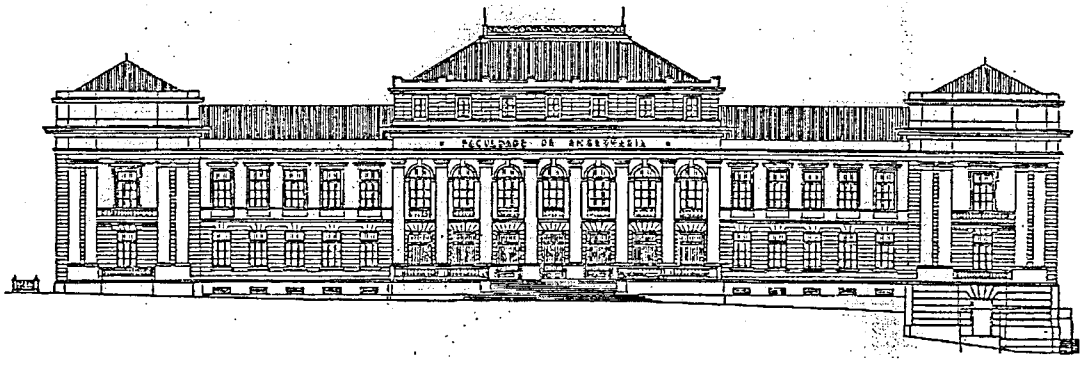
JOÃO JOSÉ DA CUNHA E SILVA PINTO FERREIRA

Suporte do Ciclo de Vida dos Sistemas Integrados de Fabrico através de Modelos Executáveis sobre Infra-estruturas de Integração



FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

1995

JOÃO JOSÉ DA CUNHA E SILVA PINTO FERREIRA

Suporte do Ciclo de Vida dos Sistemas Integrados de Fabrico através de Modelos Executáveis sobre Infra-estruturas de Integração

Dissertação apresentada à Faculdade de Engenharia da Universidade do Porto para obtenção do grau de Doutor em Engenharia Electrotécnica e de Computadores e realizada sob orientação científica do Prof. Doutor José Manuel de Araújo Baptista Mendonça, Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES
FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
1995

O trabalho de investigação apresentado nesta dissertação foi apoiado pelo programa PRODEP, medida 4.2, acção de formação 12.19.5

Aos Meus Pais

Resumo

Nesta dissertação são tratados a concepção, o desenvolvimento e a utilização de modelos executáveis no âmbito da Integração de Sistemas de Fabrico ou Fabrico Integrado por Computador (CIM), área esta na qual têm sido realizados múltiplos esforços em vários domínios da tecnologia e das ciências humanas. O trabalho realizado situou-se numa zona de fronteira abordando as arquitecturas abertas para o CIM, as infra-estruturas de integração, e particularmente a modelação e controlo do sistema integrado de fabrico através de modelos executáveis.

A multidisciplinaridade existente no ambiente empresarial e fabril deverá reflectir-se na utilização de modelos de vários tipos que, uma vez integrados, retratem a realidade da forma mais adequada ao apoio à decisão. As facilidades de modelação devem cobrir não só o sistema físico de fabrico, i.e. máquinas, transportadores, armazéns, etc., mas ainda o sistema de informação e o sistema organizacional. Além disso, a modelação destes três aspectos distintos deve ser tratada de uma forma integrada e global, levando à necessidade da modelação das interfaces existentes entre cada um dos referidos sistemas.

O trabalho realizado no âmbito desta dissertação enquadra-se neste contexto, tendo sido dividido em duas partes que se complementam. A primeira dessas partes situa-se no âmbito das infra-estruturas e arquitecturas de integração para sistemas de fabrico, correspondendo a trabalho de investigação realizado primeiro no decorrer do projecto europeu SHOP-CONTROL e posteriormente no âmbito da construção do PROFIT (um sistema integrado para a gestão das operações fabris). A segunda parte está relacionada com a modelação do sistema de informação e a sua integração com o modelo do sistema físico de fabrico e o fluxo de materiais.

No âmbito das infra-estruturas de integração, apresenta-se a contribuição realizada para a especificação da arquitectura bem como para a especificação e desenvolvimento de infra-estruturas de suporte do SHOP-CONTROL, assim como de ferramentas de apoio à construção de aplicações de monitorização de condição do processo de fabrico.

Em relação à segunda parte do trabalho desenvolvido, são abordadas as Arquitecturas do Equipamento de Fabrico, do Sistema de Informação e Humana e Organizacional. É dada especial relevância à modelação tanto do sistema físico de fabrico (Arquitectura do Equipamento de Fabrico) como do sistema de informação de gestão e controlo (Arquitectura do Sistema de Informação). De facto, é na modelação da Arquitectura do

Sistema de Informação e na sua integração com a Arquitectura do Equipamento de Fabrico que se encontra a principal contribuição do autor. A Arquitectura Humana e Organizacional é também abordada mas de forma muito breve, uma vez que as primitivas de modelação desenvolvidas pelo autor estão a ser utilizadas na construção de uma ferramenta de modelação, simulação e análise sócio-técnica em Sistemas Integrados de Fabrico.

A ferramenta desenvolvida implementa um conjunto de blocos construtivos básicos para a modelação da Arquitectura do Sistema de Informação usando uma técnica de descrição formal, o SDL ("Specification and Description Language"). Estes blocos construtivos básicos de suporte à linguagem de especificação formal, foram construídos usando primitivas já disponíveis numa ferramenta comercial usada na modelação do sistema físico de fabrico e do respectivo do fluxo de materiais. Este facto tornou possível a construção de modelos do sistema de informação fabril que, através da introdução de conceitos de interacção do tipo "sensor" e "actuador", foi possível integrar com o modelo do sistema físico de fabrico. O sucesso desta integração permite que seja actualmente possível a construção de modelos integrados onde existe uma verdadeira interacção entre os modelos executáveis do sistema de informação e do sistema de fabrico. Resultou deste modo uma Bancada de Modelação (BM), fornecendo um ambiente integrado de modelação e simulação onde é possível descrever e analisar a interacção entre o fluxo de informação de controlo e de gestão e o fluxo de materiais inerentes ao processo de fabrico.

A BM suporta pois a especificação, desenvolvimento, teste, integração e validação de componentes e subsistemas do sistema de informação no âmbito dos Sistemas Integrados de Fabrico. A ligação da BM à infra-estrutura de integração SHOP-CONTROL permite realizar o teste e validação na fábrica, de aplicações fabris, autónomas e cooperantes num ambiente distribuído. Esta integração vertical permite a integração da BM com outras aplicações de gestão fabril e a sua interacção com a instalação fabril propriamente dita, facto que possibilita a reutilização do núcleo do modelo de simulação fabril em execução na BM no controlo efectivo das operações de fabrico.

Abstract

This thesis deals with the development and use of executable models within the scope of Integrated Manufacturing Systems or Computer Integrated Manufacturing (CIM), the latter being the area where many research efforts have been made both from the technological and from the human behaviour points of view. The research work undertaken lies in the borderline of three domains: open architectures for CIM; integrating infrastructures; and particularly executable models in the control of the actual integrated manufacturing system.

The manufacturing enterprise environment involves many different issues that should be perceived and dealt with through the use of different types of models, the integration of these models representing the best environment for decision support. The modelling facilities should cover not only the physical manufacturing system, i.e. machines, conveyors, warehouses, etc., but the corresponding information and organisational systems as well. Moreover, the modelling of these three different aspects should be managed globally, therefore requiring their interfaces to be modelled.

The work undertaken within the scope of this thesis fits in the previously described framework, and was divided into two complementary parts. The first of these parts deals with architectures and infra-structures for manufacturing systems integration, putting into focus the work undertaken at first during the European project SHOP-CONTROL and later in the construction of PROFIT (an integrated shop floor control system). The second part is concerned with the information system modelling, and its integration with the physical manufacturing system model and the inherent material flow.

Concerning the first part of the work, this document presents the author's contribution for the specification, design and development of the SHOP-CONTROL architecture and supporting infra-structures, as well as for the construction of process condition monitoring tools.

In the second part, the Manufacturing Equipment, Information System and Organisational Architectures are a major subject. Special attention was given to the modelling of both the actual physical manufacturing system (Manufacturing Equipment Architecture), and the information system managing and controlling it (Information System Architecture). The main contribution of the author lies in fact in the modelling of the Information System Architecture, and in its integration with the Manufacturing Equipment Architecture. The Human and Organisational Architecture is also dealt with briefly, as the

modelling primitives developed by the author are currently being used to build a modelling tool allowing the socio-technical simulation and analysis of integrated manufacturing systems.

The developed modelling tool encompasses a basic set of building blocks supporting an existing formal description technique SDL (Specification and Description Language), which are presently being used to model the Information System Architecture. These constructs supporting the formal description technique were built re-using basic building blocks provided by a commercial tool for modelling the manufacturing physical system and corresponding material flow. This enabled the integration of both the information and the manufacturing physical systems models through the use of interaction concepts such as *sensor* and *actuator*. The success of this integration allows the construction of integrated models, where the actual interaction across the several executable models may simulated. The result was therefore a modelling workbench (MW) providing integrated modelling and simulation facilities of both the material flow and the management / control information flow models related with the actual manufacturing process.

This MW is actually supporting the specification, development, test, integration and validation of information technology components and sub-systems within the scope of integrated manufacturing systems. The existing link between the MW and the SHOP-CONTROL integrating infra-structure further enhances these facilities, by enabling *on site* test and validation of shop floor management applications in a distributed environment. This vertical integration, which allows the communication of the MW with the existing shop floor management applications, supports real-time interaction with the actual shop floor, thereby allowing the reuse of the simulation control kernel in the actual control of shop floor operations.

Résumé

L'aide à la décision dans l'environnement de l'entreprise doit s'appyuer sur l'utilisation integrée de différents modéles issus de l'ensemble des sous systèmes y existants. Les outils de modelation doivent prender en considération les trois aspects suivants: les aspects physiques (machines, transporteurs, magasins, etc..), les aspects d'information et les aspects d'organisation d'un processus de fabrication. Par ailleurs la modelation de ces trois aspects différents doit être traité de façon integrée et global, conduisant à un besoin de définition/spécification des interfaces existentes entre ces systèmes.

Ce travail cible la conception le development et l'utilisation de modéles executables dans le cadre de l'integration de systèmes en FAO (Fabrication Assisté par Ordinateur). Il est centrée dans la modelation et contrôle de systèmes integrées de fabrication avec des modéles executables, est fondée dans les concepts d'architectures ouvertes en FAO ("CIM") et d'infrastructures d'intégration.

Le mémoire se décompose en deux parties complémentaires. La premiére concerne les infra-strutures et architectures d'integration pour des systèmes de la fabrication et à été developpé dans le cadre de recherche d'un project européen SHOP-CONTROL, qui à débouché dans le produit d'aide à la conduite de procedés industriels (PROFIT). Ici, on constate une contribuition dans le probléme de la spécification de l'architecture bien que les infra-strutures de support du SHOP-CONTROL ansi que les outils necessairs à la construction des applications de suivi de conditions du procés de fabrication. La deuxiéme activitée est en rapport avec la conception du modèle du système d'information et son integration avec le modèle du système physique de fabrication. Par rapport á la deuxiéme activité developpé on voit les architectures de l'equipement de fabrication , du système d'information, et l'architecture humaine et d'organisation. D'autre part, on met en évidence la modelation du système physique et du système d'information de gestion et contrôle (architecture de l'equipement de fabrication et architecture du système d'information respectivement).

C'est dans la modelation de l'architecture du système d'information et son integration avec l'architecture de l'equipement de fabrication que l'on trouve la principale contribution de ce travail. L'architecture humaine et d'organisation et abordée de façon élémentaire, étant donné que les primitives de modelation du système d'information son été utilisées dans la construction d'un outil d'analyse socio-technique en systèmes integrées de fabrication.

L'outil developpée est constitué par un ensemble de modules de construction élémentaire et utilise une technique de description formal le SDL ("Specification and Description Language"). Ces modules de construction sont le suport du langage de specification formal, et ont été basées sur des primitives existantes dans un système de simulation et de modélation de flux de materiaux.

L'introduction des notions d'interaction type "récepteur" et "opérateur" a permis la construction de modéles du système d'information et son intégration avec le système physique de fabrication.

Le rassemblement de ces concepts permet la construction de modéles où existe une interaction entre les modéles executables du système de fabrication et d'information.

Le résultat fut un atelier de modélation (AM) qui offre un environnement integrée de modelation et simulation, supportant simultanément le flux d'information de gestion et contrôle et le flux de materiaux du processus de fabrication.

D'autre part il support le developpement, la spécification, l'essay, l'integration et la validation des composants/soussystèmes dans les systèmes integrées de fabrication. La liaison entre l'atelier (AM) et l'infrastructure d'intégration du project SHOP-CONTROL offre la possibilité de realisation d'essays dans le cas réel, prennant en compte les contraintes y existantes. Cette integration verticale permet la utilisation du noyau du modèle en exécution dans l'atelier pour le contrôle effectif des operations de production.

Agradecimentos

Quero manifestar em primeiro lugar o meu grande e sincero reconhecimento ao Sr. Prof. Doutor José Manuel Mendonça, meu orientador científico, que deste sempre me tem privilegiado com o seu valioso apoio e confiança, enriquecendo o meu trabalho com os seus ensinamentos e estímulos.

Ao INESC e ao Grupo de Automação Industrial e Gestão de Energia, liderado pelo Sr. Prof. Doutor José Manuel Mendonça, agradeço o excelente ambiente de trabalho que me foi proporcionado. Aos meus colegas do projecto SHOP-CONTROL, Luís Carneiro, João Pascoal Faria, Pedro Martel, José António Torres, Helena Brás, Mário Nunes e Rui Picas, quero também agradecer a sua preciosa colaboração.

Aos meus colegas Lucas Soares e Ângelo Martins agradeço os comentários que ao longo destes dois últimos surgiram das nossas discussões.

Ao Sr. Prof. Doutor João Falcão e Cunha quero também exprimir de um modo especial a minha gratidão, pela organização e coordenação do processo de candidatura à submedida 4.2, acção 12.19 do programa PRODEP assim como ao Sr. Prof. Doutor Raul Moreira Vidal a sua disponibilidade e o seu parecer, que em conjunto com o parecer do Sr. Prof. Doutor José Manuel Mendonça tornaram possível a atribuição da bolsa de doutoramento do PRODEP, que me foi atribuída pelo período de dois anos. Esta permitiu-me a participação em conferências nacionais e internacionais, dando-me possibilidade de contactar com a comunidade científica que se dedica ao estudo dos problemas abordados neste trabalho de dissertação.

Finalmente aos meus pais, que entre muitos outros ensinamentos me transmitiram que "*há muitas maneiras de medir a vontade humana, sendo a mais exacta e a mais segura a que se exprime na pergunta: de que esforço seremos nós capazes ?*", o meu obrigado por tudo.

A todos os outros, que de alguma forma facilitaram a realização deste trabalho, quero expressar o meu agradecimento.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

AE	Actividade da Empresa
AEF	Arquitectura do Equipamento de Fabrico
AFP	Análise do Fluxo da Produção
AHO	Arquitectura Humana e Organizacional
AICIME	"Automotive Industry CIME"
AID	"AMICE Implementation Description"
AMICE	"European CIM Architecture" (ECIMA, lido no sentido inverso), Projecto ESPRIT nº 5288
ARIS	"Architecture of Integrated Information Systems"
ASI	Arquitectura do Sistema de Informação
BD	Base de Dados
BM	Bancada de Modelação
BOM	"Bill Of Materials"
CAD	"Computer-Aided Design"
CAE	"Computer-Aided Engineering"
CAM	"Computer Aided Manufacturing"
CAP	"Computer Aided Planning"
CAPP	"Computer Aided Production Planning"
CAQ	"Computer-Aided Quality Control / Assurance"
CASE	"Computer Assisted Software Engineering"
CAT	"Computer-Aided Testing"
CCITT	"Comité Consultatif International Telegraphique et Telephonique"
CD	Centro de Decisão

CEFSM	"Comunicating Extended Finite-state Machines"
CEN	Comité Europeu de Normalização
CIM	"Computer Integrated Manufacuring"
CIM-BIOSYS	"CIM Building Integrated Open SYStems"
CIME	"CIM and Engineering"
CIMOSA	"Open Systems Architecture for CIM"
CO	Centro de Operação
COSIMA	"Control Systems for Integrated Manufacturing", Projecto ESPRIT nº 477
CPM	"Critical Path Method"
CQ	Controlo Qualidade
CRP	Conjunto de Regras de Procedimento
DCE	"Distributed Computing Environment"
DFD	"Data Flow Diagram"
DM	Domínio
ECMA	"European Computer Manufacturers Association"
EDI	"Electronic Data Interchange"
EDT	"Estelle Development Toolset"
EIS	"Executive Information Systems"
ESPRIT	"European Strategic Program for Research in Information Technology"
EV	Eventos da Empresa
FIFO	"First-In-First-Out"
FMS	"Flexible Manufacturing Systems"
GERAM	"Generic Enterprise Architecture and Methodology"
GERT	"Graphical Evaluation and Review Technique"
GIM	"GRAI Integrated Methodology"
GR	Gestor de Regras
GRAI	"Graphe de Résultats e Activités Interreliés"
GROPE	"Graphical Representation of Protocols in Estelle"
GTF	Gestor de Transacções Fabris
HFP	"Hierarquia Funcional de Purdue"
I&D	Investigação e Desenvolvimento
IEM	"Integrated Enterprise Modelling"

IF	Instalação Fabril
IFAC	"International Federation of Automatic Control"
IFIP	"International Federation For Information Processing"
IiM	"Integration in Manufacturing"
IMS	"Inteligent Manufacturing Systems"
ISF	Integração do Sistema de Fabrico
ISO	"International Standards Organization"
ITU	"International Telecomunications Union"
IU	Interface com o Utilizador
JIT	"Just-in-Time"
LOTOS	"Language Of Temporal Ordering Specification"
MADIF	Módulo de Acesso Distribuído à Instalação Fabril
MDI	Modelo de Descrição da Implementação
MDR	Modelo de Definição de Requisitos
MEP	Modelo de Especificação do Projecto
MES	"Manufacturing Execution Systems"
MMS	"Manufacturing Message Specification"
MOO	Modelação Orientada ao Objecto
MRP	"Material Requirements Planning"
MRP-II	"Material and Resource Planning"
MSC	"Message Sequence Chart"
MSI	"Manufacturing Systems Integration - Research Institute"
ODBC	"Open Data Base Connectivity"
ODP	"Open Distributed Processing"
OE	Objectos da Empresa
OSF	"Open Software Foundation"
OSI	"Open Systems Interconnection"
PC	Computador Pessoal
PD	Processos do Domínio
PERA	"Purdue Enterprise Reference Architecture"
PERT	"Program Evaluation and Review Technique"
PLC	Controlador Lógico Programável
PN	Processo do Negócio

POO	Programação Orientada ao Objecto
PP&C	"Production Planning and Control"
PROFIT	"PRoduction Optimization throught Flexible Integrated software Tools"
QBD	Quadro de Bordo Dinâmico
REAL-I-CIM	"REAL-time Interactive shop floor CIM", Projecto ESPRIT nº 8865
RIC	REAL-I-CIM
RPC	"Remote Procedure Call"
SA	"Structured Analysis"
SA	Serviço de Apresentação
SA-RT	"Structured Analysis - Real-Time"
SADT	"Structured Analysis and Design Technique"
SAF	Sistema Avançado de Fabrico
SC	Serviços Comunicação
SCADA	"Supervisory Control and Data Aquisition"
SDL	"Specification and Description Language"
SEW-OSA	"Systems Engineering Workbench centred on CIMOSA"
SFC	"Shop Floor Control"
SFCS	"Shop Floor Control Systems"
SGI	Serviço de Gestão de Informação
SGOF	Sistema de Gestão de Operações Fabris
SHOP-CONTROL	Projecto ESPRIT nº 5478
SIF	Sistema Integrado de Fabrico
SIG	Serviço de Informação Global
SN	Serviços do Negócio
SPC	Controlo Estatístico do Processo
SQL	"Structured Query Language"
STM	Sistema de Transmissão de Mensagens
TDF	Técnicas de Descrição Formal
TQM	"Total Quality Management"
UAR	Unidade de Acesso à Rede
UoD	Universo do Discurso
VO	Vistas de Objectos

Capítulo 1

INTRODUÇÃO

1.1 Âmbito da dissertação

Nesta dissertação são tratados a concepção, o desenvolvimento e a utilização de modelos executáveis no âmbito da Integração de Sistemas de Fabrico ou do Fabrico Integrado por Computador (CIM), área esta na qual têm sido realizados múltiplos esforços em vários domínios da tecnologia, das ciências humanas e do comportamento. Estes trabalhos têm sido realizados em áreas muito distintas, representativas da evidente multidisciplinaridade e abrangência do tema, tais como: as arquitecturas abertas para a integração dos sistemas de fabrico; as infra-estruturas de integração; a modelação, especificação e análise dos processos do negócio; a engenharia concorrente; o suporte ao trabalho colaborativo; os sistemas de informação cooperativos e as bases de dados federadas; os formatos para a troca electrónica de informação; a capitalização do conhecimento e sua protecção; a cooperação homem-máquina; a estandardização; a engenharia da empresa; as abordagem integradas globais aos sistemas de fabrico ("holonic manufacturing systems"); etc.

De entre as diversas áreas de estudo mencionadas, esta dissertação concentra a sua atenção nas arquitecturas abertas para os Sistemas Integrados de Fabrico, nas infra-estruturas de integração, e particularmente na modelação e controlo do sistema integrado de fabrico através de modelos executáveis sobre uma infra-estrutura de integração.

O âmbito do trabalho apresentado neste documento está intimamente ligado ao domínio de aplicação dos chamados "Sistemas de Execução Fabril" ou "Sistemas de Controlo de Chão-de-fábrica", onde a diversidade dos problemas resulta na necessidade de ferramentas de decisão flexíveis baseadas em modelos adequados, organizadas segundo uma arquitectura aberta e suportadas por uma infra-estrutura de informação transportável. Além disso, a experiência mostra também que cada projecto de integração é um novo caso, sendo assim premente a necessidade de arquitecturas de referência que proporcionem linhas

mestras de condução dos trabalhos e um elevado nível de uniformização de procedimentos e de estandardização de componentes.

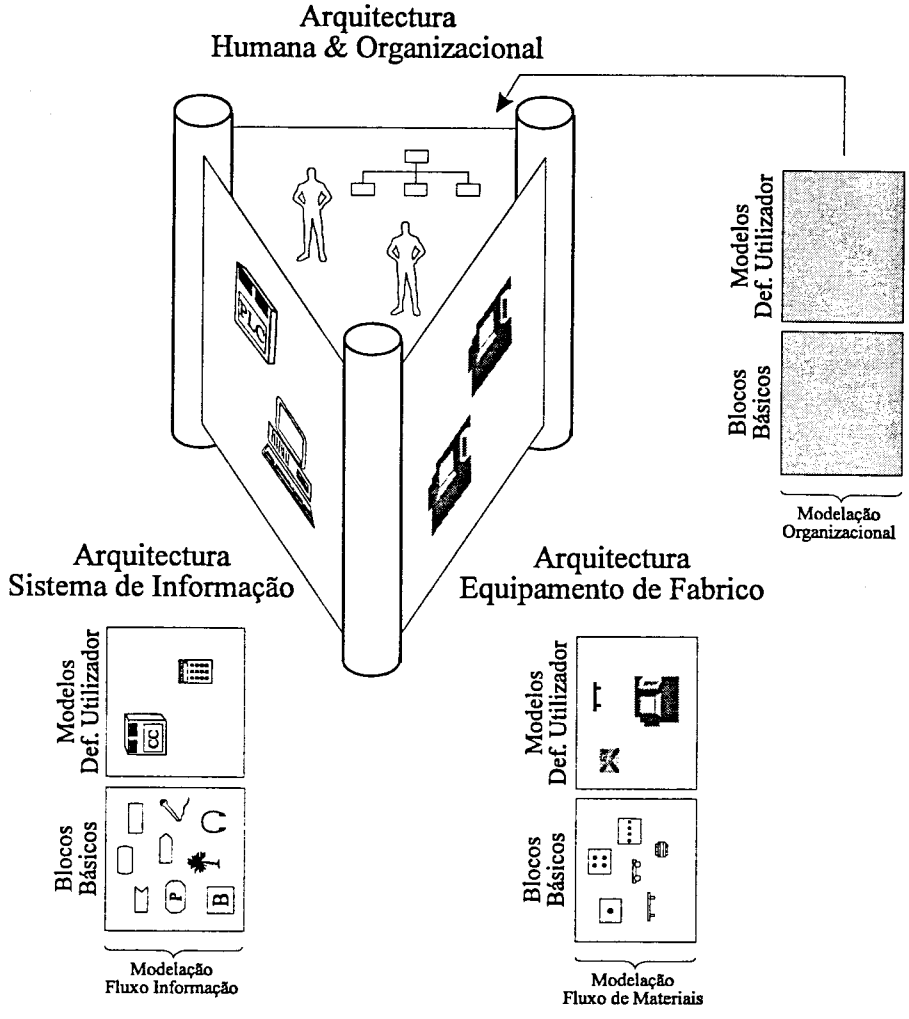


Figura 1.1 - O Ambiente Integrado de Modelação do Sistema de Fabrico

A multidisciplinaridade existente no ambiente empresarial deverá reflectir-se na utilização de modelos de vários tipos que, uma vez integrados, retratem a realidade da forma mais adequada ao apoio à decisão. De acordo com as três arquitecturas de implementação apresentadas na arquitectura de referência de Purdue, as facilidades de modelação devem cobrir não só o tradicional sistema físico de fabrico, mas ainda o sistema de informação e o sistema organizacional da empresa [1]. Além disso, a modelação destes três aspectos distintos que caracterizam a empresa de fabrico deve ser tratada de uma forma integrada e global, levando à necessidade da modelação das interfaces existentes entre cada um dos referidos sistemas. O cenário apresentado pela figura 1.1 ilustra a posição relativa daqueles sistemas e das suas arquitecturas através da apresentação das suas interfaces. É assim visível a arquitectura do equipamento de fabrico, apresentando interface com a arquitectura da organização (através dos operadores dos recursos físicos de fabrico), e com a arquitectura do sistema de informação (interface incluindo componentes tecnológica e humana). Por sua vez, a arquitectura do sistema de informação apresenta uma interface com a Arquitectura Humana e Organizacional constituída pelos operadores humanos. O apoio à decisão suportado pelas facilidades de modelação passa deste modo pela utilização de ferramentas que permitam a especificação, projecto, implementação, avaliação e teste de

soluções através da simulação: soluções para o "layout" do sistema de fabrico, para a arquitectura do sistema de informação, para as funcionalidades do software de controlo e decisão, e para a organização da empresa.

1.2 Trabalho realizado

O trabalho realizado no âmbito desta dissertação foi dividido em duas partes que se complementam. A primeira dessas partes enquadra-se no âmbito das infra-estruturas e arquitecturas de integração para sistemas de fabrico, trabalho de investigação realizado no decorrer do projecto ESPRIT ("European Strategic Program for Research in Information Technology") SHOP-CONTROL, e posteriormente no âmbito da construção do produto PROFIT (um sistema integrado para a gestão de operações fabris). A segunda parte do trabalho desenvolvido está relacionada com a modelação da arquitectura do sistema de informação e a sua integração com o fluxo de materiais.

No âmbito das infra-estruturas de integração, existiu de facto contribuição significativa do autor na especificação da arquitectura bem como das infra-estruturas de suporte ao projecto SHOP-CONTROL [2,3,4]. Ainda consideradas no âmbito do SHOP-CONTROL como infra-estruturas num sentido muito abrangente do termo, foram especificadas e desenvolvidas uma aplicação de Supervisão de processos de fabrico bem como um Módulo de Acesso Distribuído à Instalação Fabril, ferramentas de apoio à construção de aplicações de monitorização de condição ao nível da instalação fabril [4,5]. O surgir do projecto ESPRIT REAL-I-CIM permitiu que a referida aplicação se Supervisão evoluísse de acordo com conceitos inovadores de "Quadro de Bordo Dinâmico" [6].

A figura 1.1 ilustra as três perspectivas segundo as quais um sistema integrado de fabrico deve ser observado e tratado. No âmbito desta dissertação, serão abordados estes três aspectos, com especial relevância para a modelação do sistema físico de fabrico (Arquitectura do Equipamento de Fabrico) e do sistema de informação de gestão e controlo (Arquitectura do Sistema de Informação). De facto, é na modelação da Arquitectura do Sistema de Informação e na sua integração com a Arquitectura do Sistema de Fabrico que se encontra a principal contribuição do autor. A Arquitectura Humana e Organizacional é também abordada embora de forma muito breve, uma vez que as primitivas de modelação desenvolvidas no âmbito deste trabalho estão a ser utilizadas na construção de uma ferramenta de modelação, simulação e análise sócio-técnica em Sistemas Integrados de Fabrico. Nesta ferramenta, é modelado não só o sistema de informação, mas também a estrutura organizacional da empresa e a sua interacção com o próprio sistema de informação [7].

No decorrer da execução deste trabalho, foi implementado um conjunto de blocos construtivos básicos para a modelação da Arquitectura do Sistema de Informação usando uma técnica de descrição formal. Estes blocos foram construídos usando primitivas básicas já disponíveis numa ferramenta comercial de suporte à modelação do sistema físico de fabrico e do respectivo do fluxo de materiais. Este facto tornou possível a construção de modelos do sistema de informação fabril que, através da integração de conceitos simples de interacção como "sensor" e "actuador", puderam ser integradas com o modelo do sistema físico de fabrico. O sucesso desta integração permite que actualmente seja possível a construção de modelos integrados onde existe uma verdadeira interacção entre os modelos executáveis do sistema de informação e do sistema de fabrico. Resulta deste modo um ambiente de modelação e simulação que suporta de facto a interacção entre o fluxo de

informação de controlo ou gestão e o fluxo de materiais. O trabalho desenvolvido resultou entretanto na agora disponível Bancada de Modelação (BM), um ambiente integrado suportando a especificação, desenvolvimento, teste, integração e validação de componentes e subsistemas no âmbito dos Sistemas Integrados de Fabrico. A ligação da BM à infra-estrutura de integração do SHOP-CONTROL permite que o teste e validação de aplicações fabris autónomas mas cooperantes num ambiente distribuído se realize no ambiente de modelação / simulação da BM, incluindo quando necessário interacções com o sistema real através da referida infra-estrutura. Esta infra-estrutura suporta a interacção da BM com outras aplicações de gestão fabril (integração horizontal), e facilita a sua comunicação com a instalação fabril propriamente dita (integração vertical), facto que permite à reutilização do núcleo do modelo de simulação fabril em execução na BM no controlo efectivo das operações de fabrico.

Ao rever o estado-da-arte, a utilização de modelos executáveis surge como uma consequência natural da necessidade de ferramentas que permitam a construção de sistemas à medida e de baixo custo, e que acompanhem o ciclo de vida completo do sistema integrado de fabrico. Ultrapassada a tendência inicial para a utilização de modelos de simulação descartáveis, os modelos que é agora possível construir tirando partido das ferramentas estado-da-arte podem ser reutilizados na gestão e controlo diário das operações na instalação fabril, desde que a sua execução seja suportada por uma infra-estrutura integradora. Neste contexto, e dadas as semelhanças existentes, é reutilizado o núcleo de controlo do modelo de simulação no controlo do próprio sistema de fabrico que esse mesmo modelo representa.

1.3 A Organização desta Tese

Esta dissertação encontra-se organizada em sete capítulos. No presente capítulo faz-se uma breve introdução ao tema da dissertação, ao seu âmbito e às motivações científicas e de aplicação para a realização deste trabalho de investigação.

No capítulo segundo intitulado *Fabrico Integrado por Computador*, será abordada, numa perspectiva histórica, a influência das tecnologias da informação na evolução dos sistemas de fabrico até aos dias de hoje. O capítulo conclui com uma visão actual do sistema de fabrico discreto, onde se procura a utilização racional dos factores de produção. São finalmente abordadas as diversas actividades desenvolvidas no seio de um sistema de fabrico, tendo em atenção os seus horizontes temporais, bem como os fluxos de materiais, informação (técnica e de gestão) e financeiros envolvidos.

O capítulo terceiro, denominado *Arquitecturas de Integração*, apresenta as arquitecturas de suporte à integração das actividades de fabrico e empresariais mais representativas do estado-da-arte. Neste contexto, é dada particular relevância às arquitecturas alvo de discussão em grupos de trabalho internacionais, tais como o IFAC/IFIP (International Federation of Automatic Control / International Federation For Information Processing), ou com relevância no âmbito da normalização, e em particular no Comité Europeu de Normalização no qual o autor tem tido parte activa desde Fevereiro de 1995 como perito Português ao CEN/TC 310/WG 1 (Advanced Manufacturing Technologies CEN/TC 310 /WG 1 Systems Architecture).

O capítulo quarto, intitulado *Infra-estruturas de Suporte a Sistemas de Gestão de Operações Fabris*, apresenta os requisitos colocados pelo sistema integrado de fabrico às infra-estruturas de suporte. Uma vez apresentados os requisitos genéricos, são explanados

alguns exemplos de implementação de infra-estruturas disponíveis actualmente sob a forma de protótipo. Ainda neste contexto é apresentada a infra-estrutura do produto comercial PROFIT na génese do qual o autor teve uma intervenção activa ao longo do projecto ESPRIT que lhe deu origem, o SHOP-CONTROL. No âmbito do PROFIT são apresentadas as contribuições do autor na aplicação MADIF (Módulo de Acesso Distribuído à Instalação Fabril) e na aplicação de Supervisão.

O capítulo quinto, intitulado *Suporte à Gestão de Operações Fabris através de Modelos Executáveis*, começa por apresentar as necessidades, do ponto de vista do suporte de metodologias e ferramentas de modelação, colocadas pela construção de um sistema integrado de fabrico. Estabelecidas as necessidades, os temas da modelação do sistema de fabrico e do sistema de informação são desenvolvidos com algum detalhe. São finalmente apresentados dois exemplos de modelos executáveis de suporte ao sistema de fabrico, respectivamente os modelos propostos pela arquitectura CIMOSA e pelo PROFIT. No âmbito da apresentação do PROFIT é detalhada a metodologia seguida na construção e utilização do modelo fabril.

O capítulo sexto, intitulado *Modelação do Sistema de Informação - Integração dos Fluxos de Materiais e de Informação*, apresenta o trabalho desenvolvido pelo autor no âmbito da modelação da Arquitectura do Sistema de Informação bem como a sua integração com a Arquitectura do Equipamento de Fabrico.

No sétimo e último capítulo, são finalmente apresentadas as conclusões do trabalho realizado. Neste contexto, são apontadas algumas direcções possíveis para a futura evolução do trabalho até este momento realizado e que se enquadram no âmbito da presente dissertação.

Capítulo 2

FABRICO INTEGRADO POR COMPUTADOR

2.1 Considerações históricas no âmbito do Fabrico Integrado por Computador

Sempre que pensamos na indústria, pensamos sobretudo em produtos. Os produtos reflectem os nossos desejos, medos, julgamentos e preconceitos. É assim consensual que os produtos industriais, bens de equipamento ou de consumo, tais como máquinas, robots, automóveis, aviões, computadores, aparelhos de TV, canetas e fotocopiadoras, mudaram radicalmente o mundo em que vivemos. No entanto, esta ideia só transmite parte da verdade. Do ponto de vista histórico, não são os produtos mas sim os métodos da sua produção que deram forma à história da nossa cultura. As roupas já eram tecidas na antiguidade, mas foi o tear mecânico que pela primeira vez transformou os têxteis num produto de consumo. Os métodos de produção de ferro e de aço, desenvolvidos no Próximo Oriente alguns anos antes de Cristo, influenciaram o curso da história mais do que qualquer batalha ou governante que se sucederam ao longo de todos esses anos. Justifica-se assim a relevância de um breve estudo da indústria do fabrico.

2.1.1 Um Século de desenvolvimento de Sistemas de Fabrico

O processo de industrialização que se iniciou no século passado nasceu na sequência da exploração da máquina a vapor consideravelmente melhorada por James Watt. Foi, por outras palavras, resultado de uma mudança na tecnologia da produção. Uma vez que a energia usada para mover a maquinaria e equipamentos de fabrico era no início transmitida por meios mecânicos, todas as máquinas tiveram que se concentrar num mesmo local. Provocou-se assim o desenvolvimento da fábrica. A industrialização foi pois o resultado de uma mudança significativa do papel da energia como factor de produção. A energia sempre fez parte de todo o processo de fabrico, passando nesse momento a ser possível disponibilizá-la em praticamente qualquer ponto em grandes quantidades [8].

Na segunda revolução industrial, tipificada pela produção em massa para a indústria automóvel nos anos 30, 40 e 50, o factor trabalho adquiriu um peso dominante. Este facto deveu-se principalmente às crescentes exigências de qualificações da população trabalhadora. O surto de desenvolvimento associado provocou o aumento da taxa de produção líquida homem/hora, mas estabeleceu também a base para o método da partilha dos lucros e abriu caminhos ao aumento do poder da greve [9].

Estes factos levaram na Alemanha a um aumento real dos salários *per capita* entre 1950 e 1975, superior a dez vezes a taxa de crescimento em períodos anteriores de igual duração. Estava a viver-se a idade da racionalização durante a qual as técnicas de produção nos mais variados ramos da indústria sofreram mudanças radicais. É assim altura de observar que os sucessos da primeira revolução industrial e da industrialização propriamente dita atingiram então um ponto de saturação [8].

As limitações da produção em massa foram todavia ultrapassadas, uma vez que na década de 70 se assistiu à denominada terceira revolução dos processos de produção, com a era da automação (figura 2.1). A evolução dos meios de processamento electrónico da informação, proporcionando serviços de valor crescente com custos progressivamente decrescentes, deu origem a um novo factor de produção que, tal como o trabalho e a energia, sempre esteve presente nos processos de fabrico: a informação, inerente ao próprio processo de fabrico e aos processos de controlo e gestão a ele associados.

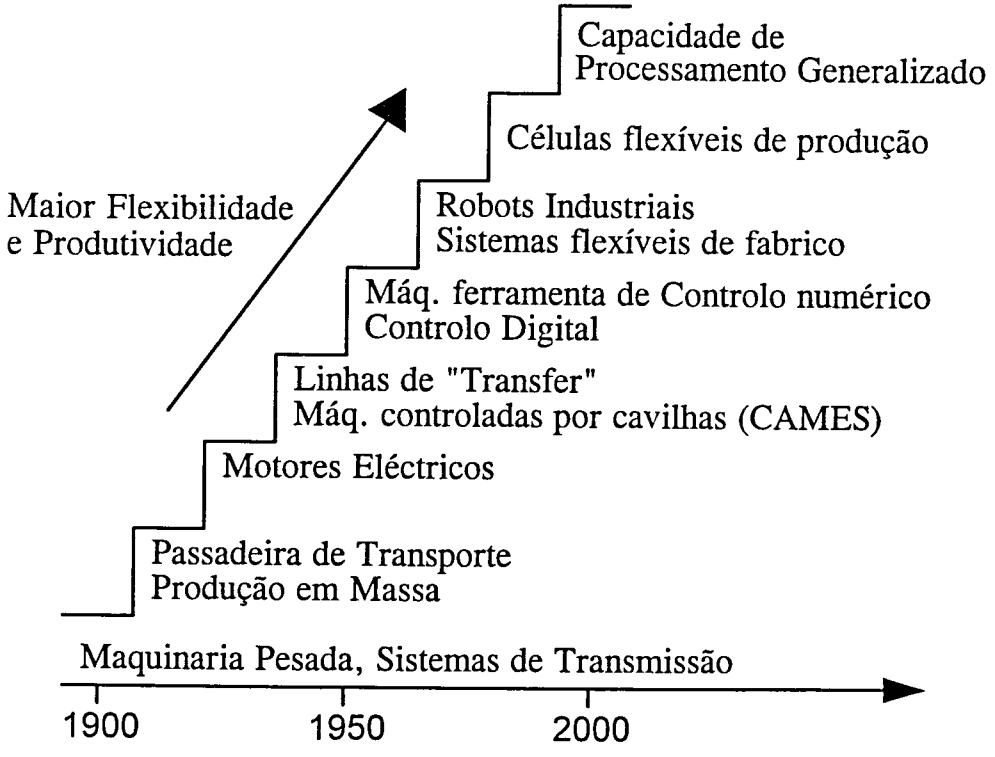


Figura 2.1 - Estádios de Desenvolvimento das Tecnologias no Fabrico [8]

Assim sendo, surge a imagem do sistema de fabrico como um sistema dinâmico em que fluxos de energia controlados por fluxos de informação actuam sobre fluxos de materiais, de forma a transformar matéria prima e componentes em produto acabado. Esta imagem nada diz sobre a natureza da transformação da informação, e deve observar-se que

nem todos os processos de automação e controlo necessitam da aplicação de técnicas de processamento electrónico da informação.

Primeiro em tarefas de automação de maquinaria e equipamento de fabrico e transporte; depois no apoio a tarefas de engenharia, planeamento e controlo de fabrico e gestão empresarial; e finalmente na integração dos múltiplos processos, unidades funcionais e pessoas da empresa industrial, o computador teve e tem um impacto considerável em quase todas as actividades relacionadas com o fabrico. Em muitas situações provocou mesmo a alteração das estruturas organizacionais e tornou necessária a adopção de novas estruturas de gestão. Por outro lado tarefas de fabrico e funções de gestão mudaram drasticamente, uma vez que o computador é preferencialmente capaz de realizar ou controlar de uma forma mais eficiente, não só tarefas repetitivas "mecanizáveis" como ainda funções de suporte especializado e dito "inteligente" a processos de decisão em ambientes complexos e dinâmicos.

Nos anos 80, e na sequência desta evolução, surge o conceito de Fabrico Integrado por Computador ("Computer Integrated Manufacturing", CIM) introduzindo um ambiente caracterizado por agregar a informação utilizada para apoiar as actividades da empresa, tanto do ponto de vista físico como do ponto de vista lógico, através de sistemas de processamento de informação interligados por uma rede de comunicações. As empresas industriais de sectores mais avançados tecnologicamente, mas também pioneiros na competição em mercados globais, rapidamente reconheceram a necessidade de incrementar continuamente as suas capacidades no CIM explorando os potenciais ganhos em produtividade e competitividade daí resultantes. Foi o caso das indústrias de processos, electrónica e automóvel, as primeiras a investir fortemente na automação e na informatização. A aproximação usual consistia todavia no desenvolvimento de interfaces específicos para estabelecer a ligação entre as múltiplas "ilhas de automação" existentes associadas aos equipamentos automatizados de fabrico e aos primeiros sistemas informáticos que surgiram para apoio da engenharia e da gestão administrativa. Esta estratégia resultou na existência de múltiplos componentes CIM (por exemplo sistemas, pacotes de software, equipamentos), cuja transferência, adaptação e redesenho para outros sectores industriais e empresas de menor dimensão e inferior nível tecnológico se desenrolou com atraso e sobretudo na última década.

A tecnologia, tanto ao nível dos equipamentos de fabrico como dos sistemas de informação de apoio ao fabrico deixou pois de estar exclusivamente ao alcance das empresas de grande dimensão e com avultados recursos financeiros, tipicamente em sectores industriais de capital intensivo. A última década foi com efeito a da "democratização" das tecnologias da automação e informação na indústria, conforme é possível constatar por exemplo com os sectores tradicionais da indústria portuguesa (figura 2.2).

As indústrias de engenharia e de fabrico vão à cabeça na batalha para a sobrevivência económica da Europa. Com a aproximação do próximo século estas indústrias estão a conhecer uma revolução sem precedentes. A globalização do fabrico e dos mercados pressiona as empresas no sentido da redução dos seus custos, assim como do aumento da sua capacidade de resposta às solicitações do mercado para a entrega dentro de prazos e de produtos com elevada qualidade fabricados à medida. Em simultâneo, a crescente preocupação das sociedades modernas com aspectos ambientais e sociais coloca às indústrias um grande desafio na adaptação dos produtos e dos processos produtivos a estes novos requisitos [10].

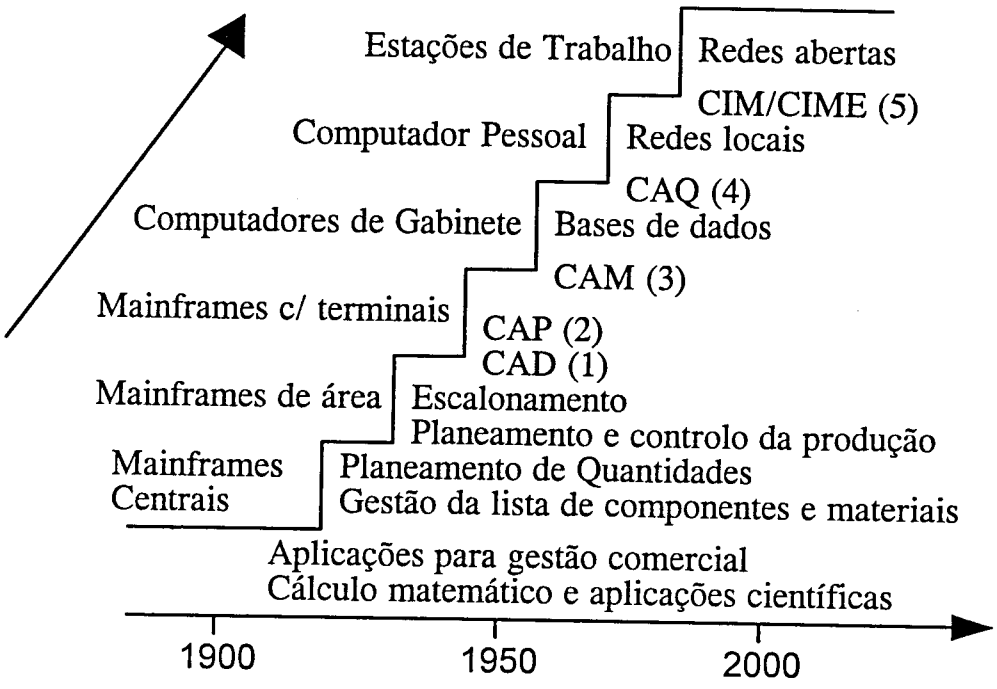


Figura 2.2 - Processamento da Informação no Fabrico: descentralização progressiva, com facilidades crescentes de processamento, interface e integração[1] [8]

2.1.2 O esgotar dos modelos tradicionais e os novos paradigmas

Neste cenário de globalização da economia a automação/informatização local esgota-se rapidamente, pois os múltiplos processos envolvidos no fabrico e no negócio tornam-se mais complexos e sobretudo distribuídos temporal e geograficamente. A próxima década exige pois abordagens inovadoras que passam pela alteração das estruturas organizacionais e das estratégias de gestão à medida que a empresa avança na re-engenharia gradual dos seus processos do negócio. Neste contexto, não deve ser esquecida a cada vez maior interligação entre o projecto do processo, o projecto do produto e o fabrico (níveis planeamento e execução) em ambientes intra- e inter-empresariais [11]. Assim o exigem novos conceitos e paradigmas como o JIT ("Just-in-Time"), Gestão da Qualidade Total ("Total Quality Management", TQM), melhoria contínua ("Kaizen"), "produção limpa" ("Lean Production"), ou a empresa virtual [8,12,13]. As necessidades de comunicação e coordenação estendem-se a todos os processos, desde o marketing e vendas, passando pelas compras e pela subcontratação e todas as fases de fabrico, até à expedição do produto para o cliente.

Estes requisitos dos anos 90 levaram à intensificação do uso das tecnologias de informação nos modernos sistemas de fabrico e de engenharia. A sua aplicação engloba actividades de projecto e engenharia, de planeamento e controlo da produção, assim como do controlo do equipamento de fabrico e dos sistemas logísticos de distribuição e aprovisionamento. No entanto, a nível industrial, o investimento em tecnologias de informação é ainda menos de um quarto do investimento em outros sectores,

---

[1] (1) CAD - "Computer-aided design", (2) CAP - "Computer-aided planning", (3) CAM - "Computer-aided manufacturing", (4) CAQ - "Computer-aided quality control" (5) CIM/CIME - "Computer Integrated Manufacturing and Engineering"

nomeadamente nas áreas de serviços. As previsões apontam todavia para que a utilização das tecnologias da informação (TI) a nível industrial cresça mais rapidamente que o mercado global das TI, à medida que evoluem as infra-estruturas de suporte às novas estratégias de fabrico [10].

A evolução futura dos Sistemas de Fabrico Integrado por Computador não pode ser prevista com exactidão, dependendo dos mais variados aspectos, tais como o desenvolvimento das tecnologias básicas, i.e. do hardware e do software, a capacidade de adaptação das novas tecnologias computacionais às técnicas de produção (convencionais ou não), a simplificação e estandardização dos processos de produção, etc. Um aspecto importante é certamente a cooperação entre as várias áreas do conhecimento de engenharia necessárias à concepção, projecto e implementação de um sistema integrado de fabrico.

A instalação de um sistema integrado de fabrico deverá trazer ao seu utilizador benefícios consideráveis em comparação com os sistemas tradicionais. Assim sendo, são benefícios típicos: redução dos custos de projecto em 15-30%, redução do tempo de permanência do produto na oficina fabril em 30-60%, aumento de produtividade em 40-70%, produtos de melhor qualidade e redução de não-conformidades em 20-50%, projecto do produto mais adequado (através da simulação de um grande número de variantes), etc.

Estes ganhos esperados deverão ser estimados e controlados em termos da evolução dos valores apresentados pelos indicadores de desempenho adequados, eles próprios factores cruciais de competitividade que determinam a prazo a sobrevivência / reforço da empresa num mercado com exigências crescentes. Porém, estes ganhos só serão atingidos através de uma abordagem integrada e global que contemple as diferentes vertentes de uma empresa industrial (processos de fabrico, gestão e organização de recursos humanos, etc.), e reúna as competências necessárias, internas e externas à empresa, para levar a cabo projectos de mudança.

2.1.3 A contribuição dos projectos cooperativos de I&D no contexto Europeu

A Europa tem continuamente procurado a melhoria e a flexibilização dos sistemas de produção, entendidos aqui como incluindo a globalidade da empresa (processos de fabrico e do negócio) e mesmo as suas relações na cadeia de valor. O mercado em cada um dos diversos países Europeus, e em algum caso mesmo a nível de toda a Europa, torna-se pequeno levando à necessidade de produção de lotes de dimensão cada vez menor, à diversificação de produtos, etc.. Assim, a Europa tem ganho experiência em sistemas flexíveis de produção e tem vindo a desenvolver vários conceitos de integração. Em paralelo, tem sido feito um esforço na utilização de recursos humanos de qualidade e formação crescentes, operários, engenheiros ou gestores, adequadamente integrados no sistema.

A União Europeia tem vindo a investir em I&D na área de CIM desde o início dos anos oitenta, através de diversos programas, nomeadamente o BRITE, EUREKA e CRAFT, mas sobretudo através do programa ESPRIT num dos seus sub-programas ou domínios, denominado primeiro CIM, depois CIME e agora IiM-"Integration in Manufacturing"). Os tópicos principais cobertos desde o ESPRIT-I (1980) até ao ESPRIT-III (1995) abrangeram as mais diversas áreas. Enquanto que no primeiro programa ESPRIT teve particular ênfase a robótica, as redes de comunicação, as infra-estruturas, a automação, assim como a modelação do produto e do processo, o ESPRIT-II centrou-se no utilizador final (produção "human centered"), na gestão e controlo do processo de fabrico, no planeamento e escalonamento assim como na mecatrónica. O programa ESPRIT-III continua por seu lado

dando particular relevância à modelação do produto e do processo, e aposta de uma forma crescente nas novas áreas da empresa virtual, dos sistemas distribuídos, dos sistemas de apoio à decisão inteligentes e da re-engenharia dos processos do negócio, quase abandonando áreas como a robótica, a automação ou as redes industriais. Esta evolução traduz de facto o redireccionar do esforço de investigação no âmbito do ESPRIT nos últimos quinze anos.

Apesar de algumas críticas sobre a rentabilidade dos investimentos de dinheiros públicos em programas europeus de I&D, uma avaliação recente dos resultados do programa ESPRIT conclui ter sido precisamente a área de CIME aquela que maior quantidade de resultados tangíveis e transferências para a indústria produziu [10,11].

Nos últimos anos teve também lugar uma iniciativa exploratória focada nas actividades de I&D relevantes para o fabrico envolvendo pela primeira vez a participação conjunta de europeus, americanos e japoneses, o programa IMS ("Inteligent Manufacturing Systems") [14].

2.2 O CIM na Indústria de Fabrico

2.2.1 As ferramentas informáticas de suporte aos Sistemas Integrados de Fabrico

A engenharia dos sistemas de produção consiste numa abordagem unificadora à tecnologia do fabrico (fluxo de materiais) e à gestão da produção (fluxo de informação) [15]. No dicionário Lello Universal pode ler-se que "o principal agente de produção é o conjunto das forças e meios naturais; à natureza vem juntar-se outro agente, o trabalho; finalmente, as reservas constituídas pelas economias em dinheiro, as máquinas, os instrumentos de trabalho, etc. constituem o terceiro agente de produção, o capital".

Produzir significa criar algo de novo com uma determinada utilidade. A utilidade constitui a unidade de medida do valor económico do produto, seja ele tangível (um objecto) ou intangível (um serviço). O produto surge da transformação das entradas do sistema produtivo, chamadas factores de produção. Se os processos de transformação ou conversão são puramente tecnológicos, então eles são chamados processos de produção. A produção poderá então ser definida como: um processo de produzir bens, incluindo produtos tangíveis e serviços intangíveis, criando desta forma alguma utilidade através de algum valor acrescentado. A produção pode assim ser vista como um sistema entrada / saída [15].

A definição de produção dada engloba dois aspectos: a produção técnica, onde o processo de conversão usa os meios de produção para fornecer um produto / serviço; a produção económica, onde existe a criação de um produto com um valor superior ao dos elementos de entrada. Do ponto de vista da empresa de fabrico, a produção é entendida como a transformação de matéria-prima em produtos através de um conjunto de aplicações de energia, cada uma das quais afecta do ponto de vista físico ou químico as características dos materiais.

O significado original do termo manufactura utilizado geralmente para significar fabrico era o de fazer as coisas à mão. No entanto, o seu significado actual traduz-se melhor pela "conversão de um projecto num produto acabado", referindo-se o fabrico ao acto físico de realizar o produto. Assim sendo, a empresa de fabrico é caracterizada por um conjunto

de actividades inter-relacionadas e operações envolvendo o projecto, selecção de materiais, planeamento, produção, controlo da qualidade, gestão, e marketing.

As actividades de um sistema de Fabrico Integrado por Computador (CIM) incluem não só o fabrico assistido por computador, ou CAM ("Computer Aided Manufacturing"), e o projecto assistido por computador, ou CAD ("Computer-Aided Design") mas ainda o planeamento / gestão da produção assistida por computador (CAP, "Computer Aided Planning").

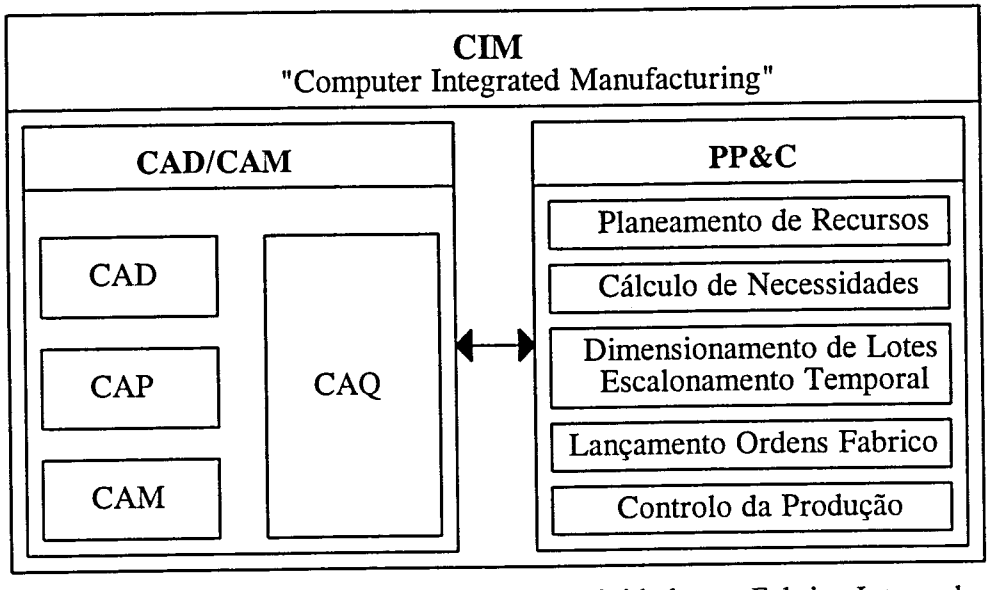


Figura 2.3 - O Suporte Informático das Principais Actividades no Fabrico Integrado por Computador

A figura 2.3 apresenta de forma muito simplificada como se relacionam as diversas actividades fabris hoje em dia suportadas por sistemas informáticos:

- O CAD ("Computer-Aided Design") é a actividade que engloba o suporte por computador do projecto do produto e do processo de fabrico, assim como dos cálculos de engenharia. Uma vez que a engenharia está também envolvida no desenvolvimento e teste de produtos, geração de programas de controlo numérico (NC) e outras actividades apoiadas por computador, o termo CAE ("Computer-Aided Engineering") também é muitas vezes utilizado com um significado mais abrangente.
- O CAP ("Computer-Aided Planning") inclui, além do cálculo de recursos e materiais, as actividades de geração do "dossier de fabrico" para a realização do produto na linha de produção. Este plano descreve os processos de fabrico, gamas operatórias, percursos e as sequências relacionadas com a realização do produto.
- O CAM ("Computer-Aided Manufacturing") engloba definição das funções do computador no controlo do sistema de produção na instalação fabril, incluindo o controlo directo do equipamento de produção e a gestão de materiais, ferramentas, acessórios. Em alguns casos são ainda incluídas funções de controlo da qualidade e de gestão da manutenção relacionadas com cada equipamento específico.
- O CAQ ("Computer-Aided Quality Control / Assurance") combina todas as actividades de controlo da qualidade no sistema de fabrico. Em alguns casos também é utilizado o termo CAT ("Computer-Aided Testing"), o qual tem um

significado bastante mais restritivo dado limitar-se aos aspectos de inspecção automática.

- O CAD/CAM designa as actividades conjugadas do CAD e CAM e frequentemente funcionalidades de CAQ.
- O PP&C ("Production Planning and Control") é uma actividade organizacional, abrangendo um conjunto de funções tais como: o planeamento dos recursos de produção, o cálculo de necessidades, o faseamento temporal das operações, a emissão e lançamento de ordens de fabrico e o controlo da produção.
- O conceito de CIM / CIME combina todas as actividades anteriores, integrando os requisitos do processamento de informação técnica e operacional da empresa industrial. Uma apresentação mais detalhada dessas actividades e do suporte informático respectivo é feita na figura 2.4 através do denominado modelo Y de Scheer [16]. As tarefas de gestão das operações fabris são estruturadas na cadeia de Planeamento e Controlo da Produção (PP&C) representada no braço esquerdo do Y. As actividades de projecto e engenharia, caracterizadas e apoiadas pelos vários conceitos de CAX, surgem no braço direito do Y. A integração destas áreas coloca às empresas exigências particularmente elevadas ao nível da organização, constituindo ainda um desafio aos produtores de hardware e de software que deverão coordenar os seus esforços no desenvolvimento de novos sistemas e componentes, e aos consultores e empresas de serviços responsáveis pela manutenção, actualização e integração dos sistemas de fabrico.

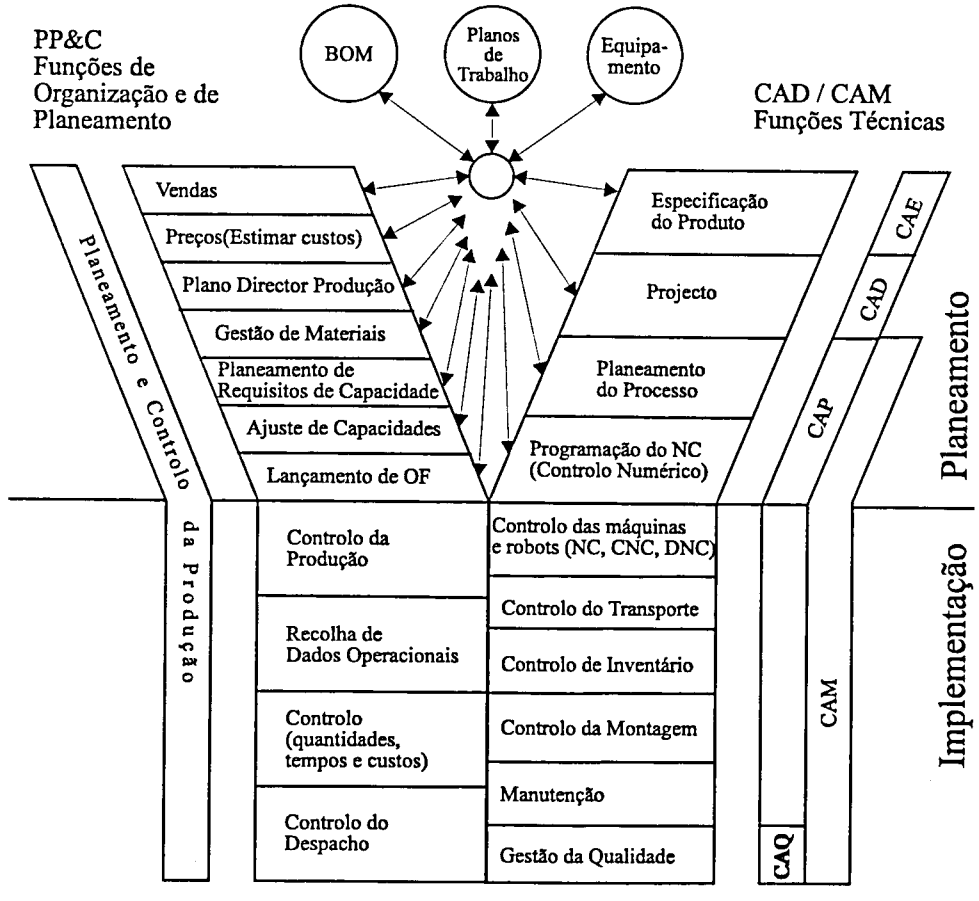


Figura 2.4 - Modelo em Y de Scheer [16]

2.2.2 Conceitos básicos: factores de produção e produtividade

Um sistema fabril pode ser visto como um sistema entrada / saída (figura 2.5) [15]. As entradas do sistema de fabrico, os factores de produção, podem ser divididos em quatro categorias, cada uma delas desempenhando um papel essencial: os objectos de produção, o trabalho produtivo, os meios de produção e a informação. Os objectos de produção são aqueles sobre os quais são executadas as actividades de produção: os materiais primários são convertidos em produtos durante o processo de produção (e.g. matéria prima e componentes); os materiais auxiliares são adicionados aos primários, tais como tintas, óleos ou electricidade consumidos durante o fabrico. O trabalho produtivo consiste na utilização das capacidades humanas, sendo o esforço físico e/ou mental colocado em jogo por um operador durante a actividade de produção. Os meios de produção são os meios através dos quais os objectos de produção são convertidos em produtos, podendo ser classificados em meios directos ou indirectos. Os meios directos de produção actuam directamente, realizando transformações, sobre a matéria-prima (máquinas, equipamento, ferramentas, etc.), enquanto que os meios indirectos não actuam directamente no processo produtivo (terrenos, estradas, edifícios, armazéns, etc.). De acordo com [15], a informação de produção consiste nos métodos de produção, constituídos pelos procedimentos técnicos usados na implementação eficiente e económica dos processos de fabrico.

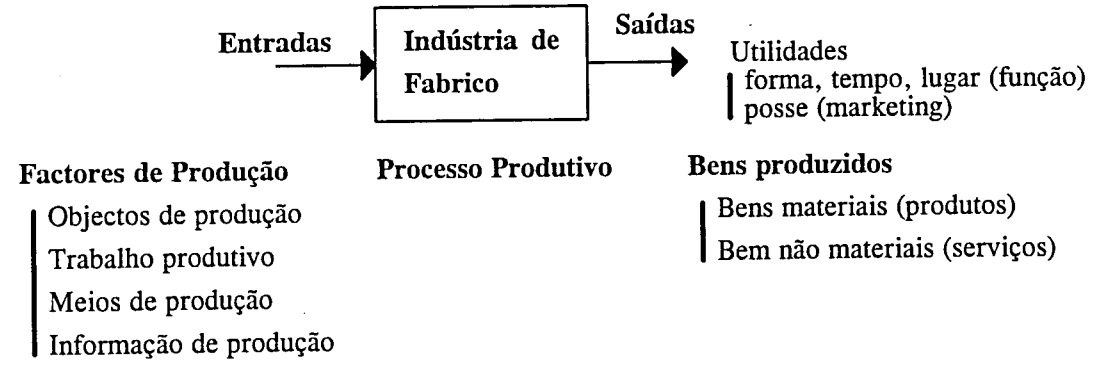


Figura 2.5 - A Indústria de Fabrico [15]

A saída do processo de produção corresponde à criação de uma utilidade de forma, tempo, lugar e posse com um determinado valor. Como exemplo, temos a actividade de fabrico que fornece uma utilidade de forma, enquanto que a distribuição física dos produtos fornece uma utilidade de lugar. A utilidade de tempo é por exemplo dada através do serviço telefónico. Finalmente, a utilidade de posse traduz a satisfação relativa do utilizador derivada do uso do produto. Nesta sequência de raciocínio, podemos chegar à situação onde a utilidade pode não ser positiva, como por exemplo no caso de subprodutos do processo produtivo que podem ser nefastos para o ambiente. Por este motivo, deve ser evitada a produção de "não-utilidades" traduzidas na super-produção ou na produção de coisas sem utilidade (i.e. produto não-conforme).

O fabrico integrado por computador está essencialmente concentrado nas indústrias de fabrico, i.e. sector secundário, cujas saídas são produtos. O valor dos produtos é avaliado através de três aspectos: a função e qualidade, o custo de produção e preço, a quantidade e o tempo de produção. O objectivo primário de uma indústria de fabrico é realizar produtos na quantidade, com a função e qualidade desejadas, rapidamente e ao menor custo possível. A gestão da produção é a actividade da empresa responsável pelo fornecimento de um plano adequado para a produção eficiente. Cada um dos aspectos referidos está relacionado

respectivamente com técnicas de controlo da qualidade, controlo de custos e controlo da produção (e / ou do processo de fabrico). Por controlo do processo de fabrico entende-se o controlo das máquinas e equipamentos inerentes ao processo físico propriamente dito, enquanto que o controlo da produção se refere ao âmbito mais alargado da gestão e cálculo dos materiais bem como da sua logística.

O processo de conversão dos factores de produção, em especial de matérias primas em produtos, é chamado o processo de produção ou processo de fabrico. O processo de produção é geralmente constituído por sucessivos estados de produção, sendo as diversas operações executadas sucessivamente em várias estações de trabalho. O fornecimento das diversas utilidades, de forma, tempo e lugar, ocorre pois sucessivamente ao longo do processo produtivo. Do ponto de vista global deve notar-se que, para exploração óptima dos recursos de fabrico, vários produtos são lançados em produção em simultâneo e ocorrem em paralelo operações sobre componentes ou materiais posteriormente agregados em processos de montagem.

A medida da eficiência do processo de produção é chamada produtividade, definida como a razão entre as entradas e as saídas. O conteúdo e dimensões das entradas / saídas levam ao aparecimento de vários indicadores: a produtividade física, onde as saídas são medidas em quantidades; o valor (económico) da produtividade, onde as saídas são medidas em valor monetário; e o factor de produtividade, onde a produtividade é calculada em relação aos factores de produção indirectos tais como a produtividade do trabalho, do terreno, etc. A produtividade total está por seu lado relacionada com todos os factores de produção, sendo uma medida global que representa a eficiência da empresa de fabrico.

A produtividade como indicador de desempenho global depende de parâmetros medidos através de indicadores de desempenho ao nível do fabrico (tais como tempos de ciclo, custos da não-qualidade, rendimentos de operações), e de serviço (tais como os tempos de resposta a encomendas).

2.2.3 Vistas estruturantes sobre os Sistemas de Fabrico

Falamos em produção como um sistema entrada/saída. Por sistema entende-se um todo organizado, onde o efeito sinérgico leva a que o seu desempenho total seja superior à soma dos desempenhos parciais dos seus subsistemas. O conceito de sistema de fabrico foi criado durante a revolução industrial em Inglaterra há cerca de duzentos anos. No início deste século, Taylor (1911) apresentou a sua visão da gestão do sistema de fabrico [17].

Os sistemas de fabrico podem ser descritos pelos três aspectos que os caracterizam: o estrutural (ou estático), onde o sistema é considerado como um conjunto de unidades mais simples reconhecíveis assim como pelas ligações entre essas mesmas unidades; o transformacional (ou funcional), onde as entradas são as influências exercidas pelo exterior no sistema (incluindo possíveis perturbações), e as saídas a forma como o sistema influencia o seu exterior enquanto que procura maximizar a produtividade da transformação; o procedimental (ou dinâmico), onde o sistema consiste num determinado conjunto de estados relacionados para cada um dos quais existe uma tarefa específica a executar.

O aspecto estrutural do sistema de fabrico descreve-o como um conjunto unificado de recursos, incluindo trabalhadores, máquinas, ferramentas de produção, equipamento de manuseamento de material e outros aparelhos suplementares. O aspecto estrutural do sistema forma assim a estrutura estática espacial da instalação fabril, isto é a sua planta ou "layout". O padrão de planta da instalação fabril é escolhido tendo por base o número de

produtos P e a quantidade a produzir Q, conduzindo às seguintes distribuições espaciais típicas:

- *"layout" orientado ao produto* ("flow-line" ou "production line") se Q/P for elevado, caso típico de produção em massa. Os recursos de produção assim como os serviços auxiliares estão colocados de acordo com a denominada linha de produção.
- *"layout" orientado ao processo* se Q/P for baixo, ou seja, se a produção for de pequenos lotes, as máquinas do mesmo tipo estão agrupadas em centros de trabalho numa determinada área da fábrica.
- *tecnologia de grupo* o *"layout" celular* é usado para os valores médios de Q/P, onde uma grande variedade de produtos pode ser agrupada em diversas famílias. Estas famílias são assim produzidas em lotes de produtos similares e as máquinas são agrupadas para se ajustarem a este tipo de produção.

O aspecto transformacional (ou funcional) do sistema de fabrico é definido como o processo de transformação/conversão dos factores de produção, em particular da matéria-prima em produtos acabados, tendo como objectivo a máxima produtividade. Este aspecto apresenta a vista do fluxo de materiais do sistema produtivo.

O aspecto procedimental do sistema de fabrico baseia-se na descrição procedimental do sistema, constituindo o chamado ciclo ou cadeia de gestão da produção: o *planeamento* que consiste na selecção, entre várias alternativas, das linhas de acção futuras; a *implementação* consistindo na execução das actividades de acordo com o plano estabelecido; e o *controlo* que realiza a medida e as correcções do desempenho das actividades por forma a assegurar que os objectivos do plano estão a ser atingidos. Deste modo, o sistema de fabrico prepara planos, implementa as actividades produtivas para a conversão de matéria prima em produtos, e controla o processo com o objectivo de reduzir ou eliminar os desvios entre o desempenho efectivo do sistema e o planeado. A este procedimento chama-se *gestão da produção* e consiste nos cinco passos seguintes:

- o *planeamento agregado da produção*, que determina os tipos de produtos e as quantidades a serem produzidas durante períodos especificados;
- o *planeamento do processo de produção*, que estabelece os processos de produção (existem diversas sequências) que transformam a matéria-prima em produtos acabados;
- o *escalonamento da produção*, que fixa o processo de produção final e define o faseamento temporal para cada tarefa contida no processo de produção, indicando quando ela é executada sobre que lote de matéria prima, componente ou "em-curso-de-fabrico", em que máquina e por que operador;
- a *implementação da produção*, responsável pela efectiva produção de acordo com o plano estabelecido;
- o *controlo da produção*, que procura detectar situações em que existem desvios entre o escalonamento de tarefas planeado e o andamento real das operações, actuando para os corrigir.

Este ciclo de gestão traduz-se de facto no fluxo de informação dos denominados níveis táctico e operacional necessário à produção eficiente. A um nível mais elevado surge o planeamento estratégico, função responsável pelo tratamento de problemas estratégicos existentes entre o sistema de fabrico e o ambiente em que ele se insere. Realiza deste modo o planeamento de longo prazo, os planos de marketing e vendas, a política de preços do produtos, etc..

Observando em mais detalhe o procedimento de fabrico, podemos dividi-lo na fase de projecto do produto, fase de planeamento de processo e fase de implementação da produção (figura 2.6, [15]) três fases correspondentes ao braço direito do modelo em Y de Scheer e realizadas respectivamente segundo horizontes temporais de longo / médio (estratégico) médio / curto (táctico) e curto (operacional) prazo. Na fase de projecto são primeiro realizadas as especificações técnicas a que o produto deve obedecer para satisfazer as necessidades do mercado, decididas como resultado de uma actividade de investigação e desenvolvimento, de transferência de tecnologia ou de engenharia, seguindo-se o seu projecto que incluí a definição das suas funções e dos aspectos ergonómicos. Surge então o planeamento do processo com o objectivo de realizar os produtos de uma forma eficiente e económica, sendo decidido o encaminhamento ("routing") do processo de fabrico e as máquinas, bem como a selecção das ferramentas e dos suportes de material. No processo de implementação da produção, é recebida a matéria-prima, as peças são maquinadas e os produtos que o exijam são montados e/ou inspeccionados. Finalmente, os produtos acabados serão expedidos e distribuídos. Torna-se deste modo evidente a importância de uma abordagem do sistema como um todo.

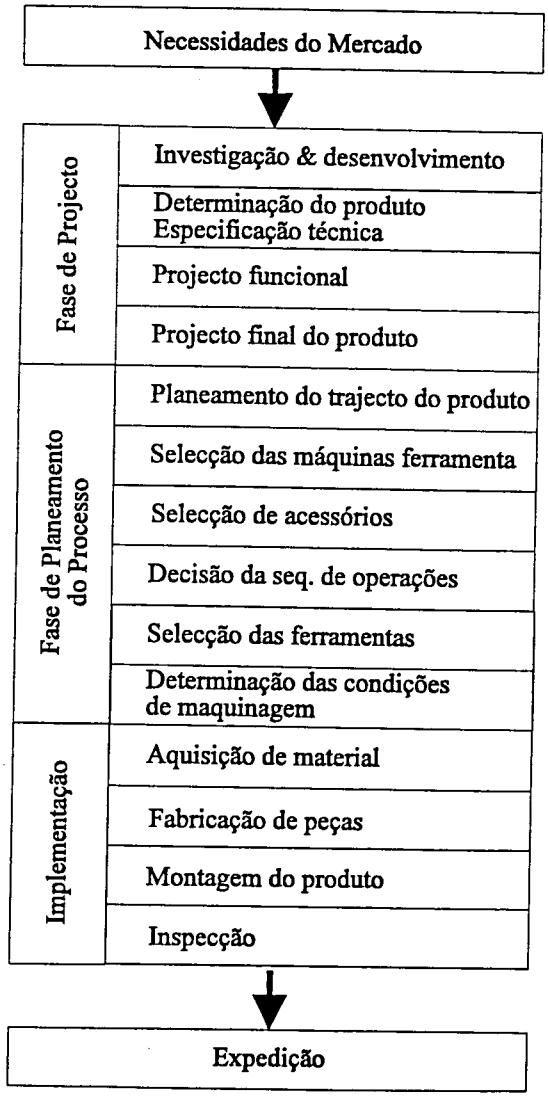


Figura 2.6 - Funções no Sistema Integrado de Fabrico (exemplo de um processo discreto)

Na Engenharia dos Sistemas de Fabrico a unificação completa do *fluxo de materiais* (processo de fabrico) e do *fluxo de informação* (gestão da produção), é da maior importância para a execução eficiente e económica das actividades de produção. Esta abordagem unificada e integrada, chamada Engenharia dos Sistemas de Fabrico evidência os quatro aspectos seguintes: (1) clarifica o conceito de sistema de fabrico bem como as suas funções e estruturas básicas, incluindo o problema do projecto do produto e processo e em particular do fluxo de materiais; (2) trata da optimização do sistema de fabrico, com particular relevo para a decisão óptima ao nível da produção; (3) trata o controlo e a automação do sistema de fabrico; (4) aborda o processamento da informação de produção para os sistemas de fabrico dentro de janelas temporais bem definidas.

2.2.4 Problemas de Decisão nos Sistemas de Fabrico

Nos anos 80, havia produção para um mercado assegurado, i.e. a empresa de fabrico produzia um produto "já vendido". Hoje em dia este cenário mudou e as empresas têm que se certificar de que o mercado existe antes de fabricar. De facto, hoje em dia o mercado tomou o poder que anteriormente pertencia à empresa de fabrico sendo ela agora controlada pelos desejos e pelo tempo do cliente. É assim da maior importância que os departamentos da empresa de fabrico responsáveis pelo projecto, planeamento e gestão da produção disponham dos meios adequados à resolução dos problemas de decisão em que estão envolvidos. Nos parágrafos que se seguem serão abordados alguns dos problemas de decisão que ocorrem na empresa de fabrico ao nível do planeamento agregado, do planeamento do processo de produção, do escalonamento da produção e do controlo da produção.

Problemas de decisão no planeamento agregado

A combinação óptima de produtos determina a combinação adequada dos tipos de produtos que podem ser produzidos com a capacidade existente. Se a capacidade não for suficiente, um determinado conjunto de pedidos não poderá ser satisfeito. Neste caso deve então ser escolhida a combinação de produtos óptima para a produção (i.e. quais os tipos de produtos e quais as quantidades a serem produzidas). Algumas das políticas usadas procuram por exemplo maximizar o lucro total obtido pela produção face às restrições, e em outros casos procura-se maximizar o cumprimento dos prazos de entrega das encomendadas com eventual sacrifício dos lucros.

A análise de requisitos (cálculo de necessidades) determina as quantidades de materiais a serem encomendadas / produzidas para cada produto durante um determinado período. Técnicas quantitativas da programação matemática, programação linear, etc., são uma ajuda preciosa na resolução destes problemas. Na prática, este problema é resolvido por pacotes de software MRP / MRP-II ("Material Requirements Planning" / "Material and Resource Planning") embora tipicamente de forma deficiente (geralmente pressupondo capacidade infinita e tempos de transporte nulos, e ignorando o estado actual da instalação fabril, pelo que é muitas vezes inútil o planeamento fornecido por um MRP-II).

A análise da dimensão dos lotes é pertinente em situações de produção intermitente, nos casos em que o ritmo de consumo é baixo comparado com o ritmo de produção. Nesta situação, o produto será produzido periodicamente e numa quantidade tal que satisfaça as necessidades para um determinado período de tempo. Assim sendo, a decisão da produção (dimensão económica do lote e ciclo de produção óptimo) é tomada tentando minimizar os

custos totais de produção tomados como a soma dos custos de preparação, fabrico e armazenamento.

O nível de produção uniforme, bem como a sua adaptação às flutuações das exigências do mercado, é da maior importância para que se maximize a utilização da instalação fabril. As políticas de gestão para atingir estes objectivos são: (1) adaptação temporal, absorvendo a flutuação dos pedidos através do aumento ou diminuição dos tempos de produção; (2) adaptação dos stocks, absorvendo a flutuação dos pedidos através do controlo do nível de stocks; (3) adaptação intensiva, absorvendo as flutuações de pedidos através do aumento ou diminuição da intensidade da produção; (4) adaptação quantitativa, absorvendo a flutuação dos pedidos através da subcontratação das quantidades de produção que não podem ser fabricadas com a capacidade instalada.

A complexidade dos problemas é agravada com perturbações internas (tais como avarias de máquinas e absentismo) ou externas (tais como encomendas tardias urgentes, fornecimentos atrasados levando à rotura de stocks), exigindo uma constante adaptação às realidades que exige a flexibilização destas políticas tornando os problemas de decisão de difícil resolução.

Problemas de decisão no planeamento do processo de produção

O *planeamento do processo* tem como objectivo obter o fluxo de materiais óptimo nas operações de conversão da matéria-prima no produto acabado. Esta função inclui respectivamente o projecto do processo e o projecto das operações. O projecto do processo realiza a análise do fluxo de materiais e selecciona a estação de trabalho adequada onde cada operação deve ser executada. Os caminhos óptimos, minimizando o tempo e o custo total da produção, poderão ser determinados usando técnicas de programação dinâmica. O projecto da operação estabelece o tipo de operação (o seu conteúdo e método) a ser executada para cada um dos processos de produção. Neste contexto, procura-se reduzir os tempos de inactividade das máquinas assim como dos operadores.

Ao nível do *planeamento do "layout" da instalação fabril* reside a preocupação com a distribuição espacial dos recursos de produção de forma a permitir o melhor fluxo de materiais. A distribuição dos recursos na instalação fabril ("layout") depende principalmente dos tipos de produtos e das suas quantidades. O planeamento de "layouts" é normalmente realizado usando um conjunto de heurísticas, sendo o melhor seleccionado entre um conjunto de várias alternativas e podendo ser utilizadas ferramentas de simulação na análise comparativa.

Na *análise das condições óptimas de fabrico*, procura-se que o sistema de fabrico funcione de uma forma eficiente e económica, sendo necessário que seja determinado qual o seu ponto óptimo de funcionamento. Esta análise será feita tanto para os sistemas com uma única fase de produção como para os que possuem várias fases, segundo o critério da máxima taxa de produção, mínimo custo de produção e máxima taxa de lucro.

Problemas de decisão no escalonamento da produção

O problema de escalonamento é um problema de afectação a cinco dimensões, ou seja deverá ser afectada a operação correcta à máquina correcta e operador correcto, fazendo chegar a tempo os materiais correctos e na sequência correcta.

O sequenciamento de tarefas determina a ordem pela qual as diversas tarefas são executadas numa máquina, célula ou linha. O escalonamento das operações determina essa ordem e estabelece o plano temporal para o processamento das diversas tarefas nas várias máquinas, podendo esta afectação ser apresentada sob a forma de um gráfico de Gantt.

Numa linha de montagem por exemplo, as várias estações de trabalho encontram-se normalmente dispostas ao longo do seu comprimento. Deseja-se assim um fluxo de materiais uniforme, procurando equalizar os tempos das diversas operações realizadas nas estações de trabalho ao longo da linha. Deste modo são minimizados os tempos de inactividade das diversas estações de trabalho. O equilíbrio das linhas de produção procura atingir o mínimo tempo de ciclo ("cycle-time"), o mínimo número das estações de trabalho, e o melhor agrupamento dos elementos de trabalho.

Num escalonamento de longo-prazo, tal como a construção de uma nova fábrica, de um navio, etc., não são significativos períodos muito curtos tais como segundos ou minutos. Nestas situações, o escalonamento de um projecto a longo prazo é realizado usando técnicas tais como PERTs ("Program Evaluation and Review Technique"), CPM ("Critical Path Method") ou ainda o GERT ("Graphical Evaluation and Review Technique"), suportadas por pacotes de software standard.

Problemas de decisão no nível da execução das operações

Na *automação flexível*, um dos problemas mais importantes actualmente é a implementação de sistemas de produção flexível, i.e. produção de múltiplos produtos em pequenos lotes em oposição à produção de um produto único. Para atingir este objectivo, é necessária a disponibilidade de equipamento de produção muito flexível capaz de realizar automaticamente diferentes tarefas: (1) maquinagem, através centros de maquinagem capazes de trocar automaticamente entre si conjuntos de ferramentas, e de realizar múltiplas operações de fabrico; (2) carga e descarga de material, através de um braço mecânico ou robô, podendo ainda existir um sistema de "palettes" para o transporte da peça trabalhada; (3) transportadores ("transfer-lines"), usados no transporte da matéria-prima do armazém para a produção, entre fases da produção, ou entre a produção e o armazém de produtos acabados, podendo ser utilizados passadeiras rolantes, ou veículos filo-guiados, conhecidos por AGVs ("Automated Guided Vehicle"); (4) armazenamento automático, de matéria-prima, peças, stock do em-curso-de-fabrico e produtos acabados, podendo as "palettes" e as passadeiras rolantes ser usados como armazenamento temporário ("buffers") entre fases da produção.

A produção automatizada pode então ser construída usando máquinas-ferramenta muito flexíveis bem como sistemas automáticos de manuseamento de materiais, sendo o conjunto controlado e coordenado por um sistema computorizado. Este todo é designado por Sistema de Fabrico Flexível (FMS - "Flexible Manufacturing Systems"). Um FMS tem duas funções: a primeira é a função de produzir, i.e. trabalhar as peças com diversas formas e com vários processos de maquinagem de uma forma contínua de acordo com instruções de maquinagem recebidas de um computador (sistema CAD / CAM), sendo depois descarregadas da máquina e transferidas para o centro de maquinagem seguinte ou para a saída. A outra função é a do processamento da informação. Os dados/informação para as diversas instruções de fabrico estão armazenados, gerados, transmitidos e controlados pelo computador. O armazenamento das peças em produção assim como a troca de ferramentas devem ser executadas quando necessário. Assim sendo, um FMS é um sistema automático de fabrico que controla tanto o fluxo de materiais como o fluxo de informação em tempo-real.

Problemas de decisão no controlo da produção

O controlo exige necessariamente a monitorização do estado do sistema a controlar. Tratando-se do sistema de produção esta função permite não só a realimentação de

informação vinda da instalação fabril, mas ainda a implementação de acções de correcção sempre que seja detectados desvios entre o planeado e o realmente executado.

O *controlo da logística* intra-empresarial é a função de controlo do fluxo de materiais, que inclui: (1) o *controlo das compras*, ou seja da aquisição dos factores de produção; (2) o *controlo da produção*, com o controlo dos tempos (e datas de entrega) e das quantidades a serem produzidas; (3) o *controlo da qualidade*, i.e. o controlo e garantia do nível desejado da qualidade dos produtos acabados; (4) o *controlo das vendas*, i.e. do grau de execução dos objectivos das vendas; (5) o *controlo de stock*, i.e. controlo do excesso / insuficiência de stocks.

O *controlo dos recursos de produção* controla os factores de produção, incluindo o controlo dos recursos (pessoas e máquinas) e dos custos da produção.

2.3 O Sistema de Fabrico

Nos parágrafos que se seguem é apresentado de uma forma breve o sistema de fabrico, sendo descritas as diversas unidades funcionais existentes num sistema de fabrico discreto. O texto é uma visão pessoal mas inclui algumas ideias estruturantes extraídas da bibliografia (particularmente Hitomi [15], Scheer [16] e Storr [18]).

2.3.1 Introdução

Num sistema de fabrico integrado por computador são combinadas funções fundamentais que deverão ser executadas em conjunto de uma forma eficiente e coordenada. Na figura 2.7 estas funções são apresentadas formando um todo coerente, sendo identificados os múltiplos fluxos de informação tecnológica e de gestão, assim como os fluxos de materiais e financeiros inerentes [15,18]. Nos parágrafos que se seguem é abordado de uma forma sistemática o papel de cada uma das funções referidas, bem como as interacções entre elas.

Um sistema integrado de fabrico é constituído por um conjunto de funções interligadas através de um complexo sistema de comunicações. Este conjunto integrado de funções, inclui o suporte ao planeamento estratégico, planeamento organizacional, escalonamento e controlo da produção e contabilidade. Neste sistema, os fluxos de informação, financeiros e de materiais deverão ser controlados de uma forma integrada por forma a melhorar o desempenho da empresa na sua função de servir o mercado de consumidores com um produto de alta qualidade e desta forma assegurar a sua viabilidade económica.

A vista global sobre o sistema de fabrico é apresentada na figura 2.7, acompanhada de duas escalas horizontais indicando respectivamente a forma como as diferentes actividades se agrupam na cadeia / ciclo de planeamento e controlo, e o horizonte temporal em que se inserem.

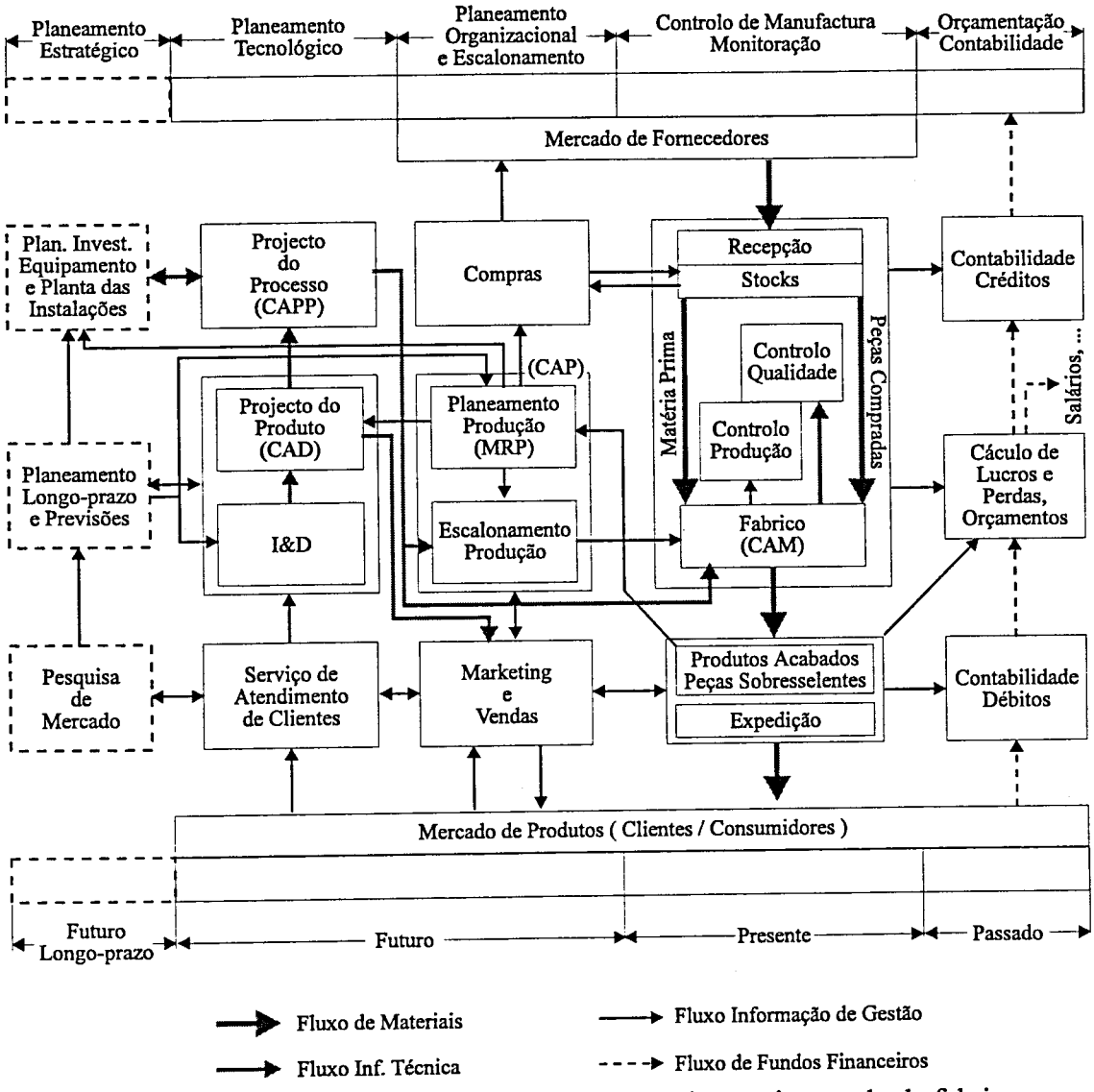


Figura 2.7 - Interacção entre as funções básicas num sistema integrado de fabrico para produção de lotes da pequena e média dimensão

2.3.2 Planeamento Estratégico

O conhecimento do mercado potencial, do desenvolvimento demográfico, das tendências técnicas e comerciais, das inovações nos produtos, do futuro fornecimento de recursos, da disponibilidade de mão de obra, das condições financeiras e da sua competitividade é essencial à empresa. A pesquisa de mercado é uma função básica responsável pela recolha de informação disponível em publicações e pela consulta a especialistas. Esta informação é então avaliada com métodos matemáticos e científicos de previsão, do âmbito da estatística e da teoria de decisão. Os resultados deste trabalho são utilizados nas previsões e no planeamento de longo prazo necessários à tomada de decisões estratégicas. Esta função está intimamente ligada aos serviços de recepção de encomendas e de marketing, na avaliação de novos métodos de venda e tratamento das queixas e anseios dos consumidores.

No planeamento de longo prazo é determinado o produto ou a linha de produtos a ser comercializada. A empresa deverá saber qual a taxa de produção anual, as características do produto, a localização e a disposição física da área de produção, assim como o tipo de

equipamento de fabrico exigido. A taxa de produção vai ainda determinar o método de fabrico, os processos, o tipo de material a usar e os procedimentos para o controlo da qualidade. Se a decisão for a de equipar a produção com processos renovados de fabrico, devem ser encontradas estratégias para que a introdução do novo equipamento não interfira com a produção em curso. Quando é realizado o planeamento de uma nova instalação fabril, devem ser tomados em consideração: a localização, os caminhos de transporte óptimos, a existência de mão de obra, o fornecimento de matéria prima e infra-estruturas de baixo custo. O efeito das previsões será sentido em todas as funções da empresa, motivo pelo qual uma tarefa desta envergadura deverá envolver todos os departamentos da empresa.

