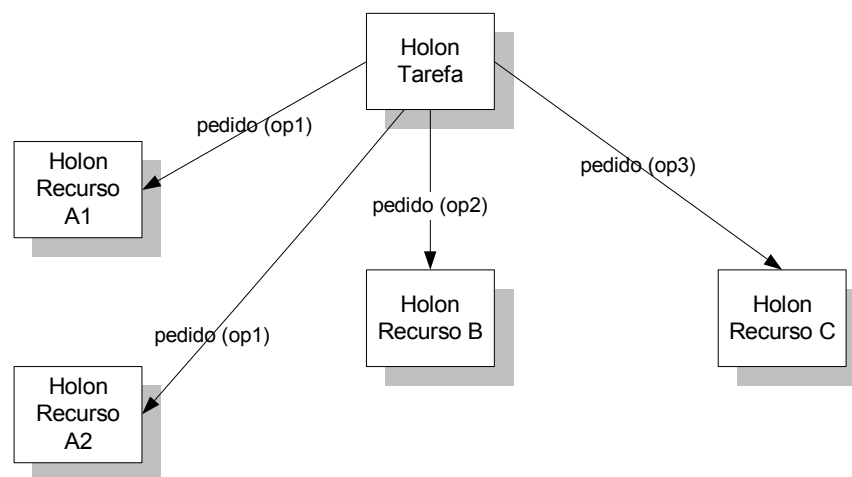


Figura 6.6 – Requisição de recursos

Mensagem de Requisição de Serviços

O Holon de Tarefa irá contactar cada Holon de Recurso para cada operação constante do plano com uma mensagem com o seguinte teor:

pedido(DataLimite, OpId, Quant, JanelaTemporal, Pred, Succ)

onde *DataLimite*, *OpId*, *Quant*, *JanelaTemporal*, *Pred* e *Succ* denotam respectivamente, a data considerada para conclusão da tarefa, a identificação da operação requerida, a quantidade de produtos a fabricar (*i.e.*, número de vezes que a operação tem que ser repetida), a janela temporal para o escalonamento, a lista de recursos contactados para a realização das operações imediatamente anteriores à que está a ser considerada no plano e, a lista de recursos contactados para as operações imediatamente posteriores a esta. Esta informação sobre antecessores e sucessores é necessária para os recursos poderem coordenar entre si os intervalos de tempo em que faz sentido escalonar cada operação.

6.2.1.3 Influência Directa

Esta é a primeira fase de propagação de restrições onde os recursos a montante influenciam os recursos a jusante, para a determinação do limite inferior da janela temporal.

Quando um *Holon de Recurso* recebe uma mensagem *pedido* verifica a sua agenda para as actividades já escalonadas e, constrói uma lista com os intervalos livres, onde é possível escalonar uma nova operação. Em seguida, os Holons de Recurso sem antecessores irão passar a lista de intervalos livres influenciada para os recursos que os sucedem (mensagem *influencia_directa*) (Figura 6.7, adaptada de: [Sousa *et al.*, 1999b]). Esta lista influenciada é obtida através do deslocamento à direita dos intervalos livres em n unidades de tempo (correspondente ao tempo necessário para executar a operação requisitada).

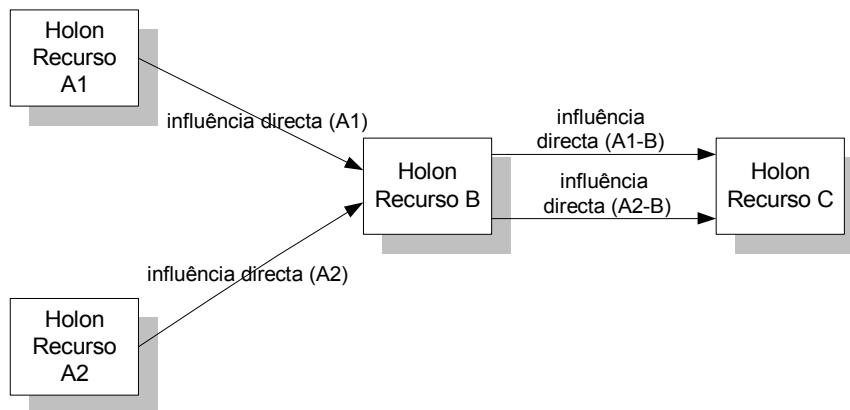


Figura 6.7 – Influência directa

Para cada mensagem *influencia_directa* que um Holon de Recurso recebe dos recursos antecessores, o Holon de Recurso procederá a uma combinação entre a sua lista de intervalos livres e a lista de intervalos recebida, obtendo assim uma nova lista de intervalos, cujo limite inferior reflecte as restrições impostas por esses recursos. Essa nova lista será influenciada e passada para os recursos sucessores. Ao passar a lista influenciada de intervalos, cada holon passa também o caminho correspondente aos recursos já utilizados. A fase de influência directa termina quando todos os recursos sem sucessores receberem as listas influenciadas dos seus antecessores.

Mensagem de Influência Directa

Um Holon de Recurso envia a outro Holon de Recurso seu sucessor a lista influenciada de intervalos livres numa mensagem com o seguinte teor:

influencia_directa(TId, OpId, Dur, LIntervalos, Caminho)

onde *TId*, *OpId*, *Dur*, *LIntervalos* e *Caminho* denotam respectivamente, o identificador do Holon de Tarefa, o identificador da operação requisitada ao Holon de Recurso receptor da

mensagem, a duração da operação requisitada ao holon emissor e, a lista influenciada de intervalos livres desse recurso e a lista de todos os recursos antecessores.

6.2.1.4 Influência Inversa e Proposta

Nesta segunda fase de propagação de restrições, os recursos a jusante influenciam os recursos a montante para a determinação do limite superior (efectivo) da janela temporal. A fase de influência inversa é semelhante à influência directa, mas no sentido oposto.

Após combinarem a sua agenda com a lista de intervalos recebida dos antecessores, os Holons de Recurso sem sucessores irão dar início à fase de influência inversa. Estes holons irão passar a lista de intervalos livres influenciada para os recursos que os precedem (mensagem *influencia_inversa*) (Figura 6.8, adaptada de: [Sousa *et al.*, 1999b]). Esta lista influenciada é obtida através do deslocamento à esquerda dos intervalos livres em n unidades de tempo (correspondente à duração da operação requisitada).

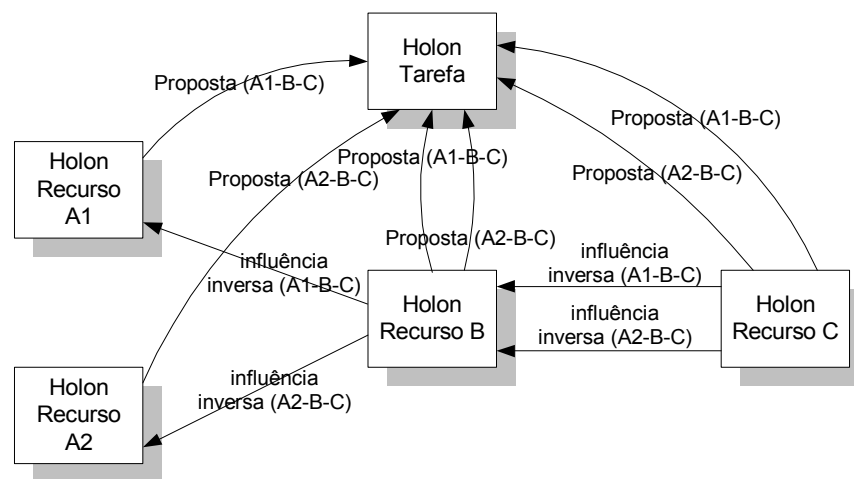


Figura 6.8 – Influência inversa e ofertas

Para cada mensagem *influencia_inversa* recebida com uma lista de intervalos de tempo possíveis de utilização, cada Holon de Recurso irá combinar essa lista de intervalos com a sua própria lista de forma a determinar o limite inferior (efectivo) de escalonamento, gerando assim a lista final de intervalos livres. Após influenciar essa lista e a passar aos recursos utilizados nas operações precedentes, o Holon de Recurso enviará a proposta de escalonamento ao Holon de Tarefa (mensagem *proposta*). É enviada uma proposta para cada combinação possível de recursos para executar o plano de produção. O envio da lista influenciada aos recursos antecessores leva em consideração o caminho por onde a mensagem chegou (ou seja, uma mensagem que tenha sido originada pela combinação de recursos *A1-B-C* na fase de influência directa irá passar apenas

por esses recursos na fase de influência inversa). Esta fase termina quando todos os recursos sem antecessores receberem as listas influenciadas dos respectivos sucessores.

Mensagem de Influência Inversa

Um Holon de Recurso envia a outro Holon de Recurso seu antecessor a lista influenciada de intervalos livres numa mensagem com o seguinte teor:

influencia_inversa(TId, OpId, LIntervalos, Caminho)

onde *TId*, *OpId*, *LIntervalos* e *Caminho* denotam respectivamente, o identificador do Holon Tarefa, o identificador da operação requisitada ao Holon de Recurso receptor da mensagem, a lista influenciada de intervalos livres desse recurso e, a combinação de recursos que deu origem à solução que esta mensagem de influência inversa materializa.

Mensagem de Proposta de Escalonamento

Quando um Holon de Recursos elabora a lista final de intervalos livres, envia essa proposta ao Holon de Tarefa com uma mensagem com o seguinte teor:

proposta(OpId, Duração, LIntervalos, Custo)

onde *OpId*, *Duração*, *LIntervalos* e *Custo* denotam respectivamente, a identificação da operação requisitada, a duração do intervalo de tempo necessário para fabricar a quantidade requisitada de produtos, a lista de intervalos livres onde é possível escalonar e, o custo que este recurso atribui à execução daquela operação.

6.2.1.5 Contratação

Após receber as propostas de escalonamento de cada um dos Holons de Recurso contactados, o Holon de Tarefa analisa as propostas e escolhe a combinação de recursos a utilizar (caso existam recursos alternativos para uma ou mais operações) e qual o intervalo de tempo mais indicado para a utilização de cada um dos recursos. Após a selecção o Holon de Tarefa envia uma mensagem a cada Holon de Recurso indicando a decisão tomada quanto à proposta recebida (Figura 6.9, *adaptada de*: [Sousa *et al.*, 1999b]).

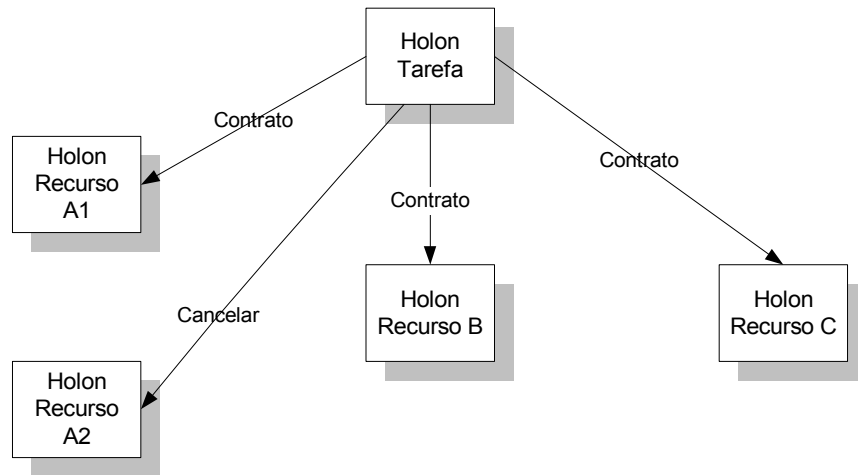


Figura 6.9 – Conclusão da negociação

A selecção da combinação de recursos a usar (bem como do intervalo de tempo a utilizar) é baseada em heurísticas (*e.g.*, maior folga) e no custo total da solução para a tarefa a executar com base no custo de cada proposta recebida e escolhida – equação (6.2).

$$\text{custo-da-solução} = \sum_{i=1}^n \text{custo}(S(i)) \quad (6.2)$$

onde, n é o número de operações do plano, $S(i)$ é uma função que devolve a proposta seleccionada para efectuar a operação i e $\text{custo}()$ é uma função que devolve o custo de uma proposta.

Mensagem de Contratação

O Holon de Tarefa comunica a aceitação da proposta a um Holon de Recurso com uma mensagem com o seguinte teor:

contrata(OpId, IntervaloSeleccionado, PredSel, SuccSel)

onde *OpId*, *IntervaloSeleccionado*, *PredSel* e *SuccSel* denotam respectivamente, a identificação da operação anteriormente requisitada, o intervalo seleccionado pela tarefa de entre os intervalos propostos pelo recurso, a lista de recursos seleccionados para as operações antecessoras e, a lista de recursos seleccionados para as operações sucessoras.

Mensagem de Cancelamento de Requisição

O Holon de Tarefa comunica a rejeição da proposta de um Holon de Recurso com uma mensagem com o seguinte teor:

cancela(OpId)

onde *OpId* denota a operação anteriormente requisitada.

6.2.2 Tratamento de Conflitos

Durante a execução simultânea do protocolo por vários Holons de Tarefa pode surgir um conflito quando mais do que uma tarefa requisita a realização de uma operação a um dado recurso em janelas temporais sobrepostas [Sousa e Ramos, 1998] [Sousa e Ramos, 1999a].

Por exemplo, considere-se um cenário em que há duas tarefas a executar, T_1 e T_2 , e dois recursos, R_A e R_B , no qual T_1 inicia a negociação de op_1 com R_A e R_B . Tanto R_A como R_B analisam as suas agendas e encontram um intervalo livre para escalonar op_1 informando T_1 desse intervalo. T_1 analisa então as propostas de R_A e R_B para decidir a qual dos recursos atribuir a execução de op_1 . Entretanto, R_A recebe um pedido de T_2 para uma outra operação (op_2). Suponha-se agora que a única maneira de executar op_2 cumprindo o prazo imposto por T_2 passa por utilizar parte do intervalo proposto a T_1 . R_A tem um *Problema de Indecisão*, pois ainda não recebeu resposta de T_1 , confirmando ou não a utilização do intervalo proposto.

Dois senões podem ocorrer:

1. R_A considera o intervalo como reservado para a execução de op_1 de T_1 informando T_2 que não pode executar op_2 . Caso T_1 decida contratar R_B , o recurso R_A perde ambas as oportunidades de contrato quando seria possível executar op_2 , visto que o intervalo foi rejeitado por T_1 .
2. R_A considera o intervalo livre e informa T_2 que pode executar op_2 . Caso T_1 e T_2 decidam contratar R_A o recurso fica comprometido com a execução de duas operações, mas vai falhar a execução de uma delas devido à impossibilidade física de executar duas operações em simultâneo.

Este tipo de problema é denominado Problema de Indecisão podendo ser solucionado de uma forma simples, que passa por acrescentar alguns procedimentos extra à cabeça do protocolo. A Figura 6.10 (*adaptada de*: [Sousa *et al.*, 1999b]) apresenta a nova sequência de mensagens trocadas na fase de anúncio, para contornar o problema de indecisão.

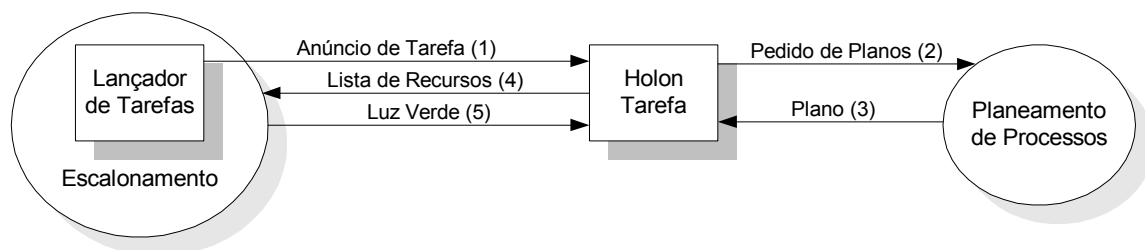


Figura 6.10 – Alteração ao PRCPR: tratamento de conflitos

Os passos (1), (2) e (3) são exactamente os mesmos descritos na secção 6.2.1.1. Após receber o plano de produção a utilizar o *Holon de Tarefa* constrói a lista de recursos a contactar, mas antes de negociar com cada recurso, envia essa lista ao *Holon de Escalonamento* (mensagem *lista_de_recursos*).

Para evitar conflitos o Holon de Escalonamento possui uma lista de recursos em negociação e uma outra lista de tarefas em espera ordenada por prioridade. O Holon de Escalonamento conhece as janelas temporais que estão a ser negociadas para cada recurso, colocando em espera as tarefas que desejam aceder a esses recursos, caso possuam uma janela temporal que se sobreponha às janelas temporais em negociação. Este mecanismo permite a negociação em simultâneo da execução de várias tarefas, apenas impedindo a negociação simultânea de tarefas em janelas temporais sobrepostas. Na prática, este mecanismo é semelhante ao de exclusão mútua na programação concorrente.

A prioridade para a execução de uma tarefa T é dada pela equação (6.3), baseada no valor total das encomendas que deram origem a essa tarefa (ε_i), bem como na prioridade associada aos clientes dessas encomendas. $\Delta(v)$ denota uma operação de normalização, aplicada a valores monetários, por forma a que a adição de montantes (monetários) ligados às prioridades associadas à execução de encomendas, faça sentido.

$$prioridade(T) \stackrel{\text{def}}{=} \Delta \left[\sum_i total(\varepsilon_i) \right] + \Delta \left[\sum_i prioridade(cliente(\varepsilon_i)) \right] \quad (6.3)$$

Quando não há possibilidade de conflitos o Holon de Escalonamento sinaliza os Holons de Tarefa em espera que podem iniciar a negociação, através de uma mensagem *luz_verde*.

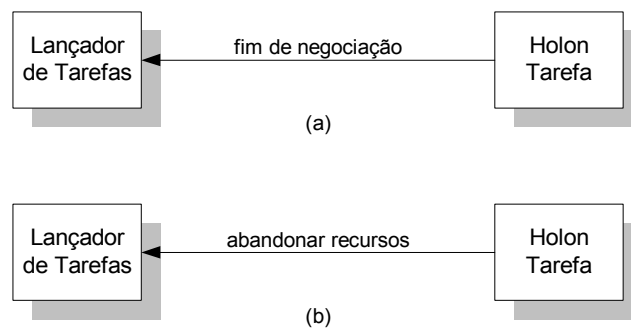


Figura 6.11 – Alterações ao PRCPR: (a) fim de negociação; (b) abandono de recursos

Para actualizar a lista de recursos em negociação o Holon de Tarefa deve indicar ao Holon de Escalonamento, no final da execução do protocolo, que concluiu a negociação (mensagem *fim_de_negociação*, Figura 6.11a). tendo em vista a optimização do processo, o Holon de Tarefa pode também durante a execução do protocolo informar o Holon de Escalonamento de qualquer

recurso que afinal já não vai contactar (mensagem *abandonar_recursos*, Figura 6.11b, *adaptada de* [Sousa e Ramos, 1998]).

Mensagem com a Lista de Recursos

O Holon de Tarefa avisa o Holon de Escalonamento sobre quais os recursos que vai contactar com uma mensagem com o seguinte teor:

lista_de_recursos(LRecursos, JT)

onde *LRecursos* denota a lista de Holons de Recurso que esta Tarefa vai contactar e *JT* denota a janela temporal que vai ser negociada.

Mensagem de “Luz Verde”

O Holon de Escalonamento comunica ao Holon de Tarefa que pode iniciar a negociação com a seguinte mensagem:

luz_verde

Mensagem de Fim de Negociação

O Holon de Tarefa avisa o Holon de Escalonamento que terminou a negociação com a seguinte mensagem:

fim_de_negociação

Mensagem de Abandono de Recursos

O Holon de Tarefa avisa o Holon de Escalonamento que não vai entrar em contacto com determinados recursos usando uma mensagem com o seguinte teor:

abandonar_recursos(LRecursos)

onde *LRecursos* denota a lista de Holons de Recurso que esta Tarefa já não vai contactar.

6.2.3 Tratamento de Excepções

As fases descritas anteriormente correspondem à execução do protocolo para a contratação de serviços entre tarefas e recursos. No entanto, após o estabelecimento do contrato, as condições do sistema podem mudar de tal forma que seja necessário renegociar o escalonamento (*e.g.*, avaria de um recurso, entrada de tarefa prioritária).

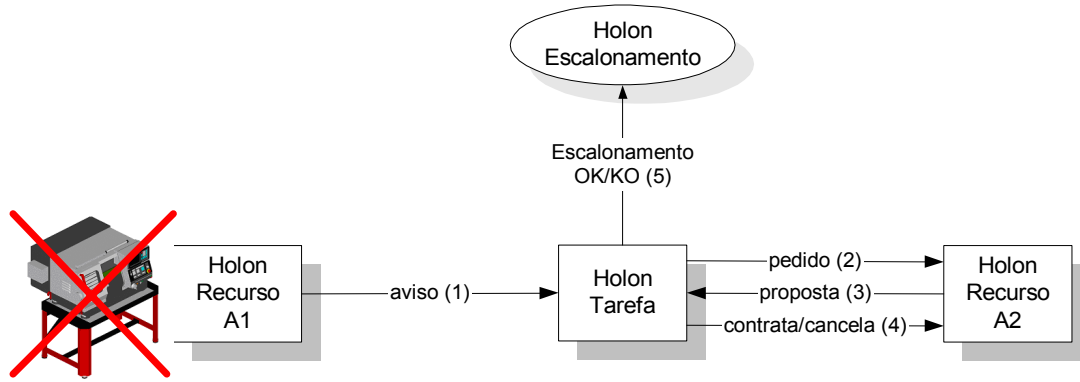


Figura 6.12 – Exemplo de avaria de um recurso

Quando um *Holon de Recurso* detecta, por exemplo, uma avaria que não pode ser recuperada (em tempo útil), informa os *Holons de Tarefa* que o tenham contratado da impossibilidade de cumprir o contrato. Após receber a mensagem de *aviso* do Holon de Recurso, o Holon de Tarefa vai tentar escalonar as operações em causa com outros Holons de Recurso (Figura 6.12, *fonte*: [Sousa e Ramos, 1999a]). Em certas situações, o Holon de Recurso pode não informar os Holons de Tarefa da avaria e tentar recuperá-la, executando as operações contratadas com um pequeno atraso e/ou executando-as de forma a recuperar o tempo perdido (*e.g.*, utilização de uma maior velocidade de operação).

6.2.4 Análise da Complexidade do Protocolo

Nesta secção é efectuada uma análise ao protocolo acabado de descrever, do ponto de vista de complexidade do problema (número de soluções), número de mensagens trocadas e tamanho total das mensagens trocadas.

Complexidade do Problema de Escalonamento

O problema de escalonamento com n operações em m máquinas é um problema NP-completo que resulta numa explosão combinatória do espaço de soluções. O protocolo descrito vai gerando as várias combinações possíveis de recursos durante as fases de influência (*i.e.*, o número de planos de produção alternativos tendo em conta os vários recursos capazes de efectuar cada operação).

Tendo em conta que um plano \mathcal{P} é um grafo constituído por uma sequência de operações ou por várias sequências que ocorrem em paralelo [Ramos *et al.*, 1998] [Rocha, 1999], que pode ser descrito pela seguinte gramática BNF:

$$\begin{aligned}
 \text{PLANO} &::= \text{PAR} \mid \text{SEQ} \\
 \text{PAR} &::= \text{par}(\text{Lista-SEQ}) \\
 \text{SEQ} &::= \text{seq}(\text{Lista-ARGSEQ})
 \end{aligned}$$

$$\begin{aligned}
 \text{Lista-SEQ} &::= \text{SEQ} \mid \text{SEQ}, \text{Lista-SEQ} \\
 \text{Lista-ARGSEQ} &::= \text{ARGSEQ} \mid \text{ARGSEQ}, \text{Lista-ARGSEQ} \\
 \text{ARGSEQ} &::= \text{Operação} \mid \text{PAR}
 \end{aligned}$$

Sabendo-se que o número de combinações num dos ramos sequenciais é dado pela equação (6.4), e o número de combinações em ramos paralelos dado pela equação (6.5), em que n é o número de operações no ramo sequencial em questão, $R(i)$ é uma função que devolve o número de recursos capazes de executar a operação i daquele ramo, k é o número de ramos paralelos, e $\text{ramo}(i)$ é uma função que devolve cada um dos ramos do grafo em questão.

$$\prod_{i=1}^n R(i) \quad (6.4)$$

$$\sum_{i=1}^k \chi(\text{ramo}(i)) \quad (6.5)$$

O número de combinações possíveis, denotado $\chi(\mathcal{P})$, é então uma consequência lógica do programa em lógica que a seguir se apresenta, no qual o predicado $n_combinações(\text{PlanoBNF}, N)$ denota $\chi(\text{transformar}(\mathcal{P}))$, sendo que $\text{transformar}(\mathcal{P})$ é uma função que constrói uma representação do plano \mathcal{P} na gramática referida em epígrafe.

$$\begin{aligned}
 n_combinações(\text{par}(\text{LSeq}), N) &\leftarrow \\
 &\quad n_combinações_par(\text{LSeq}, N) \\
 n_combinações(\text{seq}(\text{LArgSeq}), N) &\leftarrow \\
 &\quad n_combinações_seq(\text{LArgSeq}, N) \\
 n_combinações_par([], 0) & \\
 n_combinações_par([\text{seq}(\text{LArgSeq}) \mid T], N) &\leftarrow \\
 &\quad n_combinações(\text{seq}(\text{LArgSeq}), X) \wedge \\
 &\quad n_combinações_par(T, Z) \wedge \\
 &\quad \text{atribui}(N, X + Z) \\
 n_combinações_seq([], 1) & \\
 n_combinações_seq([\text{par}(\text{LSeq}) \mid T], N) &\leftarrow \\
 &\quad n_combinações(\text{par}(\text{LSeq}), X) \wedge \\
 &\quad n_combinações_seq(T, Z) \wedge \\
 &\quad \text{atribui}(N, X * Z) \\
 n_combinações_seq([Op \mid T], N) &\leftarrow \\
 &\quad n_rec_op(Op, X) \wedge \\
 &\quad n_combinações_seq(T, Z) \wedge \\
 &\quad \text{atribui}(N, X * Z)
 \end{aligned}$$

A complexidade do problema traduz-se então na análise de complexidade do programa anterior. O melhor caso ocorre quando se tem uma situação onde o plano de produção \mathcal{P} contempla a execução de uma única operação, sendo portanto $\chi(\mathcal{P})$ igual ao número de recursos capazes de efectuar essa operação – vide (6.6). Um dos casos particulares ocorre quando só existe

um recurso para a execução de todas as operações do plano, havendo portanto apenas uma combinação possível – *vide* (6.7). No pior dos casos, o plano \mathcal{P} é dado por uma sequência de n operações em que cada operação possui um número igual de recursos capazes de a efectuar, sendo por isso o número de combinações possíveis igual a r^n – *vide* (6.8).

$$n = 1 \Rightarrow \chi(\mathcal{P}) = R(1) \quad (6.6)$$

$$\forall(i) R(i) = 1 \Rightarrow \chi(\mathcal{P}) = 1 \quad (6.7)$$

$$(\forall(i) R(i) = r \wedge r > 1) \Rightarrow \chi(\mathcal{P}) = \prod_{i=1}^n r \Leftrightarrow \chi(\mathcal{P}) = r^n \quad (6.8)$$

A medida de complexidade do problema para o pior caso é dada por $O(r^n)$ ²¹ ou, mais genericamente, por $O(\max(R(i))^n)$, onde $\max(R(i))$ denota a função que devolve o valor máximo da função $R(i)$ aplicada ao plano \mathcal{P} .

Complexidade do Protocolo

O número total de mensagens trocadas para uma execução do Protocolo de Rede de Contrato com Propagação de Restrições depende directamente do número de combinações existentes e é dado pela equação (6.9), onde n é o número de operações do plano e $\psi(\mathcal{P})$ é o número de mensagens trocadas durante a fase de influência directa ou inversa, dado pela equação (6.10).

$$M_{prepr} = \left(\sum_{i=1}^n R(i) \right) + [2 \times \psi(\mathcal{P})] + [2 \times (\psi(\mathcal{P}) + \chi(\mathcal{P}))] \quad (6.9)$$

$$\psi(\mathcal{P}) = \sum_{i=1}^{n-1} (R(i) \times \chi(\text{truncar}(\mathcal{P}, i)) \times R(\text{succ}(i))) \quad (6.10)$$

O primeiro termo desta adição corresponde ao número de mensagens enviadas para requisição de serviço, o segundo termo corresponde ao número de mensagens trocadas na fase de influência directa e inversa (uma mensagem de influência para cada recurso a aplicar de seguida (sucessor) por cada mensagem de influência recebida de um recursos antecessor) e o último termo corresponde ao número de mensagens de proposta dos Holons de Recurso e resposta do Holon de Tarefa (uma proposta por cada mensagem de influência recebida e uma resposta por cada proposta). A função $\text{truncar}(\mathcal{P}, i)$ constrói um plano baseado em \mathcal{P} truncado até à operação i (exclusive), ou seja, um plano composto por todos os ramos de operações antecessoras da operação i . No caso de operações sem antecessoras devolve uma sequência vazia (já que

²¹ A notação O permite indicar a complexidade de um algoritmo representado por uma função $f(n)$. Mais propriamente, $f(n) = O(g(n))$ quer dizer que $c \cdot g(n)$ é um limite superior para $f(n)$. Ou seja, existe uma constante c tal que $f(n) \leq c \cdot g(n)$ para valores de n suficientemente grandes [Skiena, 1997].

$\chi(seq(\emptyset)) = 1$, elemento neutro da multiplicação), e $succ(i)$ denota a função que devolve a operação que sucede à operação i .

A Figura 6.13 apresenta três planos de produção, utilizados como exemplo para análise da complexidade do PRCPR. Na figura as operações são representadas por círculos tendo na parte superior a identificação da operação e, na parte inferior, o número de recursos capazes de efectuar essa operação. Na Tabela 6.1 são apresentados os valores de $\chi(\mathcal{P})$, $\psi(\mathcal{P})$ e M_{prcpr} para cada um desses planos.

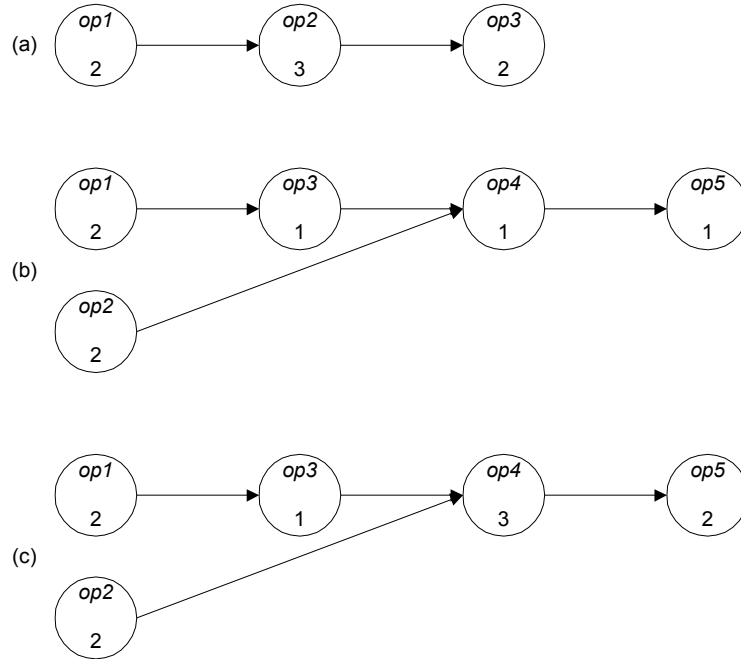


Figura 6.13 – Exemplos de planos para estudo da complexidade do protocolo RCPR

Tabela 6.1 – Valores de $\chi(\mathcal{P})$ e M_{prcpr} para os planos exemplo

Plano	Termo	Valor
(a)	$\chi(\mathcal{P})$	$2 \times 3 \times 2 = 12$
	$\psi(\mathcal{P})$	$2 \times 1 \times 3 + 3 \times 2 \times 2 = 18$
	M_{prcpr}	$(2+3+2) + 2 \times 18 + 2 \times (18 + 12) = 7+36+60 = \mathbf{103}$
(b)	$\chi(\mathcal{P})$	$[(2 \times 1 \times 1) + (2 \times 1)] \times 1 = 4$
	$\psi(\mathcal{P})$	$2 \times 1 \times 1 + 2 \times 1 \times 1 + 1 \times 2 \times 1 + 1 \times 4 \times 1 = 10$
	M_{prcpr}	$(2+2+1+1+1) + 2 \times 10 + 2 \times (10 + 4) = 7+20+28 = \mathbf{55}$

Plano	Termo	Valor
(c)	$\chi(\mathcal{P})$	$[(2 \times 1 \times 3) + (2 \times 3)] \times 2 = 24$
	$\psi(\mathcal{P})$	$2 \times 1 \times 1 + 2 \times 1 \times 3 + 1 \times 2 \times 3 + 3 \times 4 \times 2 = 38$
	M_{prcpr}	$(2+2+1+3+2) + 2 \times 38 + 2 \times (38 + 24) = 10+76+124 = \mathbf{210}$

Conforme se pode verificar, o número de mensagens trocadas é elevado, mesmo para planos relativamente simples e com poucos recursos alternativos. Em cenários reais é teoricamente possível executar todas as operações em cada um dos recursos existentes na instalação fabril (devido à polivalência dos recursos), e além disso, os planos de produção consistem normalmente em dezenas ou centenas de operações.

Conforme foi dito anteriormente, o pior caso para a execução do algoritmo ocorre quando se tem um plano sequencial de n operações com r recursos para cada operação. Assim sendo, M_{prcpr} é majorada da seguinte forma:

$$M_{prcpr} \leq n \cdot r + 2 \times 2 \times r^n + 2 \times (2 \cdot r^n + r^n) \Leftrightarrow M_{prcpr} \leq n \cdot r + 10r^n$$

A complexidade do PRCPR (no que toca ao número de mensagens trocadas) para o pior caso é $O(r^n)$, pois a componente linear $n \cdot r$ pode ser ignorada para valores de n suficientemente grandes, devido à componente exponencial r^n .

Devido à explosão combinatória do problema e à correspondência do envio de uma mensagem de influência para cada mensagem de influência recebida (o que torna o número de mensagens trocadas directamente relacionado com $\chi(\mathcal{P})$), é lícito considerar como hipótese uma outra abordagem, em que se diminua o número de mensagens trocadas.

Alteração ao Protocolo na Tentativa de Diminuir o Número de Mensagens Trocadas

A natureza distribuída da arquitectura favorece um protocolo de interacção semelhante ao descrito, onde cada mensagem recebida é processada e dá origem a uma ou mais respostas (granulosidade fina). No entanto, devido à explosão combinatória de soluções para o problema, o protocolo pode ser modificado, de modo a que cada recurso envie apenas uma mensagem com todas as combinações, ao invés de enviar uma mensagem por combinação [Sousa *et al.*, 1999b]. Trata-se então de equacionar uma outra solução, onde se diminui o número de mensagens, mas aumenta-se necessariamente o tamanho das mensagens trocadas (granulosidade grossa).

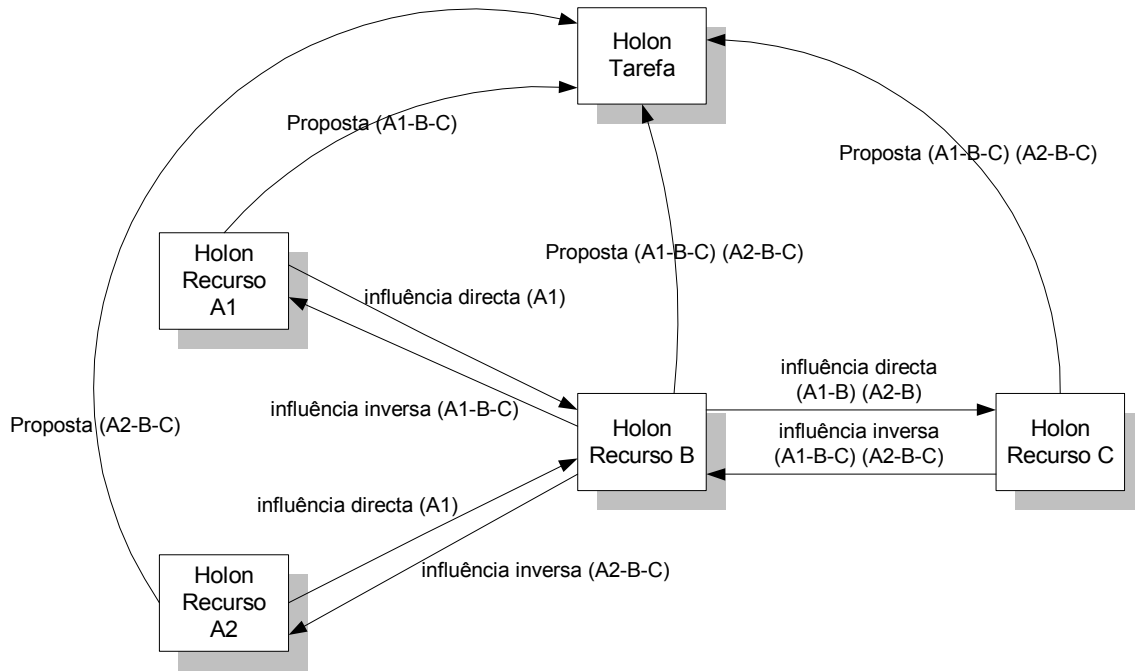


Figura 6.14 – Alterações na fase de influência directa, influência inversa e proposta

A Figura 6.14 apresenta as fases de influência directa, inversa e proposta para esta variante do PRCPR. Conforme se pode observar na figura, apenas é trocada uma mensagem por todas as mensagens recebidas na fase de influência directa. Na fase de influência inversa o número de mensagens enviadas corresponde ao número de mensagens recebidas na fase de influência directa. Além disso, o número de mensagens de proposta é agora apenas de uma por Holon de Recurso.

As mensagens de influência directa, influência inversa e proposta passam a ter os teores apresentados em seguida:

influencia_directa(TId, OpId, LComb)

influencia_inversa(TId, OpId, LComb)

proposta(OpId, Duração, LCombProposta)

onde *TId*, *OpId* e *Duração* denotam respectivamente, o identificador do Holon Tarefa, o identificador da operação requisitada ao holon de recurso receptor da mensagem, e a duração do intervalo de tempo necessário para executar as quantidades indicadas de produtos. O atributo *LComb* denota, por seu lado, uma lista de pares (*LIntervalos*, *Caminho*) em que *LIntervalos* é a lista influenciada de intervalos livres para esse recurso e *Caminho* é a lista de todos os recursos antecessores, e o atributo *LCombProposta* denota uma lista de tuplos na forma (*LIntervalos*, *Caminho*, *Custo*) onde *LIntervalos* é a lista de intervalos livres onde é possível escalonar e *Custo* representa o custo que este recurso atribui à execução dessa operação.

Complexidade da Variante do Protocolo RCPR

O número de mensagens trocadas na execução desta variante do PRCPR é representado por M_{prcpr}^2 e calculado pela equação (6.11).

$$M_{prcpr}^2 = 3 \times \left(\sum_{i=1}^n R(i) \right) + 2 \times \left(\sum_{i=1}^{n-1} (R(i) \times R(succ(i))) \right) \quad (6.11)$$

O número de mensagens de proposta e de resposta é agora igual ao número de requisições (daí o factor de multiplicação três (3) no primeiro termo da equação (6.11)). O número de mensagens nas fases de influência directa e inversa fica neste caso apenas dependente do número de recursos capazes de efectuar cada operação e do número de recursos capazes de efectuar as operações sucessoras.

Na Tabela 6.2 são apresentados os valores de M_{prcpr}^2 para os planos tomados como exemplo e dados pela Figura 6.13. Conforme se pode verificar, o número de mensagens trocadas é agora substancialmente menor.

Tabela 6.2 – Valores de M_{prcpr}^2 para planos exemplo

Plano	M_{prcpr}^2
(a)	$3 \times (2+3+2) + 2 \times (2 \times 3 + 3 \times 2) = 21+24 = \mathbf{48}$
(b)	$3 \times (2+2+1+1+1) + 2 \times (2 \times 1 + 2 \times 1 + 1 \times 1 + 1 \times 1) = 21+12 = \mathbf{33}$
(c)	$3 \times (2+2+1+3+2) + 2 \times (2 \times 1 + 2 \times 3 + 1 \times 3 + 3 \times 2) = 30+34 = \mathbf{64}$

A Figura 6.15 apresenta em forma gráfica a evolução do número de mensagens trocadas com o aumento do número de operações de um plano. Os dados correspondem a uma situação em que se tem um plano composto por uma sequência de n operações, em que cada operação possui r ($= 2$) recursos capazes de a efectuar²². O gráfico utiliza uma escala logarítmica, em que M_{prcpr} é dado pela linha com marcas circulares, M_{prcpr}^2 pela linha com marcas quadradas e $\chi(\mathcal{P})$ pela linha mais esbatida.

