

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.

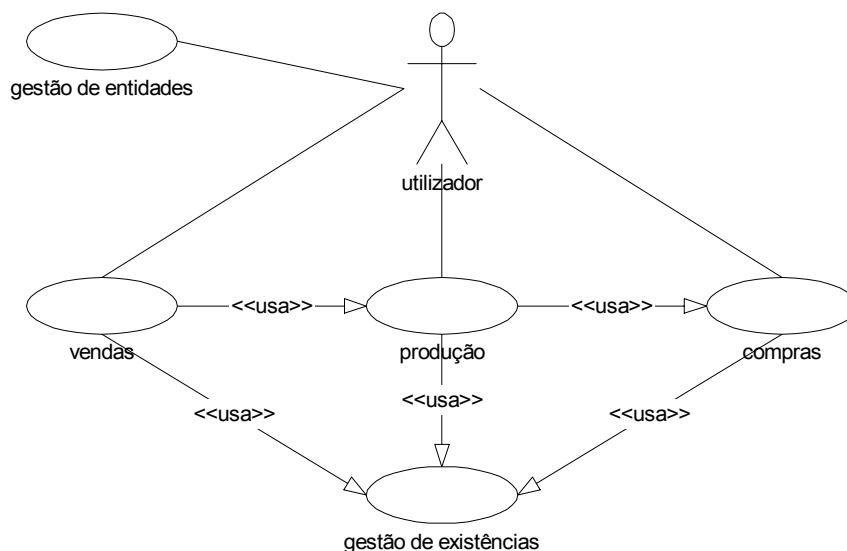


Figura 5.1 – Principais áreas funcionais do sistema

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.

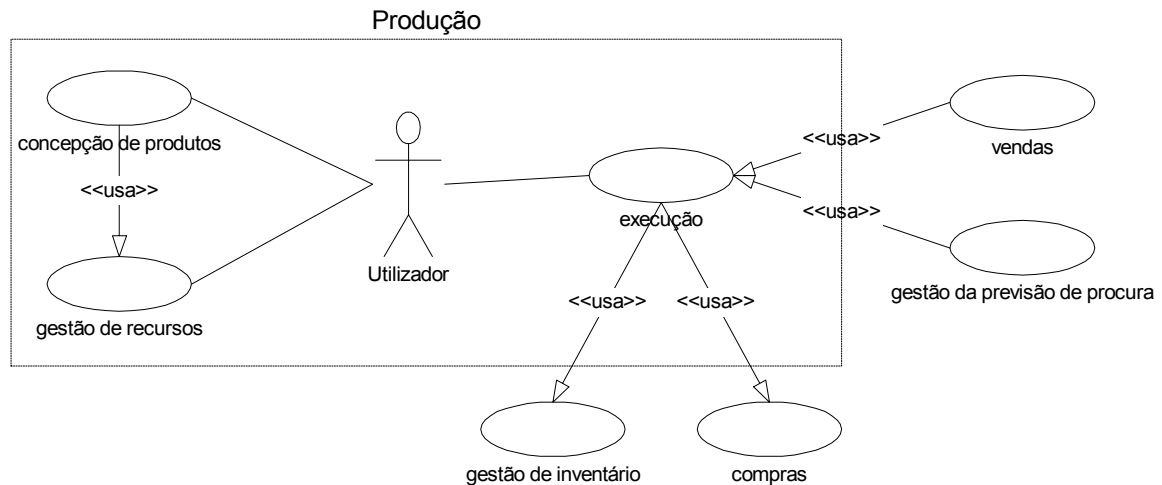


Figura 5.2 – Caso de utilização para a função ‘Produção’

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).