A actividade de planeamento dos investimentos em equipamento e infra-estruturas é responsável pela selecção dos equipamentos e do software a ser instalado. Com este objectivo, é estudada a relevância de cada recurso de fabrico, de transporte ou zona de armazenamento para o sistema produtivo. São realizados planos para a disposição dos recursos na instalação fabril, e para a incorporação de equipamento novo e antigo tendo como pano de fundo um determinado conjunto de estratégias de controlo. Esta actividade de planeamento e de avaliação de alternativas pode ser fortemente apoiada por sistemas periciais e pela simulação da produção para cenários alternativos para os quais é avaliado o retorno do investimento. Este planeamento da melhor configuração para a produção ("layout planning"), poderá ainda levar em consideração a possibilidade de futuras alterações no tipo de produtos. O resultado do planeamento deverá apresentar a configuração da zona de fabrico, as máquinas e equipamentos necessários, o equipamento auxiliar, os sistemas de transporte, o sistema computacional de controlo, os programas de controlo e de planeamento, a rede de comunicações na fábrica, o equipamento necessário ao controlo da qualidade, e ainda o resultado de estimativas de cálculo do retorno do investimento para os diferentes cenários possíveis.

2.3.3 Planeamento Tecnológico

O serviço de atendimento de clientes constitui a interface entre o cliente e a sua ordem de fabrico, devendo permitir o seu seguimento ao longo do fabrico e a consequente informação sobre o estado da sua encomenda. Através deste serviço, o fabricante pode introduzir as alterações de engenharia (especificação) e ajustar as quantidades e datas de entrega de acordo com os desejos do cliente. Tem particular interesse para o cliente a forma como a sua encomenda está a progredir na fábrica, identificando atrasos devidos a problemas de estrangulamento na produção ou falta de peças ou matéria prima. A expedição, embora possa ser realizada para qualquer actividade de fabrico, normalmente interactua com o planeamento e processamento das ordens de fabrico e com o controlo de produção.

A engenharia (responsável pela I&D e pelo projecto do produto) é uma das actividade chave do sistema de fabrico. É a engenharia que determina a função e o projecto de um produto, tendo deste modo uma influência crucial sobre o processo de fabrico a ser seleccionado. Uma vez completa a engenharia do produto, cerca de 70% do custo da sua produção está desde logo fixado. Por este motivo, a cooperação entre a engenharia e a produção deve ser a mais próxima e eficaz possível, tendo levado a crucial importância deste ponto à emergência dos novos conceitos de engenharia simultânea. A engenharia apoia o seu trabalho em documentação escrita em livros, catálogos, standards, bases de dados de produtos, programas de simulação e de CAD. Importantes fontes de conhecimento para a engenharia vêm do marketing, do controlo da qualidade, e do serviço de apoio ao cliente. Estas actividades exigem uma monitorização constante na busca de novas ideias

para os produtos e na resposta a queixas dos clientes. A engenharia concebe, projecta e testa os produtos. Para a produção de produtos em massa, a engenharia também é responsável pelo teste piloto da execução da sua produção. Os documentos mais importantes produzidos pelos responsáveis da engenharia são: os desenhos descrevendo o produto, as suas dimensões, tolerâncias, materiais, acabamentos e procedimentos de controlo da qualidade; a lista de materiais (BOM, "Bill Of Materials"), descrevendo a estrutura dos produto e dos seus componentes. Usando sistemas avançados de apoio à engenharia (CAE), é possível a produção automática de documentos da responsabilidade do planeamento do processo de fabrico, tais como preparação das folhas de processo (gamas operatórias), com a descrição do processo e sequências de fabrico e a geração automática dos programas de controlo numérico.

O planeamento do processo (construção das gamas operatórias do produto) de fabrico utiliza os desenhos das peças a serem fabricadas produzidos pela engenharia, a sua constituição em termos de materiais e componentes necessários (BOM) bem como o número de peças encomendadas, e determina os processos de fabrico necessários à produção das peças em questão assim como a sua sequência, com a eventual definição de variantes. Os resultados deste planeamento (onde são estabelecidos os processos de transformação / maquinagem, as sequências, os parâmetros de cada operação, as máquinas, as ferramentas, os acessórios e os tempos e custos de fabrico) são elaborados recorrendo a informação sobre: recursos de fabrico (máquinas, ferramentas e acessórios); processos de fabrico e gamas operatórias disponíveis; lista de variantes de produtos; parâmetros de operação; tempos de operação. Os processos seleccionados dependem muito do tamanho da peça, do seu projecto, do material utilizado e do trabalho necessário. Um factor importante é a quantidade de trabalho exigida pela peça. Se ele for elevado, será tentada a utilização de máquinas mais sofisticadas para assim minimizar o seu manuseamento e transporte. Em sistemas mais sofisticados, o responsável por esta operação poderá introduzir um código que caracterize a forma da peça a ser transformada, obtendo como resposta uma sugestão para o processo de fabrico. Esta actividade poderá ainda fornecer os programas de comando numérico para a execução das operações de fabrico, caso eles não tenham sido produzidos pela engenharia.

2.3.4 Planeamento Organizacional e Escalonamento

O marketing é uma actividade orientada para o consumidor, e procura estabelecer e melhorar o posicionamento da empresa no mercado competitivo. O marketing tenta deste modo servir as necessidades e desejos do cliente, tendo uma influência decisiva na determinação do produto ou linha de produtos, e das suas características. Como resultado do contacto próximo com os clientes, o marketing conhece as suas preferências, queixas e conceitos de qualidade.

No contexto de uma empresa industrial, o objectivo é a venda dos seus produtos pela qual o marketing é responsável. São os lucros provenientes das vendas que permitem a viabilidade económica da empresa. Assim sendo, são as vendas que estabelecem a combinação de modelos dos produtos a serem produzidos, as estratégias de preços, a especificação dos produtos e a sua qualidade. O departamento de vendas e o serviço de atendimento de clientes são a interface entre o mercado e a engenharia, o planeamento, a produção e o controlo da qualidade.

Quando uma ordem é enviada para o fabrico, ela vai competir pela utilização dos recursos de fabrico com um elevado número de outras ordens por forma a que os seus

prazos de entrega sejam respeitados. Por este motivo, são invocados procedimentos complexos de escalonamento de ordens de fabrico de forma a que seja uniformizada a utilização dos recursos e a que sejam cumpridos os prazos de entrega. Tentar-se-á reduzir os tempos de preparação das máquinas e de manuseamento e transporte das peças, procurando que em cada máquina seja realizado o número máximo de operações. Além disso, os algoritmos de escalonamento podem ainda tentar minimizar o transporte de material entre duas operações através da selecção de recursos próximos para operações consecutivas. O escalonamento compreende assim funções cruciais de dimensionamento de lotes, sequenciamento e afectação de recursos.

O escalonamento da produção consiste nas três seguintes operações: o faseamento temporal, que garante que as peças são todas produzidas dentro dos prazos de entrega estabelecidos; o escalonamento de recursos, que procura uma utilização uniforme de recursos evitando grandes filas de espera e estrangulamentos; a distribuição de material, que assegura que todas as peças e materiais necessários estão no posto de trabalho certo na altura certa.

Quando uma determinada peça não pode ser produzida dentro dos limites de tempo necessários, ou quando é necessária um tipo de peça standard, deverá decidir-se se ela vai ser fabricada, se vai ser comprada a terceiros (subcontratação ou "outsourcing") ou se as duas situações deverão ocorrer em simultâneo. No caso onde a opção é a subcontratação a terceiros, pode subcontratar-se o fabrico apenas, ou o fabrico e a engenharia dos componentes a incluir na produto final. Dependendo da situação escolhida, a decisão deverá ser negociada em alturas diferentes, em função do tipo de indústria e da complexidade dos produtos. No caso de um componente standard o seu fabrico poderá ser subcontratado mais tarde, podendo o seu fornecimento ser mesmo realizado "Just-in-Time". Na situação onde a engenharia é também subcontratada, a sub-contratação surge mais cedo no tempo durante a fase de planeamento tecnológico.

A monitorização e o controlo do fabrico são assim da maior importância. O controlo é responsável por garantir que o fluxo de informação sobre o processo activa o fluxo de materiais na sequência correcta de forma a que todas as operações estejam sincronizadas. A monitorização por seu lado fornece a realimentação ao controlo que verifica se os eventos ocorrem ou não como o planeado. Na tabela 2.1 temos diversas actividades de monitorização da instalação fabril com a informação recolhida para cada uma delas e os respectivos relatórios produzidos.

A actividade de compras e recepção de material é responsável pela compra de matéria prima, peças standard necessárias à produção e componentes a serem adquiridos a fornecedores externos. Existem múltiplas estratégias, podendo as peças ser encomendadas a pedido ou para stock. A decisão entre estas duas alternativas depende naturalmente da disponibilidade das peças no mercado e do custo dos stocks. Se a sua disponibilidade no mercado for baixa, então será boa estratégia manter um stock de segurança para que não surjam quebras na produção com a consequente e inevitável ruptura das entregas. Na produção em massa, é comum o contrato com fornecedores para que as peças ou matérias primas apenas sejam entregues no momento em que são necessárias para montagem ou transformação, de acordo com estratégias de produção "Just-in-Time".

Quando alguma peça ou material é pedido para a produção, primeiro é verificada a existência em stock; se necessário, são enviados pedidos de propostas de fornecimento a possíveis fornecedores. Assim que as propostas chegam, será realizada a encomenda em face da melhor proposta.

Tabela 2.1 - Actividades de Monitorização da Instalação Fabril [18]

Informação recolhida	Actividade	Relatórios
Identificação dos Empregados Hora de chegada e de saída	Controlo de presenças	Presenças Tempos de permanência Horas extraordinárias
Despacho de ordens de fabrico Início de tarefa, fim de tarefa e tempo de preparação Expedição Pedido de assistência Atribuição de recursos Alteração de atribuição	Atribuição de Tarefas	Relatório de actividade da instalação fabril Relatórios de fim de turno Relatórios de problemas Identificação de problemas
Pedidos de material Trocas de material Armazenamento de material Monitorização de equipamento Controlo de equipamento	Atribuição de materiais	Relatório de materiais Atribuição de deslocação Relatório de actividade Relatório de problemas Stock Stocks intermédios Relatório do estado e dos problemas de equipamento
Programas NC ou outros para máquinas Monitorização de recursos (máquinas) Tempos de preparação de recursos Tempos de reparação	Controlo dos recursos Máquinas	Níveis de actividade Avarias Produtividade Identificação de problemas Assistência de Manutenção
Procedimentos controlo qualidade (CQ) Programas de inspecção e teste do CQ	Controlo da Qualidade	Relatórios CQ Identificação de problemas Tendências

Quando a encomenda é recebida, a quantidade entregue, assim como outros requisitos de inspecção da qualidade, são verificados contra o contrato de fornecimento realizado, sendo a entrega aceite se todos requisitos do contrato forem cumpridos. Em algumas situações algumas das peças não satisfazem os requisitos, sendo então realizada uma nova encomenda. Completada a inspecção, as peças recebidas são introduzidas para produção ou para stock.

A inspecção por amostragem é obviada nos casos em que a confiança no fornecedor o permite, sendo garantia da qualidade assegurada através de auditorias.

2.3.5 Controlo de Fabrico e Monitorização

O sistema de produção perfeito não deveria necessitar de stocks. O capital investido no stock está parado e como tal não é aplicável na operação da instalação fabril. No entanto, na prática o stock é necessário como arma contra as rupturas de fornecimento, ou contra a perda de oportunidades de venda por falta de matéria prima para encomendas inesperadas. Existem diversos tipos de stocks, os stocks para matéria prima, para peças adquiridas, para peças acabadas, para produtos acabados e para peças sobresselentes. Além disso, existe ainda o stock no decorrer do processo e a ele inerente que funciona como *buffer* entre operações (stocks do em-curso-de-fabrico). Através de um sistema de gestão de stocks, todo este stock deve ser gerido e controlado nos níveis mínimos. Uma informação importante para o stock são os tempos associados ao processamento da encomenda e ao próprio fabrico no "chão-de-fábrica" ("lead-time", "throughput-time", "cycle-time").

A gestão da qualidade é uma função de supervisão e controlo apresentando interfaces com a maior parte das actividades da empresa. A qualidade do produto deve ser pensada no momento do seu projecto. Existe normalmente um conflito de interesses quando a qualidade é observada de diversos pontos de vista. Assim sendo: o cliente pretende um produto versátil, com um tempo de vida longo e que preencha por completo as suas necessidades; a engenharia deseja conceber um produto com elevadas características do ponto de vista técnico, que seja fiável e que apenas apresente as características que o tornam útil; a produção quer utilizar métodos de fabricação e materiais económicos; e as vendas anseiam por uma maior fatia de mercado, com um melhor produto e mais barato. A decisão de compromisso que se deseja atingir nem sempre é fácil. Ligando tudo isto, devem ser estabelecidos padrões de qualidade, e todas as operações na instalação fabril devem obedecer a um apertado procedimento de controlo da qualidade.

As actividades ligadas à gestão da qualidade, são orientadas por normas internacionais (ISO9000/1/2/3), por normas impostas por clientes (e.g. FORD Q101), standards internos à empresa, programas e procedimentos de teste e algoritmos para avaliação da informação recolhida, cálculos da capacidade do processo e das tendências, etc.. Usando ferramentas diversas, os parâmetros do produto ou processo a vigiar e os limites de controlo recomendados, o controlo da qualidade recolhe a informação que será tratada e apresentada sob a forma de um conjunto de relatórios cobrindo os produtos aceites e rejeitados, as tendências da qualidade, históricos de qualidade, custos da qualidade, e o desempenho do processo.

A manutenção é definida pela AFNOR ("Association Française de Normalisation") (NF X 60-10) como: "o conjunto de acções ou operações que permitam manter ou re-estabelecer um bem dentro de um estado específico ou na medida necessária para assegurar um serviço necessário" [19,20]. Embora nos primeiros tempos a manutenção fosse reduzida a uma actividade menor subsidiária da produção, ela é actualmente um serviço autónomo dentro da empresa. Para além da manutenção correctiva, surge a manutenção preventiva, de início apenas baseada numa manutenção sistemática, caracterizada por intervenções anteriores à ocorrência da avaria e efectuada em função da vida esperada dos equipamentos e a partir das leis de degradação respectivas e dos dados resultantes de experiências anteriores. Mais tarde surge um novo ramo da manutenção preventiva, a manutenção condicionada, definida como a manutenção preventiva subordinada a um tipo de evento pré-determinado (como por exemplo o auto-diagnóstico, ou a informação de um sensor) revelador do estado de degradação do equipamento com o fim de prevenir avarias inesperadas.

A manutenção preventiva tem vindo também a ser introduzida como procedimento corrente para equipamento de produção standard. A informação dos planos de manutenção dos recursos de fabrico é fornecida ao planeamento da produção com a informação da sua disponibilidade. A manutenção deverá manter um registo com a informação do desempenho do equipamento e dos problemas surgidos.

2.3.6 Orçamentação e Contabilidade

A contabilidade cobre três aspectos, o crédito, o controlo e o débito, e é responsável por manter toda a informação relativa à saúde financeira da empresa. Os orçamentos de operação são estabelecidos por um período de um ano para toda a empresa e para cada uma das suas subsecções. As unidades operacionais poderão ser divididas em centros de custo com um orçamento definido onde são debitadas as despesas realizadas. Periodicamente

devem ser realizados relatórios sumário onde são apresentados os lucros ou prejuízos. Não é simples a avaliação do desempenho destes centros de custo, uma vez que a função atribuída a um centro de custo leva a que ele seja um simples consumidor do orçamento. Por outro lado existe uma distinção que deve ser feita entre os custos fixos e os custos variáveis, sendo estes últimos os mais difíceis de estimar. O custeio industrial pode apresentar variantes tais como por exemplo a contabilização por produto ou por actividade ("activity based costing"), com objectivos de planeamento e controlo mais versáteis.

2.4 Conclusão

A evolução das relações entre a empresa industrial e o mercado e o consequente desafio que lhes é colocado explica a evolução que se tem vindo a assistir no âmbito da gestão e da organização fabril. Nos anos 80, havia produção para um mercado assegurado. Hoje em dia este cenário mudou e as empresas têm que se certificar de que o mercado existe antes de fabricar. Neste contexto, o grande mercado da subcontratação funciona para fabricar o que já foi vendido. Este é o resultado de um ambiente competitivo global a nível mundial. O mercado tomou o poder que anteriormente pertencia à empresa de fabrico sendo ela agora controlada pelo tempo e pelo cliente. Vive-se assim um clima de "guerra económica" e também uma nova revolução industrial que provoca ondas de entusiasmo, de novas ideias, de resistência à mudança, de imaginação, de revisão dos valores tradicionais, e de busca de novos modelos.

Neste ambiente efervescente de novas ideias e conceitos podemos tentar resumir a essência do pensamento futuro e das direcções que ele poderá tomar nas seguintes palavras chave: Integração, Fluxo, Evolução, Autonomia, Inteligência, Flexibilidade e Simplificação [12]. A Integração impõe-se como um dos conceitos fundamentais nas novas organizações industriais, apresentando a tendência da evolução onde o "óptimo" se mede ao nível da empresa global e não ao nível de cada um dos seus componentes ou partes. O Fluxo pretende transmitir uma visão dinâmica do controlo na empresa, através da gestão e aumento do fluxo do produto ao longo de todo o seu ciclo de vida (desde o momento em que é projectado até ao produto acabado e eventualmente reciclado). A noção de tempo, velocidade e de débito permite a medida da capacidade de reacção da empresa às solicitações do mercado, tornando o conceito de fluxo fundamental na gestão e no controlo. A Evolução ou "Melhoria Contínua" (kaizen) aponta para uma segunda dimensão no tempo, significando que nada é definitivo, apresentando os produtos, os processo e as organizações um ciclo de vida muito curto. Por este motivo, a gestão da evolução deverá ser uma actividade que envolve todas as pessoas da empresa. A Autonomia e a Inteligência caracterizam o comportamento de cada parte elementar do Sistema Industrial Total, constituindo a chave para o desenvolvimento de uma empresa adaptável e com capacidade de resposta. Neste contexto, são as pessoas que constituem o principal factor no ciclo de melhoria. A Flexibilidade caracteriza a capacidade que cada parte do sistema total tem de atingir a maior capacidade de reacção global, por exemplo capacidade: de tratar flutuações no volume e mistura de produtos; de reduzir "lead-times" e tempos de lançamento de novos produtos; de reagir a avarias de equipamento; de evoluir rapidamente para um novo ambiente de fabrico. Temos finalmente a Simplificação, que hoje em dia se acredita condicionar fortemente o desempenho industrial, tornando-se na pedra angular de todas as actividades de automatização e modernização. É simplificando que se conseguem economias que permitirão o financiamento da necessária evolução. Simplificar significa

acima de tudo observar o funcionamento da empresa de uma forma que permita a constante busca de soluções que permitam eliminar as verdadeiras causas do mau funcionamento. Implica naturalmente o exame cuidado das situações existentes procurando-se eliminar todas as actividades que não acrescentam qualquer valor ao produto. É neste ambiente em constante mudança que se procura a utilização racional dos factores de produção, optimizando o processo de fabrico e evitando a super-produção ou a produção de coisas sem interesse.

O conceito dos anos 80 de fabrico integrado por computador (CIM) veio disponibilizar ferramentas integradas suportando as múltiplas actividades da empresa de fabrico, tentando de uma forma integrada dar resposta a este conjunto de problemas. Uma vez provado que a solução do problema não está de facto na automatização global, o CIM ou mais exactamente a Integração dos Sistemas de Fabrico procura providenciar a todos os níveis a informação correcta, atempada e no local desejado de forma a permitir o apoio à decisão.

O fabrico integrado por computador foi apresentado neste capítulo como uma disciplina essencialmente interdisciplinar, envolvendo todas as pessoas da empresa, portanto com diferente formação e com interesses e objectivos individualmente diversos. A integração do sistema de fabrico é de facto complexa, e por este motivo tem-se vindo a estudar diferentes arquitecturas de integração e de projecto. Estas arquitecturas têm por objectivo propor um modelo de abstracção da realidade que facilite o tratamento da complexidade da integração tema esse que irá ser abordado no capítulo seguinte.

Capítulo 3

ARQUITECTURAS DE INTEGRAÇÃO

3.1 Introdução

O tema deste capítulo surge naturalmente na sequência dos assuntos abordados no capítulo anterior. A complexidade de um sistema de fabrico, assim como a complexidade de um projecto de integração numa empresa de fabrico, exige a combinação de esforços multidisciplinares conjugados para que se atinjam os objectivos propostos com sucesso e o mínimo de esforço. Neste contexto, são da maior importância o estudo e a reflexão crítica sobre as denominadas arquitecturas de integração existentes. Esta importância reflectiu-se nos trabalhos desenvolvidos por um grupo de especialistas da IFAC/IFIP (*International Federation of Automatic Control / International Federation For Information Processing*) na área das Arquitecturas para a Integração das Actividades de Fabrico.

Neste contexto, serão apresentadas as três arquitecturas mais conhecidas, a "Purdue Enterprise Reference Architecture" (PERA), a "GRAI Integrated Methodology" (GIM) e o "Open Systems Architecture for CIM" (CIMOSA). Serão ainda apresentadas a "Architecture of Integrated Information Systems" (ARIS) e a "Integrated Enterprise Modelling" (IEM). A IEM foi desenvolvida no instituto Fraunhofer-IPK, que tem realizado um grande esforço para a normalização junto de grupos de trabalho como o Comité Europeu de Normalização (CEN).

Finalmente, e face à sua importância na tentativa de constituir uma arquitectura reunindo as principais facetas das acima referidas PERA, GIM e CIMOSA, será apresentada a metodologia GERAM ("Generic Enterprise Architecture and Methodology").

3.2 Arquitecturas de Suporte à Integração das Actividades Empresariais e de Fabrico

3.2.1 O Problema da Integração da Empresarial

A integração empresarial é simultaneamente um problema de tecnologia e de estratégia. O seu objectivo fundamental é a construção de uma empresa reactiva, capaz de actuar em tempo real e de uma forma adaptativa, e respondendo globalmente às necessidades do utilizador, devendo ainda apresentar uma flexibilidade que lhe permita responder adequadamente a mudanças tecnológicas e sociais [21].

A denominada empresa integrada coordena as suas actividades e todas as decisões estratégicas, tácticas e operacionais através de um fluxo de informação eficiente e atempado e de uma organização que lhe permite a utilização da informação de uma forma óptima. Além disso, a empresa integrada deverá ainda apresentar outros atributos, tais como: estrutura organizativa adaptável e de fácil manutenção, utilização e reutilização eficiente dos recursos disponíveis, incluindo as capacidades em recursos humanos, e os bens de investimento nas tecnologias da produção e da informação.

O desenvolvimento de um sistema integrando os fluxos de informação inerentes ao processo de fabrico, às actividades de suporte ao fabrico e à organização da empresa é uma tarefa complexa que se depara com três principais problemas. Em primeiro lugar, o custo e o risco de um desenvolvimento deste tipo poder ser muito elevado se não existir experiência anterior. Como segundo problema, o conhecimento exigido para a realização de um projecto deste género não existe geralmente dentro da própria empresa. Por último, o desenvolvimento de novas estruturas organizacionais e de informação pode ou não ser viável, ou podem essas estruturas resultar demasiado frágeis se não assentarem nas tecnologias geralmente disponíveis no mercado para o tipo de empresa.

Surge deste modo a ideia do desenvolvimento de arquitecturas de referência que permitam enquadrar e guiar projectos de integração, ideia esta baseada no facto de uma parte substancial dos projectos de integração terem aspectos comuns a qualquer tipo de empresa. Assim sendo, a parte comum poderia ser identificada, estandardizada e reutilizada, evitando sucessivos re-desenvolvimentos. Poder-se-ia ainda capitalizar com a utilização em novos projectos da experiência e dos resultados ganhos em projectos anteriores. Uma vez aceites, as arquitecturas de referência poderiam ser suportadas por ferramentas e metodologias de análise, projecto e implementação, e por uma gama de produtos compatíveis, que tornariam a tarefa de integração possível com um menor esforço em tempo e custos.

Existe uma multiplicidade de arquitecturas e de modelos apresentados na literatura, descrevendo e estruturando as tarefas de integração das actividades de fabrico e processos de negócio a eles associados. No entanto, só algumas delas tratam de facto o "como" se faz a integração, assim como "o que é" necessário à integração. A grande maioria concentra-se na descrição da estrutura de um sistema de controlo por computador, assim como na conectividade das diversas funções envolvidas, i.e., concentram-se n' "o que é" necessário à integração (arquitectura tipo I), não se referindo ao "como" realizar a integração. Na caracterização de "como" se faz a integração deverá ser descrita a arquitectura do próprio projecto, a ser respeitada no processo de integração, abrangendo todo o ciclo de vida, i.e., análise, projecto, construção, instalação e operação da empresa integrada (arquitectura tipo II). Do ponto de vista do utilizador, a arquitectura tipo II é a de maior relevância, uma vez que apresenta a relação entre as tarefas de projecto envolvidas no conceito, na análise de

projecto, na construção, na colocação em serviço e na operação da empresa integrada. A arquitectura tipo II deverá ser acompanhada por uma metodologia que guie o utilizador, ou seja, deve ser apresentado não só um modelo do processo de desenvolvimento mas também a respectiva metodologia. Por metodologia entende-a aqui se um conjunto de linhas mestras, técnicas e procedimentos que podem ser seguidos pelo seu utilizador na implementação de uma determinada política de integração da empresa, e que são geralmente desenvolvidos juntamente com uma arquitectura de referência. Esta metodologia deverá ainda especificar ou sugerir quais as ferramentas computacionais ou outras, assim como técnicas a serem utilizadas na sua aplicação.

Temos assim o problema tecnológico da integração no fabrico de certa forma resolvido pelas arquitecturas de referência que descrevem a estrutura do sistema de controlo e a interligação entre funções envolvidas e fornecem ferramentas (tipo I). Estas ferramentas serão então da maior utilidade na concretização de uma estratégia onde se estabelece o "como" se faz a integração (tipo II). O actual cenário nos ambientes de fabrico torna evidente a grande necessidade de recorrer a Arquitecturas de Referência que estabeleçam "como" se projecta, desenvolve, implementa e usa a empresa integrada de fabrico, não sendo de forma alguma suficiente o projecto do sistema de informação.

A "Task Force" IFAC/IFIP que trabalha em Arquitecturas para a Integração das Actividades de Fabrico identificou que o utilizador potencial tem uma grande necessidade de um manual de instruções que lhe indique "como" projectar, desenvolver, implementar e utilizar a integração da empresa e do fabrico [21].

Das múltiplas arquitecturas existentes actualmente, apenas as três seguintes abrangem por completo ciclo de vida do projecto do Sistema Integrado de Fabrico [21]:

- A arquitectura CIMOSA ("Open System Architecture for Computer Integrated Manufacturing"), foi desenvolvida pelo consórcio Europeu AMICE no âmbito de projectos ESPRIT 688, 2422 e 5288. Com o trabalho iniciado em 1984, a CIMOSA propõe uma arquitectura para o projecto de integração (associada ao conhecido cubo CIMOSA) onde são enumerados os modelos necessários, e apresenta o conceito de cooperação entre o ambiente de engenharia e o ambiente operacional da empresa [22, 23, 25].
- A arquitectura GIM ("GRAI-Integrated Methodology"), foi desenvolvida no Laboratório GRAI da Universidade de Bordéus em França. Este trabalho resultou dos trabalhos realizados por este laboratório na área da gestão da produção desde 1974, tendo a sua forma actual sido atingida em 1984, altura em que passa a ser conhecido como GIM. O GIM é caracterizado pela utilização do modelo GRAI ("Graphe de Résultats e Activités Interreliés"), definindo a sua utilização para quatro sistemas cooperantes: os sistemas de decisão, de informação, operacional e físico. Além disso, a abordagem estruturada do GIM é caracterizada pela definição do ciclo de vida do projecto de integração [26, 27, 28].
- A arquitectura PERA, "Purdue Enterprise Reference Architecture", e a relacionada "Purdue Methodology", foram desenvolvidos na Universidade de Purdue no âmbito de um trabalho realizado em cooperação com a indústria pelo "Purdue University Consortium for CIM". A arquitectura de Purdue é caracterizada pela estruturação do ciclo de vida da empresa em fases correspondendo a tarefas que o cobrem por completo, assim como pela representação explícita do papel do elemento humano na empresa [1, 29].

A integração do fabrico não se limita à automação e / ou à aplicação computacional por si só, embora tanto a informatização como a automação estejam normalmente envolvidas na

sua implementação. A integração da empresa de fabrico consiste na aquisição, processamento, armazenamento e utilização de dados e/ou informação dos processos de negócio e fabrico e do seu ambiente, por forma a optimizar a sua operação de acordo com critérios estabelecidos pelos elementos responsáveis pela sua gestão. Esta optimização exige a integração dos fluxos de matérias primas, produtos intermédios, e produtos acabados, bem como a organização e gestão adequadas dos recursos e dos procedimentos de fabrico.

Nos parágrafos seguintes são apresentadas com algum detalhe as arquitecturas referidas atrás. Outras arquitecturas, tais como a ARIS ("Architecture of Integrated Information Systems") e IEM ("Integrated Enterprise Modelling"), embora não seleccionadas pela "Task Force" IFAC/IFIP são também tratadas como consequência da sua relevância em produtos comerciais no primeiro caso, e para o grupo de estandardização CEN/TC310 WG1 no segundo. No final, serão apresentadas algumas conclusões relativas à relevância de cada uma destas arquitecturas.

3.2.2 "Purdue Enterprise Reference Architecture", PERA

No decorrer de um projecto de três anos (1989-1992), um consórcio de empresas industriais desenvolveu em conjunto com a Universidade de Purdue uma metodologia de implementação de um Sistema Integrado de Fabrico (SIF), com o objectivo de tornar essa implementação mais simples e aumentar a sua probabilidade de sucesso. Esta metodologia tem como suporte a denominada Arquitectura Empresarial de Referência de Purdue [30].

A Arquitectura de Referência

A Arquitectura de Referência de Purdue é uma arquitectura tipo II onde se descreve "como" se realiza o projecto de integração. A figura 3.1 ilustra as diferentes fases propostas por esta arquitectura. Principiando pela caracterização da Entidade de Negócio ("CIM Business Entity"), que poderá ser entendida como a Unidade de Negócio nos meios empresariais a tratar, são identificados os objectivos da integração assim como a visão da gestão, estratégia e objectivos para aquela unidade. Na denominada Fase Conceptual são realizadas outras considerações sobre filosofias / estratégias de operação e implementação ou acções obrigatórias na selecção dos processos ou fornecedores, e são definidas políticas para o fabrico, pessoal e sistema de informação. São assim estabelecidas as estratégias operacionais e os objectivos para as Arquitecturas de Informação (AI) e de Fabrico (AF) assim como para a sua implementação

Na Fase de Definição, são utilizadas as orientações estratégicas da gestão (da Fase Conceptual) para o estabelecimento dos requisitos. São assim estabelecidos os requisitos para a implementação da AI que vai permitir a execução das políticas operacionais da empresa. São ainda estabelecidos os requisitos para a produção de produtos ou serviços a serem gerados pela empresa. Surgem deste modo dois tipos distintos de requisitos, identificando tarefas relacionadas com a informação ou relacionadas com o fabrico. Estas tarefas ou módulos funcionais vão permitir respectivamente a execução dos requisitos impostos à AI e à AF, sendo finalmente agrupadas e inter-ligadas formando redes de fluxo de informação, materiais e de energia. Estas redes podem surgir como diagramas de fluxos de dados ou de processos ilustrando as suas conexões. Até este ponto não são realizadas quaisquer considerações relativamente à posição dos operadores humanos no sistema.

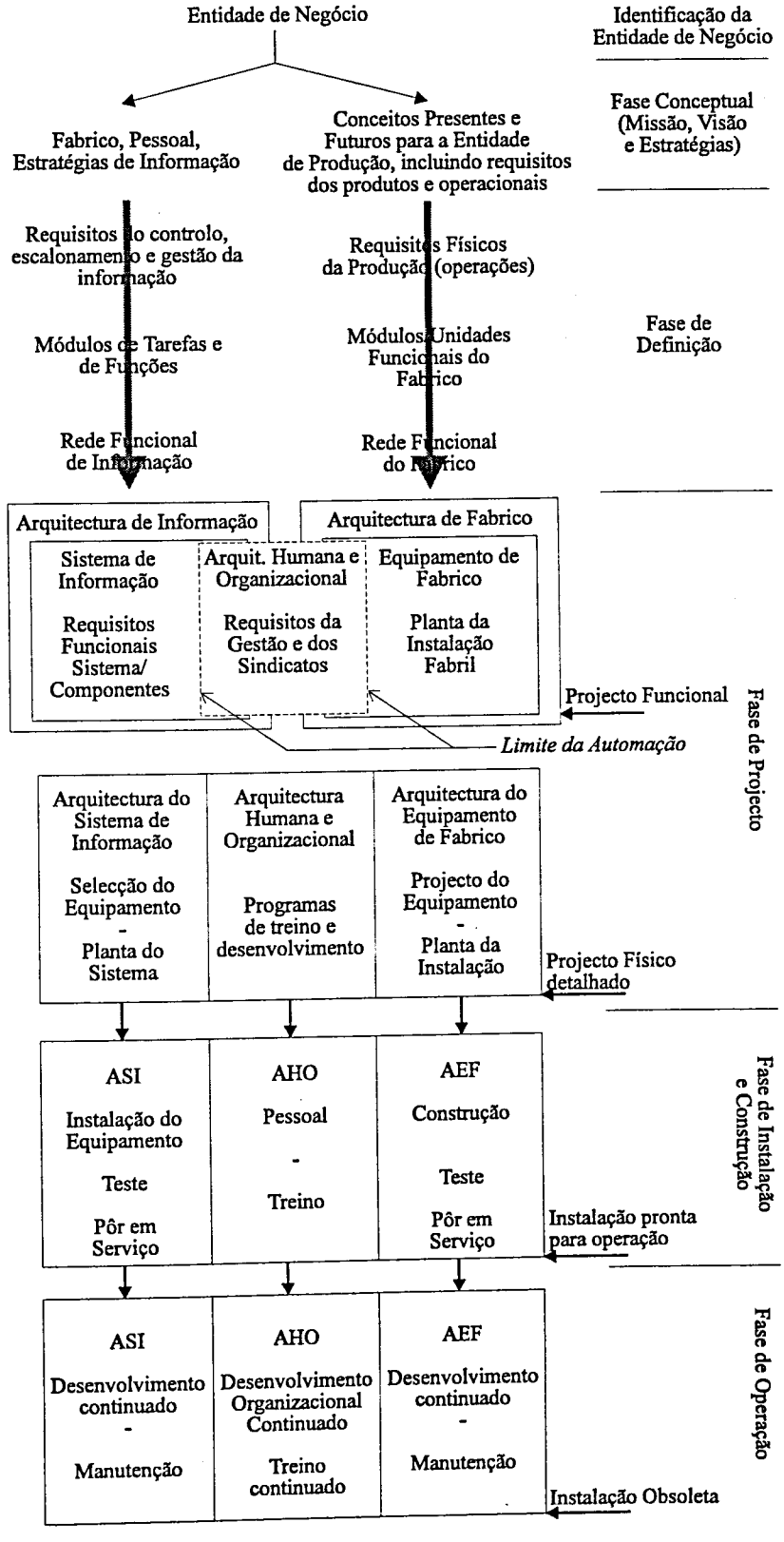


Figura 3.1 - Arquitectura de Referência de Purdue

Uma vez decidida a implementação, são identificadas quais as tarefas de cada uma das vertentes (informação / fabrico) que serão realizadas pelas pessoas distinguindo-as das tarefas que serão realizadas pelo equipamento de fabrico ou pelo sistema de informação. Na parte final da Fase de Definição caracteriza-se assim o posicionamento humano na Arquitectura do Sistema Informação (ASI) e do Equipamento de Fabrico (AEF), constituindo-se a chamada Arquitectura Humana e Organizacional (AHO). As tarefas que restam na ASI serão executadas pelo sistema de informação, i.e. pelas redes de comunicação, bases de dados, software, etc. Por outro lado, as que restam na AEF são executadas pelo equipamento de fabrico na instalação fabril. Em empresas cujo negócio não inclua o fabrico, a AEF é substituída pela Arquitectura do Serviço. As duas especificações funcionais, que se reflectem nas redes funcionais definidas na fase anterior, são assim convertidas nas três sub-arquitecturas de implementação da PERA, cada uma delas formando por si só ambientes onde são utilizadas ferramentas, modelos e métodos adequados aos desenvolvimentos e implementações específicos. Estas sub-arquitecturas apresentam assim a sua contribuição própria para o projecto de integração empresarial.

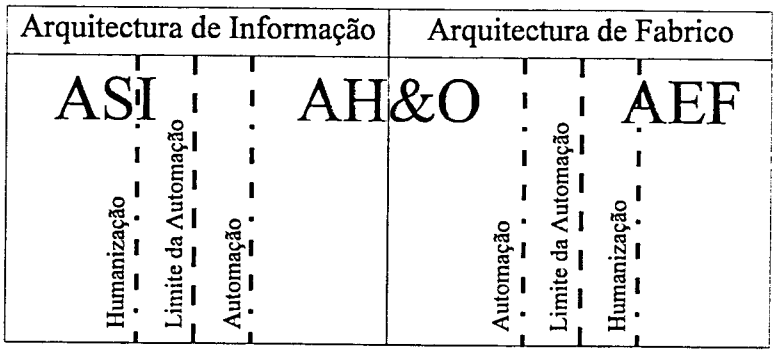


Figura 3.2 - As linhas da Automação e da Humanização entre as três Arquitecturas de Implementação de Purdue

As arquitecturas anteriores são separadas por uma linha definindo as interfaces existentes entra cada uma delas. A figura 3.2 ilustra estas linhas, representando as interfaces entre as arquitecturas de implementação na Arquitectura de Projecto de Purdue. A primeira é a linha da Automação ("Automatability") que estabelece o limite absoluto até ao qual a tecnologia é de facto capaz de automatizar as tarefas e as funções do sistema integrado de fabrico (SIF) da entidade de negócio em causa. Esta linha está limitada pelo facto de muitas das tarefas exigirem intervenção humana com capacidades de flexibilidade, adaptação e inovação constantes. A segunda linha é a chamada linha da Humanização ("Humanizability") que mostra até que limites os operadores humanos podem de facto ser usados na implementação de tarefas e funções no SIF, e tem de se encontrar com a linha anterior em completa sobreposição. Este limite é restringido pelas capacidades humanas, tais como tempos de resposta, capacidade de compreensão, visão e força física, etc. Nas fronteiras ASI/AHO e AEF/AHO surge assim uma linha virtual, a do Limite da Automação ("Extent of Automation") (ver figura 3.1) que estabelece de facto qual a fronteira planeada para a automação na entidade do negócio em estudo. A determinação da linha Limite da Automação leva em consideração factores económicos, sociais (costumes, leis e directivas, regras dos sindicatos) e tecnológicos.

As três arquitecturas apresentadas são sub-arquitecturas da Arquitectura de Referência de Purdue. São chamadas arquitecturas uma vez que por si só elas formam um conjunto de

quadros para a utilização de ferramentas, modelos, etc., para o desenvolvimento de cada uma das suas contribuições para o SIF da empresa em estudo. É neste contexto realizada a Fase de Projecto Detalhado: do equipamento e do software da ASI; das capacidades a desenvolver em cursos de formação e na organização da AHO; dos componentes, processos e equipamento da AEF.

Surge assim de uma forma lógica o processo de implementação, teste e colocação em serviço das Arquitecturas definidas nos parágrafos anteriores. A gestão da evolução é também contemplada na Fase de Operação onde é previsto o desenvolvimento continuado das diversas arquitecturas assim como da sua manutenção.

Método de Representação das Tarefas e dos Módulos na Arquitectura de Purdue

O modelo de Referência de Purdue, tal como muitos outros métodos de análise, utiliza uma abordagem topo-base (i.e. do geral para o particular, ou da função para a tarefa). A abordagem base-topo é também utilizada assegurando a correcta definição e descrição das tarefas ou funções disponíveis e identificando funções ou tarefas comuns provenientes de outras fontes. Esta última abordagem assegura ainda que a descrição dos requisitos de implementação está completa. É então realizada a descrição detalhada de uma tarefa individual, função ou macro-função, a ser utilizada na subsequente análise topo-base.

A figura 3.3 ilustra o módulo básico utilizado na caracterização de uma tarefa, função, ou macro-função em particular. Os módulos assim definidos na abordagem base-topo podem então ser usados na subsequente análise topo-base. A tabela 3.1 apresenta os diferentes tipos de tarefas que podem surgir na construção do modelo, bem como as respectivas entradas, saídas, parâmetros, armazenamento e processo de transformação [31].

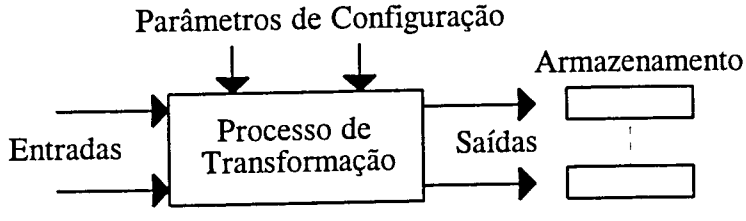


Figura 3.3 - Definição de uma Tarefa (PERA)

Uma tarefa de informação é por exemplo o processo de escalonamento da produção envolvendo operações sobre informação (dados) e um algoritmo (com características determinísticas ou estocásticas), podendo assim ser considerado um bloco computacional. O algoritmo é aqui referido no sentido lato, podendo significar tanto uma expressão matemática como um procedimento programável.

Para um processo de fabrico, as tarefas básicas incluem operações genéricas e irredutíveis. A modularidade a este nível é atingida através de um conjunto de directivas, nomeadamente: a modularidade será estabelecida ao nível da tarefa ou da função; e todas as transformações físicas, químicas, posicionais ou de forma estão localizadas no nível mais baixo da denominada Hierarquia Funcional de Purdue (HFP) [31]. A HFP é realizada usando os blocos construtivos da figura 3.3 para o estabelecimento dos diversos níveis hierárquicos de controlo, por exemplo: nível físico / processo (nível 0) - estação de trabalho (nível 1) - célula (nível 2) - instalação fabril (nível 3). A utilização dos referidos blocos construtivos básicos, permite a especificação funcional dos diversos módulos envolvidos bem como das interfaces que entre eles forem surgindo. Além disso, um módulo poderá agrupar um conjunto de transformações relacionadas, por exemplo as realizadas no mesmo

equipamento ou posto de trabalho, e incluir também o equipamento necessário para as executar.

Tabela 3.1 - Descrição do Processo de Transformação em função do Tipo de Tarefa

Tipo de Tarefa	Entrada	Saída	Processo de Transformação	Parâmetros	Armazenamento
Informação	Dados, incluindo informação temporal	Saídas calculadas	Algoritmos	Numéricos	Base de Dados (pode não existir)
Fabrico	Material e / ou energia	Material transformado ou energia	Processo de fabrico incluindo equipamento	Físicos (valores de propriedades físicas)	Stock (pode não existir)
Humana ou Organizacional (não automatizada ou mecanizada)	Requisitos, especificação, informações disponíveis, incluindo informação temporal	Oral, escrita ou movimento físico	Tarefa escrita, declaração	Regras, conceitos, princípios, ferramentas, etc.	Base de Dados (biblioteca ou ficheiro; pode não existir)

A Metodologia de Purdue

Os conceitos básicos desta metodologia para o CIM incluem:

- a necessidade de um *Plano Director* da transição para o novo sistema antes de se iniciar a implementação de qualquer sistema integrado;
- a disponibilidade de um *Manual de Instruções* que guie e simplifique a construção do Plano Director;
- a inclusão no Plano Director da proposta do novo sistema integrado, sob a forma de um conjunto de projectos com prioridades associadas;
- a existência de uma *Arquitectura de Referência* que estabeleça o quadro para o desenvolvimento e utilização do Manual de Instruções, para a utilização do Plano Director resultante e para a implementação da Proposta do novo sistema integrado.

O *Manual de Instruções* está publicado com o título "An Implementation Procedures Manual for Developing Master Plans for Computer Integrated Manufacturing" [1].

A Arquitectura de Referência de Purdue cobre ainda as necessidades que mais cedo ou mais tarde a empresa vai sentir quando desejar introduzir alterações (melhoramentos) nas suas práticas de negócio, nos métodos de tratamento da informação ou nas tecnologias do fabrico ou do processo. A Arquitectura fornece assim uma quadro para todos os Planos Directores do sistema integrado.

O Plano Director compara o estado presente da empresa (Presente, "*AS-IS*") com o estado futuro desejado (Futura, "*TO-BE*") para caracterizar o caminho de transição entre os dois estados. O caminho de transição é estabelecido confrontando os diagramas das duas arquitecturas presente e futura [1,30].

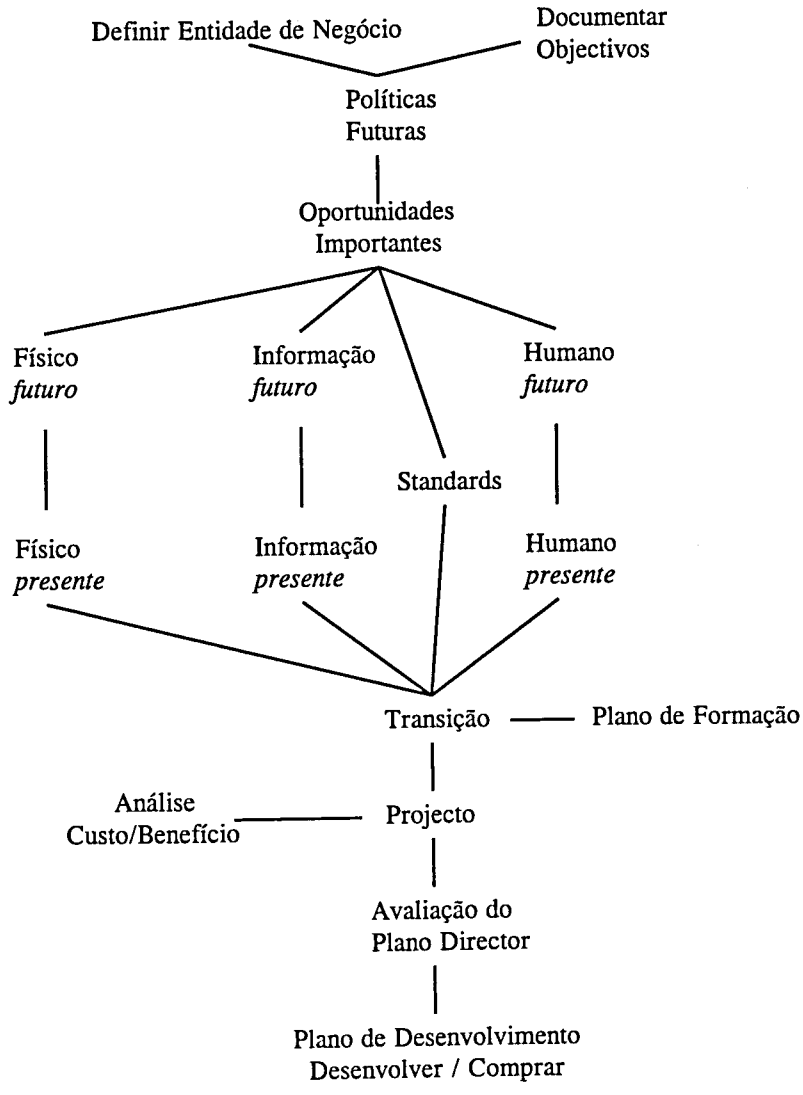


Figura 3.4 - Fluxo de Informação e Faseamento no Plano Director da arquitectura PERA

Conclusão

Como é possível constatar, a Arquitectura de Purdue permite a introdução de todos os aspectos essenciais à caracterização da empresa assim como à definição do seu percurso de evolução. São deste modo tomados em consideração: os processos de tomada decisão, as actividades, os processos do negócio, a troca de informação, e os fluxos de materiais e de energia.

A arquitectura PERA disponibiliza duas classes de arquitecturas: as arquitecturas do sistema físico (tipo I) e a arquitectura (quadro) do projecto de desenvolvimento (tipo II), permitindo assim a caracterização do ciclo de vida completo da empresa. Esta arquitectura faz a separação da análise funcional em duas vertentes: a análise funcional da informação combinando a gestão e o controlo dos dados; a análise funcional do serviço ou sistema físico de fabrico. Por outro lado, a PERA define o posicionamento humano na empresa através da distribuição dos dois tipos de tarefas (de processamento de informação e de fabrico), introduzindo a denominada linha do Limite da Automação e o ciclo de vida dos componentes humanos, de informação e de fabrico. Define ainda um verdadeiro modelo de referência, uma vez que faz a extensão do conceito de fabrico ao conceito de serviço, resultando na substituição da AEF pela Arquitectura do Serviço, e faz corresponder às

arquitecturas tipo I a estrutura do ciclo de vida do sistema CIM (figura 3.1 na página 35). Finalmente, permite que sejam atribuídas às diversas fases da arquitectura as ferramentas mais adequadas à execução das tarefas envolvidas em cada uma delas e estabelece um Plano Director, que consiste num conjunto de procedimentos compilados sob a forma de um manual contendo a descrição dos passos essenciais à implementação do sistema CIM.

A Arquitectura de Referência de Purdue pode ser estendida de forma a cobrir qualquer tipo de empresa, e não somente as empresas de fabrico, ou programa de implementação de um sistema integrado.

3.2.3 GIM "GRAI Integrated Methodology"

A metodologia GIM ("GRAI-Integrated Methodology") foi desenvolvida no Laboratório GRAI da Universidade de Bordéus em França. A metodologia é caracterizada pela utilização do denominado modelo GRAI [28], e por uma abordagem estruturada à derivação das especificações de projecto partindo da análise de requisitos. Esta metodologia que permite a definição da especificação do chamado "Sistema Avançado de Fabrico" (SAF) [26], levando em consideração os requisitos de integração dos vários aspectos do sistema, é composta por um conjunto consistente de componentes: um modelo de referência (o modelo GRAI), estabelecendo a estrutura genérica do sistema integrado de fabrico a estudar; vários formalismos de modelação, permitindo a descrição dos vários componentes do sistema; uma abordagem estruturada, estabelecendo um conjunto de passos para a migração do sistema actual para o sistema futuro; e critérios de avaliação de desempenho, que permitem a avaliação do sistema de diversos pontos de vista.

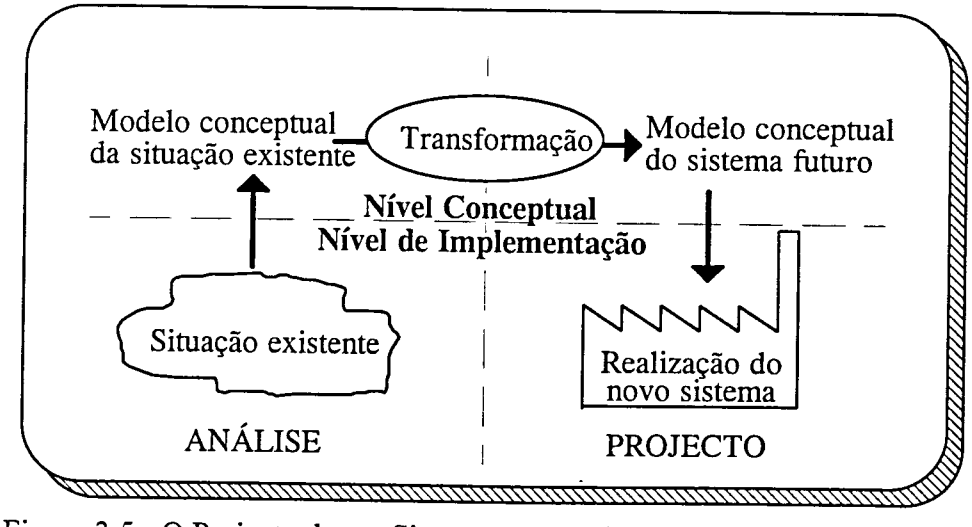


Figura 3.5 - O Projecto de um Sistema Avançado de Fabrico segundo a GIM

O objectivo da metodologia GIM é o de fornecer uma abordagem estruturada ao suporte da fase de Análise e Projecto de um sistema integrado de fabrico (SIF) (figura 3.5). A fase de análise orientada ao utilizador é precedida pela chamada fase de iniciação, cujo objectivo consiste na definição dos objectivos da empresa, a apresentação da metodologia aos seus gestores, a definição do domínio de estudo, a realização do primeiro diagnóstico, a definição das pessoas a envolver, a formação dos participantes e finalmente o planeamento dos recursos para a fase seguinte. Surge então a fase de especificação orientada ao utilizador (figura 3.6), consistindo na construção do modelo conceptual do sistema existente (fase de análise) e da sua transformação no modelo conceptual do sistema futuro (fase de projecto).

A segunda é uma fase mais orientada à especificação técnica, sendo então realizada a "tradução" do modelo conceptual do sistema futuro em três categorias de especificação: Tecnologia da Informação (hardware e software), Tecnologia de Fabrico (equipamento de fabrico) e Organização (sistema físico e estrutura de gestão) [26].

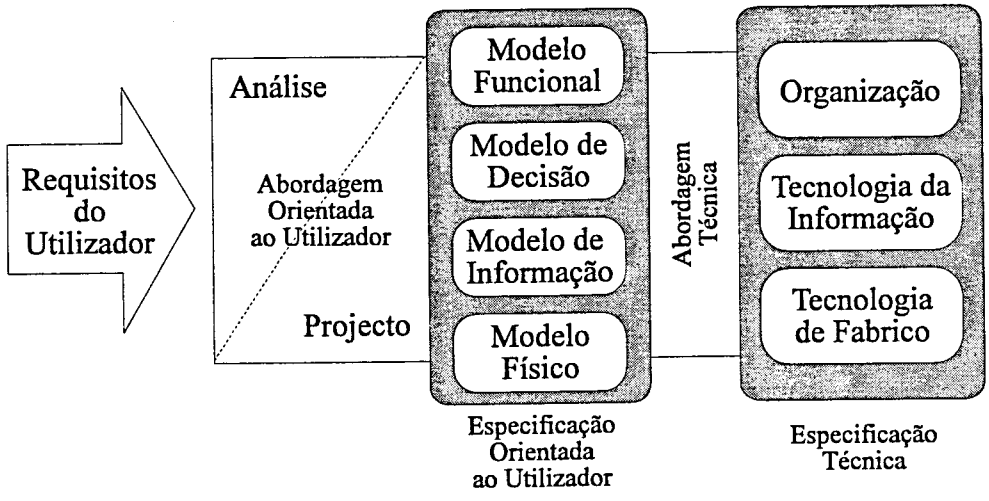


Figura 3.6 - A abordagem estruturada da GIM

O Modelo de Referência do Sistema CIM: o Modelo GRAI

Além dos aspectos metodológicos, a GIM tem associada um modelo de referência que estrutura de uma forma genérica o sistema a ser estudado. Utiliza assim o modelo GRAI, o qual suporta o projecto de um sistema integrado de fabrico através de vários formalismos de modelação que permitem a descrição dos diversos componentes do sistema.

O objectivo do modelo GRAI é dar uma descrição genérica do sistema de fabrico focando a parte de controlo do sistema (gestão da produção, no sentido lato). O controlo do sistema de fabrico é tratado de início de um ponto de vista global, e depois ao nível dos centros de decisão. O modelo GRAI é suficientemente genérico para ser usado em qualquer sistema de fabrico (discreto ou contínuo) e ainda nos serviços.

O Modelo GRAI baseia-se na teoria dos sistemas, na teoria do controlo e na teoria do controlo hierárquico [26]. Ao nível conceptual, o modelo GRAI é decomposto em três sistemas (figura 3.7): o Sistema Físico, o Sistema de Decisão e o Sistema de Informação.

No Sistema Físico, as matérias primas e os componentes são transformados em produtos, traduzindo-se a sequência de transformações num fluxo de materiais através dos meios de produção (máquinas e equipamentos) organizados de acordo com diversos critérios de agrupamento.

O Sistema de Decisão está estratificado segundo determinados critérios em níveis de decisão, cada um deles composto por um ou vários centros de decisão (CD na figura 3.7). Nesta organização em camadas temos naturalmente diferentes horizontes temporais, apresentando os níveis mais elevados comportamentos de características dominantemente periódicas. O nível mais baixo da hierarquia de decisão, onde se encontra a interface com o sistema físico, tais como os sistemas de controlo numérico ou os controladores lógicos programáveis, apresenta um comportamento sobretudo conduzido por eventos constituindo o denominado o Sistema Operativo.

A figura 3.7 ilustra a decomposição hierárquica dos centros de decisão. Na definição da hierarquia de decisão do sistema devem existir critérios de decomposição. Assim sendo,

consideram-se dois eixos ortogonais, correspondendo o vertical ao eixo de coordenação entre níveis distintos, e o horizontal ao eixo de sincronização entre operações, decisões ou funções num mesmo nível. O critério de decomposição para o eixo vertical ou de coordenação é simplesmente temporal e leva em consideração o horizonte e o período da decisão. O horizonte caracteriza o período de validade da decisão, sendo o período o intervalo de tempo ao fim do qual deverão ser tomadas novas decisões, surgindo deste modo o conceito de horizontes deslizantes. O critério para o eixo horizontal ou de sincronização é um critério funcional e resulta da teoria da gestão da produção, onde são identificadas três actividades funcionais: de gestão do produto, de planeamento do processo e de gestão de recursos.

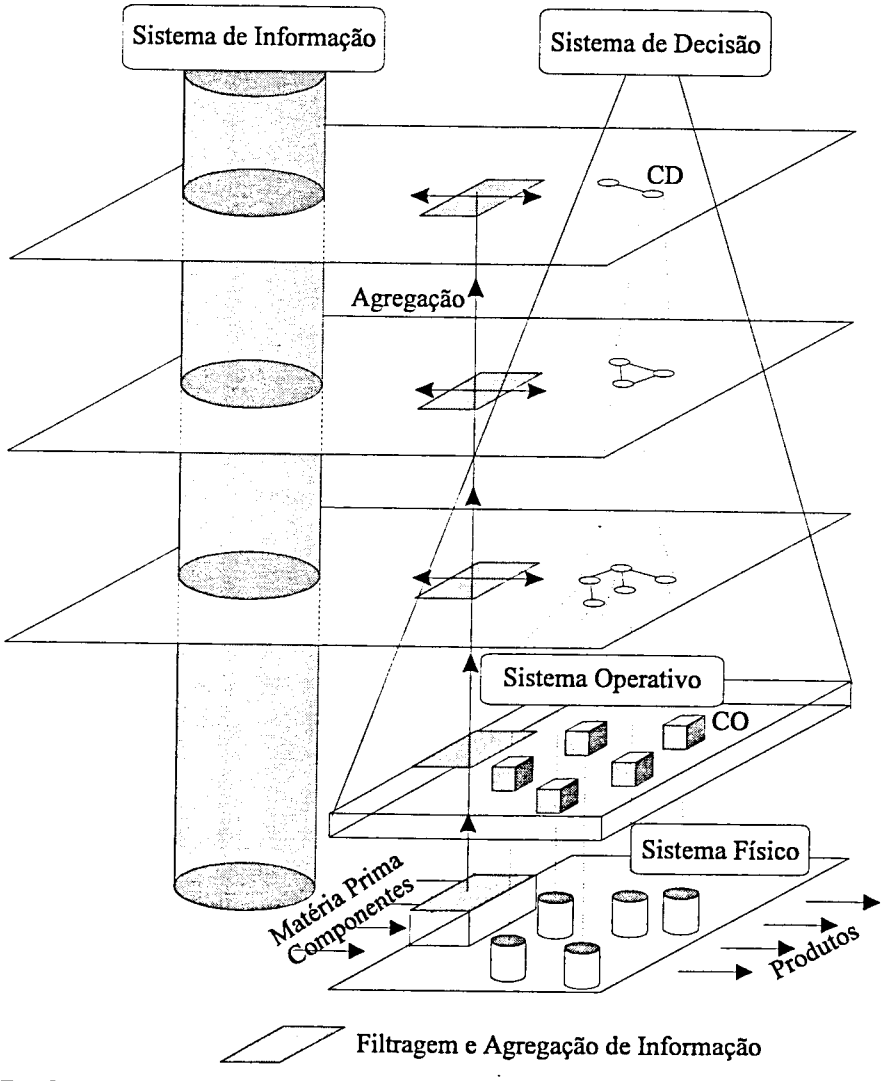


Figura 3.7 - O Modelo Conceptual Global do GRAI (os CD são descritos na figura 3.8)

O Sistema Operativo é composto por elementos básicos do sistema de decisão chamados *Centros de Operação* (CO), não sendo aqui adequada a utilização de critérios temporais usados nas actividades periódicas. A este nível, o critério para a decomposição conduzida por eventos é a complexidade da sincronização entre os fluxos de materiais e a disponibilidade dos recursos e matérias primas ou componentes. Cada CO está intimamente relacionado com a utilização dos recursos e participa em diversas funções. Cada CO (figura 3.7) deve assim sincronizar todas as actividades funcionais através do eixo de sincronização definido para as actividades periódicas associadas. Dada a proximidade entre os CO e o

processo físico é definido um terceiro eixo de decomposição, o "eixo do processo físico" a partir do qual é possível localizar no processo cada um dos centros de operação (a ponteado na figura 3.7, ligando os CO no Sistema Operativo aos elementos correspondentes no Sistema Físico).

O Sistema de Informação contém toda a informação necessária pelo sistema de decisão, a qual deve ser estruturada hierarquicamente de acordo com a estrutura dos centros de decisão. Cada CD pode para qualquer nível ser decomposto usando o mesmo critério de decomposição: elemento físico, elemento de decisão e elemento de informação (figura 3.8). O elemento físico (tratando-se de facto do sistema físico controlado) apresenta a vista do sistema físico no nível de decisão considerado, uma vez que por exemplo o encarregado não vê o sistema de produção da mesma forma que o gestor da produção. O bloco de decisão é composto por um elemento de decisão (que pode ser automático ou ser realizado pelo operador humano), juntamente com um conjunto de estruturas que o ajudam à tomada de decisão. O papel deste elemento traduz-se na comparação entre o resultado esperado e o obtido no nível inferior, ajustando se necessário, ou solicitando o ajuste ao nível superior se tal não for possível neste nível. No Sistema de Informação, cada nível recebe informação agregada dos níveis inferiores para ser utilizada pelo CD do nível em questão (figura 3.8). Cada nível é pois responsável pela agregação da sua informação e pelo seu envio ao nível imediatamente superior.

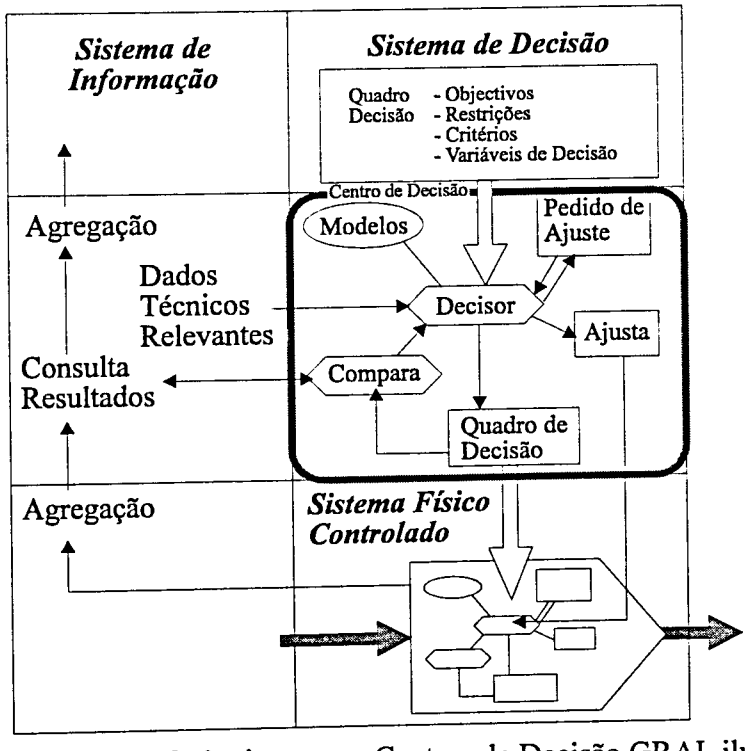


Figura 3.8 - Modelo de referência para os Centros de Decisão GRAI, ilustrando a sua integração com o Sistema de Informação e o Sistema Físico

O Quadro de Modelação do GIM

No projecto de um "Sistema Avançado de Fabrico" (SAF), a complexidade exige estruturação. Devem ser definidos os vários conceitos de suporte e os modelos a serem construídos. Com o objectivo de assegurar a consistência, que o modelo está completo e ainda a integração entre os conceitos e os modelos, a metodologia GIM propõe um quadro

de modelação onde são posicionados todos os modelos necessários para a análise, projecto e implementação do SAF. O quadro de modelação estabelecido pela metodologia GIM é apresentado na figura 3.9 e definido segundo duas dimensões, as vistas e os níveis de abstracção.

Como foi referido, o modelo GRAI está dividido em três sistemas: o sistema físico, o sistema de decisão, e o sistema de informação, cada um deles fornecendo respectivamente as vistas física, de decisão e de informação. A estas três vistas do modelo GRAI, a metodologia GIM adicionou a vista funcional, caracterizando as principais funções do sistema de fabrico assim como os fluxos (de qualquer tipo) trocados entre elas.

O quadro de modelação para cada um destes sistemas do modelo GRAI estabelece três níveis de abstracção para a sua análise estruturada, o nível de conceito, de estrutura e de realização. O nível de conceito é construído sem qualquer consideração organizacional ou técnica e destina-se a responder ao "o quê?"; o nível de estrutura por seu lado já integra informação organizacional procurando responder às questões "quem?", "quando?" e "onde?"; o nível de realização é o mais específico, integrando já restrições do ponto de vista técnico e procura responder ao "como?". Como ilustra a figura 3.9, o GIM não integra ainda qualquer método para o tratamento deste último nível de abstracção, apresentando apenas orientações para a fase de implementação.

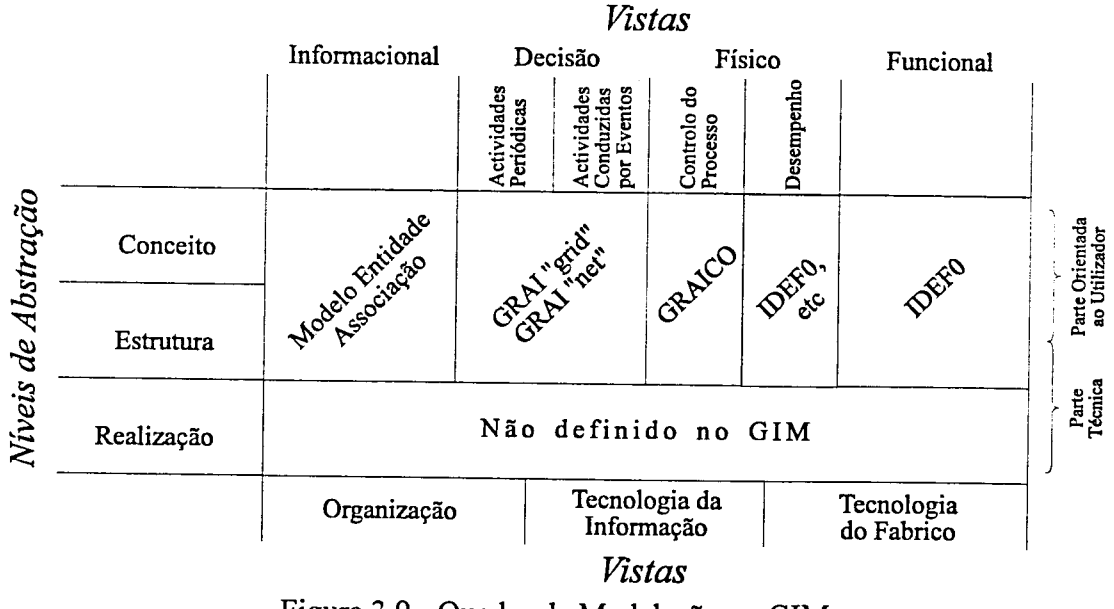


Figura 3.9 - Quadro de Modelação no GIM

Como pode observar-se na figura 3.9, a vista de informação traduz-se na realização de um modelo conceptual da informação através do modelo entidade-associação. O modelo de decisão é por seu lado caracterizado através das grelhas GRAI ("GRAI grids") ao nível global e pelas redes GRAI ("GRAI net") ao nível do detalhe, no seu conjunto usadas na definição da estrutura do modelo de decisão. Na vista física e no nível conceptual, onde se caracterizam os componentes do sistema em termos de funcionalidades, são usados os formalismos IDEF0 e S/R (Stock/Recurso, "Stock/Resource"), assim como Tecnologia de Grupo, Análise do Fluxo da Produção (AFP), e técnicas de simulação. Numa primeira fase o IDEF0, o S/R, a simulação e a AFP são usados para na determinação dos elementos básicos e da estrutura por forma a seja dada resposta aos requisitos de desempenho, sendo o GRAICO usado posteriormente na definição da estrutura óptima de controlo [26,32,33].

Conclusão

O GIM fornece uma especificação do sistema em função dos requisitos do utilizador em termos de organização, tecnologia da informação e tecnologia do fabrico. A abordagem estruturada proposta pelo metodologia GIM é constituída por quatro fases - iniciação, análise, projecto e implementação - que permitem a construção do novo sistema. Uma das principais características desta metodologia é divisão da fase de projecto em duas sub-fases: a do projecto orientado ao utilizador e a do projecto técnico (figura 3.9). Esta separação foi realizada por forma a poupar aos utilizadores a necessidade da compreensão dos detalhes do projecto técnico, apenas necessários na fase de desenvolvimento e implementação do futuro sistema. Aos utilizadores é pedida a validação da "especificação orientada ao utilizador" para que seja garantida uma solução eficiente e adequada para os seus requisitos / problemas.

3.2.4 CIMOSA

A indústria do fabrico está actualmente na fase final da migração de um paradigma de produção em massa e da automação do fabrico para os novos paradigmas que elegem o fabrico flexível e a gestão da mudança como alvos cruciais a perseguir. Estas modificações geram uma crescente necessidade de informação actualizada, no local exacto, no momento apropriado, e na quantidade e formato necessários. Esta informação, criada nos mais variados locais ao longo do ciclo de vida do produto, é necessária em muitos outros locais para que se realize uma eficiente execução e conclusão de todos os processos do negócio.

A forma mais eficiente de identificar a informação necessária e de obter o acesso a essa mesma informação, é através da utilização de modelos de operação da empresa cobrindo não só os referidos processos mas ainda as suas relações internas e externas. No entanto, para que se atinja a suficiente flexibilidade na manutenção dos modelos e na sua contínua actualização, eles deverão tornar-se numa ferramenta não só de suporte ao planeamento das actividades mas também à execução. Para que este objectivo seja atingido, a engenharia do modelo deve apoiar-se em standards que garantam a sua inteligibilidade nos diferentes sectores industriais e, mais importante ainda, a sua reutilização e transferência.

Foi neste contexto que o consórcio europeu ESPRIT AMICE desenvolveu, validou, verificou e introduziu na indústria e nos comités de estandardização os conceitos de engenharia empresarial de gestão e de controlo operacional e monitorização suportados por modelos baseados na arquitectura CIMOSA ("Open Systems Architecture for CIM") [22,23,25,34,35]. Estes conceitos foram ainda validados por diversos projectos ESPRIT, associações internacionais como IFIP e IFAC, organizações independentes em vários países (China, França, Alemanha, Hungria, Japão, Suíça, e outros), e finalmente pelas organizações membros do consórcio AMICE. Grupos de estandardização nacionais, europeus e internacionais (CEN, ISO) iniciaram também trabalho para o estabelecimento de standards baseados nestes conceitos.

Desde 1986 o consórcio AMICE, constituído por parceiros onde estão representados os principais vendedores de tecnologia da informação e grandes empresas industriais europeias, obteve vários contratos para a realização de projectos de I&D no âmbito do programa ESPRIT. Ao longo destes anos, desenvolveu os conceitos do CIMOSA, e providenciou um conjunto básico de blocos construtivos a serem utilizados na modelação empresarial e no controlo e monitorização da empresa com base nesses modelos. O trabalho realizado inclui ainda a definição de uma infra-estrutura integradora, assim como a

especificação de requisitos para as ferramentas de modelação do CIMOSA. A infra-estrutura e as ferramentas do CIMOSA têm vindo a ser validadas em vários casos de estudo nas empresas do consórcio, do sector aero-espacial, automóvel, e fornecedores de tecnologias da informação. Além disso, outros projectos europeus, tais como o CIMPRES (EP5532), CODE (EP5499) e VOICE (EP5510), têm vindo a validar os conceitos desenvolvidos pelo CIMOSA. Estes projectos partilham os seus resultados e experiências através de um grupo de interesse do "CIM-EUROPE" organizado e patrocinado pela Comissão das Comunidades Europeias.

O projecto CIMOSA produziu informação para grupos de estandardização, tanto nas áreas da modelação empresarial como das infra-estruturas de integração. Neste último caso, são exemplo os trabalhos de estandardização realizados no âmbito do OSI ("Open Systems Interconnection") e ODP ("Open Distributed Processing"). Foi assim atingido um quadro de estandardização europeu preliminar (ENV 40 003 [36]) que serviu de base para os trabalhos de estandardização internacional da ISO TC 184 SC5 WG1 [37]. Esforços recentes na cooperação no contexto da integração empresarial entre a União Europeia e os Estados Unidos levaram a que o CIMOSA fosse reconhecido como uma considerável contribuição para a integração empresarial.

O CIMOSA desenvolveu um trabalho pré-normativo para a modelação empresarial baseada em processos e na aplicação destes modelos na gestão, controlo e monitorização da empresa. A modelação baseada em processos traduz uma forma excelente de estruturar e identificar a informação criada e usada durante o processo de fabrico permitindo torná-la disponível a partir qualquer local da empresa em qualquer instante e na quantidade e formato necessários para uma determinada fase do processo.

As empresas modernas são muitas vezes constituídas por diversas áreas funcionais isoladas. As interacções entre estas áreas funcionais podem ser limitadas como resultado de estruturas de dados incompatíveis e / ou entradas e saídas de funções também incompatíveis, com a consequente duplicação de serviços e de informação. Estes problemas levam ao aparecimento de dificuldades operacionais e de gestão, reduzindo a capacidade da empresa para responder ao ambiente em mudança. Uma das formas de ultrapassar estes problemas permitindo um processamento da informação consistente, é adoptar uma infra-estrutura de informação que promova a flexibilidade e a eficiência operacional. O CIMOSA fornece tal infra-estrutura, providenciando vários serviços comuns às múltiplas aplicações (por exemplo, gestão de dados e comunicações). Fornecendo também uma metodologia de modelação, permite por outro lado que a operação da empresa seja descrita de uma forma global e estruturada. Este modelo pode ser utilizado para monitorizar o desempenho da empresa, assim como para avaliar o resultado de eventuais propostas de alteração da infra-estrutura.

Quadro de Modelação Empresarial

O objectivo do quadro de modelação empresarial do CIMOSA é o de fornecer um conjunto de linguagens com uma semântica unificada suportando a descrição e a integração empresariais. Isto significa que para integrar dois componentes de um sistema CIM (sejam aplicações, máquinas ou pessoas), estes componentes devem partilhar uma definição comum dos objectos que trocam entre si, i.e. deverão "falar" linguagens compatíveis. Existe deste modo uma necessidade para um referencial comum (ou mecanismos de identificação semântica) para a integração empresarial, além de uma infra-estrutura de integração. É isto que caracteriza o quadro de modelação do CIMOSA.

Quando a empresa é modelada, existem vários aspectos e pontos de vista que devem ser estruturados e que não podem ser examinados num quadro unidimensional. O CIMOSA identifica um quadro tridimensional oferecendo a capacidade de modelar os diferentes aspectos da empresa segundo três eixos (figura 3.10): os blocos construtivos, os modelos e as vistas.

O quadro de modelação é apresentado através da Arquitectura de Referência do CIMOSA (figura 3.10) em três níveis de modelação: o nível genérico, o parcial e o particular. O eixo dos Blocos Construtivos tem por objectivo o reforçar da utilização de uma linguagem de descrição comum (definida pelos blocos construtivos genéricos). Força deste modo a utilização de componentes standard, bem como a reutilização sempre que possível, dos modelos parciais existentes reduzindo-se assim o ciclo de desenvolvimento dos módulos. O quadro de modelação do CIMOSA fornece uma arquitectura de referência constituída por um catálogo de blocos construtivos e agregações de blocos construtivos, os chamados modelos parciais reutilizáveis. Introduz ainda o conceito de Vistas, que permite que se trabalhe num subconjunto do modelo e não sobre o modelo total, o que contribui para a redução da complexidade vista pelo utilizador. A decomposição em vistas, traduz o facto de a empresa dever ser modelada em termos de funções (do seu sistema físico e de decisão) que utilizam dados e informação, assim como materiais, componentes e recursos (o seu sistema físico) que operam sob responsabilidades definidas (como parte do sistema de decisão). A modelação no CIMOSA está assim estruturada por forma a descrever a operação de uma empresa, ou de parte dela, através de quatro vistas separadas mas inter-relacionadas da empresa: a Vista Funcional, a Vista de Informação, a Vista dos Recursos e a Vista Organizacional.

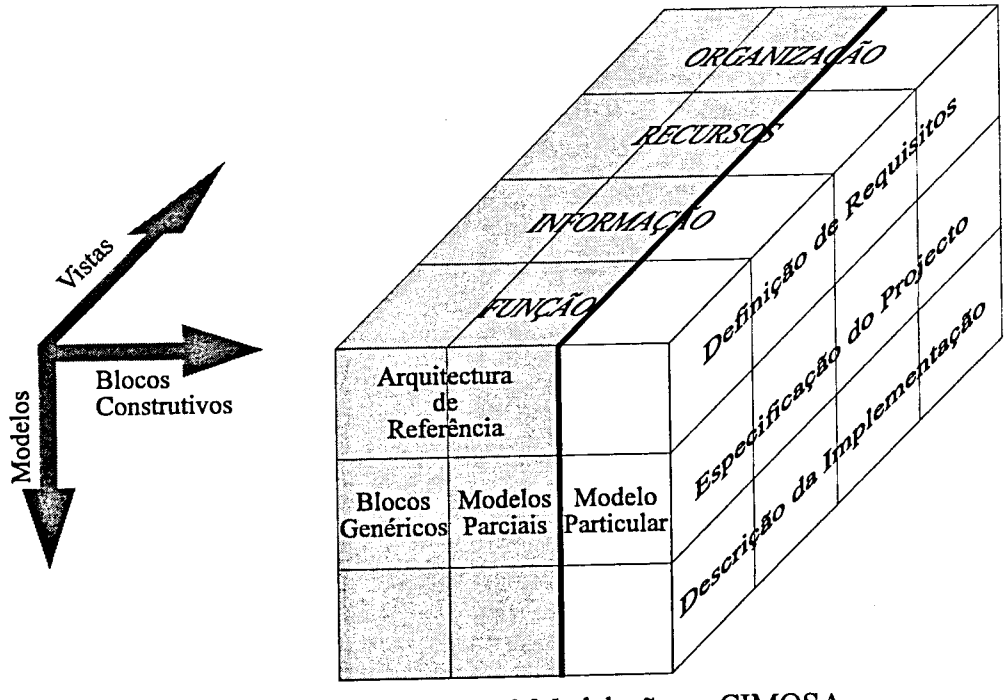


Figura 3.10 - Abordagem à Modelação no CIMOSA

O quadro de modelação do CIMOSA providencia as linhas mestras de orientação para a criação de modelos da empresa e do correspondente sistema integrado de fabrico de uma forma coerente. A abordagem de modelação do CIMOSA é baseada na Arquitectura de Referência a partir da qual são derivadas as Arquitecturas particulares e desenvolvidos os modelos da empresa.

A Arquitectura de Referência do CIMOSA suporta a Modelação de todo o Ciclo de Vida das operações empresariais (apoiando a Definição de Requisitos, a Especificação do Sistema e a Descrição da Implementação), sendo o objectivo deste eixo impor uma abordagem estruturada ao processo de desenvolvimento e implementação do Modelo derivado da definição dos requisitos

CIMOSA, Modelação Empresarial [35]

De forma a limitar o denominado UoD (Universo do Discurso), i.e. o âmbito de um modelo particular, o CIMOSA fornece ao utilizador a possibilidade de definir Domínios, i.e. áreas consistentes da funcionalidade do negócio que podem ser facilmente integradas. O CIMOSA vê uma empresa como um sistema dinâmico constituído por Domínios (i.e., subsistemas) contendo Processos do Domínio (PD) interactuantes, gerando, consumindo ou simplesmente usando Eventos da Empresa (EV) e objectos (sob a forma de vistas de objectos). Os PD são activados por eventos (sejam eles acontecimentos do mundo real solicitados ou não). Assim sendo, o paradigma de modelação do CIMOSA é baseado em processos conduzidos por eventos.

A figura 3.11 ilustra a forma como o CIMOSA vê a modelação da empresa. O Domínio (DMx) captura objectivos e restrições particulares ao UoD. As suas fronteiras são identificadas por conjuntos de relações entre domínios. As funcionalidades globais dos domínios são identificadas pelos Processos do Domínio (PDx.y). São ainda identificados os eventos que disparam os PD e os objectos (descritos pelas vistas de objectos) que atravessam a fronteira do domínio. O encapsular dos PD no interior dos domínios assegura ainda a consistência entre processos CIMOSA e não-CIMOSA, facilitando assim a integração com sistemas legados e a migração de uma situação existente para uma situação compatível com as soluções do CIMOSA. Os domínios e os PD são blocos construtivos essenciais do CIMOSA, sendo indispensáveis à definição de ambientes estruturados de fabrico integrado.

Os PD definem a maior parte das funcionalidades e do comportamento da empresa. São processos individuais, disparados por eventos que controlam a forma como são executadas as Actividades da Empresa (AE) de acordo com o chamado Conjunto de Regras de Procedimento (CRP - "Procedural Rule Set"). O CRP representa assim o comportamento do PD, representando as AE as suas funcionalidades. Os objectivos e as restrições (Regras Declarativas) representam o conhecimento do negócio (i.e., procedimentos, políticas e regras). As AE usam, processam e criam Objectos da Empresa (OE) (objectos físicos ou de informação) definidos no modelo como Vistas de Objectos (VO). Com o objectivo de facilitar a estruturação do modelo CIMOSA, é ainda introduzido o conceito de Processo do Negócio (PNx.y.z), que torna possível a decomposição do PD em sub-processos reutilizáveis. Deste modo, o PD completamente definido traduz-se numa rede de AE ligada por um CRP, surgindo uma característica interessante do CIMOSA: o fluxo de controlo.

As VO traduzem aspectos específicos dos objectos (incluindo os objectos de recursos) apresentando-se sob a forma como são vistos pelo utilizador e usadas pelas funções da empresa. Os objectos são formalmente descritos como Objectos da Empresa (OE). As VO podem ser construídas sobre VO (que serão sub-vistas), e OE podem ser construídos a partir de outros OE (que serão sub-objectos). Ambos contêm Elementos de Informação, que são propriedades elementares dos objectos. Estes elementos de informação estão sujeitos a regras de integridade responsáveis pela verificação das restrições aplicáveis à informação.

As VO podem ser utilizadas na representação de objectos físicos (i.e. o próprio objecto físico) ou de objectos abstractos (i.e. informação). Por exemplo, a parte física de uma

máquina ferramenta é uma VO Física, enquanto que a sua descrição no sistema de informação é uma VO de Informação do mesmo OE. As VO podem ser utilizadas no modelo do CIMOSA como entradas funcionais ou de controlo, ou ainda como saídas funcionais das AE (e portanto também dos Processos do Negócio e do Domínio) (R na figura 3.11). Uma grande vantagem desta facilidade é o facto de o CIMOSA poder facilmente representar: o fluxo de informação sob a forma de diagramas de fluxos de dados se só for considerada a VO de Informação; ou fluxos de materiais (e.g. peças e materiais, etc.) se só forem considerados VO Físicos, sendo este segundo ponto apenas relevante no modelo de definição de requisitos.

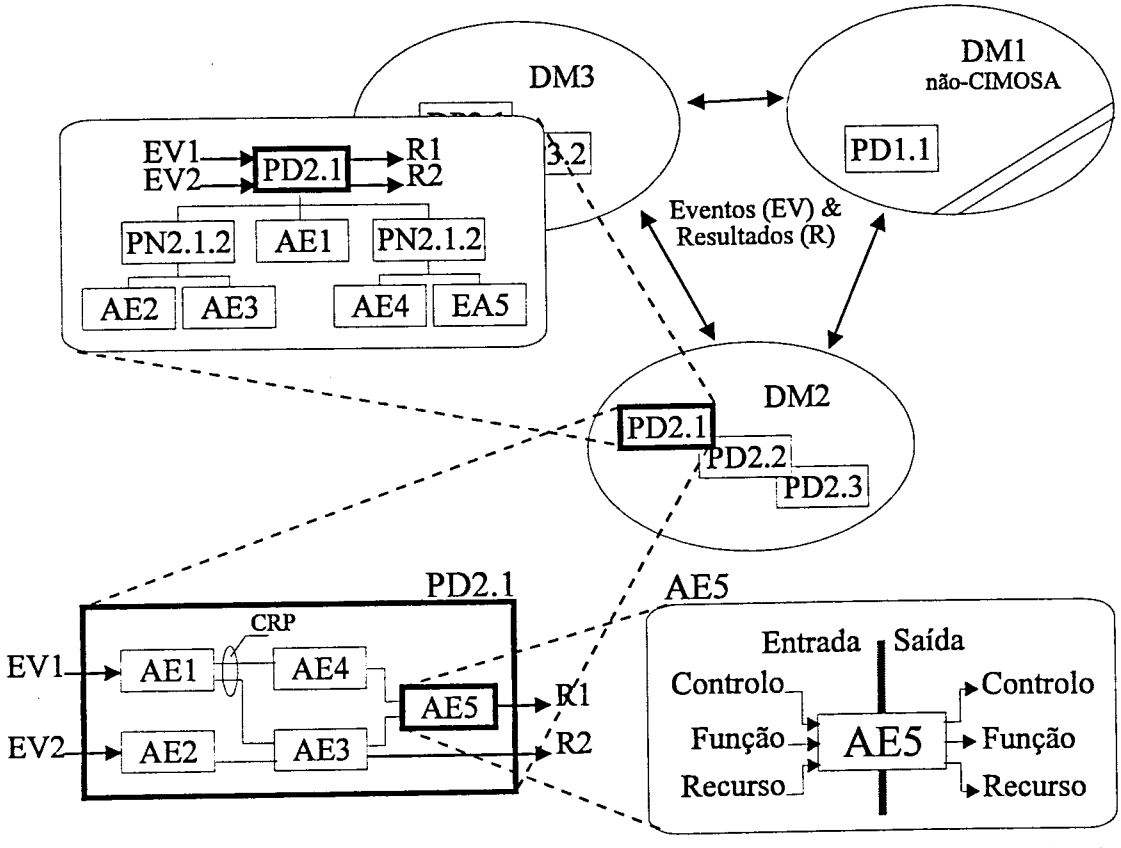


Figura 3.11 - Decomposição de um Processo de Domínio numa rede de Actividades da Empresa

O Âmbito da Modelação no seio da Empresa

A definição do âmbito de um modelo particular é da responsabilidade do utilizador. O CIMOSA está todavia concentrado na integração do sistema de fabrico, i.e. pretende modelar as funcionalidades que devem ser integradas e controladas durante a sua execução. O CIMOSA não impõe no entanto quaisquer restrições ao âmbito do modelo no contexto da empresa, podendo ser usado na modelação de toda a empresa ou apenas das áreas seleccionadas, os Domínios.

O nível de detalhe do modelo é também deixado ao critério do utilizador, permitindo o CIMOSA a construção de modelos mais ou menos detalhados conforme as necessidades. Uma definição pouco detalhada implica que apenas são definidos os domínios, identificados os seus PD bem como os eventos que os disparam, além das principais AE e CRP. No modelo detalhado, a granularidade será necessariamente maior, sendo modelados os Domínios e as suas relações, todos os PD e respectivos eventos de disparo, assim como a

sua estrutura interna em termos de AE. As regras declarativas devem ainda ser completamente definidas de acordo com as necessidades do processo de monitorização e de controlo da operação.

O Processo de Desenvolvimento do Modelo

O modelo particular tem uma relação com o mundo exterior (através de Eventos e de Vistas de Objectos), traduzindo uma descrição estática da funcionalidade da empresa bem como o seu comportamento dinâmico. O modelo particular é visto como sendo desenvolvido ao longo das seguinte quatro fases derivadas umas das outras, e para as quais o CIMOSA fornece blocos construtivos aplicáveis:

- Definição de requisitos, da qual resulta o Modelo de Definição de Requisitos (MDR)
- Especificação do projecto do sistema, verificando os requisitos estabelecidos, de que resulta o Modelo de Especificação do Projecto (MEP) derivado do anterior MDR.
- Descrição da implementação do sistema de acordo com opções de projecto e resultando no Modelo de Descrição da Implementação (MDI). Mais uma vez, o MEP é completado por forma a traduzir o sistema final a ser utilizado na operação.
- Emissão formal do modelo, seguida da validação e verificação de que ele fornece o modelo adequado a ser utilizado na condução da operação, controlo e monitorização da empresa.

Na criação do modelo particular, a sequência definida traduz a abordagem topo-base com o objectivo do projecto e análise do sistema futuro ("TO-BE"). É assim a sequência adequada à definição de um novo sistema de fabrico. A abordagem base-topo também podem ser aplicada na modelação do sistema presente ("AS-IS"), com o objectivo de avaliar soluções alternativas. Neste caso, a análise começa com a descrição da implementação actual a partir da qual são avaliados projectos de mudança alternativos.

A Infra-estrutura de Integração

O MDR é reestruturado, optimizado e simulado ao nível da especificação do projecto, só depois sendo traduzido no MEP e MDI. Este modelo de implementação é um modelo executável sobre uma infra-estrutura de integração. A infra-estrutura de integração, além de fornecer os mecanismos para a integração transparente de componentes heterogéneos (acesso transparente à informação, e gestão da infra-estrutura de comunicações), providencia também os meios para controlar (gerir) o ciclo de vida do processos através da execução de modelos compatíveis com o CIMOSA. Esta gestão dos modelos executáveis suportando o coordenação da empresa, garante a execução consistente dos Processos do Domínio concorrentes e é realizada pelo seguinte conjunto de entidades que serão tratadas em detalhe no próximo capítulo: a Entidade de Informação, a Entidade do Negócio, a Entidade de Apresentação, a Entidade de Serviços Comuns e a Entidade de Gestão do Sistema.

A Interacção do CIMOSA com o Mundo Real

O CIMOSA distingue explicitamente o ambiente de engenharia da empresa e o seu ambiente de operação, bem como as tarefas associadas à actualização e manutenção da empresa, sublinhando a necessidade da engenharia da empresa da mesma forma que é necessária a engenharia do produto. Assim sendo, as fases do ciclo de vida da engenharia da empresa poderão traduzir-se na emissão de novas versões de processos de operação, incluindo tanto a implementação como a validação do modelo.

A figura 3.12 ilustra a utilização do CIMOSA na engenharia do modelo assim como no controlo da operação e monitorização. Usando a Arquitectura de Referência são produzidos os Modelos Particulares da Empresa sob o controlo da Engenharia do Modelo de Implementação da Empresa. Este último modelo será construído usando ferramentas CAE que apoiam o ciclo de vida do sistema CIMOSA. O Modelo Particular de Implementação uma vez completo é usado directamente na condução das operações, monitorizando e controlando todo ou parte do ciclo de vida do produto assim como a implementação dos correspondentes processos do negócio. A infra-estrutura de integração estabelece a ligação aos recursos da empresa, sendo esta ligação indispensável à execução do modelo assim como à sua manutenção em tempo-real (actualizações e extensões).

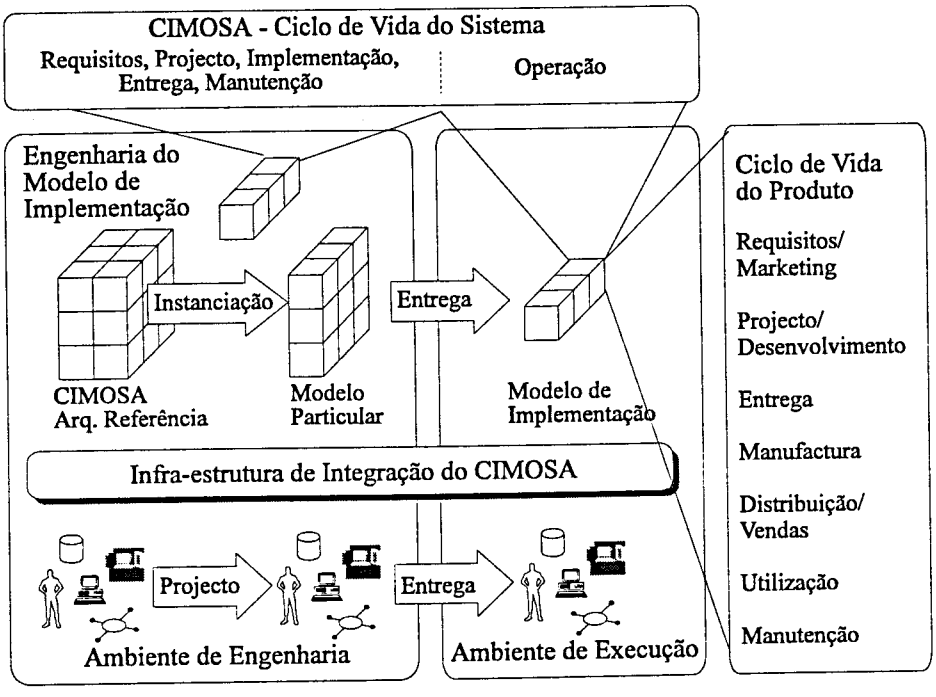


Figura 3.12 - A Aplicação do Conceitos do CIMOSA - ambientes de engenharia e de operação

Validação do CIMOSA, Aplicações

O CIMOSA permite elevada flexibilidade nas actividades de engenharia empresarial pela rápida construção e avaliação de modelos, através da simulação de situações alternativas de operação e da implementação directa da solução final. Estes factos foram explicitamente comprovados pelo projecto AMICE no decorrer de uma aplicação piloto numa fábrica da FIAT. Na comparação com a aplicação de outras metodologias estado-da-arte à divisão de fabrico e montagem de caixas de velocidades da FIAT os benefícios do CIMOSA foram evidentes [25]: melhor estruturação e maior detalhe relativos à informação, recursos e organização; menor tempo de modelação (cerca de três vezes menor); maior reutilização de partes do modelo (para uma aplicação semelhante o tempo de modelação pode ser reduzido cerca de oito vezes).