²² No Apêndice B é apresentado o modelo matemático e respectiva folha de cálculo utilizada para a geração dos dados deste gráfico e do da Figura 6.16.

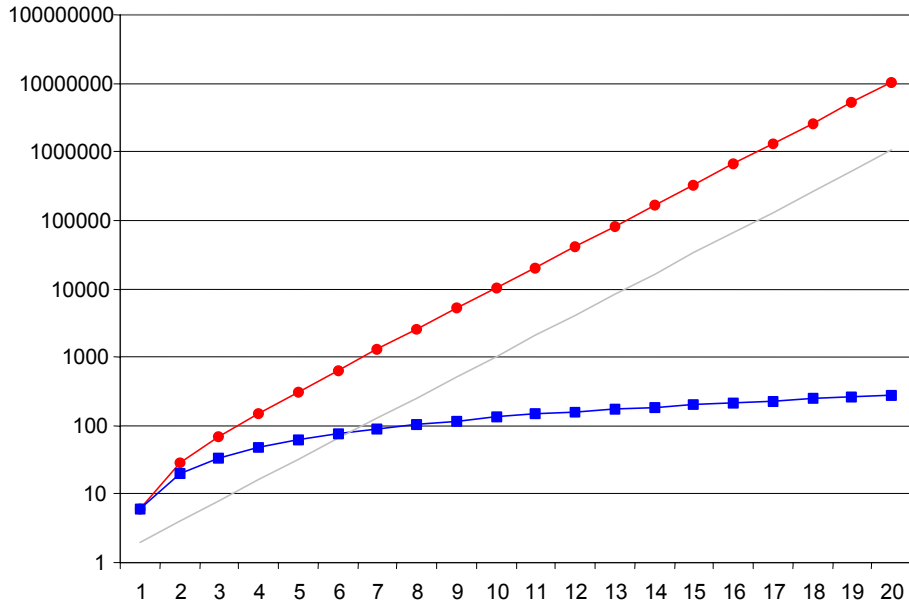


Figura 6.15 – Evolução do número de mensagens com o aumento de operações

Conforme se pode ver $\chi(\mathcal{P})$ e M_{precpr} crescem exponencialmente com o número de operações, enquanto que M_{precpr}^2 cresce linearmente. Daqui se pode observar que a variante do PRCPR coloca uma carga menor no sistema em termos de número de mensagens trocadas ($M_{precpr}^2 \ll M_{precpr}$).

Como exemplo ilustrativo, a razão $\frac{M_{precpr}^2}{M_{precpr}}$ para $n = 10$ é de 0,07 enquanto que para $n = 20$ é de 2.59×10^{-5} .

Conforme foi dito anteriormente, o pior caso na utilização do algoritmo ocorre quando se tem um plano sequencial com n operações e com r recursos para cada operação. Nesta situação, M_{precpr}^2 é majorada da seguinte forma:

$$M_{precpr}^2 \leq 3 \cdot n \cdot r + 2 \times (n-1) \times r^2 \Leftrightarrow M_{precpr}^2 \leq 3nr + 2nr^2 \Leftrightarrow M_{precpr}^2 \leq n \cdot (3r + 2r^2)$$

A medida de complexidade da variante do PRCPR é dada por $O(n)$, pois não só r é constante, como é sempre possível encontrar uma constante $c > (3r + 2r^2)$.

Cálculo do Tamanho Total das Mensagens Trocadas

Para comparar as duas abordagens ao PRCPR vai-se então efectuar o cálculo do comprimento das mensagens trocadas num e noutro caso. Para simplificar, assume-se que o tamanho da lista influenciada de intervalos, o caminho e a lista de propostas é sempre igual para qualquer par $\langle \text{recurso}, \text{operação} \rangle$ (uma situação que pode teoricamente acontecer quando todos os recursos

possuem a agenda de actividades completamente livre). O tamanho da mensagem de influência directa ou inversa é então dado pela equação (6.12) e o tamanho da mensagem de proposta pela equação (6.13), onde Cab , LdI , $Cam(n)$ e LdP denotam respectivamente, o tamanho do cabeçalho da mensagem, o tamanho da lista de intervalos influenciados, o tamanho da lista com a combinação de recursos de uma solução e, o tamanho da lista com os intervalos propostos.

$$T_{inf} = Cab + LdI + Cam(n) \quad (6.12)$$

$$T_{bid} = Cab + LdP \quad (6.13)$$

Para a variante do PRCPR, o comprimento da mensagem de influência directa ou inversa é dado pela equação (6.14) e o da mensagem de proposta pela equação (6.15) onde \emptyset denota o conjunto vazio.

$$T_{inf}^2(i) = Cab + \chi(trunc(\mathcal{P}, i)) \times (LdI + Cam(n)) \quad (6.14)$$

$$T_{bid}^2(i) = \begin{cases} Cab + \chi(\mathcal{P}) \times LdP & succ(i) \neq \emptyset \\ Cab + \chi(trunc(\mathcal{P}, i)) \times LdP & succ(i) = \emptyset \end{cases} \quad (6.15)$$

O comprimento de todas as mensagens trocadas nas fases de influência inversa e directa e proposta no PRCPR é então dado pela equação (6.16) e, no caso da variante do PRCPR, pela equação (6.17).

$$T_{prcpr|inf+bid|} = 2 \times \psi(\mathcal{P}) \times T_{inf} + (\psi(\mathcal{P}) + \chi(\mathcal{P})) \times T_{bid} \quad (6.16)$$

$$T_{prcpr|inf+bid|}^2 = 2 \times \left(\sum_{i=1}^{n-1} (R(i) \times R(succ(i)) \times T_{inf}^2(i)) \right) + \left(\sum_{i=1}^n R(i) \times T_{bid}^2(i) \right) \quad (6.17)$$

A Figura 6.16 apresenta em modo gráfico uma comparação entre $T_{prcpr|inf+bid|}$ (barras claras) e $T_{prcpr|inf+bid|}^2$ (barras escuras). O eixo primário do gráfico (lado esquerdo) tem uma escala logarítmica e denota o comprimento das mensagens, enquanto que o eixo secundário (do lado direito) tem uma escala linear e denota a razão $\frac{T_{prcpr|inf+bid|}^2}{T_{prcpr|inf+bid|}}$ (linha escura).

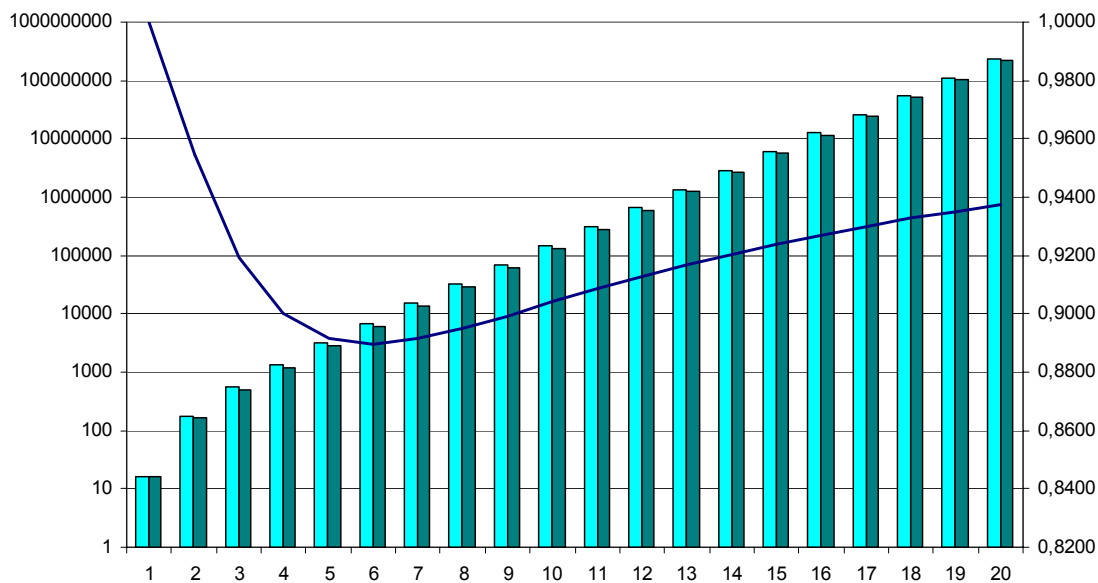


Figura 6.16 – Tamanho total de mensagens

O gráfico corresponde a valores encontrados para um plano de produção sequencial, de n operações com um número igual de recursos $r (=2)$ para efectuar cada operação, e considerando-se $Cab = 2$, $LdI = 10$, $Cam(n) = 2 \cdot (n-1)$ e $LdP = 6$.

Daqui se pode verificar que o comprimento das mensagens trocadas na variante do PRCPR é menor que no original ($T_{prcpr|inf+bid}^2 < T_{prcpr|inf+bid}$). Também se verifica que a razão entre os termos

das duas séries é crescente, o que implica que o ganho ($G = 1 - \frac{T_{prcpr|inf+bid}^2}{T_{prcpr|inf+bid}}$) vá diminuindo com

o aumento de n . Por exemplo, para $n = 20$, o valor médio de $\frac{T_{prcpr|inf+bid}^2}{T_{prcpr|inf+bid}}$ é 0,9195 e o valor médio

de G é 0,0805; para $n = 50$, o valor médio de $\frac{T_{prcpr|inf+bid}^2}{T_{prcpr|inf+bid}}$ é 0,9426 e o valor médio de G é 0,0574.

6.3 Funcionamento dos Holons

Na secção 5.2 foi apresentada a arquitectura para um sistema de produção de nova geração e na secção 5.3 foram formalmente caracterizados os vários holons da arquitectura do ponto de vista de conhecimento, objectivos e ciclo de vida. Nesta secção, esses holons serão considerados do ponto de vista de procedimentos que a si têm associados. A especificação destes procedimentos terá apenas em conta a função de Escalonamento, escolhida como caso de teste para o protótipo a

desenvolver. Assim sendo, os Holons de Tarefa e de Recurso são descritos com maior detalhe que os restantes.

O arquétipo de holon apresentado na secção 5.2.2 é bastante genérico. Os holons a considerar nas subsecções seguintes denotam especializações desse arquétipo; ou mais concretamente, agentes de *software*, reactivos (orientados por mensagens) com estado interno e com funcionalidades definidas aquando da concepção do sistema.

6.3.1 Arquétipo de Holon

Esta secção apresenta funcionalidades comuns a todos os holons, principalmente relacionadas com o processamento de mensagens para gestão das holarquias. No âmbito deste trabalho, apenas são focadas as holarquias pré-definidas na arquitectura.

Além do tratamento de mensagens para gestão de holarquias, todos os holons possuem uma funcionalidade que lhes permite sinalizar ao Holon de Serviços de Directório que estão em execução, enviando, a intervalos regulares, uma mensagem com o seguinte teor:

bater_do_coracao

A Figura 6.17 corporiza o algoritmo de processamento de mensagens de gestão de holarquias existente em todos os holons.

```

func_msg_hms( $\gamma$ ,  $\mathcal{M}$ )  $\stackrel{\text{def}}{=}$ 
  seja  $\gamma$  a identificação do holon interlocutor
  seja  $\mathcal{M}$  uma mensagem enviada pelo holon  $\gamma$ 
  seja  $Act$  a identificação da operação a executar na lista de membros
  seja  $\eta$  a identificação de um holon
  seja  $H$  a identificação de uma holarquia
  início
    se  $\mathcal{M} = \text{hms\_adicionar}(\gamma)$  então
       $Membros \leftarrow \text{trata\_adicionar}(\gamma)$ 
       $\text{enviar\_msg}(\gamma, \text{r\_hms\_adicionar}(\gamma, Membros))$ 
    senão se  $\mathcal{M} = \text{hms\_remover}(\gamma)$  então
       $\text{trata\_remover}(\gamma)$ 
    senão se  $\mathcal{M} = \text{hms\_act\_membros}(Act, \eta, H)$  então
       $\text{trata\_act\_membros}(H, Act, \eta)$ 
    fim se
  fim

```

Figura 6.17 – Algoritmo de processamento de mensagens de gestão de holarquias

As mensagens referentes à gestão de holarquias são as seguintes:

- *hms_adicionar*(*Id*) – indica que um holon pretende adicionar-se a esta holarquia;
- *hms_remover*(*Id*) – indica que um holon pretende retirar-se desta holarquia;

- *hms_act_membros(Id-Holarquia, Operação, Id-Holon)* – indica que uma holarquia sofreu uma alteração (adição ou remoção) na sua lista de membros.

O programa em lógica apresentado em seguida, materializa as actividades do algoritmo anterior:

```

trata_adicionar(Holon, Membros) ←
    todo(Holarquia) ∧
    holarquia(Holarquia, Membros) ∧
    actualiza_holarquia(Holarquia, [Holon | Membros]) ∧
    avisa_todos(hms_act_membros(adicionar, Holon, Holarquia), Membros) ∧
    enviar_msg(Holon, r_hms_adicionar(Holon, Membros, ok))

trata_remove(Holon) ←
    todo(Holarquia) ∧
    holarquia(Holarquia, Membros) ∧
    retira_elemento(Holon, Membros, NovosMembros) ∧
    actualiza_holarquia(Holarquia, NovosMembros) ∧
    avisa_todos(hms_act_membros(remove, Holon, Holarquia), Membros)

trata_act_membros(Holarquia, adicionar, Holon) ←
    parte_de(Holarquia) ∧
    holarquia(Holarquia, Membros) ∧
    actualiza_holarquia(Holarquia, [Holon | Membros])
trata_act_membros(Holarquia, remove, Holon) ←
    parte_de(Holarquia) ∧
    holarquia(Holarquia, Membros) ∧
    retira_elemento(Holon, Membros, NovosMembros) ∧
    actualiza_holarquia(Holarquia, NovosMembros)

```

No programa anterior, o predicado *actualiza_holarquia* modifica a base de conhecimento do holon referente aos membros de uma dada holarquia e o predicado *avisa_todos* envia uma mensagem de alteração da constituição da holarquia aos seus membros constituintes. Esses holons actualizarão a informação referente à constituição da holarquia ao processarem a mensagem *hms_act_membros*.

A operação dos holons para integração nas holarquias é bastante simplificada, sendo ignorados aspectos como a verificação de identidade do holon que executa o pedido de adesão a uma holarquia e o seu “direito” de pertença a essa holarquia. Nesta implementação todos os holons que desejem ser adicionados ou removidos de uma holarquia são aceites ou retirados, partindo-se do pressuposto que todos os holons fornecem informação válida; *i.e.*, todos os holons cumprem o pressuposto de veracidade. Em situações mais próximas da realidade, seria necessário validar o pedido de adesão, verificando se o holon requerente pode ou deve de facto pertencer à holarquia.

6.3.2 Serviço de Directório

O *Holon de Serviços de Directório* é um auxiliar, funcionando como uma base de dados centralizada com informação sobre a identificação de holons e anúncio das suas reais potencialidades [Sousa *et al.*, 1999b].

A Figura 6.18 corporiza o algoritmo para o funcionamento do Holon de Serviços de Directório. Nesta implementação são ignorados aspectos de segurança, tais como verificação da identidade de quem faz o registo e principalmente de quem faz a remoção do registo, admitindo-se sempre que a informação é válida e de confiança (pressuposto de veracidade).

```

func_directorio()  $\stackrel{\text{def}}{=}$ 
  seja  $\gamma$  a identificação do holon interlocutor
  seja  $\mathcal{M}$  uma mensagem enviada pelo holon  $\gamma$ 
  seja  $\mathcal{H}$  a lista de habilidades do holon  $\gamma$ 
  seja  $h$  a identificação de uma habilidade
  seja  $\eta$  a identificação de um holon
  seja  $\mathcal{V}$  o valor de verdade da prova de um axioma
  início
    enquanto verdadeiro fazer
      receber_msg( $\gamma, \mathcal{M}$ )
      se  $\mathcal{M} = \text{registra}(\gamma, \mathcal{H})$  então
        regista( $\gamma, \mathcal{H}$ )
      senão se  $\mathcal{M} = \text{remover\_registo}(\gamma)$  então
        apaga_holon( $\gamma$ )
      senão se  $\mathcal{M} = \text{faz}(\eta, h)$  então
        demo fornece( $\eta, h$ )  $\mapsto \mathcal{V}$ 
        enviar_msg( $\gamma, \text{r\_faz}(\eta, h, \mathcal{V})$ )
      senão se  $\mathcal{M} = \text{o\_que\_faz}(\eta)$  então
        o_que_faz( $\eta, LH$ )
        enviar_msg( $\gamma, \text{r\_o\_que\_faz}(\eta, LH)$ )
      senão se  $\mathcal{M} = \text{quem\_faz}(h)$  então
        quem_faz( $h, LH$ )
        enviar_msg( $\gamma, \text{r\_quem\_faz}(h, LH)$ )
      senão se  $\mathcal{M} = \text{bater\_do\_coracao}$  então
        trata_bater_do_coracao( $\gamma$ )
      senão se  $\mathcal{M} = \text{em\_execucao}$  então
        quem_esta_em_execucao( $LH$ )
        enviar_msg( $\gamma, \text{r\_em\_execucao}(LH)$ )
      fim se
    fim ciclo
  fim

```

Figura 6.18 – Algoritmo de funcionamento do holon de serviços de directório

A interacção com este holon é feita através de um pequeno e simples protocolo de pedido-resposta de acordo com as seguintes produções (mensagens):

- *registra*(*Id*, *ListaFuncionalidades*) – anuncia o potencial de um holon existente no sistema;
- *remover_registo*(*Id*) – remove informação sobre um holon no sistema;
- *faz*(*Id*, *Funcionalidade*) – permite saber se um holon fornece ou não um determinado serviço;
- *o_que_faz*(*Id*) – permite conhecer o potencial de outros holons no sistema;
- *quem_faz*(*Funcionalidade*) – permite obter a lista de holons do sistema capazes de fornecer determinado serviço;
- *bater_de_coracao* – anuncia a disponibilidade de um holon, sinalizando que o holon se encontra em execução;
- *em_execucao* – permite obter a lista de holons registados no sistema e em execução.

Quando o Holon de Serviços de Directório recebe uma mensagem (*i.e.*, conjunção de termos lógicos ou teoremas) vai tentar obter a partir da sua base de conhecimento o valor de verdade correspondente ao pedido e enviar o resultado ao interlocutor. O programa em lógica dado a seguir materializa esta situação.

```

registra(H, LHab) ←
    ¬servicos(H, _) ∧
    adiciona_holon(H, LHab)
registra(H, LHab) ←
    actualiza_holon(H, LHab)

quem_faz(Hab, LH) ←
    fornecedores(Hab, LH)
quem_faz(Hab, [])

o_que_faz(H, LHab) ←
    servicos(H, LHab)
o_que_faz(H, [])

trata_bater_do_coracao(Holon) ←
    agora(Instante),
    adiciona_actualiza_em_exec(Holon, Instante)

quem_esta_em_execucao(LHolons) ←
    agora(Instante),
    todas_as_solucões(H, realmente_vivo(Instante, H), LHolons)

realmente_vivo(Actual, H) ←
    esta_vivo(H, UltimoBater) ∧

```

$$\text{atribui}(Z, \text{Actual} - \text{UltimoBater}) \wedge \\ \text{realmente_vivo_aux}(Z, H)$$

$$\text{realmente_vivo_aux}(Z, _) \leftarrow \\ Z < 300 \\ \text{realmente_vivo_aux}(_, H) \leftarrow \\ \text{remover_em_exec}(H)$$

No programa anterior *registra*(*H*, *LHAB*) acrescenta ou actualiza a base de conhecimento com informação sobre as operações que um holon pode executar (através dos predicados *adiciona_holon* e *actualiza_holon*). O predicado *apaga_holon* remove da base de conhecimento todos os itens de informação referentes a um dado holon. O predicado *quem_faz*(*HAB*, *LH*) tem como função dar a conhecer a existência na base de conhecimento de axiomas do tipo *fornecedores* para a operação em questão. De modo semelhante, *o_que_faz*(*H*, *LHAB*) dá a conhecer a existência ou não na base de conhecimento de axiomas do tipo *serviços* para o holon em questão. O predicado *trata_bater_do_coracao*(*Holon*) acrescenta ou actualiza a base de conhecimento do Holon de Serviços de Directório com informação sobre o instante em que um dado holon se sinalizou como estando em execução, usando para tal termos do tipo *esta_vivo*(*Holon*, *Instante*). Finalmente o predicado *quem_esta_vivo*(*Lholons*) tem como função dar a conhecer todos os holons registados no sistema, que se encontram em execução e sinalizarem esse facto há menos de um determinado tempo (e.g., 300 segundos). Caso esse tempo seja ultrapassado, a informação sobre esse holon estar em execução é removida da base de conhecimento.

6.3.3 Holon de Produto

Um *Holon de Produto* denota um item do catálogo de produtos fabricados pela empresa, sendo essencialmente responsável por fornecer informação relativa ao processo de fabrico aos outros holons do sistema [Sousa *et al.*, 1999b].

A interacção com este holon é feita através de um simples protocolo do tipo pedido-resposta, de acordo com as produções (mensagens):

- *obter_arvore* – permite obter a árvore de composição de cada produto a fabricar;
- *obter_BOM* – permite obter a lista de material para execução de 1 (um) produto de um dado tipo;
- *obter_planos*(*Critério*) – permite obter os planos de produção do produto de acordo com critério(s) preestabelecidos (e.g., critérios de qualidade).

A Figura 6.19 denota o algoritmo que corporiza o funcionamento de um Holon de Produto (apenas estão referidas as funcionalidades necessárias para o Planeamento de Produção e Escalonamento).

```
func_produto ()  $\stackrel{\text{def}}{=}$   
  seja  $\gamma$  a identificação do holon interlocutor  
  seja  $\mathcal{M}$  uma mensagem enviada pelo holon  $\gamma$   
  seja  $Crit$  um critério de otimização de um plano de produção  
  início  
    enquanto verdadeiro fazer  
      receber_msg( $\gamma$ ,  $\mathcal{M}$ )  
      se  $\mathcal{M} = \text{obter\_planos}(Crit)$  então  
        obter_planos( $Crit$ ,  $LP$ )  
        enviar_msg( $\gamma$ , r_obter_planos( $C$ ,  $LP$ ))  
      senão se  $\mathcal{M} = \text{obter\_arvore}$  então  
        expande_arvore( $Arv$ )  
        enviar_msg( $\gamma$ , r_obter_arvore( $Arv$ ))  
      senão se  $\mathcal{M} = \text{obter\_BOM}$  então  
        expande_BOM( $BOM$ )  
        enviar_msg( $\gamma$ , r_obter_BOM( $BOM$ ))  
      fim se  
    fim ciclo  
  fim
```

Figura 6.19 – Algoritmo de funcionamento de um holon de produto

Quando o Holon de Produto recebe uma mensagem vai tentar obter a partir da sua base de conhecimento o valor de verdade correspondente ao pedido e enviar o resultado ao interlocutor. O programa em lógica dado a seguir materializa esta situação.

```
obtem_planos( $C$ ,  $LP$ )  $\leftarrow$   
  todas_as_soluções(plano( $Id$ ,  $C$ ,  $A$ ,  $LOp$ ), plano( $Id$ ,  $C$ ,  $A$ ,  $LOp$ ),  $LP$ )  
  
expande_arvore( $Arv$ )  $\leftarrow$   
  todas_as_soluções( $s(S$ ,  $C$ ,  $Q)$ , composicao( $S$ ,  $C$ ,  $Q$ ),  $L$ )  $\wedge$   
  trata_arv( $L$ ,  $Arv$ )  
  
trata_arv([], [])  
trata_arv([ $s(0$ ,  $C$ ,  $Q)$  |  $T$ ], [componente( $C$ ,  $Q)$  |  $T2$ ])  $\leftarrow$   
  trata_arv( $T$ ,  $T2$ )  
trata_arv([ $s(P$ ,  $0$ ,  $Q)$  |  $T$ ], [produto( $P$ ,  $Q$ ,  $Arv$ ) |  $T2$ ])  $\leftarrow$   
  enviar_msg(id_srv_dir, quem_faz(produto( $P$ )))  $\wedge$   
  receber_msg(id_srv_dir, r_quem_faz(produto( $P$ ),  $HP$ ))  $\wedge$   
  enviar_msg( $HP$ , obter_arvore)  $\wedge$   
  receber_msg( $HP$ , r_obter_arvore( $Arv$ ))  $\wedge$   
  trata_arv( $T$ ,  $T2$ )  
  
expande_BOM( $BOM$ )  $\leftarrow$   
  todas_as_soluções( $s(S$ ,  $C$ ,  $Q)$ , composicao( $S$ ,  $C$ ,  $Q$ ),  $L$ )  $\wedge$   
  trata_BOM( $L$ ,  $X$ )  $\wedge$   
  agrega_BOM( $X$ , [],  $BOM$ )
```

```

trata_BOM([], [])
trata_BOM([s(0, C, Q) | T], [s(C, Q) | T2]) ←
    trata_BOM(T, T2)
trata_BOM([s(P, 0, Q) | T], T2) ←
    enviar_msg(id_srv_dir, quem_faz(produto(P))) ∧
    receber_msg(id_srv_dir, r_quem_faz(produto(P), HP)) ∧
    enviar_msg(HP, obter_BOM) ∧
    receber_msg(HP, r_obter_BOM(L1)) ∧
    mult_BOM(L1, Q, L2) ∧
    trata_BOM(T, LTemp) ∧
    juntar(Ltemp, L2, T2)

```

No programa anterior, o predicado *obtem_planos(C, LP)* permite que se construa uma lista com os planos de produção de um certo produto, que por sua vez obedecem a um determinado critério de optimização. O predicado *expande_arvore(Arv)* corporiza a árvore de composição desse produto, em que cada nó da árvore é um componente ou uma subárvore de um produto. O predicado *expande_BOM(BOM)* permite que se obtenha a lista de componentes utilizados no produto e subprodutos com as respectivas quantidades. Estes predicados recorrem a alguns predicados auxiliares, nomeadamente, *mult_BOM(Lista-Original, Quantidade, List-Final)* que devolve a lista de materiais de um subproduto com as quantidades actualizadas do número de componentes necessários e *agrega_BOM(Lista-Original, Aux, Lista-Final)* que agrega as quantidades dos elementos correspondentes a um mesmo componente.

6.3.4 Holon de Escalonamento

A operação do *Holon de Escalonamento* descrita nesta secção cinge-se ao tratamento de conflitos, nomeadamente, na resolução do Problema de Indecisão.

A Figura 6.20 denota o algoritmo que materializa o funcionamento do Holon de Escalonamento.

```

func_trata_conflitos() def
  seja  $\gamma$  a identificação do holon interlocutor
  seja  $\mathcal{M}$  uma mensagem enviada pelo holon  $\gamma$ 
  início
    enquanto verdadeiro fazer
      receber_msg( $\gamma$ ,  $\mathcal{M}$ )
      se  $\mathcal{M} = \text{lista\_recursos}(LR, TW)$  então
        se  $\text{possivel}(LR, TW)$  então
          enviar_msg( $\gamma$ , luz_verde)
          act_em_negociação ( $LR, \gamma, TW$ )
        senão
          coloca_em_espera( $\gamma, LR, TW$ )
        fim se
      senão se  $\mathcal{M} = \text{fim\_de\_negociacao}$  então
         $LR \leftarrow \text{obter\_estado}(\gamma)$ 
        libertar_tarefas( $\gamma, LR$ )
      senão se  $\mathcal{M} = \text{abandona\_recursos}(LR)$  então
        libertar_tarefas( $\gamma, LR$ )
      fim se
    fim ciclo
  fim

```

Figura 6.20 – Algoritmo de funcionamento do holon de escalonamento

Quando o Holon de Escalonamento recebe uma mensagem *lista_de_recursos* vai verificar se é *possível* iniciar a negociação com esses recursos e, em caso afirmativo, envia ao Holon de Tarefa a mensagem de “Luz Verde”, actualizando o estado dos recursos em negociação. Em caso negativo, essa tarefa é colocada em fila de espera. Este processo é descrito pelo programa em lógica dado a seguir.

```

possivel([], _)
possivel([Rec | T], TW) ←
  em_negociacao(Rec, LRecTW) ∧
  possivel_um_recurso(TW, LRecTW) ∧
  possivel(T, TW)
possivel([Rec | T], TW) ←
  ¬em_negociacao(Rec, _) ∧
  possivel(T, TW)

possivel_um_recurso(_, [])
possivel_um_recurso(TW, [s(Tarefa, RecTW) | T]) ←
  ¬sobrepoe(TW, RecTW) ∧
  possivel_um_recurso(TW, T)

```

No programa anterior o predicado *em_negociacao* verifica se um dado recurso se encontra actualmente em negociação e em caso afirmativo devolve a lista de intervalos em negociação e o predicado *sobrepoe* verifica se dois intervalos de tempo são sobrepostos.

Caso receba informação sobre o fim de negociação ou o abandono de recursos, o Holon de Escalonamento vai actualizar o estado de negociação desses recursos e libertar eventuais tarefas em fila de espera (Figura 6.21).

```

func_liberar_tarefas( $\gamma$ , LR)  $\stackrel{\text{def}}{=}$ 
  seja  $\gamma$  a identificação do holon interlocutor
  seja LR o conjunto de recursos libertados
  início
    liberar(LR,  $\gamma$ )
     $\langle t, LRe, TWe \rangle \leftarrow \text{pode\_libertar}()$ 
    enquanto  $t \neq \text{nenhuma}$  fazer
      enviar_msg( $t$ , luz_verde)
      act_em_negociação(LRe, TWe)
       $\langle t, LRe, TWe \rangle \leftarrow \text{pode\_libertar}()$ 
    fim ciclo
  fim

```

Figura 6.21 – Algoritmo de funcionamento do holon de escalonamento: libertar tarefas

Com este procedimento é possível começar-se por *libertar* os recursos associados à tarefa em questão e, de seguida tentar libertar cada uma das tarefas em fila de espera, usando para tal o predicado *pode_liberar* que permite obter informação associada a uma tarefa em fila de espera que possa ser libertada face aos recursos agora livres, e, o predicado *act_em_negociação* que actualiza a lista de recursos em negociação.

6.3.5 Holon de Tarefa

Os *Holons de Tarefa* denotam as ordens de fabrico enviadas à instalação fabril para a execução de n itens de um determinado produto, sendo responsáveis por garantir o seu escalonamento [Sousa e Ramos, 1998] [Sousa *et al.*, 1999b].

Como o objectivo deste tipo de holons é o de efectuar o escalonamento da ordem de fabrico e monitorizar a sua execução, após o arranque, o holon passa, essencialmente, por dois estados (Figura 6.22):

- *em negociação*, onde é efectuado o escalonamento da tarefa através de uma negociação operação a operação com os vários recursos disponíveis;
- *em acompanhamento*, onde o holon vai recebendo informação da execução da tarefa dada pelos recursos utilizados.

A Figura 6.22 apresenta o diagrama de estado para os Holons de Tarefa.

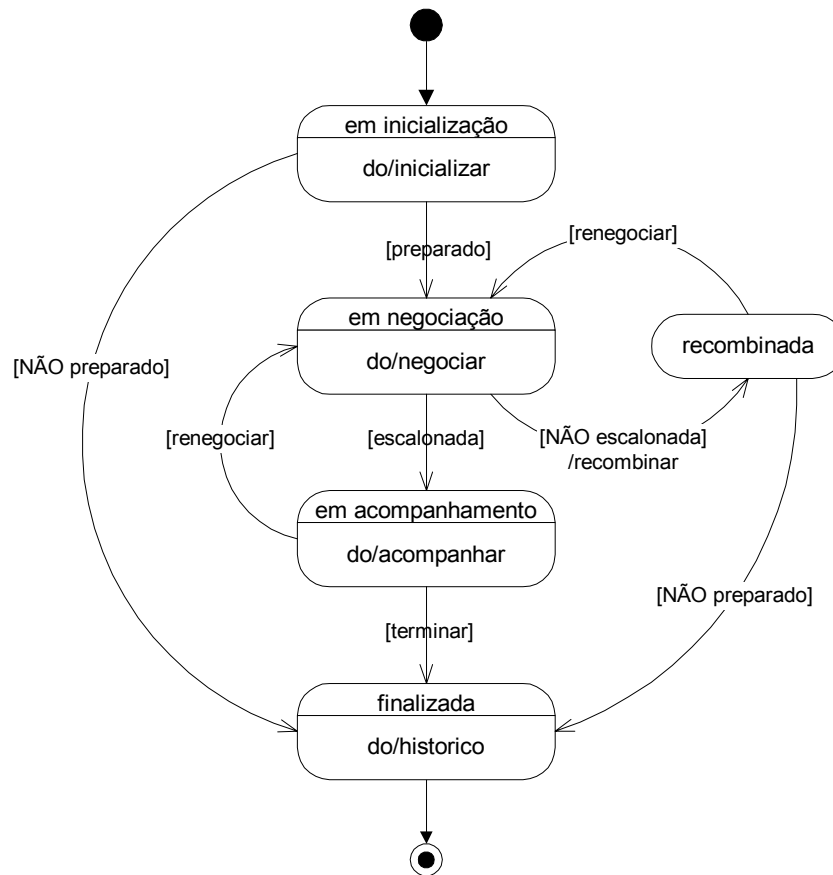


Figura 6.22 – Diagrama de estado para holons de tarefa

O funcionamento geral de um Holon de Tarefa é dado pelo algoritmo da Figura 6.23 que traduz o diagrama de estados apresentado na Figura 6.22.

```

func_tarefa() def
início
  inicializar
  se preparada então
    enquanto ¬terminar fazer
      negociar
      se ¬escalonada então
        recombinar
      senão
        acompanhar
      fim se
    fim ciclo
  fim se
  historico
fim
  
```

Figura 6.23 – Algoritmo de funcionamento de um holon de tarefa

O Holon de Tarefa começa por obter informação sobre o que fazer (algoritmo da Figura 6.24) e em seguida negocia a sua execução com os Holons de Recurso (algoritmo da Figura 6.25), recorrendo para tal a variante do PRCPR apresentada no capítulo anterior. As várias propostas recebidas dos Holons de Recurso são avaliadas e seleccionadas (algoritmo da Figura 6.26) e, caso não seja possível efectuar o escalonamento, o Holon de Tarefa irá recombina os recursos disponíveis para tentar nova negociação (algoritmo da Figura 6.27). Caso seja possível efectuar o escalonamento, o Holon de Tarefa acompanha a sua execução (Figura 6.28) e, no fim, guarda o seu estado na base de dados de histórico.

A Figura 6.24 descreve a fase de inicialização de um Holon de Tarefa.

```

tarefa_inicializar()  $\stackrel{\text{def}}{=}$ 
  seja HE a identificação do holon de escalonamento
  seja  $\mathcal{P}$  um plano de produção de um produto
  seja  $\mathcal{R}$  o conjunto de recursos necessários para efectuar o plano  $\mathcal{P}$ 
  início
    registar
    receber_msg(HE, anuncio(TId, NOF, PId, Qt, DataLimite, Attr))
    trata_anuncio(TId, NOF, PId, Qt, DataLimite, Attr)
    aderir_holarquias_pre_definidas
     $\mathcal{P} \leftarrow \text{obter\_plano}(\textit{PId}, \emptyset)$ 
     $\mathcal{R} \leftarrow \text{obter\_lista\_recursos}(\mathcal{P})$ 
    se  $\mathcal{R} \neq \emptyset$  então
      inserir preparada
    senão
      inserir  $\neg$ preparada
    fim se
  fim

```

Figura 6.24 – Algoritmo de funcionamento de um holon de tarefa: inicialização

Após a criação da ordem de fabrico e respectivo Holon de Tarefa pelo Holon de Escalonamento, o Holon de Tarefa recebe a mensagem *anuncio(TId, NOF, PId, Qt, DataLimite, Attr)* que especifica o que há a fazer. Após processar essa mensagem o holon regista-se no serviço de directório e adere às holarquias pré-definidas a que pertence; *i.e.*, a holarquia de escalonamento. Em seguida, obtém a informação necessária ao seu funcionamento, ou seja, o plano de produção e os recursos passíveis de serem utilizados em cada operação do plano. As várias funcionalidades presentes no algoritmo são dadas pelo programa em lógica a seguir.

```

trata_anuncio(TId, NOF, PId, Qt, DataLimite, Attr)  $\leftarrow$ 
  inserir(tarefa(TId, NOF, PId, Qt, DataLimite))  $\wedge$ 
  adiciona_attr(Attr)

adiciona_attr([])
adiciona_attr([a(P, V) | T])  $\leftarrow$ 
  inserir(atributo(P, V)  $\wedge$ 
  adiciona_attr(T)

```

```
obter_plano(PId, Attr) ←  
    enviar_msg(id_plan_processos, obter_plano(PId, Attr)) ∧  
    receber_msg(id_plan_processos, r_obter_plano(PId, Attr, Plano)) ∧  
    inserir(Plano)  
  
obter_lista_recursos(Operações-Plano, LRec) ←  
    constroi_LR(Plano, LRec) ∧  
    validar_LR(LRec)  
obter_lista_recursos(_, [])  
  
constroi_LR([], [])  
constroi_LR([nodo(Id, Op, rec(H), _, _, _) | Plano], [(Op, [H]) | LR]) ←  
    constroi_LR(Plano, LR)  
constroi_LR([nodo(Id, Op, oper, _, _, _) | Plano], [(Op, LH) | LR]) ←  
    enviar_msg(id_srv_dir, quem_faz(operacao(Op))) ∧  
    receber_msg(id_srv_dir, r_quem_faz(operacao(Op), LH)) ∧  
    constroi_LR(Plano, LR)
```

No programa anterior *trata_anuncio* processa a mensagem de *anuncio* com informação de carácter geral sobre a tarefa. A actividade *obter_plano* contacta o Holon de Planeamento de Processos para a obtenção do plano de produção do produto em questão. Finalmente, a actividade *obter_lista_recursos* constrói a lista de recursos a contactar para cada uma das operações do plano.

O algoritmo que corporiza o funcionamento dos Holons de Tarefa para a fase de negociação é apresentado na Figura 6.25.

```

 tarefa_negociar() def
  seja  $\Lambda$  o conjunto de recursos contactados para cada operação do plano  $\mathcal{P}$ 
  seja  $\Theta$  o conjunto de propostas recebidas
  seja  $HE$  a identificação do holon de escalonamento
  seja  $\mathcal{R}$  o conjunto de recursos necessários para efectuar o plano  $\mathcal{P}$ 
  seja  $op$  uma operação do plano  $\mathcal{P}$ 
  seja  $\mathcal{P}$  o plano do produto a executar
  seja  $\mathcal{T}$  a lista de operações contratadas por este holon
  início
     $\Lambda \leftarrow \emptyset$ 
     $\Theta \leftarrow \emptyset$ 
    demo tarefa( $TId$ ,  $NOF$ ,  $PId$ ,  $Qt$ ,  $DataLimite$ )
    enviar_msg( $HE$ , lista_recursos( $\mathcal{R}$ ,  $JT$ ))
    receber_msg( $HE$ , luz_verde)
    para cada  $\langle op, LH \rangle \in \mathcal{R}$  fazer
       $Pred \leftarrow antecessores(op, \mathcal{P}, \mathcal{R})$ 
       $Succ \leftarrow sucessores(op, \mathcal{P}, \mathcal{R})$ 
      para cada  $h \in LH$  fazer
        enviar_msg( $h$ , pedido( $TId$ ,  $DataLimite$ ,  $op$ ,  $Qt$ ,  $JT$ ,  $Pred$ ,  $Succ$ ))
         $\Lambda \leftarrow \Lambda \cup \{ \text{contactado}(h, op) \}$ 
      fim ciclo
    fim ciclo
    para cada  $\text{contactado}(op, LH) \in \Lambda$  fazer
      para cada  $h \in LH$  fazer
        receber_msg( $h$ , proposta( $op$ ,  $Dur$ ,  $LComb$ ))
         $\Theta \leftarrow \Theta \cup \{ \text{prop}(h, op, Dur, LComb) \}$ 
      fim ciclo
    fim ciclo
     $\Phi \leftarrow \text{avaliar_ofertas}(\Theta)$ 
     $\mathcal{T} \leftarrow \text{enviar_respostas}(\Lambda, \Phi)$ 
    enviar_msg( $HE$ , fim_de_negociacao)
  fim

```

Figura 6.25 – Algoritmo de funcionamento de holon de tarefa: negociação

O Holon de Tarefa começa por entrar em contacto com cada um dos recursos capazes de efectuar cada uma das operações do plano, requisitando o serviço pretendido. Em seguida, o holon aguarda pelas propostas de cada Holon de Recurso contactado (uma proposta por cada pedido enviado, já que utiliza a variante do PRCPR). Após a recepção das propostas procede à sua avaliação e selecção, terminando com o envio de contratos para os Holons seleccionados para a execução de cada operação.