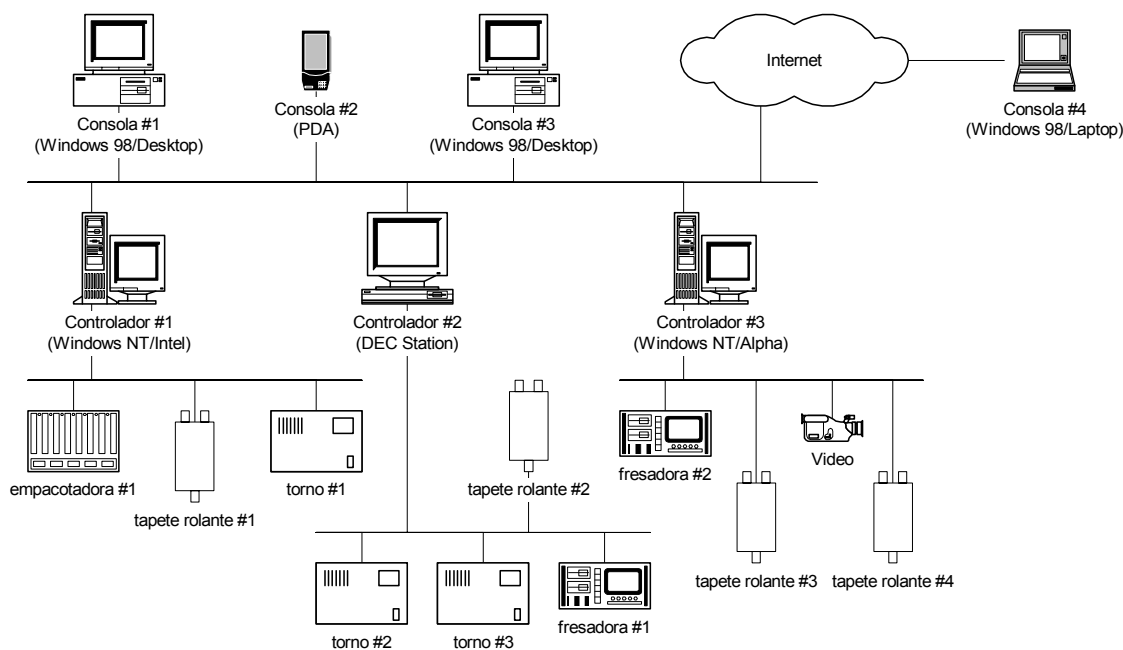


Figura 5.3 – Um possível cenário de exploração

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.

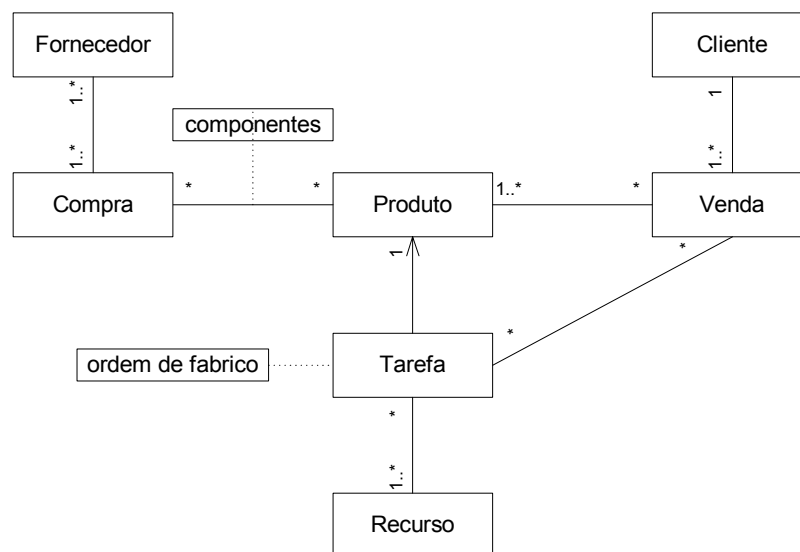


Figura 5.4 – Entidades intervenientes no sistema e seus relacionamentos

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).