Os conceitos do CIMOSA foram validados e verificados num total de oito implementações piloto realizadas pelos projectos ESPRIT AMICE, CIMPRES, CODE e VOICE, cobrindo tanto a parte da engenharia do modelo como o controlo e monitorização da operação baseada no modelo.

Conclusão

O CIMOSA fornece um quadro completo de suporte à engenharia e à posterior operação e re-engenharia de sistemas de fabrico complexos. A tabela 3.2 apresenta algumas vantagens e desvantagens da sua utilização [38]. No coração do CIMOSA existe por um lado o modelo integrado da empresa capturando e estruturando as características relevantes do sistema integrado de fabrico, e por outro a infra-estrutura de integração suportando a operação da empresa de uma forma integrada.

Tabela 3.2 - Vantagens e Desvantagens da utilização do CIMOSA de acordo com [38]

Vantagens do CIMOSA	**Desvantagens do CIMOSA**
Abrange todo o ciclo de vida do sistema.	Metodologia complexa.
Cobre aspectos funcionais, de informação, de recursos e organizacionais.	A verificação da consistência, coerência bem como a comprovação de que o modelo está completo é realizada com base em regras.
Desenvolvimento de um modelo executável.	Inter-relação entre as vistas [38].
Ambiente de engenharia e de execução.	Definição pobre da simulação.
Decomposição funcional e descrição do comportamento.	Infra-estrutura de integração ainda em discussão.
Arquitectura de referência.	Falta de exemplos concretos.

No decorrer do desenvolvimento do CIMOSA, o consórcio AMICE estabeleceu desde 1985 um conjunto de conceitos importantes [34]:

- o Processo do Negócio (PN), conceito separado do de Actividade da Empresa (AE), usado no projecto e organização dos sistemas de fabrico, por forma a evitar a criação de ilhas de automação (princípio da separação entre a funcionalidade da empresa e o seu comportamento);
- o de modelo integrado da empresa, conduzindo à unificação semântica dos conceitos, linguagens e calão usados pelos diversos actores do sistema CIM;
- o de ciclo de vida do sistema CIM, identificando claramente o modelo de definição dos requisitos, o modelo de especificação e o modelo executável;
- o de vistas de modelação, com o objectivo de melhor tratar a complexidade do sistema através de janelas focadas sobre determinados aspectos da empresa e evitando a dispersão por outros aspectos;
- o de modelos parciais, usado na identificação de modelos comuns a empresas de sectores industriais específicos (princípio da reutilização dos modelos);
- o de modelos executáveis, usados no controlo ou pelo menos na monitorização da execução dos processos que necessitam de ser controlados por computador;
- o de técnicas de especificação formal aplicadas ao projecto e coordenação de modelos coerentes e executáveis de processos de negócio em ambientes distribuídos;
- o de infra-estrutura de integração CIM, constituída por um conjunto de serviços cooperantes com o objectivo de tornar transparente a tecnologia de informação e de reduzir o peso da heterogeneidade dos sistemas;
- o de gestão da mudança na re-engenharia do modelo da empresa acompanhando a sua evolução, um dos desafios a levar em consideração nos anos que se aproximam.

As referidas contribuições do CIMOSA constituíram valiosas fontes de informação para trabalhos em curso no âmbito de grupos de trabalho internacionais para a normalização da "integração e modelação empresarial" (ENV 40 003 [39]).

Comparada com outras abordagens à modelação tais como o IDEF [40], ou o GIM, o quadro de modelação fornecido pelo CIMOSA providencia a unificação semântica sob a forma de um único modelo integrado abrangendo as vistas fundamentais da empresa para os diversos níveis da modelação [34]. Relativamente à arquitectura ARIS [41], uma das que a seguir se aborda de uma forma breve, esta é considerada em [34] como uma derivação simplificada do CIMOSA para os sistemas de informação no fabrico.

3.2.5 Outras Arquitecturas

ARIS - "Architecture of Integrated Information Systems"

A ARIS define uma arquitectura para o desenvolvimento de sistemas integrados de informação, optimizados e convertidos em implementações adequadas ao processamento electrónico de informação [42]. A arquitectura integrada do sistema de informação do negócio é derivada do modelo geral da cadeia de processos a ele inerentes.

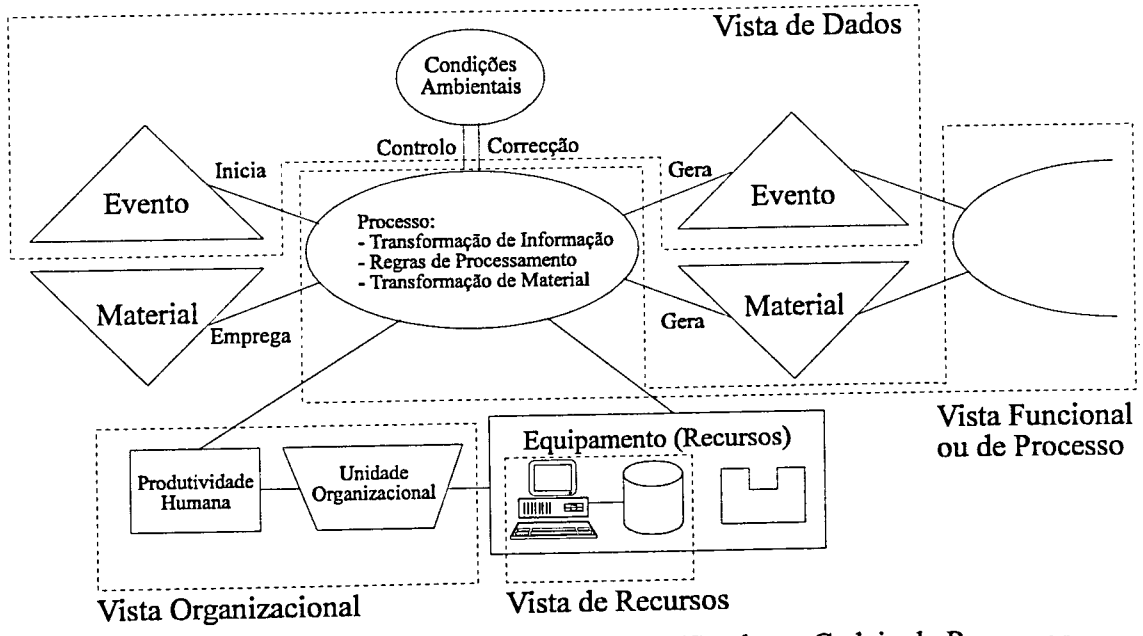


Figura 3.13 - As Vistas da Arquitectura ARIS sobre a Cadeia de Processos

A figura 3.13 ilustra uma cadeia de processos onde pode ser observada a chegada de um evento (e.g. ordem de produção #1234) assim como uma chegada de material (e.g. material M#7). O referido evento inicia o processo de produção usando a matéria prima, os recursos humanos pertencentes a uma determinada unidade organizacional e os recursos de produção (⌨). O processo de produção gera naturalmente outros eventos (por exemplo, fim de execução da ordem #1234) assim como material (peças) que poderão ser utilizados como entradas para outros processos produtivos. Como resultado da função do processo existem alterações ambientais, tais como a criação de uma nova entidade (produto, cliente) ou a alteração do valor de um atributo de uma entidade existente.

Dependendo do tipo de processo, assim poderá dominar o processamento da informação ou do material. No processamento de materiais domina a transformação da

matéria prima, enquanto ao processamento de informação estão associados processos administrativos, tais como tratamento das ordens de produção, planeamento e projecto ou contabilidade. Existe no entanto uma forte inter-relação entre o processamento de materiais e de informação, particularmente visível nos modernos processos técnicos de produção, como é o caso das máquinas de controlo numérico realimentando o sistema de informação relativamente ao andamento do processo de produção.

A arquitectura ARIS, tendo como objectivo a simplificação do modelo da cadeia de processos, exclui das suas vistas a existência individual dos materiais, dos recursos de produção e dos recursos humanos envolvidos directamente no processo produtivo. Cada um deles passa a fazer parte das condições ambientais da cadeia de processo, significando esta abordagem que, do ponto de vista do processamento da informação, o processo produtivo é apenas visto através de alterações do valor dos dados como resultado das actividades produtivas. O equipamento de suporte da tecnologia de informação é no entanto mantido no modelo dada a sua relevância na ligação entre o sistema de informação e o operador.

Na figura 3.13 é ainda possível observar a forma como surgem as vistas sobre o modelo da cadeia do processo. Os eventos, assim como as condições ambientais, são representados por dados constituindo deste modo a Vista de Dados. A descrição das regras de processamento assim como a estrutura do próprio processo constituem a Vista do Processo ou Funcional. O conjunto dos operadores e as unidades organizacionais a que estão afectados constituem por seu lado a Vista Organizacional. Temos finalmente a Vista de Recursos, onde é visível a infra-estrutura do sistema de informação.

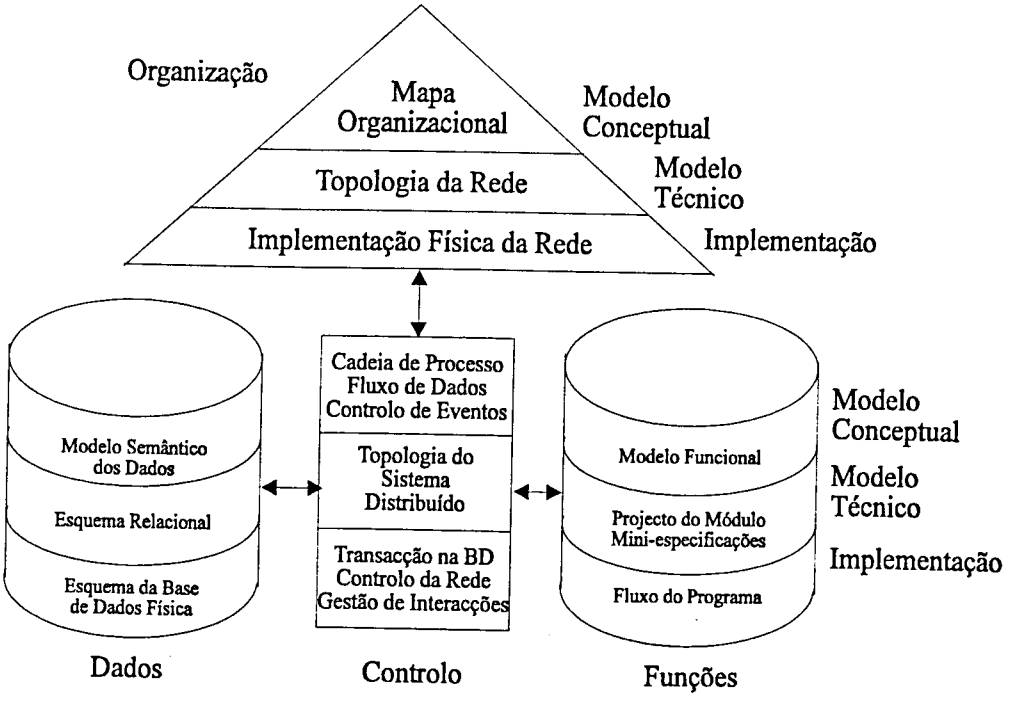


Figura 3.14 - Arquitectura ARIS

Com a formação das quatro vistas descritas, foi perdida a evidente relação entre elas. Este facto levou à criação de uma quinta vista, a Vista de Controlo, utilizada para caracterizar as relações entre os múltiplos pontos de vista do modelo (figura 3.14).

Na ARIS, é finalmente estabelecido um meta-modelo constituído por quatro elementos: o modelo geral do processo (Vista do Processo, figura 3.13), o modelo da arquitectura, o modelo de informação ARIS e o repositório de dados (figura 3.15).

O modelo geral do processo contém a estrutura básica da descrição do processo e estabelece os níveis (modelos conceptual, técnico e de implementação) que vão ser descritos na arquitectura (ARIS, figura 3.14). Para cada vista da arquitectura, os modelos de informação contêm os objectos a serem descritos assim como as suas relações, e são criados usando a linguagem descritiva do modelo entidade-associação. É assim gerado o modelo de informação ARIS a ser armazenado no repositório de informação. Neste repositório de informação é descrito o modelo das aplicações mas não o seu conteúdo, por exemplo para uma aplicação de Planeamento e Controlo da Produção (PP&C-"Production Planning and Control"), os elementos CLIENTE, PEÇA e EQUIPAMENTO, são armazenados como entidades associadas ao modelo da aplicação PP&C, enquanto que as suas instâncias específicas, ou seja a informação relativa aos clientes, às peças e aos equipamentos propriamente ditos estarão armazenados nas bases de dados das aplicações de PP&C.

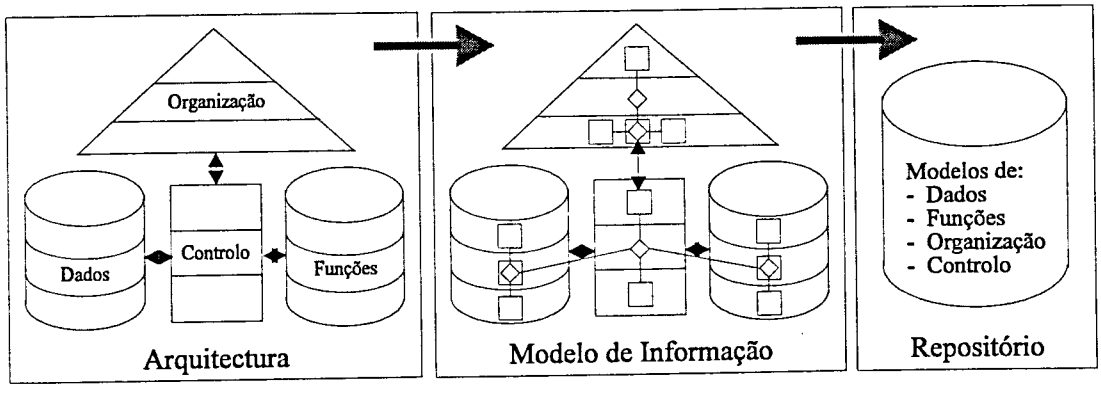


Figura 3.15 - Meta-modelo ARIS

IEM - "Integrated Enterprise Modelling"

O método de Modelação Empresarial Integrada ("Integrated Enterprise Modelling" - IEM), desenvolvido no instituto Fraunhofer-IPK (Berlim, Alemanha), utiliza a modelação orientada ao objecto na representação dos processos do negócio, das estruturas organizacionais, assim como do sistema de informação [43,44]. O IEM fornece um modelo a ser utilizado no planeamento e optimização dos processos e das estruturas organizacionais no interior da empresa.

Do ponto de vista da técnica de modelação, o IEM recorre à modelação orientada ao objecto, permitindo a integração de diferentes vistas da empresa num único modelo consistente bem como a fácil adaptação do modelo às mudanças ocorridas na empresa. As classes genéricas Produto, Recurso e Ordem, fornecem ao utilizador a base para a construção dos seus modelos IEM, sendo particularizadas na construção de novos modelos. Esta particularização pode ser realizada directamente partindo das classes básicas ou de subclasses daquelas classes básicas.

O núcleo do modelo IEM compreende duas vistas principais, a Vista sobre o Modelo do Processo e a Vista sobre o Modelo de Informação, concentrando-se a primeira vista sobre as tarefas e processos do negócio, e a segunda sobre os dados armazenados nos

objectos. Estas vistas estão naturalmente ligadas, referindo-se aos mesmos objectos e actividades, embora os representem sobre diferentes perspectivas.

No IEM, tudo o que se passa na empresa de fabrico é descrito sob a forma de uma actividade. Esta actividade, processa e modifica os objectos anteriormente classificados como Produtos, Recursos e Ordens. A execução de uma qualquer actividade implica o seu planeamento e escalonamento directo ou indirecto, sendo realizada por um Recurso com capacidades para a executar. O IEM estabelece assim três níveis para a descrição da actividade designados respectivamente por Acção, Função e Actividade (figura 3.16). A Acção é uma descrição independente do objecto de algum trabalho ou negócio, a descrição verbal de alguma tarefa, fase de processo ou procedimento. A Função descreve o processamento dos objectos, como uma transformação de um estado inicial, num outro estado dito final. A Actividade especifica a Ordem que controla a execução da Função e os Recursos responsáveis pela execução daquela Função. A figura 3.16 ilustra o Modelo de Actividade Genérico do IEM, representando os objectos das classes Produto, Ordem e Recurso e indicando a sua interacção.

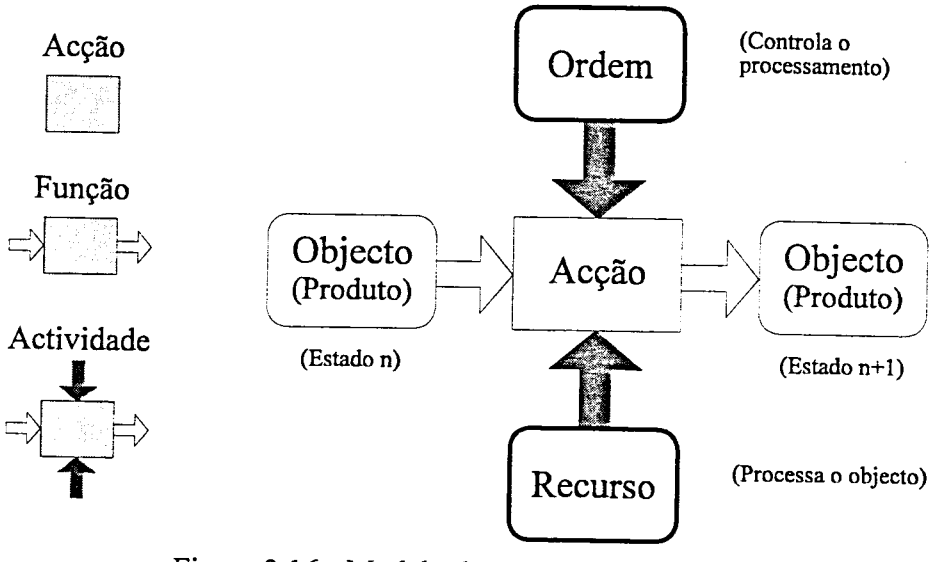


Figura 3.16 - Modelo de Actividade Genérico

Este método é actualmente suportado por uma ferramenta de modelação em MS-Windows 3.1. A sua relevância é além disso evidenciada pelo documento, elaborado pela "CIMOSA Association" e pelo IPK, comparando os blocos construtivos do CIMOSA e do IEM com o objectivo de providenciar informação ao grupo europeu de estandardização CEN TC310/WG 1 [43].

3.2.6 Discussão das Arquitecturas para a Integração das Empresas de Fabrico

Neste parágrafo são realizados alguns comentários às três arquitecturas PERA, CIMOSA e GRAI-GIM tendo como base o documento de Theodore Williams [21]. São ainda apresentados comentários relativos às arquitecturas ARIS e IEM extraídas respectivamente o livro publicado por Scheer "Computer Integrated Manifacturing, Towards the Factory of the Future"[16] onde é comparado o ARIS e o CIMOSA, e o documento "Comparison CIMOSA-IEM Modelling Constructs and methodology" realizado conjuntamente pela "CIMOSA Association" e pelo IPK [43].

CIMOSA

Ao consórcio AMICE deverá ser reconhecido o elevado mérito de ter, desde muito cedo no projecto, decidido que deveria ser o mais formal possível na definição e descrição de todos os aspectos do CIMOSA. Este princípio de base tinha por objectivo último o atingir a executabilidade em computador de todos os elementos associados à arquitectura, i.e. blocos construtivos, modelos, ferramentas, técnicas, etc.

O consórcio AMICE restringiu inicialmente o âmbito de aplicação do CIMOSA aos sistemas discretos. Além disso, restringiu a sua aplicação às instalações fabris onde todo o equipamento de produção deveria ter o seu próprio sistema de controlo. Deste modo, o CIMOSA aborda apenas os níveis de gestão da produção e do controlo supervisor. O controlo digital directo (DDC) dos diferentes processos de fabrico é tratado através dos controladores locais alimentados com instruções de operação provenientes do sistema CIMOSA.

O CIMOSA estabelece e descreve o ciclo de vida do sistema CIM. No entanto, não desenvolve suficientemente os seus conceitos para que eles se tornem verdadeiramente numa metodologia a ser utilizada juntamente com a arquitectura.

Como resultado do objectivo inicial de obter uma descrição formal da empresa passível de ser executada por computador, o CIMOSA é, de entre as três apresentadas, a arquitectura descrita da forma mais formal. Este facto coloca no entanto obstáculos à sua compreensão e facilidade de leitura por parte de potenciais utilizadores sem conhecimentos nas áreas da ciência da computação ou afins.

O CIMOSA desenvolve-se sobre dois ambientes, o Ambiente de Engenharia Empresarial e o Ambiente de Execução Empresarial, apoiando-se num conjunto de serviços de sistema básicos especificados na infra-estrutura de integração. O primeiro destes ambientes formaliza o desenvolvimento dos modelos da empresa e a sua conversão em módulos executáveis. O segundo formaliza o teste, a validação e a aceitação dos programas resultantes como adições ou alterações no sistema de informação da empresa em operação. A infra-estrutura estabelece a forma como estes módulos interagem de modo a que sejam executadas as funções do sistema integrado.

O CIMOSA é de entre as arquitecturas de referência apresentadas aquela que tem sido alvo de maior estudo e discussão, além de ser o alvo das maiores atenções com o objectivo de ser tornar um standard internacional.

GRAI-GIM

O GRAI-GIM apresenta uma metodologia já bem desenvolvida para a aplicação da sua arquitectura, assim como ferramentas de suporte ao processo de desenvolvimento e integração empresarial.

O GRAI-GIM tem vindo a restringir a documentação descritiva do sistema a discussões sobre o desenvolvimento do sistema computacional e sobre o hardware e software relacionado com a implementação.

O GRAI-GIM tem, até hoje, sido aplicado apenas no âmbito dos sistemas discretos fabrico. Por outro lado, o ciclo de vida do sistema integrado de fabrico coberto pelo GRAI-GIM ainda não abrange a construção, teste e instalação do sistema ou mesmo a sua evolução. No entanto, não existe nenhuma restrição inerente à metodologia que a impeça de ser aplicada a um outro tipo de indústria, ou de cobrir o restante ciclo de vida.

Tal como o CIMOSA, o GRAI-GIM trata o elemento humano como mais um recurso com determinadas funcionalidades e capacidades físicas, e não são igualmente

contempladas as relações humanas, os programas de treino, os requisitos dos sindicatos, ou os detalhes da organização humana.

No desenvolvimento do GRAI-GIM, foi desenvolvido no laboratório GRAI um conjunto de ferramentas e técnicas com grande potencial em estudos de integração empresarial, tais como GRAI-GRID, GRAI-NET, ECOGRAI e o modelo GRAI entre outros. Tem ainda utilizado técnicas e ferramentas desenvolvidas por outros, tais como o IDEF0 [26].

Os modelos descritivos do GRAI-GIM são sem qualquer dúvida os melhores no caso de se desejar obter um conjunto de documentos para o desenvolvimento de um projecto de integração do sistema de fabrico.

O GRAI-GIM poderá ser classificado como estando entre o modelo de Purdue e o CIMOSA no que se refere ao grau dos formalismos usados e à consequente facilidade de compreensão para um não especialista em ciências da computação.

Purdue

A arquitectura de referência de Purdue, bem como a metodologia associada, constitui uma forma informal de descrever como os seus utilizadores devem progredir ao longo das diversas fases do processo da integração da empresa.

Por ser uma descrição informal é de todas as arquitecturas apresentadas a de mais simples compreensão, não exigindo qualquer conhecimento anterior nas áreas da ciência da computação. Este facto é de fácil constatação ao ter em conta a simplicidade da representação do faseamento do processo de integração.

A introdução na arquitectura de Purdue da Arquitectura Humana e Organizacional permite que sejam introduzidos na discussão os aspectos humanos assim como a forma como eles afectam a integração.

Da mesma forma, a Arquitectura do Sistema de Fabrico permite que sejam tratados detalhes relativos à integração de fluxos de materiais, de energia, optimização do "layout" e muitos outros factores a considerar além da integração da informação.

A arquitectura de Purdue pode ser estendida de forma a cobrir todos os tipos de indústria, e tratar efectivamente todas as empresas independentemente da sua missão. Entre as três arquitecturas, esta é a que apresenta a mais ampla documentação.

A esta arquitectura falta todavia o tratamento matemático e a técnicas de modelação necessárias quando se deseja a sua transferência para um modelo computacional.

A complementaridade da PERA e da CIMOSA é evidente, fornecendo a primeira a arquitectura do projecto e a CIMOSA as ferramentas.

Restantes Arquitecturas

Comparando a ARIS com o CIMOSA tendo como referência a figura 3.10, poderá observar-se o seguinte [42]:

- Tendo em atenção os modelos (eixo vertical), os conceitos envolvidos são os mesmos embora o seu conteúdo não seja equivalente: a arquitectura ARIS estabelece a optimização dos requisitos do utilizador no modelo de definição de requisitos, deixando para a especificação do projecto os detalhes relacionados com critérios de desempenho do processamento da informação, ao contrário do CIMOSA que apenas se preocupa com a optimização dos requisitos na fase de projecto.
- Do ponto de vista dos blocos construtivos (eixo horizontal), a arquitectura ARIS não estabelece os modelos genéricos (standards) nem os modelos parciais (referentes a

um determinado tipo de indústria), sendo sempre desenvolvido um modelo de informação geral para cada modelo desenvolvido.

- Atendendo à terceira dimensão (vistas do sistema), tanto o CIMOSA como a ARIS separam o sistema de fabrico em múltiplas vistas. No CIMOSA não existe no entanto a vista de controlo cuja falta leva a que as diversas vistas não sejam mutuamente exclusivas.

A comparação entre o IEM e o CIMOSA realizada em [43] demonstra que os blocos construtivos de ambas as linguagens de modelação podem ser confrontados e traduzidos em ambos os sentidos. Constatou-se ainda que os processos de modelação apesar das semelhanças apresentavam algumas diferenças. Assim sendo, um objectivo importante será que num cenário real os modelos tenham as mesmas capacidades quer sejam gerados pelo IEM quer pelo CIMOSA. No entanto este ponto apenas poderá ser verificado através de exemplos sobre cenários comuns.

Conclusão

A "Task Force" IFIP/IFAC para as Arquitecturas para a Integração das Actividades de Fabrico fez em 1993 uma análise detalhada das três principais arquitecturas de integração empresarial que abrangiam por completo ciclo de vida de um projecto de integração: o CIMOSA, o GRAI-GIM e a PERA. Este grupo considerou então que cada uma destas arquitecturas poderia ser desenvolvida para constituir uma Arquitectura de Referência à qual seria associada a metodologia de integração apropriada, recomendando no entanto que deveria ser realizado um esforço de consolidação das melhores características de cada uma delas. Esta estratégia de consolidação deveria proporcionar a melhor metodologia para a integração de futuros sistemas.

A primeira proposta neste sentido foi realizada por Peter Bernus e Lazlo Nemes, ambos membros deste grupo de trabalho, e apresentada num encontro de trabalho em Junho de 1994 em Viena. A esta arquitectura mais completa, construída tendo em atenção as recomendações do grupo de trabalho acima referido, foi dado o nome GERAM ("Generic Enterprise Architecture and Methodology").

No contexto das arquitecturas de integração, foi também realizada uma breve introdução às arquitectura ARIS e IEM. Estas referências justificam-se pela sua ampla divulgação nos meios industriais e universitários germânicos. Relativamente à arquitectura ARIS, existe uma ferramenta CASE desenvolvida pela IDS e recentemente colocada no mercado pela SAP aparentemente com grande sucesso comercial. Embora o IEM também disponha de uma ferramenta de modelação, as razões da sua apresentação prendem-se com a sua relevância para o CEN/TC 310 WG.1 com a apresentação do documento "Comparison CIMOSA-IEM Modelling Constructs and Methodology" (draft) [43] na reunião do CEN TC310/WG1 de Fevereiro de 1995 (Copenhaga).

3.3 GERAM - Metodologia e Arquitectura Genéricas de Referência para a Integração Empresarial

3.3.1 Introdução

As recomendações do grupo de trabalho IFIP/IFAC para as Arquitecturas de Integração Empresarial deram origem à construção da Metodologia e Arquitectura Genéricas de

Referência para a Integração Empresarial, GERAM ("Generic Enterprise Reference Architecture and Methodology"), consolidando as melhores características das principais arquitecturas: CIMOSA, GRAI-GIM e PERA. A arquitectura proposta tem também por objectivo assistir os consultores no decorrer de seu trabalho de ajuda aos utilizadores. Deverá ainda ajudar à coordenação dos resultados dos trabalhos de desenvolvimento de metodologias e de ferramentas, e na identificação dos produtos adicionais mais adequados à obtenção de melhores ambientes de desenvolvimento. Este quadro deverá ainda providenciar o ambiente adequado para a caracterização das necessidades de desenvolvimento de novas teorias e conceitos na área da integração empresarial.

Tal como proposto por Bernus e Nemes, o GERAM é dividido em seis componentes principais:

- A Arquitectura Empresarial de Referência Genérica (GERA-"Generic Enterprise Reference Arquitecture"), no essencial concentrada no ciclo de vida da empresa, sendo esse ciclo de vida considerado como um processo de projecto. A arquitectura deste processo deverá identificar quais os resultados intermédios assim como os componentes do processo de projecto. Será assim considerado como o modelo do ciclo de vida do sistema.
- A Metodologia Genérica de Engenharia Empresarial (GEEM-"Generic Enterprise Engineering Methodology"), que descreve do ponto de vista genérico os processos de integração da empresa. Por outras palavras, a metodologia consiste num modelo detalhado do processo com instruções para cada passo do processo de integração.
- Ferramentas e Linguagens Genéricas de Modelação Empresarial (GEMT&L-"Generic Enterprise Modelling Tools and Languages"). A engenharia da empresa integrada é altamente sofisticada, implicando uma gestão, projecto e implementação multi-disciplinares, no decorrer das quais vão surgir várias formas de descrição e diferentes modelos da empresa a integrar. Deste facto resulta que seja necessária mais de uma linguagem de modelação.
- Modelos Empresariais Genéricos (GEM-"Generic Enterprise Models"), capturando conceitos que são comuns a todas as empresas. Assim sendo, o processo de engenharia da empresa poderá posteriormente usá-los e testá-los na construção de modelos para uma empresa específica.
- Módulos Empresariais Genéricos (GEM-"Generic Enterprise Modules"). Estes módulos são produtos considerados implementação standard de componentes que têm grande probabilidade de serem usados em actividades de integração empresarial, tanto no projecto como na própria empresa. Estes módulos genéricos podem ser configurados de forma a constituir blocos mais complexos a serem utilizados pela empresa individual.
- Teorias Empresariais Genéricas (GET-"Generic Enterprise Theories"), que descrevem os aspectos mais genéricos dos conceitos relacionados com a integração empresarial. Chamadas geralmente teorias ontológicas podem também ser considerados meta-modelos, uma vez que consideram factos e regras sobre os factos e regras dos modelos das empresas [45].

A proposta de Bernus e Nemes inclui ainda um aspecto interessante no processo de integração genérico, que resulta do facto de o modelo genérico proposto poder não só ser utilizado na descrição da empresa de fabrico propriamente dita como também de outras três "empresas" ou Entidades (figura 3.17). Pode assim descrever de forma recursiva:

- O desenvolvimento do projecto e produção do produto a ser produzido pela empresa de fabrico. O produto assim produzido é denominado *Entidade 4*.

- O ciclo de vida da empresa de fabrico propriamente dita (conceito, projecto, construção e operação). A empresa de fabrico é chamada *Entidade 3*.
- O ciclo de vida da empresa de engenharia que desenvolveu o projecto da empresa de fabrico. A empresa responsável pela engenharia é chamada *Entidade 2*.
- O processo de gestão estratégica da empresa (ciclo de vida) que originou em primeiro lugar a necessidade e o conceito para o processo de integração da empresa de fabrico, assim como da empresa de engenharia que produziu a empresa de fabrico. A empresa estratégica, é assim chamada *Entidade 1*.

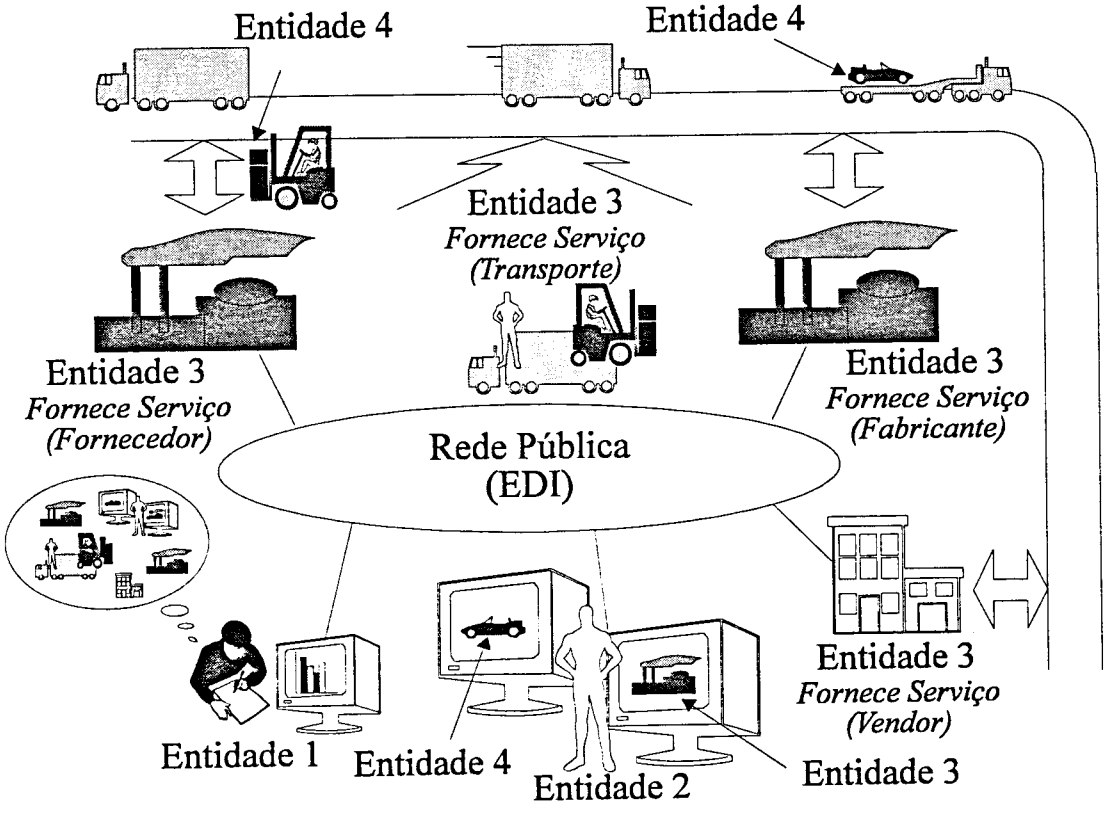


Figura 3.17 - As Entidades GERAM e respectiva interacção

Assim sendo, a Entidade 1 (a empresa estratégica) desenvolveu a Entidade 2 (empresa de engenharia), que por sua vez desenvolveu a Entidade 3 (a empresa de fabrico), que por sua vez desenvolve, produz e entrega o produto (Entidade 4) ao cliente final. Este cenário pode, com alguns ajustes, facilmente descrever qualquer conjunto de empresas bem como os seus produtos e serviços em qualquer interacção industrial ou de negócio. Bernus e Nemes propõem que a recursividade não seja estendida além destes quatro níveis, o que seria perfeitamente viável se nos reportarmos à arquitectura PERA, por exemplo onde associado ao produto (Entidade 4) poderíamos ter uma nova entidade responsável pelo projecto de engenharia do seu manual de utilização.

3.3.2 Descrição da GERAM

A GERAM baseia-se no chamado modelo matricial do ciclo de vida da empresa, desenvolvido por Bernus e Nemes para ser usado como base para a comparação e avaliação das capacidades de cada uma das arquitecturas estudadas, PERA, CIMOSA e GRAI/GIM

[45]. O quadro de análise proposto consiste numa matriz combinando cada uma das arquitecturas sobre diferentes dimensões (figura 3.18):

- três blocos verticais correspondentes ao desenvolvimento dos modelos Genéricos, Parciais e Particulares do CIMOSA;
- para cada um desses blocos verticais surge a subdivisão correspondente às vistas funcional, de informação, recursos / estrutura, decisão / organização, estabelecidas pelo GRAI-GIM;
- a separação entre os diferentes aspectos da arquitectura PERA, que se reflectem nas três arquitecturas de implementação (ASI, AEF e AH&O), estabelece a divisão horizontal de cada uma das fases do projecto;
- a fase de identificação da entidade do negócio, e as subsequentes fases conceptual, de definição de requisitos, de projecto, de implementação, de construção e de operação da arquitectura PERA, formam finalmente o ciclo de vida das três arquitecturas de implementação (ASI, AEF e AH&O).

Este modelo foi estruturado de forma a incluir a apresentação das capacidades e pontos fortes de cada uma destas arquitecturas. Uma vez que o objectivo era o desenvolvimento de uma arquitectura combinando as melhores características das três arquitecturas referidas, a matriz de avaliação torna-se ela própria na GERAM (figura 3.18). Na matriz apresentada deverá notar-se que a AEF poderá ser substituída pela arquitectura do serviço ao cliente, generalizando-se assim o âmbito da sua aplicação.

Na sequência da descrição da GERAM são caracterizados os seus principais utilizadores divididos em três grupos. Em primeiro lugar, aqueles que usam a GERAM como base para o desenvolvimento dos projectos de integração de empresas de fabrico, incluindo neste grupo os consultores de engenharia envolvidos no processo. Como segundo grupo, as entidades, indivíduos, grupos ou empresas levando a cabo desenvolvimento de novas ferramentas e metodologias de suporte à integração empresarial usadas pelo grupo anterior. Finalmente, os investigadores, académicos e outros desenvolvendo novas teorias de modelação e ontologias para providenciar o suporte teórico do trabalho realizado por ambos os grupos anteriores.

3.4 Conclusão

Neste capítulo foram apresentados e discutidos dois tipos de arquitecturas de integração de sistemas de fabrico vocacionados por um lado para resolver o problema tecnológico através da descrição da estrutura do sistema de controlo e da interligação entre funções envolvidas, e por outro suportar a definição da estratégia de "como" se faz a integração. Foi dado destaque ao trabalho realizado pelo grupo IFAC/IFIP para as Arquitecturas para a Integração das Actividades de Fabrico, o qual salientou a grande necessidade que a empresa industrial tem de que lhe forneçam um "manual de instruções" que lhe indique "como" projectar, desenvolver, implementar e utilizar a integração da empresa e do fabrico.

Em face das múltiplas arquitecturas existentes actualmente o referido grupo de trabalho do IFAC/IFIP seleccionou as arquitecturas CIMOSA, GIM e PERA como sendo as únicas que actualmente abrangem por completo ciclo de vida do projecto de integração do sistema de fabrico. No âmbito da discussão apresentada neste capítulo foram tratadas com maior detalhe as arquitecturas CIMOSA e PERA, uma vez que a documentação pública relativa à GIM é relativamente mais reduzida.

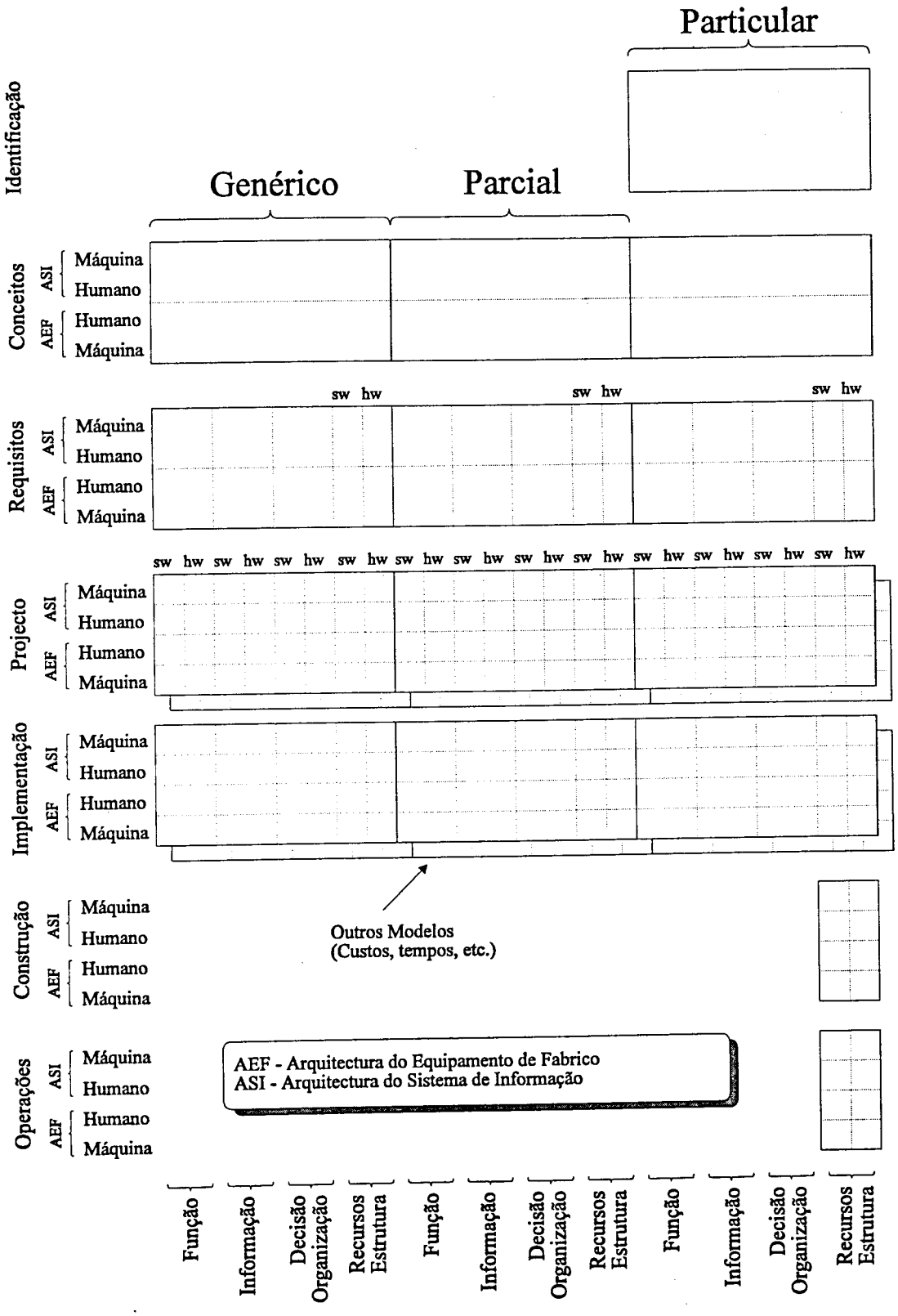


Figura 3.18 - Matriz de Avaliação GERAM

As arquitecturas CIMOSA, GIM e PERA, bem como as metodologias que lhes estão associadas, podem em qualquer dos casos ser estendidas resultando arquitecturas e metodologias completas que orientem o processo de integração empresarial. O grupo de trabalho referido sugeriu no entanto que se deveria realizar a combinação dos aspectos mais fortes das diversas arquitecturas, tirando partido das sinergias daí resultantes na construção de uma nova arquitectura. Este trabalho levou à construção de uma nova arquitectura, a "Generic Enterprise Architecture and Methodology", apresentando características particularmente interessantes para grupos de investigação que desenvolvam ferramentas para a integração de sistemas, e sendo recomendado que a arquitectura PERA deveria ser apenas utilizada na execução de projectos de integração empresarial.

Pela sua importância, a apresentação do CIMOSA será retomada nos capítulos 3 e 4, primeiro na análise da sua infra-estrutura de integração e depois no ponto de vista da utilização de modelos executáveis. Por outro lado, e no contexto da utilização de ferramentas de integração, será retomada no capítulo 4 a arquitectura de Purdue: a apresentação das tarefas e pessoas envolvidas ao longo de um projecto de integração suportará uma introdução adequada do problema da modelação do sistema de fabrico nas suas diversas vertentes.

Capítulo 4

INFRA-ESTRUTURAS DE SUPORTE A SISTEMAS DE GESTÃO DE OPERAÇÕES FABRIS

4.1 Introdução

Nos capítulos anteriores foi realizada com algum detalhe a apresentação do Sistema de Fabrico (Capítulo 2), assim como dos conceitos, metodologias e ferramentas da Integração de Sistemas de Fabrico (Capítulo 3). Na sequência dessa apresentação foram descritas as três principais arquitecturas de referência actuais de suporte à integração das actividades de fabrico, tendo sido dada particular importância à CIMOSA pela relevância do seu papel em diversos grupos de trabalho para a estandardização (Capítulo 3).

O estudo de arquitecturas de integração de sistemas de fabrico existentes demonstrou que de facto elas não fornecem ainda uma ferramenta ou conjunto de ferramentas capazes de suportar integralmente o ciclo de vida do projecto de integração do sistema de fabrico. Apesar disso a CIMOSA é, de entre as arquitecturas apresentadas, a única que propõe a existência de uma infra-estrutura de integração. A utilização prática da CIMOSA tem-se concentrado na análise para a construção de modelos empresariais, onde vem sendo aplicada com reconhecido sucesso [25]. No entanto, e apesar dos diversos projectos onde a CIMOSA esteve e está envolvida, não existe ainda um conjunto de ferramentas CASE ("Computer Assisted Software Engineering") que, de uma forma organizada, providencie um ambiente de suporte ao ciclo de vida do sistema integrado de fabrico desde a fase de definição de requisitos até à sua validação e colocação em serviço. Por outro lado, não existe também uma infra-estrutura de integração capaz de responder a todos os problemas que se colocam em projectos de integração, apesar dos trabalhos de reconhecida qualidade em curso ou já levados a cabo que poderão levar à obtenção de resultados interessantes a curto prazo.

Neste quarto capítulo serão pois apresentados os requisitos que se colocam às infra-estruturas de suporte à integração de sistemas de fabrico, sendo ainda realizada uma breve descrição de diversos sistemas / protótipos ilustrativos. Será introduzido o protótipo da infra-estrutura da CIMOSA desenvolvido no instituto WZL em Aachen na Alemanha, seguido da descrição da arquitectura proposta para o "Systems Engineering Workbench centred on CIMOSA" (SEW-OSA) pelo "Manufacturing Systems Integration (MSI) Research Institute" [46]. No SEW-OSA, pretende-se dar resposta a alguns dos problemas acima enunciados, fornecendo um ambiente integrado para a construção do modelo, que poderá ser depois executado sobre uma infra-estrutura de integração. Uma outra aproximação ao problema tem sido utilizada desde 1987 no Instituto FAW em Ulm (Alemanha), que se tem empenhado no desenvolvimento de um ambiente abrangente para a engenharia da empresa e para o desenvolvimento de aplicações [48].

Neste capítulo será ainda apresentada a infra-estrutura utilizada desenvolvida pelo projecto ESPRIT SHOP-CONTROL (EP-5478), que posteriormente suportou o produto comercial construído sobre protótipos resultantes, o PROFIT (**P**roduction **O**ptimization throught **F**lexible **I**ntegrated software **T**ools). Embora ainda em fase de discussão no interior do consórcio, serão dadas algumas perspectivas para a evolução da infra-estrutura do PROFIT no decorrer do projecto REAL-I-CIM (EP-8865). Na descrição do PROFIT será dada particular relevância à utilização da linguagem de modelação. São introduzidos descritos o ambiente e a linguagem de modelação utilizados, bem como o ciclo de vida do modelo fabril. Neste contexto, serão confrontados os conceitos fundamentais da CIMOSA com a sua aplicação prática no âmbito de instalações piloto do PROFIT. São finalmente apresentadas as aplicações de "Supervisão" (projecto SHOP-CONTROL), tal como a sua evolução no âmbito do projecto REAL-I-CIM (RIC), o "Quadro de Bordo Dinâmico", sendo ainda apresentada a aplicação de Monitorização de Condição. No âmbito do SHOP-CONTROL / PROFIT descreve-se também o "Sistema de Transferência de Mensagens", elemento fundamental da infra-estrutura, o "Gestor de Transacções Fabris" e a sua evolução para o "Módulo de Acesso Distribuído à Instalação Fabril". Das diferentes aplicações disponíveis no SHOP-CONTROL, PROFIT ou REAL-I-CIM, apenas as referidas são apresentadas pelo facto de as restantes saírem do âmbito deste trabalho.

4.2 Integração de Sistemas de Fabrico: Requisitos colocados às Infra-estruturas

4.2.1 Propriedades que devem caracterizar as Infra-estruturas de Integração

Deve-se em primeiro lugar identificar quais os atributos e qualidades que de um modo geral devem caracterizar uma infra-estrutura de integração. Estas propriedades, que devem ser consideradas sempre que se deseje definir ou projectar uma infra-estrutura para um sistema integrado de fabrico, são: a *distribuição*, a *abertura*, o suporte à *portabilidade* e à *conectividade*, a *portabilidade da infra-estrutura* de integração, a *conformidade com os standards*, a *fiabilidade*, o *desempenho* e o *suporte à migração* [23].

Pelas razões já apontadas no Capítulo 2, as aplicações informáticas, as máquinas e equipamentos de fabrico, e os recursos humanos que desempenham funções / tarefas na empresa de fabrico têm ligações a vários computadores, estando estes geralmente interligados através de uma rede de comunicações. Assim, num processo de integração global, o sistema de fabrico, necessariamente distribuído, apoia-se ou recorre a diferentes

computadores, devendo a infra-estrutura de integração de suporte a tal sistema distribuído apresentar ela própria uma estrutura distribuída.

Na selecção dos múltiplos módulos de software necessários para a construção de um sistema integrado, incluindo os módulos da própria infra-estrutura de integração, é do interesse tanto da empresa industrial como dos analistas de sistemas e dos responsáveis pelo desenvolvimento / integração, que exista uma oferta diversificada de componentes susceptíveis de serem integrados de diversas formas para construir soluções adaptadas. Os sistemas abertos são pois caracterizados pela facilidade com que podem ser constituídos pela interligação de componentes individuais, e alterados e expandidos através de alterações de parte ou partes dos sistemas iniciais. Assim sendo, a abertura baseada em standards "de facto" é uma característica desejável para a infra-estrutura de integração.

Por outro lado, é da maior importância para as entidades envolvidas que sejam portáveis (software) e de simples conectividade (hardware) em relação à infra-estrutura de integração. Esta infra-estrutura deve deste modo ser projectada favorecendo as capacidades de portabilidade e de conexão. Além da infra-estrutura de integração suportar a portabilidade dos componentes com os quais faz interface, deve ela própria ser transportável em vários aspectos (sistema operativo, linguagem, ...) em relação à plataforma onde é executada. Em qualquer dos casos anteriores, um requisito importante para a portabilidade é, sem dúvida, a conformidade com os standards genéricos ISO para a portabilidade, tal como por exemplo a conformidade com o POSIX [49]. É pois da maior relevância que uma nova infra-estrutura seja definida em conformidade com os standards existentes, requisito imprescindível para que se consiga a portabilidade e a conectividade desejadas, assim como a distribuição e a abertura do sistema.

A fiabilidade de um sistema é definida como a sua capacidade de desempenhar continuamente um determinado serviço. Além da necessidade evidente da utilização de componentes individualmente fiáveis, na fiabilidade podem ainda ser englobadas técnicas de tolerância a falhas usadas para tratar o problema da possível saída de serviço de um componente crítico. Algumas das técnicas possíveis são a replicação (de serviços e de dados) e, quando possível, a organização das interacções sob a forma de transacções.

A infra-estrutura de integração deverá naturalmente apresentar o melhor desempenho possível tendo como referência o estado-da-arte da tecnologia. Esta propriedade é determinada pela implementação específica da infra-estrutura de integração.

Por último, vem a capacidade de suporte à migração, que neste contexto significa migração no tempo, i.e. a passagem de um determinado sistema para um outro através de alterações sucessivas ao longo do tempo. Este é um requisito absoluto para os sistemas actuais onde o mote é a contínua evolução.

4.2.2 A Interface com o Utilizador

A importância de que se reveste em ambiente industrial a interface com o utilizador justifica reflexão cuidada, uma vez que ela constitui de facto a superfície de integração do ponto de vista do utilizador, e é determinante no sucesso do tipo e modo de interacção escolhidos. Embora nos primórdios da informática a importância da interface com o utilizador tenha sido frequentemente ignorada, tornou-se evidente que o sucesso das aplicações desenvolvidas se deveu de facto à enorme capacidade de adaptação do Homem. O utilizador adapta-se a qualquer interface por mais complicada e mal concebida que ela seja. Esta capacidade de adaptação inerente à pessoa humana pode eventualmente ser "ajudada" pela curiosidade natural, ou ainda por pressões externas para a utilização das

novas tecnologias, tais como ameaça de desemprego ou despromoção, caso o utilizador não mantiver a sua competitividade [50].

Na última década, o computador tornou-se entretanto num auxiliar potente e barato nas múltiplas actividades do Homem. Utilizado por um número crescente de pessoas, houve necessidade de criar uma nova definição do "bom sistema de computação", em particular no que se refere à eficiência da utilização do computador pelo seu utilizador, que é considerada o ponto chave na avaliação desse novo sistema. É óbvio que a produtividade, a satisfação pessoal, o interesse, bem como o prazer na utilização, dependem directamente do modo como cada pessoa interactua com o computador. Assim sendo, o principal objectivo passou a ser a adaptação do computador ao utilizador, evitando-se tanto quanto possível a situação inversa antes verificada.

No ambiente industrial, a interface com o utilizador reveste-se hoje em dia de particular importância, justificando o facto de nos últimos anos ter sido objecto de especial atenção. Existe assim, por exemplo, uma grande variedade de software para facilitar a configuração de uma célula de fabrico tendo vindo a desaparecer a necessidade de escrever software complexo de baixo nível para o seu controlo [51].

A grande variedade de aplicações desenvolvidas para o ambiente industrial, faz com que seja comum observar situações onde a aparente sofisticação do software esconde a simplicidade do processo, ou não salvaguarda o utilizador da sua complexidade. Nestes casos, o operador é inundado de informação ("information overload"), sendo obrigado a um grande esforço de abstracção para extrair a informação que de facto necessita para a tomada de decisão. Estes problemas ocorrem com alguma frequência quando os tecnólogos esquecem que na instalação fabril, além de típico utilizador do sistema informático, o operador está envolvido por um ambiente com solicitações múltiplas que constantemente exigem a sua atenção, e esperam a sua reacção em "tempo real".

4.3 A Infra-estrutura de Integração da CIMOSA

O principal objectivo da CIMOSA é o permitir que a empresa trate de uma forma adequada os problemas relacionados com a introdução de processos de gestão da mudança, com o aumento de flexibilidade e com a integração. Para atingir este objectivo, a CIMOSA define os três seguintes conceitos: um Quadro de Modelação (Cubo), os Ambientes da CIMOSA (de operação e de engenharia), e a Infra-estrutura de Integração. Os dois primeiros conceitos foram já introduzidos no Capítulo anterior.

A Infra-estrutura de Integração da CIMOSA: fornece interfaces standard aos componentes activos da empresa sob a forma de serviços; permite a integração das aplicações e processos componentes da empresa através do suporte operacional ao acesso à informação; permite a execução do modelo da empresa (Modelo Particular de Descrição da Implementação) para controlar, ou pelo menos monitorizar, as operações quotidianas [23].

Na CIMOSA, a infra-estrutura de integração deve basear-se em tecnologias da informação que suportem o ambiente de fabrico e permitam a execução de um modelo conforme com a CIMOSA. Temos neste contexto, um conjunto de requisitos para o primeiro nível da decomposição da infra-estrutura.

4.3.1 Requisitos Funcionais Genéricos

A infra-estrutura deve realizar a denominada *Gestão de Eventos e do Comportamento,* sendo capaz de receber e aceitar eventos da instalação fabril e de os relacionar com os domínios relevantes no modelo. Deve ainda interpretar o comportamento descrito em cada domínio do modelo, por forma a poder disparar a execução e a monitorização dos seus processos no âmbito de um determinado conjunto de recursos. A infra-estrutura de integração utiliza deste modo o modelo para a orquestração dos processos de fabrico da empresa.

A infra-estrutura de integração deve por outro lado ser capaz de suportar todas as aplicações computorizadas usadas nos processos de fabrico realizando a *Gestão dos Recursos de Fabrico*. Deve assim poder interagir com as máquinas e dispositivos de fabrico ao nível da instalação fabril de forma a instruí-los sobre as operações a serem executadas, a receber respostas (tais como pedidos de carga de ferramenta, transferência de programas de comando numérico, etc.) e a obter informação acerca da conclusão de uma operação ou da ocorrência de algum problema.

Facilidades de *Gestão da Informação* são uma necessidade evidente da empresa industrial, que trata uma gama muito diversificada de informação (ordens de fabrico, programas de controlo numérico, características técnicas dos produtos, etc.). A infra-estrutura de integração deve fornecer informação de uma forma transparente aos diversos módulos que a solicitem, escondendo a localização dos dados (locais ou remotos), o método de arquivo (sistema de gestão de base de dados ou de ficheiros), e a forma como a informação é transmitida. Estes requisitos básicos são providenciados por um conjunto de componentes da infra-estrutura chamados "Serviços de Informação", que utilizam essencialmente a informação contida na Vista de Informação do modelo CIMOSA.

A empresas dispõem ainda de redes para a comunicação entre os sistemas informáticos existentes, sendo evidente a necessidade de facilidades de *Gestão das Comunicações*. Estas redes deverão suportar a transferência de mensagens e de informação necessárias ao funcionamento activo da infra-estrutura integradora da CIMOSA. A interligação de todas estas infra-estruturas é realizada através dos denominados "Serviços Comuns".

Em resumo, a infra-estrutura integradora reage a eventos da empresa através da execução de um modelo conforme com a CIMOSA. A infra-estrutura encapsula / esconde a localização física da informação e dos recursos do cliente que requer a informação, além de fazer a gestão da ligação à infra-estrutura de comunicações. A infra-estrutura de integração da CIMOSA pode ser considerada por si só como uma entidade. Deve deste modo conter um conjunto de serviços específicos que permitam a configuração e a gestão da utilização dos seus serviços, os quais são denominados "Serviços de Gestão do Sistema".

4.3.2 Primeiro nível de Decomposição da Infra-estrutura

Nos parágrafos anteriores vimos que a Infra-estrutura de Integração (II) da CIMOSA deve satisfazer requisitos funcionais distintos. Para tal, a II é constituída por cinco conjuntos de serviços, conjuntos esses denominados entidades. A figura 4.1 apresenta estas entidades, sendo uma entidade definida em termos do conjunto de funcionalidades providenciando o contexto onde um conjunto de serviços individuais pode ser especificado. Os serviços serão caracterizados no segundo nível de decomposição da infra-estrutura de integração.

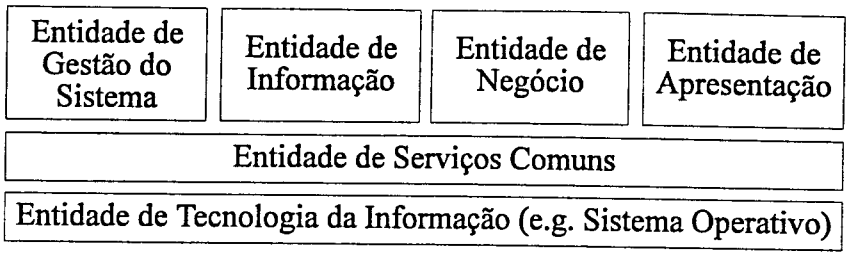


Figura 4.1 - Implementação da Infra-estrutura de Integração da CIMOSA

As entidades referidas anteriormente representam assim as funcionalidades da infra-estrutura:

- **A Entidade de Apresentação** fornece os meios para a gestão dos recursos, e tem por objectivo providenciar um conjunto de funções genéricas que suportem a integração de componentes da empresa, disponibilizando funcionalidades tais como: o suporte da interacção entre a infra-estrutura de integração e os recursos da empresa (pessoas e máquinas); a apresentação de recursos heterogéneos de uma forma homogénea aos restantes serviços da infra-estrutura; o providenciar do acesso a componentes externos; a integração entre as aplicações existentes.
- **A Entidade de Negócio** fornece a gestão de eventos e do comportamento, e tem por objectivo providenciar as funções para iniciar, monitorizar e controlar as operações da empresa, de acordo com um determinado Modelo da mesma, através do processamento do correspondente modelo executável. Assim sendo, esta entidade apresenta funcionalidades suportando: a gestão reactiva das operações da empresa (disparadas por eventos); a coordenação, sequenciamento e sincronização das operações; a gestão dos recursos da empresa; as mudanças das operações de uma forma flexível; os mecanismos de suporte à intervenção humana no tratamento de eventos de excepção.
- **A Entidade de Informação** fornece os meios para a gestão da informação e as funções genéricas necessárias ao acesso, integração e manipulação da informação. Com este objectivo, apresenta funcionalidades permitindo: o acesso generalizado à informação da empresa; o acesso transparente a meios de armazenamento heterogéneos; a manutenção da consistência e integridade da informação; a alteração de forma flexível da informação; o suporte à definição e manipulação da informação relevante; o suporte da informação do sistema e da informação do utilizador; o suporte de diferentes tipos de dados (alfanumérico e ficheiro); a definição e a garantia dos privilégios de acesso à informação.
- **A Entidade de Serviços Comuns** providencia a gestão das comunicações e fornece os serviços comuns a todas as outras entidades. Apresenta funcionalidades tais como: a troca de mensagens entre os serviços da infra-estrutura; a gestão das mensagens (atendendo, por exemplo, a prioridades); a garantia da transparência exigida pelas outras entidades; o fornecimento dos serviços comuns (tais como os serviços de nomes, de tempos e de autenticação).
- **A Entidade de Gestão do Sistema** fornece os meios para a gestão da infra-estrutura, tais como as facilidades genéricas para configurar, manter e monitorizar os componentes da infra-estrutura de integração, permitindo: distribuir e instalar as novas versões dos componentes da infra-estrutura de integração; iniciar e configurar estes componentes para a sua operação; iniciar e gerir a operação dos mesmos componentes; reconfigurar e relatar informação de segurança; registar erros e medidas de desempenho dos componentes.

4.3.3 Segundo nível de Decomposição da Infra-estrutura

O segundo nível da decomposição segue-se ao nível das Entidades e corresponde necessariamente ao nível dos Serviços. Do ponto de vista abstracto, um Serviço é composto por um conjunto de funções que descrevem qual o serviço fornecido, não sendo relevante quem o invoca ou a forma como ele é executado.

São quatro os serviços básicos fornecidos pela infra-estrutura de integração da CIMOSA, os Serviços do Negócio, os Serviços de Informação, os Serviços de Apresentação e os Serviços Comuns, sucintamente descritos nos parágrafos que se seguem [23].

Serviços do Negócio

Os denominados *Serviços do Negócio* fornecem as funções necessárias à execução do modelo da empresa, que contém tanto a vista de informação (função e comportamento) como a vista de recursos. Estes serviços são responsáveis pela gestão da execução dos processos do negócio (PN) e das actividades da empresa (AE) interpretando os conjuntos de regras de procedimento. Estes serviços realizam ainda a gestão de recursos, fazendo a reserva e o escalonamento dinâmico dos recursos utilizados pelas AE na execução das suas "Operações Funcionais".

Serviços de Informação

Os *Serviços de Informação* providenciam as funções genéricas para o acesso, integração e manipulação da informação. Existem dois serviços cooperantes, o Serviço de Informação Global ("System-wide-data service") e o Serviço de Gestão de Informação ("Data Management service"), que tratam de uma forma transparente os pedidos. O Serviço de Informação Global (SIG) providencia o acesso unificado a dados aos seus clientes, existam aqueles sobre a forma de um modelo relacional ou ficheiro, através da definição e uso de vistas SQL ou de uma linguagem de alto nível de acesso a ficheiros. Os pedidos de serviço são assim convertidos em múltiplos acessos a servidores das bases de dados distribuídas no sistema CIMOSA. Estes pedidos são expressos em SQL ou numa linguagem de gestão de ficheiros e passados ao Serviço de Gestão de Informação (SGI) que realiza a interface com o sistema de armazenamento local. Por seu lado, o SGI é responsável pela apresentação de um sistema de armazenamento virtual único ao SIG. Se necessário, a este serviço poderá ser exigido que traduza um pedido recebido em SQL na linguagem proprietária da base de dados.

Serviços de Apresentação

Os *Serviços de Apresentação* disponibilizam um conjunto de funções necessárias ao controlo da execução das "Operações Funcionais" realizadas pelos recursos heterogéneos da empresa, tais como as: Entidades Funcionais das Máquinas ("Machine Functional Entities"), Entidades Funcionais das Aplicações ("Application Functional Entities") e Entidades Funcionais Humanas ("Human Functional Entities") descritas na vista de recursos da CIMOSA. Com o objectivo de tratar os diversos tipos de serviços de apresentação, a Infra-estrutura de integração da CIMOSA providencia os "Machine Dialogue Service", "Application Dialogue Service" e "Human Dialogue Service", que tratam respectivamente do controlo da execução das operações funcionais das máquinas, das aplicações e das operações funcionais humanas.

Serviços Comuns

Os *Serviços Comuns* são introduzidos de forma fornecer as mesmas soluções para problemas comuns a todos os serviços anteriores (ver figura 4.1). A maior parte dos problemas são levantados pelas necessidades de distribuição e comunicação, e pela exigência associada de transparência. As funcionalidades identificadas para os serviços comuns cobrem os serviços de comunicação, tais como a troca de mensagens entre os componentes da CIMOSA, assim como os serviços de gestão de nomes, registo de componentes no sistema de comunicações, segurança e tempo.

Os Serviços Comunicação (SC) gerem a troca de mensagens entre os elementos constituintes da Infra-estrutura de Integração da CIMOSA. Os SC evitam desta forma que os clientes e servidores na Infra-estrutura de Integração da CIMOSA se preocupem com interfaces específicas da infra-estrutura de comunicações. Estes serviços asseguram uma transferência de mensagens segura, contemplando ainda modos de transferência síncrona e assíncrona bem como o tratamento das filas de mensagens. Garantem ainda a transparência da comunicação e do acesso a serviços, garantindo a independência do local onde os processos se encontram, sejam eles locais ou remotos. A transparência na migração das aplicações, assim como na replicação de servidores, é também assegurada; no primeiro caso as aplicações são livres de migrar de um ponto para outro da rede sem que o sistema seja reconfigurado, enquanto que no segundo caso o cliente não tem necessidade de saber qual o servidor que está a atender o seu pedido. Os SC garantem finalmente a transparência relativa à existência de processos clientes / servidores concorrentes.

Os Serviços Comuns são ainda responsáveis por evitar que, acidental ou intencionalmente, seja retirada informação do sistema de forma não autorizada, devendo ainda evitar o acesso a intrusos. A não ambiguidade do nome e endereço dos clientes e servidores na Infra-estrutura de Integração deve ser garantida em toda a extensão do sistema, sendo o registo de parceiros envolvidos na comunicação uma pré-condição para a comunicação e para a reconfiguração dinâmica do sistema distribuído.

4.3.4 Protótipo da Infra-estrutura de Integração da Arquitectura CIMOSA

A figura 4.2 ilustra o protótipo da infra-estrutura da CIMOSA desenvolvido no instituto WZL em Aachen. Trata-se de uma instalação pré-piloto usando as facilidades do OSF/DCE ("Distributed Computing Environment") [52]. Na figura é apresentada a forma como os modelos são de facto executados: um evento (EV) chega do mundo exterior (por exemplo uma solicitação à empresa) sendo introduzido no sistema CIMOSA através de um Serviço de Apresentação (SA). Este evento é então colocado pelo gestor de eventos numa fila de espera associada à classe de eventos que lhe diz respeito, sendo usado para disparar a ocorrência / execução dos Processos do Domínio relevantes (PD), disparo esse que pode eventualmente ser apenas realizado numa determinada data (eventos "atrasados"). A execução de cada PD é realizada através da execução do código que lhe está associado. Este código determina o comportamento do processo (conjunto de regras de procedimento) assim como as actividades relevantes através da invocação de operações funcionais sobre determinadas entidades funcionais. O Gestor de Recursos dos Serviços do Negócio (SN) é responsável pela libertação e atribuição de recursos. As entidades funcionais estão ligadas ao sistema através de portos, sendo a interface com o sistema realizada através dos SA. Se forem necessárias chamadas a entidades funcionais remotas, são usados os Serviços Comuns (SC) para a execução das operações através da rede de comunicações. A acesso à

informação (leitura, escrita, e actualização de vistas de objectos de informação) é realizado através do Serviço de Informação (SI).

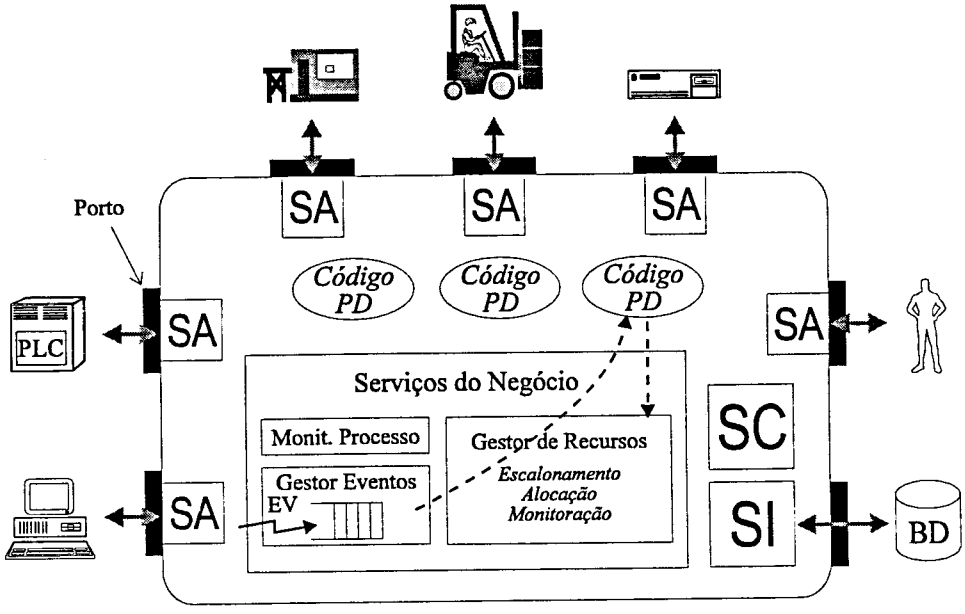


Figura 4.2 - Protótipo da Infra-estrutura de Integração da CIMOSA

4.3.5 SEW-OSA: "System Engineering Workbench for CIMOSA"

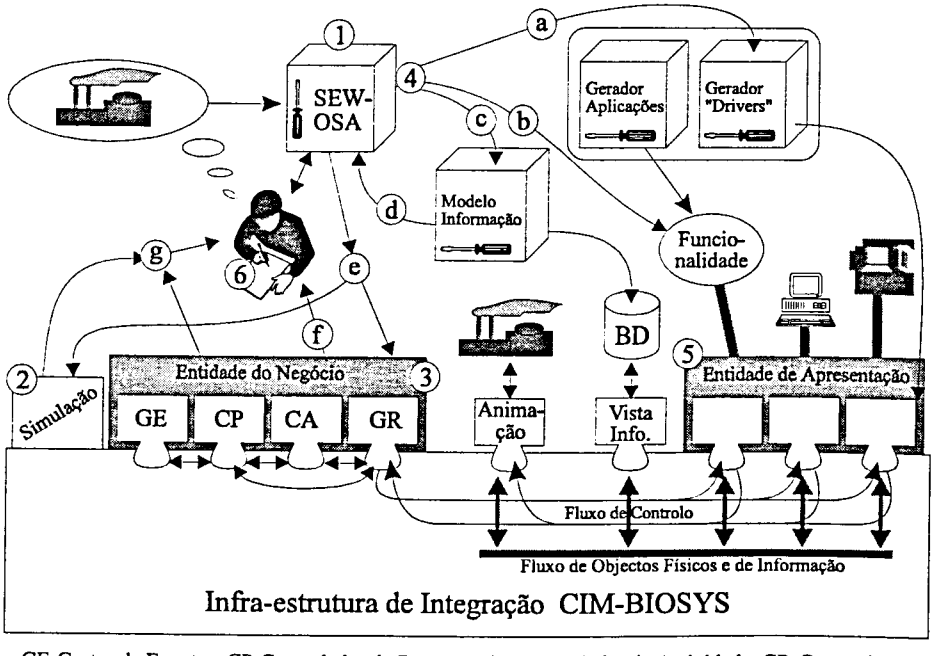
Apesar dos diversos projectos onde a arquitectura CIMOSA já está envolvida, não dispõe ainda de um conjunto de ferramentas CASE que de uma forma organizada providenciem um ambiente de suporte à migração ao longo de seu ciclo de vida, i.e. desde a fase de definição de requisitos até à colocação em serviço do sistema. No "Manufacturing Systems Integration (MSI) Research Institute" da Universidade Tecnológica de Loughborough tem vindo a ser desenvolvido o SEW-OSA ("Systems Engineering Workbench centred on CIMOSA") [46,47]. O SEW-OSA, cuja estrutura se ilustra na figura 4.3, pretende dar resposta a estes problemas fornecendo um ambiente integrado para a construção do modelo, que poderá depois ser executado sobre uma infra-estrutura de integração.

No SEW-OSA, a execução do sistema físico é conseguida pela substituição gradual dos seus componentes *emulados* no modelo pelos respectivos componentes físicos, sejam eles máquinas ou equipamentos de fabrico, programas de aplicação ou seres humanos. Como se ilustra nas figuras 4.2 e 4.3, os componentes físicos têm acesso à infra-estrutura integradora através da Entidade de Apresentação. Antes da aceitação final do sistema deve no entanto ocorrer um conjunto de iterações relativas à construção do modelo da empresa. A importante contribuição do SEW-OSA consiste em permitir a iteração através das diversas fases de construção do modelo de uma forma consistente, favorecendo a obtenção de um projecto melhorado [46]. A execução do modelo construído é conseguida através da execução de um conjunto de processos sobre uma infra-estrutura de integração industrial desenvolvida no MSI, o CIM-BIOSYS.

O CIM-BIOSYS [53] ("CIM Building Integrated Open SYStems") é uma infra-estrutura integradora que providencia os mecanismos, os métodos e as ferramentas necessárias à construção de sistemas de fabrico integrados. A estrutura do CIM-BIOSYS

permite-lhe actuar como "federador" dos standards internacionais emergentes, apresentando uma abordagem estruturada para a resolução dos problemas da fragmentação da informação, da comunicação entre processos e da interacção entre ambientes de fabrico heterogéneos.

A figura 4.3 ilustra ainda os diversos mecanismos disponibilizados pelo SEW-OSA para a construção e execução sobre uma infra-estrutura de integração de modelos conformes com a CIMOSA. Assim sendo, para uma dada empresa industrial é usado o conjunto de ferramentas do SEW-OSA em primeiro lugar (1) na construção de um modelo da empresa conforme com a CIMOSA; é então utilizado um simulador (2) que permite realizar estudos de simulação tendo como base o modelo do negócio construído. A criação da Entidade do Negócio (3) permite no âmbito do CIM-BIOSYS a construção rápida de protótipos de sistemas integrados de fabrico (e), a serem executados. O estabelecimento de ligações com outras ferramentas e serviços exteriores (4) permite tratar aspectos do projecto do sistema não contemplados no SEW-OSA, tais como a modelação detalhada dos recursos e da informação. O CIM-BIOSYS foi dotado de uma Entidade de Apresentação (5) permitindo a sua integração com os recursos da empresa. A análise dos resultados da execução do modelo fabril e a sua comparação com os resultados da simulação permite a validação do referido modelo fabril (g, 6).



Legenda: GE-Gestor de Eventos, CP-Controlador do Processo, CA-Controlador da Actividade, GR-Gestor de Recursos, a-Script de Configuração Tecnológica ("Tecnology Dependent Script"), b-Estrutura da Funcionalidade, c-Lista de Objectos de Informação, d-Lista de Atributos de Informação, e-Modelo do Negócio, f- Registo de Históricos ("logs"), g-Realimentação da execução do Modelo

Figura 4.3 - SEW-OSA: "System Engineering Workbench for CIMOSA"

A bancada de trabalho desenvolvida permite [46]: (i) a criação de modelos que capturam a informação relevante para o processo de integração do sistema de fabrico; (ii) a análise, simulação e avaliação do desempenho do modelo; (iii) a geração do código associado ao modelo que reflecte a estrutura da empresa destino, assim como a funcionalidade básica dos seus componentes; (iv) a execução automática do código gerado pela ferramenta CASE (realização rápida de protótipos) nos diversos níveis de abstracção

do processo de projecto. O SEW-OSA está actualmente a ser validado através de uma aplicação piloto numa empresa da indústria electrónica.

O SEW-OSA, em conjunto com a infra-estrutura de integração desenvolvida, o CIM-BIOSYS, constitui-se em um dos primeiros protótipos capazes de incorporar os requisitos da CIMOSA [49].

4.3.6 Ωmega: "Operational Modelling Environment and Generator for AMBAS"

Desde 1987 que o FAW, Instituto de Investigação para o Processamento do Conhecimento Aplicado, em Ulm na Alemanha, tem mantido um esforço continuado no desenvolvimento de um ambiente coerente de suporte à engenharia da empresa e ao desenvolvimento de aplicações de software para ambientes empresariais. O ambiente proposto pelo FAW estabelece a arquitectura global do sistema assim como o modelo de suporte à engenharia do software. O aspecto mais importante que levou ao desenvolvimento desta estratégia global foi o reconhecer que os múltiplos esforços de normalização não levaram à criação de um conjunto alargado de standards ao nível das infra-estruturas [54]. Na realidade, a heterogeneidade vai sem qualquer dúvida prevalecer não só em relação aos componentes da infra-estrutura mas também em relação às metodologias e a outros aspectos das tecnologias da informação [48]. Com este cenário como pano de fundo, o paradigma de software proposto pelo FAW pretende definir um quadro que suporte, de uma forma activa, as empresas na construção de um ambiente de engenharia integrado e cooperativo, assim como um ambiente de execução distribuído de informação, serviços e processos. Neste contexto não devem ser esquecidas as necessidades de migração tanto dos componentes infraestruturais como das metodologias. Chegou-se assim a uma aproximação *baseada em conhecimento* que tenta resolver o problema da heterogeneidade através da neutralização e não através da estandardização. A neutralização é assim realizada através da utilização de modelos dos diferentes componentes e na sua tradução através de mecanismos *baseados em conhecimento* [55].

O primeiro protótipo operacional do sistema de desenvolvimento completo está neste momento disponível, o Ωmega ("Operational Modelling Environment and Generator for AMBAS applications"), contendo um conjunto de componentes usados no desenvolvimento de modelos e na geração de código executável [48].

O ambiente de execução, denominado AMBAS ("**A**daptative **M**ethod **B**ase **S**hell") é compatível com a especificação da interface CORBA ("Common Object Request Broker Architecture" [56]) e vai ao encontro das necessidades da infra-estrutura de integração do CIMOSA. Fornece ainda funcionalidades adicionais, tais como o tratamento inteligente dos pedidos ao ORB ("Object Request Broker") e facilidades de integração com outros sistemas não compatíveis com o CORBA, tais como DCE, TCP/IP [52,57].

Com base no conhecimento armazenado numa base de dados orientada ao objecto denominada ObjectStore, o AMBAS suporta o desenvolvimento de novos tipos de sistemas inteligentes de apoio à decisão num ambiente distribuído através da utilização da base de métodos inteligentes do AMBAS. O AMBAS fornece às aplicações cliente uma API orientada ao problema, estas aplicações colocam os seus problemas ao sistema e o ORB inteligente procura no sistema os métodos capazes de o resolver. O AMBAS permite a distribuição de dados e de funções, e facilita a introdução de novos conjuntos de métodos no sistema. Constitui deste modo a base fundamental para o desenvolvimento cooperativo de aplicações e de processos, incluindo sistemas de apoio à decisão, com a possibilidade de transferir directamente os novos algoritmos para a utilização industrial.

A conformidade do modelo de engenharia de software estabelecido no Ωmega com a especificação da CIMOSA, bem como a incorporação dos requisitos da arquitectura do CORBA, transforma o Ωmega num segundo caso de sucesso na execução de modelos conformes com a CIMOSA sobre uma infra-estrutura de integração [55].

4.4 PROFIT, "PRoduction Optimization through Flexible Integrated software Tools"

Os desenvolvimentos que são de seguida apresentados, ao contrário dos ambientes anteriormente referidos no âmbito de objectivos mais genéricos estão, como veremos, mais vocacionados para as Pequenas e Médias Empresas em sectores discretos de fabrico e no nível de execução das operações fabris.

4.4.1 Introdução

Nesta secção, será apresentada a infra-estrutura de integração desenvolvida e utilizada no projecto ESPRIT SHOP-CONTROL (EP-5478) e posteriormente no produto comercial sobre ele construído (o PROFIT), bem como a sua evolução no projecto ESPRIT REAL-I-CIM (EP-8865). São descritos o ambiente e a linguagem de modelação utilizados, assim como ciclo de vida do sistema, com especial relevância para a aplicação dos conceitos fundamentais da CIMOSA no PROFIT.

O objectivo do projecto SHOP-CONTROL foi o de fornecer às PMEs um conjunto integrado de aplicações, no nível da execução das operações de fabrico interactuando com a instalação fabril através de equipamento de recolha de dados (terminais fabris) e redes industriais de comunicações. O conjunto de aplicações desenvolvidas oferece funcionalidades de escalonamento, despacho e controlo da produção, monitorização e supervisão da instalação fabril, gestão de alarmes, gestão da qualidade e gestão da manutenção. Ao utilizador final, seja ao nível da instalação fabril ou ao nível da gestão, é fornecida uma interface comum a todas as aplicações, que permite ainda o acesso à base de dados fabril. O projecto decorreu entre Janeiro de 1991 e Abril de 1993 com implementação de uma instalação piloto em Portugal na Cifial (fabricante de ferragens, torneiras e outros produtos metálicos).

Depois de mais de um ano de trabalho sobre resultados do projecto, um consórcio luso-alemão lançou um novo produto para a gestão integrada da instalação fabril, o PROFIT ("Production Optimization Through Flexible Integrated Software Tools"), que foi simultaneamente introduzido no mercado Português e Alemão (Fevereiro de 1994). Este produto, situa-se na área dos sistemas de execução fabril ("Shop Floor Control Systems", SFCS, ou "Manufacturing Execution Systems", MES [58]). Durante as fases de especificação, desenvolvimento e integração, foi adquirida pelo consórcio experiência considerável na especificação de aplicações fabris, na modelação da instalação fabril e no desenvolvimento de uma infra-estrutura suportando não só a integração de aplicações como um crescimento modular e aberto do sistema integrado de gestão fabril.

O PROFIT está neste momento a ser instalado em Portugal, em dois fornecedores da indústria automóvel e no âmbito da acção especial AICIME ("Automotive Industry CIME"), [59], na Alemanha e na Inglaterra. No projecto REAL-I-CIM (RIC), iniciado em Março de 1994, o PROFIT está ser usado como ferramenta de realização de protótipos em duas instalações piloto, respectivamente na indústria do calçado (Growela, Maia - Portugal) e na indústria automóvel (BMW-Lohof, Munique - Alemanha). Como ilustra a figura 4.4, o

projecto RIC beneficiou na fase de análise de requisitos e de especificação de sistema das experiências e da validação dos utilizadores do PROFIT, assim como do envolvimento de dois grupos de utilizadores em Portugal e na Alemanha. Esta informação está a ser utilizada na realização de aplicações avançadas no âmbito do RIC que poderão eventualmente ser introduzidas no produto comercial PROFIT.

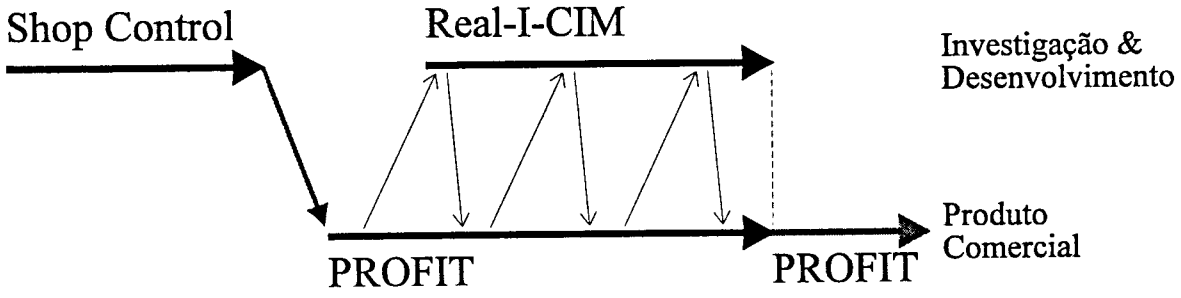


Figura 4.4 - Evolução Temporal do PROFIT em relação aos projectos ESPRIT SHOP-CONTROL e REAL-I-CIM

Nos pontos seguintes, será apresentada a infra-estrutura de integração do PROFIT. Na sequência desta apresentação, será ilustrada a evolução sofrida pela infra-estrutura durante a construção do PROFIT tendo como base o SHOP-CONTROL, assim como a posterior evolução no âmbito do REAL-I-CIM. São então apresentadas algumas das aplicações desenvolvidas no âmbito do SHOP-CONTROL nas quais o autor desta dissertação teve intervenção directa, quer na concepção e especificação quer no desenvolvimento e instalação.

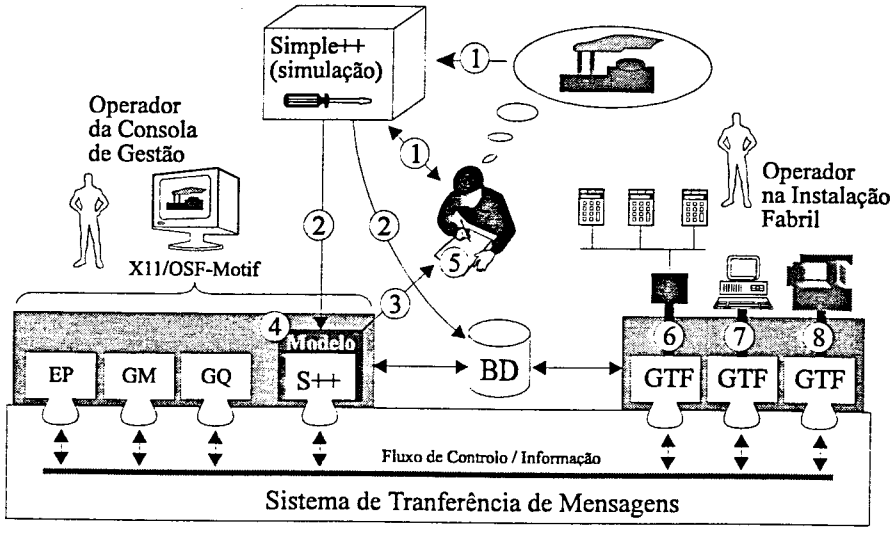
4.4.2 Plataforma Integrada de Suporte a Aplicações de Gestão Fabril

O ambiente de modelação da arquitectura CIMOSA, bem como a sua infra-estrutura de integração, constituíram alguma orientação nos desenvolvimentos realizados no âmbito do projecto SHOP-CONTROL / PROFIT. A figura 4.5 ilustra o ambiente integrado fornecido pela infra-estrutura do PROFIT, onde estão disponíveis dois ambientes, o de desenvolvimento / engenharia do modelo fabril e o de execução do mesmo modelo para controlo das operações.

O desenvolvimento do modelo fabril no Simple++ [60], uma ferramenta estado-da-arte para a modelação e simulação de sistemas discretos de fabrico, é realizado na sequência de uma análise dos processos de fabrico envolvidos na instalação fabril (1) e da produção de sucessivos modelos de monitorização / controlo e simulação para estudo e validação dos mesmos processos.

Um vez construído e validado o modelo de simulação, e dada a semelhança entre o modelo de controlo da simulação e o modelo do controlo do sistema real, é criada uma versão daquele da qual se deriva o modelo de execução do sistema de fabrico (2). Em simultâneo, é preenchida a estrutura da base de dados fabril que contém a informação actualizada sobre o modelo fabril (2). O modelo fabril é assim executado sobre uma outra instância do Simple++ (S++, na figura 4.5) que, além da ligação à Base de Dados contém uma ligação a um Sistema de Transferência de Mensagens (4), um dos principais componentes da infra-estrutura de integração do PROFIT. O modelo em execução é deste modo capaz de interagir com as restantes aplicações do PROFIT, nomeadamente as aplicações de Gestão da Qualidade e da Manutenção, de Escalonamento da Produção, ou de Gestão de Alarmes. Além disso, ao modelo fabril em execução é permitida a interacção

com a própria instalação fabril através de várias instâncias de um "Gestor de Transacções Fabril", que fornece o acesso a redes industriais de campo e terminais fabris proprietários (6), a PCs industriais (MS-DOS ou MS-Windows) (7), e a controladores de máquinas de comando numérico, PLCs, etc. (8). A realimentação resultante da execução do modelo fabril é usada tanto pela equipa de engenharia (3) para a validação, manutenção e afinação do modelo (5 e 1) como pelo operador responsável pela gestão das operações na própria fábrica.



Legenda: EP - Escalonamento da Produção, GM - Gestão da Manutenção, GQ - Gestão da Qualidade, S++ - Simple++, GTF - Gestor de Transacções Fabris

Figura 4.5 - Plataforma Integrada de Suporte a Aplicações de Gestão Fabril

A Linguagem de Modelação

A linguagem de modelação utilizada é a linguagem suportada pela ferramenta de modelação e simulação, o Simple++ [60]. Esta é uma ferramenta comercial orientada à simulação do fluxo de materiais em estudos de optimização e controlo de processos complexos de logística fabril. Trata-se de uma linguagem orientada ao objecto, suportando herança e derivação de classes partindo de outras pré-definidas. Dispondo de um conjunto básico de primitivas, o utilizador pode ir construindo novos blocos que poderão ser instanciados na construção de novos modelos tantas vezes quantas as necessárias.

As primitivas de modelação do Simple++ poderão ser divididas em dois conjuntos básicos, as *Primitivas Físicas* e as *Primitivas Funcionais*. As *Primitivas Físicas* são utilizadas para a modelação de elementos físicos, nomeadamente recursos de fabrico, transporte e armazenamento, como por exemplo: tornos de controlo numérico, passadeiras rolantes, operadores, máquinas de processamento série, máquinas de processamento paralelo, armazéns, veículos filo-guiados. As primitivas físicas permitem a caracterização dos objectos físicos, indicando por exemplo, qual a capacidade do armazém, qual a velocidade da passadeira rolante, qual o código do produto ou a sua ordem de fabrico (informação relativa a um objecto físico), qual a capacidade da máquina de processamento paralelo, etc. As *Primitivas Funcionais* caracterizam a forma como os diversos processos físicos relacionados com o fabrico se desenrolam, por exemplo o ciclo de vida de uma família de produtos ou o ciclo de trabalho de uma família de máquinas. Deste modo, enquanto o primeiro estabelece todas as operações relacionadas com o fabrico de um produto, desde a recepção da matéria prima até à expedição do produto final (gama

operatória), o ciclo de trabalho da máquina define os estados em que a mesma se pode encontrar incluindo mesmo os estados de excepção, i.e. parada, em preparação, avariada, em manutenção, e outros estados que poderão ser definidos no momento da construção do modelo.

O Modelo Fabril

O Modelo Fabril constitui sem dúvida uma peça fundamental para uma integração eficaz. No PROFIT, a modelação consiste na construção de uma representação da realidade fabril integrando as vistas física, funcional e de informação da fábrica ou sector fabril onde o sistema vai ser instalado. Estas vistas não coincidem no entanto com as da CIMOSA [61,62]. Na tabela 4.1 é caracterizada muito sucintamente a correspondência entre as vistas da CIMOSA e as vistas do PROFIT. Vejamos cada uma das situações com algum detalhe.