O envio das mensagens *pedido* aos Holons de Recurso exige o conhecimento dos holons contactados para as operações antecessoras e sucessoras. Essa informação é conseguida através do seguinte programa em lógica que define as extensões dos predicados *antecessores* e *sucessores* referenciados no algoritmo da Figura 6.25.

```

    antecessores(Op, Plano, LRec, LRecPred) ←
        no_plano(Op, Plano, nodo(I, Op, _, _, P, _)) ∧
        nos_operacoes(Plano, P, LOpPred) ∧
        quais_recursos(LRec, LOpPred, LRecPred)

    sucessores(Op, Plano, LRec, LRecSucc) ←
        no_plano(Op, Plano, nodo(I, Op, _, _, S)) ∧
        nos_operacoes(Plano, S, LOpSucc) ∧
        quais_recursos(LRec, LOpSucc, LRecPred)
    
```

Em que o predicado *no_plano*(Op, Plano, Nodo) implementa a pesquisa de um nó do plano correspondente a uma operação, *nos_operacoes*(Plano, ListaNos, ListaOperações) devolve a lista de operações correspondentes a uma lista de nós do plano e, *quais_recursos*(LRec, LOp, LRecOp) devolve a lista de recursos capazes de efectuar cada uma das operações de LOp, usando a informação recolhida na inicialização do holon (LRec).

Uma das actividades mais importantes dos Holons de Tarefa prende-se com a avaliação das propostas recebidas e escolha dos Holons de Recurso. A Figura 6.26 descreve o algoritmo que corporiza a actividade *avaliar_ofertas*.

```

    tarefa_avaliar_ofertas(Θ) ↦ Φ def
        seja Θ o conjunto de propostas recebidas
        seja Φ o conjunto de propostas avaliadas
        seja κ um critério de avaliação de propostas
        início
            Φ ← ∅
            Γ ← preprocessar(Θ)
            enquanto Φ = ∅ ∧ (κ ← critério_de_selecção()) ≠ cancelar fazer
                Φ ← escolhe-κ(Γ)
            fim ciclo
            se κ = cancelar então
                Φ ← ∅
            senão
                inserir escalonada
            fim se
        fim
    
```

Figura 6.26 – Algoritmo de funcionamento de holon de tarefa: avaliação de propostas

Esta função implementa uma forma de meta-selecção, permitindo que se efectuem escolhas das propostas com base em múltiplos critérios fornecidos por *critério_de_selecção*. Caso não seja possível encontrar uma solução segundo um determinado critério é tentado outro até estes se esgotarem.

Antes de efectuar a escolha é feito um pré-processamento ao conjunto de mensagens de proposta recebidas dos Holons de Recurso. Assim, a actividade *preprocessar* constrói uma estrutura de dados mais apropriada para a avaliação das propostas, que consiste numa lista de

produções do tipo $alt(Caminho, CustoTotal, LProp)$, onde $LProp$ denota um elemento na forma $prop(Rec, Op, Dur, ListaIntervalos, Custo)$. Desta lista são também eliminadas as soluções impossíveis, ou seja, aquelas para as quais pelo menos um dos recursos não tem intervalos de tempo livre.

De entre os critérios de selecção passíveis de serem utilizados na escolha das propostas, três são materializados a partir do programa em lógica dado a seguir: (i) primeira solução válida; (ii) solução com menor custo; e (iii) solução com maior folga temporal até à data limite de conclusão da tarefa.

$$\begin{aligned}
 &escolhe_primeiro([alt(Path, C, LProp) \mid _], ListaSel) \leftarrow \\
 &\quad atribui(LProp, ListaSel) \\
 &escolhe_menor_custo(ListaPreProc, ListaSel) \leftarrow \\
 &\quad menor_custo(ListaPreProc, alt(Path, CT, LProp)) \wedge \\
 &\quad atribui(LProp, ListaSel) \\
 &escolhe_maior_folga(ListaPreProc, ListaSel) \leftarrow \\
 &\quad limite_superior(TW, F) \wedge \\
 &\quad calc_folga(ListaPreProc, ListaPreProcFolga) \wedge \\
 &\quad maior_folga(ListaPreProcFolga, alt2(Path, Custo, Folga, LProp)) \wedge \\
 &\quad atribui(LProp, ListaSel) \\
 &atribui([], []) \\
 &atribui([prop(Rec, Op, D, [H \mid _], C) \mid T], [sel(Rec, Op, D, H, C) \mid LT]) \leftarrow \\
 &\quad atribui(T, LT)
 \end{aligned}$$

Neste programa recorre-se a outros predicados, tais como o predicado *menor_custo*, que devolve a proposta com menor custo; *limite_superior* que devolve o limite superior do intervalo no que respeita às temporais para a tarefa em causa; *calc_folga* que constrói uma estrutura de dados paralela à lista pré-processada, incluindo o valor da folga de cada proposta e *maior_folga* que à semelhança de *maior_custo* devolve a proposta com maior folga.

Através do predicado *atribui* escolhe-se o primeiro intervalo livre em cada proposta, utilizando-se assim uma filosofia de escalonamento “o mais cedo possível”. No entanto, seria possível fazer a escolha dos intervalos utilizando uma via alternativa, a de “o mais tarde possível”, que assim nos aproximaria do conceito JIT (*Just In Time*).

Após a selecção de propostas o Holon de Tarefa envia o resultado da avaliação que se fez das propostas aos diversos Holons de Recurso já contactados, enviando uma mensagem de contrato para todos os Holons de Recurso seleccionados e, uma mensagem de cancelamento para os restantes.

$$\begin{aligned}
 &enviar_respostas([], _) \\
 &enviar_respostas([contactado(RecId, Op) \mid T], LSel) \leftarrow \\
 &\quad membro(sel(RecId, Op, Dur, JT, Custo), LSel) \wedge
 \end{aligned}$$

```
antecessores_sel(LSel, Op, PredSel) ∧  
sucessores_sel(LSel, Op, SuccSel) ∧  
enviar_msg(RecId, contrata(Op, JT, PredSel, SuccSel)) ∧  
enviar_respostas(T, LSel)  
enviar_respostas([contactado(RecId, Op) | T], LSel) ←  
enviar_msg(RecId, cancelar(Op)) ∧  
enviar_respostas(T, LSel)
```

Caso o escalonamento da tarefa não seja possível, o Holon de Tarefa tenta uma nova negociação utilizando um plano alternativo (Figura 6.27). Para isso é enviada informação extra ao Holon de Planeamento de Processos sobre os planos já utilizados.

```
tarefa_recombinar()  $\stackrel{\text{def}}{=}$   
  seja  $\Pi$  o conjunto de planos já utilizados  
  seja  $\mathcal{P}$  um plano de produção de um produto  
  seja  $\mathcal{R}$  o conjunto de recursos necessários para efectuar o plano  $\mathcal{P}$   
  início  
     $\Pi \leftarrow \Pi \cup \mathcal{P}$   
     $\mathcal{P} \leftarrow \text{obter\_plano}(Pid, \Pi)$   
     $\mathcal{R} \leftarrow \text{obter\_lista\_recursos}(\mathcal{P})$   
    se  $\mathcal{R} \neq \emptyset$  então  
      inserir renegociar  
    senão  
      inserir  $\neg$ preparada  
    fim se  
  fim
```

Figura 6.27 – Algoritmo de funcionamento de um holon de tarefa: recombinar

Na Figura 6.28 é apresentado o diagrama de estado do holon para a operação *acompanhar*. Esta actividade prende-se com o acompanhamento da operação de fabrico dos produtos constantes na ordem de fabrico. A ligação com o equipamento produtivo sai fora do âmbito deste trabalho, pelo que esta actividade é especificada embora não seja implementada.

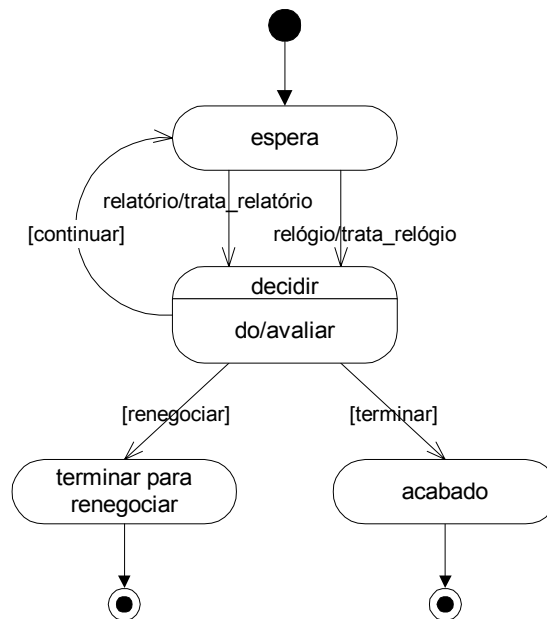


Figura 6.28 – Diagrama de estado para holons de tarefa: acompanhamento

O evento *relatório* está ligado à recepção de uma mensagem de um Holon de Recurso com informação sobre a execução de uma dada tarefa. O evento *relógio* denota um certo intervalo de tempo durante o qual não foi recebida nenhuma informação sobre o estado da execução da tarefa. A actividade *trata_relógio* envia a todos os Holons de Recurso que nada comunicaram sobre o estado em que se encontra a execução da tarefa, uma mensagem pedindo essa informação. A actividade *trata_relatório* actualiza a informação sobre a execução da tarefa, com base na informação recebida dos Holons de Recurso. A actividade *avaliar* verifica o estado de execução da tarefa comparando o estado actual com o que foi planeado, e age em conformidade: termina quando a tarefa for executada ou cancelada, ou renegoceia caso a execução da tarefa não esteja a ser cumprida de acordo com o contrato efectuado com os Holons de Recurso.

6.3.6 Holon de Recurso

Um *Holon de Recurso* denota o estado actual de um recurso (físico) da instalação fabril (*e.g.*, estado de funcionamento, actividades a efectuar) e é responsável pelo escalonamento de operações requisitadas pelos Holons de Tarefa [Sousa e Ramos, 1998] [Sousa *et al.*, 1999b].

As actividades contratadas por um Holon de Recurso estão representadas na sua agenda. Essa agenda pode ser visualizada como um gráfico de Gantt onde os rectângulos representam as operações contratadas a esse recurso. Para efectuar o escalonamento, o holon necessita de manipular a agenda, podendo efectuar sobre ela cinco operações: (i) *cálculo dos intervalos livres*;

(ii) *limite inferior*; (iii) *limite superior*; (iv) *influência directa*; (v) *influência inversa*. Estas operações são descritas em seguida.

O predicado *calc_intervalos_livres*(*AgOcupada*, *Inicio*, *Fim*, *Duração*, *AgLivre*) permite-nos obter a lista de intervalos livres do Holon de Recurso com uma determinada *Duração* mínima e atendendo à janela temporal limitada por *Inicio* e *Fim*. Esta lista pode também ser visualizada como um gráfico de Gantt, em que os rectângulos denotam tempos livres.

A Figura 6.29 dá uma caracterização pictórica desta operação para uma determinada agenda de actividades, considerando que a tarefa a executar ocupa *duas* (2) unidades de tempo, e ocorre entre os limites *a* e *b*.



Figura 6.29 – Agenda de um recurso e agenda livre

Na Figura 6.29 a primeira barra refere-se à agenda de actividades contratadas pelo holon (rectângulos escuros), ou seja, a sua agenda de intervalos ocupados. A agenda resultante tem os intervalos livres (rectângulos claros) entre os limites *a* e *b* com uma duração superior ao número de unidades de tempo necessárias (no exemplo em causa têm-se *duas*).

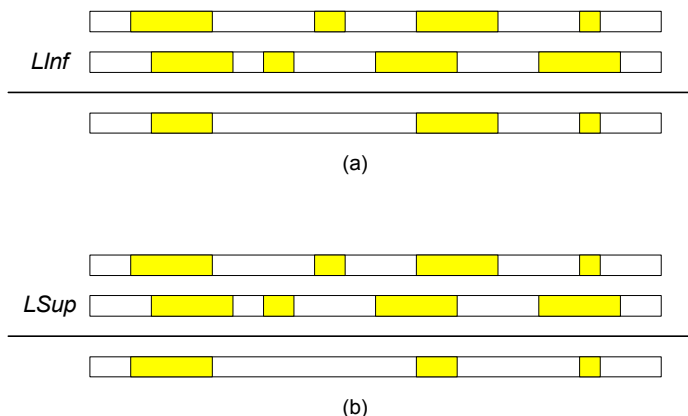


Figura 6.30 – Limitação de agendas: (a) inferior; (b) superior

A limitação de duas agendas permite criar uma nova agenda, cujos intervalos correspondem às zonas ocupadas em ambas as agendas (Figura 6.30), mas cujo limite inferior ou superior obedeça às restrições da segunda agenda. Esta operação é efectuada a partir das extensões dos predicados *limitacao_inferior*(*ListaA*, *ListaB*, *ListaResultado*) e *limitacao_superior*(*ListaA*, *ListaB*, *ListaResultado*).

As duas últimas operações sobre as agendas são a influência directa (Figura 6.31a) e a influência inversa (Figura 6.31b), que passam, respectivamente, por uma deslocação à direita ou à esquerda dos limites inferiores e superiores de cada intervalo da agenda.

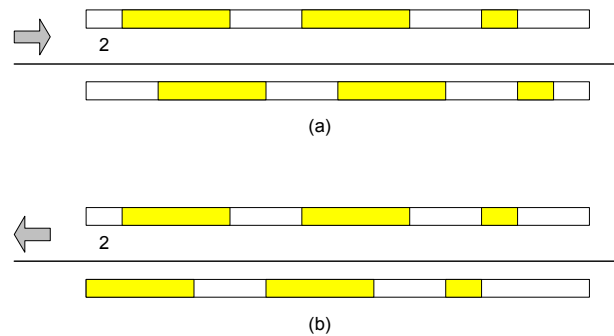


Figura 6.31 – Influência de agendas: (a) directa; (b) inversa

Estas operações são realizadas com recurso às extensões dos predicados *influencia_directa(ListaA, Duração, ListaResultado)* e *influencia_inversa(ListaA, Duração, ListaResultado)*.

Conforme o que foi referido na secção 5.3.4, os Holons de Recurso estão divididos a dois níveis: *software* e *hardware*. A Figura 6.32 apresenta o diagrama de estados para a componente de *software* de um Holon de Recurso.

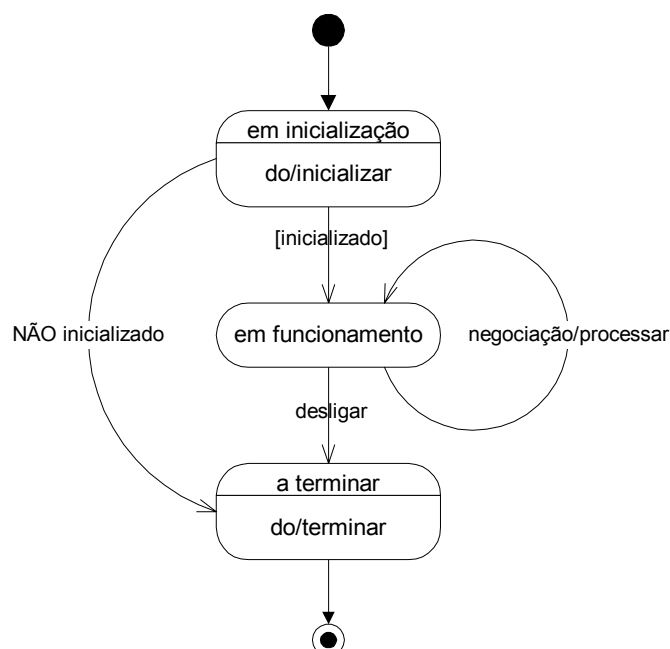


Figura 6.32 – Diagrama de estado para holons de recurso: planeamento

No momento da *inicialização*, o Holon de Recurso constrói a sua lista inicial de actividades, regista-se no serviço de directório e adere às holarquias pré-definidas a que pertence (*i.e.*, as holarquias de planeamento de processos, planeamento da produção e escalonamento). Em seguida, o holon *processa* os pedidos de *negociação* que lhe chegam dos Holons de Tarefa e dos Holons de Recurso, por forma a efectuar o escalonamento das operações inerentes à execução de uma dada tarefa (de acordo com o algoritmo apresentado na Figura 6.33).

```

func_recurso_processar() def
  seja  $\gamma$  a identificação de um holon consumidor
  seja  $\mathcal{M}$  uma mensagem recebida
  seja  $op$  uma operação a executar
  seja  $\mathcal{H}$  a lista de funcionalidades do holon
  seja  $\mathcal{A}$  o conjunto de tarefas contratadas pelo holon
  início
    enquanto  $\neg$ desligar fazer
      receber_msg( $\gamma$ ,  $\mathcal{M}$ )
      se  $\mathcal{M} = \text{pedido}(TId, DtLimite, op, Qt, tw(I, F), Pred, Succ)$  então
        se  $op \in \mathcal{H}$  então
          duracao_da_operacao( $op, Qt, DurI, DurQt$ )
           $Ag \leftarrow \text{calc\_intervalos\_livres}(\mathcal{A}, I, F, DurQt)$ 
          cria_estado( $TId, DtLimite, op, Qt, DurI, DurQt, tw(I, F),$ 
             $Pred, Succ$ )
          trata_pedido( $op, Qt, DurI, DurQt, Pred, Succ, Ag$ )
        senão
          enviar_msg( $\gamma$ , proposta( $op, 0, \emptyset$ ))
        fim se
      senão se  $\mathcal{M} = \text{influencia\_directa}(TId, op, DurI, LComb)$  então
        obter_estado( $TId, op, espera\_fw(Ag, CombAg, Pred)$ )
         $op\_pred\_maior(DurI, TId, op, LComb, LComb2),$ 
         $combinacoes(Ag, \{fw(\gamma, Op, Lcomb2)\}, LoI)$ 
         $retira\_elemento(\gamma, Pred, Pred\_Falta)$ 
         $trata\_inf\_directa(TId, op, Ag, \{LoI\} \cup CombAg, Pred\_Falta)$ 
      senão se  $\mathcal{M} = \text{influencia\_inversa}(TId, op, LComb)$  então
        obter_estado( $TId, op, espera\_bk(CombAg, Succ)$ )
         $combinacoes\_bk(CombAg, LComb, NovaCombAg)$ 
         $retira\_elemento(\gamma, Succ, Succ\_Falta)$ 
         $trata\_inf\_inversa(TId, op, NovaCombAg, Succ\_Falta)$ 
      senão se  $\mathcal{M} = \text{contrata}(op, IntervaloSel, SelPred, SelSucc)$  então
        obter_estado( $\gamma, op, Qt, \_, DurQt, DtLimite, \_, \_, \_$ ),
         $\langle i, f \rangle \leftarrow \text{escalonar}(IntervaloSel, op, DurQt)$ 
         $\mathcal{A} \leftarrow \mathcal{A} \cup \{ \text{actividade}(\gamma, op, Qt, i, f, DataLimite, por\_fazer,$ 
           $SelPred, SelSucc) \}$ 
        destruir_estado( $\gamma, op$ )
      senão se  $\mathcal{M} = \text{cancelar}(op)$  então
        destruir_estado( $\gamma, op$ )
      fim se
    fim ciclo
  fim

```

Figura 6.33 – Algoritmo de planeamento de holons de recurso

Após a recepção de um pedido, o Holon de Recurso actualiza a sua agenda de intervalos livres, e vê onde poderá escalonar a nova operação. De seguida, corre uma máquina de estados para a execução do PRCPR (dada pela Figura 6.34). Todo o funcionamento do Holon de Recurso rege-se pela execução desta máquina de estado para cada processo de negociação em que participa. Quando, finalmente, obtiver a lista final de intervalos livres para um pedido, o Holon de Recurso envia essa proposta ao Holon de Tarefa, esperando pela aceitação ou recusa da proposta.

A Figura 6.34 denota o diagrama de estados do Protocolo de Rede de Contrato com Propagação de Restrições (PRCPR) apresentado na secção 6.2 e executado pelos Holons de Recurso.

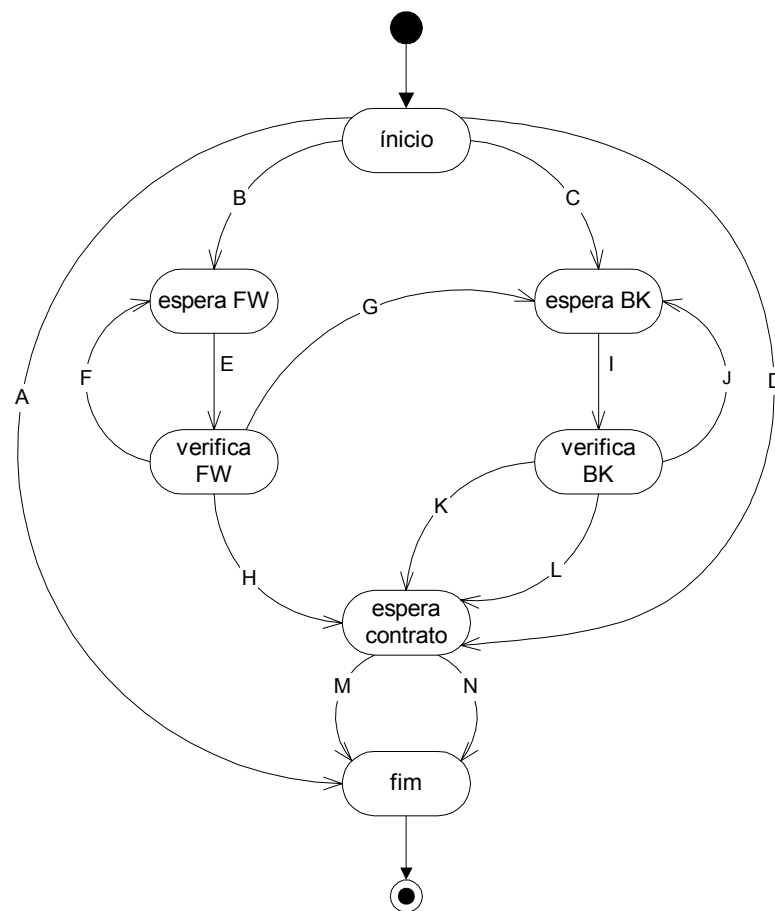


Figura 6.34 – Protocolo RCPR nos holons de recurso

Na Tabela 6.3 são apresentadas as transições desta máquina de estados, indicando-se as condições ou eventos que as viabilizam, bem como as acções a executar pelo Holon de Recurso em cada transição de estado.

Tabela 6.3 – Transições na máquina de estados do PRCPR nos holons de recurso

Transição	[Condição] Evento	Acção
A	[$Ag = \emptyset$]	Fazer proposta vazia
B	[$Ag \neq \emptyset \wedge Pred \neq \emptyset$]	–
C	[$Ag \neq \emptyset \wedge Pred = \emptyset \wedge Succ \neq \emptyset$]	Enviar <i>influencia_directa</i> aos sucessores
D	[$Ag \neq \emptyset \wedge Pred = \emptyset \wedge Succ = \emptyset$]	Fazer proposta
E	<i>influencia_directa</i>	Limitar a agenda actual com a lista influenciada de intervalos recebida dos antecessores, gerando uma combinação para cada possibilidade de combinação de recursos
F	[$falta(Pred) \neq \emptyset$]	–
G	[$falta(Pred) = \emptyset \wedge Succ \neq \emptyset$]	Enviar <i>influencia_directa</i> aos sucessores
H	[$falta(Pred) = \emptyset \wedge Succ = \emptyset$]	Enviar <i>influencia_inversa</i> aos antecessores e fazer proposta
I	<i>influencia_inversa</i>	Limitar cada combinação actual de recursos com a lista influenciada de intervalos recebida dos sucessores para a mesma combinação de recursos
J	[$falta(Succ) \neq \emptyset$]	–
K	[$falta(Succ) = \emptyset \wedge Pred = \emptyset$]	Fazer proposta
L	[$falta(Succ) = \emptyset \wedge Pred \neq \emptyset$]	Enviar <i>influencia_inversa</i> aos antecessores e fazer proposta
M	<i>contrata</i>	Escalonar na agenda de actividades
N	<i>Cancela</i>	–

O funcionamento da máquina de estados é concretizado através das extensões dos predicados *cria_estado*, *obter_estado*, *act_estado* e *destruir_estado*. Em que, *cria_estado* inicializa a máquina de estados e guarda alguma informação respeitante à negociação em questão; *obter_estado/3* permite conhecer qual o estado actual e *obter_estado/9* permite obter a informação

para uma dada negociação; *act_estado* transfere a máquina de estados para um novo estado e *destruir_estado* destrói toda a informação referente a uma negociação.

No algoritmo da Figura 6.33 a partir da invocação do predicado *combinacoes* geram-se as combinações de intervalos passíveis de utilização na execução da tarefa, de acordo com os recursos disponíveis. O predicado *combinacoes_bk* realiza a operação de limitação das listas influenciadas de intervalos livres entretanto recebidas com cada uma das combinações possíveis. A partir da extensão do predicado *op_pred_maior* trata-se o caso em que a operação antecessora tem uma duração maior que a operação requisitada ao recurso, pelo que é necessário obter uma nova lista influenciada de intervalos. Finalmente, o predicado *escalonar* determina o limite superior do intervalo em que a operação é escalonada tendo em conta a sua duração.

As acções a executar pelo Holon de Recurso para cada uma das mensagens recebidas são agora definidas através do programa em lógica apresentado em seguida.

```

trata_pedido(Tarefa, Op, _, _, _, []) ←
    enviar_msg(Tarefa, proposta(Op, 0, []))

trata_pedido(Tarefa, Op, _, DurQt, [], [], Ag_Livre) ←
    faz_oferta(Tarefa, Op, DurQt, Ag_Livre)

trata_pedido(Tarefa, Op, Dur1, _, [], Succ, Ag_Livre) ←
    influencia_directa(Ag_Livre, Dur1, Ag_Linha) ∧
    nome(EsteHolon) ∧
    enviar_fw_todos(Tarefa, [inf(Ag_Linha, [EsteHolon])], Succ) ∧
    act_estado(Tarefa, Op, espera_bk(Ag_Livre, Succ))

trata_pedido(Tarefa, Op, _, _, Pred, Succ, Ag_Livre) ←
    act_estado(Tarefa, Op, espera_fw(Ag_Livre, [], Pred))

faz_oferta(Tarefa, Op, D, Ag) ←
    enviar_msg(Tarefa, proposta(Op, D, Ag)) ∧
    act_estado(Tarefa, Op, espera_contrato(Ag))

```

O recurso à extensão do predicado *trata_pedido* ocorre quando o Holon de Recursos recebe uma mensagem *pedido*. A primeira instância do predicado materializa a situação em que o holon não possui intervalos livres para efectuar a operação. A segunda cláusula denota uma situação em que a operação requisitada é a única operação constante do plano (não tem antecessores nem sucessores). A terceira cláusula corporiza a situação em que o holon não tem antecessores, dando assim início à fase de influência directa. A quarta e última instância materializa a situação em que o holon tem antecessores, tendo por isso que esperar pelas mensagens de influência directa.

Através da invocação do predicado *faz_oferta* uma mensagem de proposta é enviada ao Holon de Tarefa e, por seu lado, através da invocação de *enviar_fw_todos* é enviada uma mensagem de influência directa a todos os recursos sucessores. Por recurso a *trata_inf_dir*

processam-se as mensagens de influência directa recebidas pelo holon. Estes acontecimentos são descritos pelo programa em lógica dado a seguir.

```

trata_inf_directa(Tarefa, Op, Ag, LComb, []) ←
    nivelar(LComb, LCombAg) ∧
    obter_estado(Tarefa, Op, _, Dur1, DurQt, _, _, Pred, Succ) ∧
    trata_inf_dir_aux(Tarefa, Op, LCombAg, Dur1, DurQt, Pred, Succ)

trata_inf_directa(Tarefa, Op, Ag, LComb, Pred_Falta) ←
    act_estado(Tarefa, Op, espera_fw(Ag, LComb, Pred_Falta))

trata_inf_dir_aux(Tarefa, Op, LCombFinalAg, Dur1, DurQt, Pred, []) ←
    enviar_bk_filtro(Tarefa, Op, Dur1, LCombFinalAg, Pred) ∧
    faz_oferta(Tarefa, Op, DurQt, LCombFinalAg)

trata_inf_dir_aux(Tarefa, Op, LCombAg, Dur1, _, Pred, Succ) ←
    influencia_directa(LCombAg, Dur1, LoI) ∧
    enviar_fw_todos(Tarefa, LoI, Succ) ∧
    act_estado(Tarefa, Op, espera_bk(LCombAg, Succ))
    
```

A primeira instância do predicado *trata_inf_directa* representa a situação em que todos os recursos antecessores já enviaram as mensagens de influência directa, tendo por isso o holon que iniciar a fase de influência inversa caso não tenha sucessores (primeira instância de *trata_inf_dir_aux*); ou então continuar a fase de influência directa caso possua sucessores (segunda instância de *trata_inf_dir_aux*). A segunda instância do predicado *trata_inf_dir* mantém o recurso à espera das restantes mensagens de influência directa dos antecessores em falta.

Por via da invocação do predicado *enviar_bk_filtro* envia-se uma mensagem de influência inversa a todos os recursos antecessores, tendo em conta o caminho que gerou cada combinação de intervalos livres. Internamente *enviar_bk_filtro* vai gerar uma agenda influenciada de acordo com a duração da operação deste recurso e a operação requisitada a cada recurso antecessor. A extensão do predicado *enviar_bk_filtro* é dada pelas produções (ou axiomas):

```

enviar_bk_filtro(_, _, _, _, [])
enviar_bk_filtro(Tarefa, MinhaOp, Dur1, LCombFinalAg, [Rec:Op | T]) ←
    dur_op_pred(Tarefa, MinhaOp, Rec, Op, DurOpPred) ∧
    max(Dur1, DurOpPred, Dur) ∧
    influencia_inversa(LCombFinalAg, Dur, LoI) ∧
    filtrar_bk(LoI, R, LBK) ∧
    enviar_msg(R, influencia_inversa(Task, Op, LBK)) ∧
    enviar_bk_filtro(Tarefa, MinhaOp, Dur1, LCombFinalAg, T)
    
```

a extensão do predicado *trata_inf_inv* é utilizada para o processamento das mensagens de influência inversa recebidas pelo holon, sendo dada pelas produções:

```

trata_inf_inversa(Tarefa, Op, LCombFinalAg, []) ←
    obter_estado(Tarefa, Op, _, Dur1, DurQt, _, _, Pred, _) ∧
    
```

$$trata_inf_inv_aux(Tarefa, Op, LCombFinalAg, Pred, DurI, DurQt)$$

$$trata_inf_inversa(Tarefa, Op, LCombAg, Succ_Falta) \leftarrow \\ act_estado(Tarefa, Op, espera_bk(LCombAg, Succ_Falta))$$

$$trata_inf_inv_aux(Tarefa, Op, LCombFinalAg, [], _, DurQt) \leftarrow \\ faz_oferta(Tarefa, Op, DurQt, LCombFinalAg)$$

$$trata_inf_inv_aux(Tarefa, Op, LCombFinalAg, Pred, DurI, DurQt) \leftarrow \\ enviar_bk_filtro(Tarefa, Op, DurI, LCombFinalAg, Pred) \wedge \\ faz_oferta(Tarefa, Op, DurQt, LCombFinalAg)$$

A primeira instância do predicado *trata_inf_inv* corporiza a situação em que todos os recursos sucessores já enviaram as mensagens de influência inversa, tendo por isso o holon que terminar esta fase caso não tenha antecessores (primeira instância de *trata_inf_inv_aux*); ou então continuar a fase de influência inversa caso possua antecessores (segunda instância de *trata_inf_inv_aux*). A invocação da segunda instância do predicado *trata_inf_inv* mantém o recurso à espera das restantes mensagens de influência inversa dos sucessores em falta.

Conforme o que foi referido anteriormente (secção 5.3.4) os Holons de Recurso estão divididos em dois níveis: *software* e *hardware*. A componente de *software* acabou de ser apresentada, quanto à componente de *hardware*, embora a ligação com o equipamento fabril saia fora do âmbito deste trabalho, esta pode ser traduzida pelo diagrama de estados da Figura 6.35.

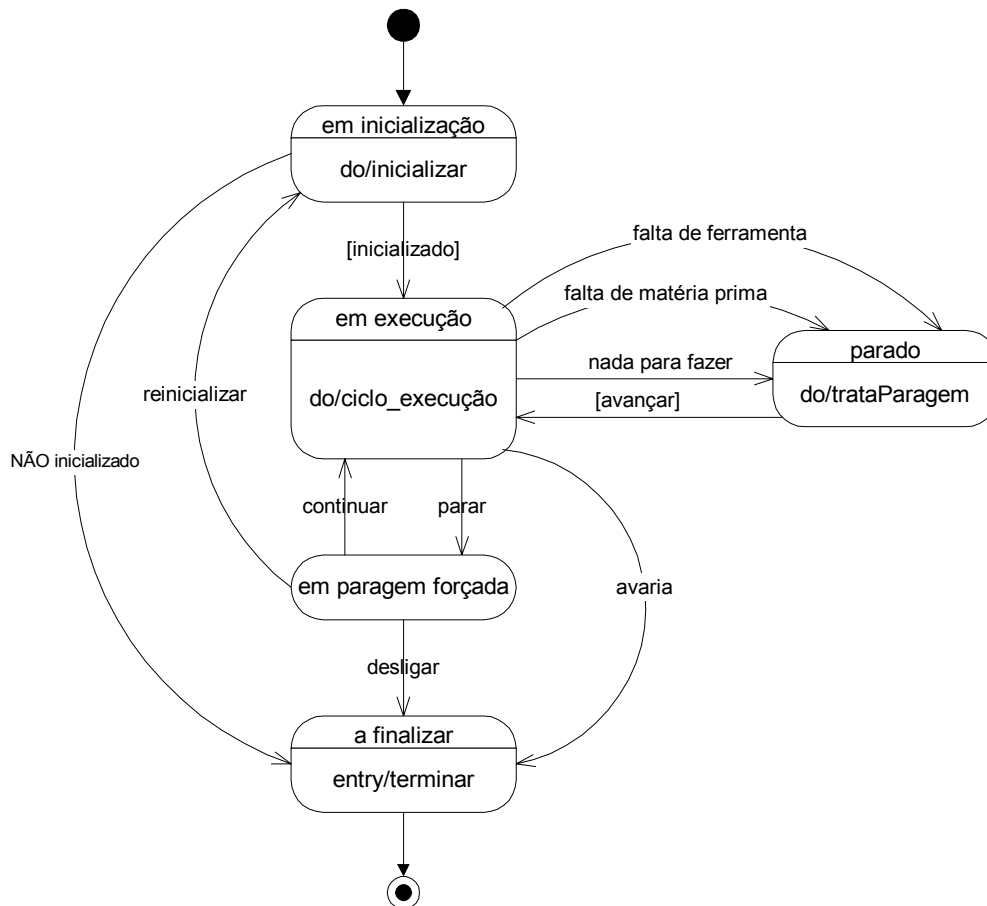


Figura 6.35 – Diagrama de estado para holons de recurso: execução

O evento *reinicializar* corporiza a operação de accionamento do botão de reinicialização do equipamento. O evento *parar*, por sua vez, denota o accionamento do botão de paragem de emergência, enquanto que o evento *continuar* indicia a anulação da operação de paragem de emergência (normalmente pressionando novamente o botão de paragem ou rodando-o). Os eventos *nada para fazer*, *falta de matéria prima* e *falta de ferramenta* denotam, respectivamente, as situações onde não há actividades a executar pelo recurso, não há matéria prima ou componentes necessários à continuação da operação em curso, e não há uma ferramenta indispensável à execução da operação em curso. O evento *desligar* refere-se à pressão do botão de desligar o equipamento. A actividade principal do recurso decorre no estado *em execução*, onde o recurso aguarda comandos enviados pelo dispositivo controlador do equipamento (e.g., carregar programa NC), executando-os de seguida.

6.3.7 Relação com o Método Original de Escalonamento

Conforme foi referido anteriormente, o caso de teste considerado para validar este trabalho foi a actividade de escalonamento tendo sido apresentado um protocolo de negociação e os algoritmos de funcionamento dos holons, tendo em vista o escalonamento de tarefas industriais.

O procedimento de escalonamento utilizado neste trabalho, resulta de uma abordagem *sui generis*, porventura distribuída, a um método original apresentado em [Almeida, 1995] e [Ramos *et al.*, 1995] para escalonamento dinâmico de tarefas industriais sujeitas a prazos de entrega.

O método original pode ser caracterizado pela aceitação da seguinte axiomática:

- utilização de uma *abordagem centralizada* com um único processo computacional, conhecendo todos os dados e efectuando todos os cálculos;
- a existência de *um único recurso capaz de efectuar cada operação*, não existindo por isso alternativas para a execução de cada operação;
- a *utilização de comportamentos* [Almeida, 1995] que definem um esquema fixo de escalonamento para as diversas operações do plano de produção, ou seja, o intervalo de tempo entre duas operações sucessivas do plano é fixo;
- não considera tempos de *setup* das máquinas, nem de transporte entre estações de trabalho;
- utilização de um *operador de intersecção* de agendas de intervalos para as fases de influência directa e inversa;
- *sugestão de intervalos de escalonamento* para um operador humano ou sistema de apoio à decisão.

A Figura 6.36a apresenta um comportamento de escalonamento para o fabrico de um produto podendo ver-se na Figura 6.36b a utilização desse comportamento para o escalonamento de uma tarefa com quantidade 3. Na Figura 6.36c pode ver-se um outro comportamento para a execução de três produtos do mesmo tipo considerando a existência de *buffers*.

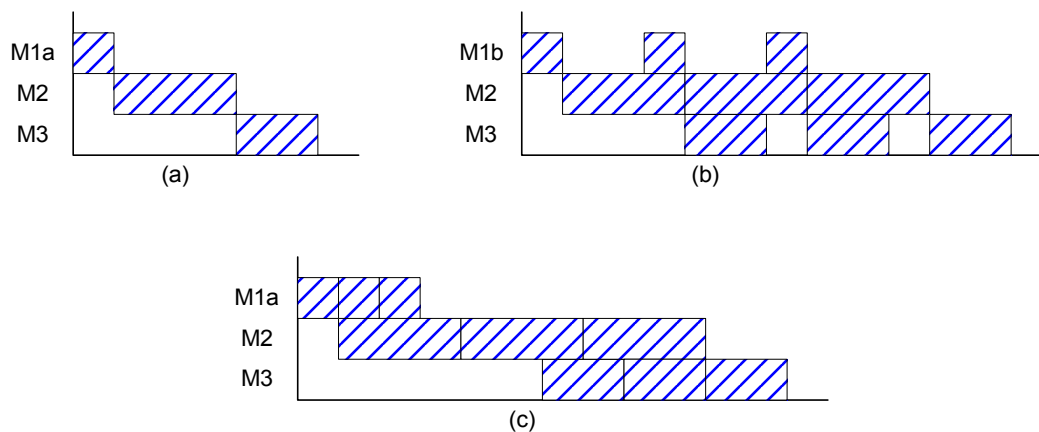


Figura 6.36 – Comportamentos de escalonamento

Conforme se pode ver através do que se expõe na Figura 6.36, o método original dependia da definição *a priori* dos comportamentos para o número exacto de produtos a fabricar, considerando o *buffer* com o comprimento necessário para evitar buracos no escalonamento.

O método de escalonamento utilizado neste trabalho pode ser dado pela axiomática:

- utilização de uma *abordagem distribuída* com vários processos computacionais (*i.e.*, os holons) efectuando parcelas do cálculo e conhecendo apenas parte dos dados do problema;
- utilização de *custos por operação por recurso* que contribui com mais um parâmetro para obtenção de soluções diferentes;
- possibilidade de existência de *múltiplos recursos para a execução de cada operação*, criando por isso várias alternativas para o escalonamento de uma tarefa tendo em conta as agendas de actividades de cada recurso e o tempo e o custo de execução de cada operação nos vários recursos alternativos;
- possibilidade de utilização de *um mesmo recurso em mais que uma operação* do plano;
- *não utilização de comportamentos*, o que permite maior flexibilidade no tempo entre operações, mas implica a existência de *buffers* (assumidos de tamanho infinito);
- não consideração de tempos de *setup* das máquinas nem de transporte entre estações de trabalho;
- utilização de um *operador para o cálculo dos limites inferior e superior* de agendas de intervalos para as fases de influência directa e inversa;
- *escolha de intervalos de escalonamento* com base em algum critério de optimização previamente definido.

Em relação ao método original podem considerar-se duas grandes melhorias, a existência de recursos alternativos para cada operação e a não utilização de comportamentos de escalonamento. A primeira, embora aumente a complexidade do problema, gerando nalguns casos uma explosão combinatória do espaço de soluções, aproxima-se mais da realidade industrial, pois cada vez mais as máquinas poderão executar um maior número de operações e cada vez mais será maior a sobreposição de funcionalidades (*i.e.*, as operações possíveis de efectuar) entre recursos. A segunda melhoria permite evitar o passo prévio de geração de comportamentos evitando o aparecimento de buracos no escalonamento em situações semelhantes à da Figura 6.36b. Adicionalmente, o intervalo de tempo entre operações sucessivas não é fixo o que permite uma maior flexibilidade no escalonamento e a utilização de intervalos de tempo que nunca seriam possíveis com o método original. Por outro lado, neste método assume-se sempre a existência de *buffers* de tamanho infinito. No método original, os comportamentos permitiam modelar a existência de *buffers* de qualquer tamanho (mas com a rigidez no escalonamento que lhes está associada).