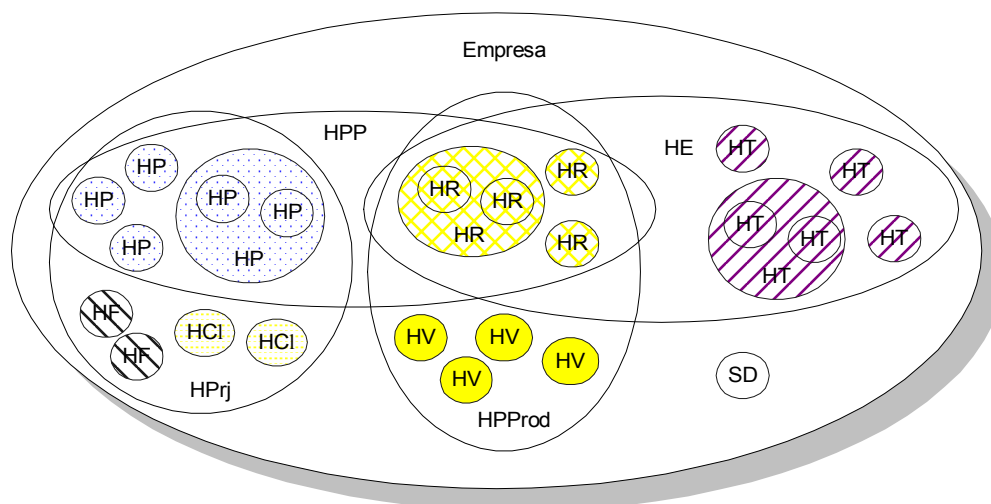


Figura 5.5 – Arquitectura holónica do sistema de fabrico

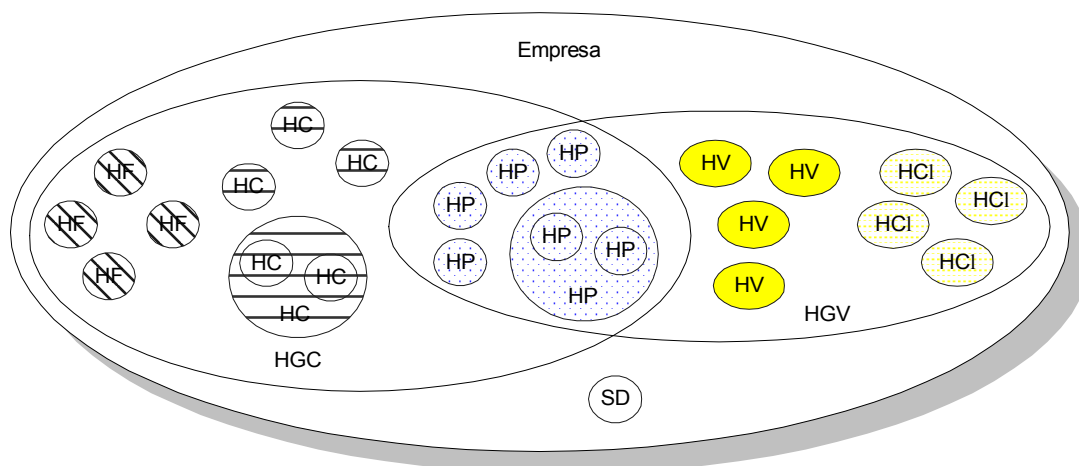


Figura 5.6 – Arquitectura holónica do sistema comercial

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.

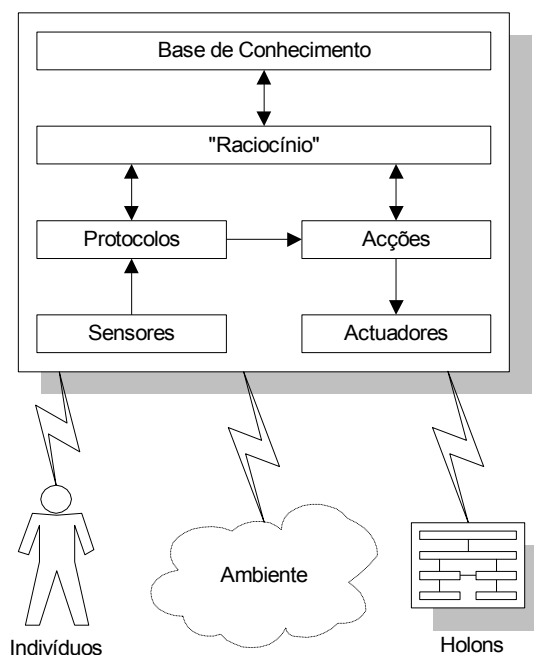


Figura 5.7 – Arquétipo de um agente

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.