Tabela 4.1 - Comparação das Vistas na arquitectura CIMOSA e no PROFIT

PROFIT / **CIMOSA**	Vista Funcional	Vista de Informação	Vista Física
Vista Funcional	√		√
Vista de Informação		√	
Vista dos Recursos	√		√
Vista Organizacional			

Vista Funcional - Na CIMOSA, a vista funcional destina-se a permitir a observação da funcionalidade da empresa tendo em vista o planeamento, controlo e monitorização das operações de fabrico. No PROFIT estas facilidades surgem na vista funcional ou de processo, onde se pode observar o estado do processo produtivo para cada classe de produtos, e também na vista física onde o processo pode ser observado com mais detalhe, por exemplo o estado actual de uma ordem de fabrico em execução num determinado recurso.

Vista de Informação - Na CIMOSA, a vista de informação permite a observação das estruturas da informação do negócio durante as operações da empresa no decorrer dos processos de planeamento, controlo e decisão. No PROFIT, a vista de informação tem de facto o mesmo objectivo. Este conceito foi no entanto estendido, estabelecendo que se está perante uma vista de informação sempre que se trate de uma vista particular sobre a base de dados. Deste modo, por exemplo a aplicação de Gestão da Qualidade fornece ao seu utilizador uma vista de informação específica sobre o processo / recursos de fabrico e / ou produtos. A vista de informação é acedida através do modelo da instalação fabril permitindo que o acesso às aplicações se faça no contexto de um elemento físico do modelo. Na figura 4.13, (página 94), é apresentado um exemplo onde uma folha de trabalho da aplicação de Supervisão é aberta no contexto de um recurso de fabrico do modelo.

Vista de Recursos - A vista de recursos da CIMOSA permite a observação dos bens da empresa necessários à execução dos processos de fabrico, incluindo a utilização do modelo para gerir (controlar e monitorizar) esses bens. No PROFIT, estas facilidades encontram-se distribuídas entre a vista funcional, onde são modelados os ciclos de trabalho das classes de recursos existentes, e a vista física, onde os atributos de cada recurso individual podem ser monitorizados e controlados.

Vista Organizacional - Esta vista permite a caracterização das responsabilidades na empresa relativamente a cada função, informação, recurso, gestão de excepções e tomada de decisão. No PROFIT, esta vista não surge de forma explícita nem implícita no modelo fabril.

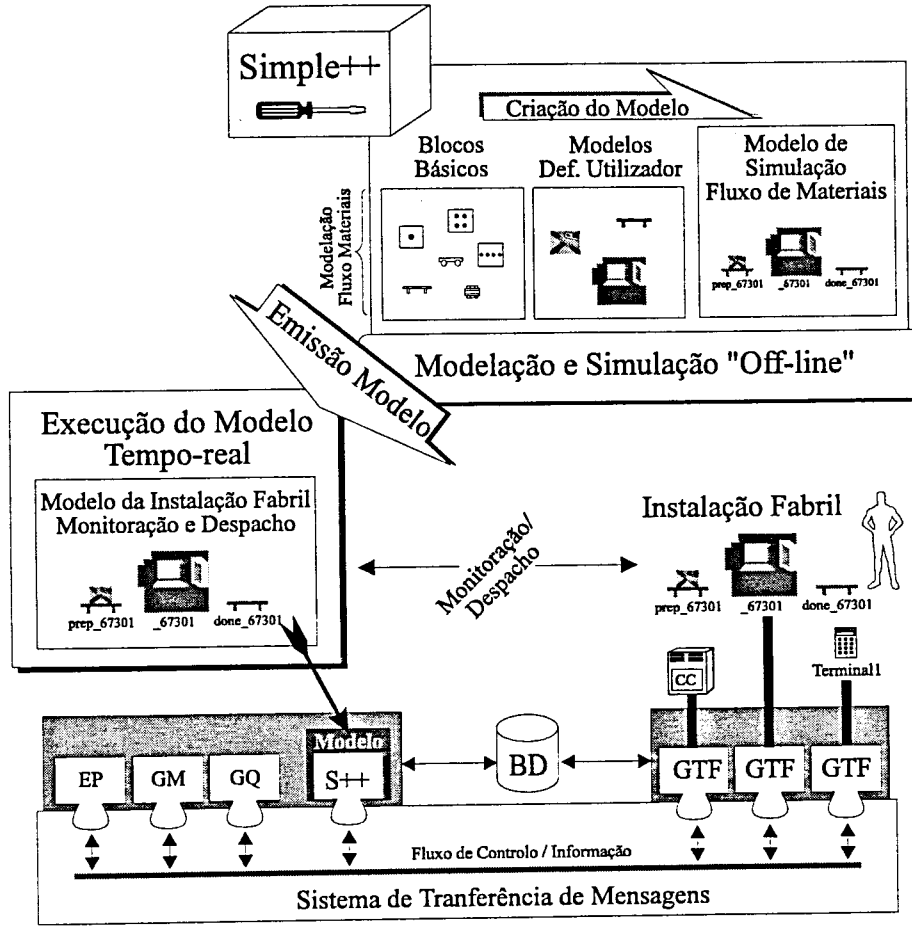
Através das diversas vistas do sistema, no PROFIT o modelo da instalação fabril permite ainda o acesso:

- à monitorização gráfica e animada em tempo real da instalação (para seguir, por exemplo, o estado actual das ordens e recursos de fabrico, ou os stocks do em-curso-de-fabrico)
- a funções de despacho e controlo da produção,
- à simulação baseada em eventos para suporte de estratégias de escalonamento da produção,
- a qualquer das aplicações do SHOP-CONTROL.

O Ciclo de Vida do Modelo Fabril

O Simple++, constitui uma infra-estrutura básica de suporte ao ciclo de vida do modelo fabril. Os dois ambientes de utilização do modelo atrás referidos assim como a sua interligação são postos em evidência na figura 4.6: o ambiente de execução em tempo real para monitorização e controlo das operações de fabrico ("on-line") e o ambiente de simulação do modelo suportando tarefas de engenharia e planeamento ("off-line").

Ao comparar com o quadro de modelação da CIMOSA, onde o cubo apresenta os três tipos de modelos empresariais (figura 3.10), são usados no PROFIT modelos que poderão ser classificados respectivamente como genéricos, parciais e particulares. A construção do modelo fabril passa com efeito pela utilização das primitivas básicas do Simple++ (modelos genéricos de componentes ou objectos fabris) na construção de modelos de objectos específicos necessários à construção do modelo da empresa ou área fabril alvo de estudo (modelos parciais). Estes modelos parciais definidos pelo utilizador serão finalmente utilizados na construção do modelo de fluxo de materiais da instalação fabril (modelo particular). Uma vez completo e validado, este modelo pode ser reutilizado (emissão do modelo na figura 4.6) no controlo da execução das operações de fabrico, suportando as funções de monitorização, despacho e controlo da produção. A execução do modelo de simulação evolui em tempo real e em paralelo com o efectivo desenrolar das operações, permitindo que o estado da produção na instalação fabril seja seguido e constantemente comparado com a situação desejada / planeada (i.e. monitorização). Esta comparação detecta eventuais desvios, e os correspondentes alarmes podem alertar o utilizador para situações em que o escalonamento de tarefas planeado fique comprometido, dando lugar a ajustes através de acções correctivas (controlo, despacho). A interacção em tempo real entre a instalação fabril e o modelo em execução ("on-line") permite ainda a actualização desse modelo sempre que surjam eventos relevantes, tais como avaria de uma máquina, atraso na movimentação de um lote em-curso-de-fabrico, ocorrência de problemas de qualidade, etc. Em qualquer dos casos é garantida a actualização em tempo útil da Base de Dados por forma a que ela espelhe constantemente o estado actual da instalação fabril.



Legenda: EP - Escalonamento da Produção, GM - Gestão da Manutenção, GQ - Gestão da Qualidade, S++-Simple++, GTF-Gestor de Transacções Fabris

Figura 4.6 - O Ciclo de Vida do Modelo Fabril no PROFIT

4.4.3 Arquitecturas

A Arquitectura em Camadas do PROFIT

O PROFIT apresenta na sua versão actual (1.0) uma infra-estrutura de integração perfeitamente estável, resultado da evolução natural do protótipo desenvolvido no decorrer projecto SHOP-CONTROL.

Os sistemas de gestão fabril são hoje em dia sistemas inerentemente distribuídos, constituídos por componentes mais ou menos independentes, trabalhando em paralelo, e comunicando e cooperando entre si. Com este objectivo, a infra-estrutura do PROFIT oferece, tanto ao utilizador final como à equipa de desenvolvimento, um conjunto de facilidades de integração das aplicações de gestão fabril. O PROFIT é pois um sistema heterogéneo distribuído, significando isto que as suas aplicações deverão ser capazes de correr em rede e em diversas plataformas de hardware e software, tais como PCs com MS-Windows ou estações de trabalho UNIX. Além disso, dada a necessidade de integração com sistemas legados, o PROFIT deve ainda permitir a ligação a terminais fabris de recolha de dados de diversos fabricantes e a máquinas MS-DOS.

Tal como ilustra a figura 4.7, o PROFIT apresenta uma interface com o utilizador comum que é utilizada para fins de modelação e simulação, monitorização e controlo, e ainda acesso às restantes aplicações. Estas fornecem ao utilizador um conjunto de serviços de gestão fabril, tais como supervisão do processo, escalonamento e controlo da produção, gestão da qualidade e da manutenção. O utilizador ao nível da estação de trabalho UNIX (figura 4.8) interage pois com o modelo fabril, em simultâneo um ponto de entrada no sistema e um ponto de acesso aos serviços PROFIT. Por exemplo, para aceder a informação de manutenção relativa a um recurso específico, o utilizador selecciona-o no modelo invocando seguidamente os serviços da aplicação de manutenção no contexto do mesmo recurso. Embora o utilizador em ambiente PC não tenha acesso ao modelo fabril, os serviços disponíveis são praticamente os mesmos que são fornecidos ao nível da estação de trabalho UNIX. Tanto o modelo da planta fabril como as aplicações, em MS-Windows ou UNIX, utilizam os serviços de comunicação fornecidos por um Sistema de Transferência de Mensagens. O acesso à base de dados PROFIT no ambiente UNIX é realizado através de bibliotecas de acesso ao cliente local do sistema gestor de base de dados. No ambiente MS-Windows o acesso ao servidor de base de dados do PROFIT é realizado através de uma interface ODBC ("Open Data Base Connectivity") (figura 4.7). Ao nível da instalação fabril, o operador poderá ter acesso a terminais de recolha de dados com maior ou menor capacidade de processamento. Enquanto que no primeiro caso poderá tratar-se de computadores pessoais adaptados ao ambiente industrial e dando acesso às diversas aplicações do PROFIT, no segundo teremos terminais muito simples dotados de um teclado industrial, "display" LCD e leitor de código de barras.

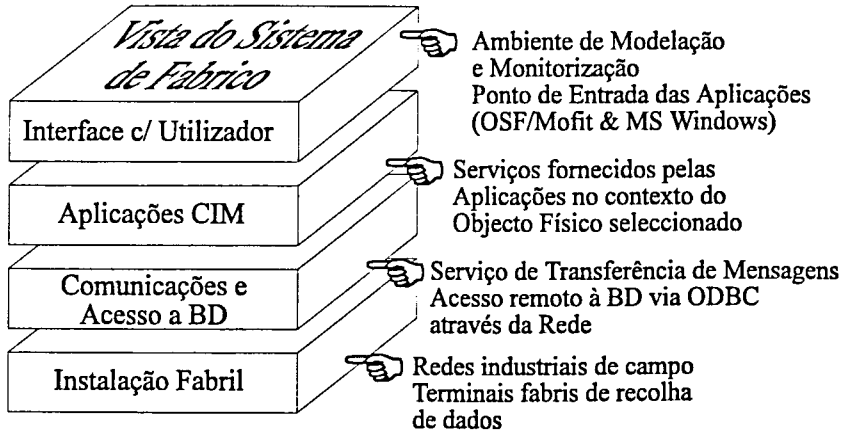


Figura 4.7 - A Infra-estrutura PROFIT em evidência no modelo por camadas

O PROFIT é assim um sistema integrado de gestão fabril dotado de uma estrutura modular que se baseia em cinco componentes fundamentais (figura 4.8):

- A interface com o utilizador, oferecendo simultaneamente a superfície de modelação e monitorização da instalação fabril e o acesso às aplicações.
- As aplicações, activadas no contexto do modelo da instalação fabril. No PROFIT, estão disponíveis aplicações de escalonamento e controlo da produção, monitorização e supervisão, gestão da qualidade e gestão da manutenção.
- A base de dados fabril, tipicamente uma infra-estrutura de integração partilhada por todas as aplicações e constituindo o repositório de informação do PROFIT. Este repositório armazena informação relativa à configuração do sistema assim como informação que tenha valor do ponto de vista histórico, além de conter sempre a

informação actualizada em tempo real sobre o estado da instalação fabril. A utilização do modelo entidade-associação na modelação da informação fabril, permitiu salvaguardar os aspectos de redundância, consistência e semântica da informação partilhada pelas diferentes aplicações.

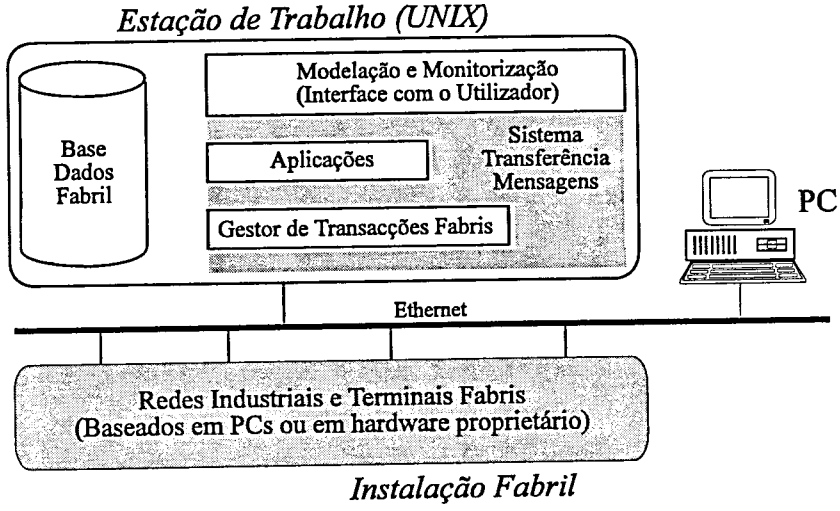


Figura 4.8 - Arquitectura do Projecto SHOP-CONTROL / PROFIT

- O Gestor de Transacções Fabris (GTF) que gere a interacção entre as aplicações referidas acima e a instalação fabril. A base de dados contém informação actualizada relativa ao estado da instalação fabril representada no modelo que é actualizada pelas aplicações e pelo GTF. É assim possível manter na base de dados uma imagem actualizada do estado da instalação na qual o GTF tem um papel particularmente importante de actualização em tempo real.
- O Sistema de Transferência de Mensagens, que suporta de uma forma transparente as comunicações entre todos os módulos / aplicações do sistema distribuídos por máquinas e ambientes diversos interligados por uma rede local Ethernet.

A Arquitectura do Sistema no projecto REAL-I-CIM

No projecto REAL-I-CIM (RIC), embora a arquitectura do sistema seja ainda ponto de discussão na altura da escrita desta dissertação, tudo aponta para uma influência forte do modelo de referência para ambientes CASE da ECMA ("European Computer Manufacturers Association") [63,64]. O modelo "torradeira" ilustrado na figura 4.9 resume as recomendações da ECMA para os ambientes CASE [65,66]. De facto, tem vindo a ser realizado considerável esforço de investigação na integração de ferramentas na área engenharia de software, esforço esse passível de ser orientado para aplicação em diferentes áreas, nomeadamente a de arquitecturas de suporte a sistemas integrados de fabrico.

A referida integração de ferramentas pressupõe os seguintes serviços básicos: o *serviço de repositório de dados*, fornecendo mecanismos para identificar e armazenar os dados e a suas relações, assegurando transparência na sua localização e fazendo a gestão do acesso concorrente; o *serviço de integração de dados*, adicionando facilidades extra ao serviço anterior, e permitindo a configuração de versões e facilidades de gestão de meta-modelos; o *serviço de gestão de tarefas*, que fornece o apoio no tratamento de tarefas (uma tarefa pode ser vista como uma sequência de operações elementares executadas por outros serviços ou ferramentas, executadas a pedido ou sempre que um estado definido por um conjunto de

pré-condições seja atingido); o *serviço de mensagens*, que providencia um mecanismo de comunicação standard usado pelas ferramentas e serviços; os *serviços de interface com o utilizador*, que permitem que a interacção entre o utilizador do ambiente CASE e todas as suas ferramentas se faça através de uma interface uniforme e consistente.

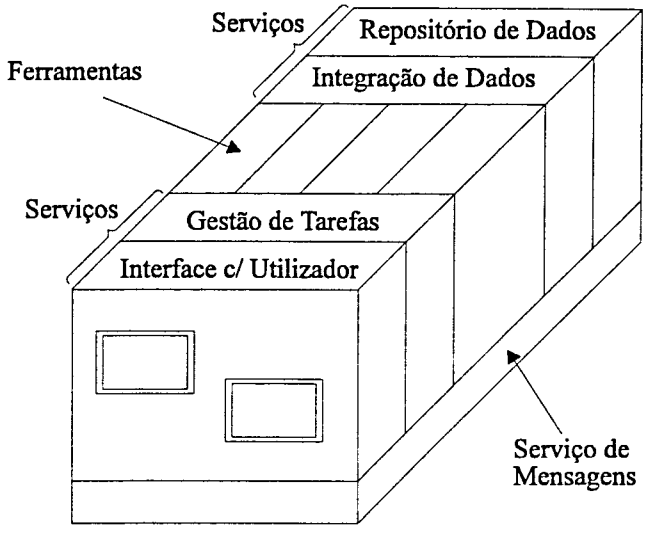


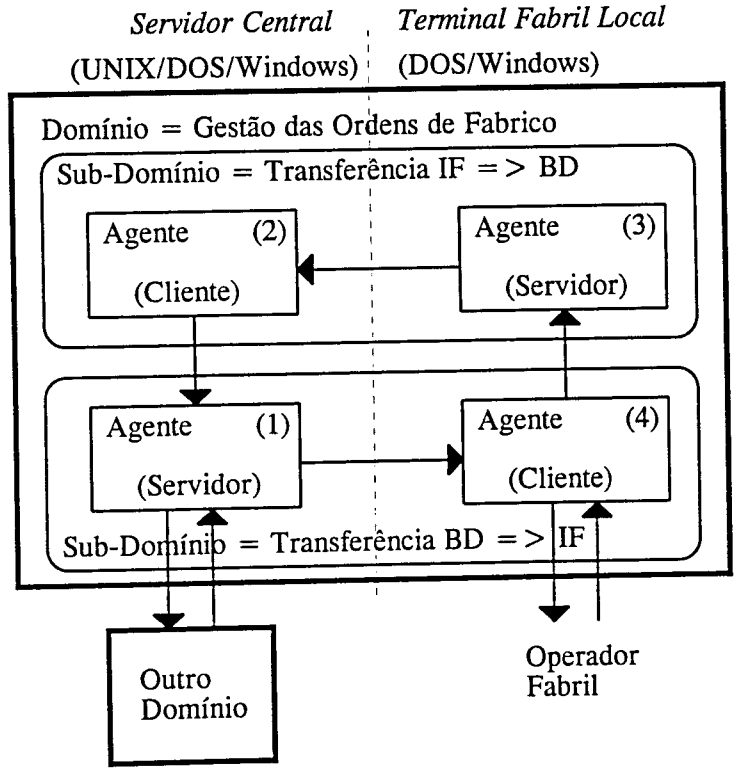
Figura 4.9 - Modelo de Referência para ambientes CASE de acordo com a ECMA

Existe ainda, conforme foi referido acima, alguma discussão no seio do consórcio do projecto RIC quanto a este tema, que é certamente polémico pela susceptibilidade de provocar o confronto entre abordagens distintas à concepção e desenvolvimento de sistemas. Por um lado surgem os defensores do conceito de uma base de dados lógica única permitindo a integração total da informação, e por outro surgem os que advogam a utilização de servidores dedicados de informação de elevado nível semântico, de acordo com missões bem definidas no âmbito da instalação fabril. No âmbito do RIC estão de facto a realizar-se esforços para minimizar ou mesmo ultrapassar as dificuldades que surgiram no decurso de instalações dos resultados do projecto SHOP-CONTROL na manutenção da estrutura normalizada da base de dados fabril. Por este motivo, a arquitectura definida pela ECMA vai certamente levar, no âmbito do RIC, a uma combinação de ambas as abordagens, recorrendo-se à utilização de múltiplos servidores de informação distribuídos sem no entanto excluir a utilização de uma base de dados lógica comum com um papel integrador.

Tendo em vista esta linha de orientação, já se encontram definidos alguns servidores que serão usados nas primeiras instalações protótipo do PROFIT no âmbito do RIC, devendo no entanto ser definidos outros a curto prazo. Com este objectivo é utilizado um método de "análise orientada ao agente" [67] onde são identificados numa primeira fase domínios, englobando missões (papéis) bem definidas(os) que no seu conjunto caracterizam as funcionalidades do RIC.

Para os servidores de carácter estruturante já definidos, foram caracterizados três domínios específicos, que englobam respectivamente a gestão das ordens de fabrico, dos recursos de fabrico e dos produtos. Na caracterização de cada um destes domínios foram identificados sub-domínios, sendo cada um deles responsável por uma missão em particular: por exemplo, transferência de informação no sentido Instalação Fabril (IF) → Base de Dados (BD) e transferência de informação no sentido BD → IF. No interior de cada um dos sub-domínios foram então separadas funcionalidades a serem corporizadas em agentes responsáveis pela sua execução. A figura 4.10 ilustra o domínio da Gestão de

Ordens de Fabrico, colocando em evidência as funcionalidades inerentes à necessária interacção entre a instalação fabril (via terminais fabris) e a base de dados. O agente responsável pelo armazenamento da informação relativa às ordens de fabrico, agente servidor (1), é consultado pelo agente cliente (4) sempre que o operador fabril executa um pedido de informação. O agente (1) também será consultado quando um outro qualquer domínio necessita de informação relativa a ordens de fabrico. Os agentes (2) e (3) funcionam, tal como no caso anterior, de uma forma complementar, i.e., o servidor armazena informação local que é transferida pelo cliente, o agente (2), para o agente (1) que se mantém desta forma actualizado.



Legenda: As setas indicam o sentido da transferência de informação

Figura 4.10 - Aplicação do método de "Análise do Sistema Orientado ao Agente" ao domínio da Gestão das Ordens de Fabrico

4.4.4 O Serviço de Transferência de Mensagens

No projecto SHOP-CONTROL foram estudados diversos métodos passíveis de serem utilizados para suportar a implementação de mecanismos de comunicação entre aplicações, nomeadamente [3]: (1) comunicação "inter-client" no X-Windows [3]; (2) comunicação entre processos UNIX através de mensagens, semáforos, memória partilhada ou "sockets" [3]; (3) filas de espera na Base de Dados. O principal problema das duas primeiras opções é o facto de provocarem o aparecimento de um grande número de ligações entre as aplicações, com todos os inconvenientes daí decorrentes. No primeiro caso, seria mesmo impossível a comunicação entre aplicações residentes em diferentes máquinas na rede. A terceira opção impõe uma sobrecarga extra ao servidor da base de dados, exigindo de qualquer forma um dos métodos anteriores como mecanismo para o disparo da

comunicação. De entre as diversas alternativas optou-se por um "Sistema de Transmissão de Mensagens" (STM) construído sobre "sockets", uma solução mais geral cuja implementação apresenta a vantagem adicional de fornecer uma interface ao nível da camada de transporte no modelo OSI [2,3,68].

Tendo sido a integração de módulos / aplicações de software distintos e independentes um dos principais objectivos subjacentes à definição do serviço de transmissão de mensagens, identificam-se de seguida as características do mesmo que tornaram possível atingir tal meta: interface de ligação muito simples entre um módulo de aplicação e o STM; identificação de cada módulo independente da máquina em que reside assim como da localização desta na rede local de comunicação de dados; implementação do conhecido conceito de caixa de correio electrónico; disponibilização do serviço de uma forma distribuída numa grande variedade de máquinas e sistemas operativos.

A figura 4.11 ilustra de forma simplificada o STM utilizado no PROFIT, mostrando os processos Agentes do STM residentes em cada uma das máquinas UNIX do sistema distribuído. Cada processo de aplicação que deseje dispor dos serviços fornecidos pelo STM deverá *registar-se* no sistema como pertencendo a uma determinada família (grupo de processos) e com um determinado nome de membro que deverá ser único no seio daquela família. A partir deste momento o processo de aplicação poderá enviar e receber mensagens cuja origem e destino são identificados pelo par (família, membro). Os ditos processos emissor e receptor são responsáveis respectivamente pela codificação e descodificação da informação contida na mensagem, uma vez que o STM é unicamente um mecanismo de transporte de mensagens não realizando qualquer interpretação do seu conteúdo.

Como ilustra a figura 4.11, os processos correndo nos PCs MS-DOS ou MS-Windows deverão ligar-se a um dos agentes residentes em uma das máquinas UNIX do sistema distribuído, agente no qual irá residir a sua caixa do correio. A independência entre o endereço lógico dado pelo par (família, membro) e o endereço físico do processo permite que uma aplicação possa ser deslocada de uma máquina para outra sem que isso implique a sua recompilação ou reconfiguração. O STM desenvolvido constitui o elo de ligação entre todos os restantes módulos do sistema, sendo o elemento da infra-estrutura fundamental no suporte à sua integração total [68]. O serviço, disponibilizado em qualquer ponto da rede local Ethernet, permite que mensagens devidamente estruturadas fluam de forma eficaz e transparente, implementando ligações de informação e de controlo entre os diversos módulos de aplicação com vista ao desenvolvimento cooperativo de um determinado conjunto de actividades.

Todas as comunicações efectuadas entre os diversos módulos do sistema assentam numa transmissão de mensagens entre dois pontos / processos. Durante a definição deste serviço também se reflectiu na facilidade de, a partir de um único ponto, ser possível desencadear a transmissão de uma única mensagem para um conjunto de pontos associados (em modo "multicast"). O suporte para a sua implementação consideraria o agrupamento dos módulos de aplicação de software em grupos, grupos esses que poderiam eventualmente ser desdobrados em sub-grupos de uma forma hierárquica. Este tipo de organização de entidades participativas num ambiente de comunicação de dados, permitiria concretizar comunicações "multicast", na medida em que um dado módulo de um grupo poderia desencadear a transmissão de uma mensagem para todos os membros do seu grupo (ou subgrupo) compondo-a e endereçando-a de uma única vez. Sendo complexo o desenvolvimento de tal facilidade, não foi considerada prioritária a sua implementação nesta versão 1.0 do PROFIT, embora se considere poder vir a ser uma facilidade interessante para desenvolvimentos posteriores do Gestor de Transacções Fabris.

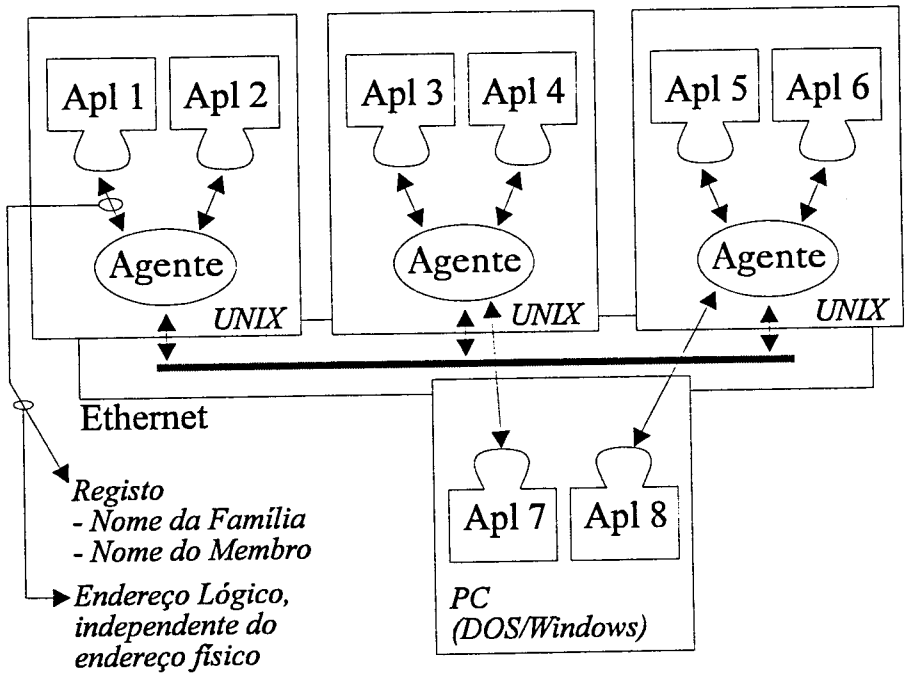


Figura 4.11 - Sistema de Transferência de Mensagens - esquema simplificado da arquitectura distribuída

Com o objectivo de permitir de uma forma simples e transparente a transferência assíncrona de mensagens, são suportadas pelo STM diversas formas de notificação da aplicação receptora de que uma nova mensagem chegou à sua caixa do correio: a sondagem ("Polling"), sinais ("Signals"), eventos X [69]. No primeiro caso, a aplicação invoca periodicamente um procedimento (*mts_mbox_size*) que lhe indica o número de mensagens que o seu agente local do STM tem para si. Este primeiro caso não se trata de facto de uma notificação, uma vez que é do destinatário a iniciativa para a sondagem à sua caixa de mensagens. Na utilização de sinais, o processo UNIX a ser notificado é interrompido pelo agente local do STM que lhe envia um sinal (SIGUSR1) sempre que chegue uma nova mensagem. Este sinal dispara no processo a notificar a execução de um procedimento definido pelo utilizador (*sigusr1_handler*) cuja função consiste em incrementar um contador (*sigusr1*) que registará o número de mensagens na sua caixa de correio. Problemas tais como o acesso mutuamente exclusivo à variável *sigusr1* foram adequadamente tratados e são expostos em [69]. A utilização de eventos X é realizada usando uma função do X-toolkit (*XtAppAddInput*) para associar a execução de um procedimento definido pelo utilizador à alteração de um "socket" atribuído à aplicação no momento em que ela se regista no STM [69].

O STM desenvolvido no âmbito do SHOP-CONTROL foi melhorado para apoiar primeiro a evolução do protótipo para o produto comercial PROFIT, e seguidamente as novas exigências da arquitectura REAL-I-CIM (RIC). A especificação das novas funcionalidades suportaram-se da experiência ganha com a sua utilização no desenvolvimento em diferentes cenários. Todavia, funcionalidades inicialmente previstas como o "broadcast" e o "multicast" foram consideradas não prioritárias pelo consórcio RIC por se ter considerado não serem suficientemente relevantes para o produto final em relação a outros esforços necessários. Foram no entanto propostas novas facilidades apresentadas na tabela 4.2.

Tabela 4.2 - Novas Facilidades do STM no PROFIT / RIC

Novas Facilidades	Descrição
Armazenamento ("Buffering")	Permite que um processo se desligue do STM durante um período de tempo limitado. Durante este período, todas as suas mensagens são armazenadas sendo-lhe entregues assim que o processo se ligue novamente ao sistema.
Monitorização	A função de monitorização permite examinar o estado do sistema distribuído que constitui o sistema de transferência de mensagens, para saber por exemplo quais as aplicações registadas no sistema, qual a dimensão da caixa de correio de cada uma das aplicações, etc.
Tamanho de mensagem ilimitado	Nas versões anteriores, a mensagem a ser enviada tinha um tamanho limite, motivo pelo qual o sistema não podia ser utilizado na transferência de grandes blocos de informação. Na versão já disponível, sempre que a mensagem exceder esse tamanho (>y bytes), ela é fragmentada em várias mensagens mais pequenas. Essas mensagens são recebidas e reconstituídas à chegada para serem entregues à aplicação destino.
Chamada Remota de Procedimentos	Sobre o STM, é emulada uma funcionalidade semelhante ao "Remote Procedure Call", permitindo um tratamento simples da Chamada Remota de Procedimentos.

Tendo em vista a utilização genérica do STM, um dos aspectos a considerar no desenvolvimento de um sistema de comunicação de dados é o do seu desempenho global. Por este motivo, a implementação em curso de cada uma destas novas facilidades vai passar por uma fase de avaliação do seu impacto no desempenho global do STM.

4.4.5 Integração de Controlo, de Dados e da Interface com o Utilizador

Introdução

Será seguidamente apresentada a classificação da infra-estrutura do PROFIT, relativamente aos três tipos de integração que normalmente se distinguem, a *integração dos dados*, do *controlo* e da *interface com o utilizador*. Para cada um destes tipos de integração são estabelecidos os níveis de *inter-operacionalidade*, *integração parcial* e *integração total*. Estes níveis de integração são incrementais, ou seja, a integração parcial é um pré-requisito para a integração total, assim como deve ser verificada a completa inter-operacionalidade para se atingir a integração parcial [65].

Integração de Dados - A integração de dados diz respeito à forma como as diferentes aplicações trocam e partilham informação. A inter-operacionalidade dos dados refere-se a aplicações ligadas através de mecanismos de transferência de dados (como o STM referido no ponto anterior) ou partilhando um repositório de informação (por exemplo, uma base de dados). Quando as aplicações têm um entendimento mútuo dos formatos dos dados, diz-se

que estão parcialmente integradas. Finalmente, estaremos em face de uma integração total se, além dos requisitos anteriores, partilharem o mesmo modelo de dados.

Integração de Controlo - A integração de controlo existe se as aplicações em questão trocarem eventos entre si. A comunicação básica de eventos pode, por exemplo, resultar no lançamento automático de uma aplicação. Neste caso, onde os eventos são apenas causadores do disparo de acções / tarefas e não contêm qualquer informação, estamos no nível da inter-operacionalidade. A integração parcial corresponde à capacidade das aplicações utilizarem informação associada a estes eventos. Estaremos em face da integração de controlo total se as aplicações cooperam para atingir um determinado objectivo ou para executar uma tarefa em particular.

Integração da Interface com o Utilizador - A inter-operacionalidade ao nível da interface com o utilizador (IU) corresponde simplesmente à capacidade de duas aplicações partilharem o mesmo dispositivo de visualização. A integração parcial da IU refere-se ao facto de duas aplicações distintas apresentarem primitivas visuais e de comportamento semelhantes. A integração total corresponde à utilização das mesmas primitivas visuais e de comportamento, conduzindo a um único modo de interacção com o utilizador. No ambiente X-Windows e OSF/Motif, esta integração total pode ser conseguida seguindo as recomendações da OSF ("Open Software Foundation") para a construção de interfaces. No entanto, a situação complicar-se-á se co-existirem dois ambientes, tais como OSF/Motif [70] e Open Windows [71], onde os objectos e os mecanismos de interacção não são sempre coincidentes.

Integração de Dados

No projecto SHOP-CONTROL foi realizada a integração total dos dados. Durante todo o processo de especificação e desenvolvimento das aplicações achou-se necessário manter a consistência do modelo de dados do sistema de informação [72]. A utilização de uma base de dados comum facilitou esta tarefa, que se traduziu na manutenção do modelo entidade associação do sistema de informação e na utilização de um dicionário de dados associado, tendo-se chegado ao modelo normalizado da base de dados [73].

A arquitectura do RIC descrita anteriormente traduz como vimos uma evolução, do conceito de sistema de informação fabril baseado numa base de dados central com total integração de dados e de controlo, para uma arquitectura cliente-servidor onde a informação se encontra distribuída por um conjunto de domínios que se identificam com um conjunto de missões bem definidas. Nesta perspectiva, não se poderá falar numa integração total de dados entre aplicações, uma vez que essa informação se encontra distribuída num conjunto de bases de dados locais associadas a um agente particular. Será de referir o aparecimento do conceito de serviço, um conceito horizontal permitindo agrupar num todo coerente funcionalidades com o objectivo de constituir uma mesma abstracção. É exemplo disso o chamado Servidor de Ordens de Fabrico ("Manufacturing Order Server") que armazena e gere toda a informação relativa à execução e estado das ordens de fabrico.

No RIC é garantida apenas uma integração de dados parcial, uma vez que as aplicações que comunicam podem ou não partilhar o mesmo modelo de dados. De facto, nesta nova arquitectura a informação é acedida através de um serviço, podendo cada aplicação construir o seu modelo de dados (privado) da forma que for mais útil e eficiente para o seu processamento local. Apesar disso, as diversas aplicações que partilham um determinado conjunto de serviços deverão todavia ser capazes de conhecer e tratar um mesmo modelo de dados desses mesmos serviços.

Maior flexibilidade e eficiência foram os objectivos que estiveram na base desta evolução de conceitos.

Integração de Controlo

Integração Total no Universo SHOP-CONTROL

No SHOP-CONTROL as aplicações são módulos independentes comunicando entre si através de mensagens e partilhando a informação armazenada na base de dados de acordo com um conjunto de regras e uma semântica comum, de forma a proporcionar um ambiente integrado. A integração das aplicações foi conseguida não só através do modelo da instalação fabril, que permite um ambiente homogéneo e coerente de acesso a todas as aplicações, mas também através de uma base de dados fabril comum e de uma infra-estrutura que permite a troca de mensagens entre as aplicações. Com o objectivo de regulamentar a comunicação entre as aplicações foi estabelecida uma "matriz de controlo", que define quais as mensagens trocada entre as diversas aplicações de acordo com uma sintaxe estabelecida. É deste modo estruturada a integração de controlo dessas aplicações, permitindo atingir a integração de controlo total [74]. Um exemplo típico de aplicação prática de uma troca de mensagens de controlo entre três aplicações no âmbito de SHOP-CONTROL, seria dado pela seguinte situação: a aplicação de controlo da qualidade detecta um problema, e diagnostica uma acção correctiva que exige a manutenção urgente de um determinado recurso de fabrico; esta aplicação envia então uma mensagem à aplicação de manutenção indicando as suas necessidades de intervenção e outra mensagem à aplicação de escalonamento informado que o recurso foi colocado fora de serviço ficando portanto indisponível.

Origem / **Destino**	Simple++ (Modelo de Simulação)	Supervisão	GTF	...	Manutenção	Qualidade
Simple++ (Modelo de Simulação)						
Supervisão						
GTF						
.....						
Manutenção						Cód_Msg Cód_Recurso Cód_Acção
Qualidade						

Figura 4.12 - Matriz de Controlo, exemplificando o conteúdo de uma mensagem enviada pela Qualidade à Manutenção

A figura 4.12 ilustra a estrutura da matriz de controlo definida no âmbito do projecto SHOP-CONTROL. Esta matriz de duas entradas define quais as aplicações que trocam mensagens (repare-se que nenhuma aplicação troca mensagens consigo própria), e quais as mensagens trocadas, incluindo o seu código identificador e os respectivos atributos, assim como os valores válidos para esses mesmos atributos. No exemplo, temos a mensagem enviada da aplicação de Gestão da Qualidade para a aplicação de Gestão da Manutenção onde, além do código identificador da mensagem, é indicado o código do recurso e da acção de manutenção que sobre ele é requerida.

Inter-operacionalidade com o exterior do Universo SHOP-CONTROL

Na integração de um novo sistema informático com um sistema ou infra-estrutura já instalados na fábrica, existe a necessidade evidente de que os sistemas existentes sejam incluídos no processo de evolução para o novo sistema integrado. O PROFIT, com a sua modularidade e abertura, pode garantir a salvaguarda dos investimentos já realizados, permitindo que as suas potencialidades sejam aproveitadas unicamente na medida das necessidades de evolução dos sistemas de gestão fabril existentes.

No âmbito do SHOP-CONTROL / PROFIT, o Sistema de Transferência de Mensagens constituiu de facto um meio privilegiado para a integração das aplicações. No entanto, a inclusão num curto período de tempo de novas aplicações substituindo módulos antigos num ambiente existente, obriga a enfrentar problemas tais como: protocolos rígidos na interacção de aplicações compiladas e não reconfiguráveis; impossibilidade de colocar dois módulos configurados para protocolos distintos a comunicar entre si; incompatibilidade entre os dicionários de dados utilizados pelas aplicações PROFIT e pelas aplicações externas a incorporar (problemas semânticos), e incompatibilidade entre os formatos de mensagens (problemas sintácticos); necessidade de estabelecer diálogo com aplicações que não possuem ligação ao STM. As primeiras instalações do PROFIT reforçaram entretanto a necessidade do estabelecimento de ligações com infra-estruturas informáticas já instaladas, nomeadamente com sistemas de planeamento da produção (MRP, CRP, stocks), de transferência de informação inter-empresarial (EDI), de recolha de dados da instalação fabril, de gestão de energia e de controlo da qualidade.

Na sequência da conclusão do projecto SHOP-CONTROL, e tendo em vista a resolução do problemas enunciados acima, foram apontados pelo consórcio alguns dos desejáveis desenvolvimentos futuros, nomeadamente o desenvolvimento de uma interface conforme com o standard MMS ("*Manufacturing Message Specification*") [75], que asseguraria a independência de redes industriais e terminais fabris proprietários. Todavia, após a realização de uma análise de mercado cuidada, o consórcio RIC decidiu que para o mercado alvo a utilização do MMS implicaria soluções mais complexas e consideravelmente mais caras.

Integração de Controlo no RIC

Ainda em discussão no seio do consórcio. São assim tópicos em estudo, a necessidade de ferramentas de configuração de interfaces com MRP's existentes, ou a flexibilização da infra-estrutura de integração por forma a permitir a construção rápida e à medida de interfaces com aplicações existentes.

Integração da Interface com o Utilizador

Integração da Interface com o Utilizador no SHOP-CONTROL

No OSF/Motif existem recomendações gerais para o desenvolvimento das interfaces gráficas, e a importância do problema justificou no projecto SHOP-CONTROL a existência de uma "Workpackage" para o efeito. Neste projecto, o desenvolvimento de uma interface com o utilizador coerente e homogénea para todo o sistema apresentava dificuldades acrescidas pelo facto de existirem equipas de desenvolvimento de software de diversos parceiros em países diferentes da Europa. No arranque do projecto foram pois estabelecidos princípios e regras gerais que, a serem seguidos, permitiriam que o sistema fosse identificado como um todo coerente [76]. A interface poderia então ser vista como uma "entidade combinada" de interface, i.e., um conjunto consistente e articulado de peças de

tipografia, símbolos, cor, desenho espacial, e uma dada sequência de diálogos interactivos, que caracteriza um sistema em particular [50,77].

Além das regras impostas à especificação das interfaces gráficas foi considerado necessário que o desenvolvimento fosse suportado por uma ferramenta comum: um protótipo de um gerador de interfaces gráficas [78]. Atrasos na estabilização dessa ferramenta, que proporcionava uma camada sobre o standard X-Windows / OSF/Motif, agravados pelo imperativo de a portar para diferentes plataformas, levaram ao seu abandono. Assim sendo, no SHOP-CONTROL as interfaces com o utilizador para as diversas aplicações foram realizadas directamente sobre OSF/Motif.

Na integração da interface com o utilizador foi apenas atingido o nível de integração parcial. Embora fossem realizados esforços no sentido de ser atingida a integração total ela não foi nunca conseguida [76], essencialmente pela dificuldade de impor e coordenar desenvolvimentos paralelos de diferentes equipas em quatro países diferentes.

Integração da Interface com o Utilizador no RIC

A existência como fundo de um produto comercial como o PROFIT impõe cuidados especiais para disciplinar o desenvolvimento das aplicações interactivas. A complexidade do problema vem ainda ampliada pelo facto de as aplicações estarem a ser desenvolvidas para os ambientes OSF/Motif (sobre X-Windows) e MS-Windows.

As dificuldades sentidas no projecto SHOP-CONTROL, que levaram mesmo à incapacidade de realizar uma interface homogénea e coerente para os diferentes utilizadores e pontos de entrada do sistema, aconselharam rigor e disciplina nos desenvolvimentos do projecto RIC. A complexidade é neste caso agravada pelo maior número de parceiros do consórcio e pela arquitectura genuinamente distribuída do RIC.

Assim sendo, foi decidido pelo consórcio que, sendo a ferramenta de modelação e simulação (Simple++) a aplicação mais estável e com maior penetração comercial de todas as aplicações do sistema, deveria ser seguido o seu "look and feel". Para além deste pressuposto seguem-se as recomendações gerais estabelecidas para o OSF/Motif. Espera-se que desta forma seja possível atingir o nível máximo de integração já referido anteriormente.

4.4.6 As Aplicações

Neste ponto não serão tratadas todas as aplicações disponíveis no PROFIT, mas apenas aquelas em cuja concepção, desenvolvimento e validação o autor deste trabalho de dissertação esteve directamente envolvido. Estas aplicações podem de facto ser vistas como peças ou componentes de carácter infraestrutural numa perspectiva lata, razão pela qual se considera que têm pleno cabimento no âmbito deste trabalho. Serão assim apresentadas as aplicações de Supervisão, e a sua migração para o Quadro de Bordo Dinâmico, bem como o Módulo de Acesso Distribuído à Instalação Fabril (MADIF). Ainda neste contexto, será exposto o conceito em estudo para a aplicação de monitorização de condição no projecto REAL-I-CIM.

A Aplicação de Supervisão

Introdução

No projecto SHOP-CONTROL [79,80], o principal objectivo estabelecido para a aplicação de Supervisão foi o de fornecer ao utilizador informação sobre o estado actual da instalação fabril (IF), e notificá-lo sempre que algo relevante ou inesperado ocorresse. Por

outro lado, a Supervisão deveria ainda fornecer meios para a análise de informação histórica sobre a IF.

Ao módulo de Supervisão, inovador na sua aproximação ao problema da supervisão de processos discretos de fabrico, foi reconhecida a sua grande flexibilidade de configuração e adaptação aos problemas específicos, assim como as potencialidades dos conceitos.

A experiência adquirida com a utilização da Supervisão, primeiro na instalação piloto do SHOP-CONTROL na Cifial e depois na primeira instalação do PROFIT na Arjal, permitiu validar a já reconhecida flexibilidade e o valor dos conceitos envolvidos. Colocou no entanto em evidência deficiências ao nível da interface com o utilizador, deficiências essas que se centravam particularmente nas capacidades gráficas limitadas (suportando apenas simples gráficos de barras). Este facto levou à revisão do anterior conceito de Supervisão e à sua reformulação no âmbito do projecto REAL-I-CIM, procurando-se agora a utilização de ferramentas standard e sacrificando alguma flexibilidade de integração.

Após esta introdução inicial, será respeitada a cronologia dos acontecimentos com a descrição da aplicação de Supervisão desenvolvida no âmbito do projecto SHOP-CONTROL e a apresentação dos requisitos básicos para a nova aplicação de Supervisão, o Quadro de Bordo Dinâmico, no âmbito do projecto REAL-I-CIM.

Funcionalidades Requeridas à Supervisão

Como ferramenta capaz de fornecer informação em tempo real sobre a IF, a Supervisão deve permitir a afixação em "tempo real" dos valores actuais de variáveis e parâmetros seleccionados entre os recolhidos na instalação fabril, além de permitir a realização de cálculos simples sobre essas mesmas variáveis. Estas são funções típicas dos sistemas SCADA ("Supervisory Control and Data Aquisition"), imprescindíveis no controlo e supervisão de processos contínuos de fabrico, mas só muito recentemente transportados para os processos discretos. Esta aplicação deveria ainda permitir a definição de regras operativas sobre as variáveis monitorizáveis, por forma a poder notificar o utilizador da ocorrência de um determinado evento ou condição pré-estabelecida. O acesso a informação histórica relativa à evolução temporal das mesmas variáveis deveria ainda estar disponível através da integração de primitivas SQL para o acesso a um servidor de base de dados.

Folha de Cálculo, o Conceito

A especificação de requisitos sumariamente apresentada acima levou à utilização, como componente nuclear da aplicação, de um gestor de regras semelhante ao motor de uma folha de cálculo [81]. Este gestor de regras foi dotado de uma interface com o utilizador semelhante à de uma folha de cálculo, e ligado às infra-estruturas que fornecem os serviços de acesso à base de dados, de transmissão de mensagens entre aplicações e de acesso à IF (figura 4.14). Como ferramenta de análise, a folha de cálculo é uma excelente escolha, permitindo ao utilizador a realização de cálculos, apresentados sob a forma tabular ou gráfica, combinando um qualquer número de valores que reflectem o estado de variáveis vindas da IF. Por outro lado, no modelo hierárquico da IF o utilizador pode definir uma folha de trabalho associada a cada recurso, por exemplo célula de fabrico, máquina ou estação de trabalho, onde é definida a informação a ser transferida da IF (de "tempo-real") ou da base de dados (histórica) (figura 4.13). O utilizador pode ainda recolher informação de qualquer outra folha(s) associada(s) a outro recurso(s) para a realização de folhas sumário referentes a grupos de recursos. Este tipo de mapa resumo pode ser obtido através de um conjunto de células de uma folha referindo células de outras folhas associadas a objectos (recursos) no nível hierárquico inferior.

A eventual dificuldade de utilização para o operador típico da instalação fabril de uma folha de cálculo poderia aconselhar a sua não utilização. No entanto, o operador do sistema deverá ser treinado para ser capaz de consultar informação de uma Tabela ou gráfico pré-definidos, e a popularidade de aplicações como o Excel ou Lotus 123 provam a evidente facilidade de utilização intuitiva dos conceitos envolvidos.

A Aplicação de Supervisão Integrada no SHOP-CONTROL

Tal como foi referido, a Supervisão fornece ao utilizador informação integrada sobre o estado actual da IF (por exemplo, estado dos recursos, nível do em-curso-de-fabrico, produtividade de uma célula / linha, ou simples contagens). Esta informação em tempo real é apresentada sob a forma de tabelas em folhas de trabalho ou gráficos, e é acessível através do modelo previamente construído da IF, simplesmente por selecção do objecto do modelo seguida da invocação da aplicação de Supervisão. Deste modo, para qualquer recurso físico seleccionado pelo utilizador na vista física do modelo fabril poderá ser activada a sua Vista de Informação [62]. A figura 4.13 ilustra com dois exemplos do acesso à Supervisão: no primeiro caso é seleccionado um objecto representando o grupo de recursos 67301/2 e a Supervisão é invocada neste contexto, dando acesso a informação condensada numa folha de trabalho sobre os recursos ou grupos de recursos que se encontram hierarquicamente debaixo daquele objecto; no segundo caso é seleccionado apenas um recurso físico no modelo e é invocada a Supervisão, sendo aberta uma folha de trabalho contendo apenas informação sobre o recurso escolhido.

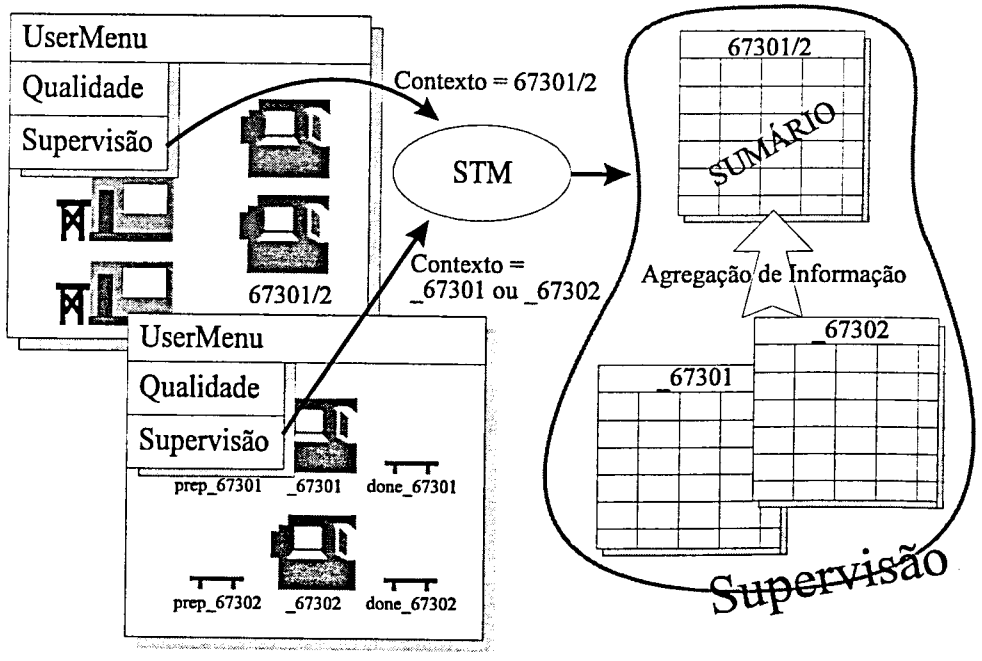


Figura 4.13 - Invocação da aplicação de Supervisão no contexto de um objecto (neste caso, um recurso) do modelo fabril

O gestor de regras de Supervisão mantém actualizado um conjunto de folhas de trabalho com informação da IF [81]. Para tal, e com o objectivo de permitir a interacção com a base de dados, com a IF, com o sistema de mensagens e com qualquer outro processo servidor exterior (figura 4.14), foi desenvolvido um conjunto suplementar de funções a serem tratadas pelo gestor de regras. Deste modo, o gestor de regras pode tratar comandos de envio de mensagens, de invocação de uma função por RPC ("Remote Procedure Call") ou primitivas em SQL, além de outras funções básicas tais como de abertura ("SheetOpen")

ou de fecho ("SheetClose") de uma folha de trabalho ou de abertura de um gráfico de barras ("Bartchart").

A utilização da Supervisão é tão simples como a de uma folha de cálculo. O utilizador, na fase de configuração da aplicação, define "macros" estabelecendo a desejada transferência de dados da instalação fabril. A folha poderá conter fórmulas, fórmulas essas incluindo "queries" SQL ou primitivas de acesso ao STM para envio e / ou recepção de mensagens, permitindo a consulta da BD (por exemplo sempre que um novo evento "chega" da IF) ou o envio de uma mensagem de controlo ou de alarme (por exemplo sempre que o valor de uma célula ultrapassa um determinado valor limite pré-fixado). O utilizador pode ainda recorrer a outras funções para abrir e fechar janelas de gráficos. Finalmente, a utilização da primitiva de RPC permite ao utilizador da Supervisão o desenvolvimento da sua própria biblioteca de funções acessível remotamente.

Estrutura e Componentes da Aplicação de Supervisão

Os diferentes módulos funcionais que compõem a aplicação de supervisão assim como a sua mecânica interna são brevemente descritos a seguir.

Interface com o Utilizador - No ambiente tipo folha de cálculo em X-Windows e OSF/Motif o utilizador dispõe de uma tabela de células onde podem ser escritas funções de cálculo, lógicas, etc., suportadas pelo Gestor de Regras (GR). A utilização destas funções permite definir como e quando um conjunto de células relacionadas devem ser actualizadas. O GR recebe regras da interface com utilizador. Estas mensagens provocam alterações no valor das células, e por sua vez a execução de todas as regras dependentes. Estas dependências poderão causar o envio de mensagens para a interface com o utilizador contendo actualizações do valor das células alteradas, mensagens para a instalação fabril através do STM, ou ainda a actualização da base de dados usando SQL. O utilizador pode ainda configurar funções para a construção de gráficos de variáveis dependentes de um dado conjunto de células, podendo desta forma analisar graficamente a evolução dos valores das células da folha de trabalho em questão.

Folhas de Trabalho - No contexto do modelo da IF, é fornecida uma folha de trabalho por recurso físico no modelo. As folhas de trabalho contêm valores pré-definidos pelo utilizador, fórmulas ou comandos macro. Existe uma folha especial de arranque ("Startup"), que permite a definição de procedimentos de arranque, tais como carregar automaticamente todas as folhas do sistema que devem ficar activas. Uma vez carregadas, estas folhas apenas serão afixadas no écran quando a Supervisão recebe uma mensagem do modelo (figura 4.13) contendo indicação para a abertura de uma determinada folha. Se a folha seleccionada estiver activa, isto é se já foi carregada anteriormente, então será apenas afixada no écran. Se a folha seleccionada não estiver activa, poderá ser lida do disco, caso exista, ou então o utilizador terá disponível uma folha limpa para a configurar como desejar. Cada folha de trabalho dispõe de um conjunto de menus que o utilizador pode utilizar para esconder a folha afixada ou ainda para a gravar. O comando para esconder a folha, apenas a torna invisível ao utilizador, ficando todas as suas regras activas.

Gráficos - A Supervisão pode mostrar a sua informação sob a forma de gráficos de barras. Quando afixados, estes gráficos são actualizados em tempo real, de acordo com o valor actual das células a que se referem, que podem por sua vez reflectir o valor de variáveis da IF ou da BD ou de resultados de cálculos sobre essas variáveis.

Gestor de Regras - A figura 4.14 ilustra a função chave desempenhada pelo GR na aplicação de Supervisão: recebe mensagens provenientes da interface com o utilizador, da IF e de outras aplicações, processando as suas regras activas que, por sua vez, provocam a

resposta a cada um destes estímulos exteriores. Deste modo, é realizada a actualização na interface das folhas afixadas e são enviadas respostas para a IF e para as aplicações. Finalmente, existe uma primitiva de acesso às funções de RPC que poderá ser usada pelo utilizador para o desenvolvimento de funções específicas que ficam disponíveis a partir do GR. A sua arquitectura aberta transforma esta aplicação num serviço tanto para o utilizador como para qualquer aplicação que esteja registada no STM.

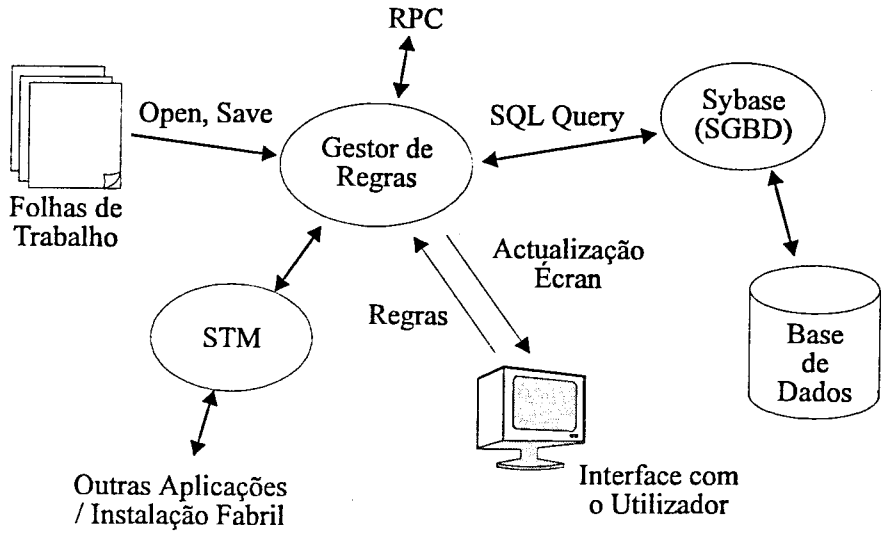


Figura 4.14 - O Gestor de Regras da Supervisão

As facilidades disponibilizadas pelo GR foram construídas sobre um conjunto de primitivas lógicas, aritméticas e de controlo de fluxo já existentes [81]. Outras funções foram definidas e desenvolvidas mais tarde para possibilitar o acesso a funções RPC, ao STM e à base de dados. O GR dispõe ainda de facilidades permitindo a execução de comandos macro na folha de cálculo.

Mais do que uma simples aplicação, e Supervisão torna-se assim uma ferramenta muito importante na monitorização e controlo de uma IF. A integração de aplicações conseguida através do STM e da base de dados comum potencia consideravelmente esta aplicação, que combina a informação vinda da base de dados, da IF e das outras aplicações num único módulo integrado de apoio à decisão.

Outras Funções do Gestor de Regras - São a seguir apresentadas algumas das funções implementadas no GR por forma a satisfazer os requisitos impostos pela Supervisão. Na tabela 4.3 apresenta-se uma breve descrição de algumas das funções do GR especificamente desenvolvidas para possibilitar a construção da aplicação de Supervisão.

Todas estas funções podem ser utilizadas isoladas ou em conjunto, de forma a permitir a definição de regras mais complexas, tais como por exemplo: a1= *if* b2*10 > 20 *then* *sendmail*("SUPER", "Supervision", "ALARMES", "Alarmes", *ack*,110,3, "s:Temperatura max excedida!",*async*). Neste exemplo, especifica-se que assim que b2*10 seja superior a 20 a regra será avaliada e a função **sendmail** executada. A execução desta regra implica que seja enviada à aplicação de gestão de alarmes (registada como: Família = ALARMES; Membro = Alarmes) a mensagem "Temperatura máxima excedida!".

Poderão assim ser configuradas regras que identificam situações irregulares activando diferentes tipos de alarmes de acordo com a gravidade do erro ou problema detectado. A utilização de um temporizador permite ainda que uma situação anormal prevaleça durante

um intervalo de tempo pré-estabelecido, sendo o alarme disparado apenas se a situação de erro persistir após se ter esgotado esse intervalo de tempo.

Tabela 4.3 - Novas Funções do Gestor de Regras

Função	Descrição
Registmailbox(), Sendmail()	Enquanto que a função **Registmailbox()** permite que o processo UNIX correspondente à aplicação de Supervisão se registe no STM com um determinado nome de membro e como pertencendo a uma dada família, a função **Sendmail** permite que o utilizador especifique o destino bem como a mensagem a ser enviada para um qualquer outro processo registado no STM.
Recepção de mensagens	Sempre que uma qualquer aplicação envia uma mensagem para a aplicação de Supervisão, os dois primeiros campos dessa mensagem devem indicar o seu destino no seio da aplicação de Supervisão, indicando respectivamente a folha e a célula destino, contendo os restantes campos a informação a enviar.
Counter()	Permite a definição de um contador numa célula cujo conteúdo pode ser incrementado de um valor definido pelo utilizador.
BarChart()	A função **BarChart** abre uma janela apresentando sob a forma de um gráfico de barras a informação contida num conjunto de células especificadas da folha. A abertura da janela contendo o gráfico pode ser feita tanto a partir de uma folha, como através de uma mensagem recebida do modelo fabril. Desta forma não é obrigatória a abertura de uma folha para que um gráfico seja afixado no écran, mas apenas que a folha em questão esteja activa.
SheetOpen(), SheetHide(), SheetSave()	As funções **SheetOpen**, **SheetHide** e **SheetSave** permitem que através de uma regra o utilizador controle a abertura e a ocultação de uma janela, assim como a sua gravação num ficheiro.
Clear(), Copy(), Move(), Rcopy()	Estas funções permitem respectivamente a limpeza, a cópia e a deslocação de células dentro de uma mesma folha ou entre folhas duas folhas de trabalho. A função **rcopy()**, permite por seu lado a cópia de uma área rectangular de células.
Initrpc(), RPC()	A função **Initrpc()** estabelece o canal de comunicação com a máquina remota para proporcionar futuras chamadas RPC. A primitiva **RPC()**, permite ao GR o acesso a funções disponíveis em servidores RPC remotos.
LookUp()	A função **LookUp()** é utilizada na pesquisa de tabelas e para conversão de constantes com base numa tabela.
split()	Dada uma sequência de caracteres com diversos campos de informação separados por um caracter conhecido, "!", por exemplo, esta função permite extrair a sequência de caracteres correspondente ao campo desejado.
Temporização	O tempo pode ser um parâmetro utilizado no disparo de regras. A célula A0 da folha de arranque ("Startup") é incrementada de segundo a segundo, e todas as regras dela dependentes são avaliadas sempre que A0 é alterada.

O Quadro de Bordo Dinâmico

Apesar das potencialidades do conceito de Supervisão apresentado, validado no decorrer do projecto SHOP-CONTROL e na primeira instalação do PROFIT, a limitação das suas facilidades gráficas bem como a sua disponibilidade apenas no ambiente UNIX levou a uma estratégia de evolução. A resolução dessas limitações obrigaria de facto a uma re-engenharia da aplicação, envolvendo considerável esforço e decisões de desenvolver /

comprar particularmente em relação às facilidades gráficas. Surgiu assim a decisão alternativa de construir uma nova aplicação distribuída em ambientes UNIX e MS-Windows que tirasse partido de informação disponível, estruturada e processada por aplicações específicas, e lhe acrescentasse um nível semântico superior. Evoluiu-se de acordo com o conceito de "Quadro de Bordo", conceito primeiro apresentado pelos Sistemas de Gestão Executiva ("Executive Information Systems"-EIS) [82] aqui "democratizado" e aplicado / aplicável a diferentes níveis de decisão no ambiente fabril.

O projecto REAL-I-CIM veio permitir a reformulação da aplicação de Supervisão no denominado "Quadro de Bordo Dinâmico"(QBD), que deverá fornecer aos diversos níveis de gestão da instalação fabril informação estruturada de forma a assistir ao óptimo desempenho das suas funções. Em termos gerais, esta aplicação deverá fornecer ao seu utilizador informação de elevado valor semântico organizada sob a forma de gráficos e tabelas representativos do processo de produção em curso.

A definição de requisitos gerais do QBD é relativamente simples: "O utilizador deverá dispor de gráficos e tabelas apresentando informação representativa do processo em execução que lhe facilite a tomada de decisões". Com base nesta definição, é possível enumerar múltiplos indicadores de desempenho habitualmente utilizados no apoio à decisão no fabrico. No entanto, e apesar de grandes semelhanças entre indicadores utilizados em vários tipos de indústrias, surgem sempre novas necessidades para cada caso em estudo.

Face à evidência da impossibilidade de prever todas as situações surgiu como fundamental o estabelecimento de um quadro geral onde o trabalho de construção de soluções específicas pudesse ser desenvolvido, contemplando simultaneamente os aspectos semânticos e tecnológicos [6]. No nível semântico foi decidido organizar conceptualmente a informação necessária às diferentes tarefas de gestão de acordo com um determinado horizonte temporal. A tabela 4.4 ilustra a estrutura de informação do modelo sistema integrado de fabrico usando a grelha de informação GRAI [83]. Esta tabela apresenta a informação tratada nas diferentes vertentes da gestão operacional (produção, qualidade e manutenção) e da gestão de recursos (materiais neste caso, embora o pessoal e o equipamento também pudessem ser incluídos).

A grelha GRAI será utilizada no enquadramento da definição de requisitos do QBD. A grelha de informação (tabela 4.4) está relacionada com a a grelha de decisão (tabela 4.5) onde se ilustra a estrutura organizacional. Tendo em atenção as tabelas 4.4 e 4.5 [83], e atendendo a que o RIC se situa nas áreas do planeamento de curto prazo e da gestão das operações de fabrico, o esforço de caracterização dos requisitos do QBD centrar-se-á nos centros de decisão mais próximos da instalação fabril, e nos horizontes temporais mais curtos (tendo como limite o horizonte de 1 a 6 semanas).

O QBD vai fornecer informação de índole geral, assim como indicadores de desempenho baseados em informação pertencendo às várias aplicações tradicionais da gestão fabril, i.e. gestão da produção, da qualidade, da manutenção e de materiais.

O QBD é, do ponto de vista tecnológico, uma aplicação cliente que deverá ser capaz de aceder à informação disponibilizada pelos diferentes servidores distribuídos pela instalação fabril, e cuja organização foi já apresentada no ponto 4.4.3. No desenvolvimento desta aplicação não será utilizado software desenvolvido no decorrer do projecto SHOP-CONTROL. Serão usadas ferramentas standard entretanto disponíveis no mercado com facilidades de configuração, de acesso a bases de dados ou a servidores genéricos remotos, que deverão permitir construir software para ambiente X-Windows / OSF/Motif e MS-Windows.

Tabela 4.4 - Grelha GRAI de informação, Estrutura da Informação no Sistema de Fabrico

Legenda: Em itálico, são indicados exemplos de possíveis indicadores de desempenho.

	Horizonte Temporal	Gestão da Produção	Gestão da Qualidade	Gestão da Manutenção	Gestão de Materiais
Empresa / Fábrica	4 anos 1 ano	plano estratégico; qt/família prod.; data de início & fim família			
Instalação Fabril	1 ano 1 mês	plano director; qt./ fam.prod./ mês *(tempo de resposta do produto, "lead time")*	plano director da qualidade para produtos e equipamento	plano director da manutenção; procedimentos de manutenção & disponibilidade do esforço / posto de trabalho *(tempo médio entre avarias por equipamento e regime de operação)*	capacidade do vendedor; qt./componente/ mês; ordens para as peças comuns; qt./ componente/ mês
	6 semanas 1 semana	Plano director de produção; qt./ tipo prod./ semana *(tempo de resposta do produto, "lead time"; tempo de processamento do produto, "cycle time"; fiabilidade de entregas JIT)*	informação da qualidade; nível da qualidade/ tipo prod/ semana; nível da qualidade/ posto de trabalho/ semana *(custos de sucata, e % de rejeitados, por produto e por posto de trabalho)*	plano de manutenção; ordens de manutenção; assistência externa; qt./componente/ semana	ordens de curto prazo; subcontratação externa; qt./componente/ semana *(nível de stocks por produto, componente ou matéria prima; número e gravidade de roturas de stock)*
	3 dias 1 dia	envio ordens de trabalho; qt./ montagem; data de entrega *(tempo de processamento do produto, "cycle time")*	análise detalhada do nível da qualidade / tipo prod/ dia; nível da qualidade/ posto de trabalho/ dia	emissão de ordens de manutenção; ordens de manut./ posto de trabalho / turno	peças reservadas & subcontratação; qt/componente/ semana
Célula de Trabalho	3 turnos 1 hora	ordens de fabrico sequenciadas	dados da qualidade por posto de trabalho; nível da qualidade / posto de trabalho *(% de não-conformidades por produto e por turno)*	estado do equipamento; acções de manutenção; ordens de manutenção / posto de trabalho	qt./peça

Os requisitos funcionais serão construídos usando o quadro geral descrito, não sendo portanto o objectivo atingir a completa especificação do QBD. Durante o projecto RIC, pretende-se estabelecer uma biblioteca de padrões ("templates") de gráficos, tabelas e indicadores de desempenho. Numa fase inicial, esta escolha baseou-se nos requisitos das instalações piloto, em bibliografia da especialidade e na experiência dos parceiros do projecto. A referida biblioteca deverá assim evoluir para cada um dos centros de decisão da tabela 4.5 com a introdução sucessiva de novos padrões.

Tabela 4.5 - Grelha GRAI de Decisão, Estrutura de Organização do Sistema de Fabrico

	Horizonte Temporal	Gestão da Produção	Gestão da Qualidade	Gestão da Manutenção	Gestão de Materiais
Empresa / Fábrica	4 anos 1 ano	identificação dos objectivos do negócio			
Instalação Fabril	1 ano 1 mês	estudo das necessidades do mercado & dos recursos de produção	determinação & análise dos procedimentos da qualidade	identificação dos procedimentos de manutenção	definição das capacidades do vendedor & "procurement" de peças comuns
	6 semanas 1 semana	estabelecimento do plano director da produção	recolha & análise de informação da qualidade	análise do desempenho das máquinas e do plano de manutenção	encomenda de curto prazo de componentes & subcontratação externa
	3 dias 1 dia	emissão das ordens fabrico	verificação & análise de amostras	emissão de ordens de manutenção	reserva de peças & subcontratação
Célula de trabalho	3 turnos 1 hora	sequenciamento das ordens de produção	medida & verificação da qualidade do produto	diagnóstico de avarias nas máquinas, e acções de reparação	selecção de material dos armazéns

4.4.7 O Acesso à Instalação Fabril

O problema da recolha e envio de dados, de e para a instalação fabril, será tratado sucintamente nos parágrafos que se seguem. Neste contexto, vamos concentrar a nossa atenção nas estratégias utilizadas no âmbito do projecto SHOP-CONTROL, na interacção com a instalação fabril usando terminais de baixo custo e com reduzida capacidade de processamento local.

Gestor de Transacções Fabris

O Gestor de Transacções Fabris (GTF) constituiu no SHOP-CONTROL o elo de ligação à instalação fabril [84]. O GTF interage com as aplicações e com os terminais de recolha de dados na instalação fabril através de mensagens (figuras 4.6 e 4.8 páginas 81 e 83). A interacção com terminais fabris exige quase sempre software dedicado pelo facto dos sistemas de recolha de dados e redes industriais de comunicação serem proprietários A estrutura do GTF é no entanto suficientemente flexível para permitir, através da alteração do módulo de acesso ao "driver" (figura 4.15), a interligação a novos sistemas de recolha de dados e de controlo da produção, garantindo deste modo a desejável abertura e independência do fabricante do equipamento.

Além desta função de interface no nível mais baixo dos protocolos de comunicação, o GTF tem por objectivo a transferência de informação entre os terminais fabris e as aplicações de gestão fabril residentes na máquina UNIX. Deste modo, o GTF é responsável pelo correcto encaminhamento das mensagens, quer elas fluam no sentido IF->Aplicação quer no sentido inverso. O GTF é ainda responsável pela actualização da base de dados assegurando que ela espelhe em cada instante o estado actual da instalação fabril. Além disso, a utilização do GTF no controlo de terminais fabris com reduzida capacidade de processamento local, permite que ele execute funções tais como por exemplo a gestão dos diálogos com o operador ou a validação de "passwords".

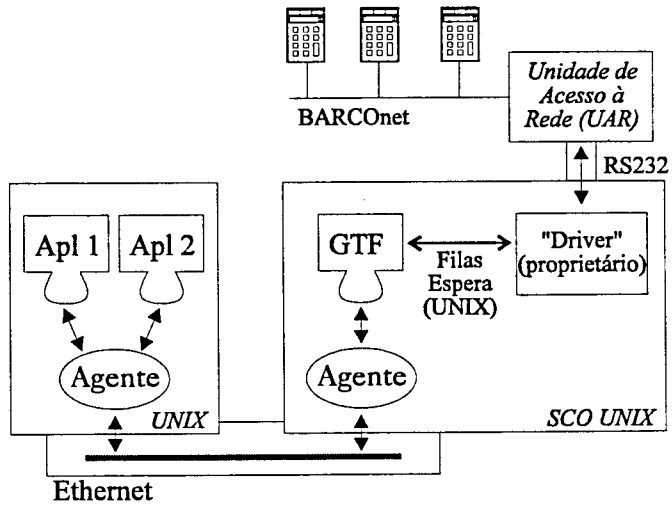


Figura 4.15 - Gestor de Transacções Fabris no interface com a infra-estrutura de recolha e de comunicação de dados do SHOP-CONTROL (arquitectura BARCO Automation)

O Módulo de Acesso Distribuído à Instalação Fabril

Objectivos

Na sequência da experiência de utilização do GTF, e com o objectivo de atingir uma maior descentralização no controlo da instalação fabril, foi desenvolvimento um novo módulo designado por "Módulo de Acesso Distribuído à Instalação Fabril" (MADIF) [4]. Este módulo tem por objectivo deslocar para a instalação fabril algumas das tarefas de processamento de informação e de controlo realizadas no SHOP-CONTROL ao nível do modelo fabril ou das aplicações a correr na estação de trabalho central (figura 4.16). Neste contexto, todas as tarefas que possam ser configuradas de uma só vez no início de cada instalação, poderão ser deslocadas para o MADIF, com vista à mais simples configuração do sistema. A utilização deste módulo apenas será relevante caso o sistema seja suportado por terminais fabris com capacidade de processamento limitado, impondo portanto restrições à distribuição das funcionalidades por diferentes máquinas sobre uma rede, o que acontece todavia na maioria dos casos. Este módulo vai permitir que naturalmente se estabeleça uma estrutura de decisão hierárquica, sendo o sistema central aliviado das decisões de menor importância que serão tomadas localmente e de uma forma distribuída ao nível dos terminais industriais (T1, T2, etc.). Repare-se que, no caso anterior, o GTF se comportava como um concentrador de terminais com tarefas de validação e encaminhamento de mensagens, sendo o processamento local muito reduzido. O MADIF pode contudo utilizar-se não só como concentrador de terminais, mas também como "front-end" de aplicações e como interface entre redes de protocolos diferentes.

O MADIF constitui uma camada suplementar de controlo sobre a UAR (Unidade de Acesso à Rede industrial), fornecendo às camadas superiores um serviço de controlo da IF conduzido por eventos (figuras 4.15 e 4.16) provenientes tanto da IF como das aplicações. Para tratar diferentes tipos de eventos, durante a instalação do sistema o utilizador escreve um conjunto de "scripts" especificando as acções a serem executadas aquando da ocorrência de cada um destes eventos. Estes "scripts" permitem a configuração de um comportamento pré-definido para este sistema de tratamento de eventos, cuja execução é suportada a um baixo nível na hierarquia de decisão do sistema, i.e. junto do processo / instalação fabril e sem intervenção do operador. Estas facilidades são particularmente interessantes para

resolver situações críticas ou de excepção em que é desejável reagir rápido e evitar que o sistema no nível superior seja inundado com grandes quantidades de informação.

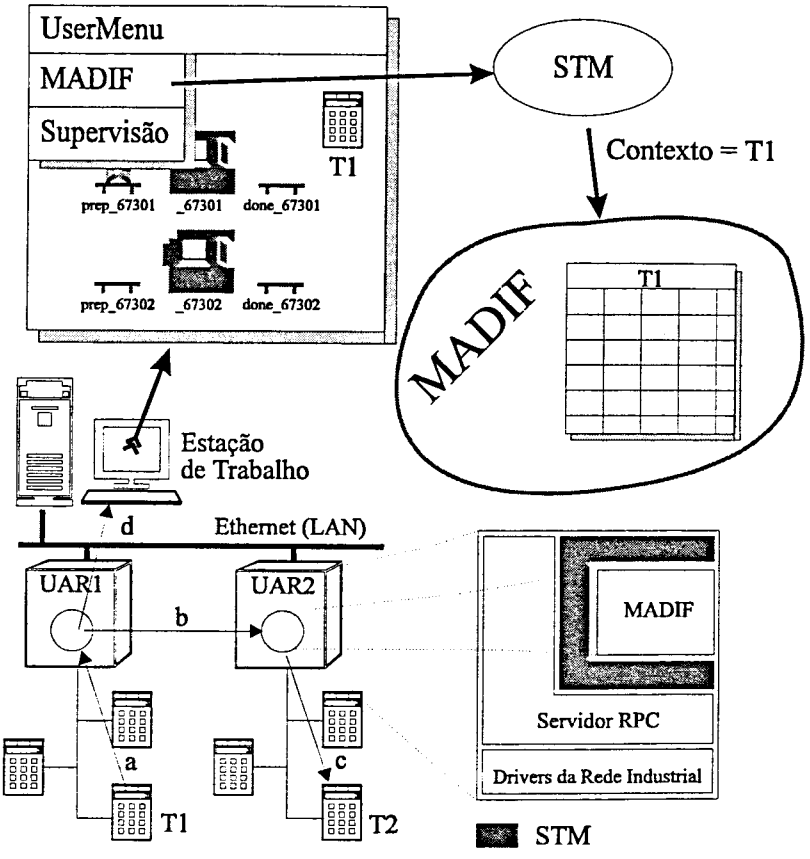


Figura 4.16 - Módulo de Acesso Distribuído à Instalação Fabril

Conceitos e Modo de Funcionamento

O modo de funcionamento do MADIF é em tudo semelhante ao da aplicação de Supervisão anteriormente descrito. Assim sendo, esta aplicação utiliza também o gestor de regras da Supervisão e o conceito de folha de trabalho à semelhança de uma comum folha de cálculo. Tal como ilustra a figura 4.16, num sistema distribuído podem coexistir várias instâncias do MADIF para suportar o acesso a diferentes redes proprietárias de terminais fabris com reduzida capacidade de processamento local.

O conceito de folha de trabalho é nesta aplicação usado como elemento estruturante. Deste modo, atribui-se a cada terminal fabril uma folha de trabalho contendo um conjunto de regras e "scripts" responsáveis pela sua gestão, bem como pela sua comunicação com outros módulos ao mesmo nível (b na figura 4.16) ou a um nível hierarquicamente superior (d na figura 4.16). Este desenvolvimento poderá ser facilitado pela transmissão de mensagens em modo "multicast" de forma a, por exemplo, ser mantida a coerência entre variáveis dependentes associadas aos diversos módulos distribuídos do MADIF. Como ilustra a figura 4.16, a selecção do ícone correspondente ao terminal T1 no modelo fabril seguida da invocação do MADIF permite o acesso à folha de configuração daquele terminal. Para cada folha de trabalho no MADIF, i.e. para cada terminal fabril, o utilizador configura conjuntos de regras e "scripts" que podem não só tratar as situações de excepção como ainda informar o utilizador ou as aplicações sempre que sejam tomadas decisões

relevantes. A reacção a situações críticas exigindo a intervenção do operador é antecipadamente configurável, podendo mesmo estabelecer-se diálogos entre este módulo e a interface com o utilizador.

Arquitectura

As aplicações do SHOP-CONTROL correm num ambiente exclusivamente UNIX, com as estações de trabalho ligadas através de uma rede Ethernet às UAR que estabelecem a interface com a(s) rede(s) industrial(ais) de recolha de dados e os terminais fabris. As UAR suportam assim o acesso à instalação fabril, constituindo o local ideal para situar o MADIF. Este módulo comunica por sua vez com um processo servidor que encapsula o "driver" de acesso à rede industrial (figura 4.17). A ligação ao STM permite a ligação com as aplicações do SHOP-CONTROL, assim como a outros módulos do MADIF residentes em outras UAR ligadas à mesma rede local (figura 4.16). A comunicação entre os diversos módulos residentes nas UAR fornece uma infra-estrutura que permite o fluxo de informação ao nível da IF, facilitando a comunicação transparente entre dois ou mais terminais fabris que poderão ser de diferentes fabricantes (tal como ilustra a figura 4.16, com o envio de mensagens de T1 para T2 via UAR1 e UAR2: a-b-c). O MADIF pode ser configurado directamente pelo operador da IF, ou ainda remotamente através de mensagens veiculadas pelo STM caso não exista terminal local.

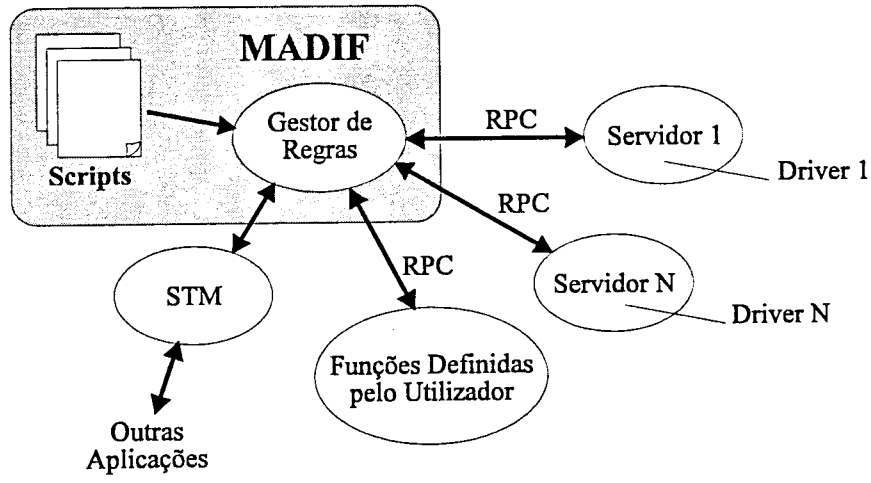


Figura 4.17 - Um Módulo do MADIF

Como ilustra a figura 4.17, é grande a flexibilidade dada ao utilizador, sendo um único módulo do MADIF capaz de fazer interface com múltiplas redes industriais através dos seus Drivers¡. Esta interface é realizada através de um servidor RPC para cada "driver", podendo o MADIF estar ligado a tantos servidores quantos os necessários. Por outro lado, estas facilidades de RPC podem ainda ser utilizadas para a construção de funções definidas pelo utilizador.

Usando o Gestor de Regras no MADIF

Os módulos do MADIF são baseados no gestor de regras (GR) anteriormente descrito, podendo o utilizador configurar a transferência de dados da IF, das aplicações e da base de dados. A chegada de informação ao GR dispara um conjunto de regras que irão estabelecer qual o "script" a ser executado. Os "scripts" são escritos numa linguagem semelhante à linguagem "C", suportando primitivas aritméticas, lógicas e de controlo de fluxo [81].

No ambiente SHOP-CONTROL, qualquer acesso à informação fabril é feito através do modelo da instalação fabril. Não sendo o MADIF uma excepção, o acesso à configuração

de um terminal faz-se seleccionando o terminal desejado e invocando esta aplicação. Neste caso abre-se uma folha de trabalho semelhante a uma folha de cálculo, que permite a configuração do MADIF (figura 4.16). Na tabela 4.6 é ilustrada uma situação típica de "run-time": no decorrer da sondagem ("polling") à rede industrial chega informação do terminal T2 da instalação fabril; esta informação é então encaminhada para a folha de trabalho "Terminal2", a qual é obtida seleccionando o objecto com o mesmo nome no modelo (que corresponde naturalmente ao objecto real na instalação fabril).

Tabela 4.6 - Tratamento da Informação da Instalação Fabril no MADIF

	Objectivo	**Acção**
1	Sondagem à fila de espera dos terminais	Na folha de trabalho "Dispatcher", é inserida a regra: B0= if Startup.A0 then RPC ('scontrol', 't_read', '', 'Dispatcher.A0') Esta regra executa a função RPC sempre que a célula Startup.A0, i.e. o temporizador, é alterada. O resultado da sondagem é colocado na célula Dispatcher.A0 com o formato: !<nº terminal>!<código>![<código>!]
2	Detecção do terminal de onde chega a informação	Na folha "Dispatcher" é colocada a fórmula: A1= Split(A0,'!',1,A2), execução da qual devolve o elemento que se segue ao primeiro '!', neste caso o número do terminal, colocando o resultado em A2.
3	Encaminhamento da informação para o terminal (folha) correspondente	Sempre que a célula "Dispatcher.A2" é actualizada, o conteúdo de A0 é deslocado para a célula A0 de uma folha que resulta da concatenação (..) de "Terminal" com A2 ('Terminal'..A2..'.A0'). Se A2=2, o conteúdo de A0 é deslocado para "Terminal2.A0"
4	Chegada da informação à folha ***Terminal$_i$***, (neste caso, o Terminal2)	Folha ***Terminal$_i$***: A informação chega a esta folha vinda do "Dispatcher". Na folha ***Terminal$_i$*** é então tratada toda a interacção referente ao terminal, tal como: execução de um diálogo e envio do resultado do mesmo a uma aplicação, o Simple++ por exemplo; registo do valor de uma variável, uma temperatura, assim como o seu controlo; envio de uma mensagem de alarme numa dada condição; etc. A configuração da folha faz-se através de regras e da execução de "scripts" associados a regras.

Conclusões

Como vemos, a flexibilidade deste módulo é muito grande permitindo uma reconfiguração simples da sua interacção tanto com a instalação fabril, através de qualquer um dos seus "drivers", como com qualquer uma das aplicações do SHOP-CONTROL.

A utilização do MADIF como evolução natural do GTF foi todavia suspensa na sequência da definição da arquitectura do PROFIT, pelo facto de terem sido escolhidos para as primeiras instalações terminais fabris baseados em PCs com razoável capacidade de processamento local. Neste contexto, a existência do MADIF não faz de facto qualquer sentido. O seu aproveitamento será decerto equacionado sempre que for necessária a

utilização de terminais de baixo custo com reduzida capacidade de processamento local, do tipo dos utilizados no projecto SHOP-CONTROL.

Monitorização da Condição do Processo de Fabrico ("Process Condition Monitorizing")

Através do GTF e / ou do MADIF podem ser implementadas aplicações simples de monitorização de condição ao nível da IF. Basta para tal configurar gamas onde se devem encontrar os valores das variáveis críticas e definir as acções correctivas respectivas [4]. Este processo de controlo pode ser estruturado recorrendo ao conhecimento heurístico de um operador experiente da IF. Estas facilidades, juntamente com funcionalidades de análise estatística podem certamente contribuir para melhorar os mecanismos de decisão, por exemplo na gestão da qualidade ou da manutenção.

A Aplicação de Monitorização de Condição no REAL-I-CIM

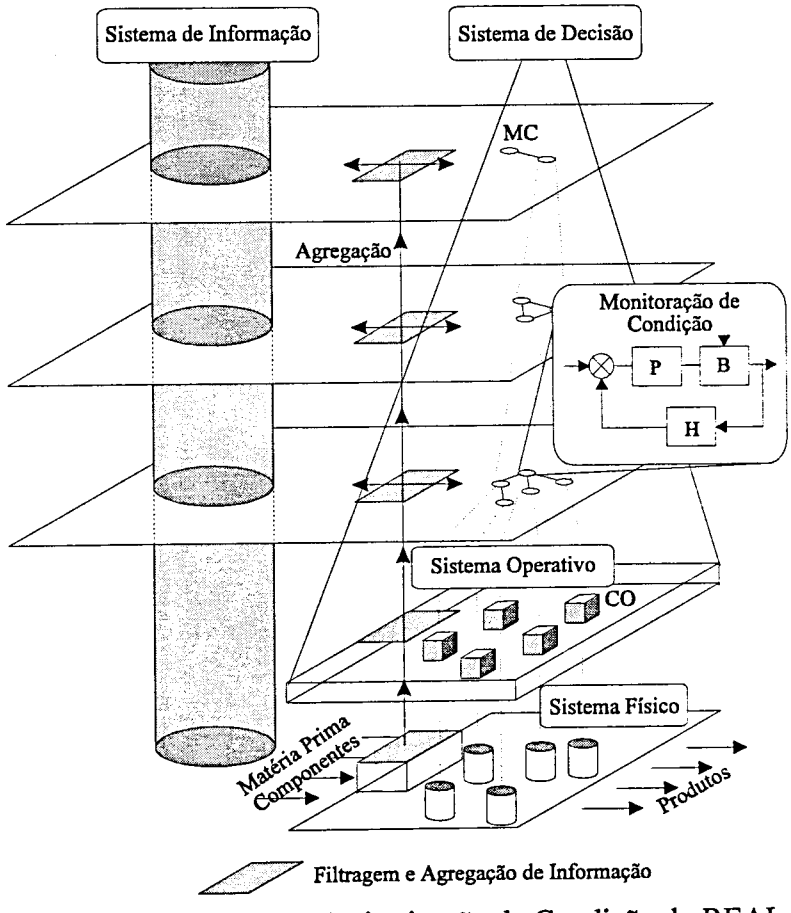


Figura 4.18 - Os Módulos de Monitorização de Condição do REAL-I-CIM

No projecto REAL-I-CIM, a aplicação do conceito de Monitorização de Condição do Processo de Fabrico surge como uma evolução e extensão das facilidades básicas propostas pelo MADIF [85]. A nova aplicação propõe agora a aplicação de conceitos da teoria do controlo na construção de malhas locais de monitorização / correcção a diferentes níveis do processo de fabrico [86]. A figura 4.18 ilustra a forma como os módulos de monitorização de condição se inserem num modelo hierárquico mas distribuído de apoio à decisão. Cada um dos módulos monitoriza e controla uma ou um pequeno número de variáveis do sistema, tais como por exemplo o nível de um stock do em-curso-de-fabrico ou o valor de um parâmetro de qualidade. O bloco H utiliza um ou mais valores de saída para derivar os valores actuais dos indicadores de desempenho a controlar, sendo estes indicadores

derivados utilizados na realimentação pelo processo de controlo P. O processo de fabrico B por seu lado, é sujeito a perturbações, tais como por exemplo avarias de máquinas, problemas de qualidade, etc..

Em cada nível de decisão, os vários módulos armazenam a informação tratada na base de dados no decorrer do seu normal funcionamento. Esta informação será usada pelas malhas de decisão dos níveis superiores, com horizontes temporais cada vez mais dilatados. O QBD atrás apresentado (em 4.4.6) faz evidentemente uso desta informação que pode ser apresentada ao utilizador de uma forma integrada e adequadamente tratada para apoio à decisão.

4.4.8 Infra-estrutura Informática

Infra-estrutura Informática do SHOP-CONTROL / PROFIT

A infra-estrutura informática do SHOP-CONTROL assentou em diversos standards *de facto* ou *de jure* actualmente existentes. Os objectivos do projecto SHOP-CONTROL incluíam o desenvolvimento de soluções de baixo custo. Por este motivo, o factor económico foi sempre levado em consideração na procura de compromissos que se reflectiram na selecção do "hardware" e do "software" a ser utilizado, especialmente no que diz respeito aos terminais fabris.

Hardware - No decorrer do projecto SHOP-CONTROL, por disponibilidade nas instalações dos diversos parceiros do consórcio, foram seleccionadas como standard para o desenvolvimento as plataformas HP 9000 nas suas diversas versões. Após a conclusão do projecto, nomeadamente com a construção do PROFIT, foi realizada a migração do software para outras plataformas, tais como a DEC (5200), IBM (RISC 6000) e PCs suportando SCO UNIX. Foram também ligados terminais fabris baseados em PCs. As comunicações entre as diversas plataformas fazem-se sobre Ethernet TCP/IP.

Software - O sistema operativo escolhido foi o UNIX pela sua disponibilidade nas diversas plataformas, inclusive em máquinas de baixo custo (PCs com SCO UNIX), e pelas inerentes facilidades de multi-processamento. O Sistema de Transferência de Mensagens foi construído sobre "stream sockets" UNIX sendo suportado em diversas plataformas (tais como por exemplo HP-UX, IBM RISC 6000, Silicon Graphics, SUN, PCs com MS-Windows e MS-DOS) [68]. O X-Windows / OSF/Motif foi o standard escolhido para interface com o utilizador, sendo suportado na maioria das plataformas de hardware, incluindo SUN e NeXT desenvolvido por terceiros. As linguagens de programação usadas foram o "C" e o "C++", tendo sido utilizados compiladores ANSI por forma a garantir uma maior compatibilidade na migração de software. Do ponto de vista da base de dados, é suportada a ligação ao Sybase (base de dados relacional central), acedida a partir de clientes UNIX ou de clientes em PCs/Windows usando ODBC.

Infra-estrutura Informática do PROFIT / REAL-I-CIM

Hardware- Suporte das plataformas HP, DEC, IBM suportando UNIX, e PCs suportando SCO UNIX, MS-DOS e MS-Windows.