Por fim, a maior diferença com o método original prende-se com a abordagem utilizada. Ao passar de uma abordagem centralizada para uma descentralizada tem-se vantagens e desvantagens. As principais vantagens relacionam-se com uma maior aproximação da realidade em termos de entidades modeladas e a diminuição de pontos de falha, já que o procedimento de escalonamento se encontra replicado em cada unidade recurso e tarefa. As principais desvantagens prendem-se com a velocidade de execução (principalmente devido à necessidade de comunicações).

No Apêndice A são apresentados os resultados de várias experiências efectuadas com o protótipo desenvolvido neste trabalho em comparação com os resultados do método original. Além dessas experiências são também apresentadas novas experiências para demonstrar a utilização de recursos alternativos.

6.4 Sistema Desenvolvido

Esta secção descreve o sistema informático concebido como protótipo para sustentação dos conceitos e arquitectura apresentados na tese. O protótipo, denominado *Fabricare* [Sousa *et al.*, 2000a], consiste numa série de aplicações informáticas para cada um dos diversos holons da arquitectura, bem como em alguns programas de suporte, tendo por objectivo o escalonamento de tarefas industriais.

6.4.1 Introdução

No Capítulo 5 foi apresentada uma arquitectura holónica para sistemas de produção e cada um dos holons constituintes dessa arquitectura foi especificado em termos de objectivos, ciclo de vida e, base de conhecimento, tendo sido em seguida descrito o seu modo de funcionamento nas secções anteriores deste capítulo.

A partir do exposto nesses capítulos partiu-se então para o desenvolvimento de um sistema computacional ao qual se deu o nome de *Fabricare* (Figura 6.37). O termo *Fabricare* é a palavra em Latim com o significado de fabricar. O sistema *Fabricare* é composto por várias aplicações e foi desenvolvido utilizando a linguagem de programação em lógica SICStus PROLOG para a implementação dos agentes do sistema e o Visual Basic para a construção da interface com o utilizador. Cada aplicação será apresentada com maior detalhe nas subsecções seguintes. A aplicação *Configuration Designer* permite definir os recursos existentes na instalação fabril e até certo ponto a disposição desses recursos na instalação fabril. Esta aplicação gera uma descrição do sistema que é utilizada pela aplicação *Deploy* para colocar em funcionamento os *Holons de Recursos*. O *Control Panel* funciona como interface para o sistema permitindo a criação de *Holons de Tarefa* através do *Lançador de Tarefas* que recolhe informação do utilizador sobre a tarefa a executar. A informação sobre os planos de produção de cada produto é colocada numa base de conhecimento usando a aplicação *Product Builder*.

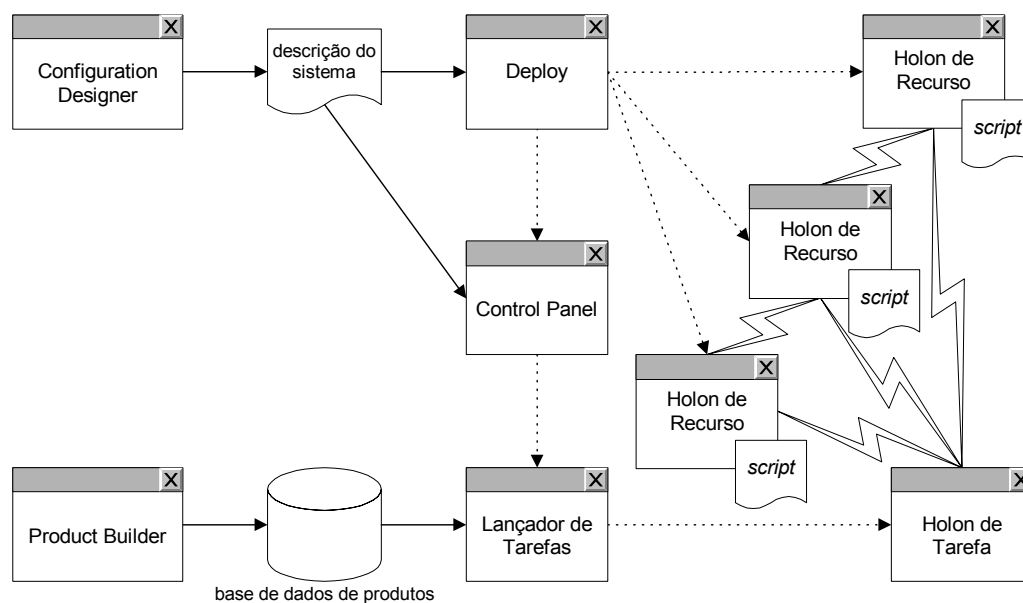


Figura 6.37 – Vista geral do protótipo *Fabricare*

A metodologia de trabalho associada à utilização do sistema é dada pela Figura 6.38.

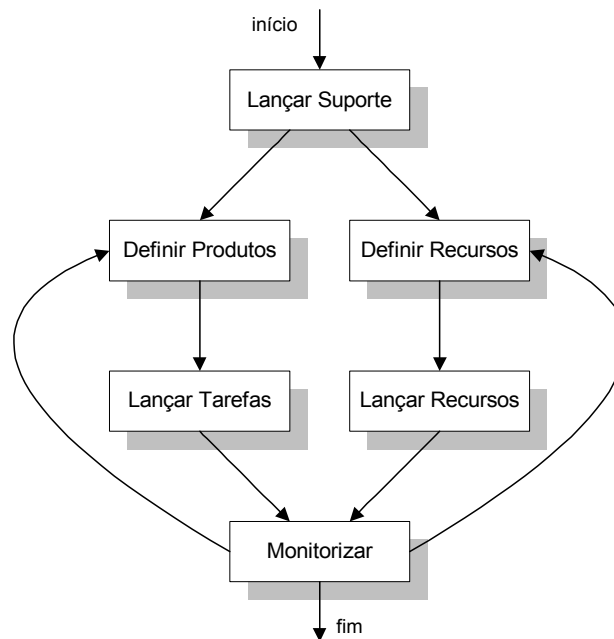


Figura 6.38 – Metodologia Fabricare

Deve começar-se por executar os mecanismos de suporte ao funcionamento do sistema (*e.g.*, o Holon de Serviços de Directório), utilizando para isso a aplicação *Deploy*. Em seguida, é possível definir os planos de produção dos produtos (através do *Product Builder*) ou a configuração dos recursos (através do *Configuration Designer*). Uma vez definida a informação sobre os produtos e recursos, é então possível colocar em execução os Holons de Recurso desejados, usando novamente a aplicação *Deploy*, e criar *Holons de Tarefa* usando o programa *Lançador de Tarefas*. Nesse momento o sistema está em funcionamento, podendo ser monitorizado (através do *Control Panel*). Eventualmente, a informação dos recursos e dos produtos pode ser alterada e o processo repetido.

6.4.2 Núcleo de Holons

Os principais componentes do sistema são obviamente os Holons de Recurso e de Tarefa, no entanto, para o seu funcionamento necessitam de alguns holons de suporte.

Se estes holons de suporte são fixos e pré-definidos, o sistema é bastante dinâmico no que toca ao número de holons de recurso e principalmente de tarefa. Os primeiros estão dependentes do ficheiro de configuração gerado pelo *Configuration Designer*, enquanto que os segundos estão dependentes da interacção com o utilizador que vai criar novas tarefas.

Holons de Suporte

O elemento de suporte às comunicações do sistema é o servidor de “quadro negro” (Figura 6.39a) que potencia a interacção entre os diversos holons existentes no sistema. Para tal, é

utilizada a tecnologia de coordenação Linda [Carriero e Gelernter, 1989a] [Carriero e Gelernter, 1989b], descrita em termos de primitivas do SICStus PROLOG, que providencia um espaço de memória partilhado (quadro negro), onde os holons escrevem e lêem as suas mensagens através dos predicados utilitários *enviar_msg* e *receber_msg*. Internamente, estes predicados utilizam uma estrutura na forma *fabricare_msg*(*Emissor*, *Destinatário*, *Conteúdo*) para representar cada mensagem lançada no sistema, em que o termo *Conteúdo* denota a mensagem propriamente dita; e.g., *lista_recursos([id_recurso_1, id_recurso_2], tw(10, 20))*. Embora as comunicações sejam efectuados via uma estrutura de “quadro negro”, de um ponto de vista conceptual cada holon emissor está a colocar uma mensagem na caixa de correio (fila de mensagens) do holon destinatário e cada holon receptor lê as mensagens que estão actualmente na sua caixa de correio.



Figura 6.39 – Interface do protótipo Fabricare: holons de suporte

Além do servidor de comunicações, há que considerar mais três holons auxiliares:

- o *Holon de Serviço de Directório* (Figura 6.39b) que implementa o serviço de directório do sistema descrito nas secções 5.3.1, 5.4.2.2 e 6.3.1;
- o *Holon de Planeamento de Processos* (Figura 6.39c) para fornecimento dos planos de produção dos produtos aos Holons de Tarefa, respondendo às mensagens *obter_plano*, conforme o que é descrito na secção 6.2.1.1. Este holon implementa também as funcionalidades de um Holon de Produto (descritas na secção 6.3.3); e
- o *Holon de Escalonamento* (Figura 6.39d) para o tratamento de conflitos entre tarefas a negociar intervalos de tempo sobrepostos para um mesmo conjunto de recursos,

conforme o que foi descrito na secção 6.2.2) e utiliza, também, os algoritmos descritos na secção 6.3.4.

Todos estes holons são programas de consola, ou seja, sem interface gráfica com o utilizador, já que a sua interacção é feita com outros holons no sistema e não com o utilizador (Figura 6.39).

Holons de Tarefa

O programa de *Holon de Tarefa* (Figura 6.40) implementa as funcionalidades de um Holon de Tarefa de acordo com a descrição apresentada nas secções 5.3.3, 5.4.2.4 e 6.3.5.

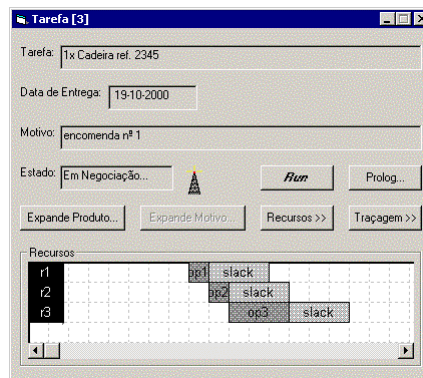


Figura 6.40 – Interface do protótipo Fabricare: holons de tarefa

Para cada tarefa criada o programa é executado, sendo a informação respeitante a cada tarefa passada ao holon que a representa, através da linha de comando do sistema operativo (correspondente à mensagem de anúncio do protocolo RCPR). No entanto, este programa é aberto no sentido em que permite a utilização de um *script* escrito em linguagem de programação em lógica PROLOG, para a definição do mecanismo de selecção de propostas a utilizar, podendo assim ser definidos novos critérios e novos mecanismos.

O seguinte programa em lógica é um exemplo de *script* para um Holon de Tarefa que define a ordem pela qual os critérios de selecção devem ser utilizados, definindo além disso, um novo critério (a nível de exemplo, pois não faz sentido usar um critério que maximize o custo de produção):

```

escolhe_criterio(maior_custo).
escolhe_criterio(primeira).
escolhe_criterio(maior_folga).
escolhe_criterio(menor_custo).

escolhe_k(maior_custo, ListaPreProc, ListaSel) :-
    maior_custo(ListaPreProc, alt(Path, Custo, LProp)),
    atribui(LProp, ListaSel).

maior_custo([S], S).
```

```

maior_custo([alt(Path, Custo, LProp)|T], alt(Path2, Custo2, LProp2)) :-
    maior_custo(T, alt(Path2, Custo2, LProp2)),
    Custo2 >= Custo.
maior_custo([S|T], S).

```

O gráfico de Gantt que é apresentado na Figura 6.40 corresponde ao escalonamento das operações da tarefa, representando cada linha um dos recursos contratados e os blocos escuros os intervalos escalonados para cada operação. Os blocos mais claros denotam as folgas que cada operação possuía no que respeita ao recurso no momento do escalonamento. É possível obter informação sobre cada uma das operações, nomeadamente o seu início, a sua duração e qual o recurso contratado. É ainda possível obter informação sobre o produto em execução por esta tarefa, nomeadamente a visualização gráfica do seu plano de produção.

Holons de Recurso

O programa de *Holon de Recurso* (Figura 6.41) implementa as funcionalidades de um Holon de Recurso de acordo com a descrição efectuada nas secções 5.3.4, 5.4.2.5 e 6.3.6.

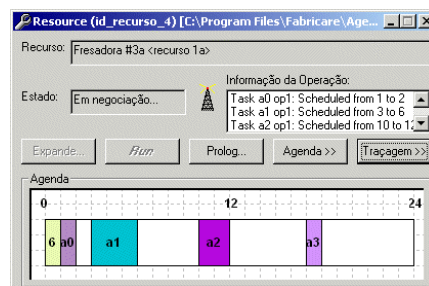


Figura 6.41 – Interface do protótipo Fabricare: holons de recurso

O gráfico de Gantt constante da Figura 6.41 corresponde à agenda de actividades do Holon de Recurso, sendo possível obter informação sobre cada uma das suas actividades (instante de início, duração, operação requisitada e tarefa contratante).

Cada Holon de Recurso é composto por um núcleo contendo as funcionalidades comuns a todos os holons e um *script* que contém as especificidades de cada Holon de Recurso individual.

O programa em lógica dado a seguir corporiza o *script* para um Holon de Recurso, informação esta que consta da base de conhecimento do holon:

```

tipo('holon-recurso').
recurso(1, 'Fresadora #3 <recurso 1>', data(2000, 11, 07), []).

habilidade(op1, 'Furar ângulo', 1, 100, 'prog-op1-r1.cnc', []).
habilidade(op24, 'Furar eixo Z', 1, 250, 'prog-op24-r1.cnc', []).

actividade(a0, op1, 1, 1, 1, data(2000, 11, 07), algum_estado, [], []).

```


Tanto o programa de Holon de Recurso como o de Holon de Tarefa possuem funcionalidades que permitem ao utilizador aceder a uma imagem da execução do programa, apresentando os eventos e os dados mais relevantes. Uma outra característica comum aos dois programas está na possibilidade de se aceder directamente ao interpretador PROLOG, utilizando a base de conhecimento do holon em questão.

6.4.3 Ferramentas de Exploração

Embora os principais componentes do sistema sejam de facto os Holons de Tarefa e de Recurso, foi desenvolvido um conjunto adicional de programas que facilitam a exploração do sistema, e que são apresentados der seguida.

Deploy

O programa *Deploy* (Figura 6.42) permite desencadear os mecanismos de suporte ao funcionamento do sistema, ou seja, o “quadro negro” para serviços de comunicações, o Holon de Serviços de Directório, o Holon de Planeamento de Processos e o Holon de Escalonamento.

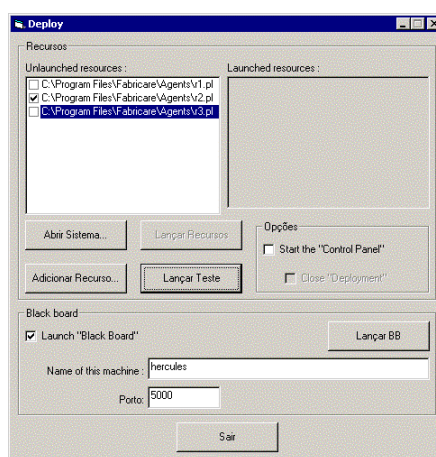


Figura 6.42 – Interface do protótipo Fabricare: aplicação “Deploy”

Adicionalmente, este programa permite também que se coloque em execução os Holons de Recurso distribuídos pelas diversas máquinas da rede informática. Para tal, um ficheiro de descrição do sistema criado pelo programa *Configuration Designer* é interpretado de forma a visualizar os recursos disponíveis. O utilizador tem então a possibilidade de escolher os recursos que pretende colocar em execução na máquina a que acede.

Lançador de Tarefas

O programa *Lançador de Tarefas* (Figura 6.43) lança os Holons de Tarefa. Este programa recolhe informação do utilizador acerca da tarefa a criar (e.g., tipo de produto, quantidade a produzir, data de entrega, cliente) e cria dinamicamente um Holon de Tarefa para a sua execução.

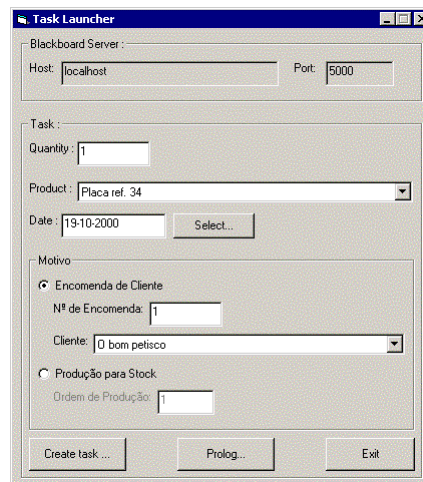


Figura 6.43 – Interface do protótipo Fabricare: aplicação “Lançador de Tarefas”

Este programa recolhe informação acerca dos produtos a fabricar e dos clientes da empresa numa base de dados do sistema.

6.4.4 Programas Adicionais

Para além do núcleo de holons e dos programas de exploração, o sistema possui também três aplicações que de forma gráfica simplificam algumas tarefas a executar por parte do utilizador.

Configuration Designer

A aplicação *Configuration Designer* (Figura 6.44) permite definir, graficamente, a topologia dos recursos na instalação fabril.

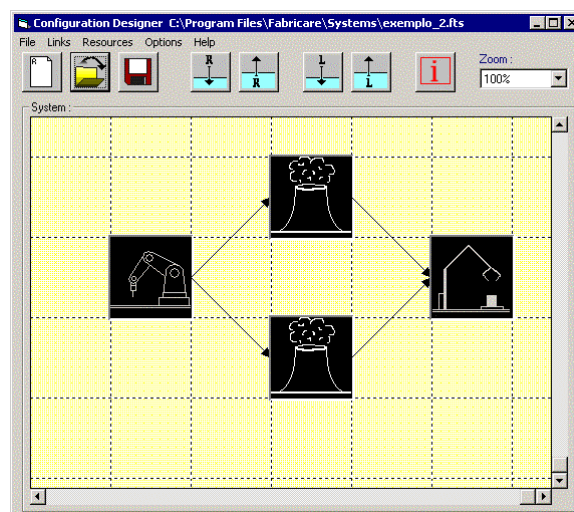


Figura 6.44 – Interface do protótipo Fabricare: aplicação “Configuration Designer”

Usando esta aplicação é possível definir, até certo ponto, a disposição dos recursos na instalação fabril e, indicar as ligações entre recursos (*e.g.*, robôs e passadeiras rolantes). Para cada recurso define-se também qual o *script* de Holon de Recurso a considerar.

Control Panel

A aplicação *Control Panel* (Figura 6.45) funciona como interface para a operação do sistema, permitindo monitorizar o funcionamento dos vários holons e oferecendo uma vista global do sistema ao utilizador. Adicionalmente, esta aplicação permite, ainda, criar tarefas através da invocação da aplicação *Lançador de Tarefas*.

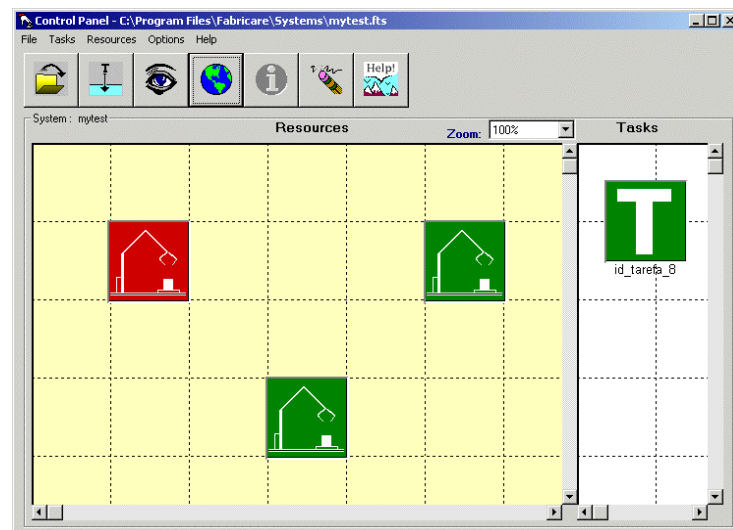


Figura 6.45 – Interface do protótipo Fabricare: aplicação “Control Panel”

No seu estado actual, permite visualizar o ficheiro de configuração, e verificar que holons de Recurso e de Tarefa se encontram em execução, questionando o Holon de Serviços de Directório, através da invocação do predicado *holons_em_execução*, definido em termos das produções:

$$\begin{aligned} \text{holons_em_execução(ListaHolons)} \leftarrow \\ \text{enviar_msg(id_srv_dir, em_execução)} \wedge \\ \text{receber_msg(id_srv_dir, r_em_execução(LHolons))} \end{aligned}$$

Uma das principais funções desta aplicação é a de fornecer uma visão global do sistema, independentemente do posto de rede onde as aplicações de Holon de Recurso se encontram a correr (Figura 6.45).

É possível observar graficamente quais os holons existentes no sistema, bem como o seu estado (activo ou desactivo), sendo também possível, questionar cada holon sobre informação respeitante à sua agenda de actividades (Figura 6.46 para os Holons de Recurso e Figura 6.47 para os Holons de Tarefa).

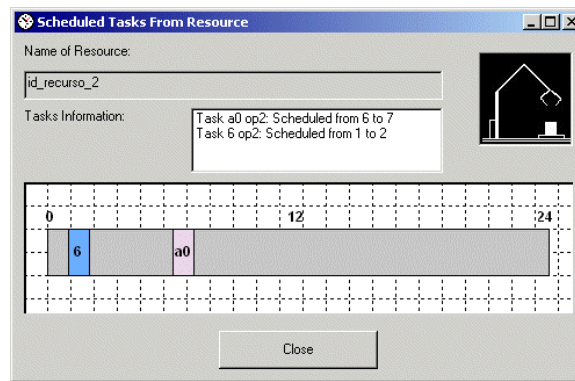


Figura 6.46 – Interface do protótipo Fabricare: escalonamento de um Holon de Recurso

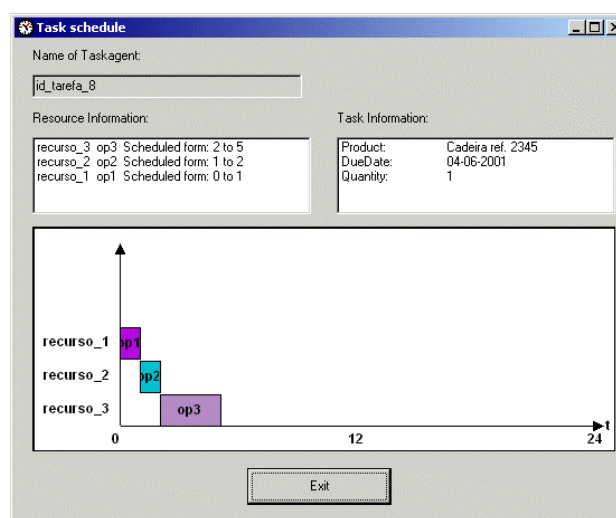


Figura 6.47 – Interface do protótipo Fabricare: escalonamento de um Holon deTarefa

Product Builder

A aplicação *Product Builder* (Figura 6.48) permite definir graficamente os planos de produção de cada produto. Esta aplicação foi criada para facilitar a criação e a edição de tais planos evitando assim o manuseamento directo das extensões dos predicados que definem um plano na base de conhecimento de um Holon de Produto.

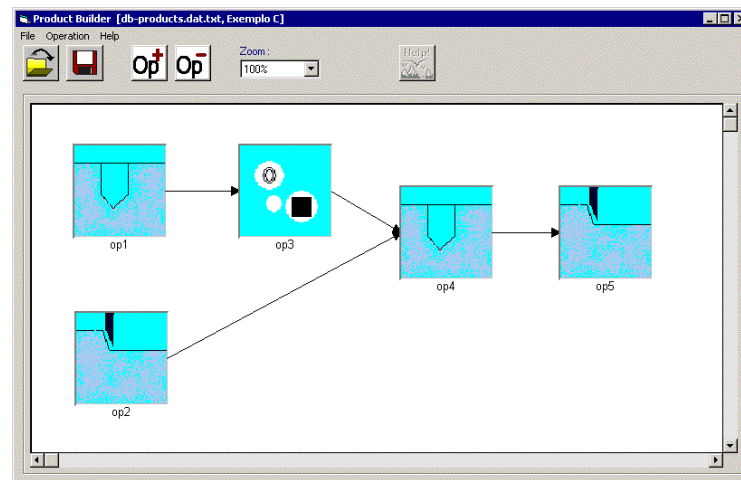


Figura 6.48 – Interface do protótipo *Fabricare*: aplicação “Product Builder”

Em condições ideais esta aplicação seria substituída pelo Holon de Planeamento de Processos que utilizaria os Holons de Produtos e de Recursos para gerar os planos.

6.4.5 Comentários ao Desenvolvimento do Protótipo *Fabricare*

Essencialmente os comentários ao desenvolvimento do protótipo prendem-se com a escolha das linguagens de programação utilizadas, Visual Basic (VB) e SICStus PROLOG (SP), e com o mecanismo de comunicação (quadro negro).

Por um lado, a utilização de PROLOG como linguagem de programação permitiu reduzir o tempo de desenvolvimento, quando comparado com outras linguagens de programação (*e.g.*, C++), devido principalmente à natureza declarativa da linguagem.

Por outro lado, a utilização de uma linguagem de programação em lógica facilitou a prototipagem, ou seja, a transição de uma especificação formal do problema (*i.e.*, dos holons) para a implementação propriamente dita. Adicionalmente, a representação de informação incompleta foi também facilitada pela utilização da programação em lógica.

Um outro factor em consideração tem a ver com a escolha da linguagem de programação em lógica PROLOG, mais concretamente, do SICStus PROLOG, dado que apresenta uma biblioteca de coordenação de eventos baseada na tecnologia Linda, que permite uma fácil implementação de quadros negros, simplificando as comunicações entre holons (no entanto, um “quadro negro” pode gerar um grave gargalo de eficiência pois introduz um elemento centralizador no sistema [Nwana *et al.*, 1996b]). As mensagens trocadas entre holons são constituídas por extensões de predicados, enquanto que, por exemplo, numa solução C++, as mensagens seriam objectos C++, implicando por isso a implementação de todo o código para a construção da mensagem, envio e

recepção pela rede, tendo especial cuidado com a reconstrução dinâmica das mensagens do lado receptor. Todas essas tarefas foram simplificadas pela biblioteca Linda.

A combinação do Visual Basic (VB) com o SICStus PROLOG (SP) permitiu a criação das interfaces gráficas com o utilizador, desenvolvidas em VB e a utilização do núcleo dos programas em PROLOG. O desenvolvimento de interfaces em VB é extremamente simples, e criou uma divisão clara entre procedimentos (manipulação de dados) e apresentação (visualização de dados).

A combinação VB/SP não se apresentou no entanto sem alguns problemas. Um dos principais foi a falta de ligação bidireccional entre o VB e o SP. Por outras palavras, é possível executar predicados PROLOG a partir do VB, mas não o contrário. A invocação de procedimentos VB a partir do PROLOG teria sido útil no que toca à actualização da interface com o utilizador. Adicionalmente, devido à não existência de um mecanismo de sinalização de existência de mensagens em espera no “quadro negro”, essa verificação teve que ser originada a partir do VB, o que implicou a existência de um ciclo temporizado para verificar a existência de mensagens, e processa-las caso existam e proceder à actualização da interface, reduzindo assim o desempenho dos holons.

Por outro lado, devido à natureza distribuída da aplicação, os testes no terreno tornaram-se difíceis de efectuar. Este facto aliado ao facto de ser impossível a invocação do *debugger* PROLOG a partir do VB tornou bastante difícil a tarefa de teste e verificação do funcionamento dos holons.

6.5 Comparação com Trabalhos Relacionados

O trabalho aqui desenvolvido foi apresentado no Capítulo 5 e em secções anteriores a esta neste capítulo, sendo agora possível efectuar uma comparação com os trabalhos de referência citados na secção 4.5.1.

É possível então, passar a uma comparação com os trabalhos de referência (Tabela 6.4, tendo por base a Tabela 4.2) segundo quatro perspectivas:

- âmbito e estrutura;
- socialização;
- informação incompleta;
- procedimento de escalonamento.

Tabela 6.4 – Comparação entre o sistema *Fabricare* e trabalhos relacionados

	Âmbito	Entidades modeladas	Socialização	Informação Incompleta	Procedimento Escalonamento
AARIA	Escalonamento e gestão da cadeia de fornecimento	Partes, Recursos e Processos Unitários	Protocolo de rede de contrato	n.a.	Algoritmo DESK efectuado pelos agentes Recurso e Parte
HMS “testbed”	Escalonamento e Controlo	Recursos, Escalonador	Holarquia; troca de mensagens entre holons	n.a.	Método centralizado efectuado pelo holon escalonador
Gou & Luh	Escalonamento	Máquinas, Células, Peças, Produtos, Coordenadores	Holarquia; troca de mensagens entre holons	n.a.	Relaxação Lagrangiana, efectuada pelos holons de Máquina/Célula e Peça/Produto
PROSA (HoMuCS)	Arquitectura de referência vocacionada para Integração empresarial (Controlo)	Produtos, Recursos, Ordens, Auxiliares (escalonador)	Holarquia; favorece a troca directa de mensagens (pedido-resposta)	n.a.	Método centralizado efectuado pelo holon escalonador
<i>Fabricare</i>	Escalonamento (integração empresarial)	Tarefas, Recursos e Produtos (fornecedores, clientes, compras e vendas)	Protocolo de rede de contrato com propagação de restrições	Representação de informação incompleta, identificação de casos e alguma manipulação	Adaptação do método de Almeida (1995) efectuado pelos Holons de Tarefa e de Recurso

Nas subsecções seguintes são elaborados alguns comentários sobre esta comparação de sistemas. Uma nota no que se refere ao trabalho PROSA, conforme foi referido, PROSA é uma arquitectura de referência e não um sistema. Por esse motivo é considerado ao longo das subsecções seguintes um sistema designado HoMuCS [Langer, 1999], que foi implementado de acordo com as orientações PROSA. Esse sistema aplica-se ao escalonamento e controlo de células de fabrico.

6.5.1 Âmbito e Estrutura (Entidades Modeladas)

No que se refere às áreas de aplicação dos vários trabalhos que são objecto de apreciação, é de realçar a escolha dos processos de escalonamento e controlo como área de teste. É também de notar a orientação para o controlo dos trabalhos oriundos de grupos com fortes raízes em Engenharia Mecânica (HMS “testbed” e HoMuCS/PROSA), ao passo que o escalonamento é o domínio preferido dos grupos com raízes nas Ciências da Computação (AARIA, Gou & Luh, *Fabricare*).

O protótipo *Fabricare* tem obviamente o seu âmbito de acção limitado ao Escalonamento, no entanto, a arquitectura proposta na secção 5.2 tem um âmbito mais abrangente, podendo ser classificada como de “integração empresarial”. Adicionalmente, embora as entidades modeladas no protótipo sejam apenas Tarefas, Recursos e Produtos, na arquitectura são também identificadas e especificadas entidades para Fornecedores, Clientes, Compras e Vendas.

Uma das principais diferenças nos trabalhos de investigação na área de produção em geral e do escalonamento em particular, são as entidades escolhidas para modelar na arquitectura [Bongaerts *et al.*, 1996]. Conforme já foi referido anteriormente, PROSA [van Brussel *et al.*, 1998] [Wyns, 1999] é uma taxonomia de Sistemas Holónicos de Produção, sendo considerada por vários investigadores na área como uma arquitectura de referência. Por esse motivo, a comparação entre as diferentes arquitecturas já referenciadas vai ser efectuada tendo em conta os quatro tipos de holons identificados na taxonomia PROSA: (i) Produtos; (ii) Recursos; (iii) Ordens de Fabrico; e (iv) Auxiliares.

O sistema AARIA encaixa parcialmente nesta taxonomia, visto modelar recursos e produtos (através dos agentes Parte). No entanto, as ordens de fabrico não são modeladas como agentes, mas sim como mensagens trocadas entre o utilizador (cliente) e o sistema. Adicionalmente, AARIA modela uma dimensão extra não existente na taxonomia PROSA, ao modelar os processos como agentes.

O sistema HMS “testbed” é um produto oriundo do mesmo grupo de investigação que desenvolveu a taxonomia PROSA, tendo inclusive estado na sua génese. O sistema HoMuCS é um produto de um grupo estreitamente relacionado com o grupo PROSA que implementou um

sistema seguindo essa mesma arquitectura. Por esse motivo ambos os trabalhos (*i.e.*, HMS “testbed” e HoMUCS) se relacionam imenso com PROSA, sendo que o sistema HoMuCS adere totalmente a esta taxonomia.

A arquitectura de Gou & Luh embora não relacionada com PROSA, encaixa perfeitamente na taxonomia dada, visto que modela produtos como holons Parte (que também representam estágios intermédios de fabrico), recursos como holons Máquina e Célula (na arquitectura PROSA não existe esta distinção hierárquica, sendo uma célula considerada como um recurso) e ordens de fabrico como holons de Produto. Adicionalmente, existem holons auxiliares, como, por exemplo o Holon Coordenador de Célula.

A arquitectura proposta neste trabalho pode também ser classificada de acordo com a taxonomia PROSA, visto que as quatro classes de holons são modeladas na arquitectura. Os Holons de Recurso e de Produto possuem o mesmo nome que as classes de PROSA, e a classe Ordem de Fabrico é representada pelos Holons de Tarefa. Adicionalmente, o Holon de Serviços de Directório é um exemplo de um holon auxiliar.

Um outro aspecto relativo à arquitectura de cada um dos sistemas referenciados prende-se com a existência de elementos centralizadores. O sistema HMS “testbed” utiliza um holon para escalonamento e como tal tem aí uma função centralizada. Gou & Luh utilizam holons coordenadores de célula que reúnem informação e participam activamente no cálculo do escalonamento. O sistema AARIA reclama a inexistência de elementos centralizadores, no entanto, na representação gráfica do sistema (Figura 4.6), existe um componente “Gestor” que não é explicado pelos autores.

O sistema *Fabricare* possui um holon que centraliza, actualmente, uma função, a gestão de conflitos. Esse holon (Holon de Escalonamento) serializa as negociações e todos os Holons de Tarefa necessitam de comunicar com ele, no entanto, a sua participação é diminuta e, apenas no início do processo de negociação. Por outro lado, a arquitectura proposta assenta numa abordagem completamente distribuída ao contrário de PROSA que favorece elementos centralizadores (só os recursos, ordens de fabrico e produtos são distribuídos por especificação da arquitectura).

6.5.2 Socialização

No que toca a mecanismos de interacção entre holons, são seguidas duas abordagens distintas, orientando-se uma mais para a negociação, e a outra para a simples troca de mensagens, com instruções de comando ou pedido de informação.

O sistema HMS “testbed” utiliza uma filosofia de pedido-resposta entre o holon de controlo e o holon de escalonamento. O holon de controlo pede ao holon de escalonamento que efectue um

plano para o escalonamento das ordens de fabrico existentes no sistema e em seguida executa esse plano. Quando não é possível efectuar as diferentes actividades (*i.e.*, operações) de acordo com o escalonamento sugerido, o holon de controlo decide o que executar requisitando ao holon de escalonamento um novo escalonamento. A execução do plano é efectuada mediante a troca de mensagens de controlo (*i.e.*, *master/slave*) entre o holon de controlo e os holons de recurso. PROSA e HoMUCS seguem uma filosofia idêntica.

Ao contrário do que foi referido em epígrafe, os sistemas AARIA, Gou & Luh e *Fabricare* privilegiam a negociação. AARIA possui uma estrutura plana, onde cada agente age em parceria, não existindo agentes hierarquicamente distintos. A interacção entre os agentes baseia-se no Protocolo de Rede de Contrato. Gou & Luh introduzem uma estrutura de decomposição (devido à natureza holónica da arquitectura) onde cada holon trabalha em parceria com os outros holons dentro da mesma holarquia mas não comunica com holons fora dessa holarquia. A comunicação consiste na troca de resultados parciais do problema de relaxação Lagrangiana.

O sistema *Fabricare* baseia-se também no Protocolo de Rede de Contrato para a negociação de operações entre os recursos e as tarefas. No entanto, devido à necessidade de cooperação (e coordenação) entre os vários recursos intervenientes na execução de um plano de produção, o PRC original é modificado por forma a incluir duas novas fases, onde os recursos trocam informação entre si, para coordenar os seus processos de escalonamentos. Todos os holons de recurso e de tarefa são vistos como pares no processo de negociação.

6.5.3 Tratamento de Informação Incompleta

Nenhum dos trabalhos de referência (PROSA/HoMUCS), HMS “testbed”, AARIA e Gou & Lu) aborda a problemática da informação incompleta. No entanto, o grupo de investigação do projecto AARIA desenvolveu um outro trabalho relativo a gestão de cadeias de fornecimento (DASch [Parunak, 1998b]) onde modela a incerteza nos canais de distribuição (nomeadamente no tempo de percurso).

Ao contrário dos trabalhos tomados para comparação (e mesmo outros trabalhos referidos no “estado da arte”), o sistema *Fabricare* permite o tratamento e representação de informação incompleta, utilizando para isso a notação descrita na secção 5.4.1. Na secção 5.4.2 foram identificados casos de informação incompleta nos sistemas de produção (mais concretamente, na base de conhecimento dos holons constituintes da arquitectura proposta), casos esses, que foram implementados nos holons do protótipo *Fabricare*, ou seja, nos Holons de Recurso, Holons de Tarefa e Holons de Produto.

Neste trabalho, contudo, essa informação não é utilizada na sua plenitude, nos processos de operação dos holons, sendo apenas utilizada pelo Holon de Serviços de Directório relativamente ao conhecimento sobre as funcionalidades de outros holons.

6.5.4 Procedimento de Escalonamento

O método de escalonamento utilizado nos sistemas referenciados, é bastante distinto, quer no tipo de escalonamento, quer na distribuição dos procedimentos pelos diversos elementos constituintes da arquitectura.

O sistema HMS “testbed” utiliza um holon de escalonamento centralizado em colaboração com um holon de controlo. Também PROSA e HoMUCS favorecem a utilização de métodos centralizados para o escalonamento, normalmente através de holons auxiliares, que efectuem um escalonamento usando uma qualquer técnica bem conhecida e fornecem esse escalonamento a cada holon de ordem de fabrico.

No sistema AARIA utiliza-se um método bastante flexível que não impõe um momento de arranque fixo para as tarefas, baseando-se antes no acordo feito com o cliente, para decidir a ordem de execução das várias tarefas da forma mais eficiente (do ponto de vista do recurso). O acordo com o cliente passa pela escolha de um ponto numa função de custo/prazo-de-entrega, por utilização do procedimento DESK [Sauter e Parunak, 1999], baseada em janelas temporais e na diminuição de custos para prazos de entrega de produtos mais alargados. Cada agente de recurso efectua o escalonamento local das operações requisitadas.

Gou & Luh utilizam um processo fundamentalmente matemático, tentando minimizar um conjunto de funções através de programação dinâmica com relaxação Lagrangiana. Em primeiro lugar cada célula calcula o seu escalonamento (propagando resultados entre os holons de máquina existentes na célula) e em seguida o escalonamento da fábrica é calculado usando os resultados das células.

Conforme o que já foi descrito anteriormente ao longo deste capítulo, o procedimento de escalonamento utilizado no sistema *Fabricare* é uma adaptação de um método centralizado. O escalonamento baseia-se no Protocolo de Rede de Contrato, sendo efectuado pelos vários recursos intervenientes na obtenção de um produto (requisitado por uma tarefa) através da propagação de restrições para determinar as janelas temporais efectivas de cada recurso.

6.6 Resumo do Capítulo

Devido à natureza distribuída da arquitectura apresentada na secção 5.2 torna-se necessário regulamentar a interacção entre os vários holons, especialmente os Holons de Tarefa e de Recursos para a actividade de escalonamento (escolhida como caso de teste para o modelo conceptual e arquitectura). Por esse motivo, na secção 6.2 foi apresentado o protocolo de negociação que rege o processo de contratação de serviços entre Tarefas e Recursos. Esse protocolo está preparado para o tratamento de excepções e, principalmente para evitar o *Problema de Indecisão*. Em seguida procedeu-se a uma análise da complexidade do protocolo no que toca ao número de soluções encontradas, ao número de mensagens trocadas e ao tamanho total das mensagens trocadas.

Tendo sido especificado o protocolo, foi então descrito o funcionamento de cada holon. Os principais algoritmos para a operação dos holons directamente relacionados com o escalonamento (*Holon de Tarefa*, *Holon de Recurso*) e de outros holons necessários (*Holon de Serviço de Directórios*, *Holon de Produto* e *Holon de Escalonamento*) foram então apresentados e explicados, estabelecendo-se a ponte entre as actividades descritas no protocolo para cada um dos holons e os seus procedimentos internos.

