

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

**UMA PROPOSTA DE MODELAGEM DE INFORMAÇÕES
PARA INTEGRAÇÃO DA MANUFATURA E
ENGENHARIA CONCORRENTE**

DISSERTAÇÃO SUBMETIDA À
UNIVERSIDADE FEDERAL DE SANTA CATARINA
PARA A OBTENÇÃO DO GRAU DE MESTRE EM
ENGENHARIA MECÂNICA

Norton Paim Moreira

FLORIANÓPOLIS
SETEMBRO/1993

**UMA PROPOSTA DE MODELAGEM DE INFORMAÇÕES
PARA INTEGRAÇÃO DA MANUFATURA E
ENGENHARIA CONCORRENTE**

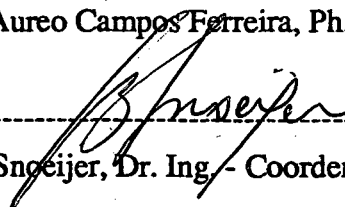
Norton Paim Moreira

**ESTA DISSERTAÇÃO FOI JULGADA PARA OBTENÇÃO DO TÍTULO DE
MESTRE EM ENGENHARIA**

**ESPECIALIDADE ENGENHARIA MECÂNICA, ÁREA DE CONCENTRAÇÃO EM
FABRICAÇÃO E APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-
GRADUAÇÃO EM ENGENHARIA MECÂNICA**

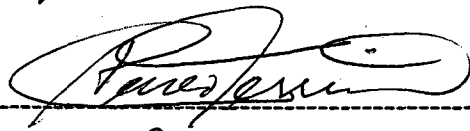


Prof. Aureo Campos Ferreira, Ph.D - Orientador




Prof. Berend Snoeijs, Dr. Ing. - Coordenador do Curso

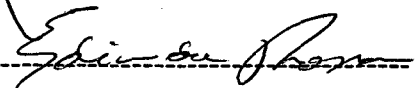
BANCA EXAMINADORA



Prof. Aureo Campos Ferreira Ph.D.- Presidente



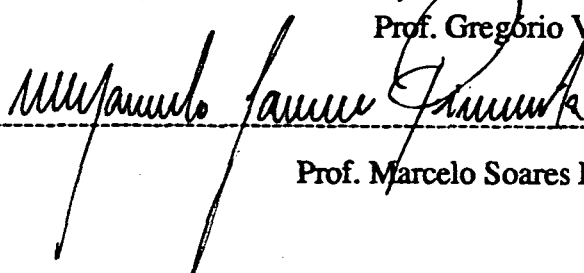
Prof. Abelardo Alvez de Queiroz, Ph.D.



Prof. Edison da Rosa, Ph.D.



Prof. Gregório Varvakis, Ph.D.



Prof. Marcelo Soares Pimenta, M.Sc.

Quem presume ter deixado, num livro, uma arte em caracteres escritos, ou quem a recebe, na suposição de que daqueles caracteres virá a sair algum conhecimento claro e duradouro, revela muita ingenuidade. [Sócrates]

Enfim, toda vez que você sentar-se em frente de uma máquina de escrever, postar-se diante do terminal ou pegar a caneta com o propósito de escrever, lembre-se que sentenças de breve extensão, amiúde logradas por intermédio da busca incessante da simplicidade no ato de redigir, da utilização freqüente do ponto, do corte de palavras inúteis que não servem mesmo para nada e da eliminação sem dó nem piedade dos clichês, dos jargões, da retórica discursiva e da redundância repetitiva — sem aquelas intermináveis orações intercaladas e sem o abuso das partículas de subordinação, como por exemplo, "que", "embora", "onde", "quando", capazes de encumpridá-las desnecessariamente, tirando em conseqüência o fôlego do pobre leitor —, isso para não falar que não custa refazê-las, providência que pode aproximar o verbo e o complemento do sujeito, tais sentenças de breve extensão, insistimos antes que comecemos a chateá-lo, são melhores e mais claras. Ou seja, use frases curtas. [Manual de Estilo da Editora Abril]

À minha companheira, Helena.

Agradecimentos

Desde a primeira vez que pensei em fazer mestrado, pesquisar, uma dissertação e coisa e tal, escrever esta página foi o momento mais esperado. Primeiro porque sem dúvida esta é a parte de um trabalho científico com maior "audiência", sendo portanto a porta de entrada para as outras tantas páginas. Segundo, porque aqui o autor é uma pessoa "definida", portanto pensa, mesmo que seja só por pensar. Por fim, porque nada neste mundo se faz sozinho, mesmo que digam por aí que as minhocas conseguem. Bom, na realidade, existe mais um motivo. Os agradecimentos, por norma, são escritos quando tudo, ou quase tudo, está pronto. Portanto, conseguimos!!!!

Dito isto, resta lembrar aqueles que fizeram com que esta dissertação acontecesse. Como já disse, desde o princípio pensei nos agradecimentos, logo existia uma "listinha" mental. Nela, muitos entraram, alguns saíram, mas com a confusão de resolver todos os problemas de última hora, acabei apagando para arrumar espaço. De qualquer forma, a lista estava muito grande para ser "inserida". O Winword não conseguiria processar. Na realidade, acho que todos os que fizeram parte deste trabalho são lembrados aqui, em cada linha, vírgula, figura e idéia. A todos, muchas gracias!

Ao pessoal do GRUCON e do GRANTE — muito obrigado por tantos "papos", idéias e críticas. Acho que estamos no caminho. Ao Manuel, muito obrigado pela didática historinha do "pinico", nas primeiras semanas de aulas de Métodos Matemáticos. Ao Prof. Aureo e Prof. Edison eu gostaria de agradecer pela orientação. Aos professores da banca, em especial ao Marcelo Pimenta e Gregório Varvakis, muito obrigado pelas sugestões que tornaram esta dissertação mais fácil de ser entendida.

Um agradecimento especial ao Joachim, da Universidade de Kaiserslautern, pela disposição em enfrentar o mato e os fósseis para ajudar na solução dos problemas com o Krisys.

Ao pessoal que ficou na torcida, umas palavrinhas — a meus pais, Seu Kleber e Dona Lila, com toda certeza não estaria aqui sem vocês. Ao meu irmãozinho Clayton e cunhadinha Aninha, acabou, viu! Ao pessoalzinho da Lagoa: Rê, Jamir, Angélica, Návala, Paulinho, Maria Grasselina, Zica, Hilário, Lenir, Íris e Tchai, hummm...sei lá, obrigado por tudo.

Por fim, à Helena, companheira de todas as horas e alegrias. Que venga la proxima!

Sumário

CAPÍTULO 1 INTRODUÇÃO.....	1
1.1 MOTIVAÇÃO E OBJETIVOS.....	1
1.2 ESTRUTURA DO TRABALHO.....	3
CAPÍTULO 2 SISTEMA DE MANUFATURA: DEFINIÇÃO E DESENVOLVIMENTOS....	4
2.1 SISTEMA DE MANUFATURA.....	4
2.1.1 <i>Definição do Sistema de Manufatura</i>	5
2.1.2 <i>Caracterização do Sistema de Manufatura</i>	7
2.1.3 <i>A Informação no Sistema de Manufatura</i>	7
2.2 ENGENHARIA CONCORRENTE	10
2.2.1 <i>Definição</i>	10
2.2.2 <i>Engenharia Concorrente ou Simultânea?</i>	12
2.2.3 <i>Estrutura básica para implantação</i>	12
2.2.4 <i>Sistema de manufatura com Engenharia Concorrente</i>	14
2.3 MANUFATURA INTEGRADA POR COMPUTADOR - CIM	14
2.3.1 <i>Definições e desenvolvimentos em CIM</i>	14
2.3.2 <i>Implantação de CIM</i>	15
2.3.3 <i>Tecnologias associadas</i>	16
2.3.4 <i>Modelos para CIM</i>	17
CAPÍTULO 3 FERRAMENTAS PARA A INTEGRAÇÃO DO SISTEMA DE MANUFATURA	21
3.1 TECNOLOGIA DE "FEATURES"	21
3.1.1 <i>Definição de "Features"</i>	22
3.1.2 <i>Identificação, Classificação e Representação de "Features"</i>	22
3.1.3 <i>Modelos para uso de "features"</i>	24
3.1.4 <i>Múltiplas "features" X "features" únicas</i>	27
3.1.5 <i>Aplicações de Tecnologia de "Features"</i>	27
3.2 CONCEITOS BÁSICOS DE ORIENTAÇÃO A OBJETOS	28
3.2.1 <i>Definição e implementação</i>	29
3.2.2 <i>Planejamento e Programação do Sistema de Manufatura Orientado a Objetos</i>	30
3.3 SISTEMAS ABERTOS, DISTRIBUÍDOS E HETEROGÊNEOS	30
3.3.1 <i>Definição e caracterização</i>	31
3.3.2 <i>Sistemas de Banco de Dados Heterogêneos</i>	32
3.3.3 <i>O Sistema de Manufatura neste contexto</i>	33
3.4 SISTEMAS GERENCIADORES DE BASE DE CONHECIMENTO	33
3.4.1 <i>O sistema Krisys</i>	34
CAPÍTULO 4 PROPOSTA DE AMBIENTE DE INTEGRAÇÃO.....	38
4.1 ANÁLISE DO PROBLEMA.....	38
4.2 AVEC — AMBIENTE ABERTO VISANDO ENGENHARIA CONCORRENTE.....	39
4.3 O CONTROLE DAS INFORMAÇÕES — GETME	42
4.3.1 <i>Requisitos para um protocolo</i>	42
4.3.2 <i>Serviços de mensagens</i>	43
4.4 INFORMAÇÕES REDUNDANTES — "OPAMBAE"	44
4.5 "INTERFACE" COM O USUÁRIO.....	45
4.6 MODELO DE FERRAMENTAS — CAX.....	46
4.7 PLATAFORMA PARA A IMPLEMENTAÇÃO.....	46
CAPÍTULO 5 MODELAGEM DA BASE DE INFORMAÇÕES.....	47
5.1 MODELAGEM DAS INFORMAÇÕES NO SISTEMA DE MANUFATURA	47
5.2 INFORMAÇÕES FUNCIONAIS E TECNOLÓGICAS	48

5.2.1 Grupos	49
5.2.2 "Yet another feature definition"	51
5.2.3 Identificação e classificação das "features"	53
5.2.4 Elementos agregados	57
5.2.5 Normas	61
5.2.6 Cálculos	62
5.3 INFORMAÇÕES DE FACILIDADES	63
5.4 INFORMAÇÕES DE ENLACE	65
5.4.1 O modelo do produto	66
5.4.2 Sistema de referência	69
5.4.3 O modelo de empresa	71
5.4.4 O modelo de aplicações	71
5.5 CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO	72
5.5.1 Por que o sistema Krisys?	72
5.5.2 Implementação da modelagem das informações	73
5.5.3 Exemplo de modelagem	76
CAPÍTULO 6 KRISYS: RESULTADOS E SUGESTÕES	79
6.1 EXPERIÊNCIA DE MODELAGEM NO KRISYS	79
6.2 KRISYS: SUGESTÕES	80
CAPÍTULO 7 ANÁLISE DE RESULTADOS: CONCLUSÕES	82
7.1 OBJETIVO PROPOSTO	82
7.2 RESULTADOS OBTIDOS	82
7.3 ANÁLISE DOS RESULTADOS	83
7.4 Sugestões de trabalhos futuros	83
7.5 CONCLUSÕES	84
CAPÍTULO 8 REFERÊNCIAS BIBLIOGRÁFICAS	85
ANEXO 1 REPRESENTAÇÃO DOS OBJETOS	94
1 OBJETO OPAMBAE	94
2 INFORMAÇÕES DE FACILIDADES	95
2.1 Ferramentas	95
3 INFORMAÇÕES DE ENLACE	97
3.2 Sistemas	97
3.2 Peças	98
3.3 Sistema de referências	99
4 INFORMAÇÕES FUNCIONAIS-TECNOLÓGICAS	101
4.1 Normas	101
4.2 Grupos	102
4.3 "Features"	102
4.4 Elementos Agregados	109
4.5 Cálculos	112
ANEXO 2 GRÁFICOS DE OBJETOS	114
ANEXO 3 GERAÇÃO DO OBJETO "POLIA-ESTICADORA"	111

Índice de Figuras

FIGURA 2.1 FUNÇÕES DO SISTEMA DE MANUFATURA.....	5
FIGURA 2.2 FUNÇÕES DA FABRICAÇÃO DE PRODUTO	6
FIGURA 2.3 COMPARAÇÃO ENTRE OS MERCADOS DOS ANOS 70 E 80.....	7
FIGURA 2.4 ESQUEMA DE REPRESENTAÇÃO DE INFORMAÇÕES	8
FIGURA 2.5 FLUXO DE INFORMAÇÕES NO SISTEMA DE MANUFATURA [HAR84].....	8
FIGURA 2.6 INFORMAÇÕES NA FABRICAÇÃO DE PRODUTO [HAR84].....	9
FIGURA 2.7 CONCEITO DE ENGENHARIA CONCORRENTE.....	11
FIGURA 2.8 FLUXO DE INFORMAÇÕES COM ENGENHARIA CONCORRENTE.....	14
FIGURA 2.9 EXEMPLO DE SI INTEGRANDO ATIVIDADES HETEROGÊNEAS.....	18
FIGURA 2.10 INTEGRAÇÃO POR "FEATURES"	19
FIGURA 3.1 RELAÇÃO ENTRE "FEATURES" CAD E CAPP	24
FIGURA 3.2 RECONHECIMENTO DE "FEATURES" [COR91].....	25
FIGURA 3.3 PROJETO POR "FEATURES" [COR91].....	26
FIGURA 3.4 ESTRUTURA PARA PROJETO POR "FEATURES" [CUN88].....	27
FIGURA 3.5 OS COMPONENTES DE UM HDDBS.....	32
FIGURA 3.6 ARQUITETURA DE KRISYS	35
FIGURA 3.7 EXEMPLO DE MODELAGEM DE CONHECIMENTO.....	35
FIGURA 4.1 AVEC — AMBIENTE ABERTO VISANDO ENGENHARIA CONCORRENTE	41
FIGURA 4.2 MODELO DE IMPLEMENTAÇÃO DO GETME	43
FIGURA 4.3 EXEMPLO DE INTEGRAÇÃO DE FERRAMENTAS	44
FIGURA 4.4 ATUALIZAÇÃO DE INFORMAÇÃO REDUNDANTE	45
FIGURA 5.1 CLASSES DE INFORMAÇÃO	48
FIGURA 5.2 OBJETOS QUE REPRESENTAM AS INFORMAÇÕES.....	48
FIGURA 5.3 EIXO PARA POLIA ESTICADORA	49
FIGURA 5.4 CLASSES DE GRUPOS	49
FIGURA 5.5 PRIMITIVA DE UM GRUPO DE CONSTRUÇÃO	50
FIGURA 5.6 RELAÇÃO ENTRE "FEATURES" EM PROJETO E PRODUÇÃO	52
FIGURA 5.7 UMA "FEATURE" VISTA SOB VÁRIOS ASPECTOS.....	52
FIGURA 5.8 "FEATURE" DECOMPOSTA EM SUPERFÍCIES	53
FIGURA 5.9 ESPECIALIZAÇÃO DE OBJETO	54
FIGURA 5.10 CLASSIFICAÇÃO DAS "FEATURES".....	54
FIGURA 5.11 "FEATURES" LAMINARES.....	55
FIGURA 5.12 CLASSIFICAÇÃO DAS "FEATURES" QUASI-SIMÉTRICAS.....	55
FIGURA 5.13 "FEATURE" RANHURA EXTERNA.....	56
FIGURA 5.14 CONSTRUÇÃO DE UMA "FEATURE" FUNCIONAL	56
FIGURA 5.15 CLASSIFICAÇÃO DE "FEATURES" DETALHE EXTERNO REGULAR	57
FIGURA 5.16 CLASSIFICAÇÃO DAS SUPERFÍCIES.	58
FIGURA 5.17 RELAÇÃO "FEATURE"/SUPERFÍCIE	59
FIGURA 5.18 SUPERFÍCIES NA "FEATURE" RANHURA INTERNA	59
FIGURA 5.19 CLASSIFICAÇÃO DE TOLERÂNCIAS	60
FIGURA 5.20 AGREGAÇÃO DE TOLERÂNCIA DIMENSIONAL.....	60
FIGURA 5.21 AGREGAÇÃO DA TOLERÂNCIA DE FORMA	60
FIGURA 5.22 TOLERÂNCIAS DE RESTRIÇÃO.....	61
FIGURA 5.23 MODELAGEM DAS NORMAS.....	61
FIGURA 5.24 MODELO DE NORMA PARA ROLAMENTOS	62
FIGURA 5.25 NORMALIZAÇÃO DE UMA INSTÂNCIA DE ROLAMENTO.....	62
FIGURA 5.26 OBJETOS CÁLCULO.....	63
FIGURA 5.27 UTILIZAÇÃO DOS ATRIBUTOS DE CÁLCULO.....	63
FIGURA 5.28 CLASSIFICAÇÃO DE MÁQUINAS-FERRAMENTAS.....	64
FIGURA 5.29 MODELAGEM DAS FERRAMENTAS.....	64
FIGURA 5.30 MODELAGEM DOS SISTEMAS DE FIXAÇÃO	65
FIGURA 5.31 MODELAGEM DOS DISPOSITIVOS DE FIXAÇÃO.....	65
FIGURA 5.32 MODELAGEM DAS INFORMAÇÕES DE ENLACE.....	66

FIGURA 5.33 HIERARQUIA DE PRODUTO	67
FIGURA 5.34 CONSTRUÇÃO DE UM PRODUTO	68
FIGURA 5.35 AGREGAÇÃO DE SISTEMA DE COORDENADAS	68
FIGURA 5.36 SISTEMAS DE COORDENADAS LOCAIS	69
FIGURA 5.37 PONTO DE REFERÊNCIA PARA PEÇAS DE REVOLUÇÃO.....	69
FIGURA 5.38 PONTO DE REFERÊNCIA PARA "FEATURES" IRREGULARES.....	69
FIGURA 5.39 MATRIZ DE TRANSFORMAÇÃO PARA OS ELEMENTOS CONSTRUTIVOS	70
FIGURA 5.40 REPRESENTAÇÃO DA LOCALIZAÇÃO DE SISTEMAS E PEÇAS	70
FIGURA 5.41 REUTILIZAÇÃO DE PEÇAS	70
FIGURA 5.42 CLASSIFICAÇÃO DAS OPERAÇÕES DE TORNEAMENTO	72
FIGURA 5.43 RELAÇÃO ENTRE "FEATURES" E OPERAÇÕES DE FABRICAÇÃO.....	72
FIGURA 5.44 GERAÇÃO ENCADEADA DE OBJETOS	74
FIGURA 5.45 ATRIBUTOS DO OBJETO GRUPO	75
FIGURA 5.46 MÉTODO PARA CONICIDADE DO FURO.....	76
FIGURA 5.47 POLIA ESTICADORA	76
FIGURA 5.48 OBJETOS QUE REPRESENTAM A POLIA ESTICADORA	77
FIGURA 5.49 ESTRUTURA DA "FEATURE" CHANFRO MENOR EXTERNO	77
FIGURA 5.50 ATRIBUTOS DA MATRIZ DE TRANSFORMAÇÃO	78
FIGURA 5.51 REPRESENTAÇÃO DE TOLERÂNCIA DE PERPENDICULARIDADE.....	78
FIGURA 5.52 ATRIBUTOS DO OBJETO PERPENDICULARIDADE	78
FIGURA 6.1 PROBLEMA NO USO DE AGREGAÇÃO	80
FIGURA 1 RELAÇÕES DE ABSTRAÇÃO NO KRISYS	114