Software - O software PROFIT/RIC deverá executar sobre os sistemas operativos UNIX e SCO UNIX, dadas as inerentes facilidades de multi-processamento. Por outro lado, a vulgarização e o baixo custo dos PCs justificam que algumas das aplicações do sistema também possam ser executadas sobre MS-DOS e MS-Windows, nomeadamente aquelas cuja utilização de forma distribuída ao nível da instalação fabril é uma necessidade

crescente. As linguagens de programação usadas são o "C" e o "C++", estando a ser utilizados compiladores ANSI por forma a garantir uma maior compatibilidade na migração de software.

4.5 Conclusão

Concluiu-se assim a abordagem iniciada no Capítulo 2 com a descrição do ambiente fabril, e onde se procurou dar uma panorâmica dos problemas envolvidos na sua gestão. Seguiu-se o Capítulo 3 com a apresentação das arquitecturas de referência mais significativas da actualidade, tendo sido salientada a forma como cada uma delas aborda os problemas da gestão fabril integrada. Neste capítulo foram ainda apresentadas as soluções mais significativas para as infra-estruturas de suporte aos sistemas de gestão fabril integrada.

Neste quarto capítulo, foi intenção do autor apresentar e discutir algumas das soluções concretas existentes actualmente no âmbito das infra-estruturas de integração empresarial. Não procurou fazer-se o seu levantamento exaustivo, motivo pelo qual nos concentramos na CIMOSA e nas suas implementações mais relevantes. Certamente que ficaram de fora outras propostas também interessantes, mas o facto de não partilharem com a CIMOSA os actuais esforços de normalização levou a que fossem preteridas. É excepção a esta regra a apresentação da implementação SHOP-CONTROL / PROFIT cujos desenvolvimentos tiveram desde sempre uma participação activa do autor.

A apresentação inicial das propriedades que devem caracterizar uma infra-estrutura de integração estabeleceu a base para a descrição dos sistemas que se seguiram. Foi assim introduzida a infra-estrutura de integração da CIMOSA, da maior importância tendo em consideração os diversos grupos de estandardização em que tem estado envolvida. Seguiram-se apresentações resumidas dos ambientes proporcionados pelo Ωmega e SEW-OSA, bem como do protótipo da infra-estrutura da CIMOSA desenvolvida no WZL. Qualquer um destes sistemas, ainda em fase de protótipo, apresenta infra-estruturas de integração e de modelação com reconhecidas potencialidades, nomeadamente no que se refere à sua conformidade com a CIMOSA. Resta no entanto provar a sua eficácia e eficiência em ambiente industrial, nomeadamente no que se refere à interface do sistema integrado com o utilizador. Ainda no contexto das infra-estruturas de suporte a sistemas de gestão de operações fabris, foram apresentados os resultados do projecto SHOP-CONTROL e o produto comercial PROFIT. O envolvimento do autor desde o início dos trabalhos de I&D (em 1990) que levaram ao nascimento deste produto (em 1994) permitiu realizar uma apresentação com uma perspectiva histórica, onde é possível a apreciar o processo evolutivo das soluções que foram sendo adoptadas. Neste capítulo foi ainda posta em evidência a actualidade dos conceitos de modelação fabril e de ciclo de vida do modelo fabril, bem como o seu suporte pela infra-estrutura do PROFIT. Finalmente, foram apresentadas algumas das aplicações do SHOP-CONTROL, PROFIT e REAL-I-CIM, cuja selecção se prendeu com o carácter de infra-estrutura integradora que lhes é inerente, e por terem sido resultado de um envolvimento directo do autor, tanto nas fases de projecto conceptual como nas de desenvolvimento, implementação e teste.

Capítulo 5

SUPORTE À GESTÃO DE OPERAÇÕES FABRIS ATRAVÉS DE MODELOS EXECUTÁVEIS

5.1 Introdução

Neste capítulo o autor apresenta e discute a utilização de Modelos Executáveis no suporte à gestão de operações fabris no âmbito dos Sistemas Integrados de Fabrico. A apresentação do ciclo de vida do sistema integrado de fabrico é, numa primeira parte do capítulo, realizada através da introdução das interfaces que caracterizam a Arquitectura do Projecto de Integração do modelo de referência proposto pela Universidade de Purdue (PERA). Esta apresentação, vai ainda permitir identificar as tarefas, as pessoas, e a documentação envolvidas num projecto de integração de um sistema de fabrico.

De seguida serão tratados alguns dos princípios básicos da Modelação para Simulação e Controlo das Operações. O problema será primeiramente abordado do ponto de vista genérico, com a definição dos objectivos da simulação e da modelação, seguindo-se o tratamento do problema específico da Modelação para Simulação e Controlo das Operações no Sistema de Fabrico. Pela sua importância no contexto deste trabalho, a modelação orientada ao objecto será alvo de um tratamento um pouco mais detalhado.

Uma vez tratado o tema da modelação para a simulação e controlo das operações no sistema de fabrico, que é na arquitectura de Purdue enquadrado no contexto da Arquitectura do Equipamento de Fabrico, será tratado o problema da modelação e simulação no âmbito da Arquitectura do Sistema de Informação. Por estar fora do âmbito deste trabalho, não será aqui tratada a modelação da terceira vertente da PERA: a Arquitectura Humana & Organizacional.

Finalmente serão abordados e discutidos os modelos executáveis do CIMOSA e do PROFIT. O primeiro pela relevância já justificada atrás, e pelo facto de ter sido no âmbito do consórcio AMICE que pela primeira vez foi tratado o problema dos modelos

executáveis. Em conclusão será realizada a apresentação do modelo executável do PROFIT, assim como a forma como ele é utilizado. Ao jeito de síntese é realizada, utilizando a matriz de avaliação GERAM, uma análise das ferramentas de integração fabril disponibilizadas pelo PROFIT.

5.2 O Ciclo de Vida dos Sistemas Integrados de Fabrico

O desenvolvimento harmonioso de projectos / actividades de integração empresarial envolve, como já foi referido nos capítulos iniciais desta tese, não só o sistema de informação mas também, e com igual importância, o equipamento de fabrico e a vertente humana e organizacional da empresa. A arquitectura PERA inclui estes três aspectos complementares através da implementação física das três sub-arquitecturas, a "Arquitectura do Sistema de Informação" (ASI), a "Arquitectura do Equipamento de Fabrico" (AEF) e a "Arquitectura Humana e Organizacional" (AH&O).

5.2.1 As Interfaces na Arquitectura de Purdue

A existência das três arquitecturas acima referidas implica a necessidade de interfaces físicas bem definidas, tanto no interior de cada uma das arquitecturas como entre cada uma delas. Na arquitectura de Purdue, podemos como vimos caracterizar dois tipos de interfaces: as interfaces que permitem a comunicação entre dois elementos / componentes do sistema integrado na empresa (Tipo 1), e as interfaces existentes entre duas fases do projecto de integração (Tipo 2).

Interfaces Tipo 1

As interfaces tipo 1 (tabela 5.1) podem ser divididas em duas categorias. Na primeira categoria, temos aquelas que *transferem a informação* entre os diferentes componentes da Arquitectura do Sistema de Informação e / ou entre estes componentes e as outras arquitecturas, sendo classificadas como desempenhando uma função de comunicação. Todas as interfaces de comunicação obrigam a uma compreensão / interpretação da informação transferida, bem como a sua eventual transformação, assim como à observância de um determinado protocolo de interacção. Na segunda categoria, incluem-se as *transferências de materiais ou de energia* entre os elementos da arquitectura de fabrico (recursos de fabrico, produtos).

Tabela 5.1 - Interfaces Tipo 1 na Arquitectura de Purdue

Interfaces	Descrição
Internas à AEF	Contempla todas as interfaces que permitem o fluxo de materiais, de energia ou de ambos, através das fronteiras entre unidades individuais do equipamento de fabrico.
Entre a AEF e ASI	A este nível temos soluções para problemas de interligação (usando cablagens veiculando sinais eléctricos) entre os equipamentos das duas arquitecturas. As cablagens ligam directamente os sensores, actuadores e controladores pertencentes à ASI aos elementos com os quais interagem directamente na AEF (tais como válvulas de controlo, motores, etc.).

(Continua)

Entre AEF e AH&O	A presença humana na AEF envolve sempre o operador, ou actuando como se fosse mais um componente do sistema de fabrico (trabalho directo) ou usando algum dispositivo para o tratamento indirecto dos materiais no processo de produção. Os denominados Factores Ergonómicos da interface Homem-Máquina são considerados como fazendo parte da AEF.
Internas à AH&O	No interior desta arquitectura só existe um tipo de interface, a interface humana. Esta interface está relacionada com um conjunto de normas para a representação e comunicação da informação. Neste contexto, a existência de um dicionário ou de outros standards semânticos, assim como de uma semântica gráfica e uma simbologia, podem tornar-se imprescindíveis num ambiente em que pessoas com diferente formação escolar e académica, e com diferentes funções e níveis de responsabilidade, se envolvem em interacções frequentemente informais.
Entre ASI e AH&O	A interacção entre a ASI e o operador / utilizador envolve sempre a transferência de dados ou de informação através de uma interface que permite a interpretação da informação transferida nos dois sentidos. A denominada interface Homem-Máquina é considerada como fazendo parte da ASI.
Internas à ASI	As interfaces internas à ASI são indispensáveis para interligar os componentes que executam as tarefas de recolha, armazenamento, processamento e distribuição da informação. Os elementos envolvidos no sistema de informação são sempre computadores ou outros dispositivos que, de ambos os lados da interface, sejam capazes de interpretar correctamente os resultados das operações executadas no outro lado dessa mesma interface. Estas interfaces devem pois obedecer a um conjunto de requisitos do ponto de vista dos protocolos de comunicação envolvidos, sendo especificados aspectos funcionais, eléctricos e mecânicos. Por outro lado, ambos os componentes envolvidos na troca de informação deverão utilizar uma sintaxe e uma semântica pré-acordadas que permitam a correcta síntese e interpretação da informação trocada.

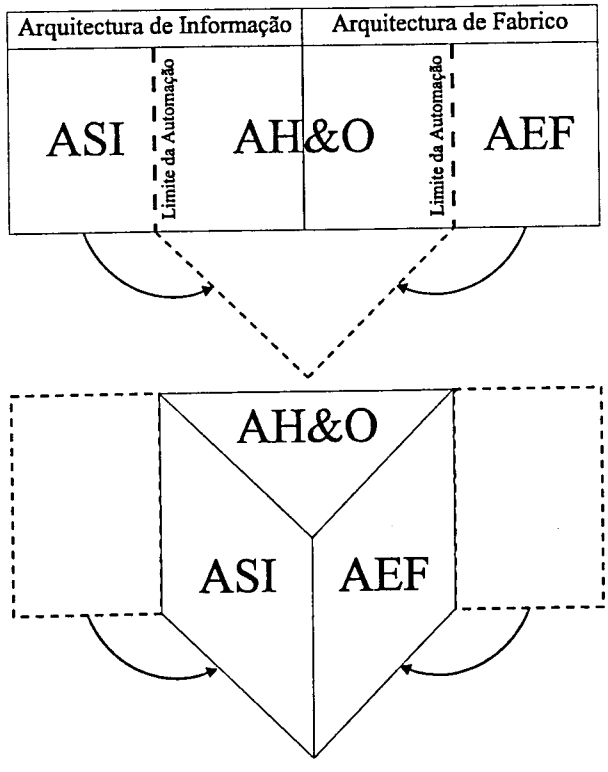


Figura 5.1 - Representação das Interfaces entre as Arquitecturas de Implementação (Tipo 1) na Arquitectura de Projecto de Purdue

A figura 5.1 ilustra a representação das interfaces tipo 1 na arquitectura de Purdue. Quebrando a representação das arquitecturas no modelo convencional da arquitectura PERA pela linha do Limite da Automação, e formando um prisma, tornam-se bem visíveis as interfaces descritas na tabela 5.1. Será ainda de observar que as interfaces são necessárias entre arquitecturas de implementação, uma vez que as interfaces são entidades físicas apenas dependentes de detalhes de implementação.

Interfaces Tipo 2

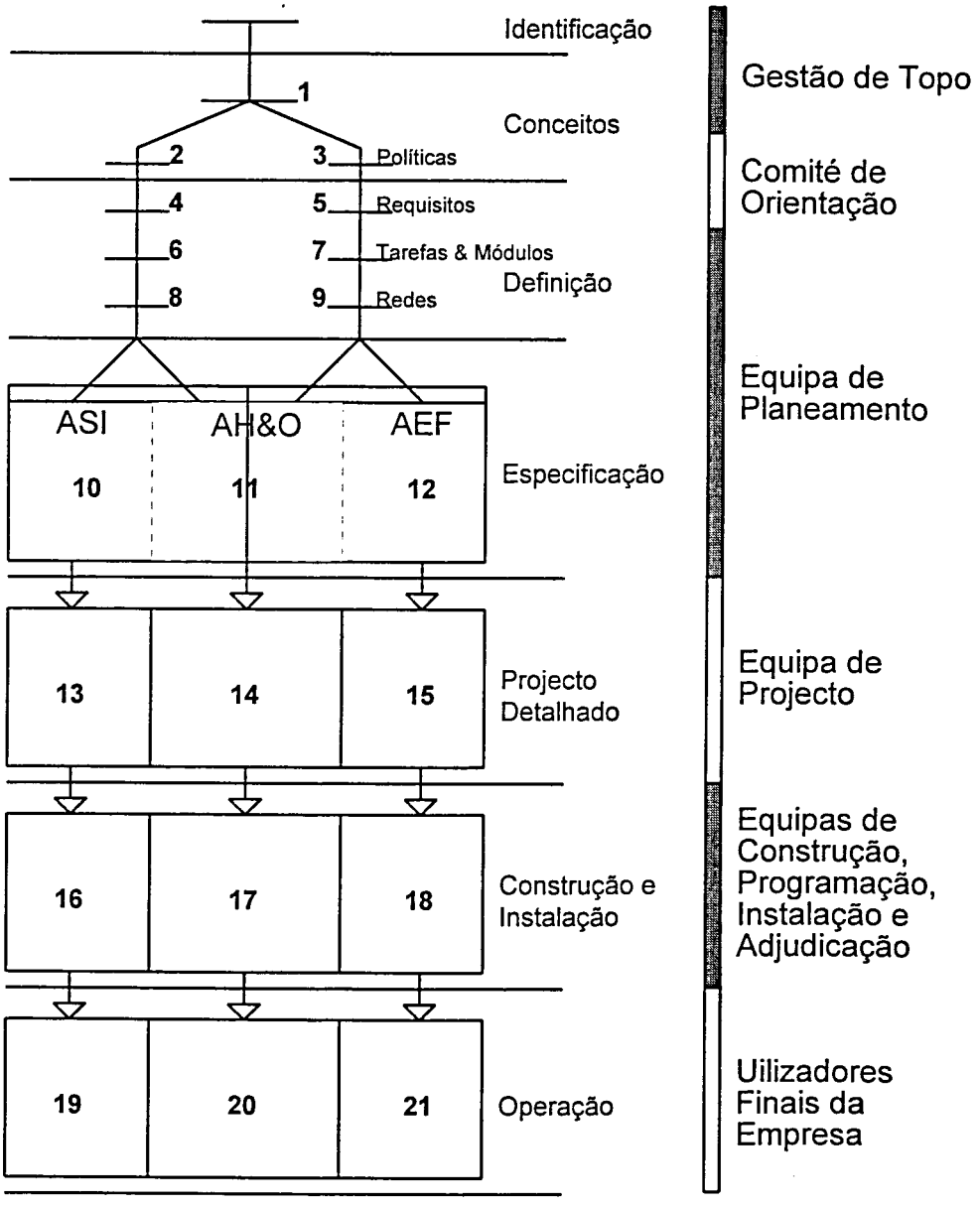


Figura 5.2 - Atribuição das Tarefas no Desenvolvimento do Projecto de Integração

As interfaces tipo 2 são as interfaces existentes entre duas fases consecutivas de um projecto de integração (figura 5.2). Surge assim um conjunto de conceitos e de requisitos relativos à transferência de informação entre as múltiplas fases do projecto de integração dizendo portanto respeito a todas as interfaces deste tipo: (1) toda a informação transferida deve ter uma representação interpretável pelas pessoas envolvidas, seja ela trocada entre grupos de trabalho humanos ou entre máquinas, quando é usada alguma forma de projecto

assistido por computador (CAD/CAE/CAXX); (2) o trabalho levado a cabo numa determinada fase da arquitectura do projecto deve-o ser no formato através do qual será transmitido à fase seguinte, por forma a minimizar o esforço global e os possíveis erros de conversão; (3) a descrição requerida é a necessária para a rápida compreensão do material transferido através das interfaces por parte dos elementos da fase seguinte, que serão responsáveis pelo seu tratamento; (4) cada uma das fases do projecto está separada da fase subsequente por uma interface que trata informação muito particular; (5) a transferência de informação entre as interfaces do projecto envolve sempre a transmissão entre pessoas ou grupos de pessoas, sendo exigida a máxima facilidade de interpretação e compreensão; (6) o grau de formalismo usado nesta arquitectura é específico a cada fase, devendo envolver a mínima complexidade de expressão necessária para assegurar a não ambiguidade da interpretação e a completa compreensão da informação transferida entre as várias fases, e devendo sempre que possível ser feita a adaptação dos formatos e dos meios de expressão à fase que os vai usar.

5.2.2 As Tarefas, as Pessoas e a Documentação envolvida na PERA

A figura 5.2 ilustra as diversas fases do desenvolvimento do projecto de integração segundo a arquitectura de Purdue, fazendo a atribuição das tarefas a cada um dos grupos específicos de pessoas envolvidas. Na tabela 5.2 são apresentados os documentos produzidos em cada fase do desenvolvimento, sendo distinguidas as situações onde surgem documentos informais, formais ou ambos os tipos.

Tabela 5.2 - Documentação por equipa num projecto de Integração

Equipa Responsável	Documento informal	Documento Formal
Gestão de Topo	Texto, listas e tabelas	Provavelmente impossível
Comité de Orientação	Texto, listas e tabelas	Provavelmente impossível
Equipa de Planeamento	Texto, listas, tabelas e diagramas	Modelos matemáticos, gráficos obedecendo a formatos e simbologia rígida, podendo este trabalho ser assistido por computador
Equipa de Projecto	Manuais, esquemas do sistema apresentados de forma não formal	Programas de CAD, bases de dados de esquemas, programas de controlo numérico, especificações formais, utilização de técnicas de engenharia de software (apoiadas por ferramentas CASE)
Equipas de Construção, Programação, Instalação e Colocação em Serviço	Manuais, esquemas do sistema apresentados de forma não formal	Gestão informática do projecto, "procurement" apoiado em computador
Utilizadores Finais da Empresa	Manuais de instrução, Quadros de Escalonamento	Instruções computorizadas, processos altamente automatizados, optimização contínua do processo, definição das tarefas

A recursividade da arquitectura de Purdue poderá ser explorada, caso seja necessário caracterizar cada uma das fases do projecto através de um novo projecto autónomo onde a

arquitectura PERA pode ser novamente aplicada. Esta abordagem permite além disso uma mais simples caracterização da interacção entre cada uma das fases do projecto.

Comparando a PERA com outros modelos existentes para o suporte ao ciclo de vida de sistemas fabris, tais como o modelo em "espiral" e em "cascata", será importante referir que de facto estes últimos não cobrem todo o ciclo de vida genérico de um sistema [31,87]. Com efeito, a parte do ciclo onde se estabelecem as ideias iniciais e os conceitos envolvidos no sistema futuro, assim como os seus requisitos, não são por eles contemplados. É pois ignorado um dos factores críticos de sucesso da implementação de um sistema fabril integrado que é o envolvimento explícito da gestão de topo. Além disso, estes modelos também não consideram explicitamente o papel das funções humanas e da arquitectura organizacional, e reduzem-se muitas vezes à implementação (para não dizer quase sempre) do sistema de informação.

5.2.3 Independência das Metodologias e Ferramentas de Modelação

A utilização da arquitectura de Purdue, deixa de imediato transparecer a sua independência das ferramentas que venham a ser utilizadas. A complexidade do projecto irá ser factor determinante na necessidade da utilização de metodologias mais ou menos sofisticadas, bem como das técnicas de modelação associadas hoje crescentemente suportadas por ferramentas informáticas adequadas.

As diversas fases da PERA ilustradas na figura 5.2 enquadram a realização de tarefas específicas, sendo cada uma delas responsável pela produção de documentação específica e adequada. Da tabela 5.2 verifica-se que a Equipa de Planeamento (executando as tarefas 6 a 12) tem necessidade de recorrer a técnicas de modelação, nomeadamente nas tarefas 10 e 12, onde são caracterizadas as arquitecturas funcionais do Sistema de Informação e do Equipamento de Fabrico. Poderão ainda ser usados modelos para a descrição funcional da arquitectura Humana e Organizacional [88].

No contexto da arquitectura de Purdue foram pois identificadas três áreas relacionadas (ASI, AH&O e AEF) onde é evidente a necessidade de modelos adequados sempre que a complexidade do sistema a analisar ou do projecto de integração a efectuar o justifique. Na última década emergiram diversos métodos de suporte a actividades de modelação empresarial, levadas a cabo com os mais diversos objectivos. Estes métodos surgiram da necessidade, sentida por profissionais das mais diversas áreas, de representar os diferentes aspectos da empresa, com o objectivo de melhor a compreender e, com base numa mais apurada percepção da realidade, nela actuar para elevar o seu desempenho. Esta necessidade de representação do mundo real, e em particular das empresas industriais, através de modelos é utilizada por especialistas em economia, investigação operacional, organização, auditoria e informática.

Desde a segunda guerra mundial que a investigação operacional propõe descrições da empresa através de modelos matemáticos essencialmente para apoio à decisão. A teoria da organização modela a empresa em função da sua estrutura, funcionamento, gestão, recursos humanos, papel na sociedade e muitos outros aspectos tais como mecanismos de coordenação. As tecnologias da informação vieram posteriormente fomentar a modelação empresarial segundo uma nova perspectiva. No início da década de 70, as empresas eram modeladas através da representação dos fluxos de dados e da estrutura das bases de dados. A utilização de sistemas de gestão de informação colocou uma enorme ênfase nos sistemas de apoio à decisão onde o modelo do sistema de informação era por si só o modelo empresarial [89].

A insatisfação relativamente ao estado-da-arte e à sua aplicabilidade levou no final dos anos 80, a novos esforços no sentido da modelação empresarial. Na Europa, por exemplo, o modelo CIMOSA, resultante de considerável esforço envolvendo empresas e instituições de I&D e pretendendo integrar vertentes distintas de uma empresa industrial, foi alvo de grande interesse por parte da comunidade científica. As novas necessidades e os avanços emergentes na modelação empresarial prendem-se com dois aspectos. Por um lado surge o conceito de produção integrada por computador, defendendo que as tecnologias da informação podem ser estendidas a todas a funções da empresa e não apenas às funções de gestão. Por outro lado houve nas últimas duas décadas um grande incremento nos métodos e técnicas de modelação formal, permitindo um melhor e mais eficaz suporte às actividades de modelação.

5.3 Modelação para a Simulação e Controlo do Sistema Integrado de Fabrico

Neste ponto será tratado o tema genérico da modelação e da simulação dos sistemas integrados de fabrico. Uma vez justificada a sua necessidade e as vantagens decorrentes da sua utilização, será introduzido o tópico da modelação de sistemas onde não só serão caracterizadas as diversas vertentes da modelação de um sistema, mas ainda a própria actividade de modelar, concluindo-se este ponto com a classificação das técnicas de modelação.

De uma forma muito breve são também referidos alguns dos desenvolvimentos recentes nas metodologias de modelação, nomeadamente apoiadas em ferramentas CASE ("Computer Aided Software Engineering"). Neste contexto é apresentada uma curta perspectiva histórica da utilização destas ferramentas nos Sistemas Integrados de Fabrico.

Levando em consideração a Arquitectura de Referência de Purdue, será então abordada a utilização dos modelos de simulação no âmbito da AEF, para o controlo das operações no sistema de fabrico, e da ASI, no apoio ao projecto, desenvolvimento e instalação do sistema de informação. Na sequência desta exposição surge a referência à utilização dos modelos para a simulação e controlo das operações no sistema de fabrico. Este tema é aqui tratado com algum detalhe, sendo nomeadamente apresentada e discutida uma arquitectura típica do Sistema de Controlo, além das vantagens / desvantagens da utilização da simulação no controlo do sistema de fabrico.

Apresenta-se ainda a utilização de modelos para o projecto e simulação do sistema de informação, assunto que é tratado com algum pormenor, incluindo referências a ferramentas CASE e a Ambientes de Desenvolvimento. Faz-se então a caracterização da estrutura típica dos sistemas de informação no ambiente industrial, à qual se segue uma exposição do uso de modelos / linguagens de especificação formal na simulação dos sistemas de informação.

Este ponto termina tratando o tema da modelação orientada ao objecto no âmbito do presente trabalho. Neste âmbito, além de serem enunciados os conceitos fundamentais relativos à programação orientada ao objecto, é introduzida a noção de bloco construtivo básico na modelação do sistema de fabrico. Em conclusão são apresentadas as vantagens da utilização da modelação orientada ao objecto aplicada aos sistemas de produção.

5.3.1 Justificação da Modelação e Simulação

De acordo com o dicionário Lello Universal, a Simulação é definida como "um método que consiste em recorrer a meios artificiais e a calculadores electrónicos para realizar

experiências sobre fenómenos económicos, políticos, etc.", sendo o Simulador definido como "um aparelho capaz de reproduzir o comportamento de um aparelho de que se deseja estudar o funcionamento, ou de um certo corpo de que se deseja seguir a evolução".

Quando é construído um modelo de simulação, é naturalmente exigida a sua adequação em termos da capacidade de análise do comportamento do sistema modelado sob um variado número de situações de operação. A simulação oferece múltiplas vantagens, permitindo que sejam realizados testes quando o sistema ainda está em desenvolvimento, além de posteriormente permitir a optimização do sistema em estudo sem interferir no seu funcionamento normal.

A simulação e os computadores estão intimamente ligados, uma vez que o modelo pode ser definido como um programa de computador. Alterando parâmetros no programa, o modelo vem alterado permitindo que diversos cenários sejam rapidamente analisados. A simulação constitui assim uma poderosa ferramenta ajudando na busca de soluções para problemas críticos durante o projecto, planeamento, implementação e operação de novos sistemas, ou ainda na análise (diagnóstico) de sistemas existentes e no estudo de alterações (prognóstico) com vista à melhoria do seu desempenho.

A complexidade dos sistemas de produção justifica a utilização de técnicas diversas de simulação, uma vez que quanto mais complexo o sistema menos apropriado se torna o seu estudo através de modelos matemáticos.

Na modelação para a simulação, é realizada a descrição completa do funcionamento do sistema através da descrição de cada um dos componentes assim como da forma como eles interagem entre si. Os métodos de simulação podem ser classificados como discretos ou contínuos, podendo a informação do modelo ser apresentada das mais diversas formas, por exemplo relatórios estatísticos, gráficos (curvas, histogramas, etc.), e podendo em alguns sistemas ser observada a evolução temporal do sistema através de animação.

A simulação implica a utilização de uma linguagem de programação que pode cair em uma de quatro possíveis categorias: (1) Linguagens de alto nível de uso geral (por exemplo Fortran, C, Pascal, SmallTalk), (2) Linguagens de simulação (por exemplo GPSS, SLAMII, SIMAN), (3) Simuladores com linguagens dedicadas para o ambiente fabril (por exemplo MAST, MAP/I, AUTOMOD, WITNESS, PS-SIM, ProModel, Simple++), (4) Linguagens usadas em métodos de especificação formal dotadas de módulos de animação da execução da simulação (por exemplo Estelle [91], LOTOS [92], SDL [24]).

No âmbito deste capítulo, vamos restringir o nosso estudo aos sistemas de gestão de operações fabris em ambientes discretos de produção, bem como aos sistemas de informação associados. Assim sendo, os parágrafos seguintes vão focar a utilização de modelos de simulação para cada um destes ambientes.

5.3.2 A Modelação de Sistemas

O Sistema

Um sistema é um conjunto de elementos organizados de acordo com uma determinada estrutura existindo interacção entre eles e com o ambiente que os envolve. Este conjunto de elementos realiza funções, transformando matéria, energia ou informação, e evoluindo no tempo segundo um determinado objectivo. O facto de esta definição ser derivada daquela que é dada para os sistemas vivos, põe em foco a sua generalidade na aplicação a sistemas complexos incluindo os sistemas de fabrico [67].

No quadro da modelação de sistemas deveremos distinguir duas categorias de modelos: o modelo lógico do sistema, apresentando a suas necessidades funcionais, dinâmicas e estáticas globais; e o modelo físico do sistema, apresentando a sua arquitectura em termos dos seus componentes reais e operacionais. A modelação do sistema poderá ainda ser caracterizada segundo as quatro dimensões apresentadas na figura 5.3: a descrição dos seus componentes, que vai dar origem à arquitectura física do sistema; a descrição das características e propriedades estáticas do sistemas incluindo os dados, a estrutura, e a informação conservada e memorizada; a descrição do comportamento do sistema do ponto de vista da sua reacção dinâmica (imediata) e da sua evolução no tempo (a um longo prazo); e finalmente a descrição das funções que o sistema deverá assegurar [67].

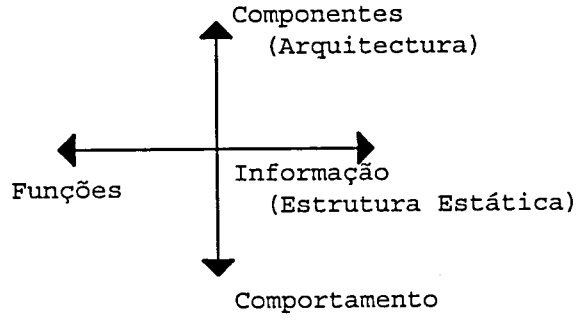


Figura 5.3 - Os Quatro Eixos da Modelação de um Sistema

A Engenharia de Sistemas

A engenharia de sistemas preocupa-se com a resolução dos problemas colocados na realização de sistemas de grande dimensão e caracterizados por elevado grau de complexidade (sistemas de telecomunicações, centrais nucleares, sistemas informáticos heterogéneos, redes de distribuição de energia, lançadores espaciais, etc.). No contexto da realidade actual, da complexidade das empresas industriais e dos ambientes envolventes resultam problemas que podem ser abordados através da engenharia de sistemas. O objectivo da engenharia de sistemas é que, uma vez realizado, o sistema possua as funções, as propriedades e o comportamento esperado. A engenharia de sistemas procura assim identificar as propriedades esperadas do sistema, prever o seu comportamento, e estruturar as suas propriedades e os seus comportamentos, de forma a que as características desejadas possam emergir de uma forma controlada dentro de limites aceitáveis.

A engenharia dos sistemas tem uma abordagem global ou, como é usual hoje em dia referir-se, *holística*, situando-se ao nível da análise, especificação e da concepção do sistema com o objectivo do seu desenvolvimento, integração e validação. A fase de análise do sistema coloca em foco a modelação do problema, tendo em vista a construção de uma especificação. A especificação é seguida pela fase de concepção do sistema traduzindo a solução ao problema, independentemente da utilização de um tipo ou outro de engenharia ou tecnologia. Surge finalmente o ponto de encontro entre as necessidades (resultado da fase de concepção) e o estado-da-arte da tecnologia disponível sobre a qual se realizará o desenvolvimento do sistema [67].

A Actividade de Modelar

Modelar é compreender! O objectivo é, no âmbito deste trabalho, a modelação das actividades da empresa (o sistema neste caso) procurando o suporte para uma melhor compreensão da realidade para a explicar / descrever ou para o apoio à decisão (tabela 5.3).

O modelo é assim uma abstracção do sistema existente num dado instante do tempo e segundo um determinado ponto de vista.

Tabela 5.3 - Porquê a Modelação ?

Modelar para Compreender	• "O quê?" antes do "Como?" • Analisar antes de construir. • Reduzir a complexidade.
Modelar para Comunicar	• Através de linguagem comum tão formal quanto possível, evitando ambiguidades. • Ganhando tempo, i.e. usando se possível modelos gráficos (não matemáticos) para uma mais fácil compreensão.
Modelar para Validar	• Usando ferramentas que ajudem a controlar a sintaxe, a coerência e a verificar se o modelo está completo. • Assegurar que modelo está de acordo com as necessidades iniciais. • Verificar, simular e executar, obtendo assim uma validação mais sistemática e rigorosa, que poderá traduzir-se na expressão do modelo usando uma linguagem formal.

O modelo tem três diferentes vertentes: a de representação, a de abstracção e a pragmática. A *vertente de representação* distingue se o modelo é isomórfico ou homomórfico. No primeiro caso, o modelo deverá representar a realidade de uma forma muito fiel, enquanto que no segundo será um modelo mais abstracto do sistema real concentrando-se na sua estrutura e nos pontos mais importantes. Assim sendo, a construção de um modelo isomórfico será, na maior parte dos casos, ou impossível ou desprovida de interesse prático. A *vertente de abstracção* refere-se à (/ao grau) correspondência que deverá existir entre o sistema real e o modelo do ponto de vista da abstracção e da agregação da realidade. A *vertente pragmática* refere-se aos objectivos e à utilidade do modelo. Como vimos, os objectivos do modelo serão o de descrever o sistema, o de o explicar e o de suportar tarefas de apoio à decisão. Um modelo simplesmente descritivo fornece uma informação comprimida acerca do estado do sistema. O modelo explicativo define o comportamento e a funcionalidade do sistema. O modelo para apoio à decisão fornece um quadro estabelecendo os objectivos e restrições da decisão relativamente ao sistema, e permite que o utilizador interactue com ele.

Na construção do modelo é normal a utilização de uma técnica de modelação. Tal técnica faz geralmente uso de um conjunto de blocos construtivos básicos e de procedimentos descritivos que estabelecem a utilização desses mesmos blocos.

No contexto da integração de sistemas de fabrico a modelação da empresa tem três objectivos: o da análise, o do projecto e o da gestão operacional da empresa. Tendo em vista estes objectivos e a definição dada para os modelos descritivos, explicativos e de apoio à decisão, fazemos corresponder: aos modelos com objectivos de análise, os modelos descritivos e explicativos; aos modelos para o projecto do sistema de informação / fabrico e

para a gestão operacional, os modelos de apoio à decisão. De facto, os modelos de análise são apenas modelos informativos, explicando o comportamento e raramente permitindo suportar decisões influenciando aquilo que realmente se está a passar, enquanto que os modelos de apoio à decisão são de facto construídos para permitir controlar o sistema [89].

A Modelação no seio da Empresa de Fabrico

O interesse da modelação empresarial numa perspectiva holística advém sobretudo da necessidade de melhor conhecer e compreender a realidade da empresa, e de incrementar uma melhor comunicação no seu seio para suporte da sua melhor gestão operacional. São pois essenciais a comunicação, assim como o tratamento e a troca / partilha de informação, sendo hoje em dia imprescindível a presença de um sistema de informação para a realização destas tarefas cruciais.

Os requisitos técnicos que se colocam à modelação empresarial são os seguintes [38]:

- Metodologia de projecto estruturada e integrada.
- Capacidade de gestão da complexidade, consistência, coerência e de verificação se o modelo está completo.
- Propriedades que permitam a decomposição funcional, a descrição do comportamento, a definição organizacional das entidades envolvidas e a ajuda às actividades de apoio à decisão.
- Capacidade para cobrir as diversas perspectivas da empresa: orientação à informação, ao processo e ao comportamento.
- Consideração de aspectos relevantes da empresa, i.e. estáticos, dinâmicos e organizacionais.
- Cobertura de todo o ciclo de vida do sistema, i.e. a evolução da empresa.
- Produção de um modelo de simulação.
- Fornecimento de um modelo de implementação executável.
- Isolamento da definição de requisitos da sua implementação.
- Mecanismos de reutilização e manutenção do modelo.

Além destes aspectos técnicos, do ponto de vista do utilizador é essencial a simplicidade da abordagem. De facto, o método de modelação deverá estar tão próximo quanto possível da forma de conceptualizar, raciocinar e de coordenar as operações da empresa, sendo de uma forma geral insuficientes as abordagens tradicionais.

Tradicionalmente os modelos eram vistos como algo que se utilizava com um único objectivo de curto prazo podendo depois ser abandonados (os denominados modelos descartáveis). A utilização da metodologia orientada ao objecto vem gradualmente tornar obsoleta esta atitude. É agora possível definir a natureza do problema e da questão que se deseja colocar, e configurar o modelo ou alterá-lo partindo de um conjunto disponível de modelos / blocos construtivos mais básicos previamente definidos e construídos. Torna-se assim possível a construção de modelos configuráveis de acordo com requisitos específicos [93].

A utilização do modelo com um único objectivo específico frequentemente restritivo, juntamente com a estratégia do modelo descartável, que é muito cara, constitui uma barreira considerável a uma maior utilização de modelos. Por outro lado, a dificuldade que os não especialistas têm na utilização de modelos é também um ponto de considerável importância por obstruir à sua vulgarização. A utilização de modelos, tradicionalmente restrita a especialistas, exigia com efeito um grande esforço tanto na aprendizagem do modelo como do seu funcionamento. Este problema tem vindo a ser resolvido através da utilização de linguagens de modelação mais evoluídas, tendo as melhores interfaces gráficas dos sistemas

de modelação também contribuído para reduzir esta barreira à sua vulgarização. Uma outra limitação das soluções obtidas através das técnicas de modelação tradicionais é a sua visão estreita de uma parte apenas de todo o sistema a ser modelado. Finalmente, raras vezes o analista segue o desempenho do sistema real e o compara com o previsto pelo modelo, e poucas vezes são introduzidos no modelo dados resultantes da operação do sistema real por forma a suportar a sua calibração e validação.

Uma nova perspectiva na utilização dos modelos aponta para a necessidade de um ciclo completo incluindo a modelação e a utilização de modelos no ambiente operacional, onde o modelo é visto como parte integral do sistema operacional / de decisão. Exige-se deste modo uma mudança de atitude perante a utilização de modelos e da simulação, afectando três tipos diferentes de utilizadores:

- Os investigadores em modelação e no desenvolvimento de modelos, que devem liderar essa mudança de atitude através do repensar do papel potencial da modelação face às tendências actuais.
- Os analistas e outro pessoal técnico que usa modelos e simulação, os quais constituem o grupo que deve estabelecer a ponte entre os investigadores e os gestores / decisores que fazem uso dos resultados da modelação / simulação.
- Gestores e decisores a todos os níveis, que são em última análise os reais utilizadores do esforço de modelação. São eles que criam e geram o ambiente no qual os modelos são desenvolvidos, aplicados e mantidos, e as novas aproximações à construção dos modelos devem levar em consideração as necessidades reais deste grupo.

Um dos conceitos novos mais importantes prende-se com o facto de que o modelo operacional de uma instalação fabril deverá ser realizado com um grande detalhe, comparável ao pormenor a que é levado o projecto de um dos produtos da empresa. A gestão da configuração do sistema deve também ser testada no modelo, garantindo-se que as alterações no modelo reflectem alterações realizadas no sistema real. Este ambiente deve ser completado através da utilização de metodologias de análise e de simulação que partilhem o mesmo modelo. Além disso, o ambiente de modelação deve ser capaz de acomodar diversos níveis de detalhe do modelo, por forma a melhor se adaptar às diferentes necessidades de modelação, assim como permitir a construção de modelos integrados em que as suas diferentes partes possam ter níveis de detalhe ou granularidade distintos.

Classificação das Técnicas de Modelação

Os métodos usados para a modelação empresarial são os mais variados, cada um deles apoiando-se em formalismos diversificados. Torna-se por este motivo difícil a avaliação e a classificação dos métodos existentes. Na realidade existe uma grande interdependência entre a metodologia usada e o seu domínio de aplicação. Conforme o problema esteja mais próximo da engenharia de software ou da gestão de operações fabris, assim poderão ser utilizadas diferentes metodologias.

Os diferentes métodos de modelação da empresa podem classificar-se em: métodos simples, múltiplos ou múltiplos agregados [38]. Os métodos simples envolvem apenas um formalismo de modelação. Os métodos múltiplos são constituídos por dois ou mais métodos simples. Os métodos múltiplos agregados são constituídos por dois ou mais métodos simples mas que são encapsulados por um nível hierárquico mais elevado. Na tabela 5.4 apresentam-se as vantagens e desvantagens de cada um dos métodos.

Tabela 5.4 - Classificação das Métodos de Modelação

Métodos	Vantagens	Desvantagens	Exemplos
Simples	fácil utilização	não cobrem o ciclo de vida do sistema, e só cobrem um aspecto isolado da modelação	IDEF0 [40]
Múltiplos	fácil utilização por serem constituídos por um conjunto limitado de métodos simples	risco de uma modelação incompleta, de inconsistência e de incoerência	MERISE [94], SADT [95], SA [96], SA-RT [97]
Múltiplos Agregados	cobrem todo o ciclo de vida e promovem a integração de todos os aspectos relevantes, gerindo a complexidade	suporte computacional mais exigente	CIMOSA (onde os princípios como o IDEF0 e as Redes de Petri [98] não estão visíveis) [38].

5.3.3 Recentes Desenvolvimentos em Metodologias de Modelação

No âmbito das metodologias de modelação têm tido lugar recentes desenvolvimentos, com o emergir de técnicas variadas na área das linguagens de modelação [93]. A consequente necessidade do suporte computacional a este tipo de linguagens traduz-se através no aparecimento de ferramentas CASE bem como de bancadas de modelação e simulação.

Os mais recentes desenvolvimentos em metodologias de modelação têm estado intimamente relacionados com Linguagens de simulação, oferecendo as mais avançadas interfaces gráficas para entrada e saída de dados e apresentando mesmo algumas delas facilidades de animação; são largamente divulgadas as seguintes: SLAM-SYSTEM, Cinema/SIMAN, Simple++, SIMFACTORY e WITNESS. Outro método, a Teoria das Filas de Espera, surgiu como o mais utilizado no estudo analítico de modelos de sistemas de produção. As Redes de Petri, constituem por seu lado uma linguagem gráfica formal para a descrição e análise de sistemas que apresentam propriedades de assincronismo e de concorrência, podendo ser usadas na representação de fluxos de informação e de controlo em sistemas de fabrico. Além disso, a introdução de transições temporizadas transformou as redes de Petri numa poderosa ferramenta de avaliação de desempenho. Um outro método consiste na aproximação ao problema da modelação através da Modelação Estruturada, utilizando técnicas matemáticas formais e um ambiente computacional para criar, representar e manipular uma grande variedade de modelos. Uma outra metodologia conhecida como DEVS-Scheme utiliza um ambiente de modelação baseado em conhecimento, combinando inteligência artificial com simulação discreta por eventos, desenvolvido para a modelação de arquitecturas de computadores e de comunicações.

A utilização de ferramentas de suporte à modelação, tais como ferramentas CASE disponíveis para múltiplas metodologias, apresenta benefícios consideráveis para os seus utilizadores tais como: as interfaces gráficas adequadas, geração automática de código, gestão das configurações do software, e várias ferramentas para a gestão da simulação e / ou do ciclo de vida do modelo. Por outro lado, o conceito mais abrangente de Bancada (de Trabalho) de Modelação/Simulação tem grandes vantagens, fornecendo ao seu utilizador, o

modelador / analista, um ambiente poderoso de trabalho através de um conjunto de ferramentas configuráveis que podem ser utilizadas para trabalhar sobre um modelo existente ou na criação de um novo modelo. Num sistema deste tipo, deverá existir uma biblioteca de objectos reutilizáveis (primitivas de software), disponíveis para a criação de modelos para fins específicos. Entre as várias ferramentas existentes, deverá existir suporte para a análise estatística tanto dos dados de entrada como dos dados de saída da simulação.

5.3.4 As Ferramentas CASE na Integração dos Sistemas de Fabrico

As exigências colocadas pela Integração do Sistema de Fabrico (ISF), tanto do ponto de vista da AEF como da ASI, leva a que o desenvolvimento deste tipo de sistemas deva ser realizado por equipas de profissionais da informática e da automação. De facto, os anos 80 caracterizaram-se pelo tratamento da ISF como um problema quase exclusivamente do âmbito das tecnologias da informação, com grande ênfase na automatização e na robotização fabris. Muitos dos sistemas desenvolvidos incluíram o desenvolvimento de software dedicado envolvendo um enorme esforço de programação. O sucesso comprovado destes sistemas dedicados determinou o conceito de então de ISF, mas foram também postas em foco as suas limitações: custos elevados, perspectiva exclusivamente tecnológica e cenários de "ilhas de automação" de difícil integração. Face ao crescimento do mercado da integração de sistemas, não existe no entanto mão de obra qualificada em quantidade suficiente para permitir a construção de todos os sistemas necessários: há falta de engenheiros qualificados para a realização deste tipo de trabalho e o tempo de desenvolvimento e integração de sistemas dedicados é muito longo. Por outro lado, nos anos 90 as prioridades tendem a centrar-se nos sistemas de apoio ao planeamento e controlo do fabrico, garantia da qualidade, diagnóstico inteligente, engenharia simultânea, re-engenharia do processo do fabrico, etc., predominando as metodologias, as ferramentas de apoio à decisão e a reorganização sobre as vertentes tecnológicas de informatização e a automação. Neste ambiente de crescente complexidade das tarefas / actividades ligadas a projectos de integração é de cada vez maior a relevância a disponibilidade de ferramentas adequadas.

Na primeira das fases acima referidas, o apoio à realização de projectos de ISF através da utilização de ferramentas CASE não teve praticamente expressão. O desenvolvimento de ferramentas CASE está muito relacionado com a evolução das tecnologias da engenharia do software. As primeiras ferramentas CASE que apareceram datam do início dos anos 80, desenvolvidas por Yourdon/DeMarco, com a análise e a especificação estruturadas, os diagramas de fluxo de informação e os dicionários de informação suportados por uma representação gráfica simples e intuitiva. Actualmente, a modelação orientada ao objecto tem vindo a ganhar importância, fornecendo os princípios para o desenvolvimento de ferramentas CASE adaptáveis, as chamadas Meta-CASE [38].

As ferramentas CASE fornecem o suporte ao desenvolvimento de sistemas informáticos através de uma metodologia formal assistida por computador, permitindo o acompanhamento do ciclo de vida do sistema desenvolvido. A utilização de tais ferramentas tem em vista acelerar o desenvolvimento e facilitar a manutenção, bem como promover / garantir tempos de desenvolvimento previsíveis e soluções fiáveis.

O suporte computacional de um método requer a descrição formal desse mesmo método. A um método de modelação está associada a respectiva linguagem, princípios e representação gráfica, além de um mecanismo de execução do modelo. Distinguimos assim

três requisitos fundamentais: uma interface gráfica e / ou textual, conceitos ou formalismos do método e um modelo de execução.

Podemos ainda distinguir dois tipos de ferramentas CASE, as ferramentas Meta-CASE, onde o ambiente de trabalho pode ser "programado" para suportar um determinado método, e as ferramentas CASE específicas, especialmente concebidas para suporte de um método de especificação em particular [38]. As ferramentas do primeiro tipo normalmente não suportam a execução do modelo, ao contrário das segundas que implementam formalismos que, como por exemplo as Redes de Petri, são directamente executáveis.

No suporte de métodos de modelação através de ferramentas CASE, existem três alternativas possíveis: (1) desenvolvimento de raiz de uma ferramenta orientada para a aplicação de um determinado método de modelação; (2) disponibilidade de uma ferramenta suportando um conjunto de primitivas básicas, que deverão ser semelhantes às do método a implementar e que o utilizador deverá então fazer corresponder às primitivas do método que deseja utilizar; (3) ambiente fornecido pela ferramenta CASE adaptável por forma a providenciar um nível de abstracção relacionado com o método que se deseja utilizar. No parágrafo 5.3.6 o tema da utilização das ferramentas CASE será retomado no âmbito do estudo dos modelos para a simulação e projecto dos sistemas de informação.

5.3.5 Modelos para Simulação e Controlo das Operações no Sistema de Fabrico

A arquitectura de referência de Purdue (figura 5.2 na página 112), preconiza na vertente dos equipamentos de fabrico (AEF) de um projecto de integração, a fase de definição e análise (7, 9), a fase de especificação (12) e a fase de projecto detalhado (15). Nestas fases realizam-se respectivamente: (7) a identificação das tarefas, módulos e macro-funções necessárias para, de acordo com os requisitos, levar a cabo o fabrico do(s) produto(s) a serem comercializado(s); (9) os diagramas de fluxos de processos, mostrando a conexão entre as funções estabelecidas no ponto anterior; (12) o projecto funcional da AEF, recorrendo a técnicas de projecto e optimização de "layouts" da instalação fabril. Só na fase (15) é realizado o projecto detalhado dos componentes, processos e equipamentos da AEF. É pois evidente a necessidade de metodologias, e se possível ferramentas de suporte, que apoiem as diversas fases do projecto.

No contexto da produção integrada por computador, o controlo do sistema de fabrico tem um papel muito importante. Conceitos recentes apontam para utilização de ferramentas de modelação / simulação com o duplo objectivo da simulação e da gestão / controlo das operações fabris [99].

O primeiro objectivo da integração do sistema de fabrico é torná-lo flexível, reactivo e de elevado desempenho, mantendo elevados os padrões da qualidade e baixos os custos. Estes requisitos colocam grandes exigências às actividades de planeamento e controlo em todas as suas vertentes. Neste ponto vamos todavia concentrar a nossa atenção na modelação do chamado "chão-de-fábrica" ("shop floor"), tendo em vista a simulação e o controlo de operações fabris.

O controlo em tempo-real da instalação fabril tem lugar no denominado nível operacional ou de execução. A este nível é gerado um plano / escalonamento detalhado de curto prazo, determinados quais os materiais / componentes que vão sofrer que operação e em que máquinas / recursos, sendo preparados ainda os planos de carga das máquinas assim como as listas de preparação de ferramentas. A informação de base para este plano provém do MRP, onde foi realizado um planeamento agregado e foram estabelecidas datas de conclusão do fabrico, assim como do estado actual da produção. O objectivo fundamental

do escalonamento é a construção optimizada deste plano detalhado, de uma forma rápida e exacta, com base nas datas de entrega, e na capacidade dos recursos e nos stocks disponíveis.

O controlo da produção preocupa-se com o despacho ou lançamento das ordens de fabrico e o seu encaminhamento, assim como com o controlo do fluxo de materiais através do "chão-de-fábrica", tirando vantagem das alternativas possíveis no âmbito do plano detalhado face ao estado efectivo dos em-curso-de-fabrico, recursos, stocks, etc.

A este nível só recentemente começaram a aparecer produtos informáticos de apoio, os denominados "Manufacturing Execution Systems" (MES) ou "Shop Floor Control Systems" (SFCS).

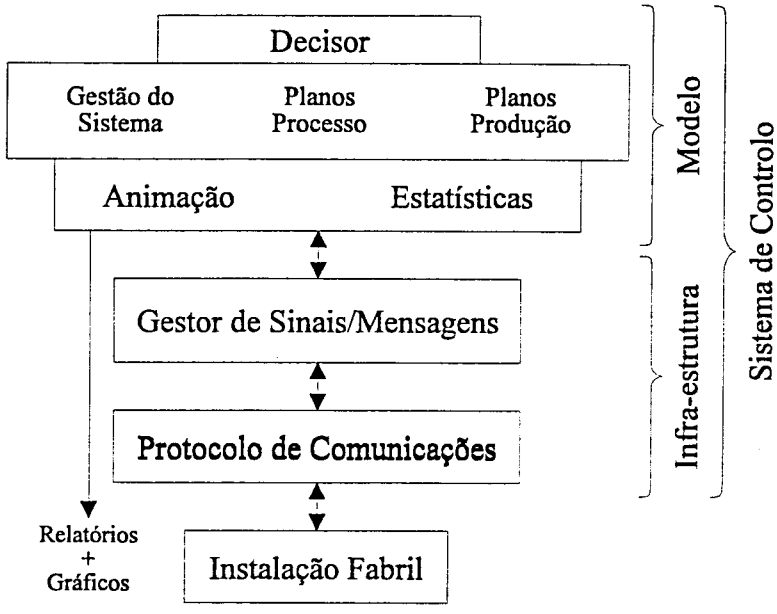


Figura 5.4 - O Modelo Fabril no Sistema de Controlo

A arquitectura do sistema desenvolvido no âmbito do Projecto ESPRIT SHOP-CONTROL, já discutida no Capítulo 4 desta dissertação, é apresentada a título de exemplo de uma forma simplificada na figura 5.4 de acordo com [99]. Uma das características avançadas deste sistema, e que o distingue das soluções actuais, é a integração das funcionalidades disponíveis nos ambientes de modelação e simulação orientadas ao fluxo de materiais, com as aplicações de escalonamento, despacho e controlo das operações fabris, aliadas a uma interacção em "tempo útil" com o "chão-de-fábrica". O sistema de controlo é, de acordo com a figura 5.4, constituído por um "modelo" sobre uma "infra-estrutura", cada um deles formado por várias partes ou blocos com funções específicas. A presença de um sistema de recolha de dados fornece ao sistema de controlo informação actualizada da instalação fabril para a actualização do seu modelo. A *gestão do sistema* é responsável pela adequada recolha e utilização da informação vinda da instalação fabril, e o *decisor* é responsável pelas decisões tomadas no decorrer do controlo em tempo-real da instalação fabril. As decisões configuradas no modelo de fabrico são tomadas de uma forma automática tendo em consideração os processos de fabrico em curso, assim como os planos de produção, e podem, sempre que necessário, ser alteradas pelo responsável da gestão do sistema. As decisões também poderão ser tomadas pelo operador por intervenção directa sobre os planos de produção detalhados. As facilidades gráficas de *animação* permitem que

a actualização em tempo-real do modelo se traduza na animação gráfica do modelo fabril, dando deste modo uma melhor percepção da evolução do sistema real e proporcionando facilidades de monitorização. As facilidades *estatísticas* podem ser utilizadas na realização de medidas sobre o modelo de simulação ou sobre a própria execução do modelo de controlo (tais como por exemplo, tempos médios de processamento, tempos médios de transporte).

O *gestor de sinais / mensagens*, é a porta de acesso ao equipamento fabril de recolha de dados, sendo ainda responsável pela conversão de formatos e pela interacção com os múltiplos protocolos existentes ao nível do equipamento fabril.

O ambiente desenvolvido no projecto SHOP-CONTROL assenta no pressuposto de que o programa (i.e. modelo) de simulação e o programa de controlo do sistema de produção são muito similares, motivo pelo qual deve ser procurada a sua reutilização. De facto, as regras (por exemplo, heurísticas de escalonamento) do sistema de produção necessariamente presentes no programa / modelo de simulação são disso um exemplo. Além disso, a maior parte dos eventos reais que ocorrem durante o controlo do processo são similares aos eventos simulados que ocorrem durante o controlo da simulação. Os modelos de simulação reutilizados no controlo do sistema de produção são denominados modelos executáveis, uma vez que a sua execução pode ser usada no controlo das operações. A figura 5.5 ilustra as áreas comuns dos ambientes de controlo e de simulação [99]. Além do mais, os simuladores apresentam módulos estatísticos e de animação próprios, que poderão ser usados com vantagem quando se deseja visualizar a evolução do sistema de produção real.

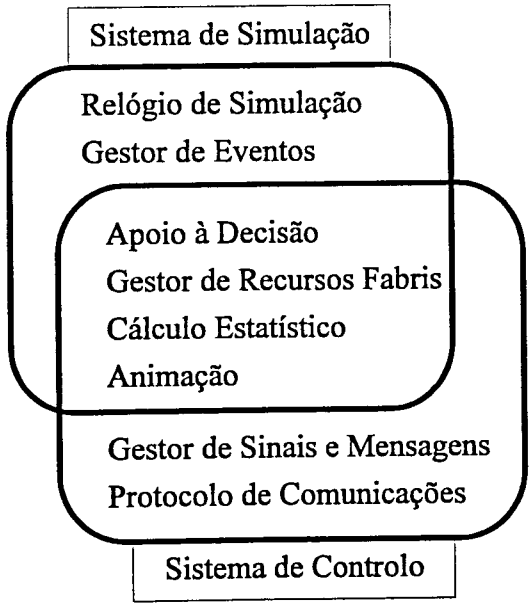


Figura 5.5 - Elementos comuns aos ambientes de simulação e controlo

Méritos da Utilização da Simulação no Controlo

Nos pontos que se seguem será realizada uma análise comparativa entre as vantagens e as desvantagens da combinação dos modelos de Simulação e de Controlo [99].

Vantagens da combinação Simulação / Controlo

- Linguagem de alto nível As linguagens de simulação são linguagens dedicadas à lógica da modelação do sistema de produção. São linguagens de programação de alto nível e de utilização mais simples do que as linguagens de uso genérico. Este tipo de linguagens parte da caracterização de blocos construtivos básicos, usados depois na construção de elementos mais complexos, numa estratégia de modelação base-topo.
- Só um programa A utilização do mesmo programa para a simulação e para o controlo poupa tempo e esforço, e permite que sejam eliminados possíveis erros na posterior conversão do programa de simulação no programa de controlo. Por outro lado, na utilização da simulação para a análise do sistema, deve existir uma completa sintonia entre a lógica do modelo de simulação e a lógica do controlo do sistema modelado (validação do modelo).
- Realização simultânea de simulação e controlo É, em alguns tipos de ambientes, possível a simulação de algumas das actividades do sistema de fabrico, coexistindo com outras actividades reais que estão a ser controladas. Além disso, a execução simultânea e em paralelo da simulação e do controlo do sistema real permite detectar em "tempo-real" desvios entre o planeado e o efectivamente cumprido.
- Lógica flexível Na fase de simulação podem ser analisadas lógicas alternativas para as estratégias da produção, por forma a optimizar o desempenho do sistema final a implementar.
- Alterações menos críticas Se após a instalação forem necessárias alterações nos modos de operação ou de simulação é relativamente simples a realização de pequenos ajustes. Comparando com os processos tradicionais, envolvendo codificação em C, Pascal ou Fortran, a realização de pequenos ajustes deixa de ser crítica. Por outro lado, dada a facilidade de alteração, pode ser vantajoso alterar as regras de produção entre a execução de dois planos de produção de acordo com a experiência de terreno.
- Reutilização de relatórios As linguagens de simulação dispõem geralmente de ferramentas estatísticas com produção de relatórios e saídas gráficas. É assim possível dispor do mesmo tipo de relatório para apresentar os resultados da simulação e da execução do sistema real, por forma a poderem mais facilmente ser comparados.
- Reutilização da animação O diagrama sinóptico da instalação fabril e as facilidades de animação são naturalmente aproveitados com a utilização do programa de simulação para a monitorização e a visualização do sistema.

Desvantagens na combinação Simulação / Controlo

- Execução mais lenta As linguagens de simulação dispõem de múltiplas funções que podem não ter qualquer interesse para o programa de controlo, facilidades essas que são normalmente a causa de uma execução mais lenta. Dependendo do nível / horizonte temporal a que o controlo estiver a ser realizado, este factor poderá tornar-se crítico.
- Mais uma linguagem A linguagem de simulação está codificada numa determinada linguagem de programação, complicando a integração com módulos definidos noutras linguagens.

De facto, a execução mais lenta terá cada vez menos significado com o aparecimento de máquinas mais rápidas e com maior capacidade de processamento. Por outro lado, as ferramentas estado-da-arte actuais dispõem já de mecanismos de comunicação com o exterior, permitindo assim a sua mais simples integração com software desenvolvido noutras linguagens. Assim sendo, comparando as vantagens e desvantagens da combinação simulação / controlo apresentadas, verifica-se que só se poderá ganhar com esta combinação aplicada a um sistema real. Na última parte deste capítulo, será apresentada a solução SHOP-CONTROL / PROFIT onde a simulação em tempo-real é usada no controlo "on-line" do sistema de produção.

5.3.6 Modelos para a Simulação e Projecto do Sistema de Informação

Introdução

Reportando-nos à vertente do sistema de informação (ASI) da arquitectura de referência de Purdue (figura 5.2 na página 112), serão agora consideradas as fases 6, 8 e 10 nas quais se realizam respectivamente: (6) a identificação das tarefas, módulos e macro-funções necessárias para responder aos requisitos colocados ao sistema de informação através de uma análise funcional; (8) a conexão dos módulos funcionais usando, por exemplo, diagramas de fluxo de informação ou outros métodos afins; (10) o projecto funcional da arquitectura do sistema de informação, recorrendo a técnicas de projecto funcional, e do modelo de informação, utilizando por exemplo o modelo entidade associação. Só no ponto 13 é realizado o projecto detalhado do equipamento e do software envolvido na ASI. É de novo evidente a necessidade de metodologias, e se possível ferramentas de suporte, que apoiem as diversas fases do projecto, sendo a utilização do CASE já vulgar na modelação do Sistema de Informação.

Cultura CASE versus Cultura dos Ambientes de Desenvolvimento

O conceito de CASE tem vindo a ser ocasionalmente associado com o mercado de ferramentas de edição de diagramas. Estes diagramas estão normalmente associados a metodologias documentadas na literatura como *projecto estruturado* [100,101,102]. Com origem na área das telecomunicações surge o SDL ("Specification and Description Language"), evoluindo para um standard na representação gráfica de sistemas de software. No contexto da linguagem Ada surgem, por exemplo, os chamados gráficos de Booch e de Buhr [103,104]. Um aspecto comum a todas estas metodologias é a disponibilidade de uma representação gráfica, tendo sido desenvolvidas ferramentas de software suportando o projecto estruturado através da utilização de notações gráficas, usando Diagramas de Fluxos de Informação (DFD), diferentes formas de Mapas de Estrutura, e métodos mais ou menos estandardizados, tais como HOOD e YOURDON [66,105].

Em oposição à denominada cultura dos ambientes CASE baseada em diagramas e em notações gráficas associadas, surge a cultura dos ambientes de desenvolvimento. Neste contexto surgem por exemplo o Interlisp e o Smalltalk, dando maior relevância ao sistema como um todo e usando conceitos de arquitectura e de integração com outras ferramentas, em oposição às ferramentas CASE orientadas para a produção de diagramas e caracterizadas pelo seu normal isolamento [66]. A tabela 5.5 faz uma comparação sumária entre estas duas abordagens alinhando algumas características marcantes de uma e outra.

Estas duas culturas apostaram em princípios distintos. A cultura dos ambientes defende um ciclo de edição-compilação-correcção bastante curto, e disponibiliza formas de suporte

à rápida realização de protótipos através da utilização de um ambiente de desenvolvimento exploratório propício a uma evolução do projecto em espiral com a produção de protótipos sucessivos. As técnicas baseadas em diagramas dão por seu lado uma maior ênfase à fase de projecto, assumindo a trivialidade da produção de código uma vez concluída esta fase, apresentando portanto um ciclo de vida tipo "cascata".

Fruto de uma influência mútua, as diferenças entre estas duas culturas têm vindo a ser atenuadas através do alargamento do domínio das ferramentas CASE ao apoio à integração através da geração de código. Por seu lado, a cultura dos ambientes evoluiu também, reforçando o suporte a todas as fases de projecto do sistema e sacrificando em parte as suas ambições de integração.

Tabela 5.5 - As duas culturas CASE [66]

Ambientes de Desenvolvimento	**Ferramentas Baseadas em Diagramas**
Interlisp, Smalltalk	Yourdon, Gane/Sarson, Jackson
Orientados à Arquitectura e à Integração	Ferramentas Isoladas
Originários na Cultura Técnica	Originárias na Cultura Administrativa
"Back-End" CASE (implementação, teste & verificação e manutenção)	"Front-End" CASE (análise de requisitos, especificação e projecto)
Ciclo curto edição-compilação-correcção	Maior investimento na fase de especificação e projecto
Realização de Protótipos	Modelo "Cascata"

Caracterização da Estrutura dos Sistemas de Informação em Ambiente Industrial

Os sistemas de informação no ambiente industrial são sistemas inerentemente distribuídos, constituídos por um conjunto de componentes executando em paralelo e comunicando (para sincronização e / ou troca de dados) através da passagem de mensagens ou partilha de informação estática. A descrição, construção e evolução destes sistemas é facilitada pela decomposição da sua estrutura num conjunto de componentes e das suas interligações, separando-a da descrição individual da estrutura de cada componente. Além disso, a reutilização dos componentes e a flexibilidade da estrutura são favorecidas se os componentes forem independentes do contexto [106].

A caracterização da estrutura do sistema distribuído, bem como a identificação dos seus componentes, pode ser realizada através de duas abordagens diferentes: a decomposição funcional e a decomposição orientada ao objecto [106]. A decomposição funcional tem como base representações do tipo dos diagramas de fluxos de informação (DFD), que descrevem o sistema através de uma configuração funcional de processos executando um conjunto de transformações sobre dados e comunicando através de fluxos de informação. A complexidade do sistema pode desta forma ser decomposta funcionalmente nos seus processos constituintes e nos fluxos de dados, sendo o sistema descrito como um conjunto hierarquizado de processos interligados. A abordagem orientada ao objecto baseia-se na identificação de objectos correspondentes a entidades reais, definindo o estado de cada objecto e a sua interacção com os outros objectos. Existem diferentes formas de abordar este problema, as quais permitem o suporte da definição de hierarquias de objectos de acordo com as suas propriedades, com a definição de classes que descrevem essas

propriedades gerais e de subclasses descrevendo a especialização de algumas propriedades. Esta hierarquia de herança, não tem no entanto qualquer relação com a decomposição funcional usada nos DFDs.

Uma aproximação interessante ao problema é ainda a combinação das duas abordagens anteriores. De facto, a modelação por objectos é apropriada para a especificação de componentes cuja estrutura coincide com a de uma aplicação, podendo a análise funcional ser usada como forma de decomposição de um sistema complexo nesses múltiplos componentes. Numa tal abordagem, é da maior importância que os componentes sejam independentes do seu contexto, não existindo nunca referência a componentes externos além da identificação da origem e do destino das mensagens trocadas. Assim sendo, é reduzido o acoplamento entre os componentes, e a informação é sempre local e não distribuída por diversos componentes.

Modelos / Linguagens de Especificação Formal e Simulação do Sistema de Informação

O desenvolvimento de sistemas de informação de grande dimensão não é simples, não só pela elevada complexidade que normalmente encerram mas também pela maior dificuldade de validação e teste. As técnicas de especificação formal permitem algum controlo sobre estes dois factores. A utilização de um conjunto de abstracções permite com efeito a redução da complexidade do problema, sendo ainda a incidência de erros reduzida através da utilização de ferramentas de projecto assistido por computador baseadas em métodos rigorosos.

As especificações informais não são adequadas por serem muitas vezes ambíguas, requerendo a comunicação directa entre os projectistas e os programadores para possível clarificação. Nos grandes projectos, este esforço conduz ao consumo desnecessário de recursos, e a problemas complexos quando o projectista responsável pela abstracção não está presente para ser consultado. Tal tipo de problemas é particularmente comum na fase de manutenção do sistema.

A especificação formal de abstracções constitui por outro lado um valioso documento de projecto porque [107]: suporta o raciocínio formal e o seu processamento automático; permite o acordo explícito entre o cliente, o projectista e os programadores relativamente ao comportamento esperado do sistema; protege os programadores que trabalham ao nível dos detalhes de implementação; e isola os utilizadores / clientes dos detalhes de programação irrelevantes.

Além disso, o projecto assistido por computador permite a detecção de erros de projecto através da simples utilização de uma linguagem com sintaxe e semântica bem definidas. Estas ferramentas verificam ainda a consistência interna da especificação, e podem ser usadas no teste de conformidade da implementação com a especificação.

Sendo reconhecido que a chave de sucesso de um sistema está na sua especificação e no seu projecto, este facto requer a existência de uma linguagem de especificação satisfazendo os requisitos seguintes [107]: ter definido um conjunto bem definido de conceitos; não ser ambígua, e ser clara, precisa e concisa; ser uma base para verificar se a especificação está completa e correcta; ser uma base para determinar se a implementação está conforme com a especificação; ser uma base para a verificação da consistência entre aplicações; utilizar ferramentas computacionais para criar, manter, analisar, simular e traduzir especificações. Estas linguagens oferecem frequentemente vantagens adicionais, tais como a utilização da especificação formal para a geração de código executável.

A linguagens de especificação formal têm vindo progressivamente a ser consideradas essenciais no desenvolvimento de sistemas complexos, tais como sistema de comunicações

com múltiplas camadas de protocolos e sistemas distribuídos. A formalização é ainda imprescindível à verificação matemática, uma restrição imposta ao desenvolvimento de sistemas envolvendo a segurança de pessoas e bens (os denominados "Safety-Critical Systems").

Na área das telecomunicações, estas linguagens de especificação formal atingiram alguma maturação, tendo já algumas delas estatuto de standards internacionais. O desenvolvimento de aplicações de telecomunicações coloca problemas com características muito especiais, dado serem aplicações distribuídas de grande dimensão executando em hardware heterogéneo. Os requisitos de rapidez de resposta obrigam ao paralelismo, e o grau de fiabilidade exigido é muito elevado. São sistemas com uma vida média muito longa (30-40 anos) devendo suportar um elevado número de variantes, colocando assim especiais exigências na sua adaptabilidade, capacidade de manutenção e de extensão. Por outro lado, o desenvolvimento e manutenção das aplicações de telecomunicações é em si mesmo um problema específico, sendo o desenvolvimento de projectos muitas vezes distribuído por vários locais em diferentes países e em várias empresas. Isto significa que têm que ser resolvidos e superados problemas relacionados com aspectos organizacionais, culturais e de métodos de trabalho. A manutenção está também normalmente distribuída por vários países, e durante a operação o sistema interage com outros desenvolvidos por outras empresas.

A crescente utilização de sistema distribuídos e a aceitação generalizada do modelo de referência OSI e dos seus protocolos standard, levaram a que na última década fossem levados a cabo trabalhos de investigação nas áreas da engenharia de protocolos e de técnicas de descrição formal (TDF) [108]. Em simultâneo com estas actividades de investigação em TDF, foram realizados esforços no âmbito da ISO e do CCITT a partir da década de 70 para o desenvolvimento de TDF standard. Após aproximadamente uma década, três TDF complementares coexistem com o seu estatuto de standards internacionais (IS) ou de recomendações: Estelle [91] (IS9074), LOTOS [92] (IS8807) definidos pela ISO, e o SDL (Z100) pelo CCITT[1] [24,109].

Conclusão

A utilização de uma ferramenta CASE clássica, apoiada por exemplo em DFDs, não permite normalmente a simulação e teste de sistemas excepto através da geração de código. A utilização de uma linguagem de especificação formal torna possível a análise e simulação de soluções alternativas, atitude na prática inviável aquando da simples utilização de uma linguagem de programação, face aos custos e tempos de desenvolvimento muito superiores. A linguagem de especificação formal oferece ainda ao seu utilizador um conjunto de conceitos bem definidos, contribuindo para a sua capacidade de produzir uma solução e raciocinar sobre ela.

5.3.7 A Modelação Orientada ao Objecto aplicada aos Sistemas de Fabrico

Alguns dos Conceitos da Programação Orientada ao Objecto

Os conceitos da programação orientada ao objecto (POO) tiveram um grande impacto na forma como o software é projectado e construído [67,110]. Segundo Meyer: "... o projecto orientado ao objecto pode ser definido como uma técnica que, ao contrário do projecto

[1]Agora denominada ITU ("International Telecomunications Union")

clássico baseado na decomposição funcional, suporta a decomposição do sistema de software em classes de objectos manipulados pelo sistema e não nas funções por ele executadas". A principal vantagem decorre de os objectos se manterem mais ou menos estáveis ao longo do tempo, ao contrário das funções que deverão evoluir para se adaptarem às novas necessidades. Do ponto de vista da simulação o paradigma orientado ao objecto tem uma outra vantagem, que decorre do facto da maior parte do código desenvolvido na construção de novos modelos ser de uso geral, uma vez que a definição dos objectos deverá ser independente das funções do sistema particular a ser modelado.

A POO envolve quatro conceitos chave, que resultam numa maior facilidade em construir software mais compreensível, alterável e reutilizável. Esses conceitos são: o encapsulamento, a passagem de mensagens, a herança e a *ligação tardia* ("late binding") O encapsulamento significa que os dados e os procedimentos de um objecto são agrupados no interior de uma fronteira não penetrável pelos outros objectos, só sendo acessíveis através de um conjunto de procedimentos definidos como parte da classe a que o objecto pertence. A passagem de mensagens é resultado imediato do encapsulamento, sendo o envio de uma mensagem a única forma de um objecto pedir a um outro a execução de um dos seus procedimentos. A herança permite uma reutilização de código de baixo nível, através da organização das classes segundo uma determinada hierarquia. A herança permite a construção de uma nova classe de objectos partindo de uma classe já existente, podendo as suas funcionalidades ser estendidas, reduzidas ou mesmo modificadas. A *ligação* ("binding") estabelece a relação entre os procedimentos e os dados por eles tratados. Ao contrário das linguagens tradicionais caracterizadas pela *ligação antecipada* ("early binding"), i.e. no momento da escrita do código / compilação é estabelecida a *ligação* entre os dados e procedimentos, as linguagens POO deixam que esta *ligação* se faça apenas no momento em que o software está em execução sendo então denominada *ligação tardia*. As diferenças existentes entre o desenvolvimento de software com uma linguagem baseada em chamadas a procedimentos e uma linguagem POO resultam pois das quatro características definidas acima. Estes conceitos envolvidos na programação orientada ao objecto podem ser estendidos à modelação para simulação e controlo de sistemas de fabrico, conforme se discutirá a seguir.

A Modelação do Sistema de Fabrico, O Conceito de Bloco Construtivo

O desempenho de um sistema de fabrico é altamente influenciado pelas estratégias de controlo usadas na sua operação. Na avaliação desse desempenho, é necessário não só considerar os componentes físicos envolvidos e a sua configuração mas também diferentes estratégias de operação. Por outro lado, existe na instalação fabril uma complexa estrutura hierárquica de decisão. As decisões são tomadas com base na informação disponível, que é muitas vezes incompleta, não exacta e atrasada. Assim, o decisor deve, em cada nível, usar um conjunto de heurísticas, a sua experiência profissional, regras da empresa e estratégias para chegar à decisão possível. As ferramentas de modelação tradicionais não disponibilizam porém o conjunto de estruturas necessárias para representar todas estas interacções, uma vez que os elementos de controlo estão muitas vezes embebidos no código modelando os componentes físicos a que respeitam.

Existem duas razões para que o modelador deva incorporar estruturas de processamento de informação e de tomada de decisão de uma forma explícita e separada [93]: facilitar uma maior reutilização do modelo e conseguir uma forma mais natural de construção do mesmo. As linguagens tradicionais não fornecem blocos construtivos que permitam a modelação distinta de elementos físicos e fluxo de materiais, fluxo de

informação e fluxo de decisões de controlo. Além disso, os elementos responsáveis pela informação e pelo controlo estão codificados e dispersos pelos diversos componentes do modelo. Por este motivo, não é facilitada a modificação nem a reutilização de modelos.

O projecto para a reutilização envolve a identificação dos comportamentos que devem ser evidenciados pelos objectos em múltiplos contextos [93]. Se um bloco construtivo do modelo executa mais do que uma das funções básicas (i.e. física, informação e ou controlo / decisão) então a sua utilização limita-se a situações em que as três funções sejam necessárias em simultâneo. Por outro lado, se for mantida uma relação um-para-um entre a funcionalidade do componente e a função por ele desempenhada, os componentes tornam-se de facto blocos construtivos básicos que podem ser combinados na construção de modelos complexos. Falta então saber como separar essas funcionalidades no contexto de um ambiente fabril.

Um sistema de fabrico pode ser completamente definido / modelado através de três componentes básicos: os objectos físicos, os de informação e os de controlo. Um *objecto físico* é um objecto com um correspondente tangível no sistema real. Um objecto pode então ser considerado físico se o objectivo primário do interesse do modelador nesse objecto forem as suas características físicas, como é o caso com matérias primas ou componentes, máquinas, transportadores, produtos, etc., mantidos no sistema informático. Um *objecto* será classificado como *de informação* se o interesse do modelador nesse objecto for o seu conteúdo de informação. Um objecto de informação é um objecto que pode não ser tangível no mundo real, por exemplo uma lista de materiais, informação de encaminhamento, etc. O *objecto de controlo / decisão* é um objecto lógico que tipicamente não tem correspondente tangível no mundo real. Estaremos face a um objecto de controlo, se a sua função principal for: (1) avaliar o estado de um sistema; (2) executar a lógica de um algoritmo; (3) assinalar uma determinada acção a ser executada.

O *fluxo de materiais* resulta da modelação dos objectos físicos de processamento, armazenamento e transporte, assim como da sequência de operações que eles realizam sobre os objectos físicos como matérias primas e componentes. A estrutura do modelo reflecte pois o "layout" fabril, as gamas operatórias e os encaminhamentos possíveis.

O *fluxo de informação* é modelado numa estrutura hierárquica; por exemplo as encomendas de um produto dão origem a pedidos aos armazéns, ordens de fabrico de componentes e finalmente ordens de execução. De um modo semelhante, o *processo de decisão* é modelado através de uma hierarquia de controlo: controlo ao nível da fábrica / unidade de produção, controlo ao nível da instalação fabril ("shop floor"), e controlo ao nível da célula de trabalho. Nestes níveis os processos de controlo / decisão são suportados por heurísticas, algoritmos de investigação operacional ou mesmo sistemas periciais, surgindo deste modo uma interacção natural entre os fluxos de materiais, e de informação / controlo na instalação fabril.

Uma outra vantagem inerente à separação dos objectos físicos, de informação e de controlo é o facto de tal permitir ao modelador pensar separadamente em cada um deles durante o desenvolvimento do modelo. Durante a construção do modelo físico, poderá pois abstrair os aspectos de informação e de controlo. Esta independência facilita a criação de modelos integrados mais realistas, e ao mesmo tempo com um maior nível de complexidade e flexibilidade.

Uma ferramenta de modelação orientada ao objecto (MOO), fazendo uso dos conceitos referidos em relação à POO, deverá fomentar a reutilização fornecendo ao seu utilizador uma base de dados de modelos. Nesta base de dados devem evidentemente estar presentes as primitivas de modelação, i.e. os blocos construtivos básicos ou primitivas físicas, de

informação e de controlo. Um objecto será uma primitiva para efeitos de modelação e simulação se não for necessária a sua decomposição num objecto mais simples.

O blocos construtivos necessários à modelação de uma empresa fabril incluem elementos genéricos normalmente encontrados no ambiente fabril, tais como máquinas, veículos de transporte de material, passadeiras transportadoras, ordens de trabalho, informação de encaminhamento, etc. A MOO pode ser aqui utilizada, por exemplo, na modelação da classe "transportador" para que as facilidades de herança sejam aproveitadas na criação de subclasses para a modelação dos vários tipos de "transportadores". Os blocos construtivos fornecem assim um nível de abstracção superior ao usado nas linguagens de simulação tradicionais, fomentando além disso a construção de modelos parciais representativos de um determinado sector industrial e de tipos específicos de organização fabril.

A utilização de primitivas genéricas fornece um ambiente de modelação mais abrangente do ponto de vista do tipo de sistemas que pode vir a ser modelado. No entanto, para que seja tirada toda a vantagem na utilização da MOO, a empresa deve procurar derivar dos modelos genéricos e parciais as suas representações particulares. Existem assim várias áreas no contexto fabril onde a MOO aplicada à simulação pode ser aplicada com vantagem, tais como o apoio a: decisões estratégicas de longo prazo (por exemplo, projecto do sistema de produção), decisões tácticas de curto prazo (por exemplo, controlo e planeamento da produção), ou ainda decisões ao nível operacional (por exemplo, escalonamento fino e reactivo).

São as seguintes as vantagens da Modelação Orientada ao Objecto aplicada aos Sistemas de Produção:

- Reutilização de Modelos — Esta vantagem advém directamente das vantagens da programação orientada ao objecto, tais como encapsulamento, herança, passagem de mensagens e *ligação tardia*.
- Múltiplos Níveis de Abstracção — Uma vez que o objectivo do estudo de simulação muda, é inevitável a utilização de múltiplos níveis de abstracção. Deste modo, o utilizador pode se o desejar simular toda a instalação fabril, uma secção ou departamento, uma célula de fabrico ou apenas um posto de trabalho. O nível de detalhe ou granularidade do modelo depende também do objectivo do estudo, e é ainda possível construir modelos heterogéneos compostos de várias partes com diferentes níveis de detalhe.
- Estrutura de Trabalho Comum — O utilizador pode usar um único ambiente para a modelação do sistema, recorrendo depois a múltiplas ferramentas de análise, tais como redes de filas de espera, redes de Petri, algoritmos genéticos, etc.

As limitações da Modelação Orientada ao Objecto aplicada aos Sistemas de Produção são, por exemplo:

- Linguagem — Os modelos orientados ao objecto são desenvolvidos usando POO, herdando de imediato todas as suas desvantagens: processamento lento e exigências pesadas relativamente a recursos computacionais.
- Disponibilidade de Bibliotecas — A real produtividade só irá certamente emergir quando estiverem disponíveis bibliotecas de modelos genéricos (i.e. as primitivas básicas de modelação), e de modelos parciais (representativos de sectores industriais específicos, como por exemplo electrónica, calçado, etc.).

5.4 Modelos Executáveis, Exemplos

5.4.1 Introdução

Na sequência do enquadramento criado nos pontos anteriores, serão agora abordados dois ambientes de integração em que é suportado o desenvolvimento de modelos fabris executáveis, o CIMOSA e o PROFIT. A escolha do CIMOSA foi já justificada pela importância da sua contribuição ao nível de diversos grupos de normalização, assim como pela sua abordagem inovadora relativamente aos modelos empresariais executáveis. Concluída uma breve apresentação do ambiente de modelação do CIMOSA, será apresentado o PROFIT, escolha esta motivada não só pelo envolvimento directo do autor mas também pela sua relevância como ferramenta avançada para construção de soluções ao nível da execução fabril.

5.4.2 A Modelação Empresarial na Arquitectura CIMOSA [23,34,35,111]

Nos anteriores capítulos 3 e 4, a CIMOSA foi já abordada tanto no âmbito das arquitecturas como das infra-estruturas de integração. Neste ponto, o tema da arquitectura CIMOSA é retomado do ponto de vista dos modelos executáveis, apresentando-se os conceitos básicos subjacentes à sua construção e utilização.

Modelação Baseada em Processos, Controlada por Eventos

Seja S o sistema de fabrico discreto que pode ser especificado através do par $S:=\{O_S,A_S\}$, onde O_S representa o conjunto de classes de objectos e A_S o conjunto de classes de actividades. Os objectos da empresa que são de facto as entidades da empresa, são definidos pela sua classe (i.e. estrutura), pelo seu estado (i.e. valores da ocorrência), e caracterizados por um identificador único.

A CIMOSA especifica o sistema discreto de fabrico como um 5-tuplo $S:=\{E_S, A_S, P_S, R_S, V_S\}$, representando cada um dos elementos um conjunto finito de classes: de eventos (E_S), de actividades (A_S), de processos (P_S), de recursos (R_S), e de vistas sobre objectos (V_S).

Nos parágrafos que se seguem, serão apresentados de uma forma breve os principais blocos de modelação da CIMOSA, as Vistas de Objectos, os Recursos, as Operações Funcionais, os Eventos, as Actividades da Empresa e finalmente os Processos. São ainda apresentadas as propriedades dos modelos da CIMOSA, e as facilidades de que dispõe para a análise do desempenho. Uma vez apresentados os blocos de modelação é exposta sucintamente a utilização desses blocos construtivos nas actividades de modelação para a definição de requisitos, a especificação do projecto, e a descrição da implementação. Esta explanação é concluída abordando o tema da execução do modelo desenvolvido sobre a infra-estrutura de integração, assim como a problemática do seu suporte computacional.

Os Principais Blocos de Modelação CIMOSA

Vistas de Objectos

Os objectos de materiais e de informação usados como entradas de controlo, entradas de funções e/ou saídas de funções de pelo menos uma actividade da empresa, são descritos através de vistas de objectos. Uma vista de um objecto é uma representação ou

manifestação física de um objecto perceptível pelo utilizador ou pelas aplicações, sendo descrita por um conjunto de propriedades. Assim sendo, a natureza do objecto poderá ser "física" se a vista representa objectos físicos (materiais, ferramentas, etc.) ou "informação" se a vista de objecto representa informação (formulários, écrans de computador, relatórios, ficheiros, mensagens, etc.).

Recursos

Os recursos representam qualquer tipo de meio físico da empresa usado na execução de tarefas (máquinas, ferramentas, sistemas de manuseamento de materiais, computadores, sistemas de gestão de bases de dados, etc.), assim como aplicações de software (sistemas de CAD, CAPP, MRP, etc.) e seres humanos, sendo o conjunto de classes de recursos R_S tal que $R_S \subseteq O_S$.

Os recursos activos oferecendo um conjunto finito de capacidades e aptos a realizarem um conjunto definido das chamadas *Operações Funcionais* formam uma classe de interesse especial para o processo de modelação da empresa. Os elementos desta classe são chamados *Entidades Funcionais* ou *Agentes*. A classe R de entidades funcionais de R_S é então caracterizada: pela definição da vista do objecto fornecendo um conjunto de propriedades descritivas (variáveis) explicitando o estado de uma classe de recursos R; pelo conjunto finito de capacidades oferecidas pelos recursos da classe R; pelo conjunto de operações funcionais (comandos básicos) que os recursos da classe R são capazes de compreender e de executar, e por uma tabela (opcional) indicando para um dado recurso de R quando e por quanto tempo ele estará afectado a actividades específicas (escalonamento de recursos).

Operações Funcionais

As operações funcionais são átomos de funcionalidade ou unidades de trabalho definidas no modelo. São executadas pelas entidades funcionais (máquinas, aplicações ou humanos), e só são utilizadas pelas actividades. São equivalentes a comandos primitivos e requerem normalmente parâmetros para passar / retornar dados ou mensagens (por exemplo, furar, mover uma peça, emitir uma lista, etc.) São equivalentes a métodos enviados a agentes no âmbito da inteligência artificial distribuída. São definidos na forma canónica como *FE.FO(lista-de-parâmetros)*, onde FO é o nome da operação funcional, FE o nome da entidade funcional que a vai executar, e a lista de parâmetros (que pode ser uma lista vazia) contém os argumentos de entrada e de saída. Um dos argumentos da lista de parâmetros pode ser utilizado para o retorno de um código após a execução da operação.

Eventos

Os eventos são acontecimentos não solicitados (sinais ou falhas) ou solicitados (pedidos, ordens, etc.) condicionando a execução das operações da empresa, (i.e. os processos de e as actividades do negócio). A classe evento E de E_S é definida como $E := (E_{id}, q, VO, t)$ onde E_{id} é o nome da classe evento, q é um predicado lógico de primeira ordem, VO a classe vista do objecto (opcional) definindo a informação transportada por esta classe de eventos, se existir, e t o instante no tempo para um evento ocorrência desta classe E em que ele aconteceu. O predicado lógico q define a condição de disparo do evento em função das propriedades de VO e das variáveis de estado do sistema. Uma ocorrência da classe evento é criada sempre que a condição q for verdadeira.

Actividade da Empresa

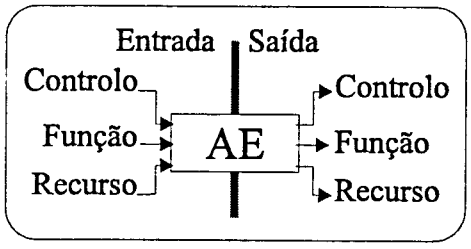


Figura 5.6 - Actividade da Empresa

As actividades da empresa, ou simplesmente actividades, representam as coisas que têm que ser realizadas, i.e. as tarefas elementares da empresa a levar a cabo usando os seus recursos. Seja A uma classe actividade de A_S, então ela é definida como:

$$A = (A_{id}, FI_A, FO_A, CI_A, CO_A, RI_A, RO_A, \delta_A, S_A)$$

onde A_{id} é o nome da classe actividade e FI_A, FO_A, CI_A, CO_A, RI_A e RO_A, são funções de entrada e de saída, controlo de entrada e de saída, e recurso de entrada e saída (figura 5.6) respectivamente (sendo que $FI_A \cup FO_A \cup CI_A \neq \emptyset$; $FI_A \cap CI_A = \emptyset$; FI_A, FO_A, CI_A, $RO_A \subseteq V_S$; $RI_A \subseteq R_S$; $CO_A \subseteq E_S$), δ_A define o comportamento da actividade, i.e. a função transformação executada pelas ocorrências de A (normalmente definida sob a forma de um algoritmo para as máquinas, ou um guião / lista de instruções para o operador humano). A função δ é tal que δ_A (FI_A, CI_A, RI_A) = (FO_A, CO_A, RO_A), e S_A é o conjunto finito de estados finais $S_A = \{s_1, s_2, ...s_{ni}\}$, $1 \leq i \leq Card(A_S)$, $ni \in N$. Os estados finais são predicados sem argumentos, indicando o estado de conclusão da actividade. Os seus valores possíveis são declarados pelo utilizador na fase de projecto, valores esses que deverão ser retornados pelas actividades no decorrer da sua execução, sendo obrigatória a sua definição no modelo. As funções de entrada e de saída fornecem respectivamente listas de vistas de objectos consumidos / transformados e produzidos pelas actividades. A entrada de controlo indica as vistas de objectos usadas pela actividade e não modificadas (informação de controlo). A saída de controlo providencia uma lista de eventos que podem ser gerados por uma actividade. A entrada de recursos indica quais as entidades funcionais a serem usadas na execução das operações funcionais da actividade. A saída de recursos fornece dados sobre o estado do recurso após a execução da actividade (opcional).

A função ES ("End State") é definida como pertencendo ao modelo, e retorna o valor do estado final da actividade: $ES(a) \in S_A$, $\forall a \in A$. Esta função é utilizada na descrição do comportamento do processo.

Processos

Os processos da empresa descrevem o seu comportamento, i.e. a ordem pela qual as actividades são encadeadas e executadas. Seja P uma classe de P_S definida como:

$$P = (P_{id}, \alpha_P, \beta_P, \delta_P, S_P)$$

onde P_{id} é o nome da classe processo, α_P é chamado o alfabeto de P e representa os passos (i.e. actividades) em que as ocorrências de P podem estar envolvidas, β_P é o conjunto de condições de disparo que levam a que o processo P inicie ($\beta_P = \{e: e \rightarrow P\}$), δ_P é o conjunto de regras de procedimento que definem o comportamento do processo, e S_P é o conjunto finito dos estados de conclusão dos processos P tais que $SP = \{s'_1, s'_2, ..., s'_{mj}\}$, $1 \leq j \leq$

Card(P_S), mj∈N. Os estados de conclusão são predicados sem argumentos indicando o estado de conclusão do processo.

A definição de processos é inspirada nos formalismos propostos por Milner e Hoare para a comunicação entre processos [112,113]. As principais diferenças que traduzem as especificidades dos processos de fabrico residem na definição de eventos, na utilização de estados de conclusão dos processos e nas regras de procedimento para o comportamento dos processos. Assim sendo, o formalismo sobre o qual assenta o CIMOSA é bastante mais rico do que uma máquina de estados finita [24,34].

O comportamento do processo P, α_P é definido por um conjunto finito de regras de comportamento da forma:

WHEN (*condição*) DO *acção*

onde a *condição* é a condição de disparo da regra representando um estado possível do sistema, o qual é definido como uma combinação lógica de predicados dependentes de eventos e de estados de conclusão. A *acção* é constituída por uma acção, processo ou actividade, um lançamento de acções em paralelo ou uma escolha de acções.

Algumas Propriedades do Modelo

Os processos de mais alto nível, ou processos mestre são chamados Processos do Domínio (PD), sendo só disparados por eventos, i.e., o seu β_P é tal que $\beta_P \subseteq E_S$. Por outras palavras, na sintaxe das suas condições de disparo não podem fazer uso das funções ES("End-state"). Cada PD deve ter um resultado final bem definido sob a forma de conjuntos de vistas de objectos.

Os PD são constituídos por sub-processos e / ou actividades, chamados Processos do Negócio (PN, "Business Processes"), e Actividades da Empresa (AE, "Enterprise Activities"). Os PN e as AE têm pelo menos um estado de conclusão caracterizado usando a função ES. Assim sendo, as classes de PN e AE podem ser reutilizadas livremente em todo o modelo.

Um evento pode ser utilizado para disparar um ou mais processos. O inverso também é verdadeiro, podendo uma acção (processo ou actividade) ser disparada por um ou mais eventos. Exemplos disso podem ser as regras:

WHEN (*condição*) DO P1 ‖ P2 ‖ P3...

WHEN (ES(P1) = s_1 AND ES(P2) = s_2 ...) DO Pn

A comunicação entre as actividades da empresa pertencentes a um dado domínio pode realizar-se de um modo síncrono ou assíncrono ao nível da operação funcional usando as operações funcionais pré-definidas:

send (a, mensagem, ok)

receive (a, mensagem, ok)

acknowledge (a)

onde 'a' é a ocorrência de uma actividade, a 'mensagem' é a mensagem enviada ou recebida, e 'ok' o código de estado. Não é necessária a indicação de qualquer entidade funcional para a execução destas tarefas, umas vez que elas são executadas de forma transparente pela infra-estrutura de integração.

O definição do comportamento do processo deve acabar sempre com o estado FINISH, i.e. as regras executadas no final devem ter FINISH como única acção. O estado final (de conclusão) de cada processo é obtido como combinação lógica dos estados finais das últimas actividades executadas nos processos.

Análise do Comportamento e do Desempenho

O fluxo de controlo e de materiais pode ser extraído do modelo integrado e convertido em redes de Petri, podendo depois ser estudadas as propriedades comportamentais do modelo.