De seguida apresentou-se o protótipo de sistema que foi implementado, seguindo as descrições efectuadas quer no Capítulo 5 quer no Capítulo 6. Este protótipo denominado *Fabricare* é composto por um conjunto de aplicações que incluem os Holons de Tarefa e de Recurso, bem como os holons auxiliares (Serviço de Directório, Planeamento de Processos, Escalonamento) e, algumas ferramentas para a exploração do sistema, permitindo colocar em execução os Holons de Recurso e a criação de tarefas. Nesta secção foram também feitos alguns comentários às diferenças entre o método original de escalonamento e o método utilizado neste trabalho, bem como alguns comentários ao desenvolvimento do protótipo.

Finalmente, foi apresentado uma comparação entre o trabalho desenvolvido e os trabalhos de referência mencionados no “Estado da Arte”. Essa comparação foi efectuada segundo quatro linhas de interesse, de acordo como os objectivos enunciados, em termos de estrutura, socialização das entidades, tratamento de informação incompleta e procedimento de escalonamento ■

CAPÍTULO 7 CONCLUSÕES



Não tenho nenhuma solução, mas é certo que admiro o problema

Ashleigh Brilliant (1933 –) escritor, cartunista, professor de história e “filósofo de rua” Inglês a viver nos EUA.

Tentou-se com este trabalho contribuir para o aumento do conhecimento científico na área dos Sistemas Inteligentes para Produção, mais concretamente nos Sistemas Holónicos de Produção. Conforme foi referido no início da dissertação, não se pretende apresentar soluções definitivas, mas sim algumas soluções pontuais, sendo neste caso o sistema *Fabricare* a face mais exposta das contribuições deste trabalho.

Para terminar a apresentação de qualquer trabalho é conveniente salientar as conclusões alcançadas, bem como criticar as suas limitações e apontar perspectivas futuras. Assim, começa-se por fazer uma síntese do que foi dito nos capítulos de enquadramento referente à produção e aos seus problemas, bem como à sociedade actual e aos requisitos que coloca nos sistemas de produção. Em seguida detalha-se as contribuições e respectivas conclusões do trabalho que permitem elaborar a *Tese* da dissertação e responder às questões colocadas na introdução. Finalmente, serão apresentadas as limitações do trabalho desenvolvido e as perspectivas de desenvolvimento futuro apontando as direcções em que se pensa seguir.

7.1 Síntese da Dissertação

No Capítulo 2 foi apresentado o tema da Produção e foram descritos os conceitos e sistemas normalmente utilizados, como por exemplo, a Produção Integrada por Computador (CIM). No Capítulo 3 foi apresentado o contexto socioeconómico dos nossos dias, evidenciando os desafios impostos à produção e terminando com a elaboração de uma lista de propriedades desejáveis para uma nova geração de sistemas de produção (secção 3.4.3). Assim, na secção 3.1 caracterizou-se o actual contexto socioeconómico em termos gerais (mercados dinâmicos, competição a nível global, estruturas empresariais em rede e predominância de alianças e colaborações com outras empresas) e, em termos de indústria (mudança da produção em massa para o fabrico flexível, reconhecimento da inovação e importância da investigação na criação de vantagem competitiva e fonte de crescimento da empresa).

A par destas características é possível observar certas tendências em termos gerais de mercados e também na área de produção [Hunt, 1989] [Kusiak, 1990] [Solberg e Kashyap, 1993] [Kalpakjian, 1995] [ISC, 1994]:

- *Produtos* – redução do ciclo de vida e da dimensão das encomendas; aumento da variedade e da complexidade com maiores requisitos de qualidade;
- *Processos* – incremento de opções em termos de materiais e processos; avanços tecnológicos mais rápidos; aumento da diversidade tecnológica; aumento do número de instalações produtivas a nível mundial; redução do tempo de colocação no mercado; necessidade de preservação de recursos naturais; restrições ambientais;
- *Práticas de gestão e factores humanos* – novas estruturas organizacionais e filosofias de trabalho (*e.g.*, teletrabalho); incremento de parcerias;
- *Mercados* – crescente globalização de mercados; as condições de mercado flutuam de forma significativa; mercados complexos e altamente competitivos; alargamento e globalização das cadeias de fornecimento;
- *Clientes* – menor fidelidade dos clientes; pretendem participação mais significativa na concepção do produto; pretendem produtos de elevada qualidade e custo reduzido; pretendem personalização de produtos para necessidades específicas.

Estas mudanças na sociedade e na economia impõem mudanças nos sistemas de produção pois, embora a Produção Integrada por Computador seja promovida universalmente, foram detectados alguns problemas no seu processo de implementação (*e.g.*, custo elevado) sendo opinião corrente que o CIM não é a resposta para os sistemas de produção do futuro.

Entre outros, foram identificados os seguintes problemas na Produção Integrada por Computador [Höpf, 1994] [Ueda, 1994] [Bongaerts *et al.*, 1995] [Parunak, 1996] [Tharumarajah *et al.*, 1996] [Gou e Luh, 1997] [Bussmann, 1998] [Kádár *et al.*, 1998] [Shen e Norrie, 1999]:

- *Inflexibilidade* – a arquitectura CIM é fixa independentemente dos requisitos de produção, sendo difícil expandir ou reconfigurar um processo para novos produtos;
- *Falta de robustez* – a eficiência não é garantida fora da gama operatória pré-definida; além disso, a existência de um controlador central cria a existência de um único ponto de falha;
- *Falta de adaptabilidade* – no que se refere a avarias ou mau funcionamento dos recursos;
- *Dificuldade de manutenção* – devido à falta frequente de dados por parte das máquinas.

Um dos principais problemas das implementações CIM é a sua estrutura hierárquica com controlo centralizado que não se adequa à produção de lotes quase unitários em mercados altamente dinâmicos em constante mudança [Kádár *et al.*, 1998].

Além dos problemas dos sistemas de produção actuais em responder às tendências observadas, os sistemas de produção também devem estar preparados para o que virá a ser a produção. Pensa-se que no futuro a produção será caracterizada por [NGM, 1997] [CVM, 1999]:

- recursos de produção distribuídos à escala planetária;
- requisitos de procura de pouca quantidade e muita variedade de produtos;
- fornecimento de soluções individuais de acordo com as necessidades específicas de cada cliente;
- integração e execução concorrente de todas as actividades da empresa;
- definição de competências nucleares e agregação temporária com outras empresas possuindo outras competências de forma a criar um todo harmonioso.

Dadas as tendências observadas nos mercados, os problemas com as actuais arquitecturas dos sistemas de produção e as previsões do que será a produção no futuro, conclui-se que a nova geração de sistemas de produção deve caracterizar-se pelas seguintes propriedades [Sousa *et al.*, 2000c] [Sousa *et al.*, 2000b]:

- *Distribuição* – o sistema deixa de ser monolítico e passa a ser constituído por várias entidades;
- *Descentralização* – as funcionalidades estão repartidas por várias entidades do sistema;

- *Autonomia* – cada entidade do sistema possui capacidade de decisão e controlo sobre a execução das suas acções;
- *Dinamismo* – o estado do sistema não é estático, mudando constantemente no que refere à estrutura, comportamentos e relações entre entidades;
- *Reactividade* – a selecção de acções é feita de acordo com as suas percepções;
- *Flexibilidade* – capacidade dos recursos efectuarem rapidamente uma mudança de processos (definidos *a priori*) com base no produto a fabricar;
- *Adaptabilidade* – capacidade que o sistema exhibe de continuar em funcionamento perante mudanças e perturbações de forma a responder às necessidades de produção e restrições momentâneas da instalação fabril;
- *Agilidade* – evolução contínua e aproveitamento de oportunidades de negócio esporádicas e instantâneas, através de alianças estratégicas;
- *Informação Incompleta* – por forma a melhor se aproximar da realidade.

Para tentar responder a estes requisitos, no Capítulo 4 foi apresentada a Teoria dos Sistemas Holónicos e mais concretamente, os Sistemas Holónicos de Produção. Este tipo de sistemas foi proposto como solução para os problemas encontrados nos sistemas de produção actuais, possuindo as características enunciadas para o desenvolvimento de novos sistemas de produção. Neste capítulo foram também apresentados os conceitos de Sistema Multiagente e de Programação em Lógica Estendida como ferramentas de implementação de tais sistemas.

Um *Sistema Holónico de Produção* (HMS) é uma holarquia que abarca a totalidade das actividades de produção desde o projecto até à venda, passando pelo fabrico, *marketing* e recepção de encomendas, para alcançar a empresa ágil de produção [Valckenaers *et al.*, 1997] [Bongaerts, 1998]. As características pertinentes de um HMS são [HMSC, URL]:

- elementos autónomos, cooperativos, reutilizáveis e autoconfiguráveis (*i.e.*, os holons);
- estrutura recursiva dos elementos (a dualidade natural dos holons);
- reconhecimento do papel importante das pessoas para o sucesso global;
- inexistência de controlo central na instalação fabril;
- integração de trabalho humano nas células de fabrico.

Do estudo dos Sistemas Holónicos em geral e dos Sistemas Holónicos de Produção em particular, conclui-se que os referidos sistemas, isto é, os HMS, possuem características intrínsecas (*e.g.*, autonomia, cooperação) que lhes permitem exibir as propriedades identificadas como requisitos da nova geração de sistemas de produção, nomeadamente:

- *Distribuição* – os holons são por definição distribuídos;
- *Descentralização* – através da autonomia e descentralização de funções no “desenho” do sistema;
- *Autonomia* – os holons são por definição autónomos;
- *Dinamismo* – através das holarquias dinâmicas;
- *Reactividade* – implementada na concepção do sistema;
- *Flexibilidade* – através da autonomia dos holons e holarquias dinâmicas;
- *Adaptabilidade* – através da autonomia dos holons e holarquias dinâmicas;
- *Agilidade* – através das holarquias dinâmicas e cooperação entre holons.

Após a descrição do conceito holónico importa abordar a problemática da sua implementação, tendo sido por isso apresentados na secção 4.4 dois mecanismos para a concretização de sistemas holónicos: *sistemas multiagente* e *programação em lógica estendida*.

É fácil estabelecer um paralelo entre holons e agentes e entre sistemas holónicos e sistemas multiagente já que há uma certa sobreposição nas características de ambos os conceitos, sendo assim, como um sistema multiagente exibe as mesmas características identificadas num sistema holónico (à excepção da estrutura holárquica) pode então ser utilizado para implementação de Sistemas Holónicos de Produção. É assim possível afirmar que:

- um holon pode ser implementado recorrendo à tecnologia dos agentes;
- até certo ponto um holon é um agente com capacidade de agregação temporária e intermediário com Humanos;
- o conceito holónico oferece a visão e a estrutura organizacional ao passo que o conceito de agência oferece a tecnologia que possibilita a implementação de tais sistemas.

Para a construção de sistemas inteligentes há que considerar a problemática da representação e inferência de conhecimento. As linguagens lógicas podem ser usadas com sucesso nesse sentido [Wooldridge, 1992] [Russell e Norvig, 1995], além disso, a Lógica pode ser usada com sucesso como uma linguagem formal para a especificação de agentes, assim como a Programação em Lógica pode ser usada para a sua implementação. No entanto, o pressuposto do mundo fechado e o pressuposto do domínio fechado, utilizados na Programação em Lógica clássica, embora

simplifiquem o desenvolvimento de sistemas inteligentes, limitam o seu campo de aplicação, pois na maioria dos casos o conhecimento total do domínio é impossível de ser obtido. Sendo que é normal a ocorrência de cenários de *Informação Incompleta* nas bases de conhecimento e nos processos de negociação [Neves, 1984] [Traylor e Gelfond, 1993] [Neves *et al.*, 1997b], os sistemas reais podem beneficiar largamente de abordagens que evitem estas limitações.

Sugeriu-se então a Programação em Lógica Estendida (programação em lógica com representação de informação negativa explícita) para a implementação dos agentes/holons, concluindo-se que:

- ao adicionar a capacidade para representação e raciocínio sobre informação incompleta a um sistema, a sua base de conhecimento passa a poder descrever o mundo real de forma muito mais correcta;
- a utilização da programação em lógica estendida com o meta-interpretador apresentado (Figura 4.5), permite a representação e manipulação de nulos, nomeadamente *nulos desconhecidos*, *nulos mutuamente exclusivos* e *nulos não permitidos* na base de conhecimento de um agente.

Reunidas estas valências (sistemas holónicos, sistemas multiagente, programação em lógica estendida) partiu-se então para a concepção de um sistema, denominado *Fabricare*, que exhibisse as características enunciadas e (pelo menos teoricamente) fosse capaz de responder aos desafios impostos ao sector produtivo numa sociedade em constante mudança.

Assim, no Capítulo 5 definiu-se uma arquitectura holónica para empresas de produção, constituída por holons que modelam as principais entidades de negócio (*i.e.*, clientes, vendas, fornecedores, compras, produtos, recursos e tarefas [Sousa *et al.*, 2000a]), possuindo as várias funcionalidades da actividade de produção (*e.g.*, escalonamento). Ainda neste capítulo, esses holons foram especificados em termos de base de conhecimento (incluindo a representação dos cenários identificados de informação incompleta).

Após a especificação do sistema, definiu-se o seu funcionamento no Capítulo 6, onde também se definiu um protocolo de negociação entre os holons de Tarefa e de Recurso com vista ao escalonamento das ordens de fabrico. Este protocolo baseia-se no protocolo de rede de contrato [Smith, 1980] [Davis e Smith, 1983] que é estendido para garantir as dependências temporais entre as operações da tarefa. Por fim, desenvolveu-se um protótipo para a actividade de escalonamento de acordo com a arquitectura, protocolo e operação definidos anteriormente.

7.2 Contribuições, Conclusões e Originalidades

Os objectivos definidos para o trabalho na secção 1.5 foram atingidos quase na totalidade, tendo a excepção sido o objectivo 5 (representação e manipulação de informação incompleta) que foi apenas parcialmente atingido, já que a utilização de informação incompleta no processo de decisão/operação dos holons é reduzida – o único holon que de facto utiliza o axioma *demo* é o Holon de Serviços de Directório ao responder a perguntas do tipo *faz(Holon, Habilidade)*.

Do resultado deste trabalho surgiram várias contribuições (relacionadas com os objectivos enunciados) que são apresentadas em seguida.

Contribuição – *Definição de uma arquitectura para sistemas holónicos de produção.*

Esta contribuição responde aos objectivos 3 e 4 (especificação de uma arquitectura e especificação de cada entidade constituinte da arquitectura, respectivamente) ao especificar uma arquitectura holónica que permite responder aos requisitos impostos (Figura 7.1).

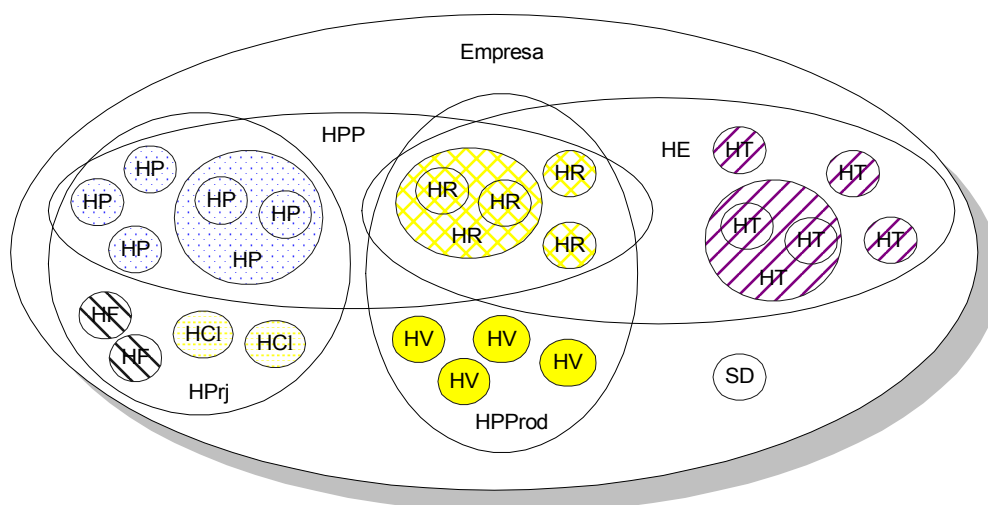


Figura 7.1 – Arquitectura holónica proposta (actividades de fabrico)

A arquitectura foi proposta na secção 5.2 sendo baseada no paradigma dos Sistemas Holónicos e abarca as áreas funcionais do fabrico (Projecto, Planeamento de Processos, Planeamento de Produção e Escalonamento), compras e vendas [Sousa e Ramos, 1997] [Sousa e Ramos, 1998] [Sousa *et al.*, 1999b]. Na secção 5.3 foi especificada a base de conhecimento e na secção 6.3 o funcionamento de cada holon da arquitectura. Sendo o escalonamento o caso de

teste escolhido, foi dada especial ênfase aos holons que possibilitam essa actividade, ou seja, os Holons de Tarefa e os Holons de Recurso.

Da arquitectura proposta conclui-se que:

- as principais entidades identificadas são: produtos, recursos, tarefas, compras, fornecedores, vendas e clientes. Enquanto outros trabalhos baseiam-se apenas na utilização da abstracção ‘Recurso’, esta proposta identifica outras entidades, tendo como principal vantagem a especialização de cada agente de acordo com um domínio (*e.g.*, tarefas, produto);
- a arquitectura proposta permite efectuar as seguintes actividades de produção: projecto, planeamento de processo, planeamento de produção, escalonamento, gestão de compras e gestão de vendas;
- as entidades principais são representadas como holons;
- as actividades de produção são representadas como holarquias;
- assenta numa abordagem distribuída e descentralizada procurando evitar elementos centralizadores de funções e favorecendo a coordenação/cooperação de vários holons para a emergência de funcionalidades;
- devido à natureza recorrente da estrutura holónica, a arquitectura proposta pode ser empregue a diversos níveis de detalhe (*e.g.*, instalação fabril vs. instalações fabris), assemelhando-se assim a um fractal e mostrando auto-semelhança;
- os holons suportam herança de conhecimento e de comportamentos (através dos axiomas *demo_α* e *tipo*);
- as holarquias criadas dinamicamente em resposta a eventos no sistema correspondem às características de dinamismo, adaptabilidade e agilidade enunciadas como requisitos para os sistemas de produção de nova geração;
- as holarquias são apetrechadas de mecanismos que facilitam as comunicações entre os seus membros (memória partilhada, “quadro negro”) e permitem obter uma visão global do subproblema em questão;
- é necessário um mecanismo que permita a identificação dos vários holons existentes, papel desempenhado pelo Holon de Serviços de directório, que além disso permite que o sistema seja flexível no que toca aos elementos existentes, ou seja, nenhum dos holons sabe à partida com que holons comunicar, mas obtém essa informação dinamicamente.

As originalidades desta arquitectura centram-se nas entidades modeladas, na abordagem distribuída e descentralizada e nas valências da arquitectura para integração empresarial e integração inter-empresarial.

Contribuição – *Notação para a representação (e demonstração de teoremas) com informação incompleta.*

A secção 5.4.1 apresenta uma notação para representação de informação incompleta tendo como base os axiomas da base de conhecimento de cada holon [Sousa *et al.*, 2000b], de acordo com o objectivo 5. Com base em [Traylor e Gelfond, 1993] [Analide e Neves, 1996] [Neves *et al.*, 1997b] e utilizando a programação em lógica estendida (com representação explícita de informação negativa), essa notação permite a representação de nulos do tipo desconhecido, nulos desconhecidos de um conjunto finito de valores e nulos não permitidos.

Na área da produção existem várias situações onde toda a informação necessária pelo sistema não está disponível (informação incompleta). Embora esta informação não esteja completamente definida, tornando assim impossível a sua utilização, o facto de ela existir no sistema é mais útil do que a sua não existência ao permitir uma melhor representação do mundo real. Da componente de informação incompleta deste trabalho pode concluir-se que:

- a notação utilizada permite a representação dos seguintes tipos de informação incompleta: negação explícita; nulos desconhecidos; nulos mutuamente exclusivos; e nulos não permitidos;
- o demonstrador utilizado permite inferir o valor de verdade de uma conjunção de termos lógicos com base numa teoria que contenha nulos destes quatro tipos;
- é possível a representação de nulos desconhecidos e nulos mutuamente exclusivos para um mesmo tipo de axioma (e.g., para uma produção do tipo *encomenda(Cliente, ListaProdutos)* é possível representar situações em que não se sabe quem encomendou e situações em que não se sabe quem entre dois clientes encomendou determinado produto).

O facto de se abordar a problemática da informação incompleta nos sistemas de produção é original. A notação utilizada, embora não seja totalmente original, apresenta algumas alterações em relação a [Analide e Neves, 1996], nomeadamente na representação de nulos não permitidos e no meta-interpretador.

Contribuição – *Identificação de situações de informação incompleta em sistemas de produção.*

Continuando na prossecução do objectivo 5, na secção 5.4.2 apresentou-se o resultado do levantamento de situações onde pode existir falta de informação nas empresas de produção. Esses casos levam em linha de conta a base de conhecimento de cada holon identificado anteriormente na arquitectura proposta [Sousa *et al.*, 2000b].

Dos casos de informação incompleta identificados neste trabalho pode concluir-se que:

- foram identificados 46 casos de informação incompleta nos sistemas de produção em geral e na base de conhecimento de cada holon da arquitectura proposta em particular;
- os casos identificados foram representados nas bases de conhecimento de cada holon objecto de implementação;
- a utilização de informação incompleta nos processo de funcionamento de cada holon é ainda limitada, ficando-se pelo Holon de Serviços de Directório.

Esta contribuição é original pois não se conhecem outros trabalhos que tenham efectuado um levantamento de situações de informação incompleta em sistemas de produção.

Contribuição – *Extensão ao protocolo de rede de contrato com cooperação entre fornecedores de serviços.*

A secção 5.4 apresenta o Protocolo de Rede de Contrato com Propagação de Restrições (PRCPR) [Sousa e Ramos, 1998] [Sousa *et al.*, 1999b] [Sousa e Ramos, 1999a] utilizado para regulamentar a interacção entre os Holons de Tarefa e os Holons de Recurso (objectivo 6).

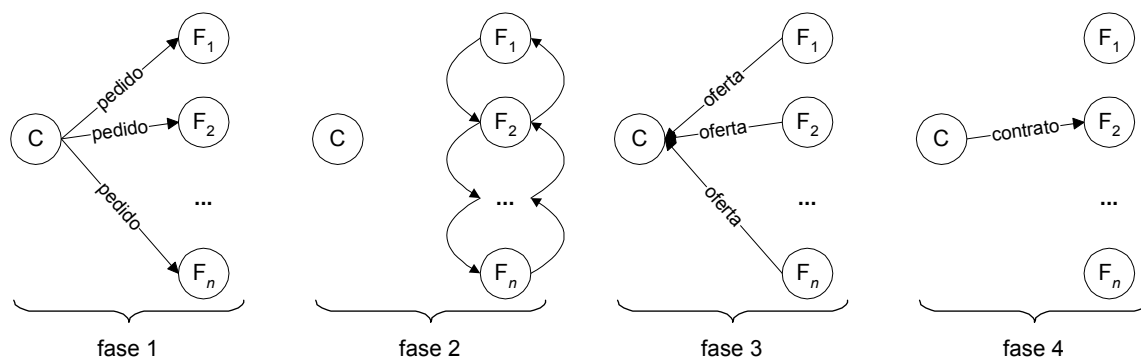


Figura 7.2 – Protocolo RCPR

A Figura 7.2 dá uma visão geral do Protocolo de Rede de Contrato com Propagação de Restrições utilizado no sistema. Este protocolo estende o Protocolo de Rede de Contrato [Smith, 1980] [Davis e Smith, 1983] ao incluir troca de informação entre os diferentes fornecedores de serviços (fase 2) para coordenar as dependências temporais existentes entre as várias operações requisitadas pertencentes a uma mesma tarefa.

O Protocolo de Rede de Contrato com Propagação de Restrições que pode ser caracterizado por:

- utilização de duas abstrações Recurso e Tarefa que permitem antever outras utilizações do protocolo que não no escalonamento de tarefas industriais;
- basear-se no protocolo de rede de contrato, mas incluindo cooperação explícita entre fornecedores de serviços (*i.e.*, os recursos) motivado pela necessidade de coordenação das dependências temporais entre as várias operações a executar numa tarefa;
- ser dinâmico no que toca aos intervenientes através da obtenção de informação sobre os holons capazes de executar cada operação necessária bem como do plano de produção;
- estar preparado para evitar conflitos entre tarefas, negociando intervalos de tempo sobrepostos num mesmo recurso (Problema de Indecisão) recorrendo à intervenção da holarquia de escalonamento na serialização das negociações;
- a existência de recursos alternativos para uma mesma operação pode originar uma explosão combinatória do espaço de soluções;
- adicionalmente, foi apresentado um método para o cálculo do número de soluções possíveis considerando um determinado plano e o número de recursos para cada operação desse plano;
- o número de mensagens trocadas cresce exponencialmente com o número de operações do plano.

Devido à explosão combinatória de soluções possíveis quando existem recursos alternativos para uma mesma operação, o protocolo RCPR original foi modificado para reduzir o número de mensagens transmitidas, concluindo-se que:

- a nova variante do protocolo tem uma evolução linear do número de mensagens transmitidas dependente do número de operações do plano;
- o ganho relativo em termos de número de mensagens trocadas desta variante do PRCPR em relação ao PRCPR original vai aumentando na proporção directa do número de operações;

- o tamanho de cada mensagem trocada (mais especificamente das mensagens de influência directa, influência inversa e proposta) é maior nesta variante do que no PRCPR original, devido à concatenação de várias soluções numa só mensagem;
- embora sejam maiores, o comprimento total das mensagens trocadas é menor na variante do PRCPR do que no PRCPR original; no entanto, o ganho relativo em termos de comprimento total de mensagens vai diminuindo com o aumento do número de operações.

Este protocolo apresenta um aspecto original ao incluir explicitamente as fases de coordenação entre holons fornecedores de recursos. Adicionalmente, contribui-se com o estudo de complexidade do problema e do protocolo.

Contribuição – *Concepção de um algoritmo distribuído para escalonamento.*

Como resultado colateral do PRCPR obteve-se um algoritmo distribuído para escalonamento dinâmico de tarefas industriais (*vide* secção 6.3.7) [Sousa *et al.*, 1999b], baseado num método centralizado, originalmente proposto por Almeida (1995) e Ramos *et al.* (1995). Este algoritmo pode ser caracterizado por:

- utilização de uma *abordagem distribuída* com vários processos computacionais (*i.e.*, os holons) efectuando partes do cálculo e conhecendo apenas parte dos dados do problema;
- utilização de *custos por operação por recurso* que contribui com mais um parâmetro para obtenção de soluções diferentes;
- possibilidade de existência de *múltiplos recursos para a execução de cada operação*, criando por isso várias alternativas para o escalonamento de uma tarefa tendo em conta as agendas de actividades de cada recurso e o tempo e o custo de execução de cada operação com os vários recursos alternativos;
- possibilidade de utilização de *um mesmo recurso em mais que uma operação* do plano;
- *não utilização de “comportamentos”* o que permite maior flexibilidade no tempo que medeia entre operações, o que implica porém a existência de *buffers* (assumidos de tamanho infinito);
- não consideração dos tempos de *setup* nem de transporte;

- utilização de um *operador de limitação inferior e superior* de agendas de intervalos para as fases de influência directa e inversa;
- *escolha de intervalos de escalonamento* com base em algum critério previamente definido (de momento os critérios definidos são: primeira proposta, proposta com menor custo e proposta com maior folga).

Embora este algoritmo se baseie no método descrito em [Almeida, 1995] e [Ramos et al., 1995], a distribuição do algoritmo, as mudanças efectuadas ao nível da combinação de intervalos e a não utilização de comportamentos são originais.

Contribuição – Desenvolvimento de um protótipo.

Para a validação dos conceitos e da arquitectura proposta neste trabalho e dando resposta ao objectivo 7, desenvolveu-se um sistema de carácter experimental (*i.e.*, um protótipo) denominado *Fabricare* (Figura 7.3, vide secção 6.3.7) desenvolvido de acordo com a arquitectura proposta (em termos de base de conhecimento e funcionamento) [Sousa et al., 1999b] [Sousa et al., 2000a].

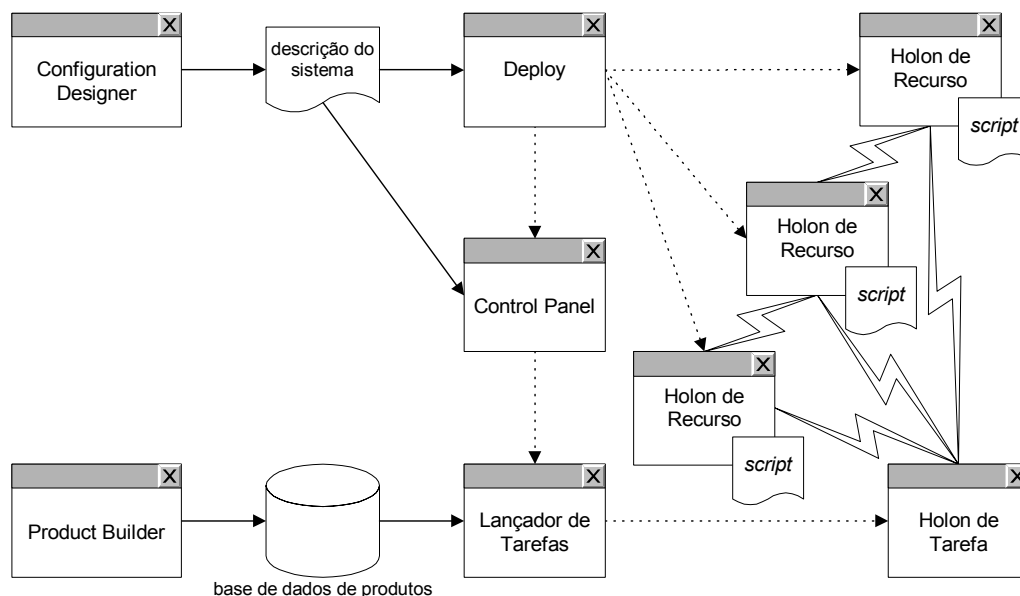


Figura 7.3 – Vista geral do protótipo Fabricare

O protótipo *Fabricare* pode ser caracterizado por:

- ser uma aplicação distribuída baseada nos paradigmas dos Sistemas Multiagente e dos Sistemas Holónicos;

- consistir numa série de aplicações divididas em três grandes grupos: directamente relacionadas com os holons da arquitectura; suporte ao funcionamento; e suporte à exploração do sistema;
- implementação dos Holons de Tarefa, Holons de Recurso, Holon de Escalonamento (apenas tratamento de conflitos), Holon de Planeamento de Processos (apenas obtenção de planos de produção) e Holons de Produto (simulados pelo Holon de Planeamento de Processos);
- utilização da linguagem de programação em lógica SICStus PROLOG para a implementação dos Programas em Lógica Estendida que compõem cada holon;
- utilização de Visual Basic para a interface gráfica com o utilizador;
- separação do conhecimento do controlo de cada holon, na forma: no Holon de Tarefa a informação acerca das tarefas a executar é obtida através da mensagem de anúncio; no Holon de Recurso através do *script* de cada recurso; e no Holon de Produto através do plano de produção;
- ser aberto e flexível no que toca ao número de holons em execução no sistema; *i.e.*, podem ser acrescentados novos holons de Tarefa, de Recurso e de Produto sem parar a execução do sistema e esses holons serão reconhecidos daí em diante (através do registo no Holon de Serviços de Directório);
- o sistema permite a utilização de vários critérios de selecção para a escolha de propostas, sendo inclusive extensível via *script*, permitindo a inclusão de novos algoritmos de selecção, o que permite pensar em critérios combinados (*e.g.*, menor custo com maior folga);
- utilização de um “quadro negro” como mecanismo de comunicação, simplificando o desenvolvimento dos diversos holons, embora introduza um elemento centralizador no sistema, o que pode gerar um “gargalo” de eficiência.

Este protótipo é original pois corresponde à implementação da arquitectura (original) que é proposta neste trabalho.

Nesta secção foram apresentadas as conclusões, bem como as contribuições do presente trabalho para o avanço do conhecimento e respectivas originalidades.

7.3 Tese

O problema que se tentou equacionar e resolver com este trabalho foi enunciado na forma de uma questão na secção 1.2 (transcrita em seguida).

Questão – *Como será possível manter a competitividade (e vitalidade) de uma empresa de Produção na ‘Sociedade de Consumo Personalizado’ do século XXI?*

De forma sucinta tudo o que foi dito ao longo deste trabalho traduz-se na tese enunciada de seguida:

Tese – *A divisão dos Sistemas de Produção em ‘pequenas’ unidades autónomas organizadas de acordo com a Teoria dos Sistemas Holónicos permite garantir a agilidade e vitalidade necessárias às empresas na nova sociedade.*

Essencialmente, defendeu-se que para fazer face à crescente personalização de bens de consumo e outras tendências da ‘Nova Economia’ (e.g., crescente complexidade de produtos com redução do ciclo de vida), é necessário construir sistemas que exibam características próprias, nomeadamente, distribuição e descentralização, autonomia, dinamismo, reactividade, flexibilidade, adaptabilidade e agilidade. Os actuais sistemas de produção integrada por computador não possuem tais características e mostram-se aquém das expectativas; no entanto, todas estas características podem ser observadas em sistemas construídos de acordo com a teoria dos sistemas holónicos. Esta teoria identifica uma entidade fundamental em todas as estruturas sociais e biológicas denominada *holon*, que é autónoma e cooperante com outros holons, sendo constituída por outros holons e fazendo parte de outros holons, apresentando por isso uma dualidade todo/parte que permite criar formas intermédias estáveis em sistemas complexos. Os sistemas holónicos de produção aplicam tais princípios aos sistemas de produção, sendo por isso o modelo sugerido para a construção de novos sistemas de produção de forma a responder aos requisitos em constante mudança que são colocados à empresa, sendo a sua grande vantagem (pelo menos teórica) a autonomia de cada holon e a sua estruturação dinâmica *ad hoc* em holarquias, permitindo assim o agrupamento de holons e a criação de subsistemas estáveis.

Na secção 1.5 foram apresentadas algumas questões adicionais que, embora já parcialmente respondidas ao longo do trabalho, serão agora abordadas de forma mais explícita.

- *Que suporte tecnológico se deve usar para a realização de sistemas de acordo com a Teoria dos Sistemas Holónicos?* – conforme o que foi já apresentado, o suporte tecnológico utilizado neste trabalho passa pela programação em lógica estendida e pelos sistemas multiagente. Esta abordagem não é obviamente a única, sendo possível recorrer a outras ferramentas, nomeadamente no que toca a linguagens de programação. No entanto, uma característica comum a todos os trabalhos que advogam a arquitectura holónica é o facto de se relacionarem com os sistemas multiagente.
- *Será possível utilizar a estrutura holónica noutras áreas que não a da Produção?* – a estrutura holónica tem as suas origens na observação de sistemas biológicos e sociais, sendo suficientemente genérica para ser utilizada noutras áreas que não a produção, em que a complexidade do sistema a desenvolver beneficie do conceito de holon como bloco de construção. Quanto à arquitectura proposta neste trabalho, é opinião do autor que a componente de escalonamento poderá ser utilizada noutras aplicações que não apenas a produção, nomeadamente em:
 - no escalonamento de projectos de construção civil, já que existem vários recursos e diferentes operações a executar, normalmente de uma única tarefa;
 - no comércio electrónico, usando tarefas que consistem apenas de uma operação (*i.e.*, comprar) e representando os vendedores como recursos, que dão as ordens de compra, indicando o custo e o prazo de entrega;
 - na atribuição de tempo de processamento a um programa numa rede de processadores, considerando para tal os processadores como recursos e o programa como tarefa, cujas operações são as unidades de código (*e.g.*, funções) a executar.
- *Que holons e holarquias há que definir de forma a abordar as várias actividades da produção de forma integrada?* – as entidades escolhidas para modelar o sistema foram os recursos, produtos e tarefas, os clientes e as vendas, os fornecedores e as compras. A escolha destas entidades deveu-se à análise efectuada das funções de fabrico e das funções comerciais, tendo sido identificados estes conceitos/entidades de negócio.
- *Existirão cenários de falta de informação e conhecimento nas actividades de produção? Se sim, quais?* – conforme foi apresentado na secção 5.4.2 existem vários cenários de falta de informação nos sistemas de produção. Os cenários identificados são fruto da experiência empírica do autor e do bom senso, sendo necessário de futuro fundamentar tais cenários com a escolha de situações reais.

- *De que forma deverão os vários elementos da arquitectura comunicar e interagir entre si?* – a interacção dos holons do sistema é feita através de um protocolo de negociação concebido propositadamente para este trabalho. A escolha de um mecanismo de negociação ao invés de um mecanismo pedido/resposta ou *master/slave* deve-se ao entendimento de que a flexibilidade e a independência (entre holons) conseguidas dessa forma são maiores.
- *Como saberão os vários elementos da existência de outros elementos?* – a informação sobre os holons existentes (em execução) no sistema é obtida através de um holon especial que implementa um serviço de directório. O conhecimento acerca da existência deste holon faz parte de cada holon, bem como as regras que definem o registo de funcionalidades no serviço de directório para assim poder ser requisitado por outros holons.
- *Como assegurar a coerência das decisões tomadas por cada elemento?* – no presente trabalho, a coerência de acções prende-se essencialmente com a relação de precedência temporal entre as operações de uma mesma tarefa, que é garantida pelo facto do protocolo de negociação utilizado incluir procedimentos explicitamente destinados para esse efeito.
- *Como formular, descrever e decompor o problema?* – podem ser considerados dois níveis neste trabalho; *i.e.*, a produção como um todo e o caso específico do escalonamento. Essencialmente foi feita uma descrição textual em termos de funcionalidades e de informação necessária às grandes entidades do sistema para suportar as várias actividades da produção. No caso do problema de escalonamento, a sua descrição é essencialmente efectuada pelo anúncio de tarefa, objecto de negociação, operação a operação, tendo em linha de conta os vários recursos existentes.

7.4 Limitações e Trabalho Futuro

O trabalho aqui apresentado tem algumas limitações (aliás dificilmente um qualquer sistema pode ser considerado como acabado e completo) que ao serem identificadas, permitem estabelecer boas perspectivas de desenvolvimento futuro.

De seguida apresentam-se algumas dessas limitações bem como sugestões sobre a forma de as ultrapassar, para além de algumas indicações sobre trabalho futuro:

- uma das grandes limitações do protótipo centra-se na falta de carácter temporal; *i.e.*, a inexistência de um “relógio interno” que modele a dinâmica do tempo no funcionamento de cada holon. Actualmente, o protótipo pode ser usado como um

escalonamento *a priori*. No entanto, a inclusão do ciclo de acompanhamento nos Holons de Tarefa e nos holons de Recurso (mesmo que simulado) permitiria desde já (embora de forma limitada) colmatar esta falha. A dinâmica temporal do sistema é um aspecto extremamente importante e será certamente alvo de trabalho futuro. Em termos conceptuais pensa-se que a utilização de *Extensões Temporais da Programação em Lógica* permitam modelar esse comportamento.

- a explosão combinatória do problema quando existem vários recursos alternativos pode tornar inviável a obtenção de uma solução em tempo útil. Aliás, essa é uma das limitações do sistema actual, pois não considera um limite máximo de tempo para gerar o escalonamento (uma situação normal quando se pretende efectuar escalonamento dinâmico *online*). Este será sem dúvida mais um dos focos de atenção em termos de trabalho futuro. O *Raciocínio Difuso* é uma solução possível para este problema, permitindo a criação de meta-regras baseadas na complexidade do problema e no tempo disponível para encontrar a solução, de forma a conduzir o processo de selecção de recursos a contactar (Figura 7.4).

se a complexidade do problema é elevada
e o tempo disponível é extremamente reduzido
então selecciona apenas um dos recursos menos ocupados para cada operação

se a complexidade do problema é elevada
e o tempo disponível é reduzido
então selecciona um subconjunto de recursos menos ocupados

se (a complexidade do problema é média
e o tempo disponível é aceitável)
ou (a complexidade do problema é elevada
e o tempo disponível é alargado)
então selecciona todos os recursos

Figura 7.4 – Exemplos de meta-regras para selecção de recursos

- também relacionado com a explosão combinatória e no sentido de minimizar o espaço de pesquisa, deve ser feito um estudo na tentativa de reduzir o número de combinações. Uma possibilidade, inspirada no algoritmo A*, é a utilização de um limite máximo de custo imposto pela tarefa e a consideração de custos parciais acumulados pelos recursos. Dessa forma, quando uma combinação parcial ultrapassa-se o limite imposto poderia ser abandonada, evitando assim a propagação de soluções “inviáveis”. De forma semelhante, o funcionamento dos Holons de Recurso poderia também ser modificado para retirar do espaço de soluções as combinações irrealizáveis, ou seja, aquelas que resultam num conjunto vazio de intervalos para um dos recursos.