Glossário

Sigla	Inglês	Português
AI	Artificial Intelligence	Inteligência Artificial
CAE	Computer Aided Engineering	Análise de Engenharia Auxiliada por Computador
CAI	Computer Aided Inspection	Inspeção Auxiliada por Computador;
CAM-I	Computer Aided Manufacturing International	Manufatura Auxiliada por Computador Internacional
CAP	Computer Aided Production	Produção Auxiliada por Computador
CAPP	Computer Aided Process Planning	Planejamento de Processos Auxiliado por Computador
CAQ	Computer Aided Quality	Qualidade Auxiliada por Computador.
CAT	Computer Aided Test	Teste Auxiliado por Computador;
CE	Concurrent Engineering	Engenharia Concorrente
CIM	Computer Integrated Manufacturing	Manufatura Integrada por Computador
CNC	Computered Numerical Control	Comando Numérico Computadorizado
DDBMS	Distributed Data Base Management System	Sistema Gerenciador de Base de Dados Distribuída
DFA	Design for Assembly	Projeto para a Montagem
DFM	Design for Manufacturing	Projeto para a Manufatura
ES	Expert System	Sistema Especialista
GT	Group Technology	Tecnologia de Grupo
HDBMS	Heterogeneous Data Base Management System	Sistema Gerenciador de Bases de Dados Heterogêneas
ICAM	Air Force Integrated Computer Aided Manufacturing Project	Projeto Manufatura Integrada e Auxiliada por Computador da Força Aérea
IDEF	ICAM Definition Flow	Definição de Fluxo de Informação - ICAM
IGES	Initial Graphical Exchange Specification	Especificação para Intercâmbio de Arquivos Gráficos
KBMS	Knowledge Base Management System	Sistema Gerenciador de Bases de Conhecimento
KRISYS	Knowledge Representation and Inference System	Sistema de Inferência e Representação do Conhecimento
MAP	Manufacturing Application Protocol	Protocolo de Aplicação para a Manufatura
MMS	Manufacturing Message	Especificação de Mensagens de Manufatura
NC	Numerical Control	Comando Numérico
OODBMS	Object Oriented DBMS	DBMS Orientado a objetos
PDES	Product Data Exchange Specification	Especificação de Troca de Definição de Produto
PDM	Product Data Modeling	Modelagem de Dados do Produto
RPC	Remote Procedure Call	Chamada Remota de Procedimentos
SI	Information System	Sistema de Informação
STEP	Standard for the Exchange of Product Model Data	Padrão para Troca de Modelo de Dados de Produto

Resumo

A presente dissertação tem como objetivo estudar o fluxo de informações no Sistema de Manufatura, de forma a propor uma abordagem computacional que auxilie na implementação de Engenharia Concorrente. Para tanto são analisados conceitos e ferramentas de modelagem. O resultado é a definição de uma plataforma computacional para o auxílio à integração da manufatura através do compartilhamento das informações.

Como proposta é descrito o Ambiente Aberto Visando Engenharia Concorrente - AVEC - que utiliza conceitos de Banco de Dados Distribuído Heterogêneos, Orientação a Objetos e Tecnologia de "Features".

A partir desta especificação é implementado o modelo de uma base de conhecimentos para a representação das informações do Sistema de Manufatura, sendo enfatizada a modelagem do produto durante seu ciclo de vida.

O resultado obtido é uma especificação computacional que contribui para o direcionamento de futuros desenvolvimentos na área de Engenharia Concorrente e integração da manufatura.

Abstract

The dissertation main goal is to study the Manufacturing System information flow, to propose a computational approach to aid in the Concurrent Engineering implementation. In this way, concepts and modeling tools have been analyzed. The result is a computational environment definition to improve the manufacturing integration through the information sharing.

As a propose is described the Open System Aiming Concurrent Engineering — AVEC—. This environment uses Feature Technology, Object-Oriented and Heterogeneous Data Base concepts.

Using this specification, a knowledge base to the Manufacturing System information representation was done. The emphasis is in the product life cycle information representation.

The results are a computational specification that contributes to lead future developments in the Concurrent Engineering and manufacturing integration area

Capítulo 1

INTRODUÇÃO

Em primeiro lugar a atenção se aplica ao movimento e só depois àquilo que se move. [Eisenstein]

Este capítulo introduz o assunto desta dissertação, situando-o no cenário nacional e internacional. São descritos os objetivos e o método utilizado. Por fim, é apresentada a estrutura do texto.

1.1 Motivação e objetivos

Uma análise da conjuntura econômica atual demonstra que o mercado mundial tende a se tornar cada dia mais competitivo, de forma que prazo de entrega, qualidade e custo são os fatores que determinam a sobrevivência de uma empresa. Segundo [REI92], as empresas estão realizando grandes mudanças na sua forma de trabalhar, utilizando novas técnicas de produção para manter uma postura competitiva no mercado.

A indústria mecânica nacional tende a se adaptar a essas inovações, pois, de outra forma, perderá ainda mais mercado para concorrentes internacionais. Mesmo que poucas empresas invistam em pesquisa e desenvolvimento, em geral estão atentas para novas tecnologias que permitam aumentar a produtividade e a qualidade de seus produtos. Uma pesquisa sobre o mercado nacional de máquinas CNC [LUC92] demonstra a disposição da indústria para a utilização desta tecnologia. Segundo a pesquisa, realizada com produtores nacionais de CNCs, foram compradas muitas máquinas durante o Plano Cruzado (1986), após o que, devido a crises renitentes na economia, o mercado sofreu forte retração. Segundo os autores da pesquisa, os usuários já ultrapassaram a fase de adaptação à tecnologia e estarão aptos a investir novamente no setor quando o caos econômico terminar. Esta opinião é reforçada por [FUO91], com relação as ferramentarias brasileiras. Isto implica que a indústria nacional, a nível gerencial, já se encontra preparada para assimilar novos conceitos de produção.

Segundo o que se observa em trabalhos de cooperação entre Universidade e empresa, nas poucas, e honrosas, tentativas de automatização realizadas na indústria brasileira até hoje, não houve maiores preocupações com o fluxo de informação como um todo. Com investimentos em máquinas CNCs e sistemas CAD de forma isolada, consegue-se apenas criar "ilhas de automação". A consequência direta desta abordagem é a descontinuidade e perda de informações causada pela incompatibilidade de padrões entre sistemas de diferentes fabricantes. Além disto, a ineficiência no tratamento da informação (como, por exemplo, sua inconsistência ou subutilização) praticamente inviabiliza qualquer tipo de integração. O uso destas tecnologias deve ser encarado como uma pequena parte das modificações necessárias no Sistema de Manufatura visando acréscimo de produtividade.

Na literatura, existem diversas abordagens para a integração das informações. Alguns propõem que o planejamento de processos assuma o papel de elemento integrador [CHA84, ROZ90], uma vez que por aí passam as informações do projeto para a manufatura. Um outro enfoque atesta que a integração deve ser

feita por padrões de transferência de informações entre sistemas CAD, como o IGES (Initial Graphics Exchange Specification) [SSE90a]. Recentemente, uma série de trabalhos têm utilizado Tecnologia de "Features"¹ como elemento integrador [YEO91, GRA88, SSE90b]. Estudos realizados no âmbito do projeto Esprit [CAM91] propõem a utilização PDES/STEP (Product Definition Exchange Specification/Standard for Exchange of Product Model Data) e conceitos de Sistemas de Informação.

Alguns produtos comerciais integram máquinas e sistemas de computador, criando uma estrutura apoiada em uma base de dados centralizada. Estes sistemas utilizam protocolos de comunicação proprietários, de forma que não existe possibilidade de troca de informações com sistemas de outros fornecedores. Estas características resultam em sérias dificuldades para interligar diversas áreas da empresa.

A passagem de um ambiente de manufatura tradicional para a utilização de novas tecnologias de produção é um tema bastante controverso, mas uma série de filosofias de gerenciamento do Sistema de Manufatura, como por exemplo Engenharia Concorrente, Manufatura Integrada por Computador (CIM), Gerenciamento da Qualidade Total e Distribuição da Função Qualidade, foram definidas neste sentido. Embora, conceitualmente, sejam boas soluções para melhorar produtividade, qualidade e custo, estas filosofias carecem de tecnologias que as implementem de fato. Isto é, não existe um produto, uma "solução de prateleira", diretamente aplicável, que resolva problemas reais da indústria em geral. Para cada caso uma série de adaptações e simplificações deve ser realizada, comprometendo os resultados.

O objetivo principal desta dissertação é compilar informações e propor uma abordagem inicial para o problema de integrar o Sistema de Manufatura a partir do ponto-de-vista de Engenharia Concorrente. A partir desta primeira tentativa, muitos outros trabalhos, aprofundando pontos específicos, devem ser desenvolvidos.

O princípio que norteia este estudo é que a complexidade do Sistema de Manufatura deve ser encarada do ponto-de-vista organizacional, trabalhando o gerenciamento do fluxo de informação. Sob este aspecto, conceitos de CIM são usados para criar um ambiente favorável à distribuição e controle das informações geradas. A partir desta plataforma são fornecidos mecanismos que facilitam a implementação de Engenharia Concorrente. No sentido de definir a plataforma são identificadas as diversas tarefas do ciclo de vida de um produto e criado um modelo que permita que estas sejam agregadas. É, então, apresentada uma proposta de plataforma em um ambiente distribuído, heterogêneo, utilizando conceitos de orientação a objetos e sistemas especialistas. Na plataforma AVEC, cada tarefa do Sistema de Manufatura tem suas informações modeladas e interage com as demais, trocando informações.

Neste sentido, as tarefas são modeladas como Sistemas Especialistas (ES) que cooperam entre si para a solução dos problemas. A representação do produto é a ligação entre as informações das diversas tarefas do Sistema de Manufatura (SM), sendo portanto o elemento principal de modelagem. Nesta abordagem, foi utilizada a Tecnologia de "Features" que permitiu uma linguagem comum entre as diferentes tarefas do ciclo de vida do produto.

A implementação do AVEC como um todo é uma tarefa longa, fruto de muito trabalho. No sentido de iniciar a implementação foi estruturada uma base de informações que descrevesse o produto e suas informações acessórias. Uma vez que as tarefas do ciclo de vida do produto são modeladas como ESs foi utilizado um KBMS na implementação da informação.

¹Neste trabalho, uma série de palavras são utilizadas sem tradução, pois, entende-se que sejam mais representativas na sua forma original. Estas palavras estarão sempre entre aspas.

1.2 Estrutura do trabalho

Antes da descrição da proposta é feita uma revisão dos conceitos que lhe servem de base (capítulos 2 e 3). A proposta é apresentada em duas partes. Primeiro é descrita a plataforma a partir de uma visão macroscópica do relacionamento entre as diversas ferramentas que atuam em conjunto (capítulo 4). Em seguida é descrita a modelagem das informações, assim como sua implementação em um Sistema Gerenciador de Base de Conhecimento (capítulo 5). São tecidas considerações sobre o sistema utilizado na modelagem da base de informações, sugerindo pontos onde este pode ter seu poder de modelagem expandido (capítulo 6). Por fim, são apresentados os resultados e considerações sobre trabalhos futuros (capítulo 7).

Capítulo 2

SISTEMA DE MANUFATURA: DEFINIÇÃO E DESENVOLVIMENTOS

Todo corpo permanece em estado de repouso ou de movimento uniforme em linha reta, a menos que seja obrigado a mudar seu estado por forças impressas nele. [Newton]

Novos desenvolvimentos tecnológicos fazem com que a manufatura se modifique a cada dia. Constantemente novas metodologias são desenvolvidas e aperfeiçoadas visando sempre melhorar qualidade e produtividade. Nesta batalha incessante, os conceitos tradicionais vão se tornando falhos. Hoje em dia, existem diversos enfoques para uma mesma atividade, tornando-se muito difícil adaptar "velhos" conceitos a novas realidades. Na literatura, não existe um consenso, nem mesmo parcial, a este respeito, logo, tenta-se, neste capítulo, adaptar formas e nomes visando, mais do que definir, situar o trabalho e o leitor.

O item 2.1 conceitua manufatura e Sistema de Manufatura, caracterizando seu estado atual e o fluxo de informações. Em seguida são analisados os conceitos de Engenharia Concorrente (seção 2.2) e Manufatura Integrada por Computador (seção 2.3).

2.1 Sistema de Manufatura

Qual a primeira idéia que surge relacionada com a manufatura? Produção de bens, centenas de máquinas funcionando sem parar? Linhas de montagem no melhor estilo Charles Chaplin, ou talvez uma fábrica de automóveis com robôs, máquinas especiais e alta produtividade? Possivelmente estas seriam as respostas para a maioria das pessoas. No entanto, a definição de manufatura é muito mais abrangente.

Na literatura, a definição começa pela transformação de matéria-prima em produto final [HIT79] [HAR84]. Outros englobam aí o tratamento das informações e serviços [GAR84]. Então, segundo Gardiner, "manufatura é a atividade pela qual materiais e informações são transformados, em bens ou serviços, para satisfazer as necessidades humanas." Definições como esta são genéricas o suficiente para englobar tanto uma indústria aeroespacial como um escritório de contabilidade.

Uma vez que não seria possível modelar tamanha generalidade, decidiu-se por restringir o trabalho a um tipo de indústria de manufatura específica: metal-mecânica. A seguir, será definido e caracterizado o Sistema de Manufatura.

2.1.1 Definição do Sistema de Manufatura

Um produto é desenvolvido seguindo uma seqüência de atividades agrupadas em seu "ciclo de vida" [REI92]. Estas iniciam na análise do mercado, passam pela aquisição da matéria-prima e se estendem através de toda gama de atividades de projeto, planejamento de processos e de produção, execução e controle dos processos de manufatura, distribuição e manutenção do produto final. Para que este ciclo se desenvolva de forma coordenada, o Sistema² de Manufatura é definido conforme o diagrama da figura 2.1 [HAR84].

O diagrama da figura 2.1 apresenta um modelo do Sistema de Manufatura. Segundo o autor, o Sistema de Manufatura se desenvolve em função de duas atividades principais: a fabricação e o "marketing"³. Além destes dois grupos principais existem as funções relativas ao gerenciamento da instituição e de suporte.



Figura 2.1 Funções do Sistema de Manufatura

As funções do Gerenciamento da Manufatura dizem respeito ao planejamento dos objetivos corporativos. Definem os produtos e serviços oferecidos pela empresa, além de controlar sua execução. As decisões tomadas a este nível são orientativas, e serão usadas para balisar as atividades desenvolvidas pelas demais funções.

A Fabricação de Produto engloba as funções de projeto, produção e suporte ao produto final. Pode-se dizer que estas são as funções básicas do Sistema de Manufatura, mas se não houver uma forte sintonia com a situação do mercado, o resultado final não será satisfatório. Por este motivo, deve existir uma forte integração desta com o "Marketing" de Produto.

As funções do "Marketing" de Produto são de venda, serviço e publicidade, ou mais especificamente:

- a) análise do fluxo de produtos no mercado, informando seu potencial para os produtos da empresa;
- b) venda dos produtos fabricados;
- c) distribuição e suporte aos produtos;

²Sistema, segundo Aurélio Buarque de Holanda Ferreira é definido como: "disposição das partes ou dos elementos de um todo, coordenados entre si e que funcionam como estrutura organizada", ou "conjunto de elementos, materiais ou idéias, entre os quais se possa definir alguma relação".

³"Marketing", segundo Aurélio Buarque de Holanda Ferreira é definido como: "conjunto de estudos e medidas que provêm estrategicamente o lançamento e a sustentação de um produto ou serviço no mercado consumidor, garantindo o bom êxito comercial da iniciativa." Baseado nesta definição, as funções de "Marketing do Produto" desenvolvem-se em torno de análise do mercado, publicidade e vendas.

- d) avaliação da realimentação do mercado para os produtos em fabricação;
- e) publicidade.

O Suporte da Manufatura realiza serviços internos a empresa, suprindo as necessidades de apoio jurídico, processamento de dados, administração de pessoal, etc.

A Fabricação de Produto, figura 2.2, inicia a partir da especificação do produto, orientada pela situação do mercado. Com estas informações é realizada a função de Processos de Engenharia, gerando uma primeira concepção do produto. A concepção do produto passa por um processo de síntese, análise e otimização, de forma a gerar o projeto do produto e suas peças.

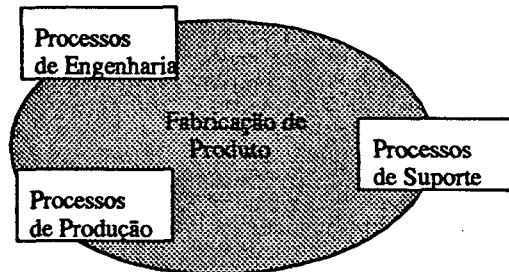


Figura 2.2 Funções da Fabricação de Produto

A função de Processos de Produção, basicamente, realiza as tarefas de planejamento, execução e controle das atividades que irão produzir as peças. As tarefas de planejamento são realizadas em duas etapas:

- a) planejamento de processos: consiste na preparação de um conjunto de instruções que descrevem como usinar uma peça de forma que satisfaça especificações do Processos de Engenharia. Pode incluir:
 - seqüência de operações
 - máquinas
 - ferramentas
 - material
 - tolerâncias
 - parâmetros de usinagem
 - acabamento superficial
 - tratamento térmico
 - dispositivos de fixação
 - superfícies de fixação
 - tempo padrão
 - critérios de inspeção
 - representação gráfica da peça durante a produção;
- b) planejamento da produção: atividade que distribui os devidos recursos de produção (incluindo mão-de-obra) para a execução das tarefas requisitadas, durante um determinado período de tempo. Esta distribuição deverá ser baseada em objetivos específicos, respeitando as restrições tecnológicas, de forma que a seqüência gerada seja ótima do ponto de vista dos critérios de avaliação e desempenho [RAB91].

A execução das atividades, definidas na etapa de planejamento, é realizada com os recursos requeridos resultando na produção das peças definidas no projeto. Estas atividades são realizadas em centros de trabalho, ou seja, agrupamentos de equipamentos no chão de fábrica onde são realizadas tarefas específicas.

Durante a produção das peças, diversas atividades de controle são realizadas para verificar o andamento da produção através dos diversos centros de trabalho. Paralelamente a estas atividades são realizadas inspeções visando garantir os requisitos de qualidade.

As funções de Processos de Suporte são várias, desde a embalagem do produto até seu armazenamento, preparação, instalação, operação e reparo, servindo de suporte tanto para a Processos de Produção como para o usuário final. Estas atividades estão integradas à Fabricação de Produto pois realimentam os Processos de Engenharia e Processos de Produção com reivindicações provenientes dos usuários.

2.1.2 Caracterização do Sistema de Manufatura

O carro do ano, o refrigerador novo, o ar-condicionado silencioso. O mercado atual é um voraz consumidor de novos produtos, exigindo que a indústria reaja rapidamente. Opitz, em 1971, já constatava a diminuição do tempo de renovação dos produtos [OPI71]. A figura 2.3 ilustra graficamente o comportamento dos produtos no mercado durante as décadas de 70 e 80. Nos anos 70, o tempo de permanência de um produto no mercado era relativamente mais longo, pois os consumidores não exigiam constantes renovações. Já nos anos 80 a redução do tempo, reflexo de imposição do mercado, foi assumida pelos fabricantes que passaram a adotar uma política de obsolescência embutida nos produtos, lançando novas versões (e.g. A', A'', B', B'', etc.) do produto com grande frequência. Uma implicação direta da diversificação da produção industrial é a redução no tamanho de lotes de peças produzidas [LOR93]. Desta forma, a indústria deve estar preparada para incrementar a produtividade de forma a acompanhar as necessidades do mercado.