A análise de desempenho de algumas partes do modelo pode também ser realizada usando Redes de Petri Temporizadas ("Timed Petri Nets") e simulação, sendo neste caso atribuída a cada actividade uma duração média e a cada condição de conclusão uma dada probabilidade. Os tempos de ciclo / processamento médios podem então ser calculados, assim como os seus limites superiores usando a análise do "caso mais desfavorável". Se desejarmos levar em consideração a disponibilidade de um recurso, também devem ser consideradas a sua taxa de avaria e o seu tempo médio de reparação. Análises deste tipo podem ser realizadas usando tanto ferramentas baseadas em Redes de Petri como ferramentas de simulação.

Modelação na Definição de Requisitos

Objectivos

O objectivo da modelação dos requisitos é a captura das necessidades essenciais do sistema integrado de fabrico a ser projectado ou re-projectado, i.e. deve ser caracterizado com a maior exactidão possível aquilo que tem que ser feito e possivelmente como pode ser feito [23]. Para tornar esta tarefa possível, a linguagem de definição de requisitos foi definida por forma a ser compreensível e utilizável pelos vários utilizadores finais da empresa.

A este nível da modelação devem ser identificados e / ou descritos os diversos processos e actividades que constituem o sistema (o comportamento do processo), assim como as condições de activação para cada processo (i.e. a identificação dos eventos relevantes). Além disso, devem ser definidas as vistas dos objectos envolvidas nas entradas e saídas das actividades e indicada a sua natureza. A definição do comportamento é no entanto deixada para a fase de projecto. É também assumido que os recursos não são ainda conhecidos, sendo a sua definição também uma tarefa de projecto. Os requisitos dos recursos são definidos através das capacidades necessárias associadas à definição de cada actividade, e exprimindo as restrições técnicas da utilização dos recursos (por exemplo, o manuseamento / transporte pode exigir o movimento de peças com um determinado peso médio a uma dada velocidade ao longo de uma certa distância).

Blocos Construtivos

A CIMOSA providencia o bloco de nível mais elevado, o Domínio, usado para decompor a complexidade do sistema. A empresa pode assim ser dividida em diversos Domínios inter-relacionados, podendo alguns deles não ser domínios CIMOSA (representando, por exemplo, sistemas legados). As interacções entre domínios são representadas por Relações de Domínio, indicando a troca de eventos e de vistas de objectos entre domínios.

Ao nível da definição de requisitos podem ainda ser usados outros blocos, tais como: eventos, vistas de objectos, processos do domínio, processos do negócio, actividades da empresa, capacidades necessárias e requisitos organizacionais. As actividades da empresa

podem neste ponto ter as suas capacidades definidas, não sendo ainda realizada a definição do comportamento da actividade e da entrada de recursos.

Modelação na Especificação do Projecto

Objectivos

O objectivo da especificação do projecto é a definição de uma "boa" solução, realizável satisfazendo o conjunto de requisitos recolhidos na fase de definição anterior. O modelo deve nesta fase ser detalhado, analisado (qualitativa e quantitativamente), e validado. A simulação por computador pode ser a este nível utilizada com vantagem na selecção de uma entre várias soluções alternativas. É nesta fase de especificação que devem ser introduzidas as questões relacionadas com tempos e ocorrências, sendo também definidos os tipos de recursos.

Blocos Construtivos

Nesta fase da modelação devem ser completadas as descrições dos comportamentos dos processos e das actividades da empresa. Para cada actividade da empresa, devem ser definidas as operações funcionais requeridas bem como os recursos a serem utilizados. As vistas dos objectos são todas analisadas com o objectivo de derivar o modelo conceptual suportando o sistema de informação, usando para a análise o modelo entidade associação ou o modelo orientado ao objecto, e sendo depois especificados os modelos externos. Aos eventos é adicionada uma etiqueta temporal e às actividades são atribuídos tempos máximos e mínimos de duração para efeitos de simulação.

O blocos construtivos utilizados nesta fase de especificação são de facto os mesmos blocos usados na anterior fase de definição de requisitos, sendo aqui expandidos por forma a abranger eventos, vistas de objectos, processos do domínio, processos de negócio e actividades da empresa. Além disso são usados blocos adicionais, tais como recursos (incluindo entidades funcionais), operações funcionais, modelos de dados conceptual e externo (usando ferramentas de análise de sistemas de informação para especificar as vistas dos objectos). São ainda usados blocos denominados unidades organizacionais para a descrição das estruturas organizacionais (níveis de decisão e centros de decisão) e das responsabilidades e autorizações sobre outros blocos construtivos.

Modelação na Descrição da Implementação

Objectivos

O objectivo da modelação na fase de implementação é a conclusão do modelo obtido na fase de especificação, em termos da identificação das entidades funcionais a serem usadas (número e localização). São ainda identificados os seus requisitos do ponto de vista de serviços de ligação e de apresentação (através da definição dos portos ou entradas/saídas do sistema), e definidas as entidades de informação em termos de estruturas relacionais usando SQL. Além disso, são caracterizados os procedimentos de tratamento de excepções a serem definidos, quer na execução de processos e de actividades quer na gestão de recursos.

Blocos Construtivos

O conjunto de blocos construtivos usado é o mesmo da fase anterior, sendo adicionada na descrição da implementação das estruturas de informação a definição dos modelos internos (usando a linguagem SQL). Os modelos externos, implementando as vistas sobre objectos, são assim descritos como vistas SQL. O bloco Domínio é estendido por forma a

acomodar as definições dos portos de entrada / saída requeridos pelo sistema, dos canais para a troca de mensagens entre actividades, bem como das variáveis globais usadas pelo comportamento das actividades. As actividades da empresa têm as suas entradas e saídas ligadas aos portos do sistema de forma a transferir a informação contida nos eventos e nas vistas de objectos. Os mecanismos de tratamento de excepções, tais como temporizações e procedimentos de excepção, são adicionados aos processos relevantes e às descrições das actividades.

A Linguagem de Implementação AID

A versão final do modelo da empresa deve ser verificada de forma automática, quanto à sua consistência e quanto ao facto de estar ou não completa. Isto só pode no entanto ser conseguido se o modelo estiver expresso numa linguagem formal tal como TIE ("Take it Easy") [114]. Além disso, deve ser assegurado: (1) o transporte do modelo de uma ferramenta CIMOSA para uma outra, (2) serviços similares para a execução do modelo oferecidos por diferentes vendedores de sistemas. Assim sendo, existe uma necessidade de uma linguagem formal com uma sintaxe e semântica precisas, para expressar o modelo de uma forma neutra, sendo este o objectivo a ser atingido pela utilização da linguagem "AMICE Implementation Description" (AID).

A AID é uma linguagem descritiva usada para exprimir o modelo da empresa através dos blocos construtivos CIMOSA. A sua estrutura tem uma sintaxe semelhante ao Pascal, apresentando declarações e chamadas a procedimentos, incorporando também extensões que permitem a declaração de mecanismos para o tratamento de excepções.

Está planeado o desenvolvimento de compiladores AID que de uma forma automática possam gerar o código executável para cada processo do domínio do modelo partindo do código fonte em AID [34].

A Execução do Modelo sobre a Infra-estrutura de Integração

O papel da infra-estrutura tem dois aspectos. Por um lado deve fornecer o suporte transparente à integração do sistema de fabrico. Por outro lado, deve providenciar o suporte para a coordenação da empresa, i.e. assegurando uma execução consistente e concorrente dos processos do domínio. Estes objectivos são atingidos através da utilização de um conjunto de serviços distribuídos fornecidos através de um conjunto de entidades, as entidades do negócio, de informação e de apresentação, assim como as entidades de serviços comuns e de gestão do sistema (assunto tratado nos capítulos 3 e 4).

Sendo um dos objectivos do CIMOSA a utilização do modelo da empresa para controlar e monitorizar a execução dos seus processos de negócio e operações, a infra-estrutura deverá obviamente suportar a execução do modelo neste contexto.

CIMOSA, Suporte Computacional

A modelação empresarial usando blocos construtivos propostos pela arquitectura CIMOSA não é, como vimos, viável sem um suporte computacional adequado. Durante o projecto ESPRIT 688 AMICE, foi já iniciado o desenvolvimento de raiz do suporte computacional para o CIMOSA. Infelizmente esta tentativa não foi bem sucedida, motivo pelo qual continua sem existir a desejada ferramenta. No capítulo anterior foi realizada uma breve descrição de algumas infra-estruturas de integração de suporte ao CIMOSA, nomeadamente o SEW-OSA / CIM-BIOSYS, o Ωmega, e a infra-estrutura desenvolvida na instalação piloto do projecto AMICE no instituto WZL em Aachen. Além deste casos pontuais, o suporte computacional para o CIMOSA continua um problema em aberto. Existem duas alternativas para a resolução deste problema. Por um lado, poderá investir-se na construção

de uma ferramenta dedicada ao suporte das primitivas de modelação CIMOSA. Embora as vantagens sejam evidentes, ver na tabela 5.6, esta tarefa não é simples e dada a sua complexidade o seu custo não será certamente desprezável. Por outro lado, poderá ser procurada a utilização de uma ferramenta CASE existente realizando a analogia entre suas primitivas de modelação e os blocos construtivos CIMOSA. De facto, embora seja inegável o risco de perder algumas das facetas inerentes à modelação CIMOSA, os custos envolvidos seriam certamente menores. Assim sendo, e levando em consideração a necessidade de facilidades já disponíveis nas ferramentas CASE tais como, por exemplo, a capacidade para verificar a coerência e consistência dos modelos bem como facilidades de simulação para a sua optimização, a segunda hipótese será certamente a mais viável para que mais rapidamente se chegue ao suporte computacional da arquitectura CIMOSA.

Tabela 5.6 - As duas soluções para o suporte computacional do CIMOSA [38]

	Usar uma ferramenta CASE existente, realizando a correlação entre as suas primitivas e as CIMOSA	**Desenvolver uma ferramenta dedicada de suporte às primitivas de modelação CIMOSA**
Desvantagens	Risco de perder as facetas do CIMOSA. Utilização complexa.	Tarefa cara e complexa.
Vantagens	Economia de recursos em termos de tempo, dinheiro e experiência. Realização rápida de um protótipo.	Verdadeiro suporte para todos os aspectos da CIMOSA ao longo de todo o ciclo de vida do sistema. Facilidade de utilização pelo facto de os formalismos da CIMOSA estarem escondidos pelo ambiente fornecido.

A consulta da tabela 5.6 leva todavia a concluir que na adaptação de uma ferramenta CASE ao CIMOSA se corre o risco de perder algumas funcionalidades importantes, além de poderem surgir pontos do ciclo de vida do sistema que não seriam cobertos. Esta solução é no entanto a mais rápida e económica, motivo pelo qual no trabalho em curso no âmbito do projecto AMICE III/P foi seguida esta abordagem. Em [38] é descrita a utilização da ferramenta CASE ARTIFEX [115] no desenvolvimento do ambiente de modelação do CIMOSA. Esta ferramenta inclui Redes de Petri Estendidas ("Extended Petri Nets") para descrição do comportamento, encapsuladas em objectos relacionados que estruturam a decomposição funcional, com diagramas entidade associação que asseguram a representação dos dados, e uma interface gráfica com o utilizador que permite o tratamento da informação.

5.4.3 A Modelação na Planta Fabril usando o SHOP-CONTROL / PROFIT

Introdução

O SHOP-CONTROL / PROFIT constitui um exemplo interessante onde a utilização de modelos para a simulação e para o controlo das operações fabris se encontram. Neste ponto serão abordados os objectivos da modelação, concentrando a atenção no nível de execução fabril. Serão neste contexto focados aspectos mais gerais relacionados com os sistemas de

gestão e controlo das operações fabris, bem como seus objectivos e relevância para a empresa industrial. Finalmente, a apresentação da linguagem de modelação do Simple++ vai ilustrar as funcionalidades do PROFIT disponibilizadas através do modelo fabril.

O nível de execução, também denominado nível oficinal ou do "chão-de-fábrica", de uma empresa industrial é considerado como o coração do sistema integrado de fabrico. O seu desempenho é crucial, pois determina ganhos ou perdas nos parâmetros ou indicadores directamente relacionados com os factores de competitividade, i.e. qualidade, custos, prazos de entrega, flexibilidade, rapidez de resposta, etc.

Os paradigmas modernos de organização dos sistemas de produção, tais como os conceitos de organização fractal ("Fractal Company") [8] ou de melhoria contínua ("Kaizen Engineering") [13], a organização em células ou grupos de trabalho [116], apontam para um maior envolvimento do pessoal a qualquer nível da estrutura hierárquica da empresa, a qual sofre entretanto grandes transformações, particularmente no que respeita à gestão dos processos e à decisão. Procura-se delegar o processo de decisão ao mais baixo nível possível, i.e. envolvendo / partilhando o poder de decisão em áreas críticas, como a qualidade ou o despacho de ordens de fabrico, com os encarregados e operadores possuidores de experiência e informação sobre muitos aspectos inexistente a outros níveis da fábrica ("shop floor empowerment"). Os sistemas de gestão fabril / oficinal, foram neste contexto identificados como um elemento crucial na cadeia de decisão, uma vez que só através deles é possível uma gestão eficiente das actividades ao nível da instalação fabril [58].

Modelação e Abordagens Orientadas à Ferramenta

Até há pouco tempo, a maior parte dos sistemas informáticos de apoio à gestão fabril eram construídos à medida das necessidades do cliente, o que os tornava muito caros e apenas acessíveis a grandes empresas, i.e. soluções específicas desenvolvidas à medida. Por outro lado, os sistemas comerciais que são actualmente vendidos, não o são em quantidades suficientes que os possam tornar suficientemente baratos para serem acessíveis às pequenas e médias empresas, e são pouco flexíveis apresentando dificuldades de adaptação às necessidades organizacionais e aos ambientes de produção específicos (soluções orientadas à aplicação). Esta flexibilidade só será alcançada se for possível a construção de um modelo do processo produtivo específico para cada empresa, modelo esse que suportaria a solução informática, sendo no entanto considerável o esforço de engenharia necessário à modelação de um sistema produtivo complexo. Assim sendo, o processo de modelação deve ser suportado de uma forma eficiente, através da utilização de um conjunto genérico de blocos construtivos que facilitem a tarefa de projecto e construção do modelo. Estes blocos construtivos deverão ser suficientemente genéricos para permitir suportar um conjunto de aplicações diversificado (perspectiva do modelador) e suficientemente específicos para facilitar tanto quanto possível a construção do modelo (perspectiva do cliente). Surgiu assim uma tendência mais recente baseada na utilização de ferramentas de modelação e desenvolvimento de aplicações que permitam modelar e realizar protótipos de soluções específicas de gestão fabril (i.e. soluções orientadas à ferramenta).

Usando esta aproximação orientada à ferramenta, o Sistema de Gestão de Operações Fabris (SGOF) resultante deverá incrementar a capacidade de decisão, ao nível da instalação fabril (gestores da produção, encarregados, operadores) e permitir a realização protótipos e a validação de soluções específicas, adequadas ao tipo de organização, e flexíveis, de forma a poder evoluir com essa mesma organização. Neste contexto, os SGOF devem providenciar a informação relevante, exacta e actualizada sobre o estado actual do

sistema produtivo, i.e. ordens, recursos de fabrico, stocks, etc., sendo este um requisito fundamental para que sejam tomadas as melhores decisões. Além disso, os SGOF estado-da-arte devem suportar os processos de decisão através da filtragem e processamento de informação da instalação fabril, identificando decisões alternativas e comparando-as de acordo com funções objectivo ou critérios configuráveis pelo utilizador. Deverão ainda suportar a gestão optimizada das operações fabris, levando em consideração aspectos temporais, de custos e de qualidade. A gestão do tempo está relacionada com o escalonamento de curto prazo e o despacho das ordens de fabrico, assim como com a monitorização do em-curso-de-fabrico ("work-in-progress tracking"). A gestão da qualidade inclui, entre outras intervenções, o controlo estatístico do processo (SPC), e a gestão de acções correctivas e de não conformidades. A gestão da manutenção inclui o suporte à manutenção correctiva e preventiva, assim como o suporte à gestão das intervenções [117].

Para que estas tarefas sejam realizadas de uma forma eficiente e integrada, o SGOF deverá apoiar-se num modelo realista da instalação fabril. O modelo deverá ser flexível para que se possa adaptar a diferentes cenários de produção incorporando todos os objectos relevantes assim como as suas relações, devendo ser acessível ao utilizador a possibilidade de acomodar alterações ao processo de fabrico [22]. De acordo com o tipo de tarefas a executar, o modelo deve ser capaz de apresentar a "vista" mais apropriada sobre o ambiente a instalação fabril [118]. Uma vez que o SGOF é constituído por um conjunto de várias aplicações de software independentes mas cooperantes, o modelo deverá actuar como elemento aglutinador de interface directo com o utilizador, permitindo um acesso contextual a essas aplicações (este aspecto já foi tratado no capítulo anterior a propósito da aplicação de Supervisão).

Linguagem de Modelação Comum e Vistas Física, Funcional e de Informação

As aplicações fabris actualmente existentes adoptam predominantemente uma estratégia de modelação orientada à aplicação ("package-based"). Neste contexto, cada aplicação cobre em geral aspectos bem definidos relativos a uma dada área de fabrico (planeamento, escalonamento, qualidade, etc.) e apresenta o seu "modelo parcial" da instalação fabril. Este "modelo parcial" resulta do esforço de configuração de software standard e de frequentes desenvolvimentos específicos. São assim remetidos para o utilizador os esforços de aprendizagem das diversas "linguagens" e "modelos" envolvidos e de integração da informação, bem como a aquisição de conhecimento específico relativo a cada aplicação.

No âmbito do projecto SHOP-CONTROL concluiu-se que seria conseguida uma melhor interface com o utilizador se a mesma linguagem de modelação fosse partilhada por todas as aplicações. Embora as diferentes aplicações sejam capazes de fornecer ao utilizador diferentes "vistas" sobre o sistema de produção, elas utilizam e partilham informação básica comum sobre matérias primas, recursos, produtos e operações. Neste sentido, o ambiente de modelação compreende uma linguagem de modelação com uma determinada gramática e vocabulário, bem como uma ferramenta de software para construção, edição e reutilização dos modelos.

O objectivo de manter o sistema aberto a futuras aplicações, levou a que o ambiente de modelação fosse dividido em três vistas [119]:

- A *Vista Física (ou Estrutural)*, que permite apresentar os objectos da instalação fabril (máquinas, transportadores, peças, produtos, trabalhadores, computadores, etc.) de uma forma próxima da realidade física, através de sinópticos com ícones das áreas fabris que fornecem "pontos de acesso" às diversas secções e objectos físicos da empresa (figura 5.7).

- A *Vista de Informação*, contendo dados relativos aos atributos dos mesmos objectos. Traduz-se numa vista sobre os diversos objectos na instalação fabril (tais como produtos e recursos), dada através das múltiplas aplicações (qualidade, manutenção, etc.) acedidas no contexto dos "pontos de acesso" do modelo (ver figura 4.13 "Invocação da aplicação de Supervisão no contexto de um objecto" no capítulo 4).
- A *Vista Funcional (ou de Processo)*, contendo informação sobre o ciclo de vida dos diversos elementos existentes na instalação fabril (máquinas, peças, produtos, trabalhadores, etc.) e definindo as operações a serem conduzidas na instalação fabril (figura 5.8).

No ambiente de modelação existente, cabe ao utilizador decidir qual o pormenor com que deseja modelar as diversas zonas da fábrica, i.e. a granularidade do modelo necessária para cada uma delas. As áreas críticas devem naturalmente ser modeladas com maior detalhe uma vez que necessitam de ser observadas, estudadas e controladas com mais rigor.

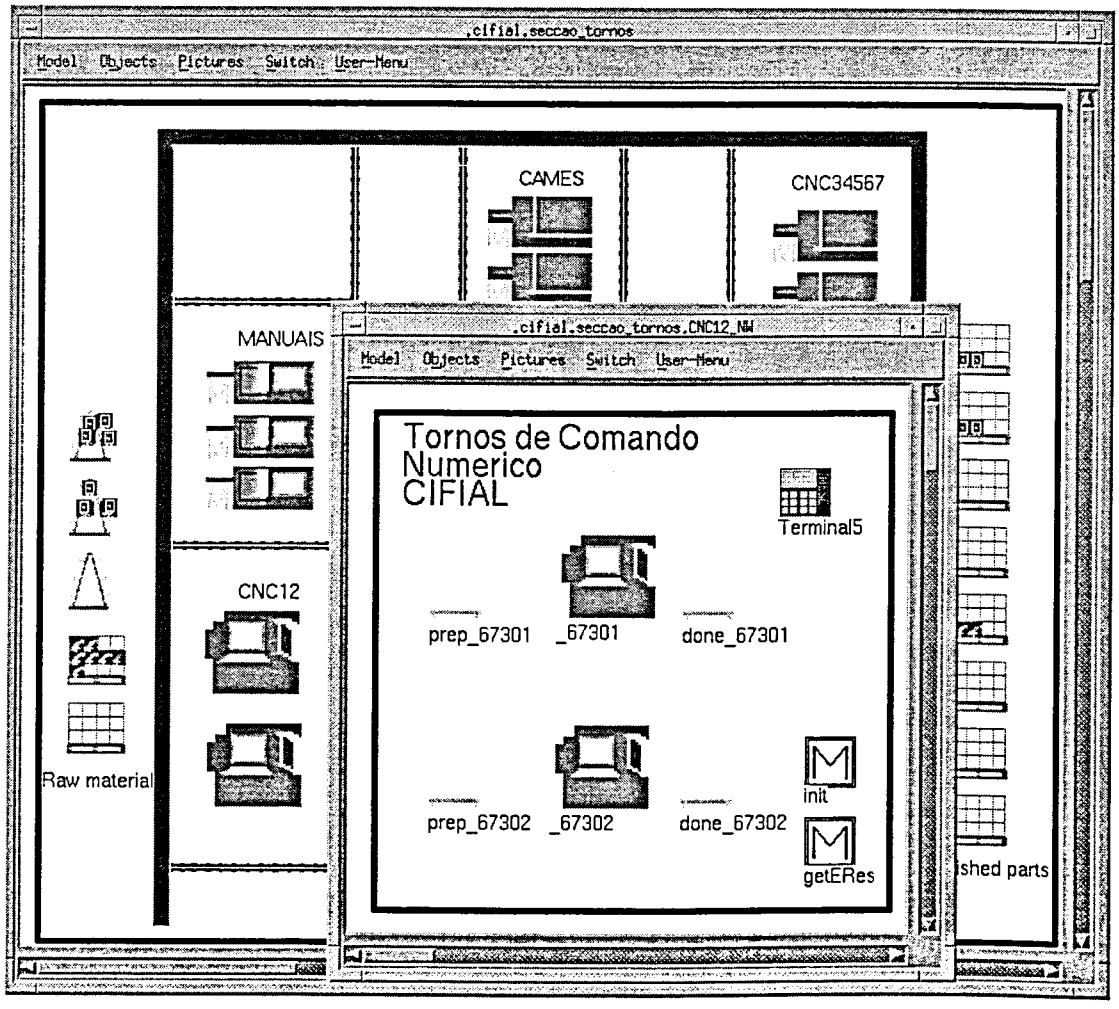


Figura 5.7 - Vista Física da Instalação Fabril, Modelo criado no Simple++ da Instalação Piloto do SHOP-CONTROL na Cifial[2]

[2]Cifial - Empresa fabricante de ferragens, torneiras, fechaduras, etc., onde os protótipos resultantes do projecto SHOP-CONTROL foram demonstrados em Abril de 1993.

Cada vista pode ser estruturada hierarquicamente, permitindo através da organização do modelo controlar a complexidade da realidade física modelada. As primitivas adoptadas para a modelação da vista física foram as disponibilizadas pelo Simple++, uma ferramenta de simulação desenvolvida no IPA-FhG[3] e depois transformada num pacote comercial de modelação e simulação fabril [60]. Estas primitivas podem ser combinadas pelo utilizador na construção de uma biblioteca de blocos construtivos mais complexos e específicos de uma empresa em particular. A cada um destes blocos construtivos está associada uma representação gráfica sob a forma de um ícone que, ao poder ser editado pelo utilizador, permite a personalização do modelo através de imagens gráficas fiéis da instalação fabril. A figura 5.7 apresenta o modelo físico de uma secção de tornos da Cifial criado usando o Simple++. Este modelo apresenta os recursos de fabrico existentes, tornos CNC de diversos tipos e idades tecnológicas, assim como os "buffers" do em-curso-de-fabrico e os armazéns de matéria prima e de produtos acabados.

As primitivas funcionais usadas na modelação do ciclo de vida dos produtos e recursos foram adoptadas do modelo de controlo do projecto de estandardização "CIM-AG" do comité de estandardização alemão "DIN"[119]. Existem basicamente dois tipos de primitivas: as primitivas de operação, usadas na modelação da interacção entre produtos e recursos; e as primitivas de sincronização, modelando o estado particular em que se encontram os recursos e os produtos num determinado instante no tempo (figura 5.8, a) e b)) [120]. A cada primitiva de sincronização é associado o número de elementos (componentes, produtos, unidades de transporte, máquinas) presentes naquela fase do ciclo de vida do produto / recurso. Assim sendo, os recursos ou produtos existentes nas primitivas de sincronização, correspondem a de certa forma a um "armazenamento intermédio", estando disponíveis para ser escalonados se necessário.

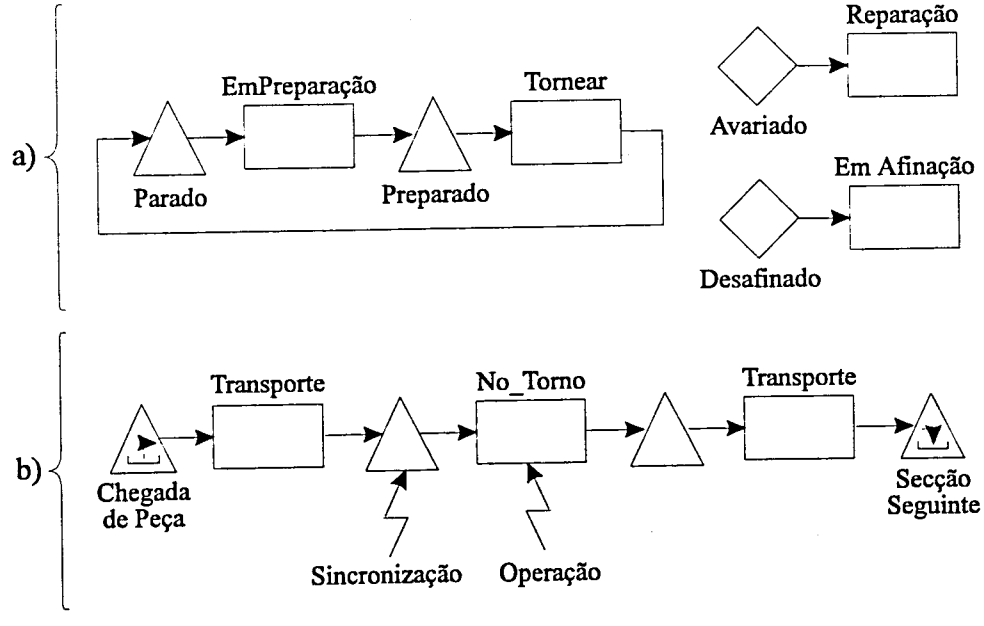


Figura 5.8 - Vista Funcional ou de Processo (Ciclo de uma classe de recursos a) ou de produtos b))

[3]IPA-FhG - Instituto Fraunhofer para as Técnicas de Produção e Automatização

A vista funcional apresenta as diversas operações executadas sobre a matéria prima desde o momento em que esta chega à secção de tornos até que o resultado do processamento é colocado no armazém de produtos acabados (figura 5.8 b)). Para cada operação no ciclo de vida do produto, o modelador indica quais os recursos físicos que são capazes de realizar a operação em questão. Comparando as figuras 5.7 e 5.8, podemos observar que a vista física da instalação fabril põe em evidência as propriedades físicas dos recursos e produtos, tais como a sua forma, cor, dimensão, capacidade e a sua localização e disposição na instalação fabril ("layout"). Por outro lado, a vista funcional descreve os encaminhamentos durante o fabrico e fornece em "tempo-real" o estado actual e a transição de estados dos diversos objectos na instalação fabril (recursos, 5.7 a) e produtos, 5.7 b)). Estas duas vistas modelam de uma forma integrada o fluxo de materiais inerente ao processo de fabrico na suas diferentes vertentes.

O diversos serviços / funcionalidades necessários à gestão das operações fabris estão intimamente ligados à vista de informação da instalação fabril, a qual providencia o quadro para a integração das diferentes aplicações (supervisão, escalonamento e controlo, gestão da qualidade e gestão da manutenção) no ambiente de modelação. A vista de informação foi criada usando o modelo entidade-associação, o qual também foi usado no suporte ao projecto e manutenção da base de dados fabril. A vista de informação é assim fornecida pelas diferentes aplicações através de vistas distintas sobre a base de dados, permitindo que o utilizador consulte os valores actuais assim como informação histórica relativa a variáveis e parâmetros da instalação fabril no contexto de cada um dos objectos e / ou níveis hierárquicos do controlo.

Complementaridade das Diferentes Vistas do Modelo

Existem no modelo dois tipos de informação crítica da maior relevância para a gestão da instalação fabril [120]: a localização dos objectos (informação física) e a informação relativa ao seu estado (informação do processo). Qualquer SGOF actual deverá fornecer este tipo de informação, grande parte da qual poderá com vantagem ser apresentada graficamente. Temos assim as duas vistas acima referidas sobre o ambiente fabril: a vista física e a vista de funcional (ou de processo).

Os blocos construtivos básicos utilizados na vista física levam em consideração as propriedades geométricas dos objectos e referem-se ao local onde os objectos se encontram fisicamente, por exemplo "qual a ordem de fabrico e em que contentor e em que palete, está no "buffer" de entrada da máquina _67301", ou "onde está localizada a ordem de fabrico urgente XPTO_47". A vista de funcional apresenta por outro lado o estado dos diversos objectos, por exemplo "máquina T avariada", "máquina S em preparação", etc., assim como as transições de estado que podem estar a ocorrer, por exemplo permitindo saber "qual o em-curso-de-fabrico do produto X disponível antes da operação Z" ou "quantas operações faltam à ordem urgente XPTO_99 antes de poder ser entregue" (figura 5.8).

Os blocos construtivos de ambas as vistas têm alguma funcionalidade embebida. Os blocos usados no modelo físico concentram-se em propriedades físicas dos objectos, tais como o comprimento de uma passadeira de transporte e o comprimento das peças a transportar, que vai limitar por exemplo o número de peças em simultâneo sobre a passadeira. Os blocos construtivos para a vista funcional focam aspectos de sincronização dos ciclos de vida das diferentes classes de produtos em produção, por exemplo uma operação de tornear exige uma máquina com um coeficiente de capacidade (C_{pk}) mínimo bem como um operador com um determinado tipo de qualificação. O comportamento interno destes blocos selecciona e afecta automaticamente a uma dada operação um dos

recursos disponíveis com as características desejadas, e se necessário assegura que o recurso esteja preparado para iniciar a operação sobre um determinado objecto. Todos estes atributos caracterizando os objectos físicos do modelo podem ser alterados no decorrer de uma corrida de simulação ou mesmo durante a sua execução em tempo-real. Deste modo, parâmetros como a capacidade da máquina, a sua velocidade de processamento, o seu tempo de preparação, ou mesmo a dimensão de um armazém, etc., podem ser modificados interactivamente.

Enquanto que a vista física é construída de acordo com a planta da instalação fabril ("layout" das máquinas, transportadores, armazéns), a vista de funcional pode ser gerada partindo da informação contida na lista de materiais (BOM) e nos planos de trabalho ("workplan") do produto, construindo assim a gama operatória. Estas duas vistas são necessariamente complementares: o controlo da vista física durante a simulação é realizado pela informação na vista de processo. Só no caso de operações que requeiram conhecimento detalhado da situação física da instalação fabril, tal como uma operação de transporte, o controlo é entregue à vista física (por exemplo para o cálculo da duração exacta do transporte baseado no comprimento da passadeira e na velocidade da mesma).

A Múltipla Utilidade do Modelo do Sistema de Produção

O modelo fabril, vai tornar-se no elemento central do SGOF sendo utilizado ao longo de todo o ciclo de vida do sistema de gestão fabril integrada.

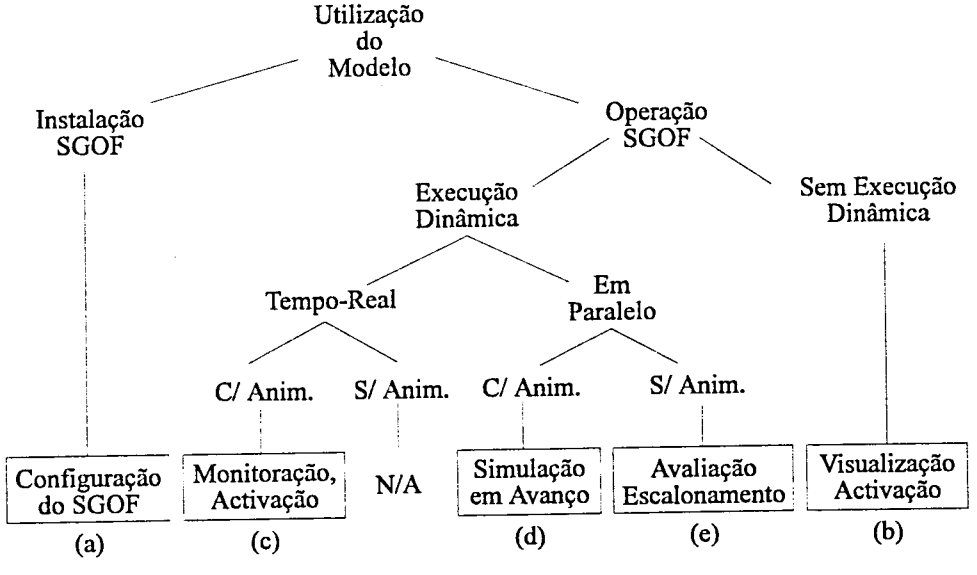


Figura 5.9 - Múltipla utilização do modelo fabril

A figura 5.9 ilustra a utilização do modelo fabril no âmbito da instalação e operação do SGOF. No período de instalação, é realizada a construção do modelo fabril, seguindo um processo em espiral de refinação (a). Relativamente à fase de operação do SGOF, temos dois modos de utilização. Num deles não existe execução da simulação, i.e. operação do modelo sem execução dinâmica em que este apenas reage aos eventos provenientes da instalação fabril e utiliza a animação da superfície gráfica, não desempenhando mais nenhum papel de controlo (b). Como exemplo, temos o caso de um evento vindo da IF solicitando ao modelo a indicação da próxima operação a ser executada, respondendo-lhe este com a indicação da tarefa que se segue, extraída da tabela com o escalonamento das tarefas para o recurso em questão. A operação do modelo fabril com execução dinâmica

pode ocorrer em dois modos distintos, execução em tempo-real ("on-line") e em paralelo ("off-line"). No primeiro caso existirá uma execução da simulação em tempo-real, apenas aplicável com a animação do modelo activada, sendo simultaneamente seguido o evoluir da produção na instalação fabril, cujo estado real será continuamente comparado com o estado desejado / planeado que é reflectido pela simulação do modelo (c). No segundo caso, teremos ainda duas formas de utilização do modelo de simulação: como ferramenta de diagnóstico e ou prognóstico (d); como ferramenta de avaliação de soluções de escalonamento propostas por um módulo de optimização (e).

Configuração do SGOF

O modelo aparece pela primeira vez no decorrer da configuração / implementação do SGOF num determinado ambiente de produção. A criação de um modelo realista leva o modelador a estudar com detalhe todos os aspectos do processo de fabrico. Nesta fase, será crítico questionar e possivelmente alterar regras e procedimentos desactualizados e / ou desejados ou introduzir simplificação. Geralmente a construção do modelo é realizada através de processos evolutivos baseados em espirais de melhorias incrementais de protótipos cada vez mais refinados e convergindo para uma versão final. Os primeiros protótipos cobrem unicamente um determinado sector piloto da instalação fabril, e a granularidade do modelo é apenas a necessária à realização de simulações em paralelo ("off-line") para comparar, por exemplo, estratégias de controlo alternativas e permitir orientar refinamentos subsequentes. Uma vez completado o modelo, ele será automaticamente guardado na base de dados do SGOF. Todas as aplicações utilizam a base de dados, tanto para o armazenamento de informação privada como para partilhar informação pública com outras aplicações. O modelo armazenado na base de dados é partilhado por todas as aplicações, formando deste modo a espinha dorsal da estrutura de informação do SGOF. Na introdução faseada das diferentes aplicações, o esforço de modelação e configuração é assim reduzido à especificidade da aplicação, uma vez que a informação de carácter geral relativa às máquinas, produtos e processos já se encontra definida no modelo de dados.

Na base de dados devem pois ser distinguidos dois tipos diferentes de informação: a informação pública partilhada por todas as aplicações (por exemplo códigos de ordens de fabrico, produtos, recursos), e a informação privada de cada aplicação (pontos de inspecção na qualidade, gamas operatórias no escalonamento, etc.). Além disso, poderá ainda ser distinguida na base de dados uma parte essencialmente estática, correspondendo maioritariamente ao modelo fabril estável, e uma outra parte contendo informação dinâmica, i.e. evoluindo no tempo, onde são registadas todas as ocorrências relevantes da instalação fabril (mudanças de estado de recursos, avanços de lotes, alterações de armazéns intermédios, alarmes, etc.).

A base de dados funciona pois como um espelho do próprio processo produtivo, armazenando informação de carácter dinâmico constantemente actualizada pelas infra-estruturas de recolha de dados. Sempre que o modelo é carregado da base de dados, é nele re-estabelecida de imediato a imagem do estado actual da instalação fabril.

Operação Global do SGOF

Tipicamente o SGOF recebe, via base de dados ou ficheiros ASCII, as ordens de fabrico geradas pelos sistemas de planeamento agregado, as quais estabelecem as necessidades de produção em resposta às encomendas colocadas pelos departamentos de marketing / vendas. Nestes sistemas (tipo MRP-II) é realizado o planeamento de médio e longo prazo, bem como o cálculo de necessidades. Em muitos casos o MRP inclui funções

de planeamento de operações assumindo capacidade infinita. O SGOF, como responsável pela gestão de operações ao nível da instalação fabril, utiliza então o escalonador na elaboração de um plano de execução detalhado. Este plano detalhado de curto prazo cobre janelas temporais variando entre algumas horas e alguns dias, e pode gerar informação, sob a forma de listas ou de diagramas de Gantt, utilizada para o posterior despacho de ordens de fabrico.

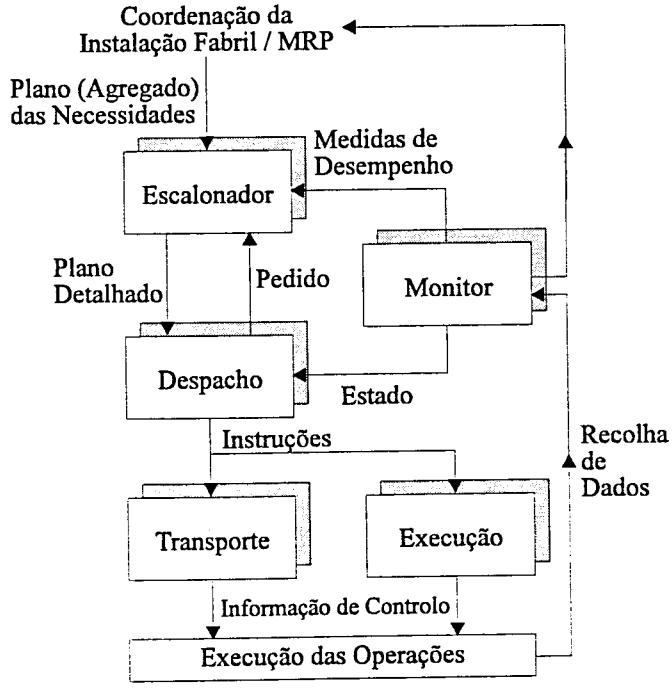


Figura 5.10 - Arquitectura COSIMA para o Controlo das Actividades de Fabrico

As tarefas de escalonamento, despacho e controlo das operações podem suportar-se de diferentes arquitecturas funcionais. Em [117], é caracterizada a proposta do projecto ESPRIT nº 477 COSIMA para a arquitectura a utilizar no âmbito do controlo das actividades de produção. Nesta arquitectura constituída pelos cinco blocos ilustrados na figura 5.10, o escalonador recebe do sistema hierarquicamente superior um plano agregado com as necessidades de produção, e elabora um plano detalhado que determina a utilização dos recursos fabris no interior de uma janela de tempo especificada. O módulo de despacho é então responsável por traduzir o plano detalhado numa sequência de acções de transporte e de execução de operações de fabrico enviando instruções de controlo do transporte dos materiais entre os centros de trabalho e da execução das operações em cada um deles. Quando estas instruções de produção ou de transporte são enviadas para a instalação fabril, podem surgir perturbações inesperadas. Assim, o principal objectivo do despacho é o de reagir ao estado actual do ambiente de produção seleccionando a melhor estratégia de afectação de recursos exequível. Para que esta função seja correctamente executada, este módulo recebe do escalonador: o plano detalhado de fabrico, com o sequência e a temporização das diferentes operações a serem executadas; a informação estática de como essas operações são executadas. Recebe por outro lado do monitor informação continuamente actualizada sobre o estado actual da instalação fabril. O despacho é assim o elemento de controlo, assegurando dentro do possível a execução do plano estabelecido pelo escalonador. O papel do monitor consiste no fornecimento da informação adequada e em "tempo útil" sobre o estado da instalação fabril, tanto ao escalonador como ao despacho, para que eles possam executar correcta e atempadamente as suas tarefas.

Na figura 5.11 é apresentada uma implementação típica do PROFIT, que acaba por ter uma estrutura funcional comparável à da arquitectura da figura 5.10. Aqui o plano de curto prazo, por exemplo diário, é obtido partindo do plano agregado fornecido pelo sistema hierarquicamente superior e armazenado na base de dados (BD). Realizado o escalonamento, quer de uma forma manual quer automática, e tendo em vista a satisfação de um conjunto de critérios de optimização e de restrições, este é armazenado numa tabela temporária utilizada para o despacho das ordens de fabrico. No exemplo de figura 5.11, o operador fabril (controlador da célula) interage com um terminal fabril para obter a próxima ordem de fabrico afectada a um determinado recurso, solicitando "qual a ordem de fabrico para o recurso _67301 ?". Este pedido ao sistema é recebido pelo monitor e enviado ao módulo de despacho que, após consultar a tabela de tarefas na base de dados (BD), responde com a indicação da ordem de fabrico a ser executada.

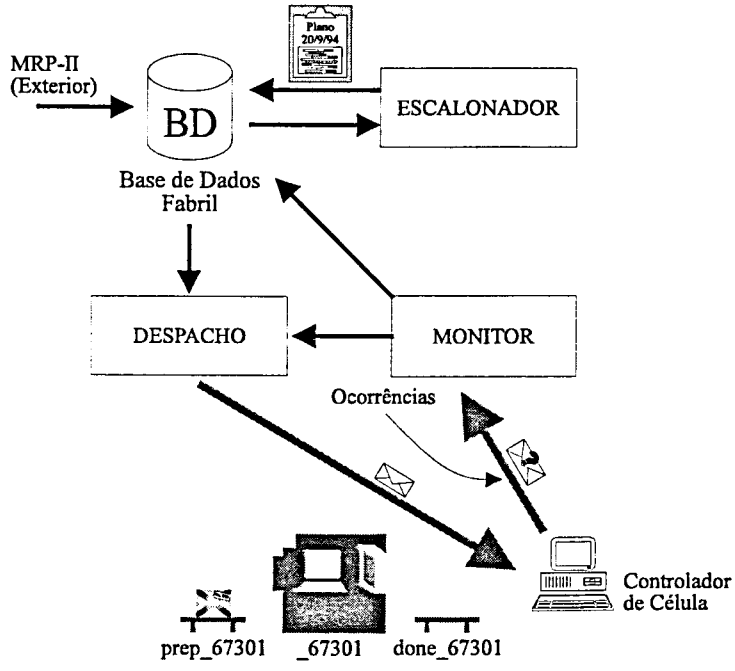


Figura 5.11 - Escalonamento e Despacho de Produção

As ocorrências críticas no fabrico, tipicamente anomalias perturbadoras do desenrolar das actividades, i.e. problemas de qualidade, avarias de máquinas, etc., são monitorizadas e automaticamente reflectidas na base de dados fabril e de imediato comunicadas às aplicações relevantes. Não existindo ainda no PROFIT um módulo de re-escalonamento automático, sempre que este tipo de problemas ocorre o gestor da produção é notificado através de um alarme podendo então tomar as decisões que ache necessárias, incluindo evidentemente a utilização do módulo de escalonamento para refazer um plano detalhado que deixou de ser válido. Assim, o conhecimento atempado das indisponibilidades não previstas permite o re-escalonamento detalhado de curto prazo (ou escalonamento reactivo) com informação fiável e actualizada.

A figura 5.12 ilustra o papel do modelo fabril do PROFIT no despacho da produção. De facto, os módulos *monitor* e o *despacho* da arquitectura lógica representada na figura 5.11 estão embebidos no modelo fabril (através das funcionalidades de monitorização do modelo, e da aplicação de Despacho, ***Di***, na figura 5.12). Observe-se que o plano detalhado a executar é guardado na base de dados fabril.

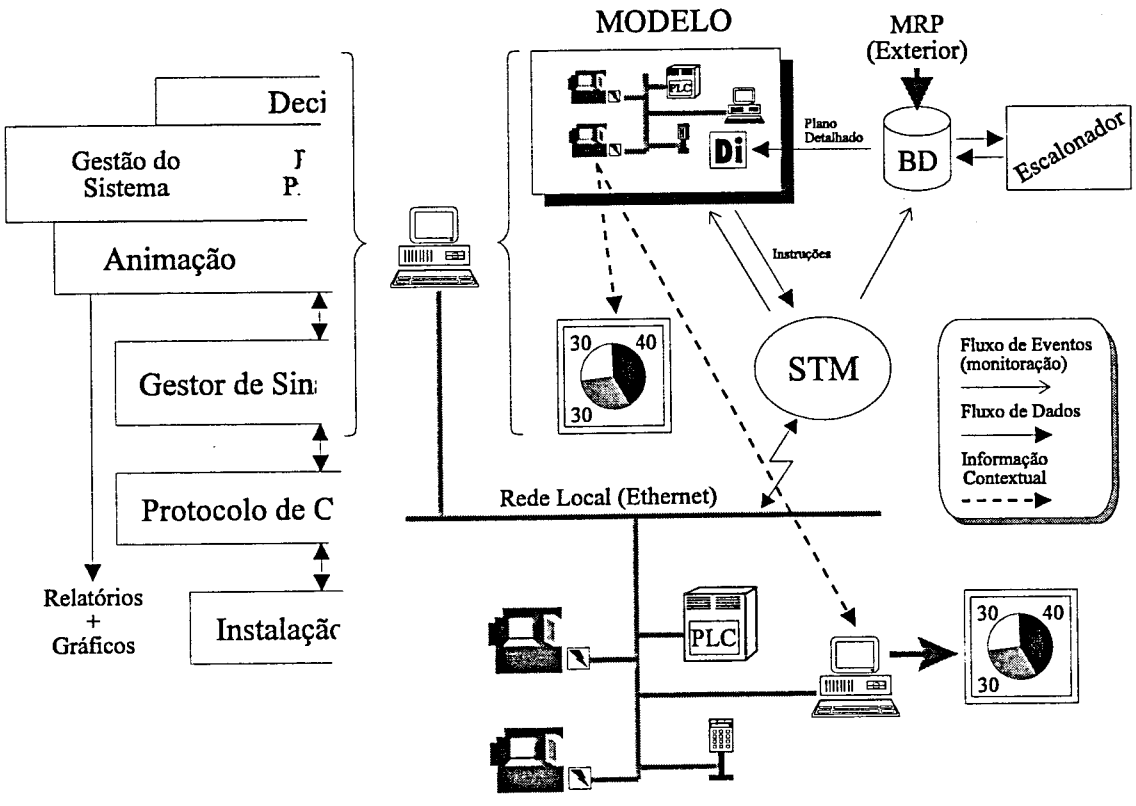


Figura 5.12 - Execução do Plano de Produção, Despacho de Ordens de Fabrico

Comparando as figuras 5.4 (página 124) e 5.12 é possível identificar as tarefas desempenhadas pelo modelo fabril. Deste modo, temos as tarefas do Decisor partilhadas pelo módulo de Despacho da produção e pelo próprio gestor da produção na sua interacção com o modelo fabril e com o escalonador. Por outro lado, o modelo disponibiliza funcionalidades de monitorização, e permite ainda a realização de cálculos estatísticos bem como a animação em tempo-real da interface gráfica.

A infra-estrutura de suporte transforma o Gestor de Sinais / Mensagens no Sistema de Transferência de Mensagens (STM) usado no PROFIT, permitindo comunicar através de redes locais, combinando uma rede Ethernet e possivelmente redes industriais proprietárias com o equipamento de recolha de dados distribuído pela instalação fabril. Ao nível da instalação fabril e através da utilização de terminais fabris (PCs, autómatos programáveis ou simples terminais de leitura óptica), o operador terá acesso à informação armazenada no modelo e / ou na base de dados apresentada sob a forma gráfica ou de relatórios adaptados à suas necessidades.

Visualização e Monitorização para Controlo do Estado de Produção Actual

Durante a operação, o SGOF pode ser utilizado como diagrama sinóptico animado da instalação fabril, à semelhança dos sistemas SCADA largamente divulgados nas indústrias de processo. A visualização pode realizar-se sem que o modelo esteja em simulação, bastando muito simplesmente que esteja activa a ligação ao sistema de recolha de dados. Sempre que um evento resultante da execução do próprio processo de fabrico é enviado ao modelo através do sistema de recolha de dados, este é actualizado através da alteração de ícones ou por modificação da posição dos objectos no modelo.

Quando à simples visualização se junta a execução do modelo em tempo-real (i.e. simulação em tempo-real), é possível em cada momento realizar a comparação do estado

actual da instalação fabril com o estado que esta teria se todas as operações tivessem sido executadas de acordo com o planeado. O modelo executa assim a simulação em tempo-real do plano de produção em paralelo com o desenrolar da execução do próprio processo na instalação fabril. Através dos terminais fabris e da sua representação no modelo realiza-se a sincronização do estado do modelo com o estado do sistema produtivo real. Entre as ocorrências de dois eventos sucessivos, que determinam os "instantes de amostragem" do sistema de fabrico[4], a simulação em tempo-real assume que a execução do sistema decorre conforme o planeado.

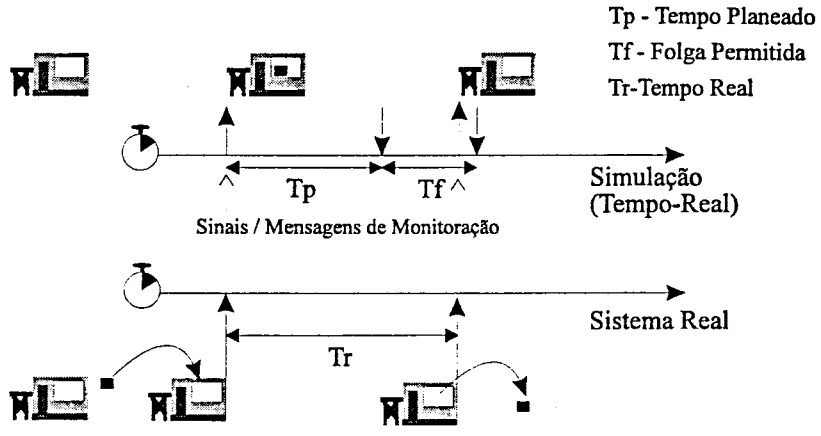


Figura 5.13 - Simulação em tempo-real: o modelo segue o processo de produção real e é actualizado por este

A infra-estrutura de suporte permite pois que seja detectado em tempo-real qualquer desvio da execução das operações de fabrico relativamente ao plano estabelecido, assim como actuar em conformidade. A detecção de desvios baseia-se na comparação dos eventos no modelo de simulação com os eventos no sistema real (figura 5.13). Como exemplo, temos a situação em que um lote de um produto inicia numa máquina um conjunto de operações que deverá estar terminado decorrido um período de tempo Tp. Neste preciso momento é colocado na lista de eventos da simulação o evento que irá provocar a saída da peça uma vez completada a operação, i.e. decorrido o tempo Tp. Se o evento proveniente da planta fabril ocorre antes do evento de simulação correspondente ao tempo planeado (i.e. Tr<Tp), o modelo é actualizado sendo o novo tempo de operação registado na base de dados para processamento estatístico e derivação de valores padrão Tp para futura referência. Se o evento vindo da instalação fabril chega depois de decorrido Tp mas antes de esgotada uma tolerância pré-determinada (i.e. Tr<Tp+Tf), é realizado o seu registo na base de dados e é feita a actualização do modelo tendo em atenção este atraso em relação a Tp (figura 5.13). Se o evento da instalação fabril muito simplesmente não chega decorrido o tempo pré-determinado (Tp+Tf), é então disparado um alarme que vai alertar o responsável pela produção e o operador ou encarregado (figura 5.14). Neste caso, a base de dados será actualizada logo que seja disparado o evento notificando a conclusão da tarefa (Tr). No PROFIT, o escalonador não reage automaticamente a este tipo de eventos, devendo qualquer decisão / acção subsequente para correcção do plano ser executada manualmente,

[4]Designação dada aos instantes de tempo em que o sistema de controlo adquire / actualiza a informação sobre o sistema controlado, por analogia com o que se passa na amostragem de um sistema físico por um sistema de controlo discreto.

embora tal decisão possa eventualmente utilizar o mesmo escalonador [121]. Os valores de Tp e Tf são lidos no plano de produção detalhado gerado pelo escalonador, sendo Tp lido directamente e Tf calculado tendo em atenção o valor de Tp e o instante previsto para o inicio da(s) operação(ões) subsequentes dependentes da conclusão desta.

Tanto o modo de monitorização como o de visualização permitem que outras aplicações (supervisão, qualidade ou manutenção) sejam activadas dentro de um determinado contexto.

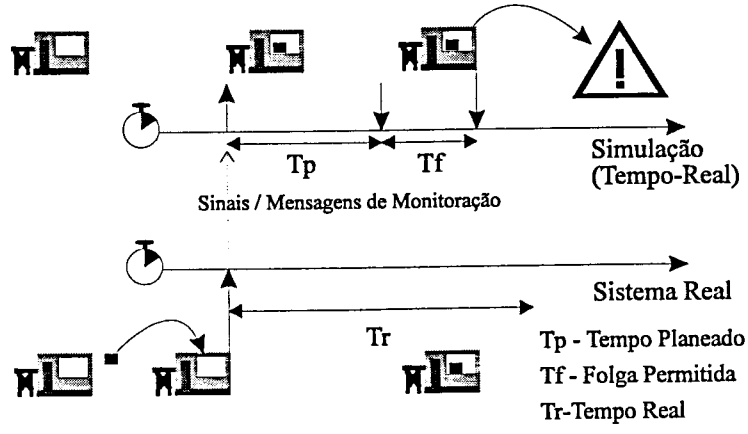


Figura 5.14 - Simulação em tempo-real: o modelo segue o processo de produção real e detecta uma situação anormal

Simulação em Avanço e Avaliação de Planos de Produção Futuros

No modo de execução "em paralelo" o modelo é executado no modo "o mais rápido possível", i.e. sendo a velocidade apenas limitada pelo desempenho do hardware disponível. Como estado inicial para a simulação, é tomado o estado actual da produção reflectido na base de dados. Este modo de funcionamento poderá ser utilizado para realizar simulações regulares para prognóstico do impacto que pode ter a ocorrência de perturbações, tais como a introdução de um conjunto de ordens de fabrico urgentes, a avaria de uma máquina, o atraso na entrega de matéria prima ou de componentes para montagem. O resultado da simulação é perfeitamente realista, uma vez que o seu ponto de partida foi o estado actual do processo produtivo, incluindo a carga das máquinas os valores dos stocks intermédios, etc. (se entretanto não se alterar no decorrer da mesma simulação).

A animação no modo de simulação em avanço tem grande interesse para o utilizador, permitindo que ele observe a evolução do modelo em execução, e com base nessa observação tome as medidas correctivas que julgue necessárias. Em alternativa, se essas medidas são tomadas de uma forma semi-automática por um outro processo será então vantajoso "desligar" a animação para que a simulação possa ser realizada mais rapidamente. Neste caso, o modelo avalia o plano de produção e reporta o valor da função objectivo a um outro processo UNIX executando em paralelo. O modelo de simulação pode ser utilizado para avaliar a qualidade de soluções / planos obtidos usando algoritmos diversos (heurísticas, algoritmos genéticos, ou outros algoritmos típicos da investigação operacional) [121]. O utilizador aguarda assim a execução de sucessivas corridas do simulador até que a função objectivo atinja um determinado nível desejado, interrompendo então o processo de optimização, e podendo nesse momento seleccionar o plano obtido como o novo plano "óptimo" para produção.

O Posicionamento do PROFIT na Matriz GERAM

Dado que o PROFIT pode funcionar como uma ferramenta de integração ao nível dos sistema de execução fabril, terá todo o interesse o estudo do seu posicionamento na matriz de avaliação GERAM. Este estudo vai permitir a constatação das áreas da integração cobertas e tomar decisões relativamente a trabalhos futuros tento em vista a cobertura de outras áreas [122].

A figura 5.15 ilustra o posicionamento do PROFIT no âmbito da matriz de avaliação GERAM. Como é possível observar na legenda da figura, foram estabelecidas cinco classes distintas segundo as quais será realizada a classificação das ferramentas disponíveis no PROFIT. São identificadas as áreas na matriz de avaliação onde exista uma vista sobre o sistema de fabrico suportada ou não por uma linguagem de modelação, ou ainda vistas particulares providenciadas por software de aplicação.

A disponibilidade de uma linguagem passível de ser utilizada na modelação da Arquitectura do Equipamento de Fabrico (AEF) foi demonstrada nos parágrafos anteriores. Ficou claro que o PROFIT disponibiliza uma ferramenta suportando a construção de modelos executáveis do sistema de fabrico, obtidos através da modelação integrada dos seus recursos físicos (vista física) e do ciclo de vida dos seus produtos e recursos fabris (vista funcional). A utilização de uma ferramenta de modelação orientada ao objecto, permite raciocinar em termos de modelos genéricos e em termos de modelos parciais para um determinado tipo de indústria, antes de os particularizar para o caso em estudo. Assim sendo, em face da utilização da ferramenta de modelação do PROFIT nas fases de projecto e de implementação, na figura 5.15 teremos a disponibilidade de uma linguagem de especificação (●) nas vistas funcionais e de recursos nos modelos genéricos, parciais e particulares. Relativamente à vista de informação na AEF não está associada qualquer linguagem, sendo esta vista providenciada através de vistas estruturadas sobre base de dados apresentadas no contexto de um recurso físico seleccionado. Deste modo, na figura 5.15 temos uma vista não suportada por uma linguagem (○) para as vistas de informação nos modelos genéricos, parciais e particulares. Será assim possível "ver" informação sobre elementos do modelo da AEF, por exemplo fabricante e capacidade de produção de uma máquina, ferramenta instalada, conteúdo de uma palete junto a uma máquina, etc..

A Arquitectura do Sistema de Informação (ASI) assume no PROFIT a forma de um conjunto de aplicações básicas de gestão fabril partilhando a base de dados com o modelo fabril. Estas aplicações (poderão ser encaradas como ferramentas) são genéricas, na medida em que existem necessidades partilhadas entre várias indústrias relativamente à gestão da qualidade, da manutenção, à gestão de alarmes e ao escalonamento. Em qualquer dos casos, e apesar da modularidade do software de aplicação desenvolvido, será sempre necessário esforço de consultadoria / engenharia na adaptação / configuração das aplicações de software aos casos particulares onde ele vai ser instalado. Não existe assim uma abordagem baseada em modelos tal como no CIMOSA, onde se pretende fazer a modelação da aplicação de acordo com a situação particular com geração automática do código executável correspondente. Em relação ao posicionamento do PROFIT na arquitectura do sistema de informação, apenas na faixa de implementação existem vistas funcionais, de informação e de estrutura / recursos para os modelos particulares, que podem ser adaptados na fase de construção (□). Neste caso particular das vistas na ASI, é patente a disponibilidade da interface com o utilizador uma vez que são já considerados os aspectos humanos da aplicação.

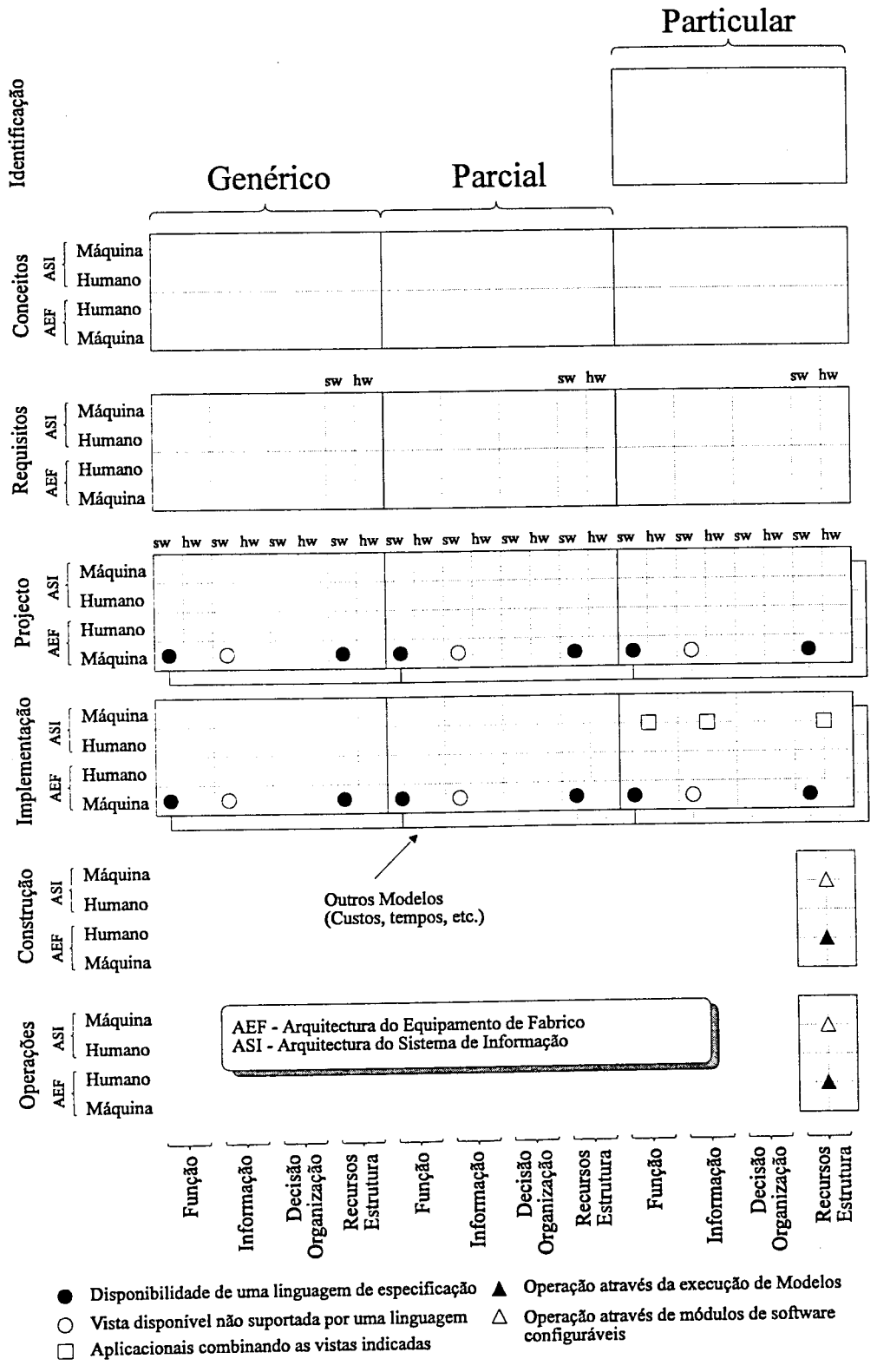


Figura 5.15 - Posicionamento Relativo das Ferramentas disponibilizadas pelo PROFIT na Matriz de Avaliação GERAM

Nas fases de construção e de operação, temos o suporte apresentado através de modelos executáveis para a AEF e de módulos de software configuráveis para a ASI. A interacção com o utilizador é considerada na construção do modelo executável da AEF, uma vez que o papel do utilizador deve ser levado em conta tendo em vista a gestão do modelo executável nas suas fases de operação e consequente manutenção.

5.5 Conclusão

Neste capítulo foram apresentadas as necessidades de modelação envolvidas num projecto de integração de um sistema de fabrico. O contexto para esta apresentação foi fornecido pela arquitectura de Purdue (PERA), onde a identificação das diversas fases do projecto de integração, bem como a identificação das interfaces entre as três arquitecturas, permitiram identificar claramente as necessidades de modelação das Arquitecturas Humana e Organizacional (AH&O), do Sistema de Informação (ASI) e do Equipamento de Fabrico (AEF). Deixando de fora a AH&O por não estar no âmbito deste trabalho, foram tratadas as necessidades de modelação no âmbito da AEF e da ASI. O papel da modelação para a simulação e sua reutilização no controlo sob a forma de modelos executáveis foi tratado em seguida, evidenciando a tendência para o abandono da filosofia dos modelos descartáveis.

Os modelos executáveis constituíram o tema central deste capítulo, tendo sido primeiro feita a apresentação do modelo proposto pela arquitectura CIMOSA, e depois pelo PROFIT. Neste contexto e a pretexto da classificação do PROFIT no âmbito da matriz de avaliação GERAM, foi realçado o facto de os modelos do CIMOSA de concentrarem nos modelos de suporte à ASI, tendo o modelo do PROFIT as suas facilidades de modelação concentradas na AEF. De facto, verificou-se que no PROFIT não existe qualquer facilidade de modelação formal no âmbito da ASI, disponibilizando-se apenas um conjunto de aplicações de gestão fabril configuráveis dentro de limites impostos pela sua arquitectura e funcionalidades.

A relativa complementaridade detectada entre as facilidades de modelação do CIMOSA e o PROFIT poderiam levar a considerar a possibilidade da sua integração. Tal resultaria na integração de dois ambientes de modelação distintos, mas não foi de facto possível por não existirem ferramentas para a geração de modelos executáveis conformes com o CIMOSA. Como veremos no capítulo seguinte, a estratégia seguida para o suporte da modelação da ASI foi a utilização de uma linguagem de modelação formal e a sua completa integração com o Simple++. Esta aproximação vai permitir caracterizar, através de modelos executáveis, as ASI e AEF bem como as suas interfaces.

Capítulo 6

MODELAÇÃO DO SISTEMA DE INFORMAÇÃO, INTEGRAÇÃO DOS FLUXOS DE MATERIAIS & INFORMAÇÃO

6.1 Introdução

No capítulo anterior foram apresentadas e discutidas com algum detalhe as actuais tendências no âmbito da integração de sistemas de fabrico, tanto do ponto de vista do próprio projecto de integração como da utilização de técnicas de modelação na simulação e controlo do sistema integrado. Foram apresentadas as técnicas de utilização mais comum no âmbito da modelação dos sistemas de informação e dos sistemas de produção, e foi salientado o papel de cada vez de maior relevo da simulação, nomeadamente como suporte à realização de modelos executáveis reutilizados no controlo efectivo das operações de fabrico.

O CIMOSA e o PROFIT foram apresentados como exemplos de arquitecturas providenciando modelos executáveis. Foi ainda realizada a análise do posicionamento do PROFIT no âmbito da matriz de avaliação GERAM, no contexto da qual se verificou que embora ele suportasse a modelação da AEF, cujo modelo era reutilizado na monitorização e controlo do sistema de fabrico propriamente dito, não dispunha de suporte à modelação da ASI. Este facto levou à ideia de utilizar uma linguagem e ferramentas de suporte à especificação / descrição formal que, sendo integráveis com o Simple++, suprimissem a lacuna de modelação existente no actual ambiente PROFIT.

Neste capítulo será assim tratado o problema da escolha da linguagem formal a usar na modelação do sistema de informação, bem como o seu suporte e integração com o ambiente PROFIT, e em particular com a sua ferramenta de modelação e simulação, o Simple++. Neste contexto, será realizada uma breve introdução à linguagem escolhida e apresentada a

justificação dessa escolha. Seguir-se-á a descrição detalhada da ferramenta de modelação construída sobre o Simple++ para suporte à modelação formal das aplicações de software na área fabril. Será dada particular relevância à utilização da Bancada de Modelação desenvolvida, com a apresentação do ambiente integrado de modelação dos fluxos de materiais e de informação, que será ilustrado com diversos exemplos.

Será ainda realizada a apresentação de outras facilidades da Bancada de Modelação em fase final de desenvolvimento, tais como a interface com uma ferramenta CASE de suporte à análise e projecto estruturados e o apoio a actividades de modelação técnico-organizacional, sendo ilustrada a reutilização das primitivas de modelação desenvolvidas pelo autor. Finalmente, é realizado o preenchimento da matriz GERAM para a Bancada de Modelação desenvolvida, levando em consideração as facilidades resultantes da integração de ferramentas realizada.

6.2 Selecção de uma Técnica de Descrição Formal para a Modelação do Sistema de Informação

As vantagens da utilização de técnicas de especificação formal são bem conhecidas, e foram já brevemente apresentadas no capítulo anterior. Uma vantagem suplementar de relevo é o facto de muitas ferramentas fornecerem facilidades de geração automática de código executável a partir de uma descrição formal, permitindo ainda recorrer à imprescindível verificação matemática do modelo sempre que esteja envolvido o desenvolvimento de sistemas de grande complexidade e importância, como é o caso de todos os sistemas que envolvem a segurança de pessoas e bens (os denominados "Critical Fail-Safe Systems").

A utilização de uma TDF standard foi o primeiro, e quiçá mais importante, requisito colocado na selecção. Outros requisitos existiram, tais como: a adequação à especificação de sistemas industriais distribuídos (caracterizados pela comunicação assíncrona através de eventos), a disponibilidade de ferramentas de geração de código, e finalmente a sua adequação à integração com o Simple++ por forma a obter-se um ambiente integrado de simulação de fluxos de informação e de materiais.

A existência desde há vários anos de TDF vocacionadas para a especificação de sistemas distribuídos na área das telecomunicações, onde algumas atingiram o estatuto de standard internacional, acabou por ser como se verá um factor determinante da selecção realizada. A partir da década de 70 foram realizados esforços dentro da ISO e do CCITT para o desenvolvimento de TDF standard, na sequência dos quais resultaram três TDF complementares com o estatuto de standards internacionais (IS) ou de recomendações: Estelle [90,91] (IS9074) e LOTOS [92] (IS8807) definidos pela ISO, e SDL (Z100) pelo CCITT [24,109]. É a seguir apresentada a análise comparativa que suportou a escolha de uma destas três TDF.

6.2.1 LOTOS

Em 1989, a linguagem LOTOS ("Language Of Temporal Ordering Specification") tornou-se um standard ISO [123], tendo por objectivo a especificação de arquitecturas OSI ("Open Systems Interconnection") e de protocolos de comunicação e serviços associados. A sua capacidade de descrição e o nível de abstracção providenciado permitem no entanto a sua aplicação a um grande número de sistemas em que a concorrência, o não determinismo, a capacidade de reacção a eventos exteriores e a transferência de informação, sejam aspectos

importantes na descrição do comportamento. As vantagens evidentes da utilização de uma representação gráfica levou ao desenvolvimento do G-LOTOS, apenas uma extensão na sintaxe do LOTOS que não afecta a semântica subjacente [123].

Em LOTOS, um sistema é considerado como um *processo*, sendo este uma entidade capaz de executar *acções internas* e de *interagir* com o ambiente através de pontos de interacção chamados *portos*. O *ambiente* de um processo P é constituído pelo conjunto de processos externos a P, que podem não estar especificados, e que com ele interagem. Cada processo P pode ainda ser considerado como sendo constituído por um conjunto dinâmico de outros processos interagindo entre eles assim como com o ambiente. Em LOTOS, um processo é caracterizado pelos seguintes três elementos: o *comportamento*, definindo a sequência temporal dos eventos que o processo é capaz de executar, sem necessariamente especificar como essas sequências são conseguidas; a *lista de portos*, representando a interface entre os processos e o seu ambiente; o *nome*, que é utilizado sempre que ele é referido no contexto de outros processos.

A linguagem LOTOS procura abstrair dos recursos e dos algoritmos de implementação. No entanto, podem também ser incorporadas decisões de implementação permitindo a modelação de aspectos funcionais. Além disso, o LOTOS pode ser utilizado na especificação da arquitectura da implementação: através da estruturação dessa especificação por forma a reflectir os paradigmas da implementação.

A utilização desta linguagem foi assim agrupada em vários "estilos" utilizados ao longo das diferentes fases do ciclo de vida do produto / sistema [124]. Assim sendo, temos que: (1) A recolha de requisitos em LOTOS é realizada através da recolha de restrições, sendo os requisitos sistematizados na composição de um conjunto de restrições. É muito utilizado um mecanismo denominado "multiway rendez-vous" para forçar a simultaneidade de um conjunto de predicados que modelam os requisitos do utilizador. A especificação não estabelece a forma de chegar à solução através da combinação de predicados, agrupando apenas os predicados que devem ser verificados para que a solução seja válida; (2) A especificação orientada aos recursos é aplicada quando se deseja considerar os recursos reais. O recursos são modelados de acordo com o paradigma da orientação ao objecto. Do ponto de vista externo (dos requisitos) o comportamento do sistema não é alterado; no entanto ocorrem agora múltiplos eventos internos por forma a dirigir o sistema no caminho correcto; (3) Na especificação orientada à implementação é usado um estilo de modelação traduzindo um fluxo único de controlo, característica das linguagens de programação não concorrentes, sendo mesmo possível a especificação de máquinas de estados.

6.2.2 Estelle

A linguagem Estelle foi criada para a especificação de sistemas distribuídos, mais particularmente protocolos e serviços de comunicações [91,125], existindo ferramentas de suporte tais como o GROPE ("Graphical Representation of Protocols in Estelle") [126] e o EDT ("Estelle Development Toolset") [127]. Uma especificação em Estelle tem por objectivo a descrição da estrutura de um conjunto de autómatos [128] comunicando entre si, sendo as acções internas de cada autómato descritas em PASCAL (implementação particular da linguagem com algumas extensões e restrições). Uma especificação em Estelle é composta por diversos módulos que poderão ser criados e destruídos dinamicamente. Tipicamente estes módulos realizam funções que podem ser executadas em paralelo, sendo a comunicação entre eles usada para a troca de informação ou como forma de sincronização da sua execução sempre que necessário. O Estelle providencia ainda o encapsulamento dos

dados e do comportamento, tornando impossível que um módulo observe ou altere a estrutura interna e o comportamento de outros módulos. A modularidade e as facilidades de encapsulamento do Estelle tornam possível a construção de especificações de protocolos complexos e de sistemas distribuídos através da combinação das descrições de objectos reutilizáveis de pequena dimensão.

Na comparação de uma especificação em LOTOS com a sua equivalente em Estelle, esta última é consideravelmente menos abstracta. No Estelle, os aspectos de comportamento são expressos sob a forma do mecanismo operacional necessário / proposto para atingir um determinado comportamento, e não somente como simples descrição declarativa do comportamento. O Estelle requer de facto a descrição do mecanismo, não restringindo este a gama dos comportamentos possíveis. Uma característica muito positiva do Estelle é o facto de as especificações resultantes apresentarem um carácter informativo para quem tiver interesse nos detalhes de implementação. Além disso, é possível produzir protótipos e realizar simulações usando as especificações em Estelle através de compiladores e simuladores. Neste contexto, poderá ser necessário fazer evoluir uma especificação abstracta para uma especificação orientada à implementação, através da substituição de alguns comportamentos por outros menos gerais e mais orientados à implementação. Esta tarefa poderá traduzir-se nomeadamente na eliminação de decisões não determinísticas, na associação de prioridades a um determinado conjunto de acções e na atribuição de valores específicos a temporizadores.

6.2.3 SDL

As duas principais características do SDL são a sua capacidade de descrever a estrutura e o comportamento de um sistema, podendo a estrutura ser dividida em estática e dinâmica [24,107].

A estrutura estática, ou distribuição do sistema, é definida em termos de *Blocos* ("Blocks") e de *Canais* ("Channels") de interligação desses blocos, sendo o Bloco visto no modelo como uma "caixa preta". Com este conjunto de conceitos, é possível a definição de interfaces claras entre as unidades estruturais do sistema. Os *Blocos* podem residir no interior de outros *Blocos,* podendo a sua estrutura ser recursivamente refinada até um ponto onde é descrita a sua estrutura dinâmica. A estrutura dinâmica é definida com a ajuda do conceito de *Processo* ("Process"), um elemento independente que reage a estímulos sob a forma de *Sinais* ("Signals"). Os *Sinais* viajam entre dois processos ao longo de "Caminhos para Sinais" ("Signal Routes").

Na implementação dos processos, o SDL baseia-se no conceito de um sistema de máquinas de estado finitas e estendidas em comunicação ("comunicating extended finite-state machines - CEFSM"). Estas máquinas de estado comunicam entre si e com o ambiente comum através de sinais trocados de uma forma assíncrona, os quais são colocados em filas de espera na entrada dos processos SDL.

Tal como na linguagem Estelle, as especificações em SDL apresentam um carácter informativo onde podem ser observados tanto o comportamento global do sistema como os detalhes de implementação. Além disso, recorrendo a compiladores e simuladores, é possível a produção de protótipos e a realização de simulações usando especificações em SDL.

6.2.4 Conclusão

Na breve apresentação das três TDF distinguem-se essencialmente dois tipos de descrições:

- a descrição operacional ("Operational semantics"), no Estelle e SDL, onde o comportamento de cada módulo é descrito sob a forma de uma sequência de comandos;
- a descrição do significado ("Denotational semantics"), no LOTOS, onde se descreve o comportamento de um determinado módulo definindo a sequência temporal dos eventos que ele capaz de executar sem necessariamente especificar como essas sequências são conseguidas.

Em [129] são analisadas as três Técnicas de Descrição Formal apresentadas acima, relativamente à sua capacidade de produzir implementações conformes com a especificação. São neste artigo realizadas diversas experiências, nas quais a especificação de um mesmo protocolo de comunicações é realizada nas diferentes linguagens (Estelle, LOTOS e SDL) e entregue a vários grupos de trabalho distintos para a realização da respectiva implementação (na linguagem ADA). Neste contexto são realizadas várias medidas com o objectivo de avaliar a adequação de cada uma das linguagens, tais como: erros de omissão ou de interpretação incorrecta; falha, ambiguidade ou erro na especificação de requisitos; projecto / implementação incorrecta da especificação apesar da correcta interpretação da mesma, e outras. São a seguir apresentadas algumas conclusões das quais se extraíram as de maior relevância para a selecção da TDF realizada no âmbito deste trabalho:

- não é possível tirar conclusões gerais relativas aos méritos de cada uma das linguagens de especificação, face ao reduzido número de experiências realizado;
- os programadores gostaram em particular dos mecanismos do LOTOS para caracterização da concorrência, embora reconhecendo a sua difícil utilização;
- os blocos construtivos usados na especificação da concorrência no Estelle e SDL não são suficientemente expressivos, tendo sido atribuída a causa dos erros na implementação às diferenças semânticas entre estes mecanismos e os do ADA;
- os utilizadores do SDL gostaram da sua representação gráfica, que permite visualizar o comportamento do sistema global;
- a representação gráfica deve contribuir para a melhor compreensão das três linguagens de especificação;
- as implementações melhores e mais estruturadas foram produzidas pelas equipas de trabalho que receberam a especificação em LOTOS, sendo de imediato seguidas por aquelas que as receberam em SDL.

A classificação dos dois tipos de especificações apresentada no início deste ponto, leva-nos todavia à exclusão do segundo tipo, ao qual pertence a linguagem LOTOS. De facto este tipo de especificação, que poderia ser realizada por um conjunto de *predicados* em Prolog [107], não é adequado aos mecanismos de implementação fornecidos no Simple++ onde os modelos são construídos através de mecanismos do tipo das máquinas de estado.

O tipo de especificação operacional é na realidade mais adequada à implementação no Simple++ (Estelle, SDL). A escolha do SDL surgiu assim de uma forma natural, como resultado imediato do facto de o Estelle não providenciar uma linguagem gráfica para a especificação do comportamento dos seus módulos.

O SDL oferece com efeito uma representação gráfica muito popular e de simples compreensão, premissa da maior importância para a comunicação entre os grupos de trabalho cooperando a diversos níveis num projecto de integração. Foi de facto a utilização

intuitiva de conceitos simples, tais como estado, entrada e saída que, juntamente com a sua representação gráfica, tornou o SDL muito popular na área das telecomunicações [107].

Não foi viável a utilização de ferramentas já existentes para a modelação em SDL, pelo facto de não tornar possível a requerida integração entre a modelação do equipamento de fabrico e dos fluxos de materiais associados e a modelação das aplicações de controlo e gestão dos fluxos de informação inerentes. O requisito de que a ferramenta de modelação em SDL fosse construída sobre o Simple++ foi viabilizado pelo facto de a sua representação recorrer a máquinas de estado finitas comunicando através de filas de espera, mecanismo este coincidente com o mecanismo de abstracção básico do Simple++.

6.3 Breve Apresentação do SDL

Actualmente, o SDL é conhecido principalmente nos meios ligados às telecomunicações onde foi originalmente desenvolvido, sendo o SDL-92 a sua última versão [133]. O campo de aplicação do SDL abrange no entanto domínios mais vastos, nomeadamente os sistemas de tempo-real, interactivos ou distribuídos, manipulando informação de comportamento e estrutura e cobrindo diferentes níveis de abstracção [130].

Esta linguagem foi projectada de modo a permitir a especificação / descrição do comportamento de um sistema em cooperação com o ambiente envolvente, assim como a especificação do seu comportamento interno. Esta facilidade permite que os diversos componentes do sistema sejam desenvolvidos de forma modular e em paralelo, característica particularmente interessante nos sistemas distribuídos. O SDL possibilita além disso o recurso aos conceitos de encapsulamento da informação e do comportamento, típicos das linguagems orientadas ao objecto [131].

O SDL cobre diversos níveis de abstracção, desde uma vista genérica global do sistema até uma vista muito detalhada de uma parte do mesmo. O SDL não pretende ser uma linguagem de implementação, embora seja possível a transformação de uma descrição em SDL numa linguagem de programação standard, como "C" ou "C++", permitindo a consequente geração de código executável [132].

A necessidade de uma completa integração de uma implementação do SDL no Simple++ levou ao seu estudo detalhado. O objectivo a atingir com a utilização do SDL no âmbito deste trabalho foi o de fornecer ao utilizador meios para modelar componentes da ASI, i.e. módulos de software de gestão / controlo fabril, nomeadamente "drivers", interfaces e núcleos de aplicações. Foi pois necessário projectar uma ferramenta de utilização eficiente e intuitiva, por forma a que o seu utilizador tirasse pleno proveito dos conceitos poderosos e das vantagens apresentadas pelo SDL.

6.3.1 Os Conceitos Fundamentais do SDL

Como foi já referido atrás, as duas principais características do SDL são a sua capacidade de descrever a estrutura e o comportamento de um sistema, podendo esta estrutura ser dividida em estática e dinâmica e representada hierarquicamente como ilustra a figura 6.1.

A estrutura estática, ou distribuição do *Sistema* ("System"), é definida em termos de *Blocos* ("Blocks") e de *Canais* ("Channels"), como se torna evidente na descrição do *Sistema* Exemplo na figura 6.1, sendo o Bloco visto no modelo como uma "caixa preta". Com este conjunto de conceitos simples, é possível a definição de interfaces claras entre as unidades estruturais do sistema. Os *Blocos* podem residir no interior de outros *Blocos,*

podendo a sua estrutura ser recursivamente refinada até um ponto onde finalmente se descreve a sua estrutura dinâmica.

A estrutura dinâmica do SDL é definida com a ajuda do conceito de *Processo* ("Process"), um elemento independente que reage a estímulos sob a forma de *Sinais* ("Signals"). O Sinais viajam entre dois processos ao longo de "Caminhos para Sinais" ("Signal Routes") (ver figuras 6.1 e 6.2).

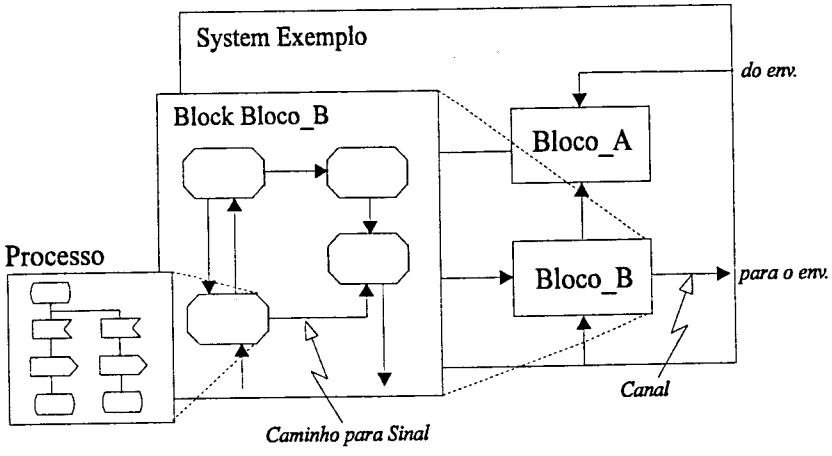


Figura 6.1 - A Hierarquia de uma Especificação em SDL

O comportamento do sistema é assim descrito no interior de um ou mais processos segundo o modelo das máquinas de estado finitas, modelo esse em que cada processo está em cada instante num estado bem definido. Através de um estímulo vindo do *Ambiente* exterior ("Environment", *env* na figura 6.1) ou do seu interior, a máquina de estado / processo é induzida a transitar para um outro estado. Este modelo é enriquecido através da utilização de um conjunto de conceitos poderosos, tais como a possibilidade de criar e destruir dinamicamente novos processos e de realizar chamadas a procedimentos durante uma transição de estado (figuras 6.2 e 6.4). Durante a transição de estado poderão assim ser executadas algumas actividades tais como, por exemplo, lançar novos processos, executar procedimentos, fazer cálculos, enviar sinais, etc..

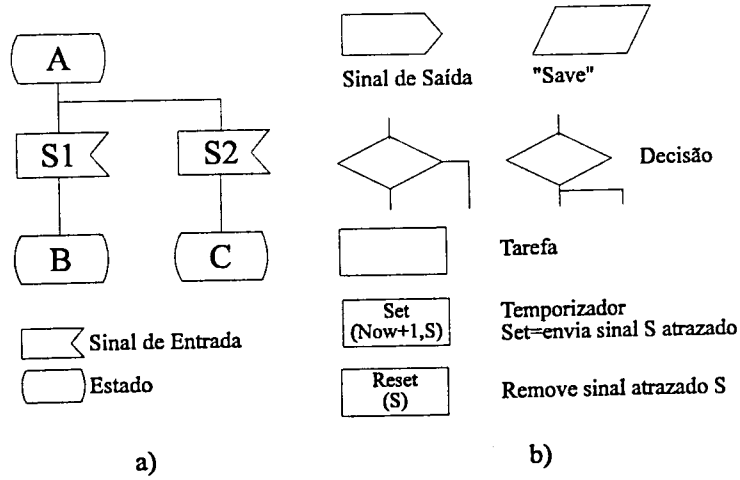


Figura 6.2 - Elementos da Máquina de Estados SDL

O SDL dispõe de tipos de dados pré-definidos (*boolean*, *char*, *integer*, *real*, *pid*, *duration*, *time*, *charstring*) que podem ser utilizados para definir novos tipos de dados abstractos ("abstract data types"). A definição de um tipo de dados requer a definição de

literais que definem os valores válidos do tipo, operadores para manipular os literais e axiomas que definem a aplicação dos operadores [133]. Deste modo é possível definir tipos de dados que se aproximam bastante do conceito de objecto.

Os sinais transportam informação entre processos, sendo o sinal na sua forma mais simples um semáforo ou uma etiqueta que estimula uma transição de estado ("transition tag"). No entanto, o sinal pode transportar outro tipo de informação, desde variáveis simples até tipos de dados abstractos. A existência de uma gama alargada de tipos de informação aceites pelos sinais facilita a especificação de formas mais elevadas de comunicação entre processos, e entre estes e o *Ambiente* onde se insere o sistema. No SDL, a troca de sinais é por natureza assíncrona, podendo a comunicação síncrona ser especificada com a ajuda de temporizadores e com a descrição do comportamento do processo.

A figura 6.2 apresenta a notação gráfica utilizada na construção da máquina de estados finita em SDL. Na figura 6.2 a) é claramente visível que, com a chegada de qualquer dos *sinais* de entrada ("insignal") S1 ou S2, o processo transita do estado A para o estado B ou C. Durante a transição, o processo pode comunicar com ele mesmo ou com o exterior, enviando um sinal de saída ("output"). O *Estado*, assim como os *Sinais* de *Entrada* e de *Saída* constituem blocos construtivos fundamentais da máquina de estados.

A figura 6.2 b) ilustra outros blocos construtivos SDL que permitem a realização de descrições mais complexas. O bloco de decisão controla o caminho da transição cuja escolha depende da avaliação de uma condição lógica. Esta construção corresponde à clássica construção de controlo "case" ou "if...then...else" nas linguagens de alto nível, diferindo aqui no facto de estar a modelar caminhos de controlo alternativos, e não o fluxo de controlo de execução de um programa.

A cada bloco construtivo *Tarefa* ("Task"), podem associar-se as atribuições que o caracterizam. Na figura 6.2 b) temos o exemplo de uma tarefa típica de temporização: ao temporizador é atribuído o valor de um determinado intervalo de tempo ao fim do qual o sinal indicado é devolvido ao processo. A utilização do temporizador permite de uma forma elegante modelar a evolução temporal do fluxo de controlo do sistema.

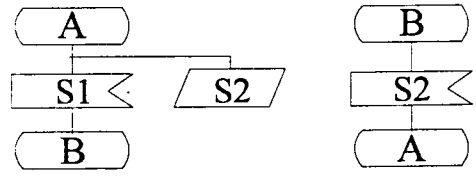


Figura 6.3 - Utilização do "Save"

A chegada de um sinal à fila de espera de um processo provoca sempre uma transição de estado, podendo todavia a máquina permanecer no mesmo estado caso o sinal não possa ser aceite no estado em causa. A transição seria assim dependente da ordem de chegada de sinais, ou em alternativa seria aumentada a complexidade da máquina de estados. Na figura 6.2 b) é apresentada a primitiva "Save" introduzida na máquina de estados para resolver este problema, a qual permite que o sinal que chega fora de ordem seja guardado para posterior utilização. Esta facilidade está ilustrada na figura 6.3. Neste exemplo, a chegada do sinal S2 antes de S1 não coloca problemas, uma vez que ele será guardado até que S1 chegue, sendo então utilizado na transição do estado B para A. Se nesta máquina de estados não for feito o "Save" do sinal S2 no estado A, a transição do estado A para o estado B dá-se sempre que S1 surgir, retornando ao estado A se entretanto S2 chegar. Assim sendo, a transição A→B→A apenas acontece se a sequência de chegada de sinais for S1→S2. O

"Save" de S2 no estado A permite que a transição A→B→A seja independente da ordem de chegada dos sinais.

A estrutura dinâmica do sistema pode também ser controlada durante a transição de estado. Os símbolos especificando estes comportamentos estão caracterizados na figura 6.4, indicando respectivamente o pedido para a criação de um novo processo de tipo determinado e a indicação de que um dado processo terminou.

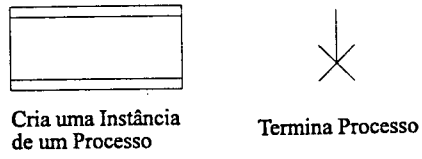


Figura 6.4 - Símbolos dinâmicos no SDL

6.3.2 As Primitivas de Descrição de um Sistema SDL, (SDL-G e SDL-PR)

Além da representação gráfica (SDL-G) exemplificada acima, o SDL tem a correspondente representação textual denominada SDL-PR ("Phrase Representation"). Neste ponto será feita uma breve descrição, acompanhada de exemplos ilustrativos, das duas representações do SDL. A sintaxe detalhada de ambas as representações pode ser encontrada em [24].

Representação do Sistema

O exemplo que se segue ilustra a representação do *Sistema* **Exemplo**, sendo claramente visível a sua interacção com o exterior. Assim sendo, temos o sinal **a** que chega do ambiente e entra no **Bloco_B** e o sinal **b** que sai do **Bloco_A** para o ambiente, respectivamente através dos canais **CH_env_B** e **CH_A_env**. A figura 6.5 a) apresenta a representação gráfica e a figura 6.5 b) a representação textual do *Sistema* **Exemplo**.