- embora os holons do sistema possuam a capacidade de representar e manipular informação incompleta nas suas bases de conhecimento, a utilização desse tipo de informação no processo de decisão/operação é ainda limitado. Um dos usos de informação incompleta com maior potencial, na opinião do autor, é o planeamento de produção a médio/longo prazo. Nesta actividade, pode-se então levar em linha de conta as encomendas subespecificadas, considerando informação estatística sobre a utilização de materiais e duração das operações para efectuar um plano prévio e tentar assim antever problemas de falta de capacidade ou material.
- os casos identificados de Informação Incompleta são baseados no bom senso e no conhecimento empírico. De futuro seria vantajoso fundamentar tais casos, bem como casos adicionais, com base num inquérito e em entrevistas com os responsáveis a vários níveis (direcção geral, direcção de produção, operários) nas empresas.
- uma outra hipótese de trabalho futuro corresponde à modelação dos custos de operação dos holons. Os custos aqui referidos não se prendem com o custo de fabrico, pois isso sai fora do âmbito do trabalho, mas sim de custos em “ciber-euros” que de alguma forma permitam representar uma Economia formada pelos diversos holons. A função de custo de cada holon poderia então depender da situação do “mercado”; *e.g.*, um Holon de Recurso muito requisitado poderia aumentar os seus preços ao passo que um outro holon pouco requisitado poderia diminui-los (lei da oferta e da procura). Adicionalmente, cada holon poderia ter uma certa riqueza pessoal que lhe permitiria a contratação de serviços a outros holons.
- conforme foi referido anteriormente, neste trabalho apenas foram abordadas as holarquias estáticas da arquitectura. Um ponto de acção será certamente a exploração de holarquias dinâmicas e um estudo concreto das suas vantagens no funcionamento do sistema. De um ponto de vista teórico o conceito é extremamente interessante, sendo talvez necessário fundamenta-lo um pouco mais através da experimentação. Nesse sentido, será necessário definir o protocolo de formação de holarquias, especificando concretamente o mecanismo de holarquias dinâmicas (formação, mensagens trocadas, controlo, cooperação, adesão, remoção e término).
- a entrada de uma tarefa prioritária no sistema (*e.g.*, encomenda urgente de um grande cliente, ou encomenda importante devido ao seu valor) pode levantar algumas questões normalmente resolvidas recorrendo ao conhecimento tácito que o Director de Produção possui sobre como resolver essas situações (envolvendo o “sacrifício” de uma ou outra tarefa menos importante para dar lugar à tarefa prioritária, ou então, recorrendo a horas extras). Na melhor das hipóteses, o escalonamento da nova tarefa não acarreta

problemas se houver capacidade livre para escalonar essa tarefa. No entanto, essa é uma situação que raramente acontece na prática, indo a nova tarefa sobrepor-se a tarefas já escalonadas (Figura 7.5a). Nesta situação é necessária uma renegociação entre o novo Holon de Tarefa e os holons das tarefas já escalonadas, em que a tarefa prioritária terá que persuadir (ou mandar) os outros holons a libertarem o tempo de recurso que lhes foi atribuído (Figura 7.5b).

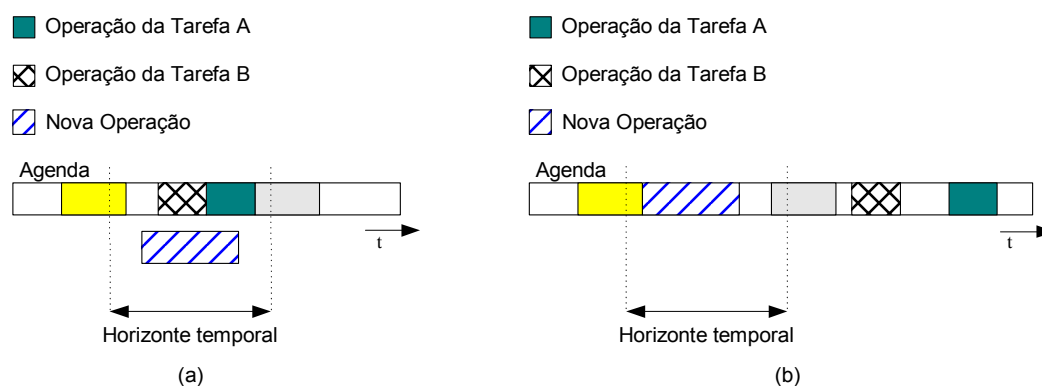


Figura 7.5 – Reescalamento para operação prioritária: (a) pedido; (b) re-arranjo

Mais uma vez, nesta situação, a utilização de raciocínio difuso, bem como a modelação de certas características mais antropomórficas nos holons (*e.g.*, egoísmo/altruísmo, capitalista/assalariado, chefe/empregado) poderá ser útil. Adicionalmente, considerando a Economia anteriormente referida, existem custos dos recursos que vão ver a agenda modificada e das tarefas que vão ceder tempo, tendo a nova tarefa que pagar a estes holons. Um factor importante a ter em conta é evitar o “efeito dominó” no reescalamento, ou seja, evitar que a mudança de uma tarefa implique mudanças em todas as tarefas. Existe uma certa sensibilidade de que os algoritmos genéticos [Holland, 1975] poderão ser úteis na minimização de alterações no escalonamento, tentando encontrar uma solução com o menor número possível de alterações.

- uma outra excepção que pode ocorrer é a alteração dinâmica de parâmetros da tarefa, nomeadamente, aumentar ou diminuir a quantidade a fabricar, atrasar ou adiantar a data de entrega. As situações de diminuir a quantidade a fabricar ou atrasar a data de entrega não acarretam grandes consequências, pois no primeiro caso, libertam tempo de recurso que pode vir a ser usado noutra tarefa e no segundo permitem uma maior folga em relação ao intervalo escalonado. O mesmo já não se pode dizer do aumento da quantidade ou do adiantamento da data de entrega, pois o escalonamento previamente efectuado pode tornar-se inviável. Nessa situação, uma negociação Tarefa a Tarefa semelhante à de Tarefas prioritárias é necessária para reescalonar a tarefa.

- a utilização de um “quadro negro” como único meio de comunicação coloca um gargalo de eficiência no sistema com o aumento do número de holons. Pensa-se que tal problema poderá ser ultrapassado recorrendo à existência de “quadros negros” distribuídos (um por cada holarquia) e canais de comunicação clássicos (*e.g.*, *sockets*) em cada holon. O uso do “quadro negro” para cada holarquia permite diminuir o número de conexões necessárias entre todos os membros da holarquia, funcionando ao mesmo tempo como área de memória partilhada que facilita a partilha de conhecimento.
- a ligação Visual Basic/SICStus PROLOG coloca também algumas limitações ao protótipo, pois introduz uma latência considerável no processamento de mensagens. Tal problema poderá ser resolvida considerando outras ferramentas para a componente de comunicações que permita um melhor desempenho da aplicação.
- uma outra limitação do protótipo centra-se na não consideração de transportes (disponibilidade momentânea e duração), bem como na modelação real dos *buffers* (dimensão e disponibilidade momentânea), da existência de ferramentas e de tempos de *setups*. Por forma a aproximar o protótipo de um sistema mais real e menos experimental, é necessário considerar estes aspectos. A existência de transportes, ferramentas e a capacidade momentânea dos *buffers* estão directamente relacionados com a modelação do carácter temporal no sistema, sendo, por isso, necessário trabalhar esse ponto em primeiro lugar.
- o sistema apresentado não tem em consideração a “experiência” anterior; *i.e.*, o sistema não aprende com vista a melhorar o seu desempenho. A *Aprendizagem Automática* poderá ser utilizada, por exemplo, nos Holons de Tarefa, criando uma memória colectiva com indicação da “credibilidade” dos recursos. Dessa forma, um Holon de Tarefa poderá decidir entre dois ou mais recursos com base não só nos intervalos e custos propostos, mas também no comportamento anterior desses recursos (*e.g.*, atraso na execução das operações, qualidade dos produtos fabricados).
- um desenvolvimento futuro refere-se à integração de três vertentes: produção, empresa virtual e comércio electrónico²³. Assim, o presente trabalho contribui com a componente de produção, que poderá ser distribuída e agrupada a um nível mais alto através da componente de empresas virtuais (formação e operação), sendo as relações de compras e vendas processadas automática e electronicamente pelo componente de comércio electrónico (*business-to-business* e *business-to-consumer*).

Conforme se pode constatar pela extensão desta secção há ainda imenso trabalho a fazer. Além das limitações, foram também apresentadas perspectivas e sugestões de novos caminhos a percorrer, alguns dos quais têm dimensão e abrangência para novos trabalhos de Doutoramento ou Mestrado, permitindo assim que este trabalho floresça, dando origem a um sistema integrado cada vez mais completo e útil para a sociedade ■

²³ Esta proposta de investigação foi aceite pela Fundação para a Ciência e Tecnologia, no projecto “*IMaGE – Integrated Tools for Computer Supported Manufacturing*” iniciado em Outubro de 2000.

Bibliografia

- Adam, E.; Vergison, E.; Kolski, C. and Mandiau, R. (1997) "Holonc User Drive Methodologies and Tools for Simulating Human Organizations". In: Hahn, W. and Lehmann, A. (Eds.) *Proceedings of the 9th European Simulation Symposium*, pp.57-61. October 19-22, 1997. Passau, Germany.
- Agre, J.; Elsley, G.; McFarlane, D.; Cheng, J. and Gunn, B. (1994) "Holonc Control of Cooling Control Systems". In *Proceedings of the Rensselaer's 4th International Conference on Computer Integrated Manufacturing and Automation Technology*.
- AIIM (URL) "Resource Center Report". AIIM International.
http://www.aiim.org/industry/resource/rc_report.html
- Albayrak, S. and Bussmann, S. (1996a) (Eds.) *Proceedings of the European Workshop on Agent-Oriented Systems in Manufacturing* (EWAOSM'96). Berlim, Alemanha, 26 e 27 de Setembro de 1996.
- Albayrak, S. and Bussmann, S. (1996b) "Introduction". In: S. Albayrak and S. Bussmann (Eds.), *Proceedings of the European Workshop on Agent-Oriented Systems in Manufacturing*, pp.i-ii. Berlim, Alemanha, 26 e 27 de Setembro de 1996.
- Almeida, A. (1995) "Escalonamento Dinâmico de Tarefas Industriais Sujeitas a Prazos de Entrega". Dissertação de Mestrado. Faculdade de Engenharia da Universidade do Porto. Setembro de 1995.
- Analide, C. e Neves, J. (1996) "Representação de Informação Incompleta". Unidade de Ensino. Departamento de Informática. Universidade do Minho. Braga, Portugal. Novembro 1996.
- Analide, C. e Neves, J. (1997) "Estruturas Hierárquicas com Herança". Unidade de Ensino. Departamento de Informática. Universidade do Minho. Braga, Portugal. Janeiro 1997.
- Atkinson, R. and Court, R. (1998) *The New Economy Index: Understanding America's Economic Transformation*. Technology, Innovation, and New Economy Project. Progressive Policy Institute. November 1998. (disponível em <http://www.neweconomyindex.org/>)
- Axelrod, R. (1984) *The Evolution of Cooperation*. Basic : New York, USA.
- Ayres, R. (1989) "Technology Forecast for CIM". *Manufacturing Review*, vol. 2(1), pp.43-52, 1989.

- Azevedo, A. and Sousa, J. (1997) "On the Design of an Order Promise System for Virtual Enterprises". *Proceedings of the Conference on Management and Control of Production and Logistics* (MCPL'97). Campinas, SP, Brazil.
- Baker, A.; Parunak, H. and Erol, K. (1997) "Manufacturing Over the Internet into Your Living Room: Perspectives From the AARIA Project". Center for Electronic Commerce Report CEC-0-9. (disponível em <http://www.aaria.uc.edu/cybermfg.ps>)
- Balasubramanian, S. and Norrie, D. (1995) "A multi-Agent Intelligent Design System Integrating Manufacturing and Shop-Floor Control". In: *Proceedings of the 1st International Conference on Multi-Agent Systems* (ICMAS'95). San Francisco, California, USA. 12-14 June 1995. (disponível em <http://ksi.cpsc.ucalgary.ca/DME/AnAgent.html>)
- Barram, D. (1994) "Keynote Address". North American Symposium – Preparing for the Full-Scale IMS Program". June 23, 1994, Dallas, Texas, USA. http://ksi.cpsc.ucalgary.ca/IMS/Dallas/IMS_Barram.html
- Baumgartel, H.; Bussmann, S. and Klosterberg, M. (1997) "Multi-Agent Co-ordination of Material Flow in a Car Plant". In *Proceedings of The Second International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agent Technology* (PAAM'97), pp.227-236. London, UK. 21-23 April 1997.
- Besant, C. and Lui, C. (1986) *Computer-Aided Design and Manufacture*. Ellis Horwood Limited: England.
- Bohez, E. and Limsombutan, B. (1999) "Curvature Analysis Based Holon For 5-Axis Milling Cutter Selection and Tool Path Planning". *Electronic Journal of the School of Advanced Technologies*, vol. 1(2). July 1999. Asian Institute of Technology. (ISSN 1513-1432) (disponível em <http://www.sat.ait.ac.th/ej-sat/articles/1.2/bh-full.html>)
- Bond, A. and Gasser, L. (1988) *Reading in Distributed Artificial Intelligence*. Morgan Kaufmann.
- Bongaerts, L.; Valckenaers, P.; Van Brussel, H. and Wyns, J. (1995) "Schedule Execution for a Holonic Shop Floor Control System". In *Proceedings of the Advanced Summer Institute* (ASI'95). 24-28 June, 1995. Lisbon, Portugal.
- Bongaerts, L.; Wyns, J.; Detand, J.; Van Brussel, H. and Valckenaers, P. (1996) "Identification of Manufacturing Holons". In: S. Albayrak and S. Bussmann (Eds.) *Proceedings of the 1st European Workshop on Agent Oriented Systems in Manufacturing*, pp.57-73. Berlin, Germany. September 1996.

- Bongaerts, L. (1998) "Integration of Scheduling and Control in Holonic Manufacturing Systems". PhD Thesis, Production and Automation Division, Katholieke Universiteit Leuven. Leuven, Belgium, December 1998.
- Booch, G. (1991) *Object Oriented Design with Applications*, Benjamin/Cummings: Redwood City, California, USA.
- Bradenburger, A. and Nalebuff, B. (1997) *Co-opetion: 1. A Revolutionary Mindset that Redefines Competition and Cooperation; 2. The Game Theory Strategy That's Changing the Game of Business*. Doubleday. (ISBN: 0385479506)
- Bradshaw, J. (1997a) "An Introduction to Software Agents". In: J. Bradshaw (Ed.) *Software Agents*. AAAI/MIT Press. (ISBN: 0262522349)
- Bradshaw, J. (1997b) "Software Agents: The Next Generation". Tutorial at the 2nd International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'97). London, UK. 21-23 April 1997.
- Brewka, G. (1996) "Well-founded Semantics for Extended Logic Programs with Dynamic Preferences". *Journal of Artificial Intelligence Research*, vol. 4, pp.19-36.
- Britannica (URLa) "Automation". Encyclopaedia Britannica Online, Britannica.com. <http://www.britannica.com/bcom/eb/article/0/0,5716,117180+1,00.html>
- Britannica (URLb) "Frederick Taylor". Encyclopaedia Britannica Online, Britannica.com. <http://www.britannica.com/bcom/eb/article/7/0,5716,73317+1+71464,00.html>
- Britannica (URLc) "Machine Tool". Encyclopaedia Britannica Online, Britannica.com. <http://www.britannica.com/bcom/eb/article/1/0,5716,115451+1,00.html>
- Britannica (URLd) "Great Depression". Encyclopaedia Britannica Online, Britannica.com. <http://www.britannica.com/bcom/eb/article/0/0,5716,38610+1+37849,00.html>
- Britannica (URLe) "Mass Production". Encyclopaedia Britannica Online, Britannica.com. <http://www.britannica.com/bcom/eb/article/4/0,5716,109274+1+106304,00.html>
- Britannica (URLf) "Industrial Relations". Encyclopaedia Britannica Online, Britannica.com. <http://www.britannica.com/bcom/eb/article/2/0,5716,43322+1,00.html>
- Britannica (URLg) "Work, History of the organization of – sophistication of mass production – scientific management". Encyclopaedia Britannica Online, Britannica.com. <http://www.britannica.com/bcom/eb/article/1/0,5716,115711+10,00.html>

- Bürckert, H.; Fisher, K. and Vierke, G. (1998) "Transportation Scheduling with Holonic MAS – The TeleTruck Approach". In H. Nwana and D. Ndumu (Eds.) *Proceedings of the Third International Conference and exhibition on The Practical Application of Intelligent Agents and Multi-agent Technology* (PAAM'97), pp.577-590. London, UK, May 23-25. (ISBN: 0952555484)
- Bussmann, S. and Müller, J. (1992) "A Negotiation Framework for Co-operating Agents". In S. Deen (Ed.) *Proceedings of CKBS-SIG*, pp.1-17. Dake Centre, University of Keele, 1992.
- Bussmann, S. (1998) "An Agent-Oriented Architecture for Holonic Manufacturing Control". In *Proceedings of the First International Workshop on Intelligent Manufacturing Systems* (IMSEurope'98), pp.1-12. Lausanne, Suíça, 15 a 17 de Abril de 1998.
- Butala, P. and Sluga, A. (1998) "A Pace Toward a Distributed Manufacturing System". In *Proceedings of the First International Workshop on Intelligent Manufacturing Systems* (IMS-Europe'98), pp.277-290. Lausanne, Suíça, 15 a 17 de Abril de 1998.
- Camarinha-Matos, L.; Afsarmanesh, H.; Garita, C. And Lima, C. (1997) "Towards an Architecture for Virtual Enterprises". In: L. Monostori (Ed.), *Proceedings of the Second World Congress on Intelligent Manufacturing Processes & Systems* (IMP&S'97), pp.531-541. Budapeste, Hungria, 10 a 13 de Junho de 1997.
- Carriero, N. and Gelernter, D. (1989a) "Linda in Context". *Communications of the ACM*, **32**(4), 1989.
- Carriero, N. and Gelernter, D. (1989b) "How to Write Parallel Programs: A Guide to the Perplexed". *ACM Computing Surveys*, September 1989.
- Castelfranchi, C. (1995). "Guarantees for Autonomy in Cognitive Agent Architecture". In: Wooldridge, M. and Jennings, N. (Eds.) *Intelligent Agents: Theories, Architectures, and Languages – Lecture Notes in Artificial Intelligence*, vol. **890**, pp.56-70. Springer-Verlag: Heidelberg, Germany.
- Chang, T. and Wysk, R. (1985) *An Introduction to Process Planning*. Prentice-Hall, Inc.
- Chen, F. and Su, C. (1996) "Vision-based Automated Inspection System in Computer Integrated Manufacturing". *The International Journal of Advanced Manufacturing Technology*, **11**(3):206-213.
- Chiariglione, L. (URL) "FIPA Rationale". Foundation for Intelligent Physical agents. http://www.csel.it/fipa/fipa_rationale.htm
- Clocksinn, W. and Mellish, C. (1981) *Programming in PROLOG*. Springer-Verlag.

- CMAF (URL) “What Do We Mean by ‘Agility’?”. Canadian Manufacturing Agility Forum.
<http://www.cmaf.com/info.html>
- Coelho, H. and Cotta, J. (1988) *PROLOG by Example: How to Learn, Teach and Use it*. Springer-Verlag.
- Coelho, H. (1995) *Inteligência Artificial em 25 Lições*. Serviço de Edição Fundação Calouste Gulbenkian: Lisboa, Portugal.
- Compton, W. (1988) *Design and Analysis of Integrated Manufacturing Systems*. National Academic Press: Washington DC, USA.
- Cooper, S. and Taleb-Bendiad, A. (1997) “CONCENSUS: A Multi-Parties Negotiation Support for Conflict Resolution in Concurrent Engineering Design”. In: L. Monostori (Ed.), *Proceedings of the Second World Congress on Intelligent Manufacturing Processes & Systems* (IMP&S’97), pp.320-324. Budapeste, Hungria, 10 a 13 de Junho de 1997.
- Courtois, A.; Pillet, M. e Martin, C. (1991) *Gestão da Produção*. LIDEL – Edições Técnicas, Lda.: Lisboa, Portugal.
- Cox, W. e Alm, R. (1998) *The Right Stuff: America’s Move to Mass Customization*. Federal Reserve Bank of Dallas (1998 Annual Report). (disponível em <http://www.dallasfed.org/htm/pubs/annual/arpt98.html>)
- CVM (1999) *Visionary Manufacturing Challenges for 2020*. Committee on Visionary Manufacturing, National Research Council (Ed.). National Academic Press: Washington DC, USA. (disponível em <http://www.nap.edu/readingroom/books/visionary/index.html>)
- Davis, R. (1980) “Report on the Workshop on Distributed Artificial Intelligence”. *SIGART Newsletter*, nº 73, pp.42-52.
- Davis, R. and Smith, R. (1983) “Negotiation as a Metaphor for Distributed Problem Solving”. *Artificial Intelligence*, vol. 20(1), pp.63-109.
- Davis, S. (1987) *Future Perfect*. Addison-Wesley, Co.
- Dertouzos, M.; Lester, R. and Solow, R. (1989) *Made in America: Regaining the Productive Edge*. MIT Press: Cambridge, MA, USA.
- Dilworth, J. (1992) *Operations Management – Design, Planning and Control for Manufacturing and Services*. McGraw-Hill, Inc.
- Drucker, P. (1994) “The Age of Social Transformation”. *The Atlantic Monthly*, November 1994. (disponível em <http://www.theatlantic.com/politics/ecbig/soctrans.htm>)

- Duffie, N.; Chitturi, R. and Mou, J. (1988) "Fault-tolerant Heterarchical Control of Heterogeneous Manufacturing System Entities". *Journal of Manufacturing Systems*, vol. 7(4), pp. 315-328.
- Durfee, E. and Rosenschein, J. (1994) "Distributed Problem Solving and Multi-agent Systems: comparisons and examples". In: *Proceedings of the 13th International Distributed Artificial Intelligence Workshop*, pp.94-104.
- Durfee, E.; Lesser, V. and Corkhill, D. (1989) "Trends in Cooperative Distributed Problem solving". *IEEE Transactions on Knowledge and Data Engineering*, vol. 1(1), pp.63-83.
- EAF (URLa) European Agility Forum. <http://www.agility-forum.de/>
- EAF (URLb) "Aims". European Agility Forum. <http://www.agility-forum.de/aim.html>
- Encarta (URLa) "Baruch, Bernard Mannes". Microsoft Encarta Online Encyclopedia. <http://encarta.msn.com/find/Concise.asp?z=1&pg=2&ti=761568198>
- Encarta (URLb) "Computer-Aided Design/Computer-Aided Manufacturing (CAD/CAM)". Microsoft Encarta Online Encyclopedia. <http://www.encarta.msn.com/find/Concise.asp?z=1&pg=2&ti=761577973#s1>
- Encarta (URLc) "Gore Vidal". Microsoft Encarta Online Encyclopedia. <http://encarta.msn.com/find/MediaMax.asp?pg=3&ti=761595762&idx=461542504>
- Encarta (URLd) "Mandelbrot, Benoit B." Microsoft Encarta Online Encyclopedia. <http://encarta.msn.com/find/Concise.asp?z=1&pg=2&ti=761579168>
- Ferber, J. (1993) "Modèle de Systèmes Multi-Agents: Du Réactif au Cognitive". In: *Proceedings of INFAUTOM'93*, pp.26-56. Toulouse, France. 18-19 February 1993.
- Ferreira, A. (1998) "Geração e Sequenciamento automático da Geometria das Áreas de Maquinagem". Dissertação de Mestrado, Universidade Portucalense Infante D. Henrique. Maio de 1998.
- Figueiredo, L. (1996) "Conversão e Execução de Planos de Processo". Dissertação de Mestrado. Faculdade de Engenharia, Universidade do Porto. Novembro de 1996.
- Finn, T.; Labrou, Y. and Mayfield, J. (1997) "KQML as an Agent Communication Language". In: Bradshaw, J. (Ed.) *Software Agents*. AAAI Press/MIT Press. (ISBN: 0-262-52234-9) (disponível em <http://www.cs.umbc.edu/kqml/papers/>)
- Fischer, K.; Müller, J.; Heimig, I. and Scheer, A. (1996) "Intelligent Agents in Virtual Enterprises". *Proceedings of the First International Conference on The Practical Application of Agents and Multi-Agent Technologies (PAAM'96)*, pp.205-223. Londres, Reino Unido, 22 a 24 de Abril de 1996.

- Fischer, K. (1999) "Agent-Based Design of Holonic Manufacturing Systems". *Journal of Robotics and Autonomous Systems*. Elsevier Science. (disponível em <http://www.dfki.de/~kuf/index.html>)
- FOLDOC (URLa) "Agent". The Free Online Dictionary of Computing. <http://www.instantweb.com/~foldoc/foldoc.cgi?query=agent>
- FOLDOC (URLb) "Computer Aided Engineering". The Free Online Dictionary of Computing. <http://www.instantweb.com/~foldoc/foldoc.cgi?Computer+Aided+Engineering>
- FOLDOC (URLc) "Manufacturer Resource Planning". The Free Online Dictionary of Computing. <http://www.instantweb.com/~foldoc/foldoc.cgi?Manufacturer+Resource+Planning>
- FOLDOC (URLd) "Manufacturing Automation Protocol". The Free Online Dictionary of Computing. <http://www.instantweb.com/~foldoc/foldoc.cgi?Manufacturing+Automation+Protocol>
- FOLDOC (URLe) "MRP". The Free Online Dictionary of Computing. <http://www.instantweb.com/~foldoc/foldoc.cgi?query=mrp>
- Foner, L. (1997) "Entertainment Agents: A Sociological Case Study". *Proceedings of the First International Conference on Autonomous Agents* (Agents'97). Marina del Rey, CA. 6-8 February 1997. (disponível em <http://foner.www.media.mit.edu/people/foner/Julia/Julia.html>)
- Fordyce, K.; Dunki-Jacobs, R.; Gerard, B.; Sell, R. and Sullivan, G. (1992) "Logistics Management System: An Advanced Decision Support System for the Fourth Decision Tier Dispatch or Short-Interval Scheduling". *Production and Operations Management*, vol. 1(1), pp.70-86.
- Franklin, S. and Graesser, A. (1997) "Is it an Agent, or Just a Program? a taxonomy for autonomous agents". In: Müller, J.; Wooldridge, M. and Jennings, N. (Eds.) *Intelligent Agents III – Agents Theories, Architectures and Languages*. Lecture Notes in Artificial Intelligence, vol. 1193, pp.21-35. Springer-Verlag. Berlin, Germany. (ISBN 3-540-62507-0) (disponível em <http://www.msci.memphis.edu/~franklin/AgentProg.html>)
- Frazer, J. and Oppenheim, J. (1997) "What's New About Globalisation". *The McKinsey Quarterly*, n° 2, p.172.
- Gantt, H. (1919) *Organising for Work*. Harcourt, Brace and Howe, New York.
- Garey M. and Johnson, D. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman: San Francisco.

- Gasser, B.; Braganza, C. and Hermann, H. (1987) "MACE: A Flexible Testbed for Distributed AI Research". In: M. Huhns (Ed.) *Distributed Artificial Intelligence*, pp.119-152. Pitman Publishing: London.
- Genesereth, M. and Ketchpel, S. (1994). "Software Agents". *Communications of the ACM*, **37**(7):48-53.
- Gentia (1997) "Agents Explained". Gentia Software Whitepaper.
- Gerber, C.; Slekmann, J. and Vlerke, G. (1999) "Holonc Multi-Agent Systems". DFKI Research Report RR-99-03. May 12, 1999. (disponível em <http://www.dfki.uni-kl.de/dfkidok/publications/RR/99/03/abstract.html>)
- Gini, M. and Boddy, M. (Eds.) (1998) *Working notes of the workshop on Agent Based Manufacturing at the Second International Conference on Autonomous Agents (Agents'98)*. Minneapolis/Saint Paul, Minnesota, EUA, 9 a 13 de Maio de 1998.
- Glistau, E. (1996) "Comparison to Other Concepts". In: S. Albayrak and S. Bussmann (Eds.) *Proceedings of the 1st European Workshop on Agent Oriented Systems in Manufacturing*, p.185. Berlin, Germany. September 1996.
- GM (WP) "Agent Based Manufacturing". White paper. General Motors Corporation.
- Good, M. (1994) "Keynote Address". North American Symposium - Preparing for the Full-Scale IMS Program. Dallas, Texas, USA, June 24, 1994. http://ksi.cpsc.ucalgary.ca/IMS/Dallas/IMS_Good.html
- Gou, L. and Luh, P. (1997) "Holonc Manufacturing Scheduling: Architecture, Cooperation Mechanism, and Implementation". In *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Tokyo, Japan. 16-20 June 1997. (disponível em <http://www.eng2.uconn.edu/msl/paper/ling-HMS/paper.html>)
- Gould, P. (1997) "What is Agility?". *IEE Manufacturing Engineer*, pp.28-31. February 1997. Institution of Electrical Engineers, UK.
- Green, S.; Hurst, L.; Nangle, B.; Cunningham, P.; Somers, F. and Evans, R. (1997) "Software Agents: A Review". Intelligent Agents Group Report. (disponível em http://www.cs.tcd.ie/research_groups/aig/iag/toplevel2.html)
- Grehan, R. (1994) "Good things in small packages are generating big product opportunities". *BYTE Magazine*. September 1994. (disponível em <http://www.byte.com/art/9409/sec6/art5.htm>)
- Groover, M. and Zimmers (jr.), E. (1984) *CAD/CAM Computer Aided Design and Manufacturing*. Prentice Hall, Inc.: Englewood Cliffs, New Jersey.

- Groover, M. (1987) *Automation, Production Systems and Computer Integrated Manufacturing*. Prentice-Hall, Inc.: Englewood Cliffs, New Jersey.
- Haddadi, A. (1995) *Communication and Cooperation in Agent Systems*. Lecture Notes in Artificial Intelligence, vol. **1056**. Springer-Verlag: Germany. (ISBN 3-540-61044-8)
- Hahndel, S.; Fuchs, F. and Levi, P. (1994) "Distributed Negotiation-Based Task Planning for a Flexible Manufacturing Environment". In: *Proceedings of the 6th European workshop on modelling autonomous agents in a multi-agent world*, pp.147-158. Odense, Denmark.
- Hatvani, J. and Nemes, L. (1978) "Intelligent Manufacturing Systems – A Tentative Forecast". In: A. Niemi (Ed.) *Proceedings of the VIIth IFAC World Congress*, vol. **2**, pp.895-899. June 12-16. Helsinki, Finland.
- Hatvani, J. (1983) "The efficient Use of Deficient Knowledge". *Annals of the CIRP*, vol. **32**(1), pp.423-425.
- Hayashi, H. (1993) "The IMS International Collaborative Program". In: *Proceedings of the 24th International Symposium on Industrial Robots*. Japan Industrial Robot Association, 1993.
- Hayes, C. (1998) "MAAP: An Agent Organization for Process Planning". In: Gini, M. and Boddy, M. (Eds.) *Working notes of the workshop on Agent Based Manufacturing at the Second International Conference on Autonomous Agents (Agents'98)*, pp.41-46. Minneapolis/Saint Paul, Minnesota, EUA, 9 a 13 de Maio de 1998.
- Hayes, R.; Wheelwright, S. and Clark, K. (1988) *Dynamic Manufacturing: Creating the Learning Organisation*. Free Press: New York, USA.
- Hayes-Roth, B. (1995) "An Architecture for Adaptive Intelligent Systems". *Artificial Intelligence: Special Issue on Agents and Interactivity*, vol. **72**, pp.329-365, 1995. (disponível em http://www-ksl.stanford.edu/KSL_Abstracts/KSL-93-19.html)
- Hazdra, T. and Pechoucek, M. (1999) "ProPlanT – A Multi-Agent System for Operative Planning of Production". In *Proceedings of The Fourth International Conference on The Practical Application of Agents and Multi-Agent Technologies (PAAM'99)*, pp.345-362. Londres, Reino Unido, 19 a 21 de Abril de 1999. (ISBN 1-902426-05-3)
- Heikkilä T.; Agostino N.; Rannanjärvi L. and Salonen P. (1999) "Feature-Based Product Modelling for Holonic Shot Blasting Systems". *Proceedings of the IEEE/IMACS CCSC Multiconference*. Athens, Greece, 4-8.July.1999.
- HMSC (URL) "HMS Questions and Answers". Holonic Manufacturing Systems Consortium. http://hms.ifw.uni-hannover.de/public/FAQ/fr_faq.htm

- Holland, J. (1975) *Adaptation in natural and artificial systems*. University of Michigan Press, 2nd edition, 1992.
- Höpf, M. (1994) “Holonc Manufacturing Systems – The Basic Concept and a Report of IMS Test Case 5”. In: J. Knudsen *et al.* (Eds.), *Sharing CIM Solutions*. IOS Press. (disponível em <http://hms.ifw.uni-hannover.de/public/Feasibil/holo2.htm>)
- Hunt, V. (1989) *Computer-Integrated Manufacturing Handbook*. Chapman & Hall. March, 1989. (ISBN 0412016516)
- IMS (URL) “Introduction to IMS – Structure of IMS”. Intelligent Manufacturing Systems Organization. <http://www.ims.org/intro/intro.htm>.
- IMS98 (1998) *Proceedings of the First International Workshop on Intelligent Manufacturing Systems* (IMS-Europe’98). Lausanne, Suíça, 15 a 17 de Abril de 1998.
- ISC (1994) “IMS: Intelligent Manufacturing Systems – A Program for International Cooperation in Advanced Manufacturing”. International Steering Committee Final Report adopted at ISC6. Hawaii, USA. 24-26 January 1994. http://ksi.cpsc.ucalgary.ca/IMS/ISC/ISC94_Content.html
- Jacobson, I. (1992) *Object-Oriented Software Engineering*, Addison-Wesley Readings: Massachusetts, USA.
- Jennings, N.; Sycara, K. and Wooldridge, M. (1998) “A Roadmap of Agent Research and Development”. *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 1, pp.275-306. Kluwer Academic Publishers, The Netherlands.
- Kádár, B.; Monostori, L. and Szelke, S. (1998) “An Object-oriented Framework for Developing Distributed Manufacturing Architectures”. *Journal of Intelligent Manufacturing – special issue on Agent Based Manufacturing*, vol. 9(2), pp.173-179. Chapman & Hall. (ISSN 0956-5515)
- Kalpakjian, S. (1995) *Manufacturing Engineering and Technology*. Third edition. Addison-Wesley Publishing Company. (ISBN: 0-201-84552-0)
- Kanchanasevee, P.; Biswas, G. Kawamura, K. and Tamura, S. (1997) “Contract-Net Based Scheduling for Holonic Manufacturing Systems”. *Proceedings of SPIE “Architectures, Networks and Intelligent Systems for Manufacturing Integration”*, pp.108-115. 15-16 October. Pittsburgh, Pennsylvania.
- Kaupunginkirjasto, K. (1999) “Arthur Koestler”. <http://www.kirjasto.sci.fi/koestler.htm>

- Kelley, R. (1999) *How to Be a Star at Work : 9 Breakthrough Strategies You Need to Succeed*. Times Books. (ISBN: 0812931696)
- Kelly, K. (1998) *New Rules for the New Economy: 10 Radical Strategies for a Connected World*. Penguin: USA. October 1999. (ISBN: 014028060X)
- Kidd, P. (2000) "Next Generation Manufacturing Enterprise Model". <http://www.cheshirehenbury.com/agility/ngmemodel.html>. September 2000.
- Kirsch, D.; Wiecezorek, D. and Albayrak, S. (1998) "Open Agent Architecture for the Realization of Holonic Manufacturing Systems". In *Proceedings of the First International Workshop on Intelligent Manufacturing Systems (IMS-Europe'98)*, pp.1-12. Lausanne, Suíça, 15 a 17 de Abril de 1998.
- Koestler, A. (1967) *The Ghost in the Machine*. Hutchinson & Co: London.
- Kollingbaum, M.; Heikkilä, T.; Peeters, P.; Matson, J.; Valckenaers, P.; McFarlane, D. and Bluemink, G. (2000) "Emergent flow shop control based on MASCADA agents". *Proceedings of IFAC Symposium On Manufacturing, Modelling, Management And Control (MIM2000)*. July 12-14, 2000, Patras, Greece.
- Kopacek, P. (ed.) (1999) *First IFAC Workshop on Multi-Agent Systems in Production (MAS'99)*. December 2-4, 1999. Vienna, Austria.
- Korper, S.; Ellis, J. and Gibson, J. (1999) *The E-Commerce Book: Building the E-Empire*. Academic Press. (ISBN: 0124211607)
- Kouiss, K.; Pierreval, H. and Mebarki, N. (1997) "Using Multi-Agent Architecture in FMS for Dynamic Scheduling". *Journal of Intelligent Manufacturing*, vol. 8, pp.41-47. Chapman & Hall.
- Kusiak, A. (1990) *Intelligent Manufacturing Systems*. Prentice-Hall, Inc.
- Lamma, E.; Moniz Pereira, L. and Riguzzi, F. (1998) "Learning with extended logic programs". *Proceedings of International Workshop On Non-Monotonic Reasoning at KR'98*, pp.99-108. Trento, Italy, May 30-June 1.
- Langer, G. (1999) "HoMuCS: A Methodology and Architecture for Holonic Multi-cell Control Systems". PhD Thesis. Technical University of Denmark, Department of Manufacturing Engineering. January 1999.
- Larouse (1995) *Enciclopédia Visual e Temática*. Selecções do Reader's Digest: Lisboa, Portugal. Janeiro de 1995.

- Laverty, J. e Demeestère, R. (1993) *Controlo de Gestão nas Empresas Industriais*. LIDEL – Edições Técnicas, Lda. : Lisboa, Portugal.
- Li, R. and Moniz Pereira, L. (1997) “Knowledge-Based Situated Agents Among Us: A preliminary Report”. In: Müller, J.; Wooldridge, M. and Jennings, N. (Eds.) *Intelligent Agents III - Lecture Notes in Artificial Intelligence*, vol. **1193**, pp.375-389. Springer-Verlag. Berlin, Germany. (ISBN 3-540-62507-0)
- Ligeza, A. (1997) “Logical Analysis of Completeness of Rule-Based Systems with Dual Resolution”. In *Proceedings of the European Symposium on the Verification and Validation of Knowledge Based Systems (EuroVAV’97)*, pp.155-165, Leuven, Bélgica, 1997.
- Lin, G. and Solberg, J. (1992) “Integrated Shop Floor Control Using Autonomous Agents”. *IIE Transactions*, vol. **24**(3), pp.57-71. July 1992.
- Lomuscio, A. and Colombetti, M. (1997) “QLB: A Quantified Logic for Belief”. In J. Müller, M. Wooldridge and N. Jennings (Eds.) *Intelligent Agents III – Agents Theories, Architectures and Languages*. Lecture Notes in Artificial Intelligence, vol. **1193**, pp.71-85. Springer-Verlag. Berlin, Germany. (ISBN 3-540-62507-0)
- Machado, J. e Neves, J. (1992) “Lógica Computacional – Volume I”. Unidade de ensino da Universidade do Minho. Departamento de Informática. Braga, Portugal.
- Maes, P. (1994) “Modelling Adaptive Autonomous Agents”. *Artificial Life Journal*, vol. **1**, n.º 1 & 2, pp.135-162, MIT Press, 1994.
- Maley, J. (1988) “Managing the Flow of Intelligent Parts”. *Robotics and Computer-Integrated Manufacturing*, vol. **4**(3/4), pp.525-530.
- Malone, T. and Crowston, K. (1994) “The Interdisciplinary Study of Co-ordination”. *ACM Computing Surveys*, vol. **26**(1), pp.87-119. March 1994.
- Martin, J. (1997) “Give ’em exactly what they want”. *Fortune*. November 10, 1997.
- Maturana, F. and Norrie, D. (1996) “Multi-Agent Architecture for Distributed Manufacturing”. *Journal of Intelligent Manufacturing*, vol. **7**, pp.257-270.
- McFarlane, D.; Marett, B.; Elsley, G.; Jarvis, D.; Wilbers, P. (1995) “Application of Holonic Methodologies to Problem Diagnosis in a Steel Rod Mill”. In *IEEE Int. Conference on Systems, Man and Cybernetics*.
- McHugh, P.; Merli, G. and Wheeler III, W. (1995) *Beyond Business Process Reengineering: Towards the Holonic Enterprise*. John Wiley & Sons: USA. (ISBN: 0-471-95087-4)

- Merchant, M. (1988) "The Percepts and Sciences of Manufacturing" *Robotics & Computer Integrated Manufacturing*, vol. 4(1/2), pp.1-6, 1988.
- Merrian-Webster (URLa). "agent". Merrian-Webster Online Dictionary. <http://www.m-w.com/cgi-bin/dictionary?agent>
- Merrian-Webster (URLb). "competing". Merrian-Webster Online Dictionary. <http://www.m-w.com/cgi-bin/dictionary?competing>
- Merrian-Webster (URLc). "coordinating". Merrian-Webster Online Dictionary. <http://www.m-w.com/cgi-bin/dictionary?coordinating>
- MESA (URL) "MESA International Glossary of Terms". Manufacturing Execution Systems Association. <http://www.mesa.org/html/main.cgi?sub=82>
- Monostori, L. (1997a) "Editorial". In: L. Monostori (Ed.), *Proceedings of the Second World Congress on Intelligent Manufacturing Processes & Systems (IMP&S'97)*, pp.vi-vii. Budapeste, Hungria, 10 a 13 de Junho de 1997.
- Monostori, L. (1997b) (Ed.) *Proceedings of the Second World Congress on Intelligent Manufacturing Processes & Systems (IMP&S'97)*. Budapeste, Hungria, 10 a 13 de Junho de 1997.
- Morley, R. and Schelberg, C. (1993) "An Analysis of a Plant-Specific Dynamic Scheduler". *Proceedings of the NSF Workshop on Dynamic Scheduling*. Cocoa Beach, FL, USA.
- Nagel and Dove (1991) "21st Century Manufacturing Enterprise Strategy". Iaccocca Institute Bethlehem P.A., 1991.
- NCMS (1996) "NCMS Collaborative Manufacturing Agenda". National Center for Manufacturing Sciences. NCMS Document 0040RE96, May 1996.
- Neves, J. (1984) "A Logic Interpreter to Handle Time and Negotiation in Logic Databases". In: *Proceedings of the ACM 1984 Annual Conference*. San Francisco, California, USA.
- Neves, J. e Machado, J. (1997a) "Rapid Prototyping in Concurrent Engineering – A Model and its Language". In: *Proceedings of the International Conference on Concurrent Engineering and Electronic Design Automation*. Erlangen, Germany. March 1997.
- Neves, J.; Machado, J.; Analide, C.; Novais, P. and Abelha, A. (1997b) "Extended Logic Programming Applied to the Specification of Multi-Agent Systems and Their Computing Environment". In: *Proceedings of the 1997 IEEE International Conference on Intelligent Processing Systems*. Beijing, China.