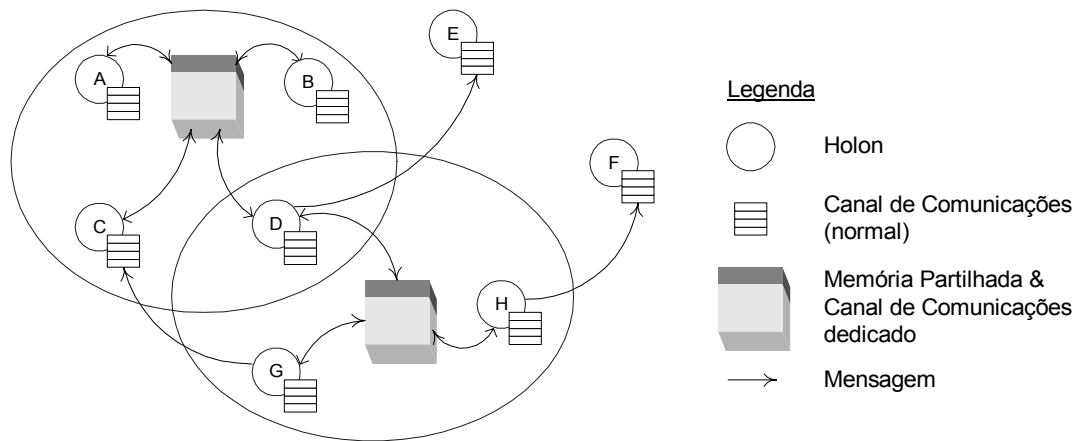


Figura 5.8 – Holarquias

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:

tipo(tipo_srv_dir)
fornece(Id-Holon, Id-Habilidade) ←
serviços(Id-Holon, L) ∧

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

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.

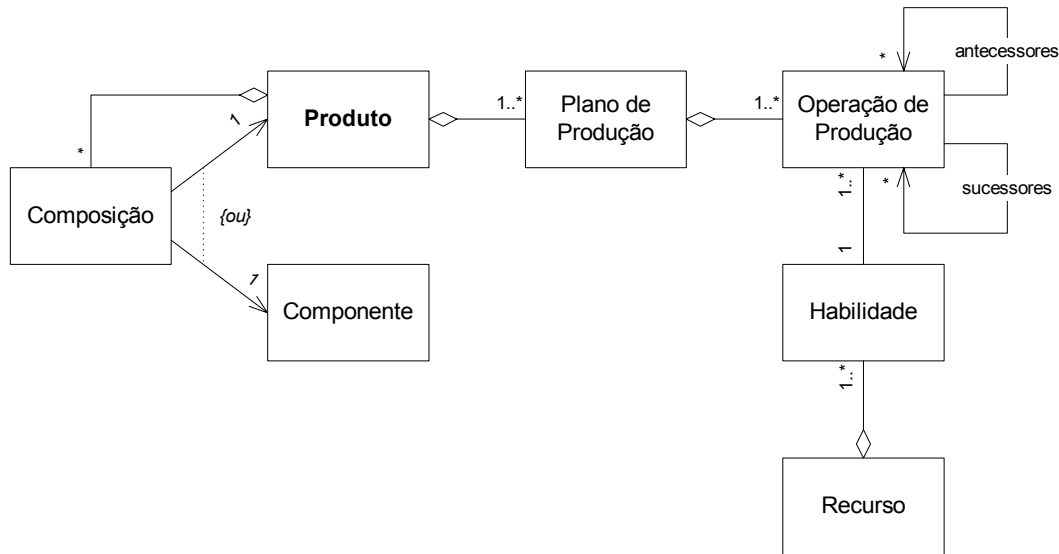


Figura 5.9 – Diagrama de classes para holons de produto

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.

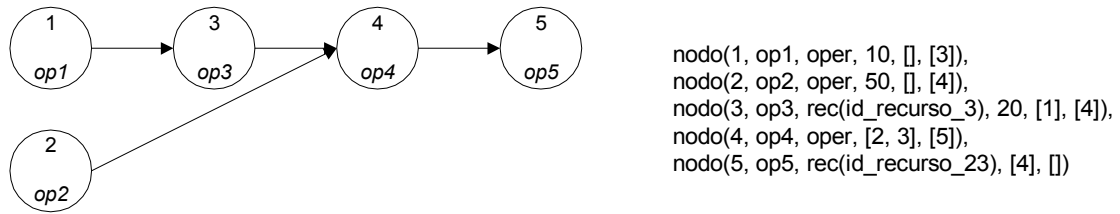


Figura 5.10 – Exemplo de representação de um plano

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.