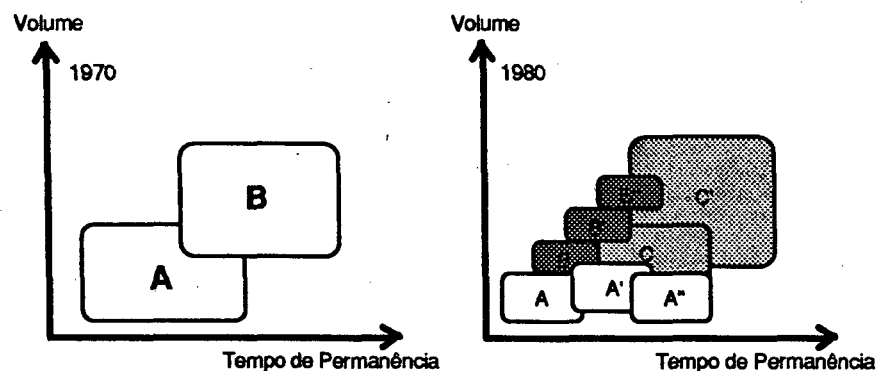


Figura 2.3 Comparação entre os mercados dos anos 70 e 80

2.1.3 A Informação no Sistema de Manufatura

Informação pode ser entendida como um dado acrescido de seu significado. Ou seja, recebido o dado, este deve ser interpretado utilizando uma série de conhecimentos intrínsecos. Sem este conhecimento, a informação não existe. Logo, informação diz respeito a dados acrescidos de conhecimento. O Sistema de Manufatura gera uma quantidade muito grande de informações, sendo fundamental que estas sejam representadas de forma a facilitar sua distribuição para as funções onde sejam necessárias.

No item 2.1.1, foram apresentadas as funções desenvolvidas no Sistema de Manufatura. Cada uma delas, durante seu desenvolvimento, gera uma série de informações que, além de utilizadas intrinsecamente, são subsídios para outras funções. Para representar as informações trocadas entre as funções, utilizou-se o modelo desenvolvido pelo projeto ICAM (Air Force Integrated Computer Aided Manufacturing Project) conhecido por IDEF₀ (ICAM Definition Flow) [HAR84]. Sua representação é bastante simples e intuitiva. Os blocos representam funções que realizam atividades, enquanto que as linhas indicam o fluxo de informações. Estas podem ser de quatro tipos, relativos a sua finalidade na função (figura 2.4). As atividades estão restringidas a operar com as "Entradas" recebidas, sob a orientação das informações de "Controle", com os recursos indicados pelos "Mecanismos". A "Saída" de um bloco servirá de subsídio para a execução de outras atividades.

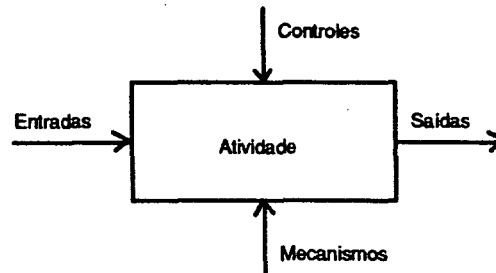


Figura 2.4 Esquema de representação de informações

A figura 2.5 apresenta a troca de informação entre as funções do Sistema de Manufatura. Basicamente, o Gerenciamento da Manufatura controla as funções de manipulação de recursos financeiros, autorizações para execução de tarefas, diretrizes de produção e concentra o estado das demais funções. A Fabricação de Produto, onde são realizadas as atividades de projeto, planejamento, produção e suporte aos produtos, concentra informações relativas aos requisitos de mercado, definição de produto, recursos para fabricação.

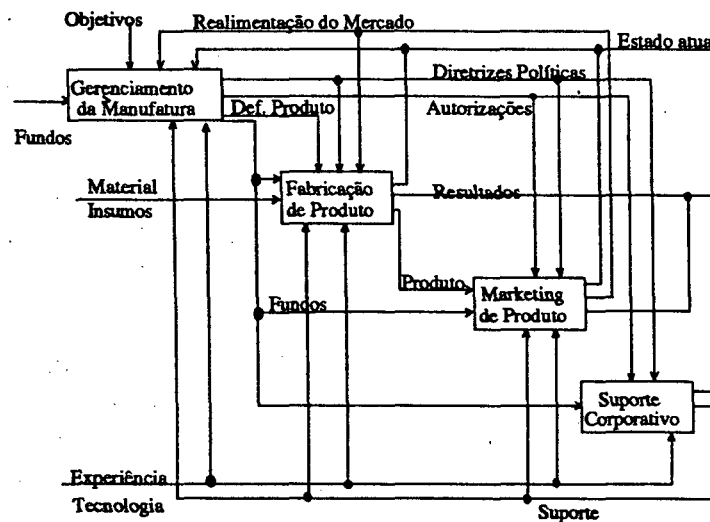


Figura 2.5 Fluxo de informações no Sistema de Manufatura [HAR84]

O Marketing de Produto é responsável por receber as informações sobre o produto, inserindo-as no mercado, ao mesmo tempo em que analisa as reações do mercado a fim de propor novos produtos. Por sua vez, o Suporte Corporativo mantém as informações relativas à legislação, economia, etc.

Alguns pontos importantes devem ser ressaltados:

- o Gerenciamento da Manufatura libera as autorizações para a execução das tarefas sem definir detalhes, apenas diretrizes;
- todas as tarefas retornam ao seu estado corrente;
- o Gerenciamento da Manufatura utiliza os objetivos corporativos e informações sobre o mercado para definir uma especificação preliminar do produto que será produzido. Existe um ciclo entre a definição, produção e marketing, visando depurar o produto;
- todas as funções tem suas atividades balisadas pelos objetivos corporativos;
- o conhecimento de especialistas, assim como de tecnologias, são mecanismos utilizados por todas as funções.
- informações sobre o produto final são transmitidas pela Fabricação do Produto ao Marketing do Produto visando dar subsídios às campanhas publicitárias.

A nível de Fabricação de Produto, o fluxo de informações ocorre conforme a figura 2.6.

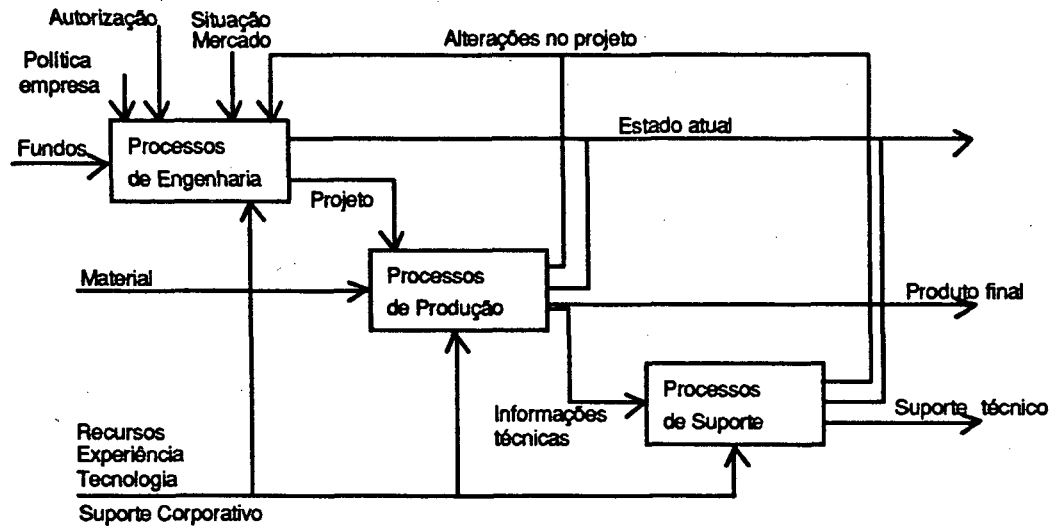


Figura 2.6 Informações na Fabricação de Produto [HAR84]

As funções de Processos de Engenharia ocorrem segundo a política da empresa e a situação do mercado, gerando o projeto do produto. Para cumprir esta tarefa são utilizados conhecimentos de especialistas. Estes conhecimentos são fruto de larga experiência no desenvolvimento de projetos, abrangendo áreas como:

- a) funcionalidade
- b) ergonomia
- c) montagem
- d) manufaturabilidade
- e) custo

Durante a fase de Processos de Engenharia as seguintes atividades são desenvolvidas:

- a) projeto do produto: o ato de projetar é uma atividade de criação da especificação de um produto realizada iterativamente, desenvolvendo um ciclo de síntese, análise e otimização. O resultado da especificação inclui geometria, topologia, tolerância, material e outros parâmetros necessários para sua produção [BED91];
- b) análise de engenharia: verifica o comportamento do projeto quando submetido a esforços. Para isso são utilizados diversos métodos numéricos que analisam, entre outros, tensão/deformação e dinâmica de mecanismos. Neste tipo de análise são usadas informações referentes ao projeto do produto, sua descrição geométrica e especificação de esforços.

A descrição completa do produto é realizada utilizando desenhos, memoriais descritivos, etc., contendo informações que serão utilizadas nas atividades desenvolvidas no Processo de Produção para planejar a execução e controle das operações de usinagem e montagem. No desenvolvimento desta função são realizadas as seguintes atividades:

- a) planejamento de processos: uma quantidade muito grande de informações é manipulada durante a geração do plano de processos. Informações referentes a geometria da peça, métodos de produção, ferramental, dispositivos, etc. Uma descrição detalhada do planejamento de processos encontra-se no capítulo 6;
- b) planejamento da produção: unindo as informações sobre os recursos disponíveis e as necessidades de produção é definido o escalonamento das operações. Segundo [BON89], as seguintes informações são necessárias para a geração do plano de produção:
 - . tamanho de lote, custo e política de produção
 - . informações dinâmicas sobre o estado da produção
 - . inventário de material
 - . processo de produção de cada produto
 - . descrição dos centros de trabalho

- . ferramental disponível por centro de trabalho
- c) geração de programa NC: a partir da geometria da peça e do plano de processos é definido o caminho das ferramentas para a usinagem da peça.
- d) usinagem e montagem: as atividades de usinagem e montagem são definidas no plano de processos e desenvolvidas nas estações de trabalho de forma a gerar o produto;
- e) inspeção: a inspeção deve ser realizada seguindo um plano estabelecido, gerando uma gama de informações que devem ser analisadas de forma a qualificar a produção.
- f) controle da produção: as informações de monitoramento dos planos de produção produzidos pelo planejamento da produção geram ações a serem tomadas e influem em diversas atividades. Devem ser realizadas correções quando necessário, modificando o plano original.

Um produto aprovado e testado, pronto para ser enviado ao mercado, necessita de suporte especializado em instalação, manutenção, operação, etc. A função dos Processos de Suporte requer que informações técnicas sejam conhecidas, de modo a preparar assistência técnica capaz de manter o produto.

Na indústria, em geral, as atividades não ocorrem seguindo procedimentos bem-definidos, criando uma estrutura complexa onde informações são perdidas ou geradas inutilmente. Mesmo nas atividades auxiliadas por computador predomina a desorganização e a dificuldade em armazenar e reutilizar informações geradas. Numa análise inicial, pode-se citar como problemas mais comuns no Sistema de Manufatura:

- a) projetos modificados e não documentados;
- b) despadronização entre as informações de cada atividade;
- c) geração de informações duplicadas ou inúteis;
- d) trabalho dobrado;

Sem dúvida que, variando de caso a caso, muitas outras informações e atividades estão presentes no Sistema de Manufatura de uma indústria metal-mecânica. Mas, dado que o objetivo deste trabalho é modelar uma plataforma genérica, sobre a qual se possa inserir quaisquer informações do Sistema de Manufatura, não é necessário que se esgote o assunto.

2.2 Engenharia Concorrente

Durante os primeiros passos da indústria automobilística, Henry Ford e Ransom Olds utilizavam, na fabricação de seus carros, equipes de especialistas unindo pessoal de projeto e produção [HYE91]. Desde há alguns anos, esta metodologia passou a ser reconhecida como um elemento de incremento da qualidade e produtividade no Sistema de Manufatura. O entendimento de Engenharia Concorrente é fundamental para que se possa tratar a informação de forma integrada.

2.2.1 Definição

Segundo Winner [REI92], "Engenharia Concorrente é um método sistematizado para o projeto concorrente e integrado de produtos com seus processos, incluindo produção e suporte. Esta abordagem procura considerar, em princípio, todos os elementos do ciclo de vida de um produto, da concepção até a distribuição".

O projeto de um produto é uma atividade que combina conhecimentos diversos, como marketing, usinagem e montagem. No enfoque tradicional, esta tarefa é realizada segundo uma seqüência de atividades estanques, em geral, realizadas por pessoas diferentes. Mais do que propor que estas atividades se desenvolvam paralelamente, ou compatibilizá-las, Engenharia Concorrente traz para a atividade de projeto

considerações sobre muitos fatores que afetam o custo total e o desempenho ao longo do desenvolvimento de um produto. Isto envolve a integração de atividades desenvolvidas em diferentes áreas funcionais da empresa. Além das necessidades do usuário, devem ser consideradas informações sobre manufaturabilidade (seleção de material, montagem, planejamento de processos), confiabilidade, testabilidade, distribuição (desmontagem, reciclabilidade), poluição ambiental, etc. [TER92].

Um estudo efetuado na Universidade do Arizona, EUA, demonstra que 70% do custo de fabricação de um produto é determinado durante o estágio de formulação do conceito de produto [BED91]. A concepção de produto não é a atividade mais demorada no ciclo de vida de um produto, mas sua influência sobre as demais é bastante intensa. Então, qualquer alteração de concepção nesta fase custa muito pouco, mas pode afetar fortemente o custo final de fabricação.

Tradicionalmente, a fase de projeto de uma peça possui uma série de problemas [BED91]:

- a) a definição de detalhes é muito demorada. Seria aconselhável que estes fossem definidos quando o projeto já estiver mais estabilizado;
- b) características de manufaturabilidade são consideradas muito tarde, somente quando o custo de mudanças é alto;
- c) planejamento da produção, manutenibilidade, confiabilidade e suporte são considerados somente após a atividade de projeto estar concluída;
- d) dados de projeto são fragmentados. A documentação inclui informações representadas de muitas formas diferentes, não sendo possível mantê-las consistentemente ao longo do ciclo de vida do produto;
- e) freqüentemente projetistas não possuem informações sobre custo.

O objetivo da Engenharia Concorrente é exatamente resolver estas deficiências já na fase de projeto, utilizando basicamente, formas de democratizar a informação, isto é, qualquer atividade que necessita informações manipuladas por outras atividades deve ter acesso a estas. Deste ponto de vista, pode-se representar Engenharia Concorrente conforme a figura 2.7. No centro da circunferência está colocado o coordenador do projeto, com a função de mediar os comentários e sugestões de reprojeção feitas pelos demais integrantes do grupo.

A linha geral do projeto é apresentada pelo coordenador aos demais participantes, e a partir daí o grupo define e redefine o projeto. Nesta "mesa-redonda" são permitidas "comunicações" em qualquer direção, visando a mais ampla divulgação das informações.

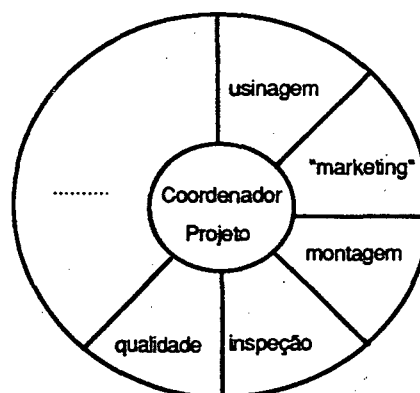


Figura 2.7 Conceito de Engenharia Concorrente

Dentro de Engenharia Concorrente uma série de metodologias são utilizadas na implementação. Projeto para a Manufatura e Projeto para a Montagem são duas das mais importantes e estudadas.

Projeto para a Manufatura (DFM) é a integração do processo de projeto com o planejamento de processos em uma só atividade. O objetivo é projetar de forma que seja fácil e eficiente produzir a peça.

Considerações sobre diretivas de manufatura são uma parte essencial do DFM. Exemplos incluem especificação de dimensão de furos dentro da capacidade das máquinas e ferramentas disponíveis, análise das possibilidades de fixação e sua relação com as tolerâncias requeridas, etc.

As diretivas de DFM são afirmações práticas da atividade de projetar, definidas empiricamente pela experiência de especialistas em projeto e manufatura. Podem ser citadas [BED91]:

- a) projeto com número mínimo de peças;
- b) projeto modular;
- c) peças multifuncionais;
- d) procurar a mínima tolerância possível entre as peças;
- e) ênfase na padronização;
- f) operações de usinagem e montagem simples;
- g) tempo de "set-ups" mínimo.

O processo de montagem é frequentemente a operação mais intensiva e soma a maior porção de custo. O conceito de Projeto para a Montagem (DFA) surgiu com o intuito de que o projeto assegure uma montagem mais simples e menos custosa.

2.2.2 Engenharia Concorrente ou Simultânea?

Os termos Engenharia Concorrente e Engenharia Simultânea tem sido utilizados como sinônimos para se referir a mesma metodologia. Geralmente o termo é associado ao escopo da definição utilizada.

Analisando a semântica das palavras, em inglês ou português, tanto concorrente quanto simultâneo significam "acontecendo ao mesmo tempo", mas concorrente tem uma idéia adicional de cooperação e concordância. Analisando os processos mais comuns de manufatura (produtos complexos), chega-se a conclusão que a palavra concorrente é mais adequada para o conceito de compartilhamento da informação e objetivos comuns apresentado anteriormente.

2.2.3 Estrutura básica para implantação

Na literatura, encontram-se exemplos de implantação de Engenharia Concorrente utilizando diversas abordagens. Para Schopbach [SCH91], a utilização de um Sistema Gerenciador de Informação de Produto é a chave para o sucesso de um ambiente de Engenharia Concorrente. Já Pennell [HYE91] aposta na utilização de grupos multi-funcionais. Para Terpenney [TER92], a implantação está fundamentada no tripé: estratégia de projeto, tomada de decisão e suporte a informação.

A metodologia de implementação é uma questão muito discutida, mas é possível classificar as abordagens em dois conjuntos: baseado em grupos multifuncionais e suporte para a informação auxiliado por computador. Estes dois são complementares e atuam juntos para que a implementação se concretize.

Grupos multifuncionais são constituídos por especialistas de diversas áreas que conjuntamente com o projetista definem o projeto do produto. O trabalho em equipe facilita a troca de informações permitindo que os possíveis problemas sejam resolvidos no momento em que são criados. A abordagem baseada em computador é muito abrangente, utilizando diversas ferramentas para representar e integrar as informações. O suporte computacional à informação inclui uma gama muito grande de tecnologias, mas basicamente constitui-se por modelagem das informações, tecnologia de "features", padrão PDES/STEP e integração de sistemas de informação.

Métodos de Engenharia Concorrente divergem do projeto de produto tradicional, no qual a informação é considerada como um elemento estanque de cada etapa do ciclo de vida de um produto.

Projetos criados usando métodos de Engenharia Concorrente utilizam o somatório das informações de todas as etapas [TER92].