- NGM (1997) *Next Generation Manufacturing – A Framework for Action*, vol. I, “Summary Report”. Next Generation Project Report, Agility Forum, January 1997. (disponível em <http://imtr.ornl.gov/NGM/ngmhome.html> e http://www.dp.doe.gov/dp_web/next_gen.htm)
- Nilsson, N. (1981) “Distributed Artificial Intelligence”. Report SRI International, Menlo Park, CA.
- NRC (1986) *Towards a New Era in Manufacturing: The Need for a National Vision*. National Research Council, Manufacturing Studies Board (Ed.). National Academic Press: Washington DC, USA.
- NRC (1990) *Research Directions and Needs in US Manufacturing*. National Research Council, Manufacturing Studies Board Draft.
- Nwana, H. (1996a) “Software Agents: A Review”. *Knowledge Engineering Review*, vol. 11(3), pp.1-40, September 1996. (disponível em <http://www.cs.umbc.edu/paperrr/ao.ps>)
- Nwana, H.; Lee, L. and Jennings, N. (1996b) “Coordination in software agent systems”. *BT Technology Journal*, 14(4), pp.79-88. October 1996.
- Okino, N. (1989) “Bionical Manufacturing Systems”. In: T. Sata (Ed.) *Organisation of Engineering Knowledge for Product Modelling in Computer Integrated Manufacture*, pp.65-81. Elsevier: The Netherlands.
- Okino, N. (1993) “Bionic Manufacturing System”. In: J. Peklenik (Ed.) *CIRP, Flexible Manufacturing Systems Past-Present-Future*, pp.73-95.
- Osborne, M. and Rubinstein, A. (1994) *A Course in Game Theory*. MIT Press.
- Parker, M. (1997) “Common Problems”. *IEE Manufacturing Engineer*, pp.10-11. February 1997. Institution of Electrical Engineers, UK.
- Parunak, H. (1987) “Manufacturing Experience With the Contract Net”. In: M. Huhns (ed.) *Distributed Artificial Intelligence*, pp.285-310. Pitman Publishing: London.
- Parunak, H. (1991) “Characterizing the Manufacturing Scheduling Problem”. *Journal of Manufacturing Systems*, vol. 10(3), pp.241-258.
- Parunak, H. (1996) “Applications of Distributed Artificial Intelligence in Industry”. In: O’Hare and Jennings (Eds.) *Foundations of Distributed Artificial Intelligence*, chapter 4. Wiley Inter-Science.
- Parunak, H.; Baker, A. and Clark, S. (1997). “The AARIA Agent Architecture”. In: *Proceedings of the 1st International Conference on Autonomous Agents (Agents’97)*. Marina del Rey, CA. 6-8 February 1997. (disponível em <http://www.aaria.uc.edu/paper.html>)

- Parunak, H. (1998a) "What Can Agents Do in Industry and Why?". In: *Proceedings of the Second International Conference on Co-operative Information Agents (CIA'98)*. Paris, France. 3-8 July 1998.
- Parunak, H. (1998b) "The DASCh Experience: How to Model a Supply Chain". *Proceedings of International Conference on Computer Science (ICCS'98)*. (disponível em <http://www.erim.org/~vparunak/papers.htm>)
- Parunak, H. (1999a) "Agents While You Wait". Tutorial presented at 4th International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'99). London, UK, 29 April 1999.
- Parunak, H.; Ward, A.; Fleischer, M. and Sauter, J. (1999b) "The RAPPID Project: Symbiosis Between Industrial Requirements and MAS Research". *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 2(2), pp.111-140. June 1999.
- PDMIC (URL) "Glossary of Product Data Management Related Terms". Product Data Management Information Center. <http://www.pdmic.com/glossary/>
- Pereira, A. (1997) "Verificação e Validação de Sistemas Baseados em Conhecimento, VERITAS - uma Ferramenta de Verificação". Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto. Julho, 1997.
- Pine, B.; Davis, S. and Pine II, B. (1999) *Mass Customization: The New Frontier in Business Competition*. Harvard Business School Press: USA. (ISBN: 0875849466)
- Pontrandolfo, P. and Okogbaa, O. (1999) "Global Manufacturing: A Review and a Framework for Planning in a Global Corporation". *International Journal of Production Research*, vol. 37(1), pp.1-19.
- Porto Editora (URLa) "agente". Dicionário da Língua Portuguesa da Porto Editora. http://www.portoeditora.pt/dol/dicio_corpo.asp?operacao=def&palavra=agente
- Porto Editora (URLb) "hierarquia". Dicionário da Língua Portuguesa da Porto Editora. http://www.portoeditora.pt/dol/dicio_corpo.asp?operacao=def&palavra=hierarquia
- Rabelo, R. and Camarinha-Matos, L. (1994) "Negotiation in Multi-Agent Based Dynamic Scheduling". *Robotics & Computer Integrated Manufacturing*, vol. 11(4), pp.303-309. Elsevier Science Ltd.
- Rabon, L. and Scheller, H. (1997) "Mass Customization Making Progress". *Bobbin*, September 1997. Bobbin Group. (disponível em <http://www.bobbin.com/media/97sept/mc.htm>)

- Ramos, C.; Almeida, A. and Vale, Z. (1995) "Scheduling Manufacturing Tasks considering Due Dates: a new method based on Behaviours and Agendas". In: *International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pp.745-751. Melbourne, Australia.
- Ramos, C.; Rocha, J. and Vale, Z. (1998) "On the complexity of Precedence Graphs for Assembly and Task Planning". *Computers in Industry*, vol. 36, pp.101-111. Elsevier.
- Rannanjärvi, L. and Heikkilä, T. (1998) "Software Development for Holonic Manufacturing Systems". *Computers in Industry*, vol.37, pp.233-253. Elsevier.
- RCBI (1998) "Trends in Manufacturing". *Technology Update*, (RCBI Newsletter), Spring 1998. Robert C. Byrd Institute for Advanced Flexible Manufacturing. (disponível em <http://www.rcbi.org/rcbi/update/spring98/trendsmfg.htm>)
- Rembold, U. (1994) *Computer Integrated Manufacturing and Engineering*. Addison-Wesley.
- Riggs, J. (1987) *Production Systems: Planning, Analysis and Control*. John Wiley & Sons, Inc.
- Riley, L. and Cox, L. (1998) "Computer Integrated Manufacturing: Challenges and Barriers to Implementation". *The Technology Interface*, vol. 2(2), (Winter 98). (disponível em <http://et.nmsu.edu/~etti/winter98/manufacturing/riley/riley.html>)
- Rocha, J. (1999) "Representação, Análise da Complexidade, Planeamento e Execução de Processos Industriais". Tese de Doutoramento. Faculdade de Engenharia, universidade do Porto. Dezembro de 1999.
- Rumbaugh, J.; Blaha, M.; Premerlani, W.; Eddy, F. and Lorensen, W. (1991) *Object-Oriented Modelling and Design*. Prentice Hall: Englewood Cliifs.
- Rumbaugh J.; Booch, G. and Jacobson, I. (1997) *The Unified Modelling Language (UML) ver. 1.0*. Rational Co.: California, USA.
- Russell, S. and Norvig, P. (1995) *Artificial Intelligence: a modern approach*. Prentice Hall: New Jersey, USA. (ISBN:0-13-103805-2)
- Sadeh, N. (1994) "Micro-Opportunistic Scheduling: The Micro-Boss Factory Scheduler". In: M. Fox and M. Zweben (Eds.) *Intelligent Scheduling*. Morgan Kaufmann, 1994.
- Samadhi, T.; Ari, M. and Hoang, K. (1995) "Shared Computer-Integrated Manufacturing for Various Types of Production Environment". *International Journal of Operations & Production Management*, vol. 15(5), pp.95-109.

- Sandholm, T. and Lesser, V. (1996) "Advantages of a Leveled commitment Contracting Protocol". *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pp. 126-133. Portland, OR, USA.
- Santos, J.; Ramos, C.; Vale, Z. and Marques, A. (1999) "VERITAS - An Application for Knowledge Verification". In *Proceedings of the 11th International Conference on Tools with Artificial Intelligence (ICTAI'99)*, pp.441-444. IEEE Press, Chicago, November, 1999.
- Santos, M. (1999) "Sistemas de Classificação em Ambientes Distribuídos". Tese de Doutoramento. Escola de Engenharia, Universidade do Minho. 1999.
- Sargent, P. (1992) "Back to school for a brand new ABC". *The Guardian*, 12 March, p.28.
- Sauter, J. and Parunak, H. (1999) "ANTS In The Supply Chain". *Workshop on Agent based Decision Support for Managing the Internet-Enabled Supply Chain (Agents'99)*. Seattle, WA, 1 May 1999.
- Schonfeld, E. (1998) "The Customized, Digitized, Have-it-your-way Economy". *Fortune*. September 28, 1998.
- Schroeder, M.; Móra, I. and Moniz Pereira, L. (1997) "A deliberative and Reactive Diagnosis Agent Based on Logic Programming". In: Müller, J.; Wooldridge, M. and Jennings, N. (Eds.) *Intelligent Agents III – Agents Theories, Architectures and Languages*. Lecture Notes in Artificial Intelligence, vol. **1193**, pp.293-307. Springer-Verlag. Berlin, Germany. (ISBN 3-540-62507-0)
- Schwartz, E. (1997) *Webnomics: Nine Essential Principles for Growing Your Business on the World Wide Web*. Broadway Books. (ISBN: 0553061720)
- Seidman, A. (1995) "Computer-Integrated Manufacturing: An Opportunity Missed", *IIE Solutions*, vol. **27**(8), (August 1995), pp.8-9.
- Shen, W. and Norrie, D. (1998) "Combining Mediation and Bidding Mechanisms for Agent-Based Manufacturing Scheduling". In *Proceedings of the 2nd International conference on Autonomous Agents (Agents'98)*, pp.469-470. Minneapolis/St Paul, USA. 9-13 May 1998.
- Shen, W. and Norrie, D. (1999) "Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey". *Knowledge and Information Systems*, **1**(2):129-156. (disponível em <http://www.acs.ualgary.ca/~wshaen/papers/survey-abm.htm>)
- Sihn, W. (1997) "The Fractal Factory: A Practical Approach to Agility in Manufacturing". In: L. Monostori (Ed.) *Proceedings of The Second World Congress On Intelligent Manufacturing Processes & Systems*, pp.617-621, Budapest, Hungary, June 10-13, 1997

- Simon, H. (1962) "The Architecture of Complexity". *Proceedings of the American Philosophy Society*, vol. **106**(6), 1962.
- Simon, H. (1969) *The Sciences of the Artificial*. MIT Press: Cambridge, Massachusetts.
- Skiena, S. (1997) *The Algorithm Design Manual*. Springer-Verlag: New York.
- Smith, J.; Hoberecht, W. and Joshi, S. (1996). "A Shop Floor Control Architecture for Computer Integrated Manufacturing". *IIE Transactions*, **28**(10), pp.783-794.
- Smith, R. (1980) "The Contract Net Protocol". *IEEE Transactions on Computers*, vol. **C-29**(12).
- Solberg, J. and Kashyap, R. (1993) "ERC Research in Intelligent Manufacturing Systems". *Proceedings of the IEEE*, vol. **81**(1), pp.25-41. January 1993.
- Sousa, P. and Ramos, C. (1997) "Proposal of a Scheduling Holon for Manufacturing". In *Proceedings of Second International Conference on The Practical Application of Agents and Multi-Agents Technologies (PAAM'97)*, pp.255-268. Londres, Reino Unido, 21 a 23 de Abril de 1997. (ISBN 0-952554-6-8)
- Sousa, P. and Ramos, C. (1998) "A Dynamic Scheduling Holon for Manufacturing Orders". *Journal of Intelligent Manufacturing – special issue on Agent Based Manufacturing*, vol. **9**(2), pp.107-112. Chapman & Hall. (ISSN 0956-5515)
- Sousa, P. and Ramos, C. (1999a), "A Distributed Architecture and Negotiation Protocol for Scheduling in Manufacturing Systems". *Computers in Industry - special issue on Life Cycle Approaches to Production Systems: Management, Control and Supervision*, vol. **38**(2), pp.103-113. Março de 1999. Elsevier, Holanda. (ISSN 0166-3615)
- Sousa, P.; Ramos, C. and Neves, J. (1999b), "Contracting Tasks Between Autonomous Resources – An application to dynamic scheduling of manufacturing orders". In *Proceedings of The Fourth International Conference on The Practical Application of Agents and Multi-Agent Technologies (PAAM'99)*, pp.345-362. Londres, Reino Unido, 19 a 21 de Abril de 1999. (ISBN 1-902426-05-3)
- Sousa, P.; Ramos, C. e Neves, J. (2000a), "Fabricare: An Integrated View of a Distributed Manufacturing System". In: E. Pagello, F. Groen, T. Arai, R. Dillmann and A. Stentz (Eds.) *Proceedings of the 6th International Conference on Intelligent Autonomous Systems (IAS-6)*, pp.423-428. Veneza, Itália, 25 a 27 de Julho de 2000.
- Sousa, P.; Ramos, C. e Neves, J. (2000b), "Manufacturing Entities with Incomplete Information". *Studies in Informatics and Control Journal*, vol. **9**(2), pp.79-88, June 2000. National Institute for R&D in Informatics. Roménia.

- Sousa, P.; Silva, N.; Heikkilä, T.; Kallingbaum, M. and Valcknaers, P. (2000c), "Aspects of Co-operation in Distributed Manufacturing Systems". *Studies in Informatics and Control Journal*, vol. 9(2), pp.89-110, June 2000. National Institute for R&D in Informatics. Roménia.
- Stuart, A. (1996) "Five Uneasy Pieces: Part 2 – Knowledge Management". *CIO Magazine*, June 1996. (disponível em http://www.cio.com/archive/060196_uneasy_1.html)
- Suppes, P. and Hill, S. (1963) *First Course in Mathematical Logic*. Blaisdell Publishing Company, New York, USA.
- Sycara, K. (1989) "Multi-agent Compromise via Negotiation". In: L. Gasser & M. Huhns (Eds.), *Distributed Artificial Intelligence 2*. Morgan Kaufmann, 1989.
- Sycara, K. (1990) "Negotiation Planning: An AI Approach". *European Journal of Operational Research*, vol. 46, pp.216-234.
- Sycara, K.; Roth, S.; Sadeh, N. and Fox, M. (1991) "Coordinating Resource Allocation in Distributed Factory Scheduling". *IEEE Expert*, vol. 6(1) pp. 29-40. February, 1991.
- Tapscott, D. and Caston, A. (1993) *Paradigm Shift – The New Promise of Information Technology*. McGraw-Hill, Inc.:New York.
- Tharumarajah, A.; Wells, A. and Nemes, L. (1996) "Comparison of the Bionic, Fractal and Holonic Manufacturing Concepts". *International Journal of Computer Integrated Manufacturing*, vol. 9(3), pp.217-226. Taylor & Francis Ltd.
- Thurow, L. (1997) *The Future of Capitalism: How Today's Economic Forces Shape Tomorrow's World*. Penguin:USA (ISBN: 0140263284)
- Traylor, B. and Gelfond, M. (1993) "Representing Null Values in Logic Programming". In: *Proceedings of the International Logic Symposium (ILPS'93)*. Vancouver, British Columbia, Canada.
- Ueda, K. (1992) "A Concept for Bionic Manufacturing Systems Based on DNA-type Information". In: G. Olling and F. Kimura (Eds.) *Human Aspects in Computer Integrated Manufacturing*, pp.853-863.
- Ueda, K. (1994) *Biological Manufacturing Systems*. Kogyochosakai Pub. Co.:Tokyo.
- Ueda, K. and Vaario, J. (1997) "The Biological Manufacturing System: Adaptation to Growing Complexity and Dynamics in Manufacturing Environment". *New Manufacturing Era*, 29th CIRP International Seminar on Manufacturing Systems, pp.39-44. Osaka, Japan.

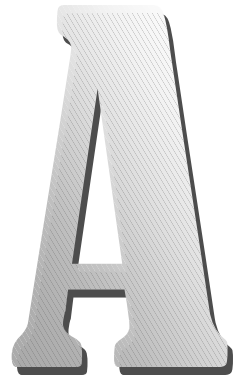
- Ulieru, M.; Stefanoiu, D. and Norrie, D. (2000) "Holonc Reconfiguration of Manufacturing Production by Fuzzy Multi-Agent Modelling". In: E. Pagello, F. Groen, T. Arai, R. Dillmann and A. Stentz (Eds.) *Proceedings of the 6th International Conference on Intelligent Autonomous Systems (IAS-6)*, pp.543-548. Veneza, Itália, 25-27 de Julho de 2000.
- USC (1990) "Making things better: Competing in Manufacturing". OTA-ITE-443. United States Congress, Office of Technology Assessment. US Government Printing Office: Washington DC.
- Valckenaers, P. and Van Brussel, H. (URL) "IMS TC5: Holonic Manufacturing Systems Technical Overview". http://hms.ifw.uni-hannover.de/public/Intro/fr_tech.htm
- Valckenaers, P.; Bonneville, F.; Van Brussel, H.; Bongaerts, L. and Wyns, J. (1994a) "Results of the Holonic Control System Benchmark at K.U. Leuven". In *Proceedings of International Conference on Computer Integrated Manufacturing and Automation Technology*, pp.128-133. Rensselaer Polytechnic Institute, Troy, NY, 10-12 October 1994.
- Valckenaers, P.; Van Brussel, H.; Bonneville, F.; Bongaerts, L. and Wyns, J. (1994b) "IMS Test Case 5: Holonic Manufacturing Systems". *IMS workshop at IFAC'94*. Vienna, Austria, June 13-15.
- Valckenaers, P.; Van Brussel, H.; Bongaerts, L. and Wyns, J. (1997) "Holonc Manufacturing Systems". *Journal of Integrated Computer Aided Engineer*, vol. 4(3), pp.191-201. John Willey & Sons, Inc.
- Van Brussel, H. (1995) "Working Group Proposal on Intelligent Manufacturing Systems". Document submitted to the Fourth Framework Programme of EC.
- Van Brussel, H.; Wyns, J.; Valckenaers, P.; Bongaerts, L.; and Peeters, P. (1998). "Reference Architecture for Holonic Manufacturing Systems: PROSA". *Computers in Industry - special issue on Intelligent Manufacturing Systems*, vol. 31(3), pp.255-276. 1998. Elsevier Science B.V.
- Van Brussel, H. and Valckenaers, P. (1999) (Eds.) *Proceedings of the Second International Workshop on Intelligent Manufacturing Systems (IMS'99)*. Leuven, Bélgica, 22 a 24 de Setembro de 1999. (ISBN 90-73802-69-5)
- Van Leeuwen, E. and Norrie, D. (1997) "Holons and Holarchies". *IEE Manufacturing Engineer*, pp.86-88. April 1997. Institution of Electrical Engineers, UK.

- Váncza, J. and Márcus, A. (1998) "Holonc Manufacturing with Economic Rationality". In *Proceedings of the First International Workshop on Intelligent Manufacturing Systems (IMS-Europe'98)*, pp.1-12. Lausanne, Suíça, 15 a 17 de Abril de 1998.
- Verbo (URLa) "agente". Dicionário Enciclopédico Verbo. Mind-Verbo & Novis. <http://enciclopediaverbo.clix.pt/cgi-bin/consulta.cgi?artigo=agente>
- Verbo (URLb) "Guerra Mundial". Dicionário Enciclopédico Verbo. Mind-Verbo & Novis. <http://enciclopediaverbo.clix.pt/cgi-bin/consulta.cgi?artigo=guerra+mundial>
- Warnecke, H. (1993) *The Fractal Company A Revolution In Corporate Culture*. Springer-Verlag: Germany.
- Weng, M. and Ren, H. (URL) "A Review of Multi-agent Scheduling". <http://www.eng.usf.edu/~ren/agentrev.html>
- White, J. E. (1994). "Telescript Technology: The Foundation for the Electronic Marketplace". White paper, General Magic, Inc., 2465 Latham Street, Mountain View, CA 94040.
- Wooldridge, M. (1992) "The Logical Modelling of Computational Multi-Agent Systems". PhD Thesis, Faculty of Technology, University of Manchester. August 1992.
- Wooldridge, M. and Jennings, N. (1995) "Intelligent Agents: Theory and Practice". *Knowledge Engineering Review*, October 1995. (disponível em <http://www.elec.qmw.ac.uk/dai/pubs/>)
- Wooldridge, M. (1997) "Agent-Based Software Engineering". *IEE Proceedings on Software Engineering*, **144**(1), pp.26-37, February 1997.
- Wooldridge, M. (1999) "Intelligent Agents: Introduction". In: G. Weiss; W. van der Hoek and M. Wooldridge (Eds.) *Proceedings of the First European Agent Systems Summer School*. University of Utrecht, The Netherlands. July 26-30, 1999.
- Wyns, J. (1999) "Reference Architecture for Holonic Manufacturing Systems – the key to evolution and reconfiguration". PhD Thesis, Production and Automation Division, Katholieke Universiteit Leuven. Leuven, Belgium, March 1999.
- Yoshikawa, H. (1993) "Intelligent Manufacturing Systems: Technical cooperation that transcends cultural differences". Information Infrastructure Systems for Manufacturing, *IFIP Transactions B-14*. 1993. Elsevier Science B.V.: Amsterdam, Holland.
- Zohoor, M. (1998) "Interactive Computer Network System Based on Computer Integrated Manufacturing". In *Proceedings of The International Conference on Collaborative Networked Learning*, February 16-18. New Deli, India.

APÊNDICES

APÊNDICE A

EXPERIÊNCIAS E RESULTADOS



Este apêndice apresenta algumas experiências realizadas com o protótipo com vista ao escalonamento de tarefas industriais. Para tal, realizaram-se as experiências do método de [Almeida, 1995] e compararam-se os resultados obtidos. Numa segunda fase realizou-se outro conjunto de experiências com vista a mostrar as novas potencialidades do método de escalonamento deste trabalho.

A.1 Experiências do Método Original

As três primeiras experiências do trabalho de [Almeida, 1995] utilizavam uma mesma tarefa com o plano de produção da Figura A.1, para produzir cinco itens numa janela temporal entre o instante zero e o instante setenta.



Figura A.1 – Plano utilizado nas experiências

A última experiência do trabalho de [Almeida, 1995] considera três tarefas em simultâneo e será apresentada na secção própria.

A.1.1 Exemplo 1

O primeiro exemplo [Almeida, 1995, pp.98-103] considera os seguintes dados de entrada:

Tabela A.1 – Condições de teste da experiência original n° 1

Recurso	Operações	Duração	Agenda inicial
M1	op1	1	[(10,15), (45,50), (65,70)]
M2	op2	3	[(20,21), (40,48)]
M3	op3	2	[(20,23), 42,51)]

Os resultados de ambos os sistemas são apresentados na Figura A.2 correspondentes aos intervalos da Tabela A.2.

Tabela A.2 – Resultados da experiência original n° 1

Recurso	Proposta método original	Proposta <i>Fabricare</i>	Custo
M1	[(20,37), (50,65)]	[(0,10), (15,36), (50,64)]	500
M2	[(21,40), (51,68)]	[(1,17), (21,39), (51,67)]	500
M3	[(24,42), (54,70)]	[(8,20), (28,42), (58,70)]	1500

No sistema *Fabricare* considerando o critério de selecção “primeira”, o resultado seria M1/(0,10), M2/(1,17) e M3/(8,20).

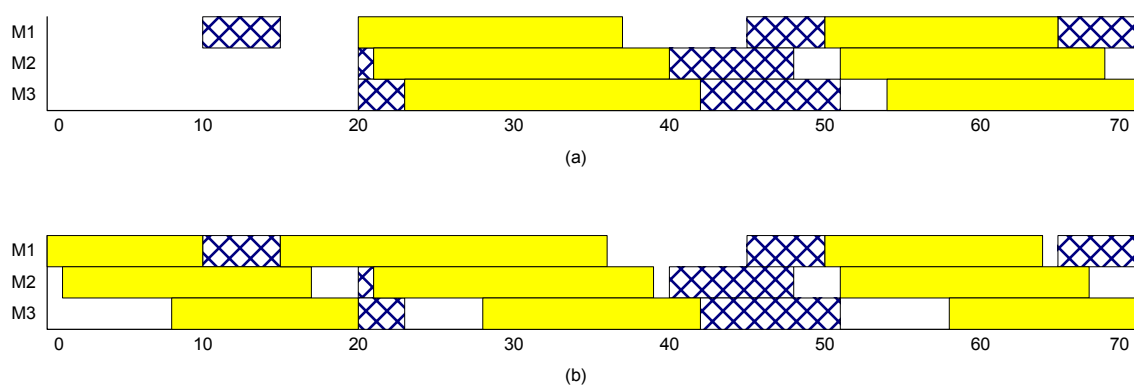


Figura A.2 – Resultado da experiência original n° 1

As diferenças de resultados advêm principalmente da utilização de comportamentos considerando “buracos” no escalonamento e *buffers* de tamanho zero, enquanto que o sistema

Fabricare considera a existência de *buffers* de tamanho infinito, permitindo-lhe assim considerar válido um novo intervalo de tempo no início das agendas dos recursos.

A.1.2 Exemplo 2

O segundo exemplo [Almeida, 1995, pp.103-109] considera os seguintes dados de entrada:

Tabela A.3 – Condições de teste da experiência original nº 2

Recurso	Operações	Duração	Agenda inicial
M1	op1	1	[(10,15), (45,53), (65,70)]
M2	op2	3	[(20,21), (40,48)]
M3	op3	2	[(20,23), (42,58)]

Os resultados de ambos os sistemas são apresentados na Figura A.3 correspondentes aos intervalos da Tabela A.4.

Tabela A.4 – Resultados da experiência original nº 2

Recurso	Proposta método original	Proposta <i>Fabricare</i>	Custo
M1	[(20,37)]	[(0,10), (15,36)]	500
M2	[(21,40)]	[(1,17), (21,39)]	500
M3	[(24,42)]	[(8,20), (28,42)]	1500

No sistema *Fabricare* considerando o critério de selecção “primeira”, o resultado seria M1/(0,10), M2/(1,17) e M3/(8,20).

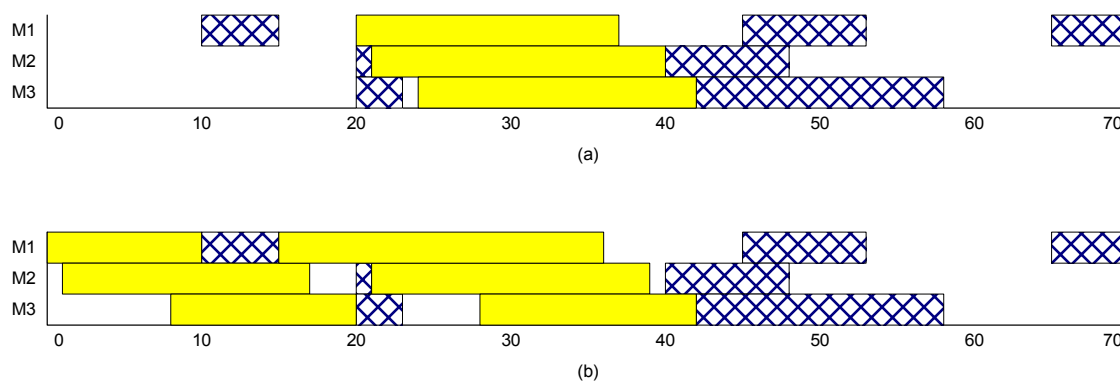


Figura A.3 – Resultado da experiência original nº 2

As diferenças de resultados advêm principalmente da utilização de comportamentos considerando “buracos” no escalonamento e *buffers* de tamanho zero, enquanto que o sistema *Fabricare* considera a existência de *buffers* de tamanho infinito, permitindo-lhe assim considerar válido um novo intervalo de tempo no início das agendas dos recursos.

A.1.3 Exemplo 3

O terceiro exemplo [Almeida, 1995, pp.109-112] considera os seguintes dados de entrada:

Tabela A.5 – Condições de teste da experiência original n° 3

Recurso	Operações	Duração	Agenda inicial
M1	op1	1	[(10,15), (35,53), (65,70)]
M2	op2	3	[(20,21), (40,48)]
M3	op3	2	[(20,23), (36,58)]

Os resultados de ambos os sistemas são apresentados na Figura A.4 correspondentes aos intervalos da Tabela A.6.

Tabela A.6 – Resultados da experiência original n° 3

Recurso	Proposta método original	Proposta <i>Fabricare</i>	Custo
M1	[]	[(0,10)]	500
M2	[]	[(1,17)]	500
M3	[]	[(8,20)]	1500

No sistema *Fabricare* considerando o critério de selecção “primeira”, o resultado seria M1/(0,10), M2/(1,17) e M3/(8,20).

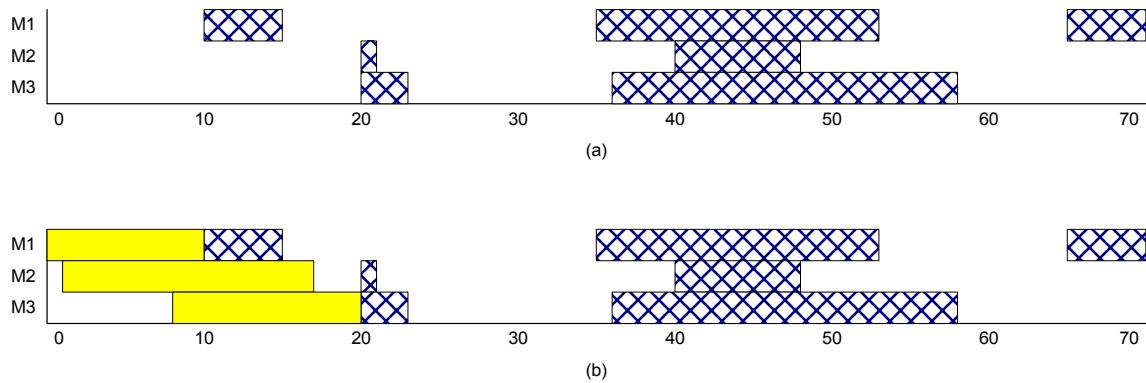


Figura A.4 – Resultado da experiência original n° 3

As diferenças de resultados advêm principalmente da utilização de comportamentos considerando “buracos” no escalonamento e *buffers* de tamanho zero, enquanto que o sistema *Fabricare* considera a existência de *buffers* de tamanho infinito, permitindo-lhe assim considerar válido o intervalo de tempo no início das agendas dos recursos, ao passo que o método original não propunha nenhuma solução.

A.1.4 Exemplo 4

O último exemplo de [Almeida, 1995, pp.123-138] considera uma situação com três tarefas para produção de dois itens de cada produto (cujos planos de produção podem ser vistos na Figura A.5) no intervalo entre o instante zero e o instante vinte.

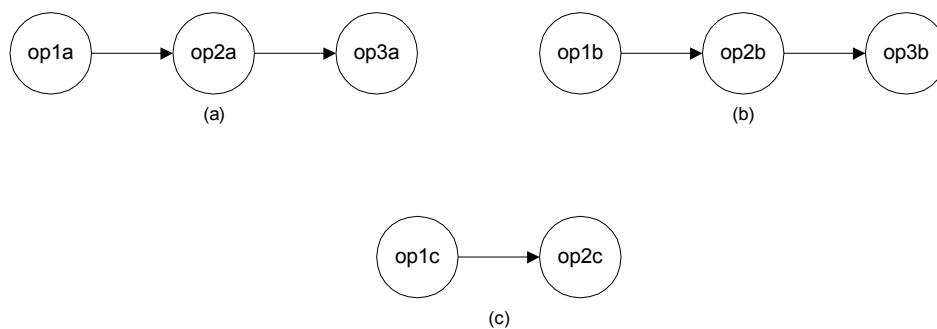


Figura A.5 – Planos utilizado na experiência n° 4

Este exemplo usa os seguintes dados de entrada:

Tabela A.7 – Condições de teste da experiência original n° 4

Recurso	Operações	Duração	Agenda inicial
M1	op1a	2	[(5,7), (11,13), (18,20)]

Recurso	Operações	Duração	Agenda inicial
	op3b	1	
	op1c	2	
M2	op2a	3	[(8,10), (15,18)]
	op1b	2	
M3	op3a	1	[(1,5), (16,19)]
	op2b	1	
	op2c	1	

Os resultados de ambos os sistemas são apresentados na Figura A.2 correspondentes aos intervalos da Tabela A.8.

Tabela A.8 – Resultados da experiência original n° 4

Tarefa	Recurso	Proposta método original	Proposta Fabricare	Custo
A	M1	[(0,5)]	[(0,5)]	200
	M2	[(2,8)]	[(2,8)]	200
	M3	[(5,9)]	[(7,16)]	600
B	M1	[(7,10), (13,17)]	[(14,18)]	200
	M2	[(4,8), (10,15)]	[(10,14)]	200
	M3	[(6,9), (12,16)]	[(13,16)]	600
C	M1	[(7,11)]	[(7,11)]	200
	M3	[(9,12)]	[(10,13)]	600

No sistema *Fabricare* considerando o critério de selecção “primeira”, o resultado seria M1/[(0,5)], M2/(2,8) e M3/(7,16) para a tarefa A, M2/(10,14), M3/(13,16) e M1/(14,18) para a tarefa B e M1/(7,11) e M3/(10,13) para a tarefa C.

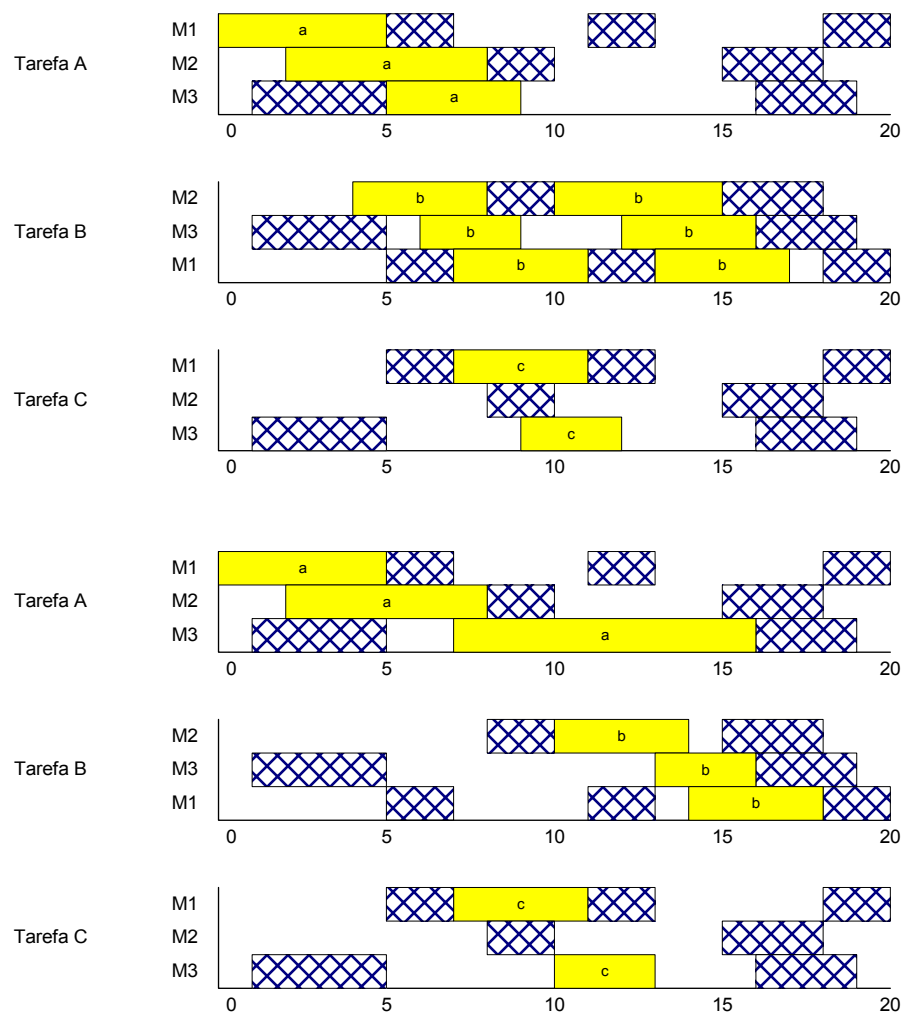


Figura A.6 – Resultado da experiência original n° 4

No método original, são utilizados comportamentos considerando “buracos” no escalonamento e *buffers* de tamanho zero, além disso, o método é executado para cada tarefa considerando sempre a situação inicial. No sistema *Fabricare* considera-se a existência de *buffers* de tamanho infinito, e adicionalmente, as várias tarefas foram executadas em sequência, quer isto dizer, que o escalonamento considerado para a tarefa *A* limitou os intervalos livres da tarefa *B* e por sua vez o escalonamento da tarefa *B* limitou os intervalos livres da tarefa *C*.

A.2 Experiências *Fabricare*

Para este segundo conjunto de experiências utilizou-se uma tarefa para produzir dois itens de um produto segundo um plano de produção de acordo com a Figura A.1 na janela temporal entre o instante zero e o instante vinte e quatro, considerando as condições de entrada descritas na Tabela A.9.

Tabela A.9 – Condições de teste das experiências Fabricare

Recurso	Operações	Duração	Custo	Agenda inicial
R1	op1	1	100	[(1,2)]
	op3	3	350	
R2	op2	1	100	[(6,7)]
R3	op3	3	300	[(7,10)]
R4	op1	1	90	[(1,2), (3,6), (10,12), (17,18)]
R5	op2	1	100	[]

A.2.1 Exemplo 1

Esta experiência considera a existência de um recurso alternativo para a execução da operação *op1*. Os resultados de ambos os sistemas são apresentados na Figura A.7 correspondentes aos intervalos da Tabela A.10.

Tabela A.10 – Resultados da experiência Fabricare nº 1

Proposta	Custo	Recurso	Intervalos Propostos	Custo Parcial
R1-R2-R3	1000	R1	[(2,20)]	200
		R2	[(7,21)]	200
		R3	[(10,24)]	600
R4-R2-R3	980	R4	[(6,10), (12, 17), (18, 20)]	180
		R2	[(7,21)]	200
		R3	[(10,24)]	600

Considerando o critério de selecção “*menor custo*”, o resultado seria R4/(6,10), R2/(7,21) e R3/(10,24).

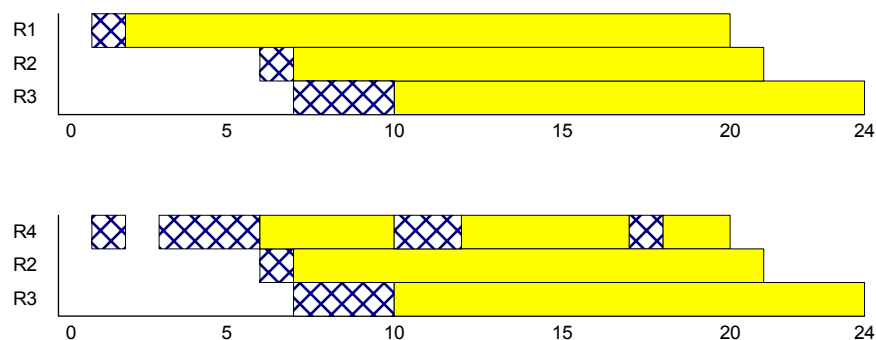


Figura A.7 – Resultado da experiência Fabricare nº 1

A utilização de recursos alternativos permite criar soluções diferentes baseadas nos intervalos livres de cada recurso. Adicionalmente, neste exemplo o custo de operação nos recursos alternativos é também diferente permitindo assim um outro factor de diferenciação.