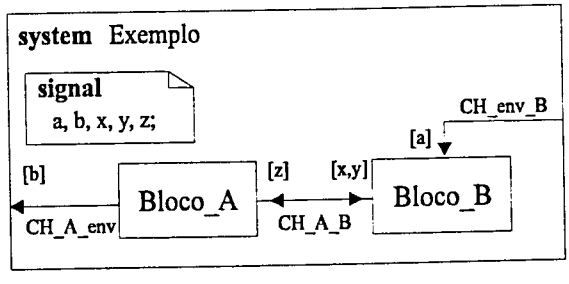


Figura 6.5 a) - Representação do *Sistema* Exemplo no SDL-G

```
system Exemplo;
   signal
     a, b, x, y, z;
   block Bloco_A referenced;
   block Bloco_B referenced;
   channel CH_A_env
     from Bloco_A to env with b;
   endchannel CH_A_env;
   channel CH_env_B
     from env to Bloco_B with a;
   endchannel CH_env_B;
   channel CH_A_B
     from Bloco_A to Bloco_B with x, y;
     from Bloco_B to Bloco_A with z;
   endchannel CH_A_B;
endsystem Exemplo;
```

Figura 6.5 b) - Representação do *Sistema* Exemplo no SDL-PR

Representação do Bloco

A representação do *Bloco* SDL é em tudo semelhante à representação do nível *Sistema*. É excepção o caso do bloco de mais baixo nível da hierarquia da especificação cujos filhos são *Processos* SDL. Deve notar-se que, a este nível, os sinais são trocados entre processos através dos Caminhos para Sinais ("signalroutes"). Torna-se todavia necessário identificar o *canal* no nível hierárquico superior ao qual o referido *signalroute* está ligado. No exemplo da figura 6.6, temos que ao *canal* CH_A_B entre os *Blocos* A e B está ligado o *signalroute* CH_env_AB.

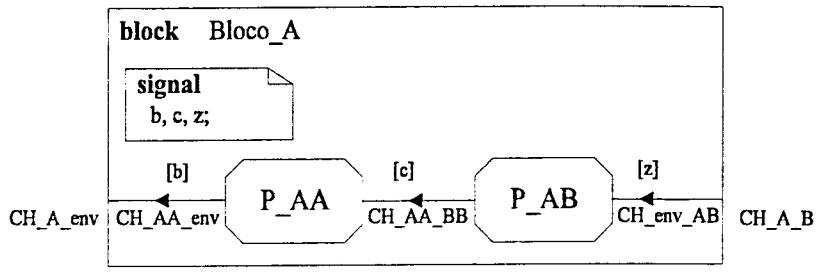


Figura 6.6 a) - Representação do *Bloco* Bloco_A no SDL-G

```
block Bloco_A;
   signal
     b, c, z;
   process P_AA referenced;
   process P_AB referenced;
   signalroute CH_AA_env
     from P_AA to env with b;
   signalroute CH_AA_BB
     from P_AB to P_AA with c;
   signalroute CH_env_AB
     from env to P_AB with z;
   connect CH_AA_env and CH_A_env;
   connect CH_env_AB and CH_A_B;
   endblock Bloco_A;
```

Figura 6.6 b) - Representação do *Bloco* Bloco_A no SDL-PR

Representação do Processo

O exemplo seguinte ilustra a forma como o *Processo* é declarado em SDL, sendo a representação dividida na área de declarações e no corpo do processo. É ainda apresentado o exemplo de uma máquina de estados na sua representação gráfica e textual correspondente (figura 6.7 a) e b)).

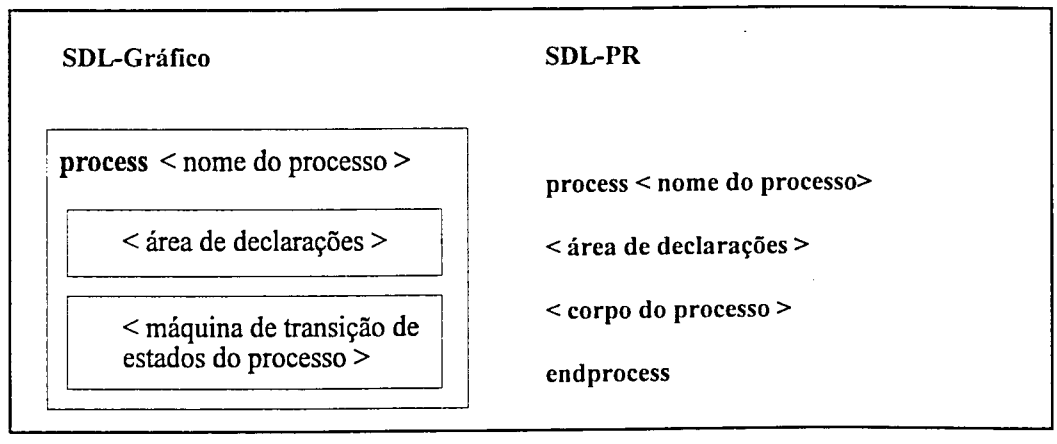


Figura 6.7 a) - Representação do *Processo* no SDL-G e SDL-PR

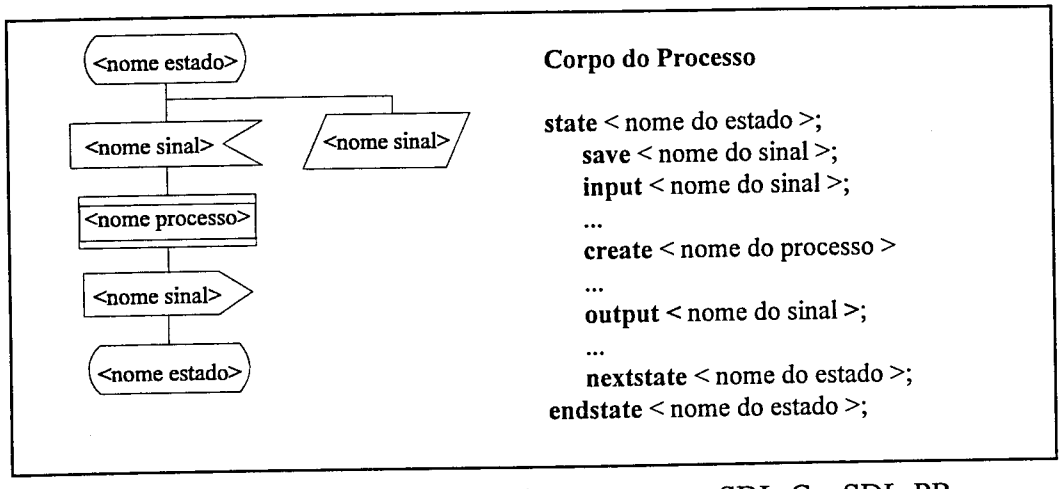


Figura 6.7 b) - Representação do *Processo* no SDL-G e SDL-PR

6.4 O Ambiente Integrado de Modelação no Simple++

A necessidade de ferramentas integradas de suporte ao desenvolvimento, teste e manutenção dos sistemas integrados de fabrico, resulta da complexidade destes sistemas cada vez mais construídos sobre arquitecturas distribuídas, onde as interacções entre aplicações fabris cooperantes, mas autónomas e independentes são críticas para o desempenho global do sistema.

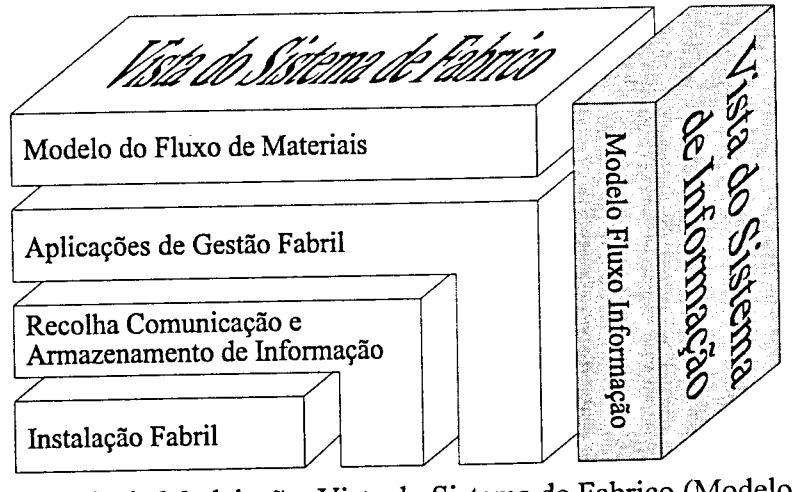


Figura 6.8 - Bancada de Modelação, Vista do Sistema de Fabrico (Modelo do Fluxo de Materiais) e Vista do Sistema de Informação (Modelo do Fluxo de Informação)

O ambiente integrado de modelação construído sobre o Simple++ no âmbito deste trabalho é de facto uma Bancada de Modelação (BM) que tenta responder a alguns dos requisitos que se colocam a essas ferramentas. A BM permite desenvolver o modelo do fluxo de informação que, juntamente com o modelo do fluxo de materiais da instalação fabril, forma um modelo integrado orientado ao objecto fornecendo uma descrição global do sistema. São incluídas não só vistas sobre os recursos de fabrico, os produtos e os fluxos de materiais, já disponíveis no PROFIT, como ainda vistas sobre o sistema de informação, agora disponíveis com a integração da ferramenta de modelação em SDL construída no Simple++.

A figura 6.8 ilustra o modelo em camadas do ambiente PROFIT, apresentando-se na camada superior o modelo do fluxo de materiais como elemento aglutinador do sistema e como sinóptico hierarquizado da instalação proporcionando pontos de entrada do utilizador na consola de gestão fabril. Possibilita ainda o acesso às diferentes aplicações do PROFIT, que se situam essencialmente no nível do controlo do fluxo produtivo e das operações de fabrico (MES / SFC), e um interface intuitivo de navegação no sistema. A vista sobre o sistema de informação surge ortogonalmente a esta estrutura em camadas fornecendo ao utilizador o modelo do fluxo de informação.

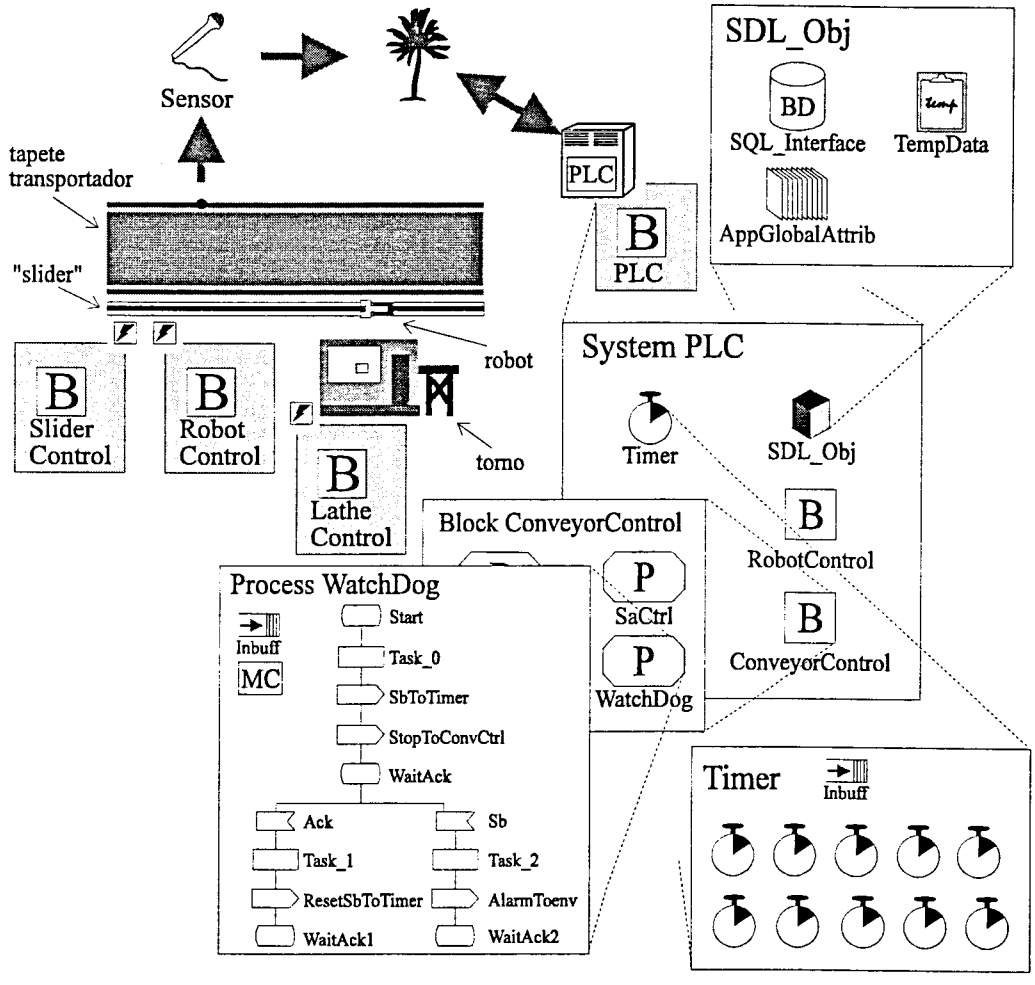


Figura 6.9 - SDL no Simple++, Especificação em SDL do Software de Controlo da Célula no PLC

A figura 6.9 ilustra através de um exemplo simples o ambiente integrado de modelação conseguido, um passo decisivo para atingir a modelação integrada dos fluxo de informação e de materiais, assim como para a obtenção de modelos integrados executáveis. A construção do novo ambiente de modelação e de simulação usando as primitivas do Simple++ será apresentada no decorrer deste capítulo.

Na BM os modelos SDL dos componentes / aplicações do sistema informação permitem, em conjunto com os modelos dos recursos de fabrico, produtos e fluxo de materiais, a construção de modelos globais de apoio ao projecto, implementação e exploração de sistemas integrados de gestão fabril [134]. Neste contexto, as aplicações

fabris podem ser desenvolvidas por forma a fornecer um conjunto de vistas particulares sobre os fluxos de informação e de materiais [135].

O exemplo na figura 6.9, extraído do modelo da célula de fabrico integrado do INESC-Porto, ilustra a forma como a integração dos dois modelos é realizada no Simple++. O modelo físico inclui neste exemplo uma secção de um tapete transportador, um robot e um torno de controlo numérico, assim como os controladores do "slider" (parafuso sem fim que desloca o robot paralelamente ao tapete), do robot e do torno ("Lathe") CNC. O modelo do fluxo de informação contém as especificações em SDL dos controladores do "slider", do robot e do torno CNC, assim como o modelo do software de controlo da célula de fabrico em execução no PLC.

A integração entre os modelos dos fluxos de materiais e de informação foi conseguida através da introdução de três novas entidades: o *Ambiente*, que permite a coexistência de várias especificações independentes comunicando entre si, e o *Sensor* e o *Actuador*, que permitem estabelecer a ponte entre o modelo físico (do fluxo de materiais) e o modelo de informação (do fluxo de informação). O *Ambiente* ("Environment") foi criado sob a forma de uma classe no Simple++ (representado pelo ícone com a forma de uma palmeira), por forma a permitir a integração de especificações de múltiplos *Sistemas* correndo em paralelo na BM e trocando mensagens (sinais) entre si que são encaminhadas pelo *Ambiente*. É óbvio que apenas uma instância dessa classe pode existir no interior de um modelo, a qual é responsável pelo encaminhamento correcto dos sinais que cada especificação lhe entrega. O *Sensor* existe, tal como o *Ambiente*, sob a forma de classe no Simple++, sendo todavia permitida a coexistência de várias instâncias deste objecto num mesmo modelo. O *Sensor* é usado para converter eventos originados no modelo físico em sinais inteligíveis para uma especificação SDL. O *Actuador* é usado na conversão de eventos com origem no sistema de informação em acções directas sobre o sistema físico. O *Actuador* actua deste modo como um "driver" de baixo nível da especificação SDL, permitindo que ela actue sobre o sistema físico sob a forma de chamada a uma função. No modelo, o *Actuador* é corporizado através de um método[1] do Simple++ definido pelo utilizador que actua sobre o modelo físico sempre que um sinal pré-determinado é enviado de uma máquina de estado SDL para o ambiente. A título de exemplo, um sinal *Alarm* enviado para o *Ambiente* pela especificação "LatheControl" poderá ser utilizado para alterar o aspecto gráfico do torno. A figura 6.9 ilustra também a especificação em SDL do *Sistema* PLC, existindo uma hierarquia similar para todas as especificações presentes no modelo.

O suporte do SDL no Simple++ foi conseguido tanto através da construção de novas classes, tais como *Sistema*, *Bloco* e *Processo*, como através da derivação de classes já existentes, como por exemplo a estação de trabalho. A construção das classes de forma a suportar a construção de uma especificação em SDL foi, como veremos, conseguida através da introdução de novos objectos inexistentes na especificação SDL convencional.

O Simple++, suportando a modelação por objectos, permite que o utilizador construa uma biblioteca de classes descrevendo as diferentes entidades típicas de uma instalação fabril, tais como máquinas, operadores, unidades de transporte, produtos, etc. Estas classes estão disponíveis para serem instanciadas no momento da criação de novos modelos fabris. A introdução de classes para a modelação de aplicações de software em SDL, permitiu a extensão deste conceito às novas entidades. Deste modo, as aplicações de software são

---

[1]Na construção de um modelo de simulação no Simple++, poderá ser necessário alterar o comportamento das primitivas estandard. Esta alteração é realizada através da associação de métodos a determinados eventos gerados por aquelas primitivas, sendo os métodos programas na linguagem SimTalk.

especificadas sob a forma de classes independentemente de um contexto, para mais tarde serem reutilizadas aquando da sua instanciação num modelo particular.

A figura 6.10 ilustra a obtenção da classe "Alarm" por derivação da classe *Sistema*. Na construção de uma nova especificação, começa-se por aplicar à classe *Sistema* a derivação de classes, obtendo uma nova classe *Sistema* derivada. Sobre esta é construída a especificação, por exemplo da aplicação de Alarmes (1), obtendo-se deste modo a nova especificação sob a forma uma nova classe ("Alarm"). Esta operação de derivação de uma classe pode evidentemente ser aplicada a uma aplicação já existente, i.e. uma nova aplicação de alarmes pode ser obtida directamente por derivação da classe "Alarm" existente na biblioteca do Simple++. O modelo orientado ao objecto facilita desta forma a construção de modelos particulares, permitindo que o utilizador obtenha por derivação tantas classes quantas as necessárias para construir a sua própria biblioteca de classes de aplicações. Estas novas classes derivadas estarão disponíveis da mesma forma que os blocos básicos, podendo blocos básicos e classes ser instanciados à medida das necessidades para a criação de novos modelos (ver (2) na figura 6.10). A especificação do software do PLC (figura 6.9) é uma instância da classe PLC existente na biblioteca de classes do Simple++. Esta especificação foi instanciada e adaptada ao contexto do modelo onde foi colocada através da configuração dos atributos globais (AppGlobalAttrib na figura 6.9). Os conceitos de derivação e de instanciação podem ainda ser estendidos à primitivas *Bloco* e *Processo*, facilitando assim a criação de novas especificações em SDL.

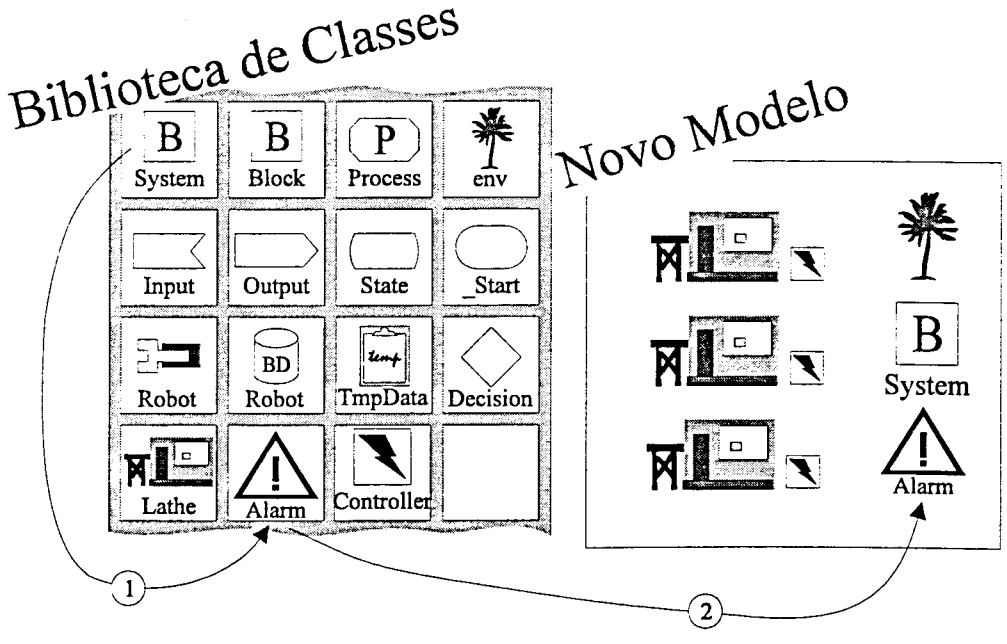


Figura 6.10 - A Derivação (1) e a Instanciação (2) de Classes no Simple++

6.5 Facilidades da Ferramenta de Modelação em SDL no Simple++

Serão apresentadas a seguir as primitivas SDL disponíveis na Bancada de Modelação (BM). A implementação de SDL no Simple++ suporta apenas um subconjunto das primitivas básicas do SDL. Na tabela 6.1, são identificadas as primitivas do Simple++ que são directamente utilizadas na construção das primitivas do SDL. São assim distinguidas as primitivas SDL que são derivadas directamente de uma primitiva básica do Simple++,

identificadas como ***Derivadas***, e aquelas que foram construídas de raiz usando naturalmente as primitivas básicas do Simple++, identificadas como ***Novas***. Este conjunto de blocos básicos SDL foi seleccionado por forma não só a permitir a construção de um protótipo demonstrativo dos conceitos apresentados, como ainda a validar a viabilidade da sua aplicação em casos concretos. Algumas facilidades não foram implementadas pelo facto de não existir, na versão do Simple++ usada na altura, um conjunto de funções do SimTalk (Linguagem de programação do Simple++) para a manipulação de objectos. São exemplo desta limitação funções permitindo eliminar dinamicamente um objecto físico (uma estação de trabalho, por exemplo), estabelecer dinamicamente ligações entre dois elementos (duas estações de trabalho), ou avaliar uma expressão no decorrer de uma simulação. A implementação do SDL no Simple++ apresenta outras particularidades, uma das quais é ilustrada por exemplo na figura 6.9, onde os identificadores **WaitAck**, **WaitAck1** e **WaitAck2** se referem ao mesmo estado (**WaitAck**). Este problema, cuja resolução é discutida mais à frente, existe pelo facto do Simple++ não permitir a coexistência de dois objectos com o mesmo nome tendo o mesmo nó da árvore como pai. Deve no entanto ser salientado que este tipo de funcionalidades, embora muito úteis no nosso ambiente integrado de modelação, não serão normalmente necessárias para o utilizador comum do Simple++. A actual implementação do SDL poderá ser melhorada na medida em que novas versões do Simple++ fiquem disponíveis.

Tabela 6.1 - Primitivas SDL Suportadas na BM

Primitiva SDL	**Classe do Simple++**	**Objectos necessários à funcionalidade da Primitiva SDL no Simple++**
System	Root_SDL_Block (***Nova***)	BuildAll(M,T), PR(M,T), SDL_Control(M,T), Signal(T), Help, Timer(M,O,T), SDL_Obj(M,T)
Block	SDL_Block (***Nova***)	SDL_Control(M,T), Signal(T)
Process	SDL_Process (***Nova***)	InBuff(O), SDL_Control(M,T), MessageControl(M), Transition(T), OutMessages(T)
Start	_Start (***Derivada***)	
Input	InSignal (***Derivada***)	
Output	OutSignal (***Derivada***)	
State	State (***Derivada***)	
Save	Embebida na classe State	
Decision	Decision (***Derivada***)	
Task	Task (***Derivada***)	

Na coluna da direita, são indicados entre parênteses os tipos dos objectos que compõem a nova classe SDL: M-Métodos, T-Tabelas, O-Primitivas para modelação de um elemento físico no Simple++

6.5.1 Classe Sistema

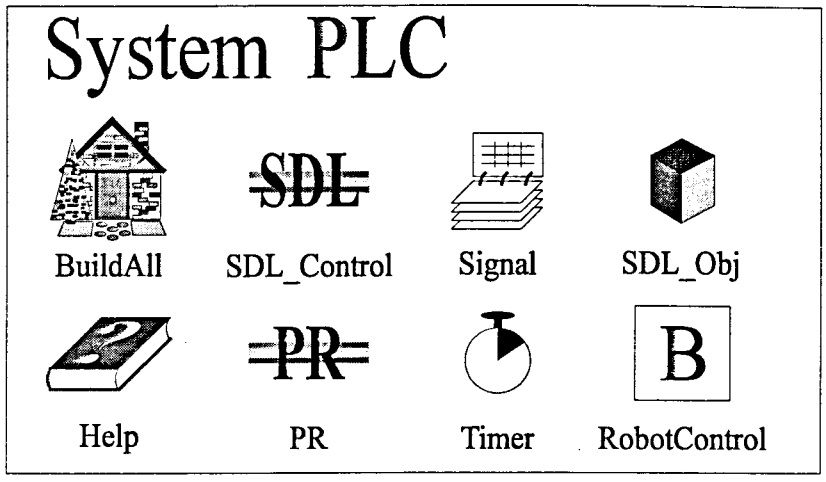


Figura 6.11 - Classe *Sistema* - contém uma instância da classe *Bloco*, RobotControl, e os objectos que disponibilizam funcionalidades de apoio à modelação e simulação

Sendo bloco construtivo do SDL, a classe *Sistema* (figura 6.11) pode ser seleccionada na biblioteca de classes do Simple++ e instanciada em qualquer ponto de um novo modelo. No exemplo da figura 6.9 (página 168), a classe *Sistema* foi instanciada para permitir construir o modelo em SDL do PLC. Várias instâncias da classe *Bloco* podem ser colocadas no interior de uma primitiva da classe *Sistema*, nomeadamente ao definir uma nova classe de aplicação ou ao modificar uma instância da classe *Sistema*. Na figura 6.9, os blocos RobotControl e ConveyorControl são instâncias da classe *Bloco* colocadas no interior de uma classe *Sistema*. No nível *Sistema* de uma especificação SDL no Simple++ existe um conjunto de objectos que não existem numa especificação SDL convencional, os quais dão acesso a funcionalidades de apoio às tarefas de modelação e simulação:

- **SDL_Control** encapsula os métodos responsáveis por iniciar a instância da classe *Sistema*.
- **Signal** é uma tabela iniciada pelo **SDL_Control**. Esta tabela, preenchida durante a execução dos métodos chamados no interior do objecto **BuildAll**, contém a informação necessária à geração da especificação SDL-PR ("Phrase Representation") do nível sistema.
- **BuildAll** é responsável pela construção automática de todas as tabelas do modelo, invocando métodos existentes no interior da classe SDL_Control para os diversos níveis da especificação, i.e. *Sistema*, *Bloco* e *Processo*. As tabelas nos diversos níveis *Sistema*, *Bloco* e *Processo* são necessárias tanto para controlar a simulação como para gerar a descrição SDL-PR da especificação.
- **PR** contém os métodos usados para a geração de um ficheiro de texto contendo a especificação no formato **SDL-PR**. A geração dessa especificação é realizada com base na informação contida nas tabelas "Signal" dos níveis *Sistema* e *Bloco*, bem como na estrutura das máquinas de estado dos vários processos da especificação. A geração do SDL-PR permite que a especificação possa ser exportada para ferramentas CASE externas, por exemplo o GEODE [136].
- **Help** contém um conjunto de objectos de texto do Simple++ com informação de ajuda à utilização do SDL no Simple++.
- **Timer** fornece à especificação um serviço de temporização.

- **SDL_Obj** esconde os atributos e outros objectos (usados, por exemplo, para encapsular o acesso a estruturas de dados complexas, tabelas, ficheiros ou bases de dados) utilizados na especificação.

6.5.2 Classe Bloco

A classe *Bloco* é instanciada no interior da classe *Sistema*, ou recursivamente no interior de uma instância de uma classe *Bloco*. Dois elementos coexistem no interior da classe *Bloco*: o *SDL_Control*, contendo procedimentos de iniciação, e o *Signal* contendo, tal como na classe *Sistema*, informação usada para a geração do SDL-PR.

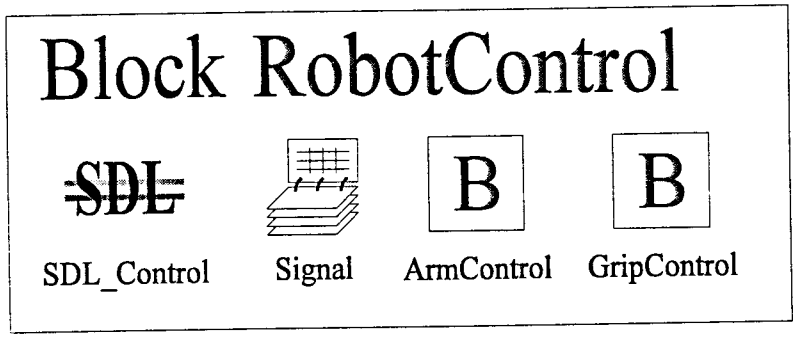


Figura 6.12 - Classe *Bloco* - contém duas instâncias da classe *Bloco* além dos objectos SDL_Control e Signal

No interior de uma instância da classe *Bloco* podem ser instanciados elementos da classe *Processo* ou da classe *Bloco*. No entanto, nenhuma instância da classe *Processo* pode coexistir com uma instância da classe *Bloco*.

6.5.3 Classe Processo

Os Objectos na Classe Processo

Esta classe apenas pode ser instanciada no interior de uma instância terminal da classe *Bloco*, i.e. as instâncias da classe *Processo* são as folhas da árvore de especificação SDL. No interior desta classe poderão ser instanciados os objectos ou primitivas apresentados na tabela 6.2. Esta tabela apresenta as primitivas SDL suportadas, as quais foram criadas por derivação da classe "Estação de Trabalho" do Simple++. Esta primitiva do Simple++ necessitou naturalmente de alterações face aos requisitos do SDL, sendo cada um dos objectos dotado dos atributos necessários à execução da especificação ou à geração de SDL-PR. Um exemplo típico é o atributo ClassID, que identifica a classe do objecto em questão.

A classe "_Start" surge instanciada sempre com o mesmo nome "Start", nome reservado no SDL. A coincidência de ser também um nome reservado no Simple++ leva a que o nome da classe seja precedido do símbolo "_".

A classe "State" no Simple++ tem algumas particularidades, não sendo possível ter dois objectos com o mesmo nome. Como vimos anteriormente no exemplo da figura 6.9 (página 168), os estados **WaitAck**, **WaitAck1** e **WaitAck2** são de facto o mesmo estado,

motivo pelo qual o *token*[2], que indica através da sua presença o estado actual da máquina de estados, deve ser colocado em **WaitAck** sempre que chegue a **WaitAck1** ou **WaitAck2**. Para tal, o atributo *Loop* contém a referência para o objecto correspondendo ao estado para onde o *token* deve ser enviado. O atributo *Loop* do estado **WaitAck1** e **WaitAck2** deve então ser a referência para o estado **WaitAck**. No caso de **WaitAck** este atributo deverá ter o valor *Null*, uma vez que o token deve lá permanecer até que um novo sinal chegue ao processo. A figura 6.13 ilustra os estados da tabela *OutMessages*, que relaciona o estado final e os *Sinais* a serem enviados como resultado da transição; onde existir a necessidade de execução do *Loop* os estados são referenciados com um *ok* na coluna *MakeLoop*. Na mesma figura, a tabela *Transition* reflecte por seu lado a relação entre o estado inicial (linha), o sinal de entrada (coluna) e o estado final da transição. A existência de um bloco de decisão é assinalada por um 'ok' na columa *Decision* desta tabela, que obriga à verificação de uma condição na transição. Uma outra situação ocorre por exemplo sempre que exista a necessidade de executar o *Loop*, assim, temos o exemplo da transição dos estados **WaitAck1** e **WaitAck2** para o estado **WaitAck** que é indicada na tabela *Transition* na coluna associada ao sinal *Loop*.

Tabela 6.2 - Primitivas SDL ao nível do *Processo*

Primitiva SDL no S++ (Classe)	Atributo da Classe e valor(es) que toma	Sintaxe para o nome do Objecto instanciado
_Start	ClassID="Start"	Start
State	ClassID="State"	<string>
	Save = {sinais}	
	Loop=Objecto	
InSignal	ClassID="InSignal"	<string>
OutSignal	ClassID="OutSignal"	<string>To<string>|env
Decision	ClassID="Decision"	<string>
	Decision= <texto a surgir na SDL-PR>	
Task	ClassID="Task"	<string>

O "Save" é mais uma situação particular. Os sinais a "Salvar" para um determinado estado são armazenados numa lista, atributo da classe do *State*, denominada "SaveStates". Este atributo contém assim a lista dos sinais a guardar para o próximo estado.

Na classe OutSignal foi exigido o endereçamento explícito para o encaminhamento dos sinais enviados a partir de um processo SDL. Não foi possível implementar o endereçamento implícito, desejável e inerentemente mais simples, devido a restrições do Simple++.

Na classe "Decision" surge um atributo onde o utilizador pode escrever uma condição a ser incluída no texto da especificação SDL-PR. A condição tem apenas um valor descritivo,

[2]O conceito de token não existe em SDL, tendo todavia sido introduzido com o objectivo de facilitar ao utilizador a visualização do comportamento da máquina de estados no decorrer da simulação.

não tendo qualquer significado para a execução da especificação por não ser possível a sua interpretação do decorrer da simulação.

Ao nível do *Processo*, o utilizador especifica o comportamento dinâmico através da construção de uma máquina de transição de estados usando as primitivas básicas indicadas na tabela 6.2. O utilizador principia por instanciar os elementos desejados ("Start" primeiro, seguido pelos estados, tarefas, sinais de entrada e saída), estabelecendo depois as ligações entre esses elementos que vão definir a máquina de estados. A figura 6.13 apresenta a classe *Processo*, assim como um conjunto de blocos básicos necessários à completa funcionalidade desta classe no Simple++:

- **Inbuff** é um "buffer" usado para modelar a fila de espera (FIFO) de entrada no processo. Esta fila tem capacidade limitada que pode ser aumentada sempre que necessário durante a execução da simulação, dependendo a sua capacidade máxima das restrições impostas pelo Simple++ ou pelo hardware disponível. Esta restrição é importante uma vez que no modelo conceptual do SDL esta fila deve ter capacidade infinita.

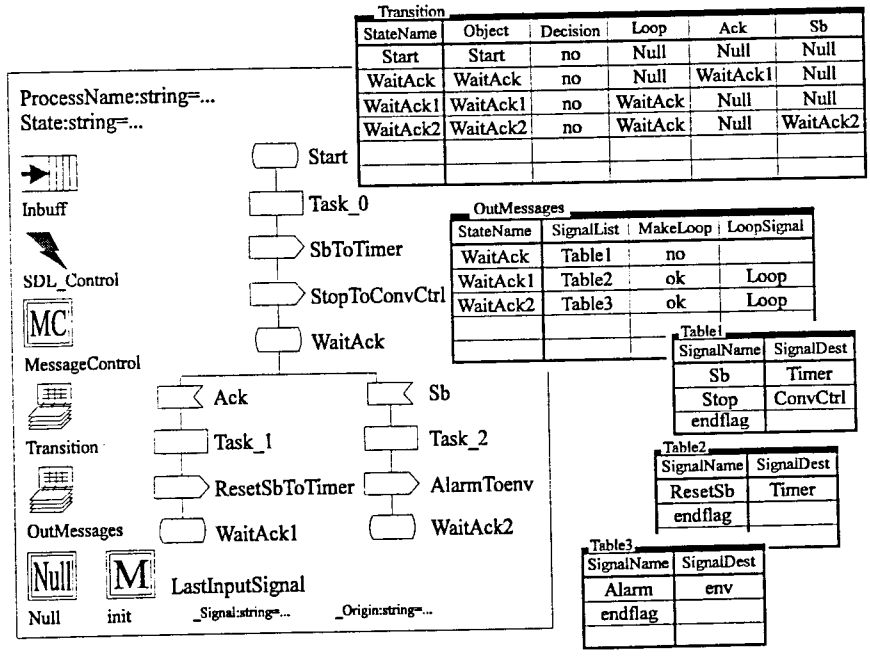


Figura 6.13 - Classe *Processo* - máquina de estados, objectos de controlo e tabelas **OutMessages** e **Transition**

- **SDL_Control** contém um conjunto de métodos de iniciação para a construção automática das tabelas **Transition** e **OutMessages**, de geração do SDL-PR, de controlo da simulação e da animação do modelo. A informação contida nestas tabelas é usada para controlar a animação da máquina de estados do processo durante a simulação.
- **Transition** é uma tabela onde é definido o estado seguinte na máquina de transição de estados, em função do estado actual e do *Sinal* de entrada.
- **OutMessages** é uma tabela que relaciona o estado final e os *Sinais* a serem enviados como resultado da transição corrente.
- **MessageControl** é um método suporta a da execução das *Tarefas* ("Task") SDL, além do tratamento de decisões e a iniciação dos temporizadores, usando a linguagem de programação do Simple++. Esta abordagem à execução das tarefas

SDL recorrendo ao SimTalk foi realizada desta forma porque: (1) o Simple++ não permite que o nome dos objectos contenham caracteres, tais como espaços ou operadores, motivo pelo qual ao utilizador não é permitido escrever qualquer expressão matemática ou chamada de função como nome do objecto; (2) embora o problema anterior possa ser remediado escrevendo a expressão definida pelo utilizador como atributo de objecto, não existia uma função do Simple++ que avaliasse a referida expressão no decorrer de uma simulação. Justifica-se assim o motivo pelo qual o utilizador é obrigado a codificar a tarefas SDL usando a linguagem de programação do Simple++ (SimTalk) no método *MessageControl*, cuja mecânica interna será explicada a seguir.

6.5.4 A Utilização do Método "MessageControl"

Este método é a única ligação existente entre a máquina de transição de estados de um *Processo* e os dados manuseados pela especificação SDL. De facto, embora a primitiva *task* seja suportada na descrição gráfica da máquina de transição de estados ela não tem qualquer papel activo durante a própria simulação, sendo esse papel desempenhado pelo método *MessageControl* e pelo código SimTalk nele embebido. A primitiva *task* é contudo levada em consideração no decorrer da geração do SDL-PR.

É em seguida analisada com algum pormenor a utilização do método *MessageControl* recorrendo a um exemplo. Neste exemplo é descrito o processo **WatchDog**, cuja funcionalidade consiste em disparar um alarme no caso de o sinal **Ack** não chegar dentro de um período de tempo pré-estabelecido em resposta ao *Sinal* **Stop**. Na figura 6.14, o estado actual da máquina de estados do *Processo* SDL é indicado pela posição do *token* (Ⓣ), que permite visualizar a evolução do estado do processo durante a simulação.

O exemplo ilustra o modelo e executado o processo **WatchDog**, cujo comportamento é resumidamente descrito a seguir. Quando a simulação arranca, ocorre uma mudança de estado de **Start** para **WaitAck** corporizada na interface gráfica pelo movimento do *token*. Nesta transição de estado, além de ser executada a tarefa **Task_0**, é enviada a ordem de paragem **Stop** ao processo **ConvCtrl** e o sinal **Sb** para activar o temporizador do **WatchDog**. Segue-se então um período de espera que acaba quando o sinal **Sb** chega do temporizador, sendo então enviado um alarme, ou quando o sinal **Ack** chega, sendo nesse caso cancelado o pedido formulado ao temporizador.

A figura 6.14 ilustra em detalhe o que se passa a partir do instante em que a simulação arranca e um *Sinal* chega ao *Inbuff* do processo **WatchDog**. Quando a simulação arranca ocorre, conforme se referiu, uma mudança de estado decorrente de uma transição (1 na figura 6.14) que é executada de imediato, levando o *token* a mover-se do estado **Start** para o estado **WaitAck**. A tarefa **Task_0** é assim executada no método *MessageControl* (1a na figura 6.14 e ***iv*** na figura 6.15), correspondendo esta tarefa ao tratamento do envio do sinal **Sb**. Neste ponto o programador utiliza o atributo global **WaitAck** (***iv***, na figura 6.15) contendo o valor do intervalo de tempo ao fim do qual o sinal **Sb** deverá ser enviado de volta do temporizador para o processo **WatchDog**. Imediatamente após a tarefa **Task_0**, onde o atributo do sinal foi colocado igual a **WaitAck** (M na figura 6.14), o sinal é de facto enviado para a fila de entrada *InBuff* do temporizador *Timer* (1b na figura 6.14). É aqui identificado um cronómetro livre, o seu tempo inicial é igualado ao valor do atributo recebido no sinal, e o sinal é movido para esse cronómetro ficando lá a aguardar a contagem decrescente (1c na figura 6.14). Após esta transição pode surgir um qualquer de dois

eventos durante a contagem decrescente do temporizador: o sinal **Ack** chega ao *InBuff* do processo **WatchDog** (2 na figura 6.14) levando a que seja feito o "reset" do temporizador onde **Sb** permanecia (2b na figura 6.14); ou chega o sinal **Sb** (T e 3 na figura 6.14), provocando a execução da tarefa **Task_2** (3a na figura 6.14) e o envio de um sinal de alarme para o ambiente (3b na figura 6.14).

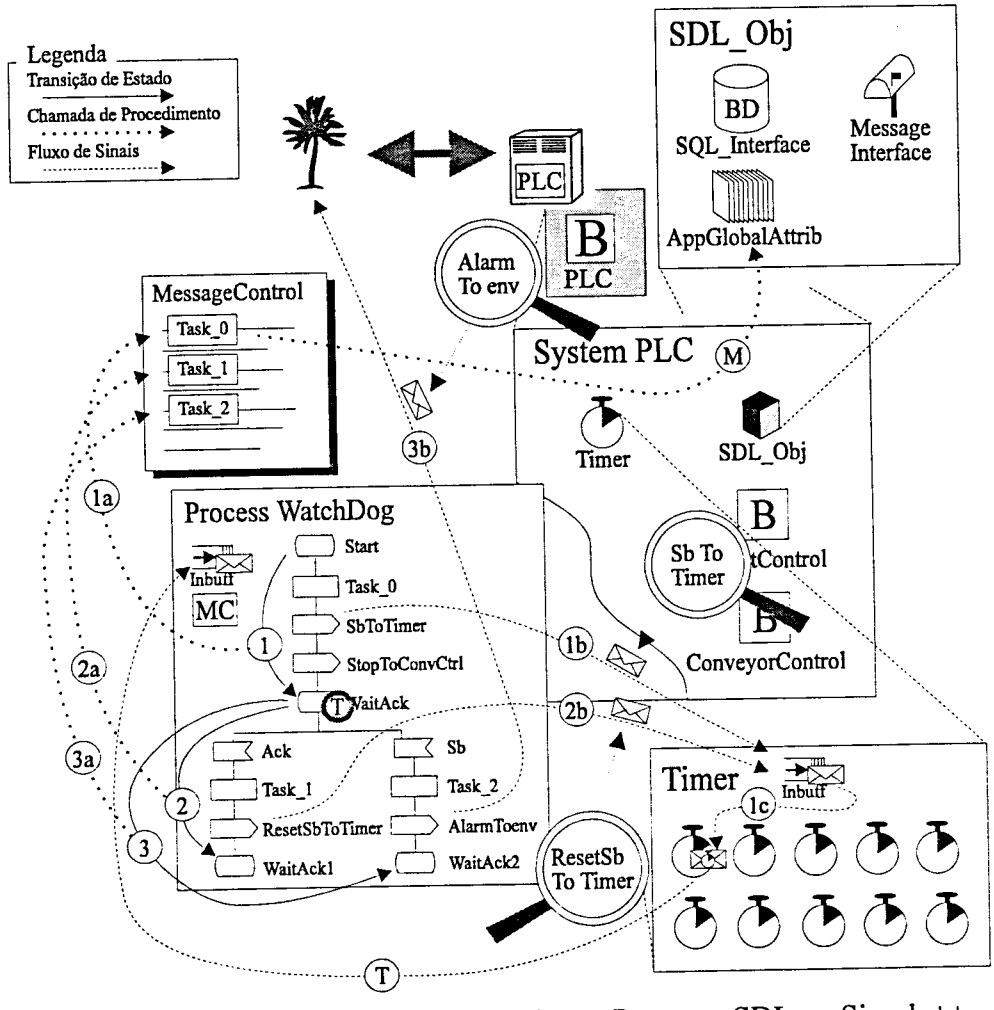


Figura 6.14 - Exemplo da Simulação de um Processo SDL no Simple++

Na figura 6.15 pode ser visto o tratamento do bloco de Decisão, que como já vimos, não tem papel activo no decorrer da simulação. Contudo, tal como para o bloco *Task*, a informação contida no atributo *Decision* do bloco decisão é usada na geração do SDL-PR (ver tabela 6.2). No decorrer de uma simulação, o método *MessageControl* é invocado sempre que exista um bloco de decisão, sendo passado o "?" no parâmetro *OutSignal* ilustrado em ***ii***. No exemplo, ***ii*** está em comentário porque, neste caso particular, quer chegue **Ack** ou **Sb** não existe dúvida possível quanto ao estado final na transição (dado que **WaitAck**, **WaitAck1** e **WaitAck2** identificam o mesmo estado). Esta parte do método *MessageControl* é útil sempre que para um único sinal de entrada tivermos mais de um estado final, por exemplo sempre que exista um ou mais blocos de decisão. Não dispondo o Simple++ de uma função que avalie uma condição de decisão, este método é invocado para testar a referida condição e desse modo decidir qual o estado final em função do sinal e valor dos atributos com ele recebidos. Por outro lado, se os atributos do sinal a serem enviados dependem de algum modo do sinal recebido, o método *MessageControl* pode também ser utilizado tal como no caso do atributo do sinal **Sb** (***iv***). Para maior simplicidade

na utilização recomenda-se que todas estas tarefas sejam convertidas na execução de um único método (M na figura 6.14) que estará disponível no interior do objecto *SDL_Obj*, existindo pois ao nível *Sistema* da especificação. O ponto ***v*** na figura 6.15 ilustra a forma como o utilizador pode configurar tanto o atributo do *Sinal* **Alarm** como a informação de encaminhamento, para que o *Ambiente* possa fazer a correcta entrega da mensagem (*Root.Alarm.receive.handler* indica o caminho no modelo até ao processo *handler* no interior da especificação SDL da aplicação de **Alarm** no interior do bloco **receive**). Se existirem atributos calculados a serem enviados pelo *Sinal* **Alarm**, o cálculo poderá ser realizado em ***iii***, sendo a secção ***v*** apenas usada para colocar o resultado desse cálculo na estrutura de dados associada à mensagem.

```
Método MessageControl
(obj :Object ; InSignal :string; OutSignal
:string) :string
-- obj contém o endereço da mensagem
-- quando este método é chamado
-- por "OutSigList", ou pelo "Dispatcher"
-- o método que chama pede para retornar
-- o estado seguinte sempre que
-- OutSignal="?"
is
local
i            : integer;
tmp,_token   : object;
SDL          : object;
do
SDL := SDL_Control.SDL_OBJ;        (i)
-- SDL aponta para o objecto SDL_Obj
-- existente em cada Classe Sistema
result := "";
if OutSignal = "?" then
-- decide qual o próximo estado
    if InSignal = "Ack" then
    -- result := "WaitAck1";          (ii)
    end;
else
-- trata os sinais a serem enviados
    if InSignal = "Sb" then
     -- Executa tarefa associada ao
     -- sinal de entrada Sb (Task_2)  (iii)
    end;
    if OutSignal = "Sb" then
     obj.NumeroAtrib :=1;             (iv)
     obj.Attributes[1]:=
      SDL.AppGlobalAttrib.WaitAck;
    end;
    if OutSignal = "Alarm" then
     obj.NumeroAtrib :=1;             (v)
     obj.Attributes[1]:=
      "PLCLateAck";
     obj.routing :=
      Root.Alarms.receive.handler;
    end;
end;
return;
```

Figura 6.15 - Exemplo da estrutura genérica do Método *MessageControl*

As classes de objectos instanciadas no interior do *SDL_Obj* devem ser construídas de forma a que cada *Tarefa* (*Task*) no SDL corresponda à execução de um único método. Assim, a primitiva *Tarefa* é apenas útil na geração do ficheiro de texto SDL-PR onde o nome da *Tarefa* no modelo surge em comentário. Desde que a cada tarefa na especificação SDL corresponda um método de um objecto no interior do *SDL_Obj*, e que cada um destes objectos tenha a correspondente imagem, por exemplo numa classe em C++, será simples a obtenção do esqueleto do código da aplicação partindo da descrição SDL-PR, assim como a obtenção do código final a ser compilado.

6.5.5 Serviço de Temporização

O conceito de tempo existe no Simple++, uma ferramenta de simulação controlada por eventos. Como funcionalidade básica é fornecida ao utilizador uma caixa de diálogo onde é possível ajustar a velocidade da simulação: variando desde uma velocidade mais lenta do que o "tempo-real", com incrementos sucessivos, até "o mais rápido possível", i.e.

velocidade limitada pelos recursos computacionais disponíveis. A animação gráfica do modelo pode ser "desligada" para que se obtenha uma maior velocidade de execução da simulação.

A linguagem SDL fornece por seu lado ao utilizador uma primitiva de temporização que pode ser utilizada por um processo para activar ou desactivar um temporizador em particular. No desenvolvimento da ferramenta de modelação SDL no Simple++, foi construído um servidor de temporizações disponível para qualquer máquina de transição de estados dentro de uma especificação, o *Temporizador* (*Timer*) no interior da classe *Sistema* (objecto representado na figura 6.11 da página 172, pelo ícone com a forma de cronómetro etiquetado *Timer*). As primitivas de activação (*Set*) e desactivação (*Reset*) estão disponíveis ao nível do processo através do envio de uma mensagem de *Set* ou *Reset* ao servidor de temporizadores. A figura 6.14 (página 177) ilustra estas primitivas ao colocar a máquina de estados num estado de espera, no qual aguarda o sinal **Ack** após ter enviado o sinal **Sb** para o Temporizador (**SbToTimer**, 1b na figura 6.14), durante um intervalo de tempo especificado no método *MessageControl*. Se o sinal **Ack** chega antes de o sinal **Sb** ser recebido do temporizador, então este deve ser desactivado através do envio de uma mensagem com o sinal **ResetSb** (**ResetSToTimer**, 2b na figura 6.14). Por outro lado, se **Sb** chega primeiro é enviado um alarme para o *Ambiente* (**AlarmToenv**, 3b na figura 6.14). A figura 6.16 complementa a descrição destas duas situações através de um Diagrama de Sequência de Mensagens (MSC-"Message Sequence Chart") apresentando a troca de sinais entre o processo e o temporizador.

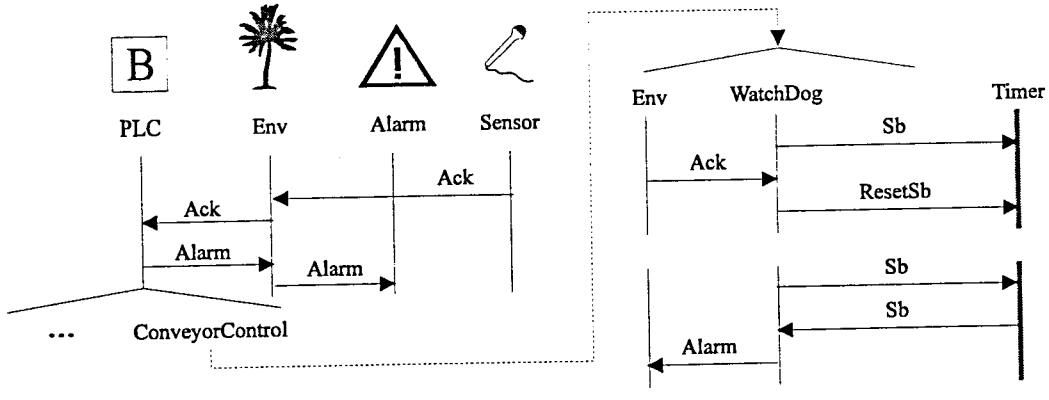


Figura 6.16 - Utilização do Temporizador, "Message Sequence Chart"

O serviço de temporização é constituído por um conjunto de cronómetros (figura 6.14 na página 177). Numa situação ideal, um cronómetro deveria ser criado dinamicamente sempre que fosse necessário e eliminado logo que terminasse a sua contagem decrescente e devolvesse o sinal. Esta solução não foi no entanto possível por não existir uma função no Simple++ que permitisse eliminar o temporizador durante a simulação, pelo facto de este ser um objecto físico no ambiente SimTalk. A solução adoptada garante um número mínimo de temporizadores, o qual pode ser aumentado dinamicamente durante a simulação em caso de necessidade.

6.5.6 Rastreio Global e ao Nível do Processo

A validação do comportamento da especificação é sem dúvida uma das tarefas de fundamental importância que se segue à sua construção. O facto de a especificação ser executável facilita esta tarefa, permitindo que ela seja testada e validada.

A execução interactiva da especificação permite ao utilizador verificar: se as sequências de teste estabelecidas na especificação informal são ou não permitidas na execução da especificação formal; se as sequências de teste obtidas na execução da especificação formal estão incluídas na descrição informal; se as sequências de teste que não foram especificadas informalmente são ou não aceites pela especificação formal. A execução passo-a-passo pode ajudar a que, para um determinado estado do sistema, o utilizador teste todas as alternativas possíveis de passagem por esse estado. Sempre que necessária uma interacção com o ambiente ela pode ser conseguida através da construção de um diálogo com o utilizador através do qual ele pode introduzir / receber a informação necessária.

A existência na Bancada de Modelação (BM) de um modo de simulação passo-a-passo, bem como a possibilidade de activar uma função de rastreio (*Trace*) associada ao envio e à recepção de sinais de um processo, providencia as facilidades básicas para a validação interactiva passo-a-passo da especificação executável. A informação de rastreio poderá ser directamente visualizada no terminal ou gravada num ficheiro de texto. Se a especificação for muito complexa, o utilizador poderá construir um conjunto de sequências a ser executadas com recolha de informação de rastreio naquele ficheiro para posterior análise.

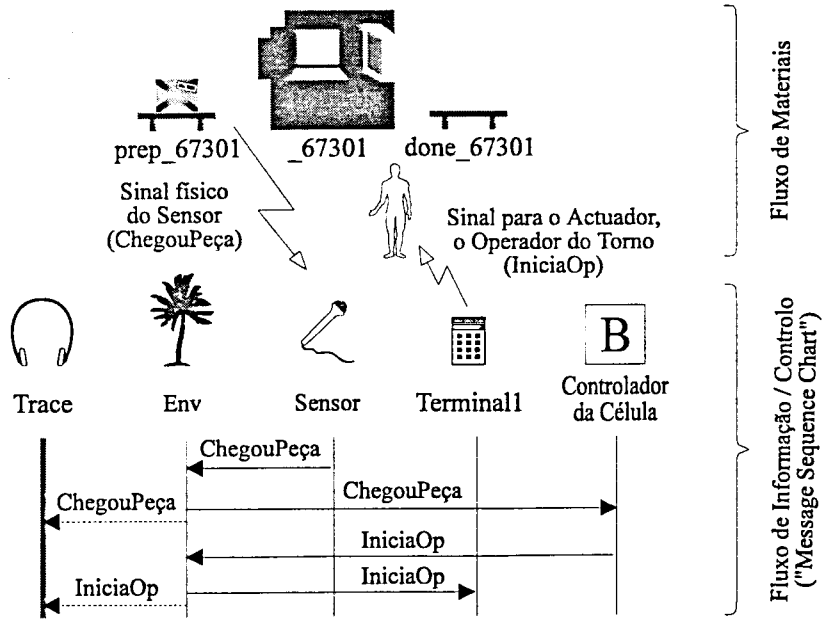


Figura 6.17 - Utilização do *Trace* para a verificação da interacção entre múltiplas aplicações

A integração entre múltiplas especificações deverá também ser testada, sendo obviamente de importância crucial no desenvolvimento integrado de aplicações fabris autónomas e cooperantes. O suporte para esta tarefa é oferecido ao utilizador através uma função de rastreio global que pode ser activada ou desactivada. Quando activada, todas as mensagens trocadas entre as várias especificações SDL no modelo são, tal como no caso anterior, visualizadas no terminal ou gravadas num ficheiro de texto com o formato do Diagrama de Sequência de Mensagens [136]. A figura 6.17 ilustra esta situação para o exemplo da célula de fabrico integrado, apresentando a chegada de um sinal vindo de um sensor via *Ambiente* para o Controlador de Célula. Como podemos constatar, o *Ambiente* notifica a função de rastreio (*Trace*) sempre que existe uma troca de sinais, podendo a sequência de eventos ser apresentada sob a forma de um MSC.

6.5.7 Avaliação da Implementação do SDL no Simple++

A escolha da linguagem de descrição formal a implementar no Simple++ teve como requisito base a viabilidade de uma implementação de facto em toda a sua extensão. A escolha do SDL pressupôs pois a viabilidade da construção de uma ferramenta de modelação sobre o Simple++.

Com o objectivo de avaliar o nível até ao qual foi levada a cabo a construção da ferramenta de modelação em SDL, propõe-se dividir o problema da especificação em SDL nos seus três aspectos fundamentais: o comportamento, a estrutura (estática e dinâmica), e os dados.

Na avaliação do protótipo desenvolvido, não deve ser esquecido o objectivo do trabalho realizado: a demonstração da viabilidade da construção de uma ferramenta de modelação do sistema de informação fabril, tendo como base uma linguagem de descrição formal, e a sua completa integração com uma ferramenta existente orientada à modelação do fluxo de materiais. O resultado fundamental daí decorrente seria a disponibilização de uma Bancada de Modelação oferecendo um ambiente integrado de modelação dos fluxos de materiais e de informação.

Comportamento

A especificação do comportamento está relacionada com a especificação da máquina de estados de cada processo SDL. Neste aspecto os objectivos principais foram atingidos no primeiro protótipo desenvolvido. Existem no entanto melhorias que devem ser introduzidas na primeira versão da ferramenta. Entre elas conta-se a utilização:

- mais racional dos blocos de *Decisão*, usando uma nova função existente na última versão (3.0) no Simple++ (*evaluate*), que permite a interpretação e execução no decorrer da simulação de uma expressão escrita num *campo* do tipo texto;
- efectiva das *Tarefas* usando, tal como no caso anterior, a função *evaluate* para executar no decorrer da simulação expressões escritas na linguagem SimTalk, substituindo desta forma o método *MessageControl*;
- da primitiva de criação de processos durante a execução (*Create*); esta primitiva não deverá ser implementada a curto prazo, embora seja já possível a criação de objectos físicos durante a execução da simulação (aguarda-se a versão 3.1 do Simple++ onde também será possível eliminar dinamicamente objectos físicos do modelo);
- de *Procedimentos,* que permitam substituir partes do corpo de um processo pela chamada a um procedimento; esta facilidade poderá ser implementada assim que, numa futura versão do Simple++, seja possível eliminar um objecto físico do modelo.;
- de *Serviços* no interior de um processo, contribuindo para uma melhor estruturação do modelo num conjunto de máquinas de estado mais simples; dentro de um mesmo processo, os serviços não podem ser executados em paralelo pois eles partilham a mesma fila de entrada de sinais, existindo para cada sinal de chegada apenas uma transição de estado possível para o conjunto de serviços presente assim como apenas uma condição de arranque; esta facilidade deverá ser implementada para a primeira versão da ferramenta de modelação em SDL;
- de *Macros*, implementando partes de máquinas de estados que, ao serem declaradas, são únicas em toda a especificação e são visíveis por todos os processos que as desejem utilizar; esta facilidade poderá ser implementada assim que, numa futura versão do Simple++, seja possível eliminar um objecto físico do modelo.

Estrutura, Estática e Dinâmica

A estrutura do modelo SDL é garantida pela própria estrutura hierárquica do Simple++. Não existe pois qualquer problema do ponto de vista da estrutura estática, à excepção da representação da máquina de estados, onde não é possível ter dois objectos com o mesmo nome como já foi referido. Este problema seria resolvido com a possibilidade de associar um nome a um objecto físico, além daquele que o identifica no universo do modelo, e permitindo que esse nome fosse visualizado na interface com o utilizador. No que se refere à estrutura dinâmica do modelo, tal como a facilidade de criar e de eliminar um objecto físico do modelo, apenas será possível a sua modificação na versão 3.1 do Simple++.

Um aspecto interessante na estrutura estática do modelo traduz-se no existência de *canais* entre os blocos de uma especificação e de *caminhos para sinais* entre *processos*. Na implementação não foi de facto viável materializar estas vias de comunicação através de "ligações" entre os objectos físicos, tendo-se por isso optado pela utilização de endereços absolutos para o encaminhamento dos sinais. Os motivos para tal são vários, sendo o mais forte o facto de não existir no Simple++ uma forma de referenciar, a não ser manualmente, o ponto de ligação de um canal de um dado nível ao canal do nível superior ao qual ele se encontra ligado (primitiva *connect* no SDL-PR). A ligação manual não era de facto viável, motivo pelo qual esta opção foi abandonada.

Dados

Poderá afirmar-se que o problema da representação dos dados foi em todo este trabalho remetido para um segundo plano. De facto, a utilização de tipos de dados abstractos e a possibilidade de o utilizador definir os seus próprios tipos de dados torna muito flexível a aproximação ao problema. Assim sendo, o utilizador do SDL no Simple++ deverá caracterizar os seus tipos de dados bem como o conjunto de operações que sobre eles se podem realizar (os denominados métodos, numa perspectiva orientada ao objecto) através da criação de classes de objectos de dados. Deste modo, a execução de uma tarefa aquando de uma determinada transição de estado deverá sempre traduzir-se na execução de uma destas operações (métodos) sobre o conjunto de atributos do objecto em questão.

Do ponto de vista da simulação, i.e. interno ao Simple++, bem como da exportação da especificação SDL com a produção do ficheiro no formato SDL-PR, a solução adoptada garante uma grande flexibilidade na conversão dos tipos de dados. Por outro lado, do ponto de vista da geração de código em linguagem C / C++ por exemplo, o modelo não contém toda a informação para a descrição da implementação da aplicação. Podem deste modo ser inseridos comentários na especificação SDL que forneçam a informação complementar de suporte à geração do código indicando por exemplo: a implementação dos tipos de dados abstractos, a inclusão de código C++ associado a tarefas SDL, a inclusão de declarações em C++, a inclusão de ficheiros C++ nas especificações SDL ou a correspondência entre o código SDL e código C++ a incluir.

A aproximação usada não é assim restritiva, deixando totalmente em aberto o caminho para futuros desenvolvimentos, nomeadamente no que se refere à geração de código executável.

6.5.8 A Geração de SDL-PR

O requisito da geração de SDL-PR advém da necessidade básica de permitir que as especificações realizadas através da ferramenta desenvolvida possam ser exportadas para

outras ferramentas SDL, com o objectivo de reutilizar as funcionalidades por elas fornecidas tais como por exemplo a geração de código executável.

A geração do SDL-PR baseia-se neste protótipo apenas na estrutura hierárquica da especificação e na estrutura da máquina de estados dos processos. Sendo usado o endereçamento absoluto para *sinais* enviados do interior de qualquer processo para um qualquer outro processo da especificação, foi estabelecido um algoritmo que, usando a estrutura hierárquica da especificação e a estrutura das máquinas de estado dos processos, gera de uma forma automática para cada bloco do sistema:

- os *canais* ligando os blocos do nível em questão, ou os *caminhos para sinais* dos processos do nível em questão, sendo a cada uma destas vias atribuído um nome único no âmbito desse nível;
- as ligação entre estas vias de comunicação de sinais e os sinais que passam em cada um dos sentidos da ligação;
- a ligação entre a via de comunicação do nível em questão e a via do nível superior à qual ela está ligada.

O algoritmo desenvolvido percorre assim a hierarquia da especificação SDL, identificando *canais* e *caminhos para sinais* bem como os *sinais* que neles transitam. Toda esta informação é, para cada *bloco* e incluindo para o nível *sistema*, armazenada nas tabelas *Signal* existentes em todas as instâncias da classe *Sistema* e da classe *Bloco* (figuras 6.11 da página 172 e 6.12 da página 173). Uma vez completa a construção das tabelas *Signal*, a geração de SDL-PR para o nível *sistema* e para os diversos *blocos* é quase imediata. A geração faz-se navegando nas referidas tabelas e percorrendo numa primeira fase a hierarquia da especificação sem tratar os processos, sendo estes tratados um a um no final. A geração do SDL-PR de cada *processo* é realizada invocando um conjunto de métodos existentes no interior do *SDL_Control* desse *processo*. Estes métodos realizam a visita recursiva da árvore da máquina de estados gerando a sua descrição em SDL-PR.

A possível geração de código a partir do SDL-PR está como vimos dependente da representação dos dados abstractos, tendo sido remetida para trabalho futuro.

6.6 Ambiente de Desenvolvimento de Aplicações de Gestão Fabril

6.6.1 Vista Sobre a Arquitectura do Sistema de Informação

Conforme apresentado na figura 6.8 (página 167), a Bancada de Modelação providencia uma vista sobre o Sistema de Informação. A Figura 6.18 apresenta essa mesma vista ilustrando o fluxo de informação através das várias camadas do sistema integrado de fabrico. Este modelo é de facto ortogonal ao modelo do fluxo de materiais do sistema integrado de fabrico, fornecendo a abstracção necessária à implementação de funções de simulação, monitorização e de gestão dos diversos componentes de informação / controlo do sistema integrado de fabrico (tipicamente aplicações de software), assim como do inerente fluxo de informação que esses componentes produzem e consomem. Estes aspectos são da maior relevância ao projectar, integrar e testar ou executar tanto os componentes individualmente como a arquitectura do sistema integrado de fabrico.

O modelo de abstracção fornecido pela linguagem de modelação apoia o ciclo de vida de cada componente suportando as diferentes fases e actividades, tais como o projecto do componente individual, a medida de desempenho, o teste e integração de componentes, e a

visualização, até à monitorização, a simulação e o controlo. Estes aspectos serão tratados nos parágrafos que se seguem.

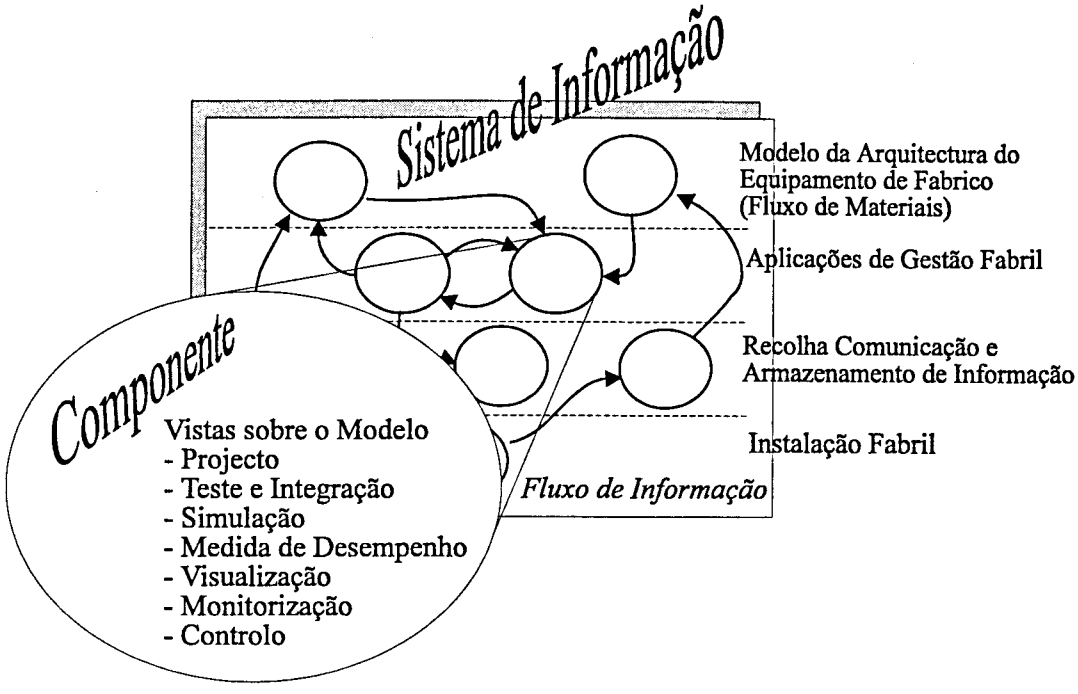


Figura 6.18 - Fluxo de Informação e Vista de um Componente através do Modelo do Sistema de Informação

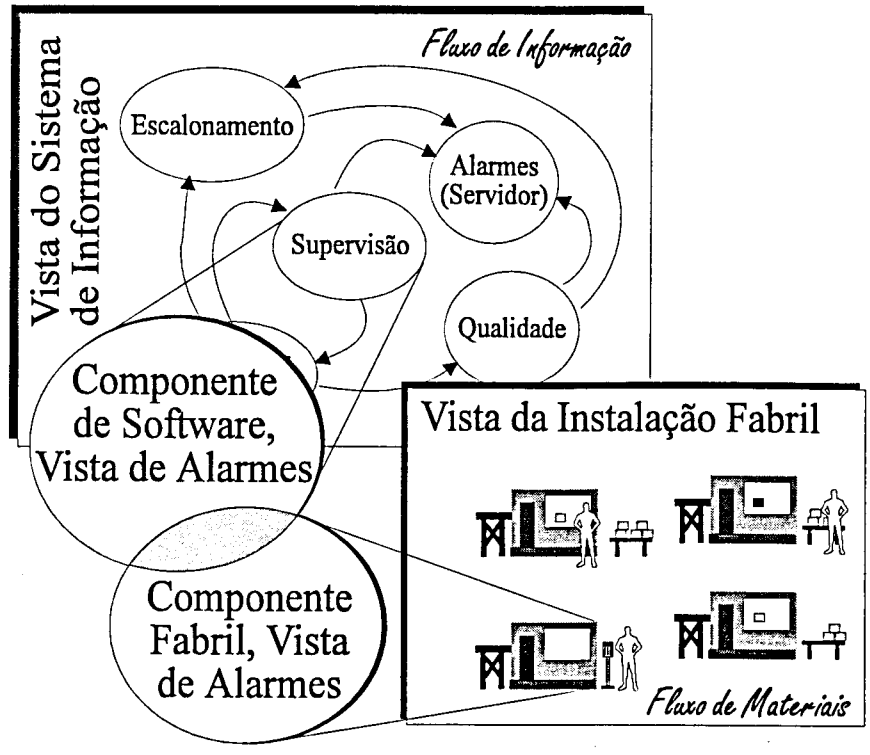


Figura 6.19 - O Modelo Integrado como Ponto de Acesso às Aplicações de Gestão Fabril

Os modelos de fluxo de materiais e de informação podem ser utilizados como pontos de acesso das aplicações de gestão fabril, sendo esta faceta do modelo integrado ilustrada na figura 6.19, onde são apresentadas vistas sobre o próprio sistema de informação, neste caso fornecidas pela aplicação de alarmes. Esta aplicação pode ter uma vista sobre a aplicação de Supervisão assim como uma vista sobre um determinado recurso de fabrico, sendo a sobreposição das duas vistas evidente sempre que existam alarmes de supervisão relativos ao recurso de fabrico em questão.

6.6.2 Suporte Integrado ao Projecto, Desenvolvimento e à realização de Protótipos

O Ambiente de Projecto

O ambiente fabril não é um ambiente onde seja simples o trabalho do integrador de sistemas, mas apresenta fortes motivos para a sua existência. Cada processo de fabrico e mesmo cada instalação fabril tem os seus requisitos próprios, e uma ferramenta que permita a máxima facilidade de configuração / adaptação às necessidades do utilizador vem decerto ao encontro dos desejos de qualquer integrador de sistemas. A Bancada de Modelação (BM) desenvolvida pretende responder aos desafios colocados pela especificidade dos problemas procurando identificá-los cedo no ciclo de desenvolvimento.

A BM pode ser utilizada para ajudar na especificação, projecto e desenvolvimento de protótipos de aplicações ou ferramentas de software, os componentes do sistema de informação fabril [137,138,139]. A construção do modelo de um componente corresponde de facto à criação um protótipo executável que pode ser usado para emular a aplicação final e que, na sequência de posteriores ciclos de evolução e refinamento, tenderá para a própria aplicação.

A BM constitui além disso uma plataforma de trabalho assistindo ao desenvolvimento integrado de aplicações cooperantes de gestão fabril. De facto, estas aplicações são inerentemente distribuídas e requerem grande flexibilidade devido ao facto de serem construídas de uma forma modular e independente, apresentando interfaces bem definidas e apresentando estruturas mais evidentes do que as aplicações centralizadas equivalentes.

O ambiente fornecido pela BM foi construído sobre dois métodos de modelação, a linguagem de modelação do Simple++ e o SDL, apresentando as vantagens e desvantagens inerentes já referidas no capítulo anterior.

O Processo de Desenvolvimento de Aplicações de Gestão Fabril Distribuídas

O desenvolvimento de sistemas distribuídos é primeiramente caracterizado pela identificação da estrutura do sistema, elemento fundamental ao seu projecto, construção e evolução, estrutura essa que deve ser descrita explicitamente e preservada durante o processo de desenvolvimento de software. A estrutura estável poderá além disso ser utilizada na descrição do projecto do sistema assim como na sua construção.

Existem fundamentalmente duas metodologias para a identificação da estrutura e dos componentes de um sistema distribuído: a decomposição funcional e a abordagem orientada ao objecto. Estas metodologias, que suportam abordagens alternativas à análise e projecto de sistemas informação, são aqui utilizadas de forma complementar tal como foi referido no capítulo 5. A análise funcional e a decomposição hierárquica são primeiro utilizadas na estruturação do sistema distribuído e na identificação dos seus componentes. A abordagem orientada ao objecto, inerente à utilização da Bancada de Modelação, é em seguida utilizada na especificação, projecto e implementação dos componentes. É importante que a

caracterização dos componentes se realize de uma forma independente do contexto, i.e. sem qualquer referência externa a outros componentes que não seja o potencial destino ou origem das mensagens, favorecendo assim a sua reutilização. Com o objectivo de conseguir independência entre os diversos componentes no sistema, a informação de estado deve ser local a cada componente e não distribuída pelo sistema. Além disso, deve ser reconhecida a importância da reutilização de classes de componentes. O resultado desta abordagem será um projecto onde coexistem componentes correspondentes a entidades existentes no mundo real comunicando entre si, sendo identificadas as suas classes bem como o tipo de dados trocados entre eles. A especificação das interfaces deve ser realizada partindo do princípio que os componentes presentes no sistema correm em paralelo, sendo os fluxos de dados implementados através da troca de mensagens. Abaixo deste nível, o comportamento de cada componente será estabelecido pelas suas máquinas de estados, as quais podem ser vistas como um terceiro paradigma de representação.

A abordagem descrita nos parágrafos anteriores poderá ser aplicada no desenvolvimento de aplicações no ambiente da BM. De facto, a utilização de classes SDL permite que sejam construídos blocos funcionais básicos fora de qualquer contexto, com informação de estado local e com uma interface de mensagens bem definida. Esta estratégia é eficaz no desenvolvimento de aplicações não interactivas em que os diversos processos presentes correm em paralelo comunicando entre si.

Quando é desejado o desenvolvimento de aplicações interactivas deverão ser incluídos no modelo do sistema os processos cliente, responsáveis pela interface com o utilizador. Se necessário, deverá também ser possível modelar o comportamento de um utilizador face à referida interface. A experiência mostra todavia que é desejável que sejam desenvolvidos protótipos da interface com o utilizador, seja em MS-Windows ou mesmo em X-Windows, por forma a que ela possa ser validada, testada e aceite pelo utilizador final.

A estratégia de desenvolvimento utilizada está evidente na figura 6.20 onde se caracteriza o âmbito da utilização da BM e se combinam as metodologias [140]. A figura 6.20 ilustra o processo de desenvolvimento de aplicações partindo da análise estruturada do sistema a desenvolver. Para esta etapa, está a ser usada uma ferramenta CASE, o Teamwork [141], assistindo à produção de uma primeira especificação funcional simples baseada em diagramas de fluxo de informação, mini-especificações (P-spec) e dicionário de dados respectivo (1). Esta fase de análise estruturada deve permitir a identificação de classes de serviços que poderão ser reutilizados em múltiplos pontos do modelo. Realizada esta identificação, são produzidas classes SDL (processos e/ou blocos) satisfazendo os requisitos dos serviços especificados, por exemplo classes fornecendo o serviço de poço, fonte ou armazém de informação, de acesso a serviços de mensagens, de cálculo específico, e muitos outros. Estas classes deverão ser construídas com suficiente generalidade (2), i.e. sem atender ao contexto em que vão ser inseridas. Será assim possível a construção de uma biblioteca de classes que podem vir a ser reutilizadas em futuros projectos. Uma vez disponíveis estas classes SDL, a especificação funcional poderá ser convertida automaticamente numa descrição em SDL do sistema (3). Esta conversão passa pela utilização da estrutura hierárquica do DFD para a produção da estrutura em árvore da descrição SDL, e pela utilização da informação existente nas especificações dos processos no DFD (P-spec) para seleccionar, instanciar e configurar as classes de processos SDL anteriormente caracterizadas (2 e 4) [140]. Uma vez construída, como ilustra o exemplo, a classe **Classe_PLC**, esta poderá então ser instanciada num modelo particular definido pelo utilizador (5), e ser usada na simulação integrada do fluxo de materiais e de informação. A

evolução natural deste processo é naturalmente a geração do código para o software especificado usando o modelo SDL, usando ferramentas SDL existentes no mercado.

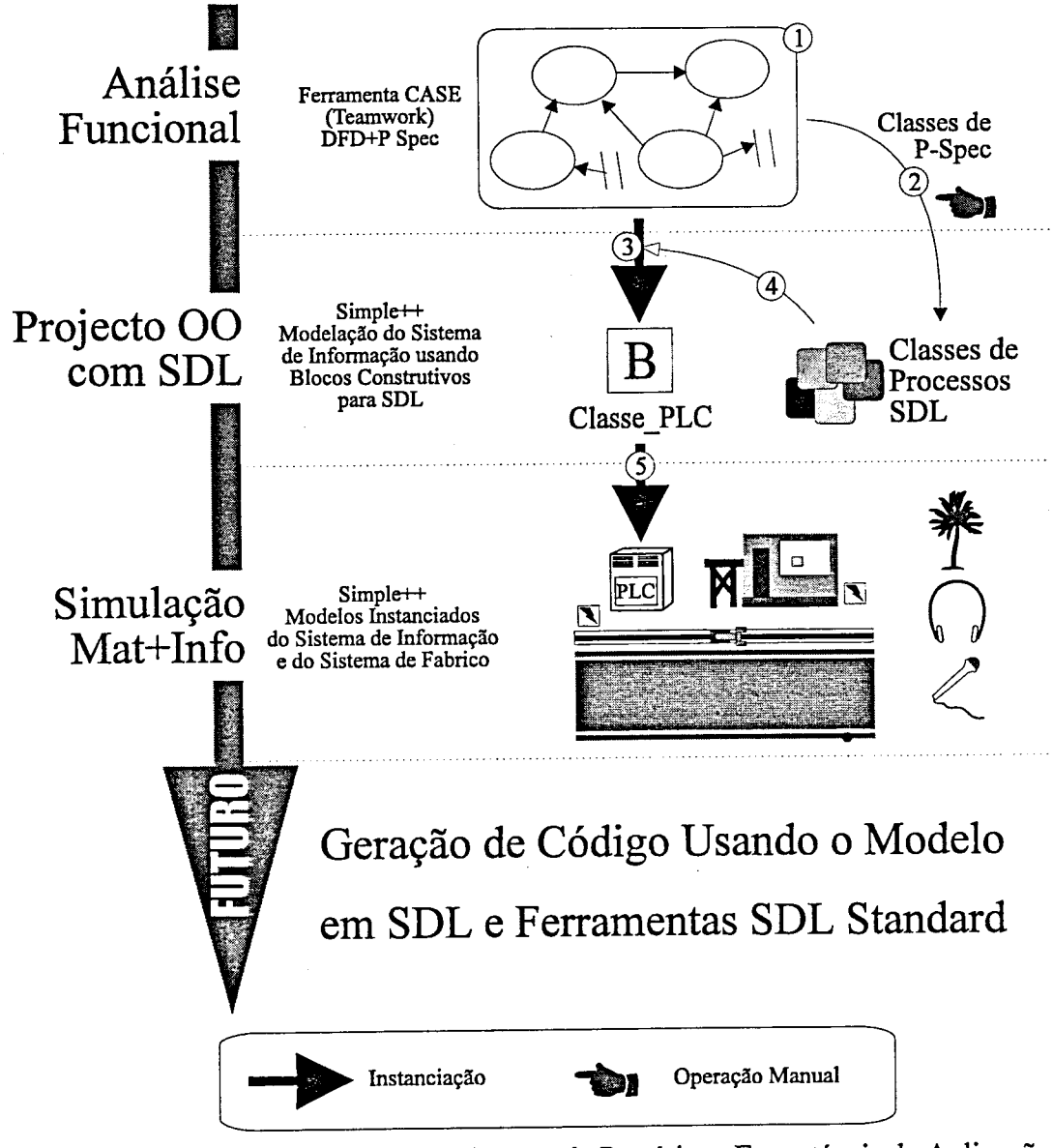


Figura 6.20 - Processo de Desenvolvimento de Protótipos Executáveis de Aplicações Industriais

6.6.3 A Utilização da Bancada de Modelação

O Ambiente de Integração, Teste e de Treino do Utilizador

A Bancada de Modelação foi construída sobre uma ferramenta normalmente utilizada para simulação de fluxo de materiais em problemas complexos de logística, planeamento de nível táctico ou sequenciamento de ordens de fabrico, o Simple++, que corre numa plataforma UNIX e ambiente OSF/Motif [60]. O Simple++ é usado no PROFIT para modelação e monitorização do fluxo de materiais, dos recursos de fabrico e do em-curso-de-fabrico ("WIP-tracking"), assim como no suporte de ferramentas de escalonamento apoiado em simulação [142]. A invulgar capacidade de modelação baseada em conceitos

poderosos, aliada às facilidades de animação gráfica, constitui a grande vantagem desta ferramenta. Além disso, a utilização do Simple++ combinada com ambiente de integração do PROFIT permite que o projecto e desenvolvimento de protótipos utilize com vantagem a infra-estrutura existente, que inclui um sistema de transferência de mensagens (STM) [2] e ligação a um motor de base de dados. O resultado é um ambiente poderoso de análise, projecto e desenvolvimento de componentes / aplicações de planeamento e controlo, assim como de teste de protótipos suportado por simulação ou utilizando componentes / aplicações exteriores já existentes.

Na sua tarefa de análise e de escolha / projecto de soluções o consultor ou integrador de sistemas é confrontado com dois ambientes distintos: o ambiente de modelação, onde se situa o modelo orientado ao problema a resolver e onde é representado cada componente do sistema real, e o próprio sistema real, i.e. a instalação fabril. Estes dois ambientes podem interagir de modo a suportar o projecto, o teste, a instalação, e a contínua evolução dos sistemas e soluções instalados. Na figura 6.21 apresentam-se os dois ambientes interactuando através dos *sensores* e *actuadores* já referidos atrás. O STM que constitui o elo de ligação entre aqueles ambientes, permite que se encaminhem mensagens (sinais) vindos dos sensores a serem utilizadas pelos modelos em execução no ambiente de modelação, tanto com o objectivo do controlo como da monitorização. Sempre que alguma acção de controlo do modelo sobre o sistema real seja necessária, o STM transfere os respectivos comandos, sob a forma de mensagens (sinais) que, uma vez entregues aos actuadores, serão transformados numa acção pré-determinada sobre o modelo físico.

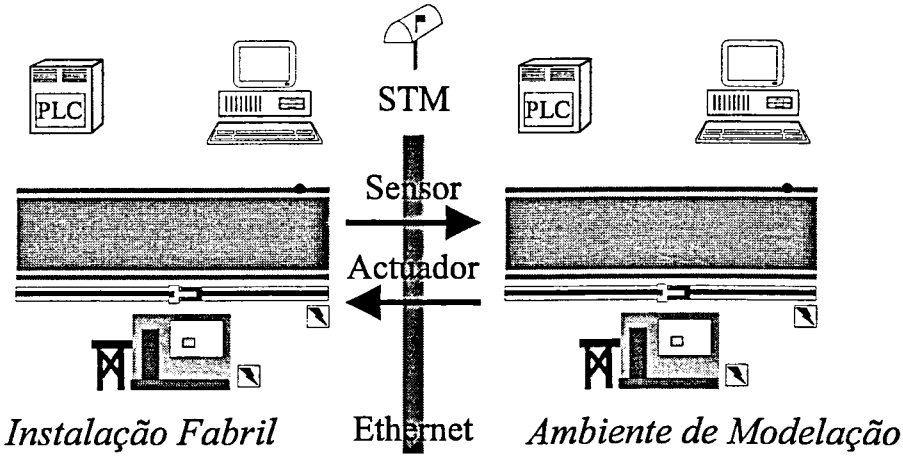


Figura 6.21 - A Interacção entre Instalação Fabril e o Ambiente de Modelação

As facilidades suportadas pelos modelos de fluxo de informação e de fluxo de materiais na BM tem duas vertentes complementares, a do apoio à decisão "off-line", desenvolvimento do modelo do sistema de informação distribuído, planeamento ou optimização de "layouts" através de simulação; e a do apoio à decisão em "tempo-real" ("on-line") através de decisões rápidas disparadas por eventos na planta fabril com objectivos de controlo.

A figura 6.22 resume os ambientes de modelação, realização de protótipos, simulação e execução disponibilizados pela BM. A existência de primitivas básicas para a construção de modelos em SDL, em conjunto com as primitivas originais do Simple++, permite a construção de modelos integrados onde é possível descrever a interacção entre os fluxos de informação e os fluxos de materiais presentes na instalação fabril. Como a figura ilustra, o ambiente de modelação e simulação "off-line" permite que o modelo do sistema possa ser

testado, inclusive através da sua interacção com protótipos da interface com o utilizador das aplicações. No exemplo é apresentado o protótipo de um terminal fabril que inclui a interacção com o modelo de simulação.

A interacção de um ou mais operadores com o modelo simulado do sistema poderá ter grande interesse sempre que se pretenda a avaliação e validação do protótipo desenvolvido. Por outro lado, pode-se ainda constituir um ambiente de treino onde o operador em formação interage com o sistema simulado, podendo ser recriadas simulações de ocorrências reais mais e menos prováveis.

A emissão ("release") do modelo corresponde à sua instalação e colocação em serviço na instalação fabril. Este processo poderá ser iterativo permitindo que, através de refinamentos sucessivos dos modelos, se chegue àquele que melhor reflecte as necessidades da instalação. A execução em tempo-real do modelo será responsável pela gestão da execução das operações de fabrico através das operações de despacho e monitorização, num modo em que o modelo executável integrado verticalmente com a instalação fabril é de facto um "modelo de controlo".

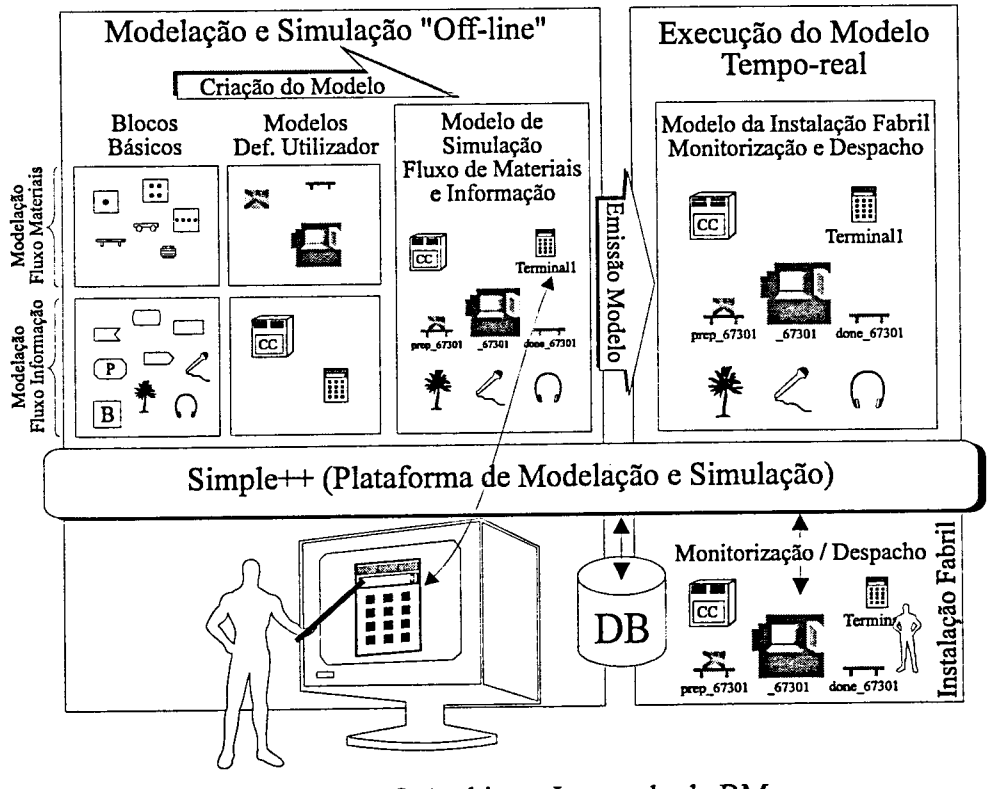


Figura 6.22 - O Ambiente Integrado da BM

Teste e Integração de Componentes

O ambiente de modelação facilita o teste e integração de componentes através da simulação, utilizando ferramentas que permitem que o modelo de um novo componente seja introduzido no sistema e testado num ambiente simulado. O rastreio ("trace") de mensagens no sistema em execução ajuda à validação do novo componente. Uma vez validado o modelo do novo componente usando a simulação, esse mesmo modelo poderá ser utilizado na geração de código para a construção do executável da aplicação, à qual se seguirá então o teste do novo componente executável.

A fase de teste do componente / módulo executável deverá permitir à equipa de desenvolvimento uma implementação mais simples do sistema através do teste módulo a módulo. O teste de um módulo é conseguido através do ensaio da interacção entre o módulo executável em teste e todos os outros módulos com os quais ele interage, que sendo simulados, são executados no ambiente da BM.

As facilidades de rastreio estão disponíveis tanto no modelo "off-line" como no modelo de "tempo-real", fornecendo ao utilizador os meios para facilmente seguir o comportamento da especificação SDL do componente (2 na figura 6.23), ou a interacção entre as especificações SDL de diversos componentes através do rastreio das mensagens trocadas através do *Ambiente* (1 na figura 6.23). Repare-se que, no exemplo da figura 6.23, o seguimento da execução do modelo do controlador de célula corresponde de facto ao seguimento do seu executável, uma vez que o modelo está a funcionar no seu modo de monitorização. O suporte durante a execução é realizado através de facilidades de monitorização, visualização, e gestão de mecanismos de medida de desempenho descritas adiante [134].

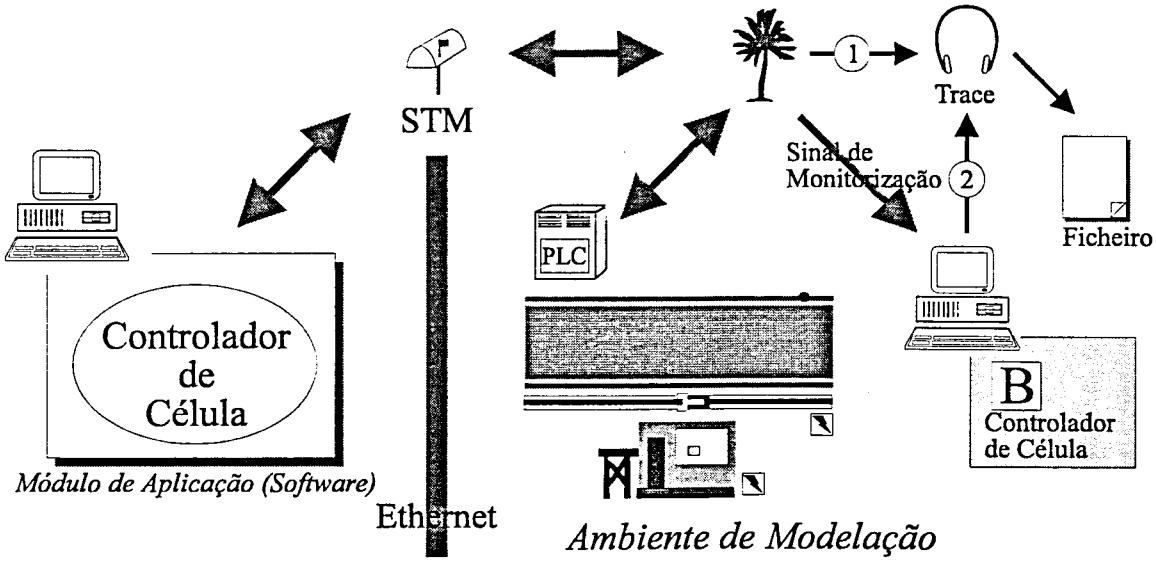


Figura 6.23 - Teste & Integração em Laboratório

As facilidades de teste e integração em laboratório apresentadas tornam mais simples o desenvolvimento de um módulo de aplicação (por exemplo, o Controlador de Célula na figura 6.23, à esquerda), permitindo que durante o seu período de desenvolvimento ele interactue com o ambiente de modelação da BM (figura 6.23, à direita).