As pesquisas em metodologias para Engenharia Concorrente tem se concentrado em alguns aspectos da real integração das informações. Grande parte dos trabalhos realizados analisam as questões da influência da manufatura no projeto, gerando uma metodologia chamada DFM (Projeto para Manufatura). Outros estudam problemas do projeto ligados a montagem - DFA (Projeto para Montagem), ou ainda Análise de Custos, quando estudados os aspectos relacionados com o custo final do produto. Segundo Terpenney [TER92], estes são apenas subconjuntos da Engenharia Concorrente, ocorrendo que muitas vezes o todo é esquecido. Sob este ponto de vista, é definida uma estrutura básica para a implantação:

- a) melhores estratégias de projeto;
- b) métodos de tomada de decisão;
- c) suporte a informação.

Estratégias de Projeto

A melhoria nas estratégias de projeto é necessária no sentido de simplificar o resultado final, tornando-o mais fácil de ser entendido e produzido. Estas estratégias tem como objetivo:

- a) simplicidade: poucas partes e geometria simples para produzir, montar, testar e inspecionar;
- b) padronização de peças e materiais: as atividades de projeto se tornam mais fáceis, reduzem o inventário e possibilitam a manipulação automática durante a usinagem e montagem;
- c) redução das tolerâncias e acabamento superficial: reduz os processos desnecessários;
- d) mínimo material (tanto em quantidade como tipos diferentes): reduz o custo de inventário e refugo;
- e) projetos modulares: permitem a reutilização de partes de um projeto em outros com características semelhantes.

Tomada de Decisão

A tomada de decisão é crucial no processo de Engenharia Concorrente. A decisão inclui desde questões de manufaturabilidade até poluição ambiental, logo são necessárias ferramentas que as auxiliem. Em geral são usados:

- a) grupos multi-funcionais
- b) Sistemas Especialistas

DFA e DFM, entre outras, são estratégias de tomada de decisão que podem tanto ser implementadas por grupos multifuncionais como, em nível menos abrangente, em computadores utilizando Sistemas Especialistas.

Suporte à Informação

A representação da informação serve como base para as atividades que se desenvolverão durante o projeto e tomada de decisão. A informação deve ser representada de uma forma tal que possa ser entendida por seus usuários.

2.2.4 Sistema de manufatura com Engenharia Concorrente

O fluxo de informações no Sistema de Manufatura se modifica para permitir a implementação de Engenharia Concorrente. Conforme apresentado na figura 2.8, adaptada de [REI92], não existe o retorno do projeto para os Processos de Engenharia após detecção de problemas nos Processos de Fabricação e de Suporte. As informações relativas a estas funções são fornecidas, em forma de recomendação, durante a fase de projeto. Ao final desta, as informações de planejamento devem estar praticamente concluídas.

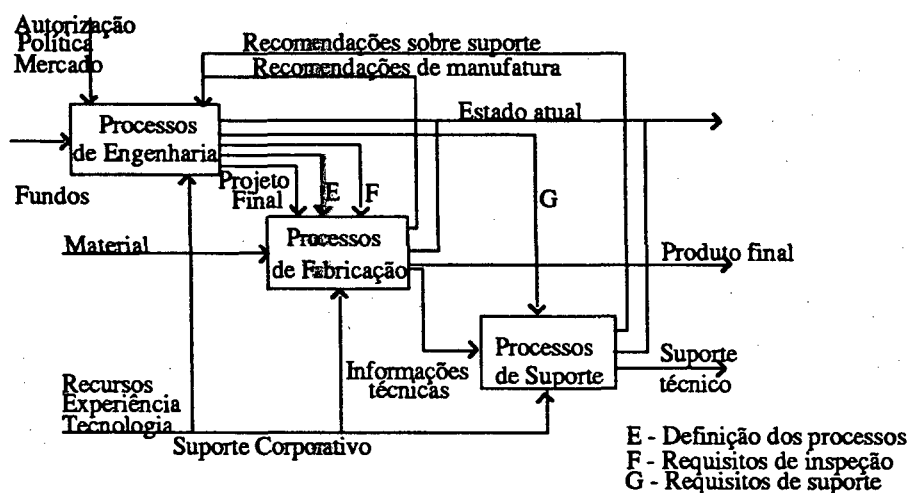


Figura 2.8 Fluxo de informações com Engenharia Concorrente

2.3 Manufatura Integrada por Computador - CIM

A Manufatura Integrada por Computador - CIM - já foi vista como a salvação da manufatura. Época: segunda metade dos anos 80. Quadro: seminários lotados, gerentes ávidos por informações, cada universidade um novo laboratório, cada empresa um conceito. Neste contexto, foram desenvolvidas diversas definições para CIM. Com o passar do tempo, a falta de um padrão, unida à inexistência de uma instituição que realmente o tenha implementado, descreditou a sigla e a abrangência da idéia.

O objetivo desta seção não é definir CIM, tampouco seus meandros político-filosóficos, mas sim analisar suas potencialidades como uma ferramenta de integração, capaz de suportar o fluxo de informações necessário ao Sistema de Manufatura.

2.3.1 Definições e desenvolvimentos em CIM

CIM foi ao longo dos últimos dez anos uma filosofia de fabricação almejada por muitas indústrias. Assim como qualquer filosofia, CIM é, conceitualmente, uma idéia abrangente e complicada, mas pode ser muito simples quando desmistificada e colocada no seu lugar - uma ferramenta de integração. Sob este ponto de vista, CIM é uma ponte para ligar as "ilhas de automação" que se espalham pelo Sistema de Manufatura, desde as funções de gerenciamento até o "marketing", passando, pela fabricação. CIM pode ser considerado, portanto, como uma grande rede de entendimento, e desta forma é fundamental para viabilizar a integração de informações e funções da manufatura.

Desenvolvimentos em microprocessadores e comunicação de dados possibilitaram a domínio da informação no local de trabalho. No ambiente de manufatura, isto se reflete em novos processos tecnológicos e ferramentas de gerenciamento, como projeto e manufatura auxiliados por computador (CAD/CAM), robôs, processamento de visão, Planejamento dos Recursos de Manufatura (MRP), Sistemas Flexíveis de Manufatura (FMS), Planejamento de Processos Auxiliado por Computador (CAPP), etc. A integração destas "ilhas de automação" da manufatura foi conceituada como sendo Manufatura Integrada por Computador - CIM [RAM92].

Em meados dos anos 80, o conceito de CIM tomou fôlego, tornando-se mais conhecido. A partir daí, muitas empresas, eufóricas com as promessas de aumento de produtividade e diminuição de gastos, passaram a trabalhar no sentido de desenvolver idéias e sistemas. Houve, inclusive, casos de empresas que passaram a vender sistemas CIM, como uma solução de prateleira. Quando a euforia passou, no inventário das perdas e ganhos, notou-se que os benefícios não foram tão grandes e principalmente, que as definições sobre o que seria Manufatura Integrada por Computador eram muitas. A confusão criada persiste até hoje, embora algum consenso esteja surgindo.

CIM não é um produto ou uma tecnologia que possa ser comprada. Antes disto, é uma ferramenta, ou uma abordagem para integração de organização e gerenciamento do Sistema de Manufatura. Este é o caminho para, organizando as tecnologias de manufatura, alcançar um fluxo de informações contínuo, eficiência, aumento da qualidade, rápido desenvolvimento de produtos e flexibilidade. Estas características tendem a tornar CIM específico para cada instituição [RAM92].

2.3.2 Implantação de CIM

O sucesso na implantação depende menos de tecnologia, em parte disponível, que de organização interna da instituição e uma clara visão dos objetivos a serem alcançados. Nesse sentido, todos os desenvolvimentos em CIM devem ser realizados com base em: [KIN88]

- a) simplificação;
- b) modificações no gerenciamento;
- c) alta taxa de utilização;
- d) respostas rápidas para necessidades do usuários e problemas internos;
- e) simulação e análise antes de execuções;
- f) treinamento de recursos-humanos.

A efetivação das mudanças necessárias, por certo, não é tarefa trivial. A participação de toda a estrutura da instituição é fundamental para o êxito da implementação. A alta gerência deve ser plenamente participante do processo, pois, as modificações a serem realizadas, a nível de estrutura organizacional, são bastante consideráveis. A simplificação do fluxo de informações deve ser a primeira tarefa a ser providenciada, e a última a encerrar.

A Jaguar Cars [WAL89], empresa britânica do ramo automobilístico, desenvolveu sua própria metodologia CIM, bastante prática e voltada as suas características. Em 1984, foi criada uma comissão sênior de direção dos trabalhos de levantamento das informações. O trabalho desenvolveu-se em equipes de estudos compostas por elementos de diferentes áreas de conhecimento e experiências. Ao final do levantamento as diretrizes foram traçadas e submetidas à diretoria para aprovação e a partir desta ratificação foram desenvolvidas uma série de atividades a longo prazo. O sistema estava em fase de testes em final de 1989. Os fundamentos que nortearam a implantação foram:

- a) utilização de sistemas de banco de dados compartilhados para o produto;
- b) um banco de dados de ordens de clientes concentra as informações de pedidos servindo de fonte para o planejamento da produção;
- c) planejamento dos recursos de manufatura criando um forte elo de ligação com fornecedores garantindo entregas sincronizadas com a produção;

- d) os cronogramas de todas as áreas sincronizados;
- e) contabilização através do estabelecimento de centros de controle de processo;
- f) utilização de sistemas de monitoração para gerenciamento.

Todas estas diretivas podem ser, basicamente, resumidas em manipulação da informação, ou seja, o processamento da informação a nível de cada atividade e o gerenciamento do fluxo de informações no sistema de manufatura como um todo.

As atividades realizadas no sistema de manufatura dificilmente podem ser completamente automatizadas, mas, certamente, algum tipo de auxílio computacional pode ser oferecido. Diversas ferramentas foram desenvolvidas com este fim, sendo conhecidas como CAX (por exemplo: CAD, CAE, CAPP).

2.3.3 Tecnologias associadas

Como já mencionado anteriormente, CIM não seria CIM sem uma série de tecnologias que produzem ferramentas de auxílio as atividades do Sistema de Manufatura. Estas desempenham suas funções desde as atividades de planejamento macro da instituição, até a monitoração de uma máquina no chão de fábrica. Segundo [AGU91], elas podem ser agrupadas nas seguintes categorias:

- a) sistemas de apoio à decisão;
- b) engenharia auxiliada por computador;
- c) projeto auxiliado por computador;
- d) manufatura⁴ auxiliada por computador;
- e) sistemas de gerenciamento da informação.

Algumas destas tecnologias, de especial interesse para este trabalho, serão analisadas detalhadamente no capítulo 3.

Sistemas de apoio à decisão - DSS

DSS envolvem as transações sobre informações de produtos e de produção, executadas por aplicações usadas nos estágios iniciais do projeto de um produto. Auxilia nas decisões, a nível corporativo, para o planejamento dos projetos a serem executados (por exemplo: sistemas de análise de mercado, modelos de simulação, etc.).

Engenharia auxiliada por computador - CAE

Similarmente a DSS, CAE envolve transações com informações de programas de aplicação usados no desenvolvimento e avaliação de especificações funcionais de produtos, peças e processos (por exemplo: avaliação, modelagem sólida, análise estrutural e de produção, estimativa de custos, etc.).

⁴Aqui, manufatura se refere as atividades de planejamento, controle e execução de um produto.

Projeto auxiliado por Computador - CAD

As tarefas realizadas no CAD são estreitamente relacionadas com o CAE, é utilizado na produção de desenhos, lista de materiais e outros conjuntos de instruções para as atividades subsequentes de produção (por exemplo: base de dados gráfica de peças, desenho, simulação gráfica interativa, base de dados de desenhos, armazenamento e acesso a documentos, edição de documentos técnicos, etc.)

Manufatura Auxiliada por Computador - CAM

O CAM desenvolve as atividades de planejamento de processos e operação, além da geração, transmissão e controle da execução, de programas de comando numérico aplicados a máquinas-ferramenta e robôs, sistema de manipulação de material, inspeção e teste. O CAM, por definição, engloba uma série de atividades, de certa forma, independentes que possuem suas próprias tecnologias associadas.

- a) CAPP - O Planejamento de Processos Auxiliado por Computador;
- b) CAP - Produção Auxiliada por Computador;
- c) CAT - Teste Auxiliado por Computador;
- d) CAI - Inspeção Auxiliada por Computador;
- e) CAQ - Qualidade Auxiliada por Computador.

Sistemas de Gerenciamento de Informações

Estas ferramentas suportam e controlam o fluxo de informações no sistema de manufatura (por exemplo redes de computadores, sistemas gerenciadores de banco de dados, etc.).

2.3.4 Modelos para CIM

O fluxo de informações do sistema de manufatura deve ser, de alguma forma, modelado visando a utilização em ambientes computacionais. Basicamente, existem duas abordagens, uma interfaceando diversos sistemas diferentes e outra integrando através de sistemas de banco de dados como elemento de ligação entre as ferramentas, sendo neste modelo, representadas todas as informações necessárias durante o ciclo de vida do produto. Na segunda alternativa existem muitas propostas de modelo diferentes. Pode-se imaginar a complexidade e o tamanho destes modelos, uma vez que a variedade de atividades diferentes, desenvolvidas no sistema de manufatura, é bastante grande.

A seguir são apresentadas algumas abordagens relatadas na literatura.

Interfaceando Ferramentas

Ao longo da década de 80, muitos sistemas de computador foram desenvolvidos para suportar empresas de manufatura. Esses sistemas eram construídos baseados em requisitos que previam funcionalidade e eficiência para atividades individuais, isso é, não foram construídos visando integração. Devido ao rápido desenvolvimento tecnológico as necessidades mudam drasticamente, aumentando a necessidade de acesso à informação. Paralelamente, os sistemas em uso geraram cultura nas empresas tornando-se difícil e custoso sua troca. A solução encontrada foi a construção de interfaces entre os diversos sistemas, muitas vezes de fabricantes diferentes. Pelletier, em [PEL91], tece as seguintes considerações a este respeito. "Um sistema de informações para a manufatura é composto de várias aplicações separadas, completamente funcionais, que, ao longo dos anos, foram 'integradas' devido a novas exigências dos

usuários. Então, o que fazemos? Integramos, usamos 'bridges', enviamos dados de um sistema para outro, desenvolvemos conexões em tempo real, encapsulamos dados, usamos transações secundárias. Em suma, usamos o único método de integração disponível no momento."

A arquitetura proposta pela IBM (IBM Poughkeepsie Logistics) [PEL91], é um exemplo desta abordagem. Neste modelo, extremamente complexo, todas as ferramentas mantêm seus bancos de dados próprios, armazenando aí as informações geradas. As necessárias trocas de informações entre diferentes sistemas são realizadas através de uma interface, ou seja, um programa que traduz um modelo para outro.

Alguns formatos de dados proprietários (de um fabricante), tornaram-se padrão pelo uso, como DXF da AutoDesk. Ao mesmo tempo foram criados formatos padrões para troca de informações entre sistemas, como IGES (Initial Graphics Exchange Specification) e mais recentemente STEP (Standard for Exchange for Product Model Data) desenvolvido pela ISO (International Standard Organization).

Essa abordagem gera alguns problemas:

- a) dados duplicados em diversos bancos de dados independentes;
- b) necessidade de atualização, manual, de diversos bancos de dados;
- c) custo de "hardware" é alto;
- d) usuários recebem dados desatualizados.

A grande parte das tentativas de implementação realizadas pela indústria utilizam este tipo de abordagem. Isso ocorre, entre outras causas, devido a cultura criada pela utilização dos antigos sistemas, o tempo já dispendido em treinamento e o custo de novos sistemas integráveis. Exemplos e abordagens similares são fartamente encontrados na literatura, [HOU91], [JAC91], [SSE90a].

Arquitetura de Sistemas de Informação

Em [CAM92], é apresentada uma proposta de utilização de Sistemas de Informação (SI) como elemento integrador para arquitetura CIM. Esta abordagem faz parte do projeto CIM-PLATO/Esprit.

Um sistema CIM, do ponto de vista de software, pode ser visto como uma arquitetura distribuída em módulos heterogêneos. Esses, são provenientes de diferentes fabricantes e desenvolvidos com base em ambientes diversos. Por distribuído heterogêneo entende-se a manutenção da autonomia de cada um dos módulos e adoção de procedimentos normalizados (regras de coexistência comum) no que diz respeito à sua integração.

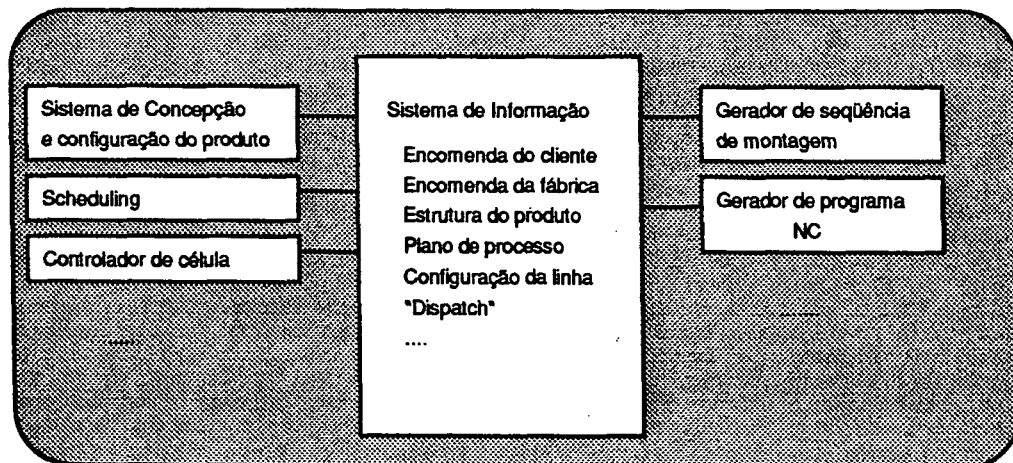


Figura 2.9 Exemplo de SI integrando atividades heterogêneas

Ao SI (figura 2.9) é atribuído o papel de núcleo de integração de diferentes atividades (representadas pelos retângulos menores), ou módulos de software, através da integração das informações provenientes de múltiplas fontes, assegurando sua coexistência e tornando-a disponível aos possíveis usuários.

Alguns problemas devem ser resolvidos para que essa arquitetura possa, efetivamente, ser colocada em prática:

- a) uma ferramenta deve poder comunicar com outras e com servidores de informação mesmo quando residam em computadores diferentes ou sobre sistemas operacionais diferentes. Para tal, torna-se necessário um ambiente de integração que disponha de meios para a troca das informações;
- b) o usuário deverá poder comunicar com os vários subsistemas de maneira homogênea, isso é, através de regras comuns de interface entre os vários módulos;
- c) a integração é baseada na definição de modelos globais bastante complexos, sendo necessário, portanto, normalizar as ferramentas de modelagem.

O ambiente de integração mencionado na alínea 'a' é realizado por troca de mensagens seguindo a filosofia Cliente-Servidor, na linha do modelo CIM-OSA (Arquitetura em Sistemas Abertos para CIM - do projeto Esprit).

Maiores detalhes sobre a arquitetura são encontrados na literatura, [ESP89], [CAM91], [OSO91].

Arquitetura baseada em "features"

Um dos maiores problemas enfrentados na integração é prover um modelo para as informações que seja representativo para todo o sistema de manufatura. A modelagem por "features", descrita em detalhes no capítulo 3, tem a capacidade de representar informações de grande parte do Sistema de Manufatura. Essa característica tem lhe assegurado um papel de destaque como elemento de integração do sistema de manufatura.