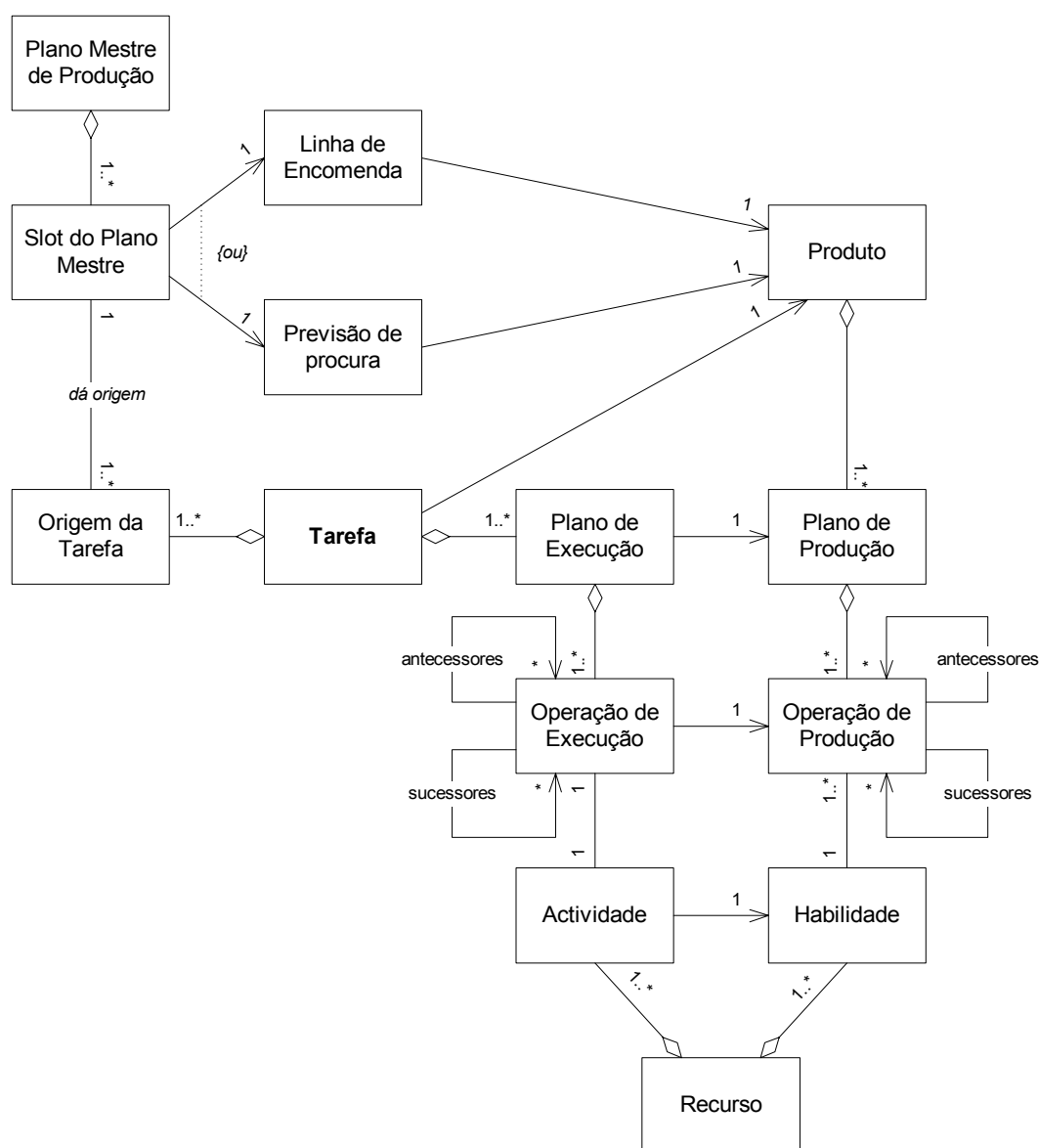


Figura 5.11 – Diagrama de classes para holons de tarefa

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 maquinaria 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.