A situação da figura 6.23 pode ser aplicada em situações mais complexas de teste de componentes / aplicações. A figura 6.24 pretende ilustrar o teste da aplicação de Alarmes, uma aplicação servidora que recebe eventos (ocorrências de alarmes) de todas as outras aplicações e é responsável pelo seu tratamento, incluindo por exemplo a gestão das confirmações e o registo de ocorrências. No teste desta aplicação deseja-se aferir a interacção / cooperação entre a aplicação de Qualidade e a de Alarmes numa situação em que as aplicações servidoras de Manutenção e de Qualidade não estão disponíveis e são por isso simuladas no ambiente da BM. No exemplo temos que na sequência da recolha de dados de qualidade (Sinal Q1) surge um problema detectado pela aplicação de Qualidade que provoca um pedido de manutenção de um recurso de fabrico, sendo então enviado o sinal M1 à aplicação de Manutenção; este problema dá origem a uma ordem de manutenção a ser executada e a manutenção faz uma proposta para o período da intervenção de

manutenção (sinal S1); o Simulador / Escalonador integra esta exigência e consolida um novo plano tentando garantir a sua execução dentro dos prazos especificados, mas acaba por responder tratar-se de uma tarefa impossível (sinal I1); é então dada a indicação à Manutenção que se trata de uma situação impossível, enviando esta um sinal à aplicação de Alarmes (sinal A1) indicando a ocorrência do problema.

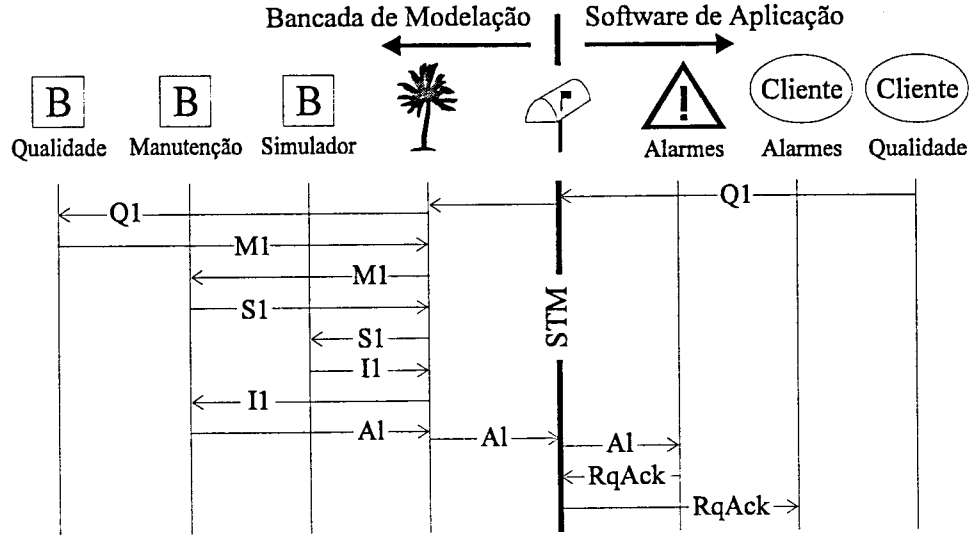


Figura 6.24 - Teste e Integração de Diferentes Componentes

Após o teste individual de cada componente, é iniciada a construção do sistema através da sua integração faseada. Esta integração deverá permitir uma mais suave implementação do sistema através da integração gradual dos seus módulos componentes. A integração passo-a-passo será executada através da substituição gradual de cada um dos módulos simulados no modelo pela aplicação executável propriamente dita. O suporte durante a execução é providenciado através de um conjunto de ferramentas permitindo a visualização e a monitorização da própria infra-estrutura de integração.

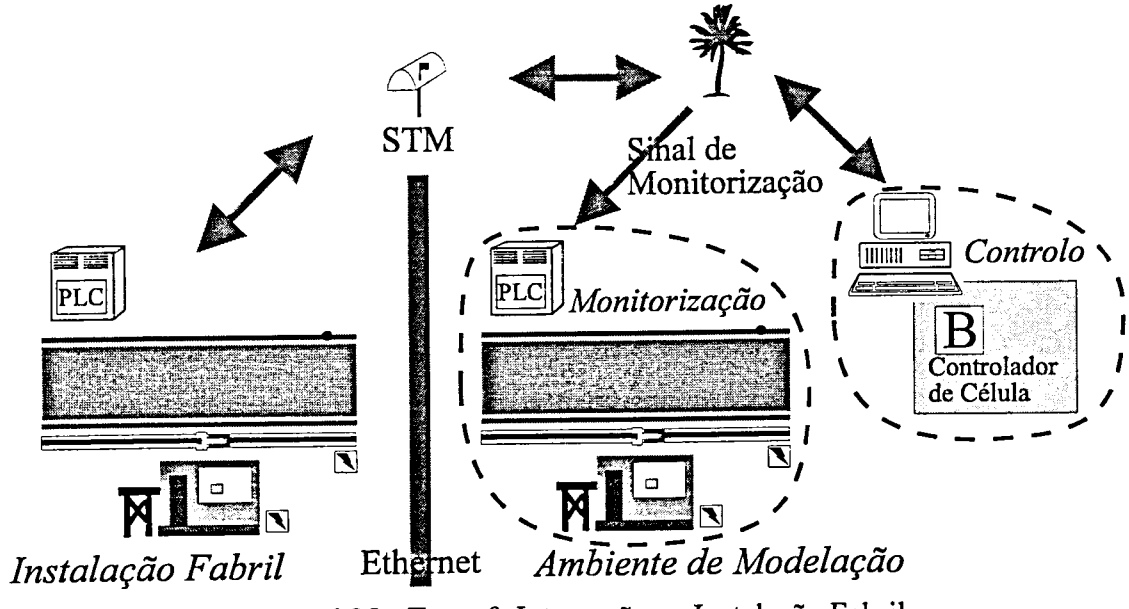


Figura 6.25 - Teste & Integração na Instalação Fabril

A figura 6.25 ilustra uma situação possível, com o modelo SDL do controlador de célula (à direita) a controlar a própria instalação fabril (à esquerda), enquanto que os outros objectos no modelo na BM são apenas utilizados para monitorização. Estas facilidades são de facto da maior relevância ao projectar, construir ou actualizar um sistema integrado de fabrico.

O ambiente de teste permite pois verificar se um novo módulo de aplicação cumpre os requisitos estabelecidos do ponto de vista: da interface com o utilizador, da comunicação com as restantes aplicações, e das suas funcionalidades. Por outro lado, torna-se possível antecipar e ajudar a resolver os problemas que normalmente surgem apenas no momento tardio e crítico da integração de sistemas. De facto, poderá dizer-se que esta integração transita da fase final de implementação para a fase de validação do modelo / especificação. Finalmente, é possível levar a cabo o teste do sistema em situações de menor probabilidade de ocorrência mas cuja análise é relevante para a garantia da fiabilidade do sistema.

Fluxo de Informação - Monitorização e Controlo

As funções de monitorização e de controlo estão de facto relacionadas, não podendo existir controlo sem monitorização do processo. A figura 5.10 do capítulo anterior apresenta o modelo de controlo da produção através da utilização das primitivas de "Transporte" e de "Execução", o qual pode aqui ser estendido ao sistema de informação. De facto, é igualmente possível reduzir o tratamento da informação no ambiente distribuído às duas acções básicas de "Transferência" e de "Processamento" da informação.

A monitorização e o controlo do fluxo de informação terão duas funções distintas, sendo a primeira o suporte ao desenvolvimento e integração de sistemas fabris distribuídos, e a segunda o apoio ao seguimento da informação durante a normal execução do sistema de informação fabril. No primeiro caso, temos a validação da descrição formal face aos requisitos informais, assim como a posterior validação da versão em código executável relativamente à especificação formal. Esta facilidade é de grande utilidade, tanto para o integrador de sistemas como para a equipa de desenvolvimento, sendo suportada pelo conjunto de ferramentas já apresentado. No segundo caso, pretende-se tratar o fluxo de informação fabril de uma forma semelhante ao fluxo de materiais. De facto, a existência de relações causa-efeito entre o fluxo de informação e o fluxo de materiais associado, e vice-versa, leva à necessidade de saber num determinado instante no tempo e para um determinado contexto: qual a informação (por exemplo, relativa à qualidade, manutenção, ordens de fabrico), o local onde ela se encontra armazenada (sobre suporte magnético ou papel), e qual a operação a que está a ser submetida (tipo de processamento). A extensão deste conceito à arquitectura organizacional pode levar à caracterização e avaliação do modelo organizacional que será abordado mais adiante neste capítulo.

Neste contexto, a monitorização fornece informação insubstituível sobre o estado global do sistema, nomeadamente: a falta de uma determinada informação (num dado momento, o atraso de um formulário deve despoletar um mecanismo de aviso à pessoa responsável e / ou seu superior hierárquico); o grau de utilização de um terminal fabril (no PROFIT / SHOP-CONTROL, os operadores da instalação fabril podem introduzir informação em qualquer terminal permitindo que a disponibilidade e carga dos terminais sejam monitorizadas); a disponibilidade dos serviços fornecidos; a carga da máquina ou da rede; etc.

Fluxo de Informação - Visualização

A visualização consiste na forma de associar aos modelos das aplicações (as especificações na linguagem formal) a sua representação gráfica [143]. A visualização tem um papel de extrema importância na comunicação humana em geral, como resultado da elevada largura de banda do sistema de visão assim como da grande velocidade com que somos capazes de reconhecer padrões visuais. A representação gráfica do fluxo de informação tem deste modo a maior importância devendo ser objecto de especial cuidado na sua apresentação ao utilizador.

Na BM, a visualização é neste momento realizada ao nível da especificação SDL através da animação das máquinas de estados dos seus processos. A visualização da interacção entre as diversas especificações presentes na BM é neste momento realizada através da simples impressão no terminal das diversas transacções na linguagem textual do MSC [136]. Esta situação é aceitável para um protótipo mas não o é de facto para a primeira versão da BM. Está assim em fase de estudo, o desenvolvimento uma classe no Simple++ capaz de emular o MSC de uma forma gráfica integrada no próprio ambiente de simulação.

Fluxo de Informação - Avaliação de Desempenho

A avaliação de desempenho está associada à definição de um modelo matemático relacionando um conjunto de variáveis do sistema que o gestor julgue relevante. A utilização da aplicação de Supervisão, quer na versão SHOP-CONTROL quer na sua nova versão de Quadro de Bordo Dinâmico do REAL-I-CIM, poderá assim ser usada neste contexto. A vantagem desta abordagem resulta na possibilidade de adaptar o modelo de apoio à decisão a cada caso particular. A título ilustrativo temos como exemplo de indicadores de desempenho do sistema de informação fabril: a medida do tempo de resposta do sistema ao utilizador, o "lead time" para transmissão ponto a ponto de informação, ou o número de transacções por unidade de tempo. Além disso, é importante dar indicação da relevância da eficiência de interacção entre o fluxo de informação e de materiais, da máxima relevância em muitas situações por afectar directamente o desempenho do sistema de fabrico. É assim crucial detectar e avaliar situações em que quer o fluxo de materiais quer o fluxo de informação são afectados, por exemplo um lote bloqueado aguardando um visto do laboratório de qualidade, ou o processamento de uma encomenda pendente por causa da avaria de uma máquina.

6.6.4 Exemplos de Utilização da Bancada de Modelação

Na sequência do exposto nos pontos anteriores são agora apresentados dois exemplos concretos de utilização da BM para trabalhos de modelação e simulação. O primeiro é extraído de um trabalho de modelação e simulação de uma Célula de Fabrico Integrado e o segundo aborda a modelação Técnico-Organizacional de Sistemas de Fabrico. Este último exemplo está a ser realizado no âmbito de uma outra dissertação de doutoramento, motivo pelo qual a sua descrição será breve, visando apenas mostrar como as primitivas de modelação SDL podem ser utilizadas num outro contexto.

Célula Integrada de Fabrico do INESC, Modelação e Simulação

A criação de um modelo na Bancada de Modelação usando o ambiente integrado descrito implica naturalmente a utilização de blocos construtivos básicos de ambas as linguagens disponíveis. Na figura 6.26 podemos observar alguns blocos construtivos para a modelação

de fluxo de materiais do Simple++ tais como: o tapete de transporte, o robot, o "slider", e o torno CNC. Por outro lado, surgem primitivas de modelação de componentes / aplicações de software baseados na especificação em SDL, tais como o PLC, o PC, o controlador do robot / "slider" e o controlador do torno CNC. Além destes objectos instanciados no decorrer da construção do modelo temos: o *Ambiente*, responsável pelo encaminhamento de mensagens entre as diversas especificações SDL no modelo; o *Sensor*, encarregado da conversão do sinal físico num sinal de informação capaz de ser interpretado por uma especificação SDL; e o *Actuador* que aparece embebido em métodos. Existe ainda uma facilidade de rastreio ("trace") onde convergem todos os sinais trocados via *Ambiente*, além de outros sinais no interior de cada especificação cuja monitorização pode ser configurada pelo utilizador.

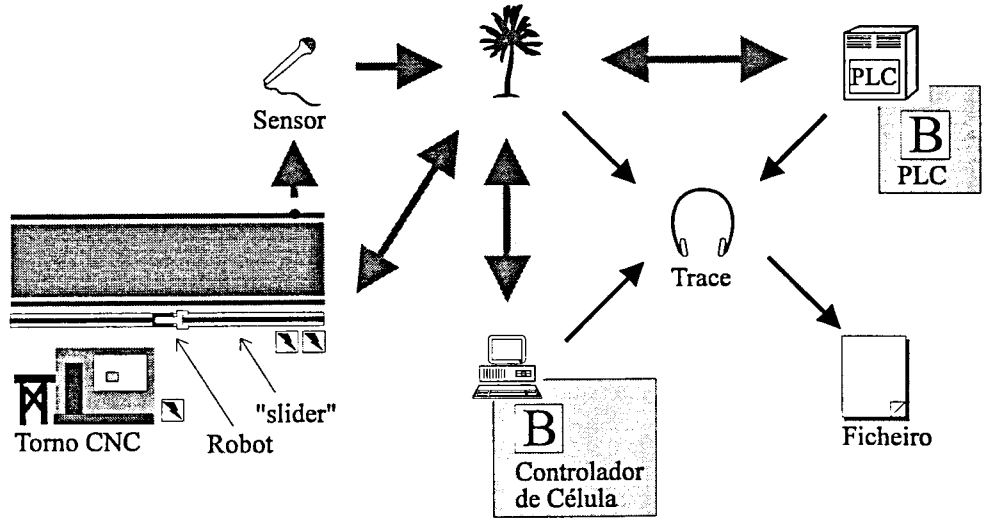


Figura 6.26 - Ambiente de Simulação da Célula de Fabrico Integrado

Um resultado imediato deste ambiente é a integração dos modelos do Equipamento de Fabrico (e do fluxo de materiais) e do Sistema de Informação (e do fluxo de informação / controlo). É a seguir examinada a forma como se atinge a integração entre o fluxo de materiais e o fluxo de informação, assim como o modo como eles interactuam ao nível da instalação fabril. Para tal será utilizado um exemplo extraído do modelo da Célula de Fabrico Integrado do INESC [144].

A figura 6.27 ilustra tanto o modelo do fluxo de materiais como do fluxo de informação para o caso de um exemplo simples. O modelo físico inclui o torno de comando numérico, o "slider", o robot, o tabuleiro, o sensor, e o tapete de transporte (a seta indica o sentido do deslocamento). O modelo de informação compreende os controladores do torno, do "slider" e do robot e o PLC.

- *Fluxo de Materiais* Quando o tabuleiro contendo uma peça, ao ser transportado pelo tapete, activa o sensor *electromecânico* de posição L é levantado do tapete ficando imóvel. O robot é de seguida deslocado pelo "slider" para a posição mais próxima do tabuleiro para imediatamente pegar na peça.
- *Fluxo de Informação / Controlo* Chega um sinal ao PLC indicando que o tabuleiro passou sobre o sensor de posição L (*Lathe*). O PLC envia um sinal ao *actuador* no *Ambiente*, resultando na execução de um método que actua sobre o objecto físico traduzindo a acção de um actuador pneumático, o qual levanta então o tabuleiro do tapete transportador (*LatheUp*). Neste momento, o PLC envia ao controlador do "slider" uma mensagem para que ele se posicione no ponto mais

próximo de tabuleiro parado (*p1c*), enviando uma confirmação (*okp1c*) depois de se imobilizar na posição desejada. O PLC recebe a mensagem *okp1c*, e de imediato envia uma outra ao controlador do robot (*P1*) para que ele pegue na peça do tabuleiro, enviando uma confirmação (*okP1*) assim que tenha a peça em seu poder.

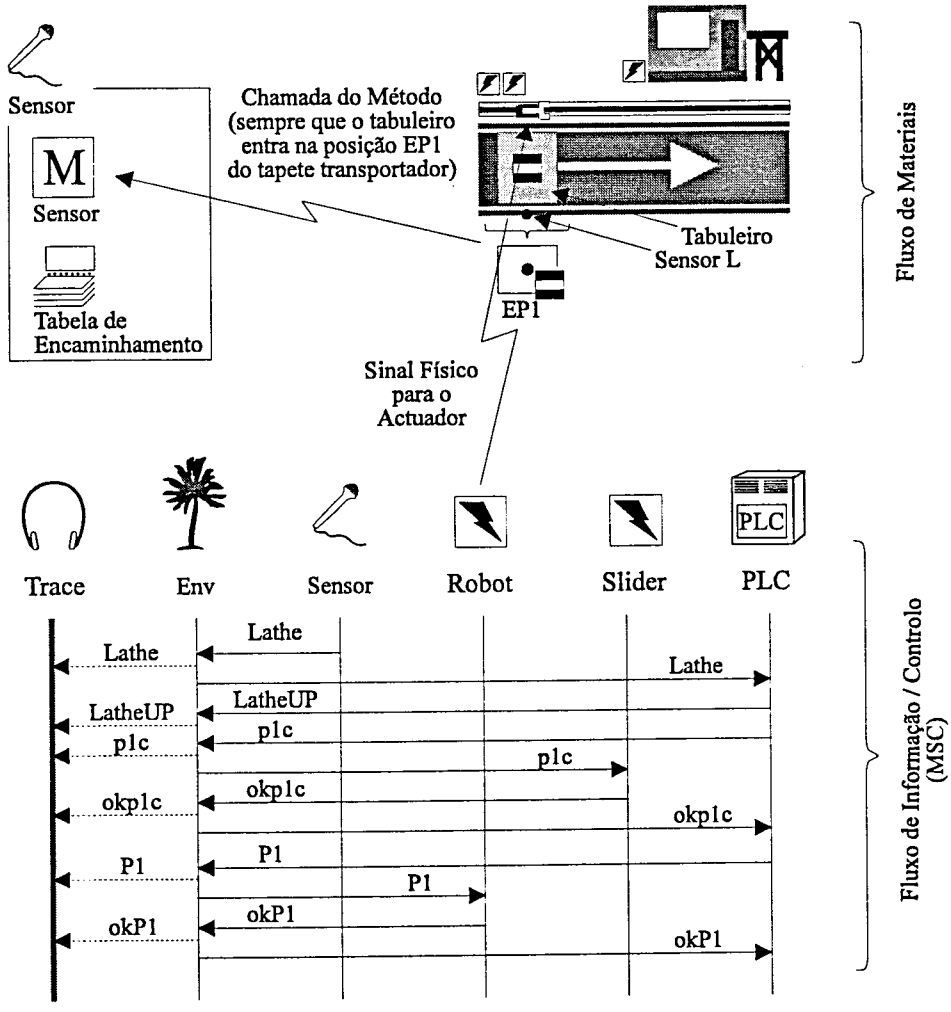


Figura 6.27 - Integração dos Fluxos de Materiais e de Informação / Controlo (Simulação)

Existe de facto um elevado grau de interacção entre a fluxo de materiais e o fluxo de informação / controlo. Considerando por exemplo o tapete transportador, vemos um bloco básico do Simple++, uma estação de trabalho, que foi utilizado para modelar a secção relevante do tapete. Do ponto de vista do modelo o tapete transportador é um elemento de transporte, um objecto com um determinado comprimento e que se descola a uma velocidade conhecida quando em movimento. Assim sendo, tendo o conhecimento do comprimento do tabuleiro sabemos quantos blocos básicos devem constituir o tapete, uma vez que cada um destes blocos apenas pode conter um tabuleiro de cada vez. A velocidade do tapete será então modelada por indicação do tempo de processamento da estação de trabalho. A esta secção do tapete chamamos EP1, e configuramos o seu controlo de entrada de forma a que o método *Sensor* seja chamado sempre que nela entre o Tabuleiro. Deste modo é feita a ponte entre o sensor electromecânico e o sinal de informação enviado através do ambiente para o PLC (*Lathe*). O sinal *LatheUP* é então enviado pelo PLC para o ambiente sendo o *actuador* responsável por agir sobre o modelo físico. Do ponto de vista da modelação, o *actuador* que deve ser visto como um "driver" de acesso ao sistema físico. O

actuador é activado através do envio de um sinal para o *Ambiente* no decorrer de uma transição de estado num processo SDL, e traduz-se simplesmente na execução de um método do Simple++ que irá actuar no modelo físico por forma a traduzir o comando identificado pelo sinal recebido (por exemplo mudar a cor do tabuleiro indicando que ele foi levantado).

Simulação e Modelação Técnico-Organizacional

Introdução e Contexto

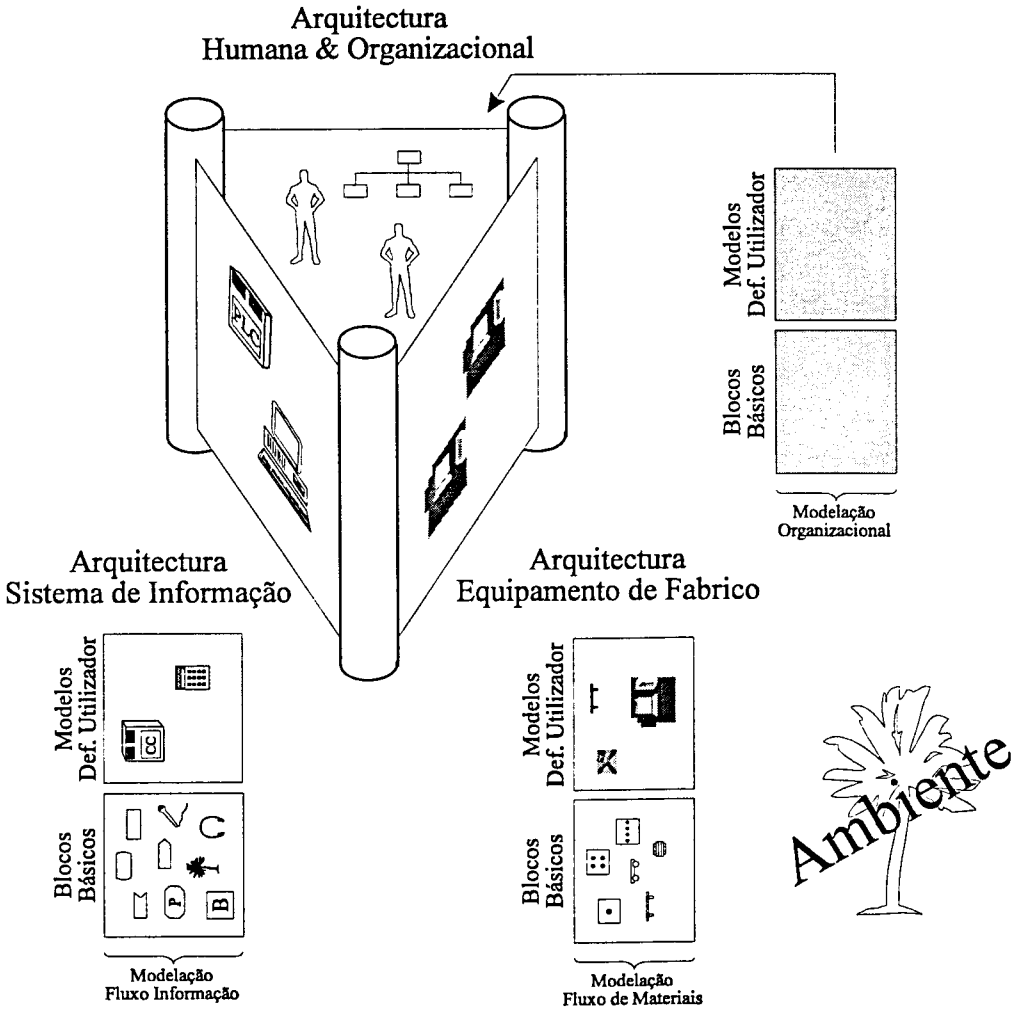


Figura 6.28 - A Utilização da Bancada de Modelação no Suporte às Arquitecturas de Implementação da Arquitectura de Purdue

No decorrer do capítulo anterior foi abordada a necessidade de dispor de ferramentas de modelação no âmbito do projectos de integração empresarial. Foi também salientada a importância de ferramentas de apoio à decisão, baseadas na modelação e na simulação, que ajudassem à avaliação de alternativas para as diversas arquitecturas de implementação da Arquitectura de Referência de Purdue (PERA) (retomadas na figura 6.28). Foi nesse capítulo dada particular relevância ao suporte da modelação da Arquitectura do Equipamento de Fabrico (AEF), tendo o corrente capítulo abordado por seu lado a modelação da Arquitectura do Sistema de Informação (ASI). Neste contexto foi estudada a

integração entre estas duas arquitecturas através da utilização dos conceitos de interacção de *sensor* e de *actuador* para modelação da interface entre elas.

A modelação da ASI bem como da AEF foi pois coberta com os anteriores desenvolvimentos descritos, restando a Arquitectura Humana e Organizacional (AH&O). Neste ponto será brevemente apresentada a reutilização das facilidades de modelação em SDL desenvolvidas pelo autor na construção de novas primitivas para suporte de trabalho de modelação técnico-organizacional. As novas primitivas de modelação, além de permitirem a modelação das unidades organizacionais, permitem como veremos modelar a interface entre a AH&O e a ASI.

Descrição

Fruto de um trabalho de investigação visando contribuir para o desenvolvimento interdisciplinar de sistemas integrados de fabrico, resultou uma ferramenta (OBsim - Simulação do Comportamento Organizacional) baseada em simulação de suporte à modelação e avaliação de estruturas organizacionais e de tecnologias de informação [7,145,146,147]. O objectivo desta ferramenta é ajudar no desenvolvimento conjunto da organização e das aplicações informáticas para suporte ao fabrico. Para isso, além dos indicadores de desempenho quantitativos básicos, tais como o tempo de realização das tarefas ou a taxa de ocupação da unidade organizacional, foram considerados critérios de análise sócio-técnica que são avaliados directa ou indirectamente pelo sistema. Exemplos destes critérios são: o grau de independência da unidade organizacional, a interdependência das tarefas, a regulação e a auto-regulação, o grau de acoplamento grupo de trabalho / sistema técnico e o grau de integração e dependência técnica.

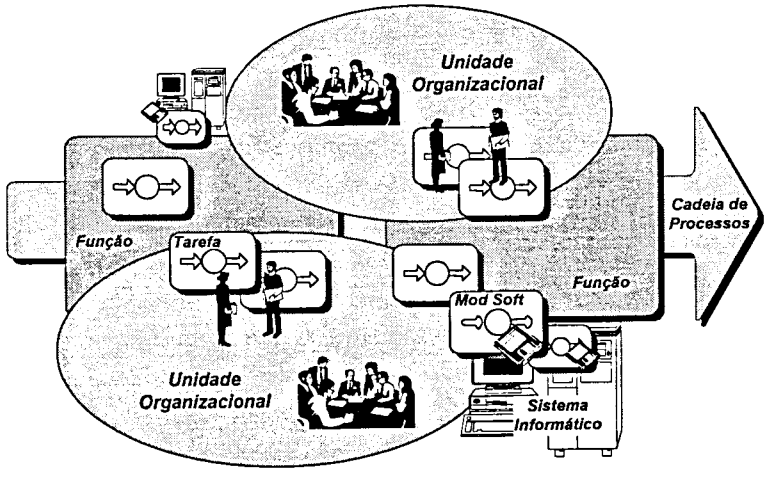


Figura 6.29 - Blocos Construtivos do OBsim

O modelo técnico-organizacional em que se baseia OBsim é delineado na figura 6.29. São mostradas as classes funcionais básicas: unidade organizacional, tarefa, função, módulo de software, sistema informático. A rede de informação é modelada por duas classes básicas: informação operacional e de monitorização. Durante a modelação é atribuído a estas classes um papel de interacção de acordo com a tarefa emissora e receptora da informação. Assim, a interacção entre unidades organizacionais, funções e sistemas informáticos pode ser modelada para posterior avaliação dos critérios de análise sócio-técnica.

Um exemplo da aplicação destes critérios é a avaliação da independência de uma unidade organizacional, um factor importante de efectividade. A minimização da dependência entre unidades organizacionais está em linha com os actuais paradigmas da

organização: ilhas de produção, grupos de trabalho autónomos, células de fabrico. O OBsim avalia este critério de uma forma estrutural e quantitativa. Quando um fluxo de informação tem um papel de interacção entre unidades organizacionais liga duas tarefas atribuídas a unidades organizacionais diferentes, representando uma ligação de dependência. A ligação pode ainda ser classificada como um fluxo operacional ou de monitorização. Durante uma sessão de simulação pode ser registada a intensidade dos fluxos de informação entre unidades organizacionais. Quanto maior a intensidade deste fluxo, menor a independência da unidade organizacional.

SDL como modelo descritivo

Na implementação do modelo técnico-organizacional desenvolvido recorreu-se à ferramenta de simulação Simple++ e sua extensão para a modelação em SDL. O modelo SDL foi usado para a descrição do comportamento das classes representando as funções organizacionais e os sistemas informáticos no âmbito do modelo técnico-organizacional.

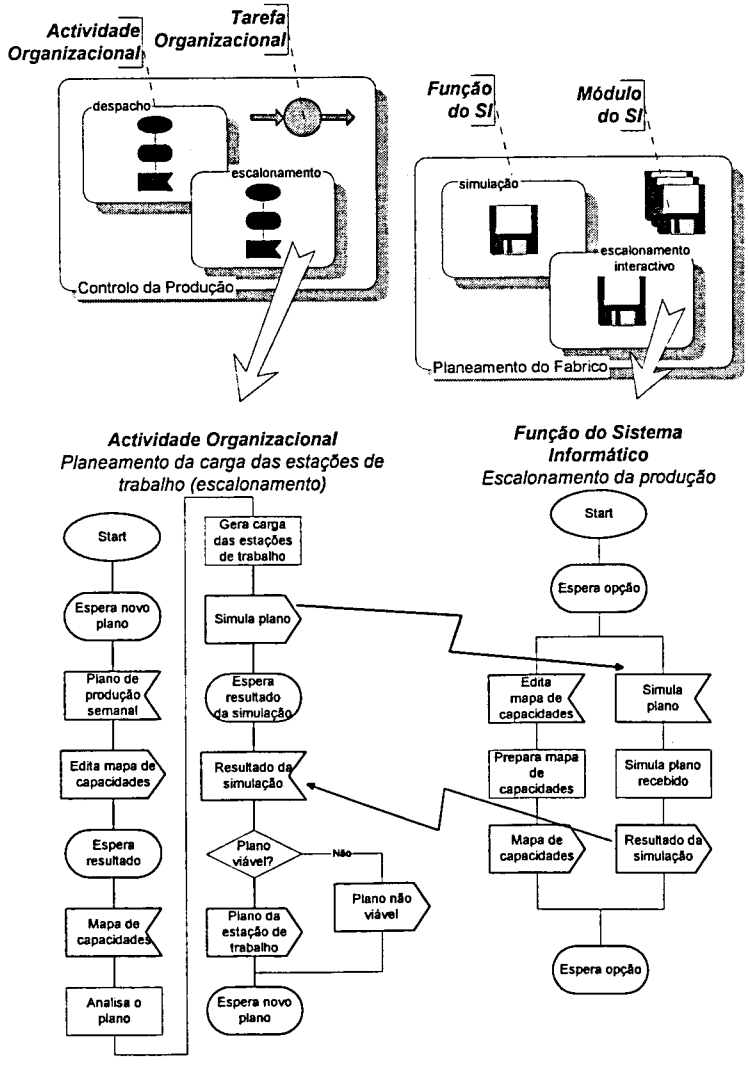


Figure 6.30 - Descrição do Comportamento das Tarefas Organizacionais e das Funções do Sistema Informático usando SDL

A estrutura de decomposição funcional dos blocos Função Organizacional e Sistema Informático reflecte a estrutura da linguagem de descrição SDL: a entidade Sistema representa uma Função Organizacional ou um Sistema de Informático. A primeira é

decomposta sucessivamente em Tarefas - traduzidas nas entidades Bloco SDL - e em Actividades - traduzidas nas entidades Processo SDL. Na figura 6.30 apresenta-se um exemplo de uma tarefa, de controlo da produção, e de uma das actividades que a compõe, o escalonamento da produção. A classe Sistema Informático é também decomposta sucessivamente em Módulos - traduzidos nas entidades Bloco SDL - e em Funções - traduzidas nas entidades Processo SDL. Novamente na figura 6.30, e considerando um sistema informático de apoio à produção, é dado um exemplo de um módulo de Planeamento do Fabrico e de uma das suas funções, o escalonamento da produção.

Os fluxos de informação entre os vários componentes do modelo são associados à entidade Sinal e modelam a interacção intra- e inter-unidades organizacionais e entre estas e os sistemas informáticos que as apoiam. Na figura 6.30 apresenta-se ainda um exemplo deste último tipo de interacção.

6.7 O Posicionamento do Ambiente Integrado da Bancada de Modelação na Matriz de Avaliação GERAM

No capítulo anterior foi analisado o posicionamento do PROFIT na matriz de avaliação GERAM. Neste parágrafo será realizado um estudo semelhante, tendo em atenção não o PROFIT mas apenas o ambiente da Bancada de Modelação (BM) [122].

A BM foi construída sobre uma plataforma de modelação de fluxo de materiais, o Simple++. O ambiente resultante providencia assim a integração entre a linguagem de modelação original (SimTalk) e um novo conjunto de primitivas de suporte à modelação em SDL. No posicionamento da BM na matriz GERAM, será ainda considerada a utilização da ferramenta de análise estruturada (CASE Teamwork) suportando a definição da estrutura do sistema distribuído, assim como a importação desta descrição para o ambiente da BM.

No capítulo anterior foi apresentada a contribuição do Simple++ para a modelação dos Produtos e da Arquitectura do Equipamento de Fabrico. Assim sendo, vamos apenas localizar as novas ferramentas disponíveis, i.e. a ferramenta de modelação em SDL e a ferramenta CASE de apoio à análise do sistema, tendo em atenção o ciclo de desenvolvimento apresentado na figura 6.20 (página 187).

Conforme ilustra a figura 6.31 a contribuição do ambiente disponível na BM do ponto de vista das ferramentas foi consideravelmente estendida relativamente ao descrito no final do capítulo anterior. Na análise e definição de requisitos, existe o apoio da ferramenta CASE na definição da estrutura do sistema distribuído assim como na caracterização dos seus componentes.

No ambiente de projecto, a linguagem de descrição formal fornece um conjunto de blocos construtivos genéricos a serem usados tanto na especificação do modelo de dados como do comportamento. Estes blocos genéricos podem ser utilizados na construção de novos blocos parciais, por exemplo um módulo servidor de alarmes, ou um seu sub-módulo, uma interface à base de dados, etc. Os módulos parciais poderão ser reutilizados na construção de novos modelos dedicados através da sua particularização, por exemplo configurando o servidor de alarmes de forma a adaptá-lo às necessidades particulares de um caso de aplicação prática. Ao nível do modelo particular, a combinação do modelo de simulação com um protótipo da interface com o utilizador da aplicação contribui para uma melhor avaliação do sistema por parte dos utilizadores finais, assim como para a identificação precoce de problemas de interacção entre o sistema e o seu operador. A ferramenta CASE apoia ainda a fase de projecto, providenciando os blocos construtivos

genéricos típicos das ferramentas de análise estruturada. Os blocos particulares fornecidos por esta ferramenta contribuem para a estruturação do sistema, fornecendo nomeadamente estruturas reutilizáveis. Estas estruturas poderão surgir com vários níveis de abstracção, podendo representar tanto sub-estruturas de um módulo como estruturas de interligação de múltiplos módulos de aplicação no ambiente distribuído.

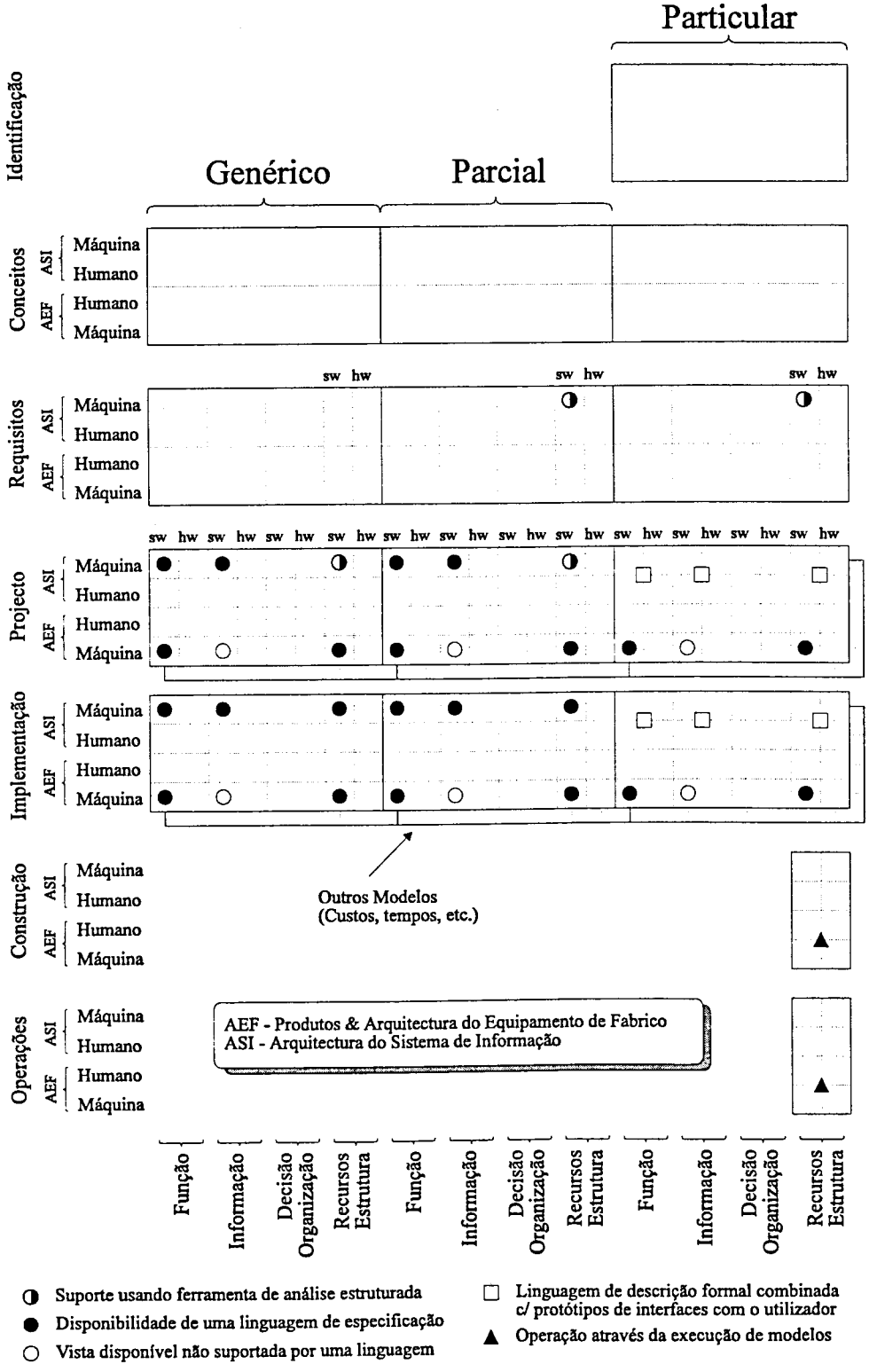


Figura 6.31 - Matriz GERAM para a BM e o ambiente envolvente

Na fase de implementação não é mais utilizada a ferramenta CASE, sendo a estrutura do seu modelo convertida numa combinação adequada de módulos instanciados de classes SDL. Na fase de implementação, embora não seja ainda suportada a geração de código, foi mantida a contribuição da BM descrita para a fase de projecto. Este facto deve-se fundamentalmente ao considerável suporte que o ambiente da BM fornece tanto ao integrador de sistemas como à equipa de desenvolvimento.

6.8 Conclusões

No capítulo anterior foram apresentadas as ferramentas de apoio à integração fabril disponíveis no PROFIT, tendo sido dada particular relevância à ferramenta de modelação usada no âmbito da modelação da Arquitectura do Equipamento de Fabrico. Uma análise no âmbito da matriz de avaliação GERAM demonstrou entretanto que o PROFIT não fornecia qualquer ferramenta de modelação da Arquitectura de Sistema de Informação. Verificou-se de facto que "apenas" estavam a ser fornecidos módulos de aplicação configuráveis integrados através de uma infra-estrutura comum.

O presente capítulo, além de abordar o tema central em torno do qual foi realizado o trabalho que deu origem esta dissertação, pretendeu dar uma resposta a este problema na esfera das ferramentas de modelação usadas no âmbito do PROFIT. Após algumas considerações iniciais relativamente ao modelo executável do sistema de informação proposto pelo CIMOSA, o único existente actualmente desenvolvido com o objectivo de responder aos problemas de integração fabril, concluiu-se que ele não é de facto executável pelo facto de nomeadamente não existirem ainda compiladores para a linguagem AID ("AMICE Implementation Description").

A utilização do CIMOSA não foi pois o caminho seguido para complementar as já existentes facilidades de modelação do PROFIT. O objectivo que se desejava atingir era não só a modelação do sistema de informação mas também a integração do modelo da Arquitectura do Sistema de Informação com o modelo já disponível da Arquitectura do Equipamento de Fabrico.

A obtenção de um ambiente integrado obrigava assim à selecção de uma linguagem de modelação do sistema de informação capaz de poder ser integrada com a já disponível e fornecida pelo Simple++. A vantagens resultantes da utilização de técnicas de descrição formal (TDF), assim como a selecção de uma TDF standard cujas primitivas permitissem a sua integração no Simple++, levou à utilização do SDL, disponibilizando uma representação gráfica simples, de fácil compreensão e rápida aprendizagem.

Uma breve apresentação do SDL, incluindo a descrição dos seus blocos construtivos, precedeu a descrição das suas primitivas básicas construídas no Simple++: as classes construídas para o suporte à modelação (classes *Sistema*, *Bloco* e *Processo)*. No decorrer dessa descrição foram apresentadas e justificadas algumas das particularidades desta implementação do SDL derivadas de restrições impostas pelo Simple++, bem como as opções tomadas. Apresentadas as funcionalidades do SDL disponíveis no protótipo desenvolvido foi feita a sua avaliação, sendo enumeradas as funcionalidades não disponíveis nesta implementação. Numa perspectiva de futuro, foram ainda indicadas quais as primitivas que a um breve prazo poderiam ser acrescentadas, bem como as facilidades do Simple++ requeridas para a sua mais eficiente implementação. Foi finalmente abordada a possibilidade de exportar uma especificação realizada em SDL no Simple++ para outras ferramentas usando o SDL-PR. A apresentação da ferramenta SDL no Simple++ culminou

com a descrição da vista que a Bancada de Modelação fornece sobre a integração do sistema de informação com o sistema de fabrico.

A descrição do ambiente de desenvolvimento de aplicações de gestão fabril seguiu-se à apresentação deste ambiente integrado de modelação. Neste contexto, foi dada particular importância à utilização do ambiente integrado no desenvolvimento, teste e integração de aplicações de software. A apresentação do ciclo de desenvolvimento, incluindo utilização de uma ferramenta de análise estruturada integrada com a BM desenvolvida, tornou evidente a vantagem da disponibilidade de blocos construtivos genéricos e parciais na realização rápida de protótipos de aplicações de software para gestão fabril.

A descrição da BM procurou dar particular relevância às vantagens resultantes da utilização do ambiente integrado. Foi assim demonstrada a utilização da BM assistindo o teste & integração passo-a-passo de aplicações, através da substituição sucessiva dos módulos simulados no modelo pelas suas versões executáveis. Neste contexto, foi demonstrada a utilização das facilidades de rastreio ("trace") na verificação e validação da interacção entre os diversos componentes do modelo em aplicações distribuídas. A utilização de dois exemplos, o "Controlador de Célula" e o teste e integração do executável da aplicação de Alarmes com as aplicações de Qualidade, Manutenção e Simulação / Escalonamento, permitiu uma melhor ilustração da forma como o ambiente integrado pode ser utilizado. Foi finalmente realizada a descrição do modelo de monitorização / controlo do fluxo de informação, fazendo um paralelo com os modelos utilizados na gestão do fluxo de materiais. A importância da visualização na análise foi também destacada. Estando fora do âmbito deste trabalho o estabelecimento de indicadores de desempenho para o sistema de informação, este aspecto foi abordado apenas numa perspectiva de utilização de uma ferramenta através da qual o utilizador do sistema deverá ser capaz de estabelecer o seu modelo matemático de avaliação de desempenho.

A avaliação última da ferramenta desenvolvida deverá sem dúvida ser realizada por um utilizador que não a conheça. Tal oportunidade permitiu que esta ferramenta fosse utilizada em duas situações distintas. No primeiro exemplo, foi construído o modelo da Célula de Fabrico Integrado existente no INESC-Porto, enquanto que no segundo as primitivas para a modelação em SDL foram reutilizadas no âmbito de um trabalho de modelação técnico-organizacional.

O sucesso evidenciado em ambos os casos na utilização do ambiente de modelação integrado, justifica sem dúvida que o protótipo desenvolvido seja transformado numa efectiva ferramenta de modelação. A construção desta ferramenta passa naturalmente pela implementação das já referidas funcionalidades do SDL ainda não disponíveis. Além disso, algumas das funcionalidades disponíveis na mais recente versão do Simple++ assim como em futuras versões, poderão com vantagem tornar mais simples a utilização das primitivas de modelação desenvolvidas. Este objectivo será certamente facilitado pela relação de cooperação existente entre o INESC e a empresa responsável pelo desenvolvimento do Simple++, a AESOP[3].

---

[3]A AESOP (Stuttgart, Alemanha) é a "software house" responsável pelo desenvolvimento do Simple++. O grupo de Engenharia de Sistemas de Produção do INESC-Porto, além de "Beta-tester" do Simple++ é parceiro da AESOP no Consórcio PROFIT e no Projecto ESPRIT 8865 Real-I-CIM.

Capítulo 7

CONCLUSÃO

7.1 Introdução

Esta dissertação tratou alguns dos problemas relacionados com a Integração de Sistemas de Fabrico. Numa primeira parte foi realizada uma introdução aos sistemas de fabrico nas suas diferentes vertentes suportada pela pesquisa bibliográfica efectuada, tendo-se seguido a apresentação das arquitecturas de referência para a integração empresarial mais conhecidas da actualidade. Neste âmbito foi dada especial relevância ao trabalho que tem sido realizado no âmbito de grupos de trabalho internacionais, tais como o IFIP, o IFAC e o CEN.

A introdução dos sistemas discretos de fabrico, bem como a restrição do âmbito desta dissertação aos denominados sistemas de execução fabril, permitiram o enquadramento do tema seguinte, o problema das infra-estruturas de suporte aos sistemas de gestão de operações fabris. Surgem no âmbito deste tema as primeiras contribuições do autor, através das aplicações de Supervisão e no Módulo de Acesso Distribuído à Instalação Fabril (MADIF), desenvolvidas no decorrer do projecto ESPRIT SHOP-CONTROL. A contribuição original do autor, consistiu na transposição de um conjunto de conceitos que tradicionalmente usados no controlo dos processos contínuos através dos sistemas SCADA ("Supervisory Control and Data Aquisition"), para os processos discretos. A aplicação de Supervisão, suportada por um gestor de regras permitindo uma configuração muito flexível, foi a raiz dos Quadros de Bordo Dinâmicos com indicadores de desempenho que integram os conceito de SCADA e de EIS ("Executive Information System"). Através do MADIF, estes conceitos, oriundos respectivamente do controlo de processos e da gestão, foram finalmente estendidos ao nível do operador e do encarregado fabril.

No quadro das arquitecturas de referência apresentadas, a Arquitectura de Referência de Purdue (PERA) assumiu particular importância, motivando a identificação das necessidades de modelação para a integração empresarial. Foram assim identificadas as três arquitecturas de implementação de Purdue, sendo constatada a relevância da modelação

para a simulação no processo de integração do sistema de fabrico. É neste contexto que são introduzidos os conceitos de modelos reutilizáveis e de modelos executáveis, salientando-se as vantagens resultantes da utilização de ferramentas de modelação orientadas ao objecto, bem como da combinação da simulação e do controlo. É também abordada com algum detalhe a modelação dos sistemas de informação e dos equipamentos de fabrico. Como ilustração e suporte destes conceitos são apresentados dois exemplos de modelos executáveis, o modelo do CIMOSA e o modelo do SHOP-CONTROL / PROFIT. Em tom de conclusão, é estudado o posicionamento do PROFIT na matriz de avaliação GERAM, sendo identificadas as áreas não cobertas pelas ferramentas disponibilizadas. O capítulo 6, no âmbito do qual é apresentada uma contribuição original do autor para a modelação integrada das Arquitecturas do Sistema de Informação e do Equipamento de Fabrico, surge como resposta à referida avaliação.

A Bancada de Modelação desenvolvida apresenta um ambiente de modelação com características originais. De facto, ferramentas baseadas na arquitectura CIMOSA, tais como o SEW-OSA, concentram a sua atenção na modelação do sistema de informação e das regras de controlo das operações não modelando o sistema físico de fabrico. A Bancada de Modelação, além de permitir a modelação da Arquitectura do Sistema de Informação usando a linguagem SDL, faculta ainda a modelação do sistema físico, i.e. recursos, produtos, fluxos de materiais, etc., usando a linguagem de simulação SimTalk do Simple++. Estes dois ambientes, suportados por linguagens de modelação adequadas, são integrados através dos conceitos interacção de *sensor* e *actuador*, resultando numa ferramenta poderosa de apoio ao ciclo de vida de um sistema integrado de fabrico.

No ponto que se segue, será detalhada a contribuição do trabalho realizado através dos resultados produzidos. É exemplo disso, o trabalho em curso de construção de uma ferramenta de modelação técnico-organizacional sobre a ferramenta de modelação em SDL desenvolvida pelo autor.

7.2 A Contribuição do Trabalho realizado

Vários foram os contributos / resultados produzidos ao longo deste trabalho. Nos parágrafos que se seguem são alinhadas as contribuições mais significativas que se reflectiram nas seguintes cinco vertentes, Protótipos & Demonstrações, Projectos, Publicações e Contributos para o Ensino:

- *Protótipos & Demonstrações*
  Foram desenvolvidos os protótipos das aplicações de Supervisão e do MADIF (descritas no capítulo quatro). A aplicação de Supervisão foi demonstrada nas várias apresentações públicas do projecto SHOP-CONTROL. Das exposições realizadas, é de salientar a sua apresentação no reunião final de avaliação do projecto SHOP-CONTROL (1993) realizada nas instalações da Cifial (Rio-Meão, Portugal), em Bruxelas no âmbito da ESPRIT ANNUAL CONFERENCE AND EXHIBITION (1992 e 1993), e no Workshop "Gestão Fabril Optimizada em Tempo Real" realizado no INETI, em Lisboa (1994). O MADIF, desenvolvido após a conclusão do projecto SHOP-CONTROL e entretanto abandonado, tem vindo a ser utilizado sempre que para demonstrações do PROFIT é necessário interligar as aplicações em ambiente UNIX a terminais com capacidade de processamento limitada.

  Foi ainda desenvolvido o protótipo da implementação do SDL no Simple++, que se traduziu na Bancada de Modelação apresentada no capítulo seis deste trabalho.

Este protótipo foi apresentado respectivamente numa conferência convidada e num workshop (intitulados respectivamente "Manufacturing and Information Systems Design through Integrated Modelling and Simulation", e "Production optimization through flexible integrated software tools") realizados na Universidade de Lima, por oportunidade da "I International Conference on Systems Engineering: The use of Information Technology and Operations Research Techniques in Business Management" (1995).

- *Projectos*
  O trabalho enquadrado nesta dissertação foi nuclear no projecto SHOP-CONTROL, tendo dele obtido motivação, razão de ser e validação e tendo nele vertido parte dos seus resultados.

  O trabalho desenvolvido foi ainda da maior importância na elaboração da proposta e do anexo técnico do projecto REAL-I-CIM, sendo disso um exemplo a evolução da aplicação de Supervisão para o actual Quadro de Bordo Dinâmico, apresentado no capítulo quarto.

  Por outro lado, a possível contribuição da Bancada de Modelação para a constituição de um ambiente de construção de protótipos, será de extrema relevância no âmbito de um projecto europeu que terá início até ao final do ano de 1995, o projecto ESPRIT 20544 X-CITTIC ("A Planning and Control System for Semiconductor Virtual Enterprises").
- *Publicações*
  Ao longo do período em que se desenvolveu este trabalho de dissertação, foram regularmente apresentadas à comunidade científica nacional e internacional os conceitos, as propostas e os resultados que foram surgindo no decorrer deste trabalho (ver referências). O debate construtivo de opiniões que emergiu destas apresentações foi certamente revelador da importância e actualidade dos problemas tratados.
- *Contribuições para o Ensino*
  As contribuições do trabalho realizado para o ensino conduziram ao estabelecimento de uma base sobre a qual se realizou uma dissertação de mestrado já terminada, e ao envolvimento na elaboração de uma proposta para uma nova disciplina do Mestrado em Engenharia Electrotécnica e de Computadores ("Sistemas Integrados de Fabrico").

7.3 Desenvolvimentos Futuros

O trabalho realizado pelo autor e apresentado nesta dissertação constituí de facto uma base para desenvolvimentos futuros. O facto de alguns desenvolvimentos estarem já em curso, como por exemplo as primitivas de modelação técnico-organizacional e a integração com a ferramenta CASE Teamwork, contribui certamente para um novo esforço na área da modelação empresarial na sequência deste trabalho.

Mas é sem dúvida a aplicação dos conceitos desenvolvidos ao longo desta dissertação à modelação da empresa virtual o desafio que se coloca a curto / médio prazo [148]. De facto, a introdução das denominadas "entidades" no âmbito da arquitectura genérica GERAM (figura 7.1), permite suportar a introdução do conceito de Empresa Virtual. O constatar da interacção existente entre as várias entidades envolvidas na empresa virtual (figura 7.1), leva a reflectir na importância de considerar não só as relações ao longo da cadeia logística

entre as denominadas entidades do tipo 3 (fornecedor do serviço / empresa de fabrico), mas também as interacções com a entidade 1 (gestão estratégica) e entidade 2 (empresa de consultadoria). Isto sem esquecer a entidade 4, o produto ou serviço produzido ou fornecido pela entidade 3, uma vez que ele constituí de facto a razão de existência das restantes entidades.

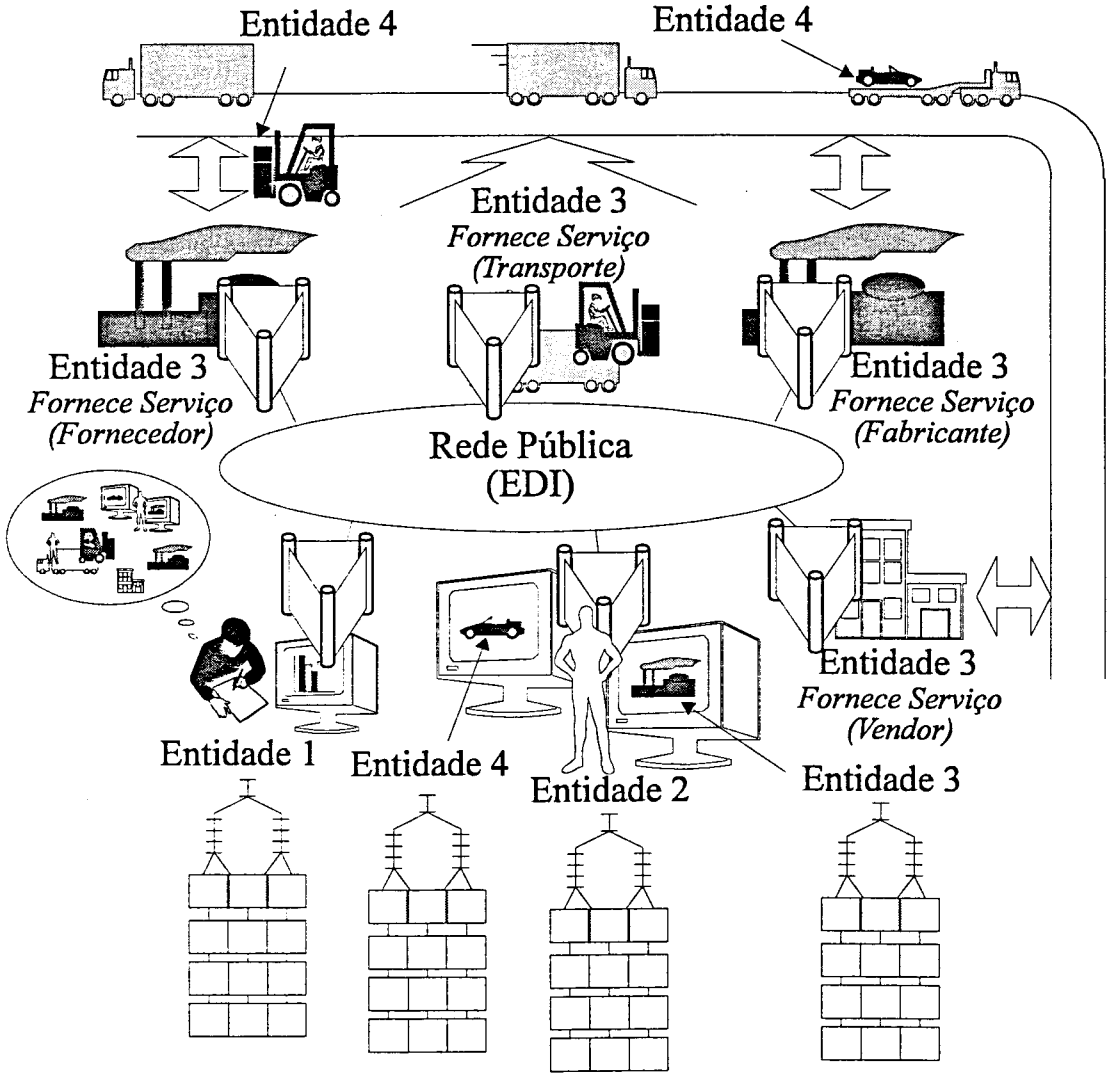


Figura 7.1 - A Modelação da Empresa Virtual

A figura 7.1 apresenta as entidades GERAM, já referidas no âmbito do capítulo 3, ilustrando ainda o facto de cada uma dessas entidades apresentar um ciclo de vida de acordo com a arquitectura de referência de Purdue. Neste contexto, a cada uma das entidades vai corresponder o ciclo de desenvolvimento das três arquitecturas de implementação de Purdue, as Arquitecturas do Equipamento de Fabrico, do Sistema de Informação e Humana e Organizacional.

A mesma figura ilustra ainda, para cada uma das entidades, o prisma representando as interfaces entre as três arquitecturas de implementação de Purdue. A definição dessas interfaces, além de ser necessária no interior da própria empresa, surge com uma maior relevância quando migramos para o conceito de empresa virtual. Neste caso, deverão ser incluídas todas as interacções que têm lugar atravessando as fronteiras da empresa, tais

como a transferência de dados através da rede pública, a transferência de produtos através de transporte físico (camiões, por exemplo), a transferência de energia, e o acordo de procedimentos organizacionais comuns.

A integração destas arquitecturas no contexto da empresa virtual levanta problemas muito interessantes e da maior relevância no âmbito da integração dos processos e actividades, exigindo tirar partido dos mais recentes desenvolvimentos tecnológicos. O conjunto de problemas envolvidos é de facto complexo, sendo da maior importância o estabelecimento de uma estratégia de migração do cenário actual para o cenário da empresa virtual que se deseja atingir. O impacto desta migração surge não só no nível tecnológico, mas ainda e sobretudo no nível organizacional. É neste contexto que a simulação pode ser utilizada com vantagem na avaliação e teste de cenários alternativos. A modelação para a simulação no âmbito da empresa virtual, deverá assim levar em consideração as múltiplas inter-relações presentes entre cada uma das entidades GERAM, e naturalmente entre cada uma das arquitecturas de implementação dessas entidades.

As ferramentas desenvolvidas ao longo desta dissertação, permitem esperar que as facilidades já disponíveis no âmbito da Bancada de Modelação, incluindo a modelação técnico-organizacional, possam ser utilizadas na modelação da empresa virtual. Estas facilidades, além de providenciarem todas as vantagens inerentes à utilização da simulação poderão servir de base ao estabelecimento e teste de novos paradigmas de integração inter-empresarial [148].

É importante observar que a relevância desta área de interesse emergente não se restringe às indústrias automóvel ou electrónica, aquelas que de uma forma mais visível corporizam o conceito de empresa virtual, mas estende-se a todas as empresas com centros de produção distribuídos e cadeias de subcontratação (indústria têxtil, de calçado, alimentar, etc.).

Referências

[1] Industry-University Consortium and Purdue Laboratory for Applied Industrial Control: An Implementation Procedures Manual for Developing Master Plans for Computer Integrated Manufacturing, Report Number 155, Purdue Laboratory for Applied Industrial Control, Purdue University, West Lafayette, IN (June 1992)

[2] J. J. Pinto Ferreira, C. Toscano, J. M. Mendonça: Infra-estruturas de Suporte aos Sistemas de Produção Integrada por Computador, Jornadas Nacionais de Projecto, Planeamento e Produção Assistidos por Computador, 17 a 19 de Maio 1993 Lisboa

[3] J. J. Pinto Ferreira, C. Toscano, L. Carneiro, J. M. Mendonça: Inter-Application Communication, ESPRIT Project 5478 SHOP-CONTROL, WP-2, task 4.3.6, February 1992.

[4] J. J. Pinto Ferreira, J. M. Mendonça: On-line Shop Floor Supervision and Control and Its Supporting Infrastructure, International Conference on Advances in Production Management Systems, APMS'93 28-30 September, Athens, Greece

[5] J. J. Pinto Ferreira, J. M. Mendonça: On-line Supervision in Shop Floor Control" Third International Conference on Flexible Automation and Integrated Manufacturing, FAIM'93 28-30 June 1993 Limerick, Ireland

[6] J. J. Pinto Ferreira : The Supervison Application in Real-I-CIM, Workpakage 3, Task 3.3, Report December 1994

[7] A. Lucas Soares, J. J. Pinto Ferreira, J. M. Mendonça: A Modelação e Simulação com Instrumentos de Análise Sócio-Técnica em Sistemas Integrados de Fabrico, 5.as Jornadas Nacionais de Projecto, Planeamento e Produção Assistidos por Computador - 30, 31 de Maio e 1 de Junho de 1995

[8] H.-J. Warnecke: The Fractal Company, Springer Verlag, Berlin u.a.

[9] Trabalho e Locais de Trabalho, em História da Vida Privada: Da Primeira Guerra Mundial aos nossos dias, Volume 5, Edições Afrontamento

[10] CIME Reaches a Turning Point, $CIME_{TM}$ GDS Publishing Ltd., Agosto/Setembro 1994

[11] Peter Wayner: EDI Moves the Data, October 1994, Byte

[12] Pierre-Marie Gallois : The new industrial challenge: Towards a "Lean Production", International Conference on Advances in Production Management Systems, APMS'93 28-30 September, Athens Greece

[13] Canon Production System, Creative Involvement of The Total Workforce, Compiled by the Japan Management Association, Translated by Alan T. Campbell, Productivy Press, ISBN 0-915299-06-2

[14] H. Yoshikawa: Intelligent Manufacturing Systems: Technical Cooperation That Transcends Cultural Differences, Information Insfrastructure Systems for Manufacturing (B-14) H. Yoshikawa and J. Gossenaerts (Eds) Elsevier Science B.V. (North Holland), 1993 IFIP

[15] Katsundo Hitomi: Manufacturing systems engineering: the concept, tis context and the state of the art, International Journal of Computer Integrated Manufacturing, 1990, Vol. 3, No. 5, 275-288

[16] August-Wilhelm Scheer: Computer Integrated Manifacturing, Towards the Factory of the Future, Second Revised and Enlarged Edition, Springer Verlag, December 1990

[17] Taylor, F. W. :Scientific Management, New York, Harper & Brothers

[18] V. Rmbold, B.O. Nnaji, A. Storr: Computer Integrated Manufacturing and Engineering, Addison-Wesley

[19] AFNOR, "Maintenance Industrielle"; Recueil de normes françaises; ed. AFNOR ("Association Française de Normalization"), 1988

[20] Paula A. S. Silva: Manutenção Condicionada Baseada na Análise de Vibrações, Tese de Mestrado em Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia de Universidade do Porto, Abril de 1994

[21] T.J.Williams et al. :Architectures for Integrated Manufacturing Activities and Enterprises, Information Infrastructure Systems for Manufacturing (B-14), H. Yoshikawa and J. Goossenaerts (Editors), Elsevier Science B.V. (North-Holland) 1993 IFIP

[22] Joriz, H.R.; Vernadat, F.B (1990): CIMOSA Part 1: total enterprise modelling and function view, Int'l Journal of Computer Integrated Manufacturing, Vol. 3, Nr., p. 144-156

[23] CIMOSA: Open System Architecture for CIM, ESPRIT Consortium AMICE (Eds.), 2nd, revised and extended edition, Springer-Verlag

[24] Ferenc Belina, Dieter Hogrefe and Amardeo Sarma: SDL with APPLICATIONS from PROTOCOL SPECIFICATION, Prentice Hall International, ISBN 0-13-785890-6

[25] Kurt Kosanke, Marco Mollo, Franco Naccari: CIMOSA and Manufacturing in Evolution, Proceedings of ESPRIT CIM-Europe, Sharing CIM Solutions J.K.H. Knudsen et al.(Eds) IOS Press, 1994

[26] G. Doumeingts, D. Chen, B. Vallespir, P. Fénié: GIM (GRAI Integrated Methodology) and its evolutions, A methodology to design and specify Advanced Manufacturing Systems, Information Insfrastructure Systems for Manufacturing (B-14) H. Yoshikawa and J. Gossenaerts (Eds) Elsevier Science B.V. (North Holland), 1993 IFIP

[27] Douneingts, G.: Methode GRAI de Comception des Systemes de Productice, These d'Etat en Automatique, Iniversite de Bordeaux, France, November 1984

[28] Douneingts, G., Vallespir, B., Darricar, D. and Roboam, M.: Design Methology for Advanced Manufacturing Systems, Computers in Industry, Vol. 9 No. 4, 271-295

[29] Reference Model for Computer Integrated Manufacturing, A Description From the Viewpoint of Industrial Automation, Williams, T.J. (Ed), CIM Reference Model Commitee, International Purdue Workshop on Industrial Computer Systems, Purdue Laboratory for Applied Industrial Control, Purdue University, West Lafayette, Indiana (may 1988); Published by Instrument Society of America, Research Triangle Park, North Carolina (1989)

[30] T.J.Williams: The Purdue Enterprise Reference Architecture (B-14), H. Yoshikawa and J. Goossenaerts (Editors), Elsevier Science B.V. (North-Holland) 1993 IFIP

[31] Hong Li: A formalization and extension of the Purdue Enterprise Reference Archhitecture and the Purdue Methodology, A Thesis Submitted to the Faculty of Purdue University by Honh Li, In partial fulfillment of the requirements for the Degree of Doctor in Philosophy, December, 1994

[32] M. Bitton, J.Colom : Shop-Floor Monitoring and Control Systems: Some Lacks to Fill up..., IFIP transactions, Advances in Production Management Systems (B-13), I.A. Pappas and I.P. Tatsiopoulos (Ed.) Elsevier Science Publishers B.V. (North-Holland)

[33] Fenie, P.: GRAICO_D: Méthode d'analyse et de conception du système d'exploitation, Thése de Doctorat, Automatique, Univertité Bordeaux I

[34] F. Vernadat: CIMOSA:Enterprise Modelling and Enterprise Integration Using a Process-Based Approach (B-14), H. Yoshikawa and J. Goossenaerts (Editors), Elsevier Science B.V. (North-Holland) 1993 IFIP

[35] CIMOSA Open Systems Architecture for CIM, Technical Baseline, Part I - Business Oriented Modelling, CIMOSA Association / Version 3.0, July 1994, ADITEC, Steinbachstr, 25 D-52074 Aachen / Germany

[36] CEN/CENELEC. Computer Integrated Manufacturing - Systems Architecture - Framework for Enterprise Modelling, ENV 40 003, Janeiro 1990

[37] ISO. Framework for Enterprise Modelling ISO TC 184 SC5 WG1, 1983

[38] A.Bienert, D.Schonenberger: Computer Aided Enterprise Modelling - A global Approach IFIP transactions, Advances in Production Management Systems (B-13), I.A. Pappas and I.P. Tatsiopoulos (Ed.) Elsevier Science Publishers B.V. (North-Holland)

[39] CEN/CENELEC. Computer-Integrated Manufacturing - Systems Architecture - Framework for Enterprise Modelling, ENV 40 003, January 1990

[40] Gary J. Colquhoun, Ray W. Baines, Roger Crossley: A state of the art review of IDEF0, Int'l Journal of Computer Integrated Manufacturing, 1993, Vol.6, No.4, 252-264

[41] Scheer, A.-W: Architecture for Integrated Information Systems (B-14), H. Yoshikawa and J. Goossenaerts (Editors), Elsevier Science B.V. (North-Holland) 1993 IFIP

[42] A.-W. Scheer: Architecture of Integrated Information Systens, Foundations of Enterprise-Modelling, Springer-Verlag 1992

[43] QCIM/WG QEM - ESPRIT CIMOSA, Comparison CIMOSA-IEM Modelling Constructs and methodology, Draft, CIMOSA Association /IPK, Document version 1.5 Date: 95-02-21

[44] K. Mertins, H. Edeler, R. Jochem, J. Hofmann: Object-Oriented Modelling and Analysis of Business Processes, European workshop on Integrated Manufacturing Systems Engineering, IMSE'94, 12-14 December, 1994, Grenoble

[45] Theodore J. Williams: Development of GERAM, a Generic Enterprise Reference Archirecture and Enterprise Integration Methodology, European workshop on Integrated Manufacturing Systems Engineering, IMSE'94, 12-14 December, 1994, Grenoble

[46] Marcos Wilson C. Aguiar, Richard H. Weston: SEW-OSA: Systems Engineering Workbench Centred on CIMOSA, The 10th ISPE/IFAC International Conference on CAD/CAM, Robotics and Factories of the Future, August 21-24, 1994 Ottawa, Canada

[47] Marcos Wilson C. Aguiar, Ian Coutts and Richard H. Weston : Model Enactment as a Basis for Rapid Prototyping of Manufacturing Systems, IMSE'94-European Workshop on Integrated Manufactuing Systems Engineering, Grenoble, France 12-14 December 1994, General Chairman: Dr. F. Vernadat

[48] Dirk Solte: A Software Engineering Paradigm as a basis for Enterprise Integration in (Multi-)Client / Server Environments, European workshop on Integrated Manufacturing Systems Engineering, IMSE'94, 12-14 December, 1994, Grenoble

[49] ISO/IEC JTC1: Portable Operating System Interface for Computer Environments

[50] J. J. Pinto Ferreira: Interface Homem-Máquina para um Sistema de Geração e Análise de Tráfego em Redes de Comunicação, Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto - 1990

[51] Hans Bukow: Graphic User Interface Impacts Control Architecture Development, Control Engeneering March 1991

[52] Introduction to OSF™DCE, Open Software Foundation Revision 1.0, Prentice Hall, Englewood Cliffs, New Jersey 07632

[53] Marcos Wilson C. Aguiar, Richard H. Weston: CIMOSA and Petri-Nets for CIM Systems Modelling and Simulation, Third International Conference on Flexible Automation and Integrated Manufacturing, FAIM'93 28-30 June 1993 Limerick, Ireland

[54] Verall: Unity Doesn't Imply Unification or Overcomming Heterogeneity Problems in Distributed Software Engineering Environments, The Computer Journal, Vol. 34, No. 6, 1991

[55] Petrie : Enterprise Integration Modelling, Proceedings os the first international conference, MIT Press Canbridge, 1992

[56] The Common Object Request Broker: Architecture and Specification, Digital Equipment Corporation, Hewlett-Packard Company Hyperdesk Corporation, NCR Corporation, Object Design Inc, SunSoft Inc. - OMG Document Number 91.12.1 Revision 1.1 Draft 10 December 1991

[57] Paul Strauss: TCP/IP Winning the Backbone Battle, Datamation, October 1993

[58] Cleaning Up the MES, The Advanced Manufacturing Research (AMR) Report, Volume VI, Number 1, January, 1992

[59] Álvaro de Oliveira, Engelbert Westkamper, Henk Bolk: AICIME - A technology and knowledge transfer methodology supported by industrial organization re-engineering and avanced CIME architecture based on ESPRIT products, Proceedings of ESPRIT CIM-Europe, Sharing CIM Solutions J.K.H. Knudsen et al.(Eds) IOS Press, 1994

[60] Simple++ User Manual, AESOP GmbH, Konigstrase 82, 7000 Stuttgart 1

[61] Becker, B., Schulte, J.: Specification for the Model Editor, Shop Floor Modelling Language, schedulling and dispatching algorithms, report for ESPRIT Project 5478, February 1992

[62] Vernadat, F.: Modelling and Analisys of Enterprise Information Systems with CIMOSA, Proceedings of Sixth CIM-Europe Annual Conference. 15-17 May 1990 Lisbon Portugal

[63] ECMA. A reference model for frameworks of computer-assisted software engineering environments. Genéve: European Computer Manufacturers Association. Décembre 1990 N° TR/55

[64] ECMA. Portable Common Tool Environment (PCTE). Genéve: European Computer Manufacturers Association. Décembre 1990 N° 149

[65] Eric Lutherer, Michel Martinez, Bruno Querenet, Joel Favrel: Towards a CIM Engineering Workbench, IMSE'94-European Workshop on Integrated Manufactuing Systems Engineering, Grenoble, France 12-14 December 1994

[66] D. Schefstrom: System Development Environments: Contemporary Concepts, em Tool Integration, Environments and Frameworks, editado por D. Schefstrom e G. van den Broek, Wiley Professional Computing 1993

[67] Philippe Larvet: Analyse des systèmes: de l'approche fonctionelle à l'approche object, InterÉditions, Paris 1994

[68] C. Toscano: Message Transmission System Specification, ESPRIT Project 5478, October 1992

[69] Helena Brás: Message Trasmission System Notification, INESC AIGE - Porto, May 1994

[70] OSF/Motif Style Guide, OSF Eleven Cambridge Center Cambridge MA 02142

[71] Open Windows User's Guide, Sun Microsystems, Inc.

[72] L. Carneiro, P. Magalhães, N. Romão, J. Mendonça: Shop Floor Data Base Specification, ESPRIT Project 5478 Workpackage 2, Task 2.1.2, July 3, 1991

[73] Paula Magalhães: Os Sistemas de Informação no Suporte à Integração de Aplicações de Gestão Fabril, Tese de Mestrado, Abril 1994

[74] B. Becker, J. Schulte, J. Arnold: Schedulling and Control Application Specification, ESPRIT Project 5478 SHOP-CONTROL, WP-5, task 5.4, February 1992

[75] ISO TC 184 Manufacturing Message Specification (MMS)

[76] J. J. Pinto Ferreira, P. Martel, J. Torres, J. M. Mendonça: Graphics Operator Interface, ESPRIT Project 5478 SHOP-CONTROL, WP-4, Tasks 4.1.1 & 4.1.2, July 1991.

[77] Aaron Marcus: Corporate Identity for Iconic Interface Design: The Graphic Design Perspective IEEE Computer Graphics & Applications, December 1984

[78] J. J. Pinto Ferreira, P. Martel, J. Torres, J. M. Mendonça: Man-Machine Interface Generator System, ESPRIT Project 5478 SHOP-CONTROL, WP-4, task 4.2, February 1992.

[79] J. J. Pinto Ferreira, P. Martel, J. Torres, J. M. Mendonça: Supervision, ESPRIT Project 5478 SHOP-CONTROL, WP-2, task 4.3.1, February 1992.

[80] J. J. Pinto Ferreira, J. Torres, J. M. Mendonça: Supervision, ESPRIT Project 5478 SHOP-CONTROL, November, 1992.

[81] Faria J.P., Ranito JV: Manual do programador parte II, linguagem de regras e comandos, user manual Saga 1.10, INESC PORTO, April 1992

[82] Robert H. Reck, James R. Hall: Designing Executive Information Systems, Handbook of MIS Management, Robert E. Umbaugh (Ed), Auerbach Publishers

[83] Wolfgang Meyer and Randolf Isenberg: Knowledge-based factory supervision: EP 932 results, International Journal of Computer Integrated Manufacturing, 1990, Vol. 3 and 4, 206-233

[84] L. Carneiro, P. Magalhães, N. Romão, J.M. Mendonça: Shop Floor Interaction Handler Specification. ESPRIT Project 5478 Report - October 28, 1992

[85] J.M. Mendonça: Condition Monitoring Concepts for Real-I-CIM, Real-I-CIM internal report.

[86] Katsuhiko Ogata: Engenharia do Controle Moderno, Prentice /Hall do Brasil

[87] Ian Sommerville, Software Engineering, Third Edition, Addison-Wesley, ISBN 0-201-17568-1

[88] ANSA, ANSA Reference Manual, Cambridge, Architecture Projects Management, Ltd, 1989

[89] J.C. Wortmann: Enterprise Modelling: purposes and means, IFIP transactions, Advances in Production Management Systems (B-13), I.A. Pappas and I.P. Tatsiopoulos (Ed.) Elsevier Science Publishers B.V. (North-Holland)

[90] ISO/TC97/SC21/WG1/DIS9074 Estelle - A Formal Description Technique Based on an Extended State Transition Model, 1987

[91] Rachid Sijelmassi, Richard J. Linn: Guidelines for using Estelle to specify OSI services and protocols, Computer Networks and ISDN Systems 23 (1992), 343-362, North-Holland

[92] L. Logrippo, M Faci, Haj-Hussein: An introduction to LOTOS: learning by examples, Computer Networks and ISDN Systems 23 (1992), 325-342, North-Holland

[93] Joe H. Mize, Hemant C. Bhuskute, David B. Pratt, Manjunath Kamath: Modelling of Integrated Manufacturing Systems Using an Object-Oriented Approach, Volume 24, Number 3, IIE Transactions, July 1992

[94] Hubert Tardieu, Arnold Rochfeld, René Coletti: Méthode Merise, Vol 1: Principes e outils, Les Editions d'Organisation, Paris 1983

[95] D. Marca, C.McGowan: SADT: Structured Analysis and Design Technique, McGraw-Hill, New York, 1988

[96] Tom DeMarco: Structured Analysis and System Specification Yourdon Press, N.Y., 1978, Prentice Hall, Englewood Cliffs, 1979

[97] Paul R. Ward, Stephen J. Mellor: Structured Development for Real-Time Systems, Prentice Hall, Englewood Cliffs, 1985

[98] T. Murata: Petri nets: Properties, analysis and applications, Proceedings of the IEEE, 77 (4): 541-580; 1989

[99] Arne Bilberg, Leo Alting: When Simulation Takes Control, Journal of Manufacturing Systems Volume 10 / No. 3

[100] Yourdon and Constantine: Structured Design, Prentice-Hall, 1979

[101] M. Jackson: Principles of Program Design, Academic Press, 1975

[102] M. Jackson: System Development, Prentice Hall International, 1983

[103] Grady Booch: Software Engineering With Ada, Benjamin-Cummings, 1987

[104] R. Ruhr: System Design With Ada, Software Series, Prentice-Hall, 1983

[105] Ian Simmonds: Aerospace Systems Software Engineering Environment, em Tool Integration, Environments and Frameworks, editado por D. Schefstrom e G. van den Broek, Wiley Professional Computing 1993

[106] J. Kramer, M. Kramer, A. Finkelstein: A Constructive Approach to the Design of Distributed Systems, Proc. of 10th IEEE ICDS, Paris, May 1990

[107] Ger van den Broek: Communication Systems Software Engineering Environment, Philips Research, em Tool Integration, Environments and Frameworks, editado por D. Schefstrom e G. van den Broek, Wiley Professional Computing 1993

[108] Son T. Voung: Guest editorial, Computer Networks and ISDN Systems 23 (1992), 323-324, North-Holland

[109] CCITT Recommendation Z.100: Specification and Description Language SDL, Blue Book, Volume X.1-X.5, 1988, ITU General Secretariat - Sales Section, Places des Nations, CH-1211 Geneva 20

[110] Grady Booch: Object-Oriented Analysis and Design with Applications, second edition, Benjamin-Cummings, CA, 1994

[111] F. Vernadat: Manufacturing Systems Modelling, Specification and Analysis, IFIP WG5.7 Working Conference on Evaluation of Production Management Methods, Gramado Brasil March 21-24, 1994, Elsevier

[112] R.A. Milner: A Calculus of Communication Systems, Lecture Notes in Computer Science 92, Springer Verlag, New York 1980

[113] C.A.R. Hoare: CSP :Communicating Sequencial Processes, Prentice Hall, Englewood Cliffs, NJ, 1985

[114] J. Goosenaerts: Enterprise Formolae and Information Infrastructures for Manufacturing (B-14), H. Yoshikawa and J. Goossenaerts (Editors), Elsevier Science B.V. (North-Holland) 1993 IFIP

[115] Artis SA: ARTIFEX-User-Guide, Artis SA Torino Italy, 1993

[116] J.T. Black: The Design of the Factory with a Future, McGraw-Hill 1991

[117] Bauer et. al: Shop Control Systems: From Design to Implementation, Chapman & Hall; ISBN 0-412-36040-3

[118] W. Wang, K. Popplewell, R. Bell: An Integrated multi-view system description approach to approximate factory modelling, Int'l Journal of Computer Integrated Manufacturing 1993, Vol. 6, Nr. 3, p. 165-174

[119] J. M. Mendonça, S. McCarthy, J. Schulte: Integration of Shop Floor Control Applications in a Real-Time Data Collection Environment, Proceedings of CIM-Europe Conference, Birmingham, May 27-29, 1992

[120] J.W.Schulte: Multi-purpose Modelling of Production Processes for Integrated Shop Floor Control, The 10th ISPE/IFAC International Conference on CAD/CAM, Robotics and Factories of the Future, August 21-24, 1994 Ottawa, Canada

[121] J.W. Schulte, B.-D. Becker: Production Schedulling using Genetic Alorithms, IPA-FhG, Stuttgart, Germany

[122] J. J. Pinto Ferreira, J. M. Mendonça: A Integração de Sistemas de Fabrico: Arquitecturas de Referência e Ferramentas de Projecto, 5.as Jornadas Nacionais de Projecto, Planeamento e Produção Assistidos por Computador - 30, 31 de Maio e 1 de Junho de 1995

[123] T. Bolognesi, E. Najm, P.A.J. Tilanus: G-LOTOS: a graphical language for concurrent systems, Computer Networks and ISDN Systems 26 (1994) 1101-1127, Elsevier

[124] José A. Mañas, Tomás de Miguel, Joaquín Salvachúa, Arturo Azcorra: Tools support to implement LOTOS formal specifications, Computer Networks and ISDN Systems 25 (1993) 815-839, North-Holland

[125] S. Budkowski, P. Dembinski: An Introduction to Estelle: A Specification Language for Distributed Systems, Computer Networks and ISDN Systems 14 (1987) 3-23, North-Holland

[126] Paul D. Amer, Darren New: Protocol visualization in Estelle, Computer Networks and ISDN Systems 25 (1993) 741-760, North-Holland

[127] S. Budkowski: Estelle development toolset (EDT), Computer Networks and ISDN Systems 25 (1992) 63-82, North-Holland

[128] Armando de Matos: Tópicos de Matemática Discreta, Faculdade de Ciências U.P., 1986

[129] Susan C. Murphy, Per Gunningberg, John P.J. Kelly: Experiences with Estelle, LOTOS, and SDL: a protocol implementation experiment, Computer Networks and ISDN Systems 25 (1991) 51-59, North-Holland

[130] Gerd Hoefner: Process Control Systems Software Engineering Environment, em Tool Integration, Environments and Frameworks, editado por D. Schefstrom e G. van den Broek, Wiley Professional Computing 1993

[131] H. Christensen, R. Gutiérrez: Modelling of Objects with ROMM and SDL'92, Proceedings of Sixth SDL Forum Darmstadt, Germany, 11-15 October 1993

[132] E. Inocêncio, M.M. Fonseca: SDL to C++ Translator for ISDN basic Rate Terminal Signalling, Proceedings of Sixth SDL Forum Darmstadt, Germany, 11-15 October 1993

[133] O. Faergemand, Anders Olsen: Introduction to SDL-92, TFL, Lyngso Allé 2, DK-2970 Horsholm, Denmark, July 27, 1992

[134] J. J. Pinto Ferreira, José M. Mendonça: Using a Decision Support System for CIM System Life Cycle Support, IFIP WG5.7 Working Conference on Evaluation of Production Management Methods, Gramado Brasil March 21-24, 1994, Elsevier

[135] J. J. Pinto Ferreira, José M. Mendonça: Using a Decision Support System for Fine Tunning an Information Technology Infrastructure in a CIM Environment", Proceedings of the Fourth International FAIM Conference, Virginia Polytechnic Institute and State University May 8-11, 1994

[136] GEODE, An Introduction to SDL and MSC, VERILOG SA, 150 rue Nicolas Vauquelin, BP 1310, 31106 TOULOUSE cedex, France

[137] J. J. Pinto Ferreira, José M. Mendonça: Supporting CIM Systems Life Cycle Through Material and Information Flow Executable Models, The 10th ISPE/IFAC International Conference on CAD/CAM, Robotics and Factories of the Future, August 21-24, 1994 Ottawa, Canada

[138] J. J. Pinto Ferreira, José M. Mendonça: Integrated Modelling Tools for Shop Floor CIM Systems, IMSE'94-European Workshop on Integrated Manufactuing Systems Engineering, Grenoble, France 12-14 December 1994

[139] J. J. Pinto Ferreira, J. M. Mendonça: Manufacturing and Information Systems Design Support through Integrated Modelling and Simulation, International Workshop on Concurrent/ Simultaneous Engineering Frameworks and Applications, 5 / 7 Abril 1995, Lisboa

[140] Ângelo Martins, J.J. Pinto Ferreira, J.M. Mendonça: Da Análise e Especificação Estruturadas ao Modelo Executável de um Sistema de Informação Fabril, PPP/AC 1995

[141] TeamWork / SA-RT User's Guide, Cadre, EUA, 1995

[142] B.D. Becker, J. Schulte: Specifications for the Model Editor, Shop Floor Modelling, Language, Scheduling and Dispatching Algorithms, ESPRIT Project 5478, Workpackage 5, February 10, 1992

[143] Gruia-Catalin, Kenneth C. Cox: Program Visualization: The Art of Mapping Programs to Pictures, Department of Computer Science, Washington University, Campus Box 1045, one Brookings Drive, St. Louis, MO 63130

[144] João A. Bastos, J.C. Marques dos Santos: Modelos de Simulação e Controlo para Células Flexíveis de Produção (FMS), 5.as Jornadas Nacionais de Projecto, Planeamento e Produção Assistidos por Computador - 30, 31 de Maio e 1 de Junho de 1995

[145] A. Lucas Soares, J. J. Pinto Ferreira, J. M. Mendonça: Organizational behaviour analysis and information technology fitness in manufacturing: Analysis through modelling and simulation" BASYS'95 - IEEE/ECLA/IFIP International Conference on Architectures and Design Methods for Balanced Automation Systems

[146] A. Lucas Soares, J. M. Mendonça: Interaction of Advanced Shop-Floor Management Systems with Production Systems Organization: An Exploratory Study, P.T. Kidd and W. Karwowski (Eds) , IOS Press

[147] A. Lucas Soares, J. M. Mendonça: Desenvolvimento de Sistemas Informáticos de Apoio à Produção: Uma perspective técnica, sociológica ou organizacional ?, VI Encontro de Sociologia Industrial, das Organizações e do Trabalho, Fundação Calouste Gulbenkian, Lisboa 29-30 de Novembro de 1994

[148] J. J. Pinto Ferreira, J. M. Mendonça: Modelling and Simulation of Manufacturing and Information Aspects of the Extended Enterprise, International Conference, Improving Manufacturing Performance in a Distributed Enterprise: Advanced Systems and Tools, Edinburgh Scotland, 13-14 July, 1995

[149] F. Vernadat: Future Directions fo CIM Deployment, European workshop on Integrated Manufacturing Systems Engineering, IMSE'94, 12-14 December, 1994, Grenoble

[150] Mary Mitchell: A Proposed Methodology for STEP Application Protocol Validation, U.S. Department of Coomerce, National Institute os Standards and Technology, NISTIR 4684

[151] Thomas H. Davenport: Saving it's Soul: Human-Centered Information Management, Harvard Business Review, Março-Abril de 1994.

[152] ISO CD 10303-1, 1992, Product Data Representation and Exchange - Part 1: Overview and Fundamental Principles.

[153] Object-Oriented Concepts, Databases, and Applications, Edited by Won Kim and Frederick H. Lochovsky, ACM Press Frontier Series 1989

[154] Joel D. Goldhar, Mariann Jelinek: Computer Integrated Flexible Manufacturing: Organizational, Economic, and Strategic Implications, Interfaces 15: 3 May-June 1985 (pp.94-105)

[155] Mary McKinly: CIM Made Simple! (Or At Least Simpler), An Approach to CIM Implementation, Proceedings of 8th CIM-Europe Annual Conference; 27-29 May 1992 Birmingham, UK

[156] Paul Ranky: Computer Networks for World Class CIM Systems, World Class Computer Integrated Manufacturing (CIM) Series, ISBN1-872631-01-0

[157] H.J. Warnecke: The Challenge for European Manufacturing Organizations, Computer Integrated Manufacturing; Proceedings of 8th CIM-Europe Annual Conference; 27-29 May 1992 Birmingham, UK

[158] J. J. Pinto Ferreira, J. Torres, C. Toscano, D. Dwyer, M. Loftus, S. McCarthy, J. M. Mendonça: Basic Application Software Package, ESPRIT Project 5478 SHOP-CONTROL, October, 1992.

[159] J Shivman: Future of manufacturing systems for SMEs: A strategic perspective, Proc, 5th CIM Europe Conf., May 17-19, 163-174

[160] Alfonso Fuggeta, Carlo Ghezzi, Dino Mandrioli, Angelo Morzenti: Executable Specifications with Data-flow Diagrams, Software-Practice and Experience, Vol 23(6), 629-653 (June 1993)

[161] C. Lischke, T. Kuhlmann, F. Lehmann, R. Cremer: Inter-Organizational Concurrent Engineering, Proceedings of 8th CIM-Europe Annual Conference; 27-29 May 1992 Birmingham, UK; pp167

[162] M. Matthiensen: Advanced Interorganisational Operations in Logistics Processes, Proceedings of 8th CIM-Europe Annual Conference; 27-29 May 1992 Birmingham, UK; pp 141

[163] Guide to Master Planning and Implementation for Enterprise Integration Programs Theodore J. Williams, (Ed) Technical report 157, Purdue Laboratory for Applied Industrial Control, Purdue University, West Lafayette, Indiana, June 1994

[164] G. Doumeingts, F. Marcotte, Y. Ducq: A Methodology to Evaluate the Best Industrial Practices in the Frame of the I.M.S. Programe: GLOBEMAN 21, IFIP Transactions on Production Management Methods Editors C. Walter, F.J.Kleiman, J.P.M. De Oliveira

[165] Jorgen P. Bansler, Keld Bodker: A reappraisal of structured analysis: Design in an Organizational Context, ACM transactions on Information Systems, Vol. 11, No. 2, April 1993

[166] Robert Orfali, Dan Harkey: Client/Server with Distributed Objects, Byte Abril 1995

[167] Tom DeMarco: Structured Analysis and System Specification, Yourdon Press, A Prentice-Hall Company, Englewood Cliffs, New Jersey

[168] Tutorial: Human Factors in Software Development, IEEE catalog EHO 185-9 Library of Congress No. 81-84180, IEEE Computer Society Catalog No. 390

[169] Brad A. Myers: User-Interface Tools: Introduction and Survey, IEEE - Software January 1989

[170] Aamir Matin, Stephen Evans: Using Information Flow Analisys to Reduce Lead Times, Flexible Automation and Integrated Manufacturing, FAIM'93 June 28-30, 1993 University of Limerick, Ireland, CRC Press Inc.

[171] Masoud Mansouri-Samani, Morris Sloman: Monitoring Distributed Systems (A Survey), Imperial College Research Report No. DOC92/93; 22 September 1992

[172] K. Marzullo, R. Cooper, M. Wood, K. Birman: Tools for Distributed Application Management, Cornell University TR 90-1136, June 1990

[173] J. J. Pinto Ferreira, J. M. Mendonça: Modelling For Monitoring and Simulation: Material and Information Flow, ERCIM Workshop, 26-28 May, 1993 Throndheim, Norway.

[174] Microsoft Excel User's Guide, Book 2

[175] Robert E. Young, Johan Vesterager: An approach to CIM system development whereby manufacuring people can design and build their own CIM systems, Int. J. Computer Inegrated Manufacturing, 1991, Vol 4, No.5, 288-299

[176] Udo Graefe, Vince Thomson: A reference model for production control, Int. J. Computer Inegrated Manufacturing, 1989, Vol 2, No.2, 86-93

[177] Steven Alter: Decision Support Systems, Current Practice and Continuing Challenges, Addison-Wesley Publishing Company, 1980