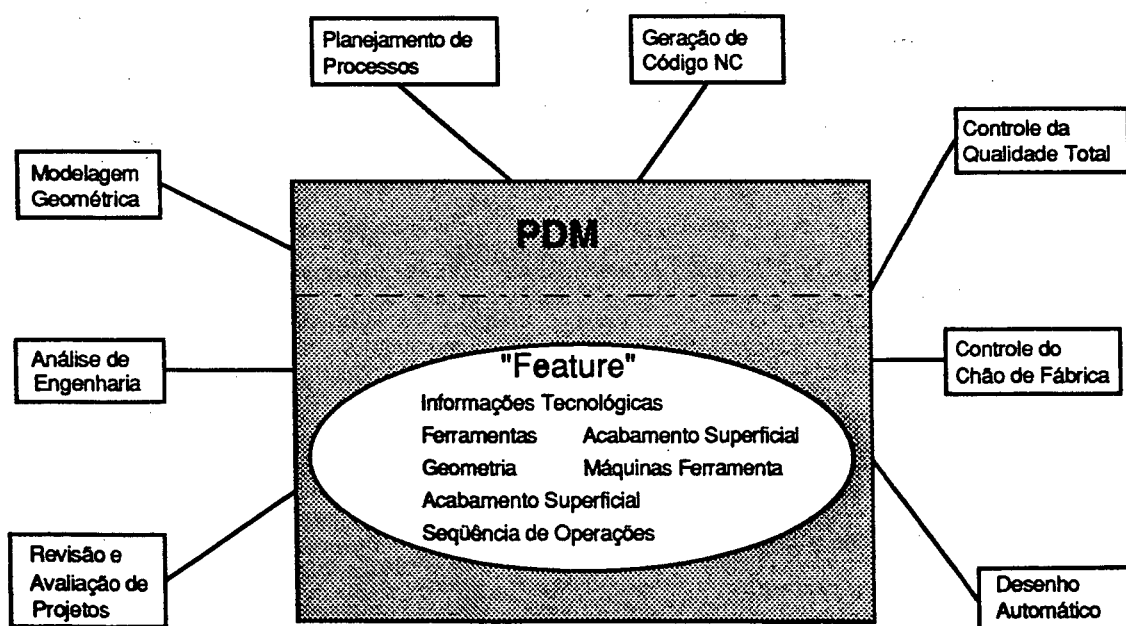


Figura 2.10 Integração por "features"

A figura 2.10 [VAR91] apresenta uma abordagem PDM (Modelagem de Dados de Produto) baseado em "features". PDM é um método pelo qual as informações de um produto podem ser representadas de tal

forma que as informações necessárias para o projeto, planejamento e produção são suportadas. Neste modelo também estão envolvidas as demais informações do ciclo de vida do produto. Nesta abordagem, "features" são utilizadas desde o projeto da peça até a geração de programa NC e controle do chão de fábrica.

Em [YEO91], é apresentado um sistema, baseado em "features" para planejamento de processos e geração de código NC de peças rotacionais. Este sistema reconhece "features" a partir de um modelo geométrico realizado em um CAD. Reconhecimento de "features" também é utilizado em [GRA88].

No capítulo 3 são apresentados trabalhos realizados utilizando Tecnologia de "Features" na modelagem de informações de manufatura.

Capítulo 3

FERRAMENTAS PARA A INTEGRAÇÃO DO SISTEMA DE MANUFATURA

S A T O R
A R E P O
T E N E T
O P E R A
R O T A S

A citação introdutória é uma inscrição latina que pode ser lida em quatro direções diferentes, sempre resultando na mesma construção. Seu sentido é enigmático — "o semeador mantém o mundo em sua órbita" — mas, a lógica e precisão com que se estrutura permitem inferir os requisitos básicos de uma boa ferramenta.

O Sistema de Manufatura na complexidade atual não seria integrável sem o uso de ferramentas que o modelem e suportem computacionalmente. Neste capítulo são apresentadas algumas destas ferramentas, com ênfase naquelas que embasam este trabalho.

3.1 Tecnologia de "Features"

Uma informação desvinculada de um contexto não é completa, permitindo dúbias interpretações. No Sistema de Manufatura é manipulada uma variada gama de informações. Destas, grande parte dizem respeito a definição do produto, sendo fundamental representá-la de forma padronizada.

A Tecnologia de "Features" tem como característica englobar as informações pertinentes a cada elemento de uma peça. Segundo [CHU88], os benefícios da utilização de "features" podem ser sintetizados em:

- a) os usuários podem expressar mais facilmente sua intenção durante o projeto, diretamente através da manipulação de "features";
- b) banco de dados de "features" permitem que sistemas dedutivos realizem tarefas como otimização heurística, análise de manufatura, etc.
- c) "features" podem armazenar conhecimento para facilitar a programação de máquinas NC, o planejamento de processos e a definição automática de malhas para elementos finitos.

Sistemas baseados em features tem demonstrado potencial para criar um ambiente de projeto, propício a aplicações como planejamento de processos e avaliação da manufaturabilidade. Várias metodologias tem sido usadas para criar modelos e bancos de dados [SHA91].

3.1.1 Definição de "Features"

Segundo Cunningham et al. [CUN88], uma "feature" "é qualquer forma geométrica ou entidade usada no desenvolvimento de um ou mais projetos ou atividades de manufatura." Essa definição é uma das tantas que se aplicam a fabricação. Muitas outras foram sendo criadas em função do uso particular em uma ou outra atividade do Sistema de Manufatura. No trabalho realizado por Shah et al. [SHA88b], estas definições são unificadas por uso e simplificadas:

- a) em modelagem geométrica - "Agrupamento de entidades geométricas ou topológicas que necessariamente são referenciadas juntas";
- b) em projetos - "Elementos utilizados na geração, análise ou avaliação de projetos";
- c) em planejamento de processos - "Formas ou atributos tecnológicos associados com operações de manufatura e ferramentas";

Não se pode dizer que exista uma definição de "feature" que seja universalmente aceita, nem que seu escopo esteja bem delimitado. No capítulo 5, é apresentada uma definição de "features", dentro do escopo deste trabalho, de forma a minimizar os problemas decorrentes das diversas interpretações dadas ao conceito de "features".

3.1.2 Identificação, Classificação e Representação de "Features"

Segundo [SHA88b], para uma eficiente utilização das potencialidades da Tecnologia de "Features", devem ser considerados três pontos principais:

- a) identificação das "features";
- b) classificação das "features";
- c) metodologia utilizada para representação das "features".

A Identificação e classificação de "features" é realizada com os seguintes objetivos [SHA88b]:

- a) definir um domínio de aplicação e, desta forma, simplificar a representação dos elementos;
- b) definir uma terminologia comum para o uso de "features";
- c) definir uma taxonomia comum, útil na definição de normas para o intercâmbio de dados.

Uma vez que não existe consenso com relação ao conceito de "features", e a identificação nele se baseia, os critérios a serem utilizados neste processo são da mesma forma conflitantes. Aumentando a polêmica, cada vez mais trabalhos são realizados nesta área. A seguir apresenta-se um resumo destes estudos.

Segundo Shah [SHA91], a identificação de "features" é realizada sobre um domínio limitado de peças, podendo ser por exemplo, o conjunto das peças rotacionais ou de produtos produzidos por uma empresa em especial. Para um domínio de problemas específicos é possível identificar um conjunto mínimo de componentes, de forma que as "features" emergem gradualmente nas atividades de projeto, fortemente relacionadas com a função da peça e podendo ser concebida antes que a peça real o seja.

Os estudos feitos pela empresa John Deere para a CAM-I⁵ [BUT86] identificam "features" fabricadas por processos de usinagem e por conformação de chapas metálicas. Com o objetivo de determinar todos os dados necessários para planejamento do processo, este trabalho define "features" em termos de sua forma genérica, parâmetros dimensionais, tolerâncias, etc. As "features" de forma são classificadas em três categorias: "features" laminares (chapas), "features" não-rotacionais (prismáticas) e "features" rotacionais. Cada uma destas categorias se subdivide em classificações, que nos níveis mais baixos são bastante específicas. Esta abordagem também comporta classificação para os materiais com base em sua composição (aço, liga de alumínio, etc), forma bruta (lâminas, barras, forjados, etc), tratamento térmico e acabamento de superfícies.

⁵ CAM-I — Computer Aided Manufacturing International, Inc. — realiza uma série de estudos relacionados ao Sistema de Manufatura.

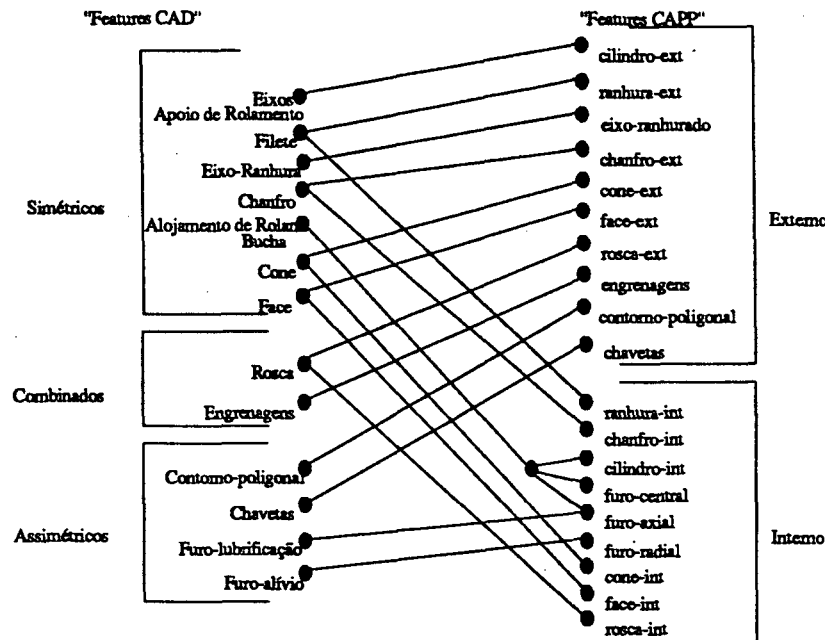


Figura 3.1 Relação entre "features" CAD e CAPP

A versão 1.0 do PDES (Product Data Exchange Standard) [SHA88b], classifica as features de forma como implícitos (procedurais), que adicionam novas informações ao modelo, e explícitos (enumerativos), que caracterizam-se pelo agrupamento de entidades geométricas. Neste caso as features implícitas são classificadas em passagens, depressões, protuberâncias, transições, etc.

Outros métodos classificam "features" com base no papel que desempenham no projeto de uma peça, como é o caso de Cunningham e Dixon [CUN88] que classificam features em estáticos e cinéticos.

O terceiro ponto crítico para a utilização eficiente da Tecnologia de "Features" é a forma de modelar e representar uma "feature" e conseqüentemente uma peça. Esta representação proporciona entre outras coisas a possibilidade de implementação computacional. As ferramentas computacionais para a representação das "features" são discutidas nos itens 3.2 e 3.4.

3.1.3 Modelos para uso de "features"

A representação de "features" em computadores pressupõe que estas tenham sido de alguma forma indicadas e nomeadas. Isso é, em algum momento, antes, durante ou depois da atividade de projeto, alguém ou alguma coisa deve identificá-la, atribuindo-lhe um nome e valores a seus atributos.

Em [SHA88b], é apresentado um exaustivo estudo sobre diversos modelos e discutidas suas diferenças.

Reconhecimento de "features"

Reconhecimento de "features" é a tarefa de extrair "features" de uma descrição geométrica. Alguns trabalhos resultaram em algoritmos que realizam esta tarefa automaticamente a partir de uma base de dados de CAD. A figura 3.2 representa esta abordagem.

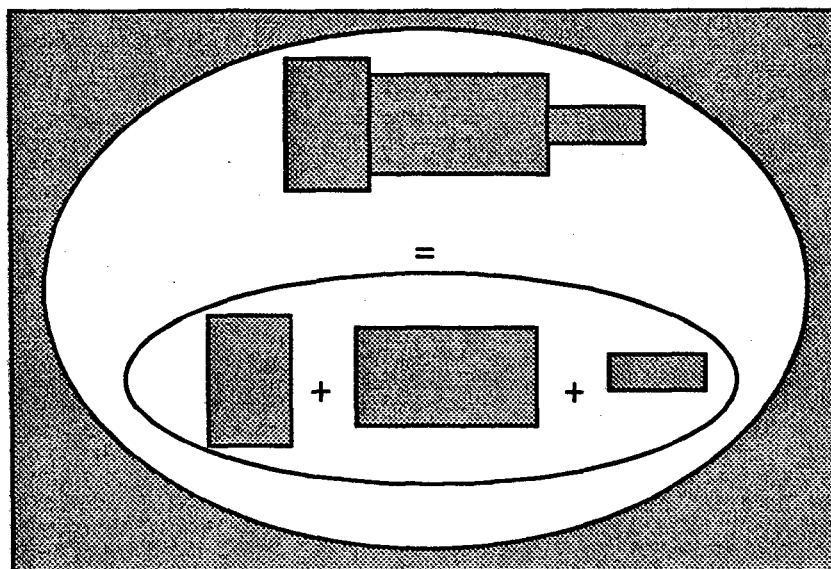


Figura 3.2 Reconhecimento de "features" [COR91]

Sahay et al. [SAH90], desenvolveram um algoritmo de extração de features direcionada para peças cilíndricas. Este domínio simplifica bastante o trabalho de extração. As "features" são extraídas utilizando uma definição baseada em propriedades geométricas. A partir desta premissa, a abordagem primeiro reconhece a existência de todas as features no contorno do objeto, sendo então classificadas como: independentes, simples ou complexa. As complexas são subdivididas em primitivas, sendo, mais tarde, classificadas utilizando Sistemas Especialistas.

Stanley et al. [STA83], utilizaram em seu trabalho uma abordagem usando reconhecimento de padrões orientado por sintaxe para extrair "features" de um banco de dados de CAD. Essa abordagem é aplicada na classificação de furos num corte seccional (2D) extraído do banco de dados de modelagem sólida (3D).

Estes trabalhos, assim como todos aqueles que necessitam de reconhecimento de padrões, necessitam de um grande esforço para tornar explícito o que está implícito em uma modelagem sólida. As informações não-geométricas, em geral, não são representadas no modelo sólido, devem da mesma forma ser identificadas, uma vez que são necessárias para diversas atividades do Sistema de Manufatura. Outro problema diz respeito a erros de interpretação durante a extração, devido a redundância de informações. Segundo Ssemakula et al. [SSE90b], o reconhecimento e extração de "features" pode ser desnecessário se existir um método que consiga reter as informações explicitamente representadas pelo projetista durante a atividade de projeto.

Diversos sistemas são relatados na literatura, [HEN88, JAR86, JAR89, LEE87, KRA88, TSU89, YEO91].

Projeto por "features"

Utilizando esta abordagem, os usuários criam modelos através do uso explícito de "features", sendo, desta forma, incorporadas desde o início do ciclo de vida do produto. A utilização deste método implica no uso de uma biblioteca de "features" e um modelador sólido baseado em "features". Diversas propostas são listadas e comentadas em [SHA88b], abaixo são citadas as mais importantes:

- a) biblioteca de "features" desvinculado do modelo geométrico: é utilizada uma linguagem de descrição de "features", através da qual são introduzidas as informações que a definem;
- b) modelagem Sólida Destrutiva (DSG): são utilizadas operações "booleanas" para construir o modelo a partir da remoção de "features" de um bloco sólido. Em geral, estas "features" são em pequeno número;

- c) modelos baseados na composição de "features": estes permitem projetar pela manipulação de "features" pré-definidas. Alguns trabalhos [CHU88, SHA88a] utilizam estas "features" com sistemas de herança baseado no paradigma de orientação a objetos. A figura 3.3 representa esta abordagem;

Vaghul et al. [KAR89], descreve um sistema para projeto e avaliação de moldes de injeção chamado IMPARD. Este sistema auxilia o usuário a projetar peças simétricas utilizando conceitos de manufaturabilidade somente para a "feature" principal da peça.

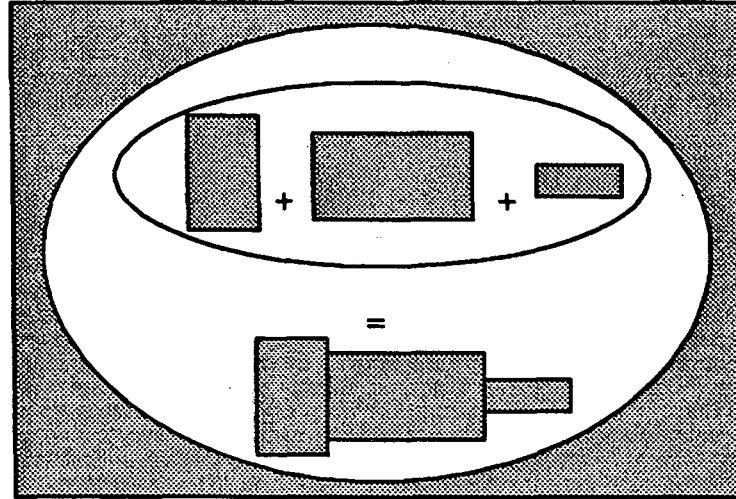


Figura 3.3 Projeto por "features" [COR91]

Em [CUN88], é apresentado um sistema CAD baseado em projeto por "features" (figura 3.4). O projetista constrói a peça manipulando "features" armazenadas em uma biblioteca. As operações realizadas sobre a peça são de inserir, modificar e apagar. A partir desta manipulação uma representação primária da peça é criada. A integridade das operações é garantida por um monitor que controla a manipulação das "features".

A representação primária é transformada em outras formas, chamadas em conjunto de secundárias, as quais são utilizadas por aplicações específicas. Esta conversão pode envolver alguma forma de reconhecimento de padrões para obter "features" mais abstratas. Por exemplo, algumas "features" podem ser divididas visando a definição de malhas de análise numérica.

A avaliação de manufaturabilidade é um dos objetivos principais da representação utilizando "features". Nesta abordagem esta análise é possível a partir das informações contidas nas "features", sem necessitar de dados geométricos do modelo sólido [KAR89].

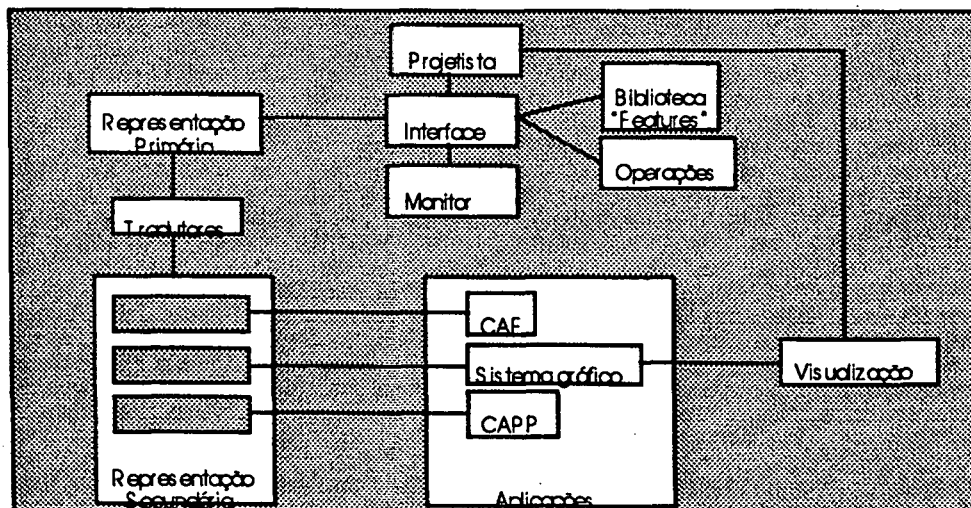


Figura 3.4 Estrutura para projeto por "features" [CUN88]

O projeto por "features" apresenta algumas dificuldades de implementação, pois, existe a necessidade de mudar constantemente a forma da peça enquanto se está projetando. Tradicionalmente o projetista concebe uma peça a partir de sua funcionalidade e o processista determina as atividades de produção. Entretanto, esta abordagem coloca o projetista frente a situação de considerar, além da funcionalidade, especificações de produção [KAR91]. As dificuldades em resolver as diferenças entre projeto e manufatura serão discutidas no próximo item.