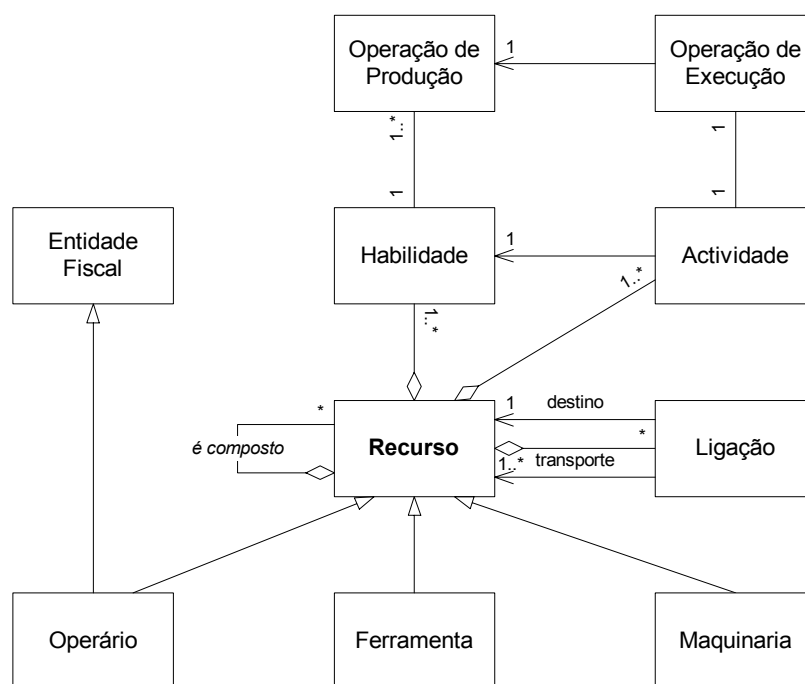


Figura 5.12 – Diagrama de classes para holons de recurso

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.

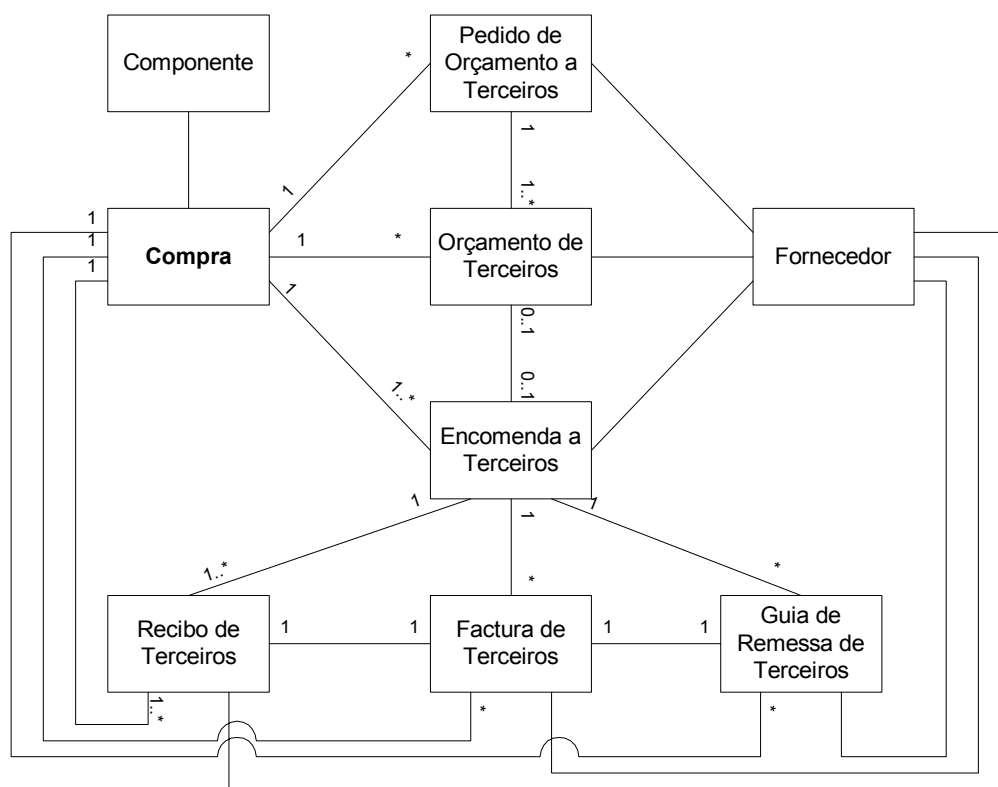


Figura 5.13 – Diagrama de classes para holons de compras

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.

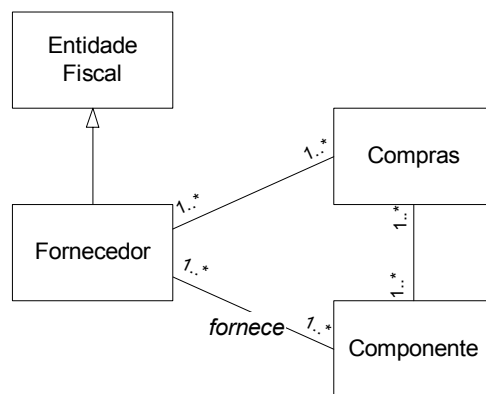


Figura 5.14 – Diagrama de classes para holons de fornecedor

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.