A.2.2 Exemplo 2

Este exemplo considera a existência de um recurso alternativo para a operação *op2*. Os resultados de ambos os sistemas são apresentados na Figura A.8 correspondentes aos intervalos da Tabela A.11.

Tabela A.11 – Resultados da experiência Fabricare nº 2

Proposta	Custo	Recurso	Intervalos Propostos	Custo Parcial
R1-R5-R3	1000	R1	[(2,20)]	200
		R5	[(3,21)]	200
		R3	[(10,24)]	600
R1-R2-R3	1000	R1	[(2,20)]	200
		R2	[(7,21)]	200
		R3	[(10,24)]	600

Considerando o critério de selecção “*menor custo*”, o resultado seria R1/(2,20), R2/(7,21) e R3/(10,24).

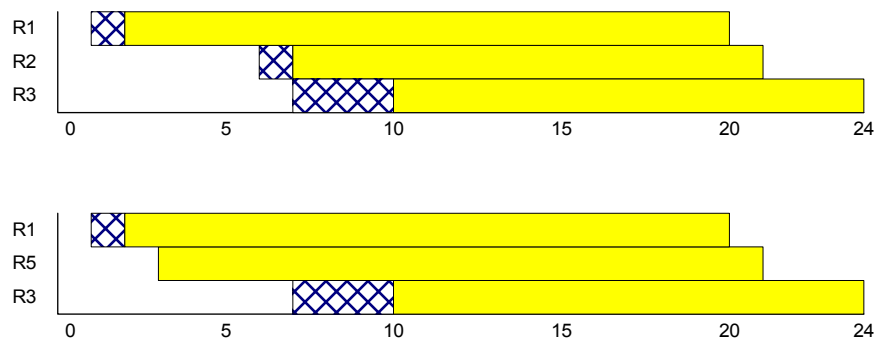


Figura A.8 – Resultado da experiência Fabricare n° 2

Neste exemplo, a utilização de recursos alternativos permite criar soluções diferentes baseadas apenas nos intervalos livres de cada recurso, já que o custo de operação nos recursos alternativos é idêntico.

A.2.3 Exemplo 3

Este exemplo considera a existência de recursos alternativos para as operações *op1* e *op2*. Os resultados de ambos os sistemas são apresentados na Figura A.9 correspondentes aos intervalos da Tabela A.12.

Tabela A.12 – Resultados da experiência Fabricare n° 3

Proposta	Custo	Recurso	Intervalos Propostos	Custo Parcial
R1-R5-R3	1000	R1	[(2,20)]	200
		R5	[(3,21)]	200
		R3	[(10,24)]	600
R1-R2-R3	1000	R1	[(2,20)]	200
		R2	[(7,21)]	200
		R3	[(10,24)]	600
R4-R5-R3	980	R4	[(6,10), (12,17), (18,20)]	180
		R5	[tw(7,21)]	200
		R3	[tw(10,24)]	600
R4-R2-R3	980	R4	[(6,10), (12, 17), (18, 20)]	180
		R2	[(7,21)]	200
		R3	[(10,24)]	600

No sistema *Fabricare* considerando o critério de selecção “*menor custo*”, o resultado seria R4/(6,10), R2/(7,21) e R3/(10,24).

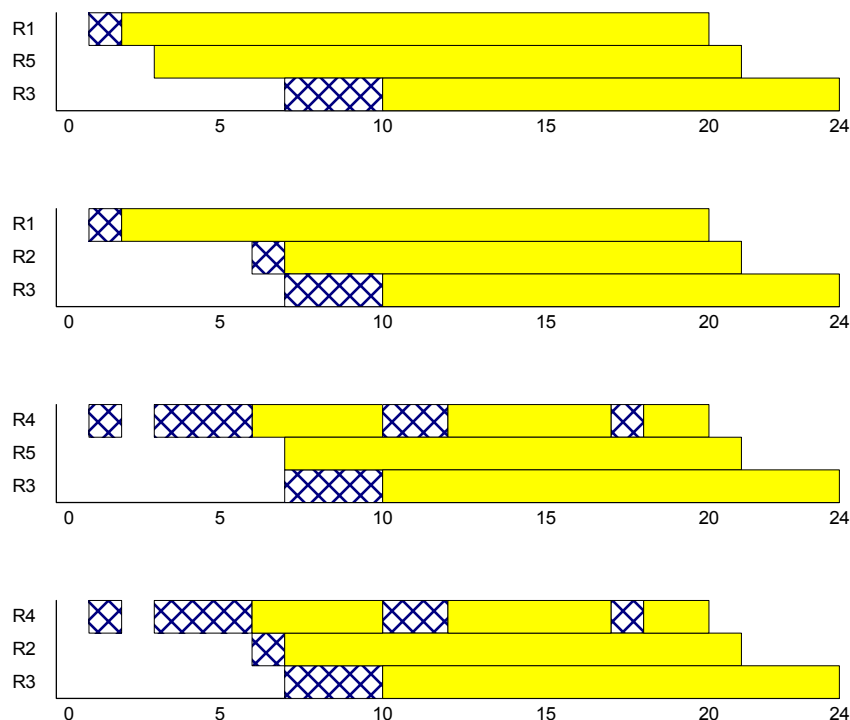


Figura A.9 – Resultado da experiência *Fabricare* n° 3

Este exemplo considera uma combinação de vários recursos alternativos para duas das operações da tarefa bem como custos diferentes de execução dessas operações.

A.2.4 Exemplo 4

Este exemplo considera a utilização de um mesmo recurso para duas operações do plano (*op1* e *op3*). Os resultados de ambos os sistemas são apresentados na Figura A.10 correspondentes aos intervalos da Tabela A.13.

Tabela A.13 – Resultados da experiência *Fabricare* n° 4

Proposta	Custo	Recurso	Intervalos Propostos	Custo Parcial
R1-R2-R1	1100	R1 (op1)	[(2,20)]	200
		R2 (op2)	[(3,6), (7,21)]	200
		R1 (op3)	[(4,24)]	700

No sistema *Fabricare* considerando o critério de selecção “*menor custo*”, o resultado seria $R1(op1)/(2,20)$, $R2(op2)/(3,26)$ e $R1(op3)/(4,24)$.

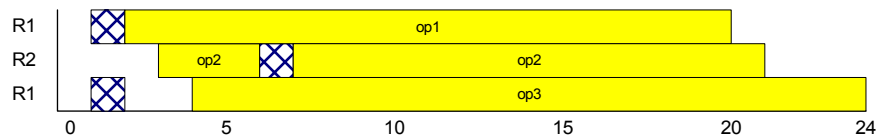


Figura A.10 – Resultado da experiência *Fabricare* nº 4

Este exemplo demonstra a utilização de um mesmo recurso para mais que uma operação, podendo ver-se nos intervalos sugeridos a influência que o recurso exerce sobre ele próprio.

A.2.5 Exemplo 5

Este exemplo considera a existência de recursos alternativos para duas operações (*op1* e *op3*), em que um dos recursos pode executar ambas as operações. Os resultados de ambos os sistemas são apresentados na Figura A.11 correspondentes aos intervalos da Tabela A.14.

Tabela A.14 – Resultados da experiência *Fabricare* nº 5

Proposta	Custo	Recurso	Intervalos Propostos	Custo Parcial
R1-R2-R3	1000	R1	[(2,20)]	200
		R2	[(7,21)]	200
		R3	[(10,24)]	600
R1-R2-R1	1100	R1 (op1)	[(2,20)]	200
		R2 (op2)	[(3,6), (7,21)]	200
		R1 (op3)	[(4,24)]	700

Considerando o critério de selecção “*menor custo*”, o resultado seria $R1/(2,20)$, $R2/(7,21)$ e $R3/(10,24)$.

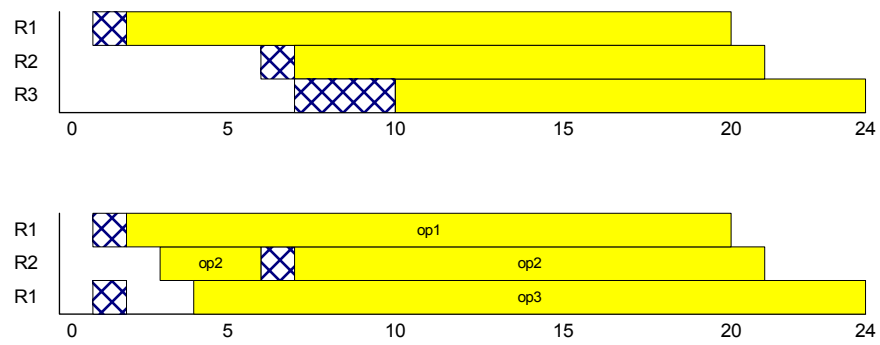
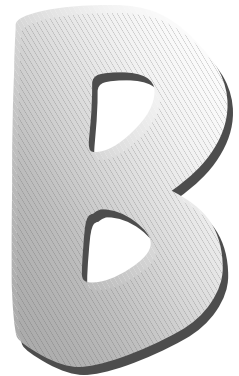


Figura A.11 – Resultado da experiência Fabricare nº 5

Este exemplo é uma combinação dos anteriores, ao usar recursos alternativos para duas operações, bem como utilizando um dos recursos para duas operações e com diferentes custos de execução ■

APÊNDICE B

MODELO MATEMÁTICO PARA ANÁLISE DA COMPLEXIDADE DE PRCPR



A análise de complexidade do Protocolo de Rede de Contrato com Propagação de Restrições efectuada na secção 6.2.4 “Análise da Complexidade do Protocolo” apresentou alguns dados em forma gráfica obtidos a partir de uma folha de cálculo com um modelo matemático descrito neste apêndice.

B.1 Simplificações

Para a análise de complexidade efectuaram-se algumas simplificações do problema, nomeadamente, considerou-se o pior caso, que consiste num plano sequencial com igual número de recursos alternativos para cada operação do plano (Figura B.1).

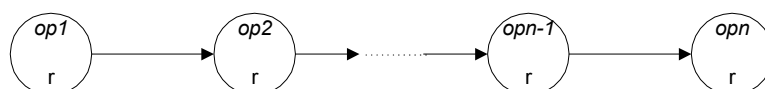


Figura B.1 – Plano exemplo utilizado para análise de complexidade

Como o plano de produção é sequencial, $\chi(\mathcal{P})$ corresponde ao produto dos vários recursos para cada operação. Como $R(i)$ é constante, isso é equivalente a uma potência.

$$\chi(\mathcal{P}) = \prod_{i=1}^n R(i) \Leftrightarrow r^n \quad (\text{B.1})$$

Por outro lado, também os termos com somatórios podem ser simplificados, pois o somatório de um termo constante é a multiplicação desse termo n vezes.

$$\sum_{i=1}^n R(i) \Leftrightarrow n \cdot r \quad (\text{B.2})$$

$$\sum_{i=1}^{n-1} (R(i) \times R(succ(i))) \Leftrightarrow (n - 1) \times r^2 \quad (\text{B.3})$$

O número de mensagens trocadas nas fase de influência pode ser simplificado de acordo com a seguinte equação:

$$\psi(\mathcal{P}) = \sum_{i=1}^{n-1} (R(i) \times \chi(trunc(\mathcal{P}, i)) \times R(succ(i))) \Leftrightarrow \sum_{i=1}^n S(i) \quad (\text{B.4})$$

$$S(i) = \begin{cases} 0 & n = 1 \\ r^n & n \neq 1 \end{cases} \quad (\text{B.5})$$

B.2 Folha de Cálculo

Para a geração dos gráficos da secção 6.2.4 utilizou-se então uma folha de cálculo (Figura B.2) cujas fórmulas utilizadas são apresentadas na Tabela B.1.

Tabela B.1 – Fórmulas da folha de cálculo

Termo		Células	Fórmula Excel
$\chi(\mathcal{P})$	1	B10...	= POWER(\$B\$1; \$A10)
$S(n)$	2	C10	0
	3	C11	= POWER(\$B\$1; 2)
	4	C12...	= C11 * \$B\$1
$\psi(\mathcal{P})$	5	D10	0
	6	D11...	= SUM(\$C\$10:C11)

Termo		Células	Fórmula Excel
$\psi^2(\mathcal{P})$	7	E10...	= (A10 - 1) * \$B\$1 * \$B\$1
N.º req.	8	F10	= \$A10 * \$B\$1
M_{prcpr}	9	G10...	= F10 + 2 * D10 + 2 * (D10 + B10)
M_{prcpr}^2	10	H10...	= 3 * F10 + 2 * E10
T_{inf}	11	I10	0
	12	I11...	= \$B\$3 + \$B\$4 + K11
T_{bid}	13	J10...	= \$B\$3 + \$B\$6
$Cam(n)$	14	K10	0
	15	K11...	= 2 * (A11 - 1)
$T_{prcpr inf+bid }$	16	L10...	= 2 * D10 * I10 + (D10 + B10) * J10
$T_{prcpr inf+bid }^2$	17	M10...	= 2 * tam_inf(\$B\$1; A10) + tam_bid(\$B\$1; A10)
delta abs.	18	N10...	= L10 - M10
delta rel.	19	O10...	= M10 / L10

As fórmulas da Tabela B.1 fazem uso de duas funções auxiliares definidas pelos seguintes programas em “Visual Basic for Applications” no caso de uma folha de cálculo Microsoft Excel.

```
Function t_inf(i As Integer, n As Integer)
    'Cab + x(P/i) * (LdI+Cam(n))
    If i = 1 Then
        x = 1
    Else
        x = Sheet1.Cells(8 + i, 2)
    End If
    t_inf = Sheet1.Cells(3, 2) + x * (Sheet1.Cells(4, 2) +
    Sheet1.Cells(9 + n, 11))
End Function
```

```
Function tam_inf(r As Integer, i As Integer) As Double
    Dim j As Integer
    Dim s As Double

    s = 0
    For j = 1 To i - 1
        s = s + r * r * t_inf(j, i)
    Next
    tam_inf = s
End Function
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	R(l)	2													
2															
3	Cab	2													
4	Ldl	10													
5	Cam(n)	2*(i-1)													
6	LdP	6													
7															
8															
9	Nº Op.	x(P)	s(n)	y(P)	y2(P)	nº req.	Mprcpr	M(2)prcpr	Tinf	Tbid	Cam	Tprcptinf-bidl	T(2)prcptinf-bidl	delta absoluto	delta relativo
10	1	2	0	0	0	2	6	6	0	8	0	16	16	0	1,0000
11	2	4	4	4	4	4	28	20	14	8	2	176	168	8	0,9545
12	3	8	8	12	8	6	70	34	16	8	4	544	500	44	0,9191
13	4	16	16	28	12	8	152	48	18	8	6	1360	1224	136	0,9000
14	5	32	32	60	16	10	314	62	20	8	8	3136	2796	340	0,8916
15	6	64	64	124	20	12	636	76	22	8	10	6960	6192	768	0,8897
16	7	128	128	252	24	14	1278	90	24	8	12	15136	13492	1644	0,8914
17	8	256	256	508	28	16	2560	104	26	8	14	32528	29112	3416	0,8950
18	9	512	512	1020	32	18	5122	118	28	8	16	69376	62396	6980	0,8994
19	10	1024	1024	2044	36	20	10244	132	30	8	18	147184	133056	14128	0,9040
20	11	2048	2048	4092	40	22	20486	146	32	8	20	311008	282564	28444	0,9085
21	12	4096	4096	8188	44	24	40968	160	34	8	22	655056	597960	57096	0,9128
22	13	8192	8192	16380	48	26	81930	174	36	8	24	1375936	1261516	114420	0,9168
23	14	16384	16384	32764	52	28	163852	188	38	8	26	2883248	2654160	229088	0,9205
24	15	32768	32768	65532	56	30	327694	202	40	8	28	6028960	5570516	458444	0,9240
25	16	65536	65536	131068	60	32	655376	216	42	8	30	12582544	11665368	917176	0,9271
26	17	131072	131072	262140	64	34	1310738	230	44	8	32	26214016	24379356	1834660	0,9300
27	18	262144	262144	524284	68	36	2621460	244	46	8	34	54525552	50855904	3669648	0,9327
28	19	524288	524288	1048572	72	38	5242902	258	48	8	36	113245792	105906148	7339644	0,9352
29	20	1048576	1048576	2097148	76	40	10485784	272	50	8	38	234880592	220200936	14679656	0,9375

Figura B.2 – Folha de cálculo utilizada para análise de complexidade do PRCPR

```

Function t_bid(i As Integer, n As Integer)
    If n = 1 Then
        x = 1
    ElseIf i = n Then
        'x(P/i)
        x = Sheet1.Cells(8 + i, 2)
    Else
        'x(P)
        x = Sheet1.Cells(9 + i, 2)
    End If
    t_bid = Sheet1.Cells(3, 2) + x * Sheet1.Cells(6, 2)
End Function

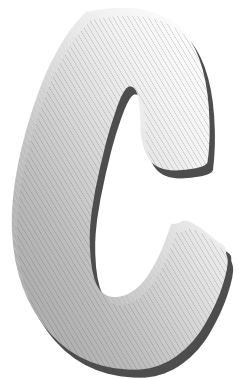
Function tam_bid(r As Integer, i As Integer) As Double
    Dim j As Integer
    Dim s As Double

    s = 0
    For j = 1 To i
        s = s + r * t_bid(j, i)
    Next
    tam_bid = s
End Function

```

■

APÊNDICE C SÍTIOS WWW



Neste apêndice são apresentados *web sites* de projectos, pessoas ou instituições relacionados com os temas abordados na dissertação. À frente do nome de cada *site* aparecem indicações sobre os temas de interesse nesse *site*, de acordo com a legenda apresentada na tabela seguinte:

Tabela C.1 – Legenda utilizada nos web sites

Ícone	Significado	Ícone	Significado
P	Produção	▲	Agentes & Sistemas Multiagente
V	Visionários na Produção	H	Sistemas Holónicos
L	Programação em Lógica	C	Interacção
E	Programação em Lógica Estendida	R	Raciocínio & Conhecimento
I	Inteligência Artificial		

C.1 Pessoas

- ▲** Agre, Phil
<http://communication.ucsd.edu/pagre/agre.html>

- HP** Bongaerts, Luc
<http://www.mech.kuleuven.ac.be/~lbongaer/>
- Δ** Castelfranchi, Cristiano
<http://pscs2.irmkant.rm.cnr.it/users/cristiano/home.html>
- ΔL** Coelho, Helder
<http://www.di.fc.ul.pt/~hcoelho/>
- Δ** Cohen, Paul
<http://eksl-www.cs.umass.edu/~cohen/home.html>
- Δ** Cohen, Phil
<http://www.cse.ogi.edu/~pcohen/>
- Δ** d'Inverno, Mark
<http://www.wmin.ac.uk/~dinverm/>
- Δ** Durfee, Edmund H.
<http://ai.eecs.umich.edu/people/durfee/durfee.html>
- ΔC** Finn, Tim
<http://www.cs.umbc.edu/~finin/>
- Δ** Fisher, Michael
<http://www.doc.mmu.ac.uk/STAFF/michael/Michael-Fisher.html>
- Δ** Franklin, Stan
<http://www.msci.memphis.edu/~franklin/index.html>
- Δ** Georgeff, Michael P.
<http://www.aaii.oz.au/>
- I** Huhns, Michael N.
<http://www.mcc.com/projects/carnot/michael-huhns.html>
- Δ** Ingrand, Félix
<http://www.laas.fr/~felix/>
- ΔL** Jennings, Nick
<http://www.elec.qmw.ac.uk/dai/>
- Δ** Luck, Michael
<http://www.dcs.warwick.ac.uk/~mikeluck/>
- Δ** Maes, Pattie
<http://pattie.www.media.mit.edu/people/pattie/>
- P** McFarlane, Duncan
<http://www-mmd.eng.cam.ac.uk/people/dcm/dcm.htm>
- Δ** Mueller, Joerg P.
<http://www.dfki.uni-sb.de/~jpm/>

- I ▲** Neves, José
<http://venus.di-ia.uminho.pt/~jneves/>
- ▲ P V** Parunak, H. Van Dyke
<http://www.iti.org/~van/>
- ▲** Petrie, Charles J.
<http://cdr.stanford.edu/html/people/petrie/home.html>
- ▲** Sloman, Aaron
<http://www.cs.bham.ac.uk/~axs/>
- ▲ H** Sousa, Paulo
<http://www.dei.isep.ipp.pt/~psousa>
- ▲** Sycara, Katia
<http://www-cgi.cs.cmu.edu/afs/cs.cmu.edu/user/katia/www/katia-home.html>
- ▲** Tambe, Milind
<http://www.isi.edu/soar/tambe>
- ▲** Traum, David R.
<http://tecfa.unige.ch/tecfa/general/tecfa-people/traum.html>
- P H** Valckenaers, Paul
<http://www.mech.kuleuven.ac.be/pma/people/pages/pvalcken.htm>
- ▲** Wooldridge, Michael
<http://www.doc.mmu.ac.uk/STAFF/mikew.html>
- P H** Wyns, Jo
<http://www.mech.kuleuven.ac.be/~jwyns/welcome.html>

C.2 Instituições e Grupos I&D

- ▲** ABE Research Group
<http://cdr.stanford.edu/ABE/>
- ▲** Agent Society Home Page
<http://www.agent.org/>
- ▲** Agents Group @ MIT Media Lab
<http://agents.www.media.mit.edu/groups/agents/>
- I** American Association for Artificial Intelligence
<http://www.aaai.org/>
- R** ARPA Knowledge Sharing Effort public library
<http://www-ksl.stanford.edu/knowledge-sharing/index.html>

- I** Artificial Intelligence @ SRI
<http://www.ai.sri.com/aic/>
- I** Australian Artificial Intelligence Institute
<http://www.aaii.oz.au/>
- A** Autonomous Agents Research Group @ CWRU
<http://yuggoth.ces.cwru.edu/>
- A** DAI @ Hebrew University, Israel
<http://www.cs.huji.ac.il/labs/dai/>
- A** DAI Unit @ QMW, London
<http://www.elec.qmw.ac.uk/dai/>
- A** Distributed Intelligent Agents Group @ Umich
<http://ai.eecs.umich.edu/diag/homepage.html>
- Enterprise Integration Laboratory @ Uni. of Toronto
<http://www.ie.utoronto.ca/EIL/eil.html>
- Enterprise Integration Technologies
<http://www.eit.com/creations/research/>
- A** FIPA - Foundation for Intelligent Physical Agents
<http://www.fipa.org/>
- P H** HMS @ K.U. Leuven
<http://www.mech.kuleuven.ac.be/pma/pma.html>
- P V** IMS @ CSIRO
<http://www.mlb.dmt.csiro.au/pages/MMS.html>
- P V** IMS @ U: Calgary
<http://ksi.cpsc.ucalgary.ca/IMS/IMS.html>
- I** Inteligência Artificial @ Universidade do Minho
<http://venus.di-ia.uminho.pt/>
- P V** Intelligent Manufacturing Systems (.org)
<http://www.ims.org/>
- Intelligent System Applications Research Group
<http://gryphon.elec.qmw.ac.uk/isag/>
- A** Internet Agents Group
<http://www.olemiss.edu/~usul/AI/iagents.html>
- R** Knowledge Systems Laboratory @ Stanford University
<http://www-ksl.stanford.edu/>
- L** Logic Group @ Stanford University
<http://logic.stanford.edu/>

- MAGMA Group @ LIFIA/IMAG
<http://cosmos.imag.fr/MAGMA/home-fr.html>
- ▲ MIT Media Lab, Software Agents Group
<http://lcs.www.media.mit.edu/groups/agents/>
- ▲ Multi-Agent Systems Group @ University of Maastricht
<http://www.cs.rulimburg.nl/~vreeswyk/mas.htm>
- RMIT
<http://www.gse.rmit.edu.au/~rsedc/holon.html>
- ▲ Sandia Intelligent Agents for Manufacturing
<http://nittany.ca.sandia.gov:8001/>
- ▲ The Multi-Agent Systems Laboratory
<http://dis.cs.umass.edu/dis.html>
- ▲ UMBC Agent Web
<http://www.cs.umbc.edu/agents/>

C.3 Projectos

- ▲ AARIA
<http://www.aaria.uc.edu/>
- ▲ Agent Building Shell @ Uni. of Toronto
<http://www.ie.utoronto.ca/EIL/ABS-page/ABS-intro.html>
- ▲ Agent Collaboration Language Project
<http://raven.cecer.army.mil/acl/welcome.html>
- ▲ Agent-Tcl
<http://minsky.med.virginia.edu/sdm7g/Projects/Python/safe-tcl/agent-tcl.txt>
- ▲ Cognitive/Agent Architecture: Title Page
<http://ai.eecs.umich.edu/cogarch0/index.html>
- Esperit IMS-WG
<http://www.mech.kuleuven.ac.be/pma/project/imswg/welcome.html>
- ▲ Fabricare
<http://www.dei.isep.ipp.pt/~psousa/fabricare/>
- GOA
<http://www.mech.kuleuven.ac.be/pma/project/goa/holonic.html>
- HMS
<http://hms.ifw.uni-hannover.de/>

- P H** HMS @ ISEP/IPP
<http://www.dei.isep.ipp.pt/~psousa/fabricare/>
- H** IMS
<http://www.ims.org>
- R** KIF: The Knowledge Interchange Format
<http://logic.stanford.edu/kif/kif.html>
- C** KQML: The Knowledge Query and Manipulation Language
<http://www.cs.umbc.edu/kqml/>
- MADEsmart
<http://www.bbtech.com/MADE/index.html>
- P A** MASCADA
<http://www.mech.kuleuven.ac.be/pma/project/mascada/welcome.html>
- P V** Next Generation Manufacturing (NGM)
<http://imtr.ornl.gov/NGM/ngmhome.html>
- R** Ontolingua: A Tool for Developing Ontologies
<http://www-ksl.stanford.edu/knowledge-sharing/ontolingua/ontolingua.html>
- OZ project @ CMU
<http://www.cs.cmu.edu:8001/afs/cs.cmu.edu/project/oz/web/oz.html>
- A** Reactive Agents Project @ Teleos Research
<http://teleos.com/projects/agents/overview.html>
- SOAR Project @ ISI
<http://www.isi.edu/soar/soar.html>

C.4 Miscelânea

- L** Bulletin of the Logic SIG (IGPL)
<http://www.mpi-sb.mpg.de:80/igpl/Bulletin/>
- Distributed Computation
<http://www.base.com/gordon/web/distribution.html>
- Formal Aspects of Computing Journal (FACS)
<http://www.cs.man.ac.uk/fmethods/facj/index.html>
- A** HTML archive of the agents mailing list
<http://www.sml.com/research/tcl/lists/agents-list.html>
- A** Intelligent Software Agents
<http://www.sics.se/isl/abc/survey.html>

- Journal of AI Research (JAIR)
<http://www.cs.washington.edu/research/jair/home.html>
- Ontologies
<http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>
- ▲ Ralph Becket's intelligent software agents page
<http://www.cl.cam.ac.uk/users/rwab1/agents.html>
- SICStus PROLOG
<http://www.sics.se/isl/sicstus.html>
- ▲ The Software Agents Mailing List FAQ
http://www.ee.mcgill.ca:80/~belmarc/agent_faq.html
- Visual Basic
<http://msdn.microsoft.com/vbasic/>
-

Índice Remissivo

A

AARIA102, 107, 108, 219, 220, 221, 222, 223, 295
Adaptabilidade5, 42, 62, 66, 67, 76, 82, 88, 89, 124, 227, 228, 229, 232, 239
Agente7, 8, 9, 12, 16, 72, 84, 85, 86, 87, 88, 90, 91, 92, 93, 94, 95, 99, 101, 102, 103, 106, 108, 109, 120, 123, 158, 179, 219, 220, 222, 223, 229, 230, 261, 267
Agente Inteligente87
Agente Móvel87
Agilidade4, 15, 59, 65, 67, 68, 69, 72, 81, 82, 124, 228, 229, 232, 239
Arthur Koestler74
Atributos holónicos80
Autonomia8, 16, 66, 73, 74, 76, 77, 80, 82, 85, 87, 89, 91, 95, 109, 113, 114, 123, 228, 229, 239

B

Base de Conhecimento9, 10, 12, 13, 14, 16, 88, 97, 98, 99, 100, 101, 108, 112, 120, 121, 122, 124, 126, 127, 129, 130, 131, 132, 133, 134, 136, 137, 139, 140, 141, 142, 143, 145, 146, 147, 148, 149, 150, 151, 152, 153, 155, 180, 182, 183, 184, 208, 213, 216, 222, 230, 231, 233, 234, 237
blackboard.....210, 217, 218, 232

C

Ciclo de vida16, 38, 48, 50, 57, 84, 112, 118, 119, 124, 125, 128, 130, 132, 133, 135, 136, 152, 155, 178, 208, 226, 239
Comando85, 91, 211, 221
Competição46, 48, 52, 56, 57, 58, 64, 79, 91, 92, 93, 108, 226
Competitividade3, 4, 15, 43, 48, 50, 51, 60, 239
Complexidade1, 2, 6, 7, 8, 13, 17, 32, 48, 50, 64, 76, 89, 95, 96, 155, 169, 171, 172, 173, 176, 207, 224, 226, 236, 239, 240, 242, 285, 288
Componente7, 20, 21, 22, 23, 25, 28, 32, 34, 35, 36, 37, 39, 49, 58, 63, 68, 72, 75, 76, 78, 80, 82, 102, 104, 112, 115, 118, 125, 127, 128, 131, 132, 133, 134, 148, 159, 173, 185, 197, 203, 204, 209, 213, 221, 233, 240, 245
Computação Distribuída.....6, 52

Controlo5, 21, 27, 31, 33, 34, 35, 36, 38, 39, 41, 43, 50, 52, 58, 61, 62, 66, 77, 78, 80, 81, 85, 87, 89, 91, 93, 95, 102, 103, 104, 106, 107, 118, 127, 144, 219, 220, 221, 223, 227, 228, 238, 243
Controlo da Produção33, 38, 39
Controlo da Qualidade.....15, 43
Controlo da Qualidade Assistido por Computador41
Controlo de Qualidade.....36
Cooperação6, 8, 16, 58, 59, 73, 77, 78, 80, 81, 82, 84, 85, 90, 91, 92, 93, 103, 108, 109, 122, 222, 229, 232, 234, 235, 243
Coopetição.....58
Coordenação6, 13, 37, 38, 40, 49, 84, 91, 92, 93, 95, 159, 210, 217, 222, 232, 235, 236

D

Descentralização9, 66, 82, 85, 94, 227, 229, 239
Desenho Assistido por Computador15, 19, 34, 40, 41, 43, 118
Dinamismo7, 15, 53, 67, 69, 82, 94, 124, 228, 229, 232, 239
Distribuição3, 14, 16, 23, 27, 32, 40, 52, 56, 65, 66, 72, 82, 84, 85, 89, 90, 91, 109, 222, 223, 227, 229, 237, 239

E

Empresa Estendida57, 58
Empresa Virtual.....58, 106, 119, 158, 245
Engenharia Assistida por Computador40
Escalonamento7, 9, 11, 12, 13, 15, 16, 17, 27, 32, 36, 39, 43, 82, 95, 102, 103, 104, 105, 106, 107, 108, 115, 118, 121, 123, 127, 128, 129, 138, 152, 155, 156, 157, 158, 160, 161, 163, 164, 167, 168, 169, 178, 184, 187, 189, 193, 194, 195, 198, 205, 206, 207, 212, 218, 219, 220, 221, 223, 224, 230, 231, 232, 235, 236, 237, 240, 241, 242, 243, 244, 271, 272, 274, 275, 277

F

Fábrica Fractal16, 73, 83, 84, 85, 109
*Fabricare*2, 12, 14, 17, 156, 207, 208, 219, 220, 221, 222, 223, 224, 225, 230, 237, 272, 273, 274, 275, 276, 277, 281, 282, 295

Flexibilidade 6, 15, 22, 24, 26, 37, 38, 42, 59, 63, 65, 67, 69, 71, 73, 76, 82, 85, 86, 95, 206, 207, 228, 229, 236, 239, 241

Flow shop 24

G

Gestão da Produção 15, 27, 43

Globalização ..1, 2, 3, 46, 47, 51, 52, 56, 226

Gou & Lu 107, 219, 220, 221, 222, 223

H

Henry Ford 49

Herbert Simon 74, 75

Hierarquia 34, 61, 62, 75, 77, 84, 106, 122, 261

Hierárquico 62

HMS 107, 219, 220, 221, 222, 223

Holarquia 8, 10, 16, 77, 78, 79, 80, 82, 104, 105, 107, 108, 112, 117, 118, 119, 121, 122, 123, 124, 130, 131, 141, 142, 152, 179, 180, 189, 198, 219, 222, 228, 229, 232, 235, 239, 240, 243, 245

Holon 8, 10, 11, 12, 13, 16, 17, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 85, 94, 95, 103, 104, 105, 106, 108, 112, 115, 117, 118, 119, 120, 121, 122, 123, 124, 125, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 150, 151, 152, 153, 155, 158, 160, 162, 163, 174, 178, 179, 180, 181, 182, 183, 185, 187, 189, 191, 192, 194, 195, 196, 198, 201, 202, 203, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 217, 218, 219, 220, 221, 222, 223, 224, 228, 229, 230, 231, 232, 233, 234, 235, 236, 238, 239, 240, 241, 243, 244, 245, 295

Holon de Cliente 117, 118, 136, 137, 151

Holon de Compra 117, 118, 131, 132, 136, 146

Holon de Escalonamento 17, 117, 118, 128, 159, 167, 168, 185, 186, 187, 189, 210, 213, 221, 224, 238

Holon de Fornecedor 117, 118, 133, 134, 137, 147

Holon de Gestão de Compras 117, 118, 148, 149, 152

Holon de Gestão de Vendas 117, 118, 151, 152

Holon de Planeamento da Produção 117, 118

Holon de Planeamento de Processos 117, 118, 129, 159, 160, 190, 194, 210, 213, 217, 238

Holon de Produto 17, 117, 118, 125, 126, 143, 183, 184, 210, 216, 217, 222, 224, 238

Holon de Projecto 117, 118

Holon de Recurso 12, 13, 17, 117, 118, 119, 123, 129, 130, 145, 158, 161, 162, 163, 164, 165, 168, 169, 171, 174, 189, 191, 192, 193, 195, 196, 197, 198, 199, 201, 203, 208, 209, 212, 213, 215, 221, 222, 224, 232, 234, 238, 242, 243

Holon de Serviços de Directório 142, 143, 181, 182, 209, 210, 213, 215, 231, 232, 234, 238

Holon de Tarefa 12, 13, 17, 117, 118, 119, 127, 128, 129, 130, 142, 144, 155, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 171, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 198, 199, 201, 208, 209, 210, 211, 213, 219, 221, 222, 224, 232, 234, 238, 242, 243, 244, 245

Holon de Venda 117, 118, 134, 135, 136, 149, 150

I

Informação Incompleta 9, 10, 13, 17, 67, 73, 95, 98, 99, 107, 108, 112, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 151, 153, 217, 218, 219, 222, 224, 228, 230, 231, 233, 234, 243

Informação Mutuamente Exclusiva ... 99, 138

Informação Negativa Explícita 99, 230

Inspeção 15, 34, 36, 43

Inteligência Artificial 6, 7, 73, 74, 86, 88, 89, 294

Inteligência Artificial Distribuída 6, 7, 8, 89, 90

Internet 2, 3, 5, 18, 50, 52, 55, 61, 72, 248, 263, 294

J

Job shop 24

Just-in-Time 193

L

Lançador de Tarefas 128, 159, 160, 208, 209, 213, 215

Linguagem Lógica 9, 96, 97, 100, 101

Lógica de Predicados 96

Lógica Predicativa 96

Lógica Proposicional 96

M

Meta-interpretador 13, 100, 101, 140, 230, 233

Mundo Aberto . 141, 142, 143, 144, 148, 151

N

Negação Explícita 16, 98, 99, 100, 109, 141, 142

Negociação 8, 9, 10, 17, 81, 85, 91, 93, 98, 108, 128, 155, 158, 166, 167, 168, 186, 187, 189, 190, 194, 198, 199, 200, 205, 221, 222, 224, 230, 241, 244

Nulo do Tipo Desconhecido 13, 138, 139, 141, 143, 144, 145, 147, 148, 150, 151, 233

Nulo do Tipo Desconhecido de um Conjunto de Valores 139, 141, 150

Nulo do Tipo Não Permitido 21, 32, 141, 148, 152, 169, 253

P

Personalização em Massa 3, 25

Planeamento da Produção 15, 16, 31, 32, 38, 43, 106, 152

Planeamento de Necessidades de Capacidade 31, 39

Planeamento de Necessidades de Material 31, 39

Planeamento de Processos Assistido por Computador 40

Plano de Necessidades de Capacidade 32

Plano de Necessidades de Material 31, 32

Plano Mestre de Produção 31

Prazo de entrega 23, 26, 27, 240

Pressuposto do Domínio Fechado 97, 98, 229

Pressuposto do Mundo Fechado 97, 98, 99, 101, 138, 141, 142, 143, 144, 145, 146, 147, 149, 150, 229

Proactividade 88

Problema de Indecisão 17, 155, 166, 185, 224, 235

Produção Assistida por Computador 15, 34, 40, 41, 43

Produção Biónica 16, 84, 85, 109

Produção em Massa 24, 48

Produção Integrada por Computador 5, 7, 15, 19, 34, 35, 36, 40, 41, 42, 43, 61, 62, 66, 84, 85, 86, 95, 109, 111, 226, 227

Produção por Lote 24

Produção por Projecto 24, 25

Produto 1, 2, 3, 5, 7, 11, 15, 16, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 33, 34, 35, 36, 37, 38, 39, 40, 41, 43, 46, 47, 48, 49, 50, 51, 54, 55, 56, 57, 58, 59, 61, 62, 63, 64, 65, 66, 67, 68, 72, 75, 79, 80, 81, 82, 84, 104, 105, 106, 107, 112, 113, 114, 115, 118, 119, 121, 123, 125, 126, 127, 129, 131, 133, 135, 137, 138, 143, 144,

148, 150, 151, 152, 156, 157, 158, 159, 160, 161, 164, 183, 185, 187, 190, 194, 205, 206, 208, 209, 210, 212, 213, 214, 216, 219, 220, 221, 223, 226, 227, 228, 230, 232, 233, 238, 239, 240, 245, 275, 277, 286

Programação em Lógica 2, 9, 12, 13, 16, 97, 98, 99, 101, 109, 138, 217, 229, 230, 233, 240, 242

Programação em Lógica Estendida 2, 9, 12, 13, 14, 16, 72, 86, 95, 100, 101, 109, 111, 228, 229, 230, 233, 240

PROSA 104, 107, 108, 219, 220, 221, 222, 223

Protocolo de Rede de Contrato 13, 91, 93, 94, 106, 108, 120, 158, 159, 222, 223, 230, 234, 235

Protocolo de Rede de Contrato com Propagação de Restrições 13, 17, 94, 158, 171, 172, 173, 174, 175, 176, 177, 178, 189, 191, 199, 211, 234, 235, 236, 285

Q

Qualidade 2, 3, 27, 33, 34, 35, 36, 38, 39, 41, 47, 48, 50, 51, 52, 55, 61, 64, 72, 133, 135, 226, 245

R

Reactividade 15, 35, 52, 53, 56, 63, 64, 65, 66, 67, 69, 82, 88, 228, 229, 239

Representação de Conhecimento 9, 14, 96, 98

S

Semi-acabado 22, 27

SICSus Prolog 208, 210, 217, 218, 238, 245, 297

Sistema de Produção 7, 15, 19, 20, 21, 22, 23, 24, 26, 27, 28, 49, 65, 68, 80, 84, 104, 111, 112, 115, 116, 120, 178

Sistema de Produção Baseado em Agentes 8, 84, 95, 102, 106, 108

Sistema Flexível de Fabrico 15, 19, 37, 38, 42, 43, 62

Sistema Holónico de Produção 5, 8, 9, 12, 14, 16, 71, 73, 79, 80, 81, 82, 83, 84, 85, 86, 95, 102, 104, 106, 107, 109, 111, 219, 220, 221, 222, 223, 225, 228, 229, 231, 239, 254, 255, 294, 295, 296

Sistema Inteligente de Produção 5, 72, 73, 79, 103, 256, 294, 295, 296

Sistema Multiagente 2, 7, 8, 9, 12, 14, 16, 72, 85, 90, 91, 92, 93, 94, 95, 109, 111, 123, 228, 229, 230, 237, 240

V

Veracidade11, 88, 124, 180, 181

Visual Basic208, 217, 218, 238, 245, 287,
297

Vitalidade..... 4, 83, 239



W

WWW 55