Segundo [CUN88], o projeto por "features", ao mesmo tempo que limita o projetista, devido ao número finito de elementos, encoraja a padronização das peças. Sendo esta responsável por aumento de qualidade e manufaturabilidade.

Varvakis [VAR91], afirma que nesta abordagem é induzido o uso de processos específicos de produção. Nas peças com longa permanência no mercado, estes processos de fabricação tendem a se tornar ultrapassados quando não modificados.

Outros sistemas que utilizam abordagem de projeto por "features" são apresentados na literatura em [UNG88, ROL89, TSA89, JUR90, SCH90, LIT91, COR91].

3.1.4 Múltiplas "features" X "features" únicas

As diferentes abordagens para o uso de "features" no Sistema de Manufatura visam, basicamente, modelar as informações de produto de forma integrada. A dificuldade está em conseguir fazer convergir as diferenças existentes entre o projeto (composição de funcionalidade) e a produção (decomposição). A nível de projeto, as "features" tem a função de representar funcionalidade através de sua forma e informações encapsuladas. Em termos de produção o objetivo principal é armazenar informações relativas a execução da peça, em termos de definição das operações, seqüenciamento, escolha das ferramentas, dispositivos de fixação, etc.

A utilização de projeto por "features" implica em divergências em um ponto básico: uma "feature" tem sua existência ligada a atividade que está sendo desenvolvida; ou uma única "feature" representa todas as informações sob o mesmo nome. Alguns autores [SHA88c, CUN88, ROL89, JUR90], descrevem as múltiplas "features" utilizando, na divisão, a atividade como critério (projeto, planejamento de processos, conformação, usinagem, etc.). Outra corrente [ECK88, TSA89, RAD91] trabalha na hipótese de unir sob o mesmo referencial as informações de um amplo espectro de atividades, utilizando a forma do elemento para caracterizá-lo, univocamente, como uma "feature".

A utilização das múltiplas "features" requer a interligação de diferentes elementos, o que pode causar erros devido a dubiedade de interpretação [TSA89]. Por outro lado, o projetista está manipulando conhecimento relativo a sua atividade. "Feature" única dispensa ligações e interpretações, mas obriga ao projetista a trabalhar com informações que não lhe dizem respeito [ECK88].

3.1.5 Aplicações de Tecnologia de "Features"

As "features" são utilizadas como modelo para representação de informações de diversas atividades do Sistema de Manufatura.

"Features" em Projeto

Projetar usando Tecnologia de "Features" implica em criar modelos usando diretamente "features". Pode-se pensar em termos de uma biblioteca de "features", de onde são extraídas informações necessárias ao

projeto. Segundo [CUN88], projetar com "features" é a forma natural de comunicação entre engenheiros de projeto e fabricação sobre topologia e geometria de uma peça projetada.

A filosofia de construção desta biblioteca determina sua eficiência. A estrutura deve permitir manutenção e expansibilidade e acesso as informações de forma direta e fácil.

Em [SHA88a], é descrito o sistema FBMS ("Feature Based Modeling System"). Esse sistema consiste de uma camada de interface para acessar as "features" e de outra para seu mapeamento. A biblioteca de "features" possui mecanismos de orientação a objetos e Sistemas Especialistas. São usados três tipos de "features": "features" de forma, "features" de precisão, "features" de material.

"Features" em Planejamento de Processos

O planejamento de processos consiste em preparar um conjunto de instruções que descrevem como fabricar uma peça, ou realizar uma montagem, que satisfaça as condições estabelecidas no projeto [BED91].

Sistemas de planejamento de processos necessitam manipular conhecimento bastante subjetivo utilizado por processistas humanos. Esse conhecimento pode ser em parte representado por "features" sobre o qual são utilizadas técnicas de Inteligência Artificial (item 3.4) gerando o plano de processos.

Em [HER91a], [HER91b], é descrito o sistema IAGE. Nele são definidos planos de trabalho que descrevem a estratégia de fabricação. O sistema é generativo-iterativo, utilizando regras de produção, com pesos, para manipular o conhecimento. Em determinados pontos-chave interage com o processista para a tomada de decisão. É gerado um conjunto de planos de processo utilizando os recursos disponíveis que definem a seqüência de operações, as máquinas-ferramenta, o sistema de fixação e caracterizam a ferramenta para cada operação. O domínio de aplicação são as peças quasi-simétricas. Ou seja, eixos de transmissão, engrenagens, flanges, etc.

"Features" na geração de código NC

A utilização de "features" facilita a geração do programa NC, uma vez que as informações para a fabricação de cada elemento da peça estão aí disponíveis. Uma abordagem bastante interessante para este problema é apresentada em [CON88].

"Features" em CAQ

A integração entre CAQ e CAM é necessária para que se possa manter a qualidade do produto em todo o lote. Os dados adquiridos na inspeção "in-process" devem ser tratados pela CAQ de forma a verificar distorções em relação aos parâmetros de tolerância definidos na fase de projeto. Essa verificação deve enviar mensagens para o CAM permitindo alterar o processo antes que a próxima peça seja processada.

A utilização de representação de conhecimento por "features" permite que a troca de informações seja feita de forma mais ágil e sem intervenção humana. Uma abordagem ao problema de interface entre CAQ e CAM é relatada por Elshennawg em [ELS89].

3.2 Conceitos Básicos de Orientação a Objetos

A utilização do paradigma de orientação a objetos vem sendo maior a cada dia. Sistemas Orientados a Objetos (OOS) tem suas raízes em linguagens de programação orientadas a objetos (por exemplo: Simula, Smalltalk, etc.) sendo que com o uso, seus conceitos básicos passaram a influenciar a Inteligência Artificial, Bancos de Dados, Engenharia de Software e CIM [KIM90].

O principal objetivo de OOS é o encapsulamento de dados e procedimentos, isso é, tanto dados como os procedimentos pertencentes ao objeto são invisíveis para o usuário. Uma interface provê os meios de disparar procedimentos internos que realizam as ações [NIE88]. Esta característica permite que sistemas complexos sejam divididos em pequenas partes autônomas hierarquicamente estruturadas, simplificando o entendimento do todo [THO89].

Nos capítulos 4, 5 e 6, uma série de conceitos de OOS são subentendidos, de forma que no próximo item apresenta-se uma curta introdução sobre o assunto. Farto material para consulta a respeito dos conceitos de orientação a objetos pode ser encontrado na literatura [STE86, NIE88, THO89, HUD89, KIM90].

3.2.1 Definição e implementação

Em sistemas orientados a objetos todas as entidades conceituais são modeladas como *objetos*. Os objetos podem ser analisados sob dois pontos de vista: interno e externo.

Objetos: por dentro

Um objeto consiste de dados internos, ou *atributos*, que armazenam seu estado atual e *métodos* que realizam operações e modificam o estado dos objetos, ou seja, modificam os valores dos atributos.

O valor de um *atributo* é também um objeto, com suas características intrínsecas. Um *objeto primitivo*, (por exemplo: um inteiro, "string", etc.) não tem atributos, o valor é o objeto em si. *Objetos complexos* contêm atributos que podem referenciar outros objetos, e assim por diante, em cadeia. O comportamento de um objeto é encapsulado em métodos, ou seja, um procedimento que realiza uma ação sobre os atributos. Um objeto pode retornar, por exemplo, seu estado atual.

As *propriedades* de um objeto, seus métodos e atributos, não são visíveis de fora do objeto. A comunicação com o exterior é realizada através de *mensagens*. Estas, constituem a interface do objeto. Uma mensagem ativa um método correspondente que realiza a ação necessária, sendo, portanto, o elemento que relaciona objetos com o mundo externo e entre si.

Objetos: por fora

Todos os objetos carregam consigo seus próprios atributos e métodos, de forma que a quantidade de informação a ser especificada e armazenada pode ficar muito grande. Por esse motivo, objetos similares são agrupados em *classes*. Todos os objetos sob a mesma classe são descritos pelos mesmos atributos e métodos, respondendo as mesmas mensagens.

As classes de objetos são estruturadas hierarquicamente em *subclasses* e *instâncias*, sendo que suas propriedades são as mesmas da classe, ou *superclasse*, a qual estão ligadas. Essa "passagem" de propriedades entre superclasse e classe é chamada *herança*.

A hierarquia de classes implementa o conceito de especialização, ou seja, existe uma relação de é-um entre classes interligadas, o nível mais baixo é uma classe mais detalhada que o nível anterior. Por exemplo, a classe veículos descreve as propriedades dos veículos de uma forma geral, e possui uma subclasse — veículos-motorizados — onde uma série de propriedades ligadas aos motores são especificadas. Neste caso, um veículo-motorizado é-um veículo.

As mensagens entre objetos são descritas por um *protocolo*, isto é, uma padronização da forma das mensagens trocadas entre os objetos;

Objetos: complementos

Algumas aplicações de orientação a objetos necessitam de uma estrutura operacional que as suporte. Os Gerenciadores de Objetos tem a função de manter, em tempo de execução, serviços do tipo: concorrência, distribuição de mensagens, nomes de objetos e segurança [NIE88]. Este tipo de serviço é fundamental para garantir independência entre objetos.

3.2.2 Planejamento e Programação do Sistema de Manufatura Orientado a Objetos

Um método efetivo para o desenvolvimento de sistemas distribuídos complexos é o uso do paradigma de orientação a objetos. Devido seu estilo intuitivo de modelagem e capacidade de encapsular informações, as diversas funções do Sistema de Manufatura podem ter seu inter-relacionamento corretamente representado [TER89].

Em [KIM90] é apresentado um Sistema Gerenciador de banco de Dados Orientado a Objetos (OODBMS) voltado para aplicações em CAD. O OODBMS ORION suporta características fundamentais para a utilização em sistemas CAD. Entre elas as mais importantes são: evolução dinâmica das estruturas de representação, controle de versões, notificação de modificações e gerenciamento de transações entre objetos. Segundo Kim et al., um DBMS tradicional, não foi projetado para trabalhar com aplicações dinâmicas como o caso das tecnologias envolvidas em CIM. A utilização de objetos permite que um objeto complexo qualquer seja representado como um conjunto de instâncias de objetos simples recursivos.

Em [PHA91], é apresentado um Sistema Especialista, utilizando representação do conhecimento baseado em objetos, para gerar um programa para máquinas de medir por coordenadas (CMM) a partir dos dados de um CAD. O sistema é construído sobre uma plataforma para Sistemas Especialistas chamada LEONARDO.

Dulli [DUL91], apresenta um OODBMS que suporta a manipulação de dados de objetos de geometria sólida. Alguns aspectos técnicos são particularmente apontados: ambiente de modelagem, arquitetura do sistema e linguagem de manipulação de dados. O sistema considera que manufatura e objetos geométricos compartilham informações geométricas para construir e representar o objeto, ou com outras informações do objeto com o propósito de avaliar a manufaturabilidade de objeto.

Um modelo de manufatura integrada utilizando orientação a objetos é apresentada em [CAR92]. Neste trabalho, os diversos objetos que compõe o ambiente CIM trocam mensagens definidas por um protocolo. Outra abordagem, bastante dedicada ao controle do chão de fábrica, é descrita em [WAG92]. Neste trabalho é utilizado o protocolo MMS ("Manufacturing Message Specification"). para modelar equipamentos inteligentes (robôs, Controladores Lógicos Programáveis, Máquinas de Comando Numérico, etc.) em termos de objetos. Estes, trocam mensagens entre si, viabilizando o controle da produção.

Uma abordagem orientada a objetos para Engenharia Concorrente é apresentada em [CHE92]. As ferramentas são construídas utilizando, como suporte, um banco de dados orientado a objetos, no qual são armazenadas as informações sobre o produto. Esta representação faz uso de "features" orientadas a objetos.

3.3 Sistemas abertos, distribuídos e heterogêneos

Quando, hoje, se fala em grandes sistemas de computador, subentende-se sistemas integrados, sendo executado em diversas máquinas, algumas delas incompatíveis entre si. Não raras vezes, os fornecedores de cada parte são concorrentes. Sistemas de computador, hoje, são sinônimo de redes, sistemas distribuídos, roteadores, "gateways", "bridges", sistemas abertos, cliente-servidor, ambientes heterogêneos, enfim, conectividade em geral. Uma análise das tendências na indústria de computação aponta claramente para a

obtenção de soluções únicas e simples para o gerenciamento de sistemas operacionais e protocolos de rede diversos [RIH92]. Facilidade de utilização e conformidade com sistemas abertos estão na ordem-do-dia.

3.3.1 Definição e caracterização

Sistemas distribuídos possuem diversas definições, cada uma delas levando em consideração um aspecto específico. Mas é possível dizer, genericamente, que em sistemas distribuídos as funções computacionais são alocadas entre vários elementos processadores. Estes podem, ou não, estar separados fisicamente [LAG86]. Segundo [SAU92a], computação distribuída é um modelo de arquitetura de computação no qual uma coleção de processos, recursos e computadores distribuídos numa rede funcionam conjuntamente para executar tarefas. Estas definições são complementadas por uma série de requisitos que devem ser satisfeitos por um sistema distribuído [SAU92b]:

- a) transparência, isto é, o usuário não necessita saber em que máquina uma informação está armazenada;
- b) modularidade, ou seja, deve ser possível aumentar gradativamente o número de recursos ligados, sejam eles físicos ou não;
- c) interoperabilidade, pois, a heterogeneidade dos sistemas, tanto a nível de software como "hardware" é inevitável;
- d) padronização para garantir a interoperabilidade;
- e) segurança no tratamento dos dados.

Sistemas Gerenciadores de Banco de Dados Distribuídos (DDBMS) utilizam os mesmos conceitos de sistemas distribuídos para o armazenamento e manipulação de informações. Estes, estendem as facilidades usuais de gerência de dados de tal forma que o armazenamento de um banco de dados possa ser fragmentado ao longo dos nós de uma rede de comunicação, sem que com isso os usuários percam uma visão integrada do modelo [CAS85].

O uso prático de um DDBMS muitas vezes impõe a necessidade do uso de heterogeneidade, isso é, sistemas distribuídos que agregam componentes diferentes entre si, ou seja, modelo de dados, linguagem de consulta e estrutura [ELM90]. A utilização deste tipo de estrutura possui diversas dificuldades. A resolução destes problemas implica na utilização de novas tecnologias — Sistemas de Banco de Dados Heterogêneos (HDBS), ou Federados. No item 3.3.2 os HDBS são descritos com mais detalhes.

Um só fornecedor, usualmente, não tem capacidade para produzir uma solução completa para um problema complexo como os do Sistema de Manufatura. Essa tendência está sendo superada pela utilização de sistemas abertos. A partir deste conceito, tanto "hardware" como "software" de diferentes fornecedores são intercambiáveis e podem ser combinados em um ambiente operacional interligado. Para que isso possa efetivamente acontecer duas premissas devem ser cumpridas no desenvolvimento das plataformas:

- a) portabilidade - um sistema é portátil se puder ser transferido para uma outra plataforma sem que para isso tenha que ser re-projetado;
- b) interconectividade - se diz de plataformas que possam ser ligadas através de alguma forma de interfaceamento.

O paradigma mais usado para a construção de sistemas distribuídos abertos é o modelo cliente-servidor. Neste, aplicações clientes requerem serviços de um processo servidor. O cliente-servidor requer que convenções de comunicação sejam pré-estabelecidas. Esse conjunto de regras é conhecido como protocolo e deve ser implementado e reconhecido por todos os participantes do sistema [SUN90, SAU92c, MOU92].

Obviamente, todos estes conceitos de nada valeriam, se não existissem as redes de computadores. A discussão sobre este assunto é bastante extensa, fugindo aos objetivos deste trabalho. Na literatura [LAG86, TEN81, SUN90] é encontrada abrangente descrição sobre o assunto.

3.3.2 Sistemas de Banco de Dados Heterogêneos

Um Sistema de Banco de Dados Heterogêneo (HDBS) é uma coleção de Sistemas de Banco de Dados (DBS) sobrepostos e cooperantes. A sobreposição está na modelagem de uma mesma entidade em mais de um DBS. Isto é uma tendência em "workstations". A interoperação de HDBS é uma forma de manter autonomia, ao mesmo tempo que distribui as informações [AHM91]. Os DBSs são integrados em diferentes níveis. O "software" que controla e coordena manipulações nos DBSs é chamado de Sistema de Gerenciamento de Banco de Dados Heterogêneo (HDBMS) (figura 3.5) [SHE90].

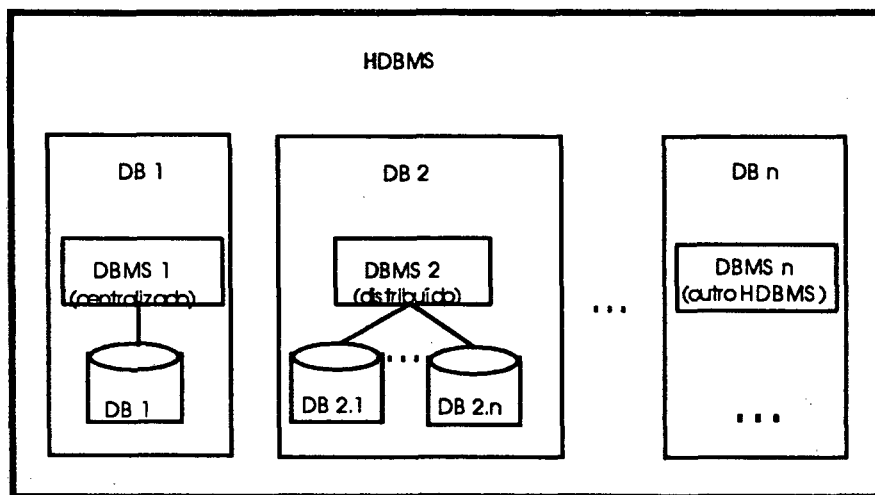


Figura 3.5 Os componentes de um HDBS

Sistemas compostos de múltiplos DBSs, dos quais HDBS é um tipo específico, podem ser caracterizados por três componentes [SHE90]:

- a) **distribuição:** os dados podem estar distribuídos entre múltiplos DBSs. E estes armazenados em um único computador ou em vários, geograficamente distribuídos ou não, de forma que, os dados podem ser repartidos de diversas maneiras. Isto inclui: particionamento de bases de dados e múltiplas cópias, de alguns ou todos os dados, mantidas não necessariamente estruturadas. Em um DBMS Distribuído, a alocação dos dados pode ser induzida deliberadamente, para tirar vantagem da estrutura. Por outro lado, no HDBMS, a maior parte da distribuição dos dados já existia, decorrência do uso de múltiplos DBMSs anteriores a construção do HDBMS;
- b) **heterogeneidade:** muitos tipos de heterogeneidade que ocorrem devido a diferenças referentes a "hardware", sistema operacional, etc., são solucionáveis a partir do uso de padrões de conectividade. Problemas de heterogeneidade a nível de DBMS, no entanto, são mais complexos. Estes podem ser:
 - heterogeneidade devido a diferenças entre DBMS: uma instituição pode ter múltiplos DBMSs, A este nível problemas com diferenças na estrutura da base de dados, restrições entre tipos de dados iguais em DBs diferentes e diferenças entre linguagens de consulta (query languages) devem ser resolvidos;
 - heterogeneidade semântica: ocorre quando existe discordância sobre, significado, interpretação ou intenção de uso de um mesmo dado. Eliminar heterogeneidade semântica é um problema bastante difícil. Tipicamente, estruturas de DBMS não possuem semântica suficiente para interpretar dados consistentemente.
- c) **autonomia:** as entidades que manipulam diferentes DBSs são geralmente autônomas. Em outras palavras, DBSs estão frequentemente sob controle separado e independente. Um DBS participante de um HDBS deve possuir diversos tipos de autonomia:
 - de projeto: refere-se a habilidade do DBS de escolher seu próprio projeto com respeito: aos dados, à representação, à concepção semântica dos dados, restrições, funcionalidade do sistema, compartilhamento de informações;
 - de comunicação: diz respeito a habilidade de um DBMS decidir querer comunicar com outros componentes de um HDBS;

- de execução: refere-se a habilidade de um DBMS executar operações locais sem interferência de operações externas;
- de associação: implica que um DBMS tem a habilidade de decidir quando compartilhar suas funcionalidades.