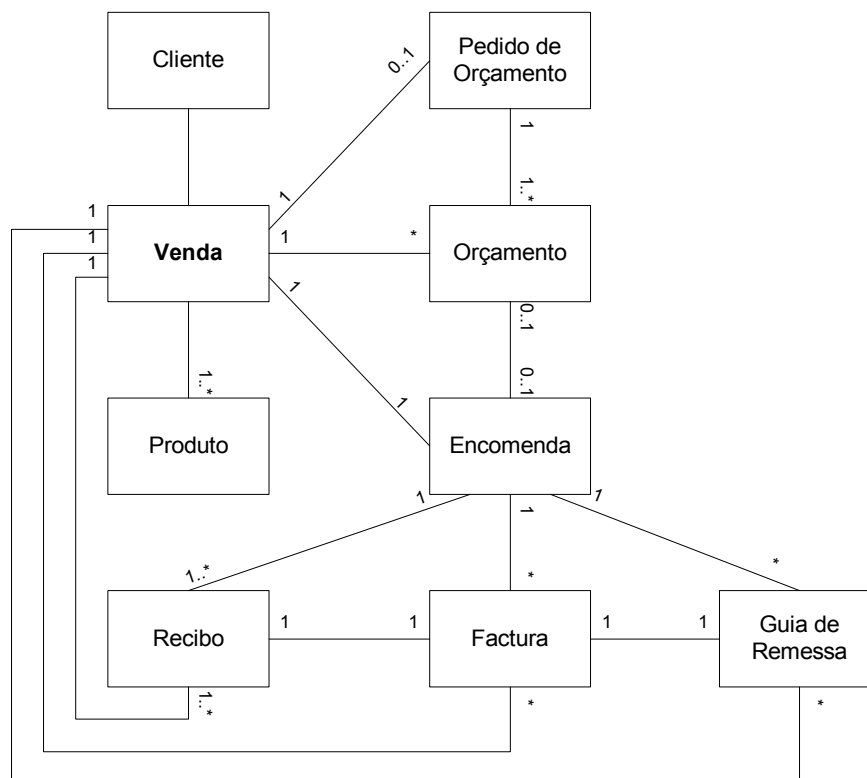


Figura 5.15 – Diagrama de classes para holons de vendas

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.

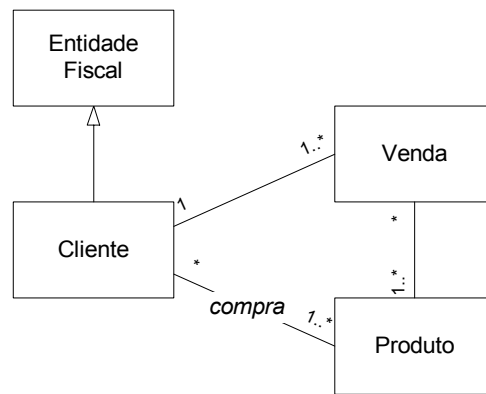


Figura 5.16 – Diagrama de classes para holons de cliente

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:

$$\neg nome(I, D) \leftarrow \\ \text{não } nome(I, D)$$

$$\begin{aligned}
 \text{nulo_n_permitido}(\text{nome}(_, _)) \leftarrow & \\
 & \text{todas_as_soluções}(\text{nome}(I, D), \text{nome}(I, D), L) \wedge \\
 & \text{comprimento}(L, N) \wedge \\
 & N > 1 \\
 \neg \text{tipo}(T) \leftarrow & \\
 & \text{não tipo}(T) \\
 \neg \text{relação}(H, R, V, P) \leftarrow & \\
 & \text{não relação}(H, R, V, P) \\
 \neg \text{holarquia}(H, L) \leftarrow & \\
 & \text{não holarquia}(H, L) \\
 \neg \text{todo}(H) \leftarrow & \\
 & \text{não todo}(H) \\
 \neg \text{parte_de}(H) \leftarrow & \\
 & \text{não parte_de}(H)
 \end{aligned} \tag{5.5}$$

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 ■