Tipos diferentes de habilidades podem ser atendidas pelas HDBMS. Estas incluem [THO90]:

- a) estrutura de integração - permite uma visão global do HDB;
- b) processamento distribuído de consulta - diz respeito a análise, otimização e execução de consultas a base de dados;
- c) gerenciamento distribuído de transações - negocia com a durabilidade de uma transação;
- d) funções administrativas - incluem tarefas como autorização, definição e submissão de restrições semânticas entre dados e gerenciamento de dicionários de dados;
- e) manipulação de diferentes tipos de heterogeneidade - ou seja, "hardware", sistema operacional, elos de comunicação, DBMSs de fornecedores diferentes e/ou modelos de dados.

Existe no mercado uma série de DBMS que suportam diversos tipos de heterogeneidade. Significantes avanços tem sido realizados nesta área, mas "certamente ainda não é possível comprar um sistema de prateleira que una todos os modelos de dados populares, suportando processamento distribuído de consulta, gerenciamento distribuído de transações." [THO90]

Sistemas que suportam interoperabilidade entre múltiplos DBSs autônomos estão surgindo. A maioria dos sistemas comerciais como SQL Server, Distributed Ingres e SYBASE, entre outros, tem evoluído nesta direção [LIT90].

3.3.3 O Sistema de Manufatura neste contexto

As aplicações de bancos de dados visando a integração do Sistema de Manufatura são bastante complexas sendo necessário a utilização de sistemas que, além de permitir sistemas distribuídos, garantam a integridade dos dados mesmo quando estes estão armazenados em diferentes sistemas gerenciadores.

A integração do Sistema de Manufatura requer a interligação de diversos aplicativos de "software", e bancos de dados. A partir destes pré-requisitos, podem-se citar os seguintes motivos para a utilização de HDBS:

- a) o ambiente do Sistema de Manufatura é geograficamente distribuído, dificultando a utilização de um computador central;
- b) as necessidades computacionais de cada função do sistema de manufatura são muito diferentes, requerendo computadores com características diferentes;
- c) uma série de banco de dados, nem sempre compatíveis, são usados no Sistema de Manufatura ligados a aplicações específicas. A interligação destas informações é um pré-requisito para a integração do Sistema de Manufatura.

As "ilhas de informação" se espalham por todo o Sistema de Manufatura, desde o nível corporativo até o chão de fábrica. Integrá-las coerentemente é o objetivo final de qualquer política de modernização do Sistema de Manufatura [BED91].

3.4 Sistemas Gerenciadores de Base de Conhecimento

A construção de Sistemas Baseados em Conhecimento (KBS) possui características próprias que inviabilizam o uso de facilidades do tipo Sistemas Gerenciadores de Base de Dados (DBMS) tradicionais. Segundo Mattos [MAT89], isto ocorre devido as seguintes razões:

- a) DBMS não suportam o processo incremental de construção de uma base de conhecimento;

- b) modelos clássicos não são satisfatórios para modelar conhecimento (perda de conceitos de abstração, frágil orientação a objetos);
- c) DBMS não suportam eficientemente necessidades de processamento de conhecimento;
- d) mecanismos de controle de integridade são frágeis (perda de manutenção automática da integridade semântica);
- e) acoplamentos entre DBMS e sistemas de conhecimento são, em geral, de baixo rendimento;

Durante os últimos anos, têm sido realizadas muitas pesquisas no sentido de desenvolver uma nova geração de sistemas capazes de efetivamente suportar a construção e processamento de KBS. Estes sistemas, unindo técnicas de AI e DBS, são chamados Sistemas Gerenciadores de Base de Conhecimento (KBMS), com características que permitam gerenciar eficientemente grandes bases de conhecimento compartilhadas.

Os KBMS devem, principalmente, suportar meios de descrever, avaliar e reestruturar o conhecimento de uma aplicação. Outras características devem ser respeitadas [MAT90]:

- a) meios para suportar o desenvolvimento incremental - Durante a construção de um KBS, o engenheiro de conhecimento deve utilizar o modelagem do conhecimento como uma ferramenta para definir dinamicamente o conteúdo da base de conhecimento (KB) sem perder as informações existentes;
- b) eliminação das diferenças entre estrutura e conteúdo na base de conhecimento - como uma consequência do requisito anterior, o modelo deve integrar, explicitamente, metac conhecimento (por exemplo, descrição das classes) e deste modo eliminar as diferenças entre estrutura e conteúdo na base de conhecimento;
- c) estabilidade da estrutura da KB durante o processamento do KBS - Durante a operação de um KBS, modificações na estrutura da KB (por exemplo, alterações em atributos, classes, etc.) são praticamente nulas. Essa característica deve ser explorada para aumentar o desempenho das aplicações;

Sobre a utilização de KBMSs na modelagem de informações técnicas, essenciais ao Sistema de Manufatura, DeBloch et al. [DE89] tece o seguinte comentário. "... o uso de AI suporta a modelagem de informações técnicas satisfatoriamente. Métodos heurísticos permitem a especificação de seleção inteligente de processos. Metodologias orientadas a objetos suportam a integração necessária dos procedimentos na descrição de objetos técnicos, permitindo a especificação de métodos de cálculos e de ações sobre elementos estáticos. "Demons" e regras podem ser aplicadas para diversos propósitos: especificar restrições de integridade, controle do andamento de projeto, etc."

KBMS é um assunto bastante amplo, não sendo necessário, para o entendimento deste trabalho, um maior detalhamento de seus meandros. A literatura é farta em apresentar exemplos de utilização [TAN90, DE89, DE91] e detalhes construtivos [MAT88, MAT89, MAT91].

3.4.1 O sistema Krisys

KRISYS (Knowledge Representation and Inference System) [MAT89, MAT91] é um KBMS desenvolvido na Universidade de Kaiserslautern, Alemanha, que reflete as considerações discutidas no item anterior.

O KRISYS tem como origem o estudo da possibilidade de utilizar um DBMS convencional como suporte para a implementação de um KBMS. A pesquisa foi realizada no sentido de colocar ao alcance do usuário as potencialidades de um sistema de representação (que se aproxima da idéia de "frames", com conceitos de abstração e orientação a objetos), bem como de um mecanismo de inferência flexível, baseado em regras de produção. O DBMS utilizado foi o INGRES, da Ingres Corporation Inc. [LEM92].

O KRISYS é arquiteturalmente dividido em módulos que refletem os aspectos de multinível, fundamental para suportar independência de conhecimento (figura 3.6) [MAT88]. O nível de aplicação corresponde interface sistema/aplicação, realizada pela linguagem KOALA (KRISYS Object-Abstraction Language). Esta linguagem permite ao usuário ou KS, uma visão funcional da KB.

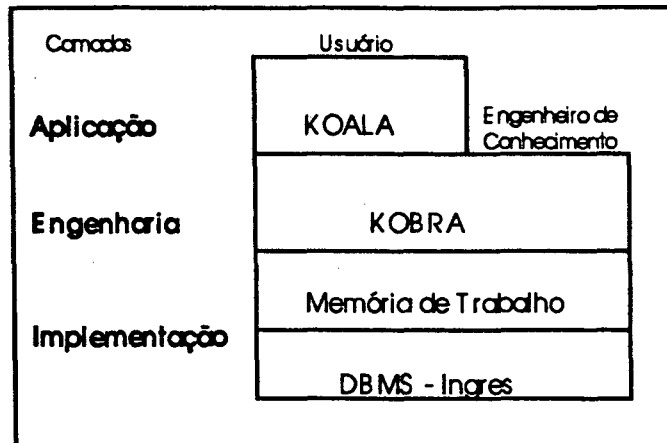


Figura 3.6 Arquitetura de KRISYS

A camada de Engenharia manipula com os aspectos descritivos, organizacionais e operacionais do modelo de representação do conhecimento. É utilizada pelo "Engenheiro de Conhecimento" quando da construção da base. Nesse nível central é implementada a linguagem para a representação de conhecimento de Krisys, chamada KOBRA (KRISYS Object-centered Representation Model).

A camada de Implementação realiza a tarefa de armazenar as informações de forma eficiente. Está construída para servir de interface entre o KRISYS e o DBMS onde são efetivamente armazenadas as informações. Esta camada está dividida em dois níveis:

- DBS Kernel : responsável pelo armazenamento e manuseio da base de conhecimento.
- Working Memory System : responsável pela redução de chamadas ao DBS, e pelo acesso mais rápido aos objetos da base de conhecimento. Funciona como um "buffer" de informações.

Representação de conhecimento no KRISYS

O modelador de conhecimento KOBRA integra, conhecimento descritivo, organizacional e operacional em um elemento básico chamado "schema". Este representa todas as entidades do domínio modelado (por exemplo, Bebidas no exemplo da figura 3.7). Um "schema" (alguns autores chamam de "frame", "unit" ou objeto) é unicamente identificado por um nome (isto é, o identificador do objeto), sendo suas características representadas por um conjunto de atributos. Atributos podem ser de dois tipos: "slots" — usados para representar as propriedades estáticas do "schema" (por exemplo, preço) e relacionamento com outros "schemas" (por exemplo, acompanhamento); ou métodos — usados para expressar aspectos comportamentais do objeto.

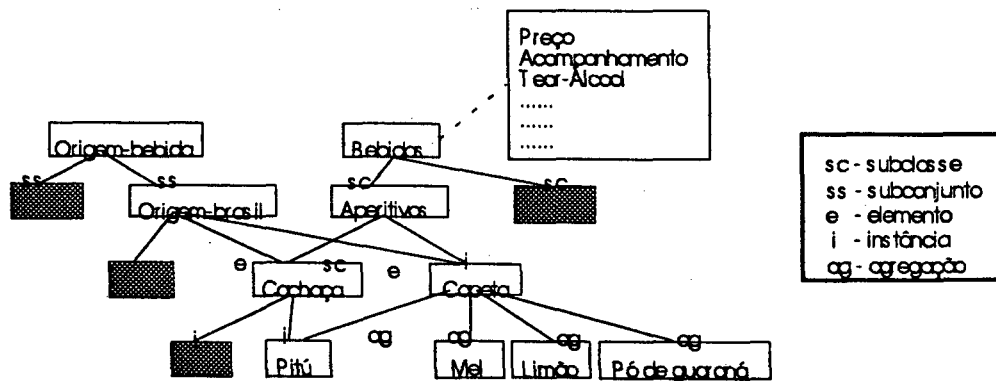


Figura 3.7 Exemplo de modelagem de conhecimento

De forma a estruturar os objetos, KOBRA suporta os conceitos de abstração de classificação/generalização, associação e agregação. Estes conceitos são incorporados ao modelo por meio de atributos especiais controlados pelo KBMS.

Cada "schema" pode se relacionar com os demais utilizando qualquer conceito de abstração. Então, classificação/generalização assim como associação ou agregação formam um grafo acíclico com raiz em um objeto definido pelo KBMS. Uma vez que cada "schema" pode ser um nó deste grafo, o KB pode ser visto como a superposição de tres grafos. O mesmo objeto pode representar uma instância ("instance") e um subconjunto ("subset") ao mesmo tempo. Na figura 3.7, o "schema" "Pitú" tanto é uma instância de "Cachaça" como elemento do conjunto "Origem-brasil".

Como já mencionado, aspectos comportamentais do objeto são representados por métodos. Estes, expressam operações, as quais podem modificar as propriedades dos objetos.

Os objetos possuem uma série de características que permitem sua estruturação e descrição de suas propriedades, mas algumas vezes é necessário que os atributos tenham uma descrição mais detalhada, que focalize e oriente o processamento executado nele. Para isso existem os aspectos. Existem oito tipos de aspectos básicos, além dos criados pelo usuário, são eles:

- a) "possible-values" - restringem o conjunto de valores de um atributo;
- b) "cardinality" - restringe o número de valores em um atributo;
- c) "restraint" - restringe os tipos de acesso a um atributo;
- d) "default" - contém valores default para um atributo;
- e) "comment" - descrição textual de um atributo;
- f) "demon" - liga um atributo a um ou mais métodos;
- g) "default-method" - contém um código "default" para um método;
- h) "diametric-reference" - utilizado em agregações.

KOBRA possui ainda dois outros conceitos de especificação de conhecimento operacional: "demons" e regras de produção. "Demons" são procedimento ligados a um atributo, os quais são automaticamente executados quando o atributo é manipulado. As regras de produção possuem raciocínio "para frente" (progressivo) ou "para trás" (regressivo) [RIC88], de acordo com a metodologia escolhida.

Os conceitos de abstração (classificação, generalização, associação e agregação) são utilizados para representar o conhecimento. Sendo, portanto, necessário alguns comentários, em [LEM92], a este respeito.

Classificação/Generalização

Uma classe pode ser, informalmente, comparada a um conceito que se aplica a um grupo de objetos. Assim, quando se diz que um objeto é uma classe, se está afirmando que os objetos pertencem a ele por este vínculo, possuem todas as características referentes ao conceito definido pelo objeto-classe. Em outras palavras, um conceito definido por um conjunto de características atribuídas individualmente a cada objeto pertencente à classe.

Existem dois tipos de vínculos entre uma classe e as que dela derivam. Primeiro é o vínculo entre classe e subclasse, onde esta última representa um conceito mais particular, hierarquicamente derivado do nível anterior. No exemplo da figura 3.5 o objeto "Cachaça" é representa um conceito mais específico que o modelado em "Bebidas", sendo portanto uma subclasse. Ou seja, "Cachaça" é um tipo de "Bebida".

O segundo tipo de vínculo acontece entre classe e instância. Neste caso, o objeto, hierarquicamente derivado, é visto como um elementar, não como um conceito mas como uma entidade, à qual se aplica o conceito definido pelas suas superclasses. O objeto "Pitú", por exemplo, é instância de "Cachaça", logo, se pode afirmar que: Pitú é uma Cachaça.

Classificação se refere ao agrupamento de instâncias sob uma classe, enquanto que generalização se refere ao agrupamento de subclasses sob uma classe. Os termos recíprocos são, respectivamente, instanciação e especialização.

Associação

Um conjunto ("set") no KRISYS, é um agrupamento de objetos que não necessariamente possuem alguma característica em comum, ou seja, não se pode dizer que o objeto "a" é um objeto "b". O fato de pertencerem a um mesmo conjunto implica que o todo possui características que os agrupam. Em resumo, ao contrário do que acontece com as classes, são os elementos que devem influenciar o funcionamento do conjunto. No exemplo, o objeto "Origem-brasil" possui uma série de propriedades que caracterizam as bebidas produzidas no Brasil, as quais são elementos do conjunto.

Agregação

Uma agregação representa a inserção de um objeto dentro de outro, ou seja, o objeto agregado faz parte do objeto agregante. Esta relação ocorre com frequência na modelagem de situações reais, onde os objetos são constituídos por uma série de partes. No exemplo, o objeto "Capeta" é constituído de diversas partes como, Pitú, Limão, Mel, Guaraná em Pó

Capítulo 4

PROPOSTA DE AMBIENTE DE INTEGRAÇÃO

Carcasse, tu trembles? Tu tremblerais bien devantage, si tu savais, où je te mène. Turenne

A citação acima tem o caráter de reforçar a importância deste capítulo. O qual trata da informação no Sistema de Manufatura podendo, então, ser considerado o centro deste trabalho. No primeiro item são analisados os requisitos para um ambiente de Engenharia Concorrente baseado na integração do Sistema de Manufatura. Em seguida é apresentada a proposta de um ambiente computacional destinado a auxiliar na integração das informações. Por fim são descritos os elementos que compõem o sistema.

4.1 Análise do problema

A música é a seleção de uma parte do espectro sonoro. Alguns ruídos devem ser sacrificados em função de outros. Na realidade, música é o equilíbrio entre os sons. Este equilíbrio é a harmonia que se desprende dos sons musicais, no modo como conciliam os sons opostos⁷. O antagonismo está presente em todas as coisas. Em um conjunto de atividades existe uma série de aspectos que se contrapõe, carregando consigo vantagens e desvantagens de um em relação aos outros. A melhor solução é aquela onde os aspectos vantajosos de cada opção estejam presentes. Harmonicamente presentes. De forma que a complementaridade seja usada para melhores resultados.

O Sistema de Manufatura (SM) tradicional pode ser traduzido por um conjunto de atividades sequenciais, executadas em áreas⁸ distintas de uma empresa. O objetivo destas etapas é definir particularidades de um produto, assim como procedimentos para sua produção. Esta prática pode levar a erros causados por desconhecimento de informações que, a princípio, seriam de uso "exclusivo" de outras áreas. Ao invés de compatibilizar informações, ou desenvolver atividades paralelamente, a Engenharia Concorrente traz para uma atividade específica informações que afetam o desenvolvimento do produto como um todo. Assim como na música, as etapas de desenvolvimento de um produto devem ser realizadas pela escolha de um aspecto ou outro das diversas atividades conflitantes, sacrificando ruídos para que sons mais limpos possam se sobressair.

⁷ Extraído de O Som e o Sentido de José Miguel Wisnik, editado pela Companhia das Letras/Círculo do Livro.

⁸ Por área de uma empresa entenda-se o departamento responsável por funções bem-definidas do Sistema de Manufatura. Por exemplo: Departamento de Projeto, Departamento de Produção, etc. Pode-se traçar um paralelo com as funções da manufatura descritas na seção 2.1.1.

As definições de Engenharia Concorrente apresentadas na seção 2.2 são voltadas para o projeto do produto. Segundo estas, a maior parte do custo é definido durante as atividades de projeto, sendo necessário concentrar os esforços e, por assim dizer, as informações (ver figura 2.7). Essa simplificação ocorre baseada no pressuposto de que quando um projeto sair da área de Processos de Engenharia estará isento de erros, estando as etapas de planejamento posterior praticamente prontas.

A experiência prática de Engenharia Concorrente, acompanhada na indústria, tem demonstrado que, quando utilizados grupos multifuncionais a nível de projeto, existem pendências que serão resolvidas somente no planejamento de processos ou escalonamento da produção. Estas decorrem da falta de informações atualizadas, ou ainda, porque sua solução pode ser bastante demorada. Problemas desse tipo ocorrem também com relação a processos de inspeção.

Na prática, Engenharia Concorrente deve prover a descentralização das informações, não só a nível de projeto (Processos de Engenharia) mas em todas as atividades do Sistema de Manufatura. No sentido de exemplificar o tipo de relação necessária entre as ferramentas são definidas, a exemplo do descrito na seção 2.2.1 algumas diretrizes importantes para um ambiente de Engenharia Concorrente. Estas, conjuntamente com as anteriores, servem de base para a especificação do sistema:

- a) informações sobre planejamento da inspeção devem estar disponíveis a nível de planejamento de processos. Isso implica em normas de qualidade assim como em procedimentos e sistemas de medição;
- b) informação de carga máquina e planejamento da produção devem estar disponíveis a nível de planejamento da produção minimizando problemas com sobrecarga de máquinas;
- c) informações sobre custos de serviços de terceiros são recomendadas tanto a nível de projeto como planejamento de processos e da produção.

A implantação de Engenharia Concorrente, baseada em grupos multifuncionais, centra o desenvolvimento de um projeto em torno de diversos especialistas. Estes mantêm o conhecimento sobre seu trabalho em seu cérebro, ou em algum tipo de "representação informal". Esta prática continua sendo centralizadora da informação gerando problemas quando a pessoa está ausente. Além disso as informações nem sempre estão disponíveis quando necessárias, acarretando em demora no desenvolvimento das atividades.

Este trabalho procura especificar as necessidades e propor metodologias de implementação de um ambiente computacional onde seja possível integrar as ferramentas computacionais que auxiliem a solução do problema descrito acima. De forma a minimizar estes efeitos, procurando resolver, basicamente, o problema da descentralização e tratamento da informação foi definido o modelo AVEC — Ambiente Aberto Visando Engenharia Concorrente.

4.2 AVEC — Ambiente Aberto Visando Engenharia Concorrente

As dificuldades descritas na seção anterior são decorrentes da concentração da informação em áreas e/ou pessoas. O uso de um ambiente computacional que dê suporte ao tratamento da informação, viabiliza sua descentralização. O ambiente computacional AVEC, aqui proposto, é baseado em conceitos de Manufatura Integrada por Computador (CIM). Assim, cada CAx (e.g. CAD, CAPP, CAM, etc.) é considerado uma ferramenta de auxílio ao Sistema de Manufatura como um todo.

Esta abordagem é complementar àquela descrita na seção 2.2.3, proposta por Terpenney [TER92] para ambientes de Engenharia Concorrente voltados para o projeto de produto. Três pontos são básicos para a sua implementação: melhoria na estratégia de projeto, auxílio à tomada de decisão e suporte à informação. Sob este enfoque, são tratadas as informações e previstas melhorias para a etapa de projeto do produto, procurando provê-lo das informações necessárias ao desenvolvimento de um produto sem inconsistências. Um ponto importante desta proposta é a generalidade em termos do tipo de auxílio pois ataca a

descentralização da informação. Normalmente, as propostas de Engenharia Concorrente são voltadas exclusivamente para as questões de manufaturabilidade e montagem.

Generalizando ainda mais esta abordagem, para o auxílio ao Sistema de Manufatura como um todo é necessário acrescentar às diretrizes propostas por Terpenny, um meio que suporte a informação de forma distribuída. Isto é, tornar a informação acessível a qualquer usuário do ambiente que a necessite. Em [TUI93], Afsarmanesh descreve os usuários de um ambiente de banco de dados distribuído heterogêneo como: "... tipicamente parte de um grupo onde outros dependem da disponibilidade de suas informações e necessitam saber como consultá-las." Como descrito na seção 3.3, esse tipo de sistema provê as facilidades necessárias para a implementação desta proposta.

Infelizmente, um sistema de banco de dados heterogêneo ainda não se encontra em fase de franca comercialização, não estando, portanto, disponível à maioria das empresas de manufatura. Assim sendo, definiu-se uma plataforma visando a efetiva implementação do ambiente. Em linhas gerais mantêm-se os requisitos básicos de um HDBMS, sendo adicionados alguns elementos de auxílio.

No sentido de especificar o ambiente, devem ser considerados os seguintes requisitos:

- a) o ambiente deve ser um sistema distribuído baseado no paradigma de cliente-servidor (ver seção 3.3). Neste enfoque, cada ferramenta é considerada responsável por prover serviços de análise e consulta às informações relativas a sua área de atuação;
- b) o grau de automatização das análises é proporcional à complexidade da tarefa. Ou seja, a ferramenta de análise pode ser tanto mais automática quanto mais simples for o processo de análise. Por exemplo, para análise de manufaturabilidade a ferramenta deve auxiliar o processista na tarefa de analisar a peça, mas não necessariamente realizá-la automaticamente;
- c) a independência de dados é uma característica essencial em sistemas distribuídos que deve ser garantida. Neste sentido, a troca de informações entre diferentes ferramentas deve ser realizada de modo a manter transparente sua estrutura interna. Da mesma forma, deve ser controlada a redundância das informações, ou seja, o mesmo dado repetido em mais de uma ferramenta;
- d) as informações sobre produto são manipuladas por praticamente todas as ferramentas, sendo necessário representá-las a partir de um modelo único de dados. Isto significa que tanto informações tecnológicas como geométricas devem ser representadas da mesma maneira, ou seja, dados complexos;
- e) a interação com o usuário num ambiente integrado deve ser realizada de maneira padrão, ou seja, a "interface" com usuário deve ser padronizada definindo uma forma comum de realizar troca de informações entre ferramentas. Além disso, facilidades como: projeto compartilhado, reuniões eletrônicas, correio e multimídia melhoram a eficiência no uso do ambiente;
- f) o ambiente deve ser configurável permitindo a inserção de novas ferramentas sem grandes traumas;

O Sistema de Manufatura, conforme descrito na seção 2.1, desenvolve atividades com diferentes meios, tanto em termos de ambiente de trabalho e cultura da área, quanto em relação ao significado dado a uma mesma informação. Na função de Processos de Engenharia, são manipuladas informações geométricas e tecnológicas, implicando em um grande volume de dados que devem, de alguma forma, ser associados. Além disso, a geração de malha e análise de elementos finitos implica em capacidade para processamento numérico, o que gera extensos arquivos. Nesta área, a informação é analisada do ponto-de-vista funcional e estrutural. Por sua vez, a função de Processo de Produção manipula informações a nível de chão-de-fábrica, ou seja: estado atual das máquinas, ferramental disponível, etc. As mesmas informações tecnológicas são analisadas como restrições para a produção. Em muitas ocasiões, informações são utilizadas no próprio ambiente de chão-de-fábrica, exigindo equipamentos dedicados para este fim. O ambiente distribuído, baseado no modelo cliente-servidor, adapta-se muito bem a esta realidade, pois mantém servidores de informação interligados, permitindo consultas por parte dos interessados. Desta forma, podemos considerar um sistema CAPP como sendo um servidor de informações sobre planejamento de processos e, ao mesmo tempo, uma ferramenta de análise, capaz de verificar o grau de manufaturabilidade do projeto de um produto. Esse enfoque organiza o caos, isolando problemas e simplificando o tratamento das informações.

Estando as informações agrupadas por área, em ferramentas providas de meios para responder a solicitações, é necessário que exista uma forma de manipulá-las externamente. O paradigma de orientação a objetos (ver seção 3.2) utiliza o conceito de troca de mensagens entre objetos para enviar ou requisitar serviços. Sob esse ponto de vista, cada ferramenta do Sistema de Manufatura é considerada um objeto e, portanto, troca mensagens com as demais. A troca de mensagens deve ser definida de maneira que todas as ferramentas possam entendê-la. Para tanto é utilizado um protocolo reconhecido por todas as ferramentas que implementa todas as requisições necessárias.

As informações sobre produto tem um papel especial no AVEC. Levando em consideração as atividades de "Gerenciamento Corporativo", "Fabricação de Produto" e "Marketing de Produto" (figuras 2.7 e 2.8 da seção 2.1.3), a grande maioria das informações que por ali fluem relacionam-se diretamente com o produto, tanto a nível de definição como de produção. Esta característica leva a considerar que o produto, ou melhor, os elementos mecânicos construtivos de um produto, sejam na realidade a informação central com a qual as demais se relacionam.

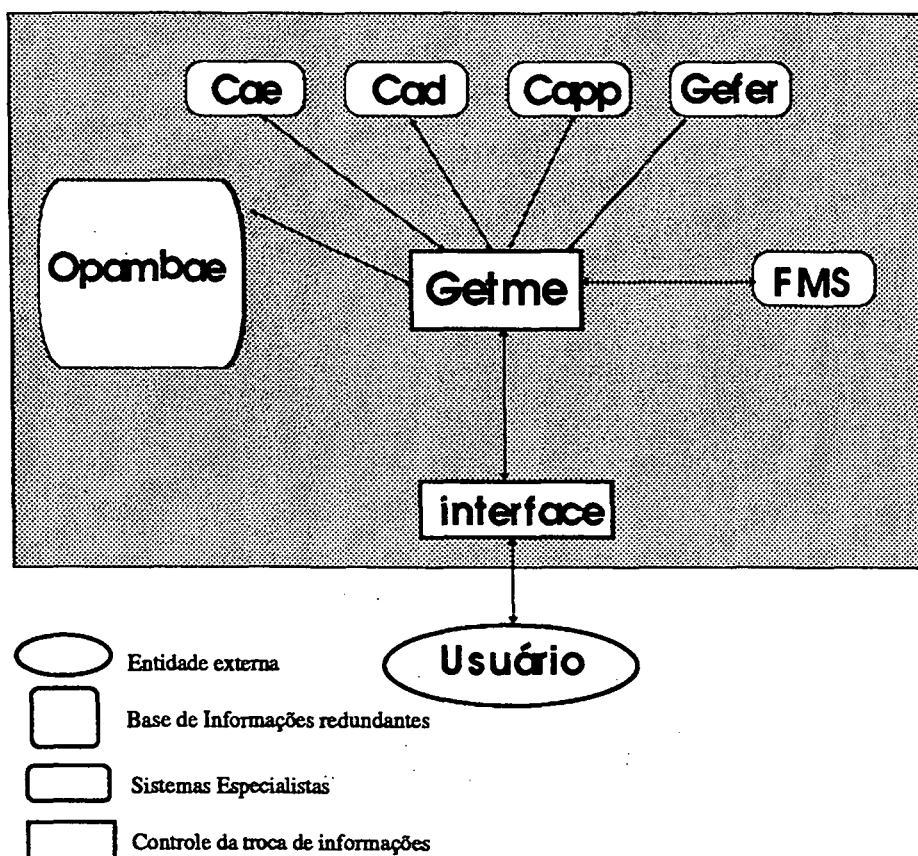


Figura 4.1 AVEC — Ambiente Aberto Visando Engenharia Concorrente

Partindo deste pressuposto, é induzida a utilização de uma representação única para modelar o produto, de forma a permitir a relação das características geométricas com as tecnológicas e seus meios de produção. A utilização de Tecnologia de "Features" (ver seção 3.1) permite definir o produto como um conjunto de elementos funcionais, reconhecidos em todo o Sistema de Manufatura. Cada um destes elementos, ou "features", armazena informações relativas ao seu próprio uso durante todo o ciclo de vida de um produto. Uma discussão detalhada sobre a modelagem das informações é feita no capítulo 5.

Usuários de computadores estão habituados a ser colocados diante de uma mudança de hábito, quando uma nova "interface" surge para ser utilizada. Aquilo que parece um "nó no cérebro" é perfeitamente evitável, se um padrão de "interface" for utilizado. "Motif" e "OpenLook" são dois desses padrões que facilitam a utilização de um novo "software", isto porque qualquer "interface" desenvolvida sob esse padrão terá elementos iguais sendo utilizados para realizar a mesma função. Por exemplo, a forma de gravar um arquivo é a mesma para todos. A utilização destes conceitos em um ambiente integrado é

fundamental para o efetivo uso das facilidades. De outra forma, a dificuldade em utilizar algumas tantas "interfaces" diferentes desestimulará seu usuário.

Resumindo o exposto acima, é possível afirmar que o AVEC é um ambiente composto por bases de conhecimento e sistemas especialistas trabalhando em um sistema heterogêneo distribuído, trocando informações. A figura 4.1 apresenta a idéia básica do Ambiente Aberto Visando Engenharia Concorrente.

Em ambientes deste tipo, a comunicação entre objetos é realizada através da troca de mensagens, garantindo desta forma a independência entre eles. Três objetos especiais tem funções específicas de suporte ao ambiente. O objeto Getme — Gerenciador de Troca de Mensagens —, suporta o fluxo de informações, mantendo transparente o redirecionamento de mensagens trocadas entre os objetos. O objeto "Opambae"⁹ é uma base de conhecimento compartilhada, utilizada como elemento aglutinador de informações redundantes no ambiente. Por exemplo, informações tecnológicas do tipo acabamento-superficial ou tolerância são utilizadas a nível de sistemas CAD e CAPP e, por este motivo, são modeladas no "Opambae", além de seus respectivos locais de origem. Neste objeto, as informações do Sistema de Manufatura são genericamente representadas. O objeto "Interface" é usado para uniformizar as "interfaces" para usuários. Deve suprir as necessidades do ambiente descritas no item "e" da especificação acima.

A seguir, os elementos principais sobre os quais se baseia esta proposta serão descritos com mais detalhes.

4.3 O controle das informações — Getme

O Getme — Gerenciador de Troca de Mensagens — tem a função de suportar o fluxo de informações entre os objetos. A troca de mensagens entre objetos deve ser realizada de forma transparente a nível de endereçamento, ou seja, para enviar uma mensagem não deve ser necessário conhecer o endereço da máquina, assim como do processo no qual um serviço é oferecido.

Todo o sistema de mensagens está baseado em um protocolo suficientemente genérico a ponto de suportar as diferenças existentes entre ferramentas. Supondo sua existência, é necessário prover cada objeto do ambiente com um serviço que manipule as mensagens, enviando-as ao destino. Conforme mostrado na figura 4.1, o Getme é o elemento central do ambiente, para o qual todas as mensagens são enviadas e a partir de onde são redirecionadas.

O objeto Getme é responsável por controlar ainda algumas funções de permissão de acesso de usuários e ferramentas a determinadas informações.

4.3.1 Requisitos para um protocolo

Um protocolo de comunicação entre objetos deve prever as necessidades de comunicação do ambiente como um todo, considerando que os serviços que cada objeto oferece são variáveis. Além disso, os objetos também são variáveis, uma vez que o sistema deve ser configurável. A definição de um protocolo do porte exigido foge aos objetivos deste trabalho, mas os pontos principais podem ser definidos desde já.

Basicamente, as mensagens devem manipular objetos e disparar seus métodos públicos. Estes métodos realizam tarefas internas e retornam resultados. Cabe lembrar que todos os elementos do ambiente são modelados como objetos, inclusive as ferramentas CAx. As características mais comuns que o protocolo deve prover são as seguintes:

⁹ "Opambae", significa todos os objetos na língua Guarani, utilizada pelos índios da região litorânea de Santa Catarina durante o Século XVIII.

- a) a manipulação de objetos, seja ele uma informação que caracteriza uma peça, ou o estado atual de uma máquina, deve ser passível de controle por parte de uma ferramenta que o necessite. Portanto, o protocolo deve suportar criação, eliminação e modificação de um objeto específico;
- b) os métodos são a porta de entrada para o interior de um objeto. Sendo possível dispará-los, todos seus serviços públicos estarão disponíveis. O protocolo deve prover serviços que permitam a verificação da consistência entre pedidos de disparo de métodos e os existentes nos objetos;
- c) o retorno dos resultados deve ser realizado na forma de mensagens. O objeto que solicitou o serviço recebe o resultado como um evento, ou seja, não necessita ficar inativo, aguardando o retorno da informação, a não ser que isto seja imprescindível.

Muitos outros requisitos devem ser identificados para que possa ser possível definir a troca de mensagem entre os objetos do ambiente. Mas, a partir destes itens básicos, a maior parte dos problemas estarão resolvidos.

Uma série de protocolos padronizados poderiam ser utilizados como base para a troca de mensagens. A Especificação de Mensagens de Manufatura (MMS) é um bom exemplo destes padrões. Definida como uma camada de aplicação do modelo Protocolo de Automação da Manufatura (MAP), tem como característica seguir o modelo servidor-cliente e ser uma especificação orientada a objetos para a troca de mensagens entre equipamentos de manufatura inteligentes, como robôs ou máquinas CNC. Da mesma forma que na abordagem utilizada neste trabalho, o MMS representa cada equipamento como sendo um objeto do sistema.

4.3.2 Serviços de mensagens

Como descrito anteriormente, o meio de troca das mensagens deve ser suficientemente flexível para ser utilizado em uma plataforma distribuída e heterogênea. Além disso, ele próprio deve estar igualmente distribuído, aumentando a segurança do sistema, uma vez que desta forma o sistema como um todo não depende apenas de uma máquina.

O Getme é implementado como uma camada de rede em cada ferramenta do ambiente. Isto é necessário para descentralizar o controle das informações. A figura 4.2 representa essa abordagem. Na prática, esta implementação não difere da idéia apresentada na figura 4.1, uma vez que a camada de rede de cada ferramenta pode ser considerada como um elemento único, ainda que distribuído.

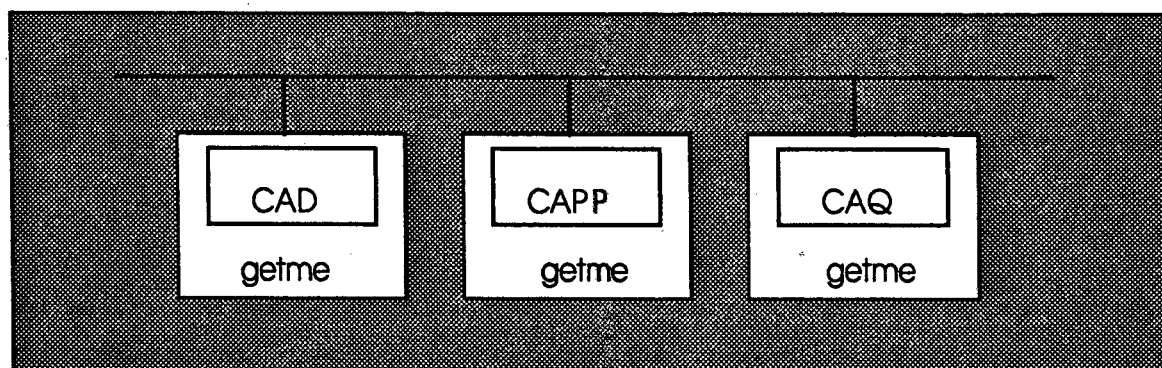


Figura 4.2 Modelo de implementação do Getme

O Getme é implementado como uma biblioteca de funções que, utilizada por cada uma das ferramentas, disponibiliza os serviços oferecidos pelo protocolo. Em geral, todos os sistemas, mesmo os comerciais, apresentam alguma linguagem de desenvolvimento que permite adaptá-los para uma atividade

específica. Estas ferramentas podem ser "integradas fortemente"¹⁰ ao sistema, utilizando o protocolo internamente. Nos casos de ferramentas para as quais não for possível utilizar esta facilidade, a saída com resultados deve ser lida por um programa tradutor, o qual decodificará a informação, disponibilizando-a via troca de mensagens. Este é um caso de "integração fraca" de sistemas. A figura 4.3 exemplifica esta abordagem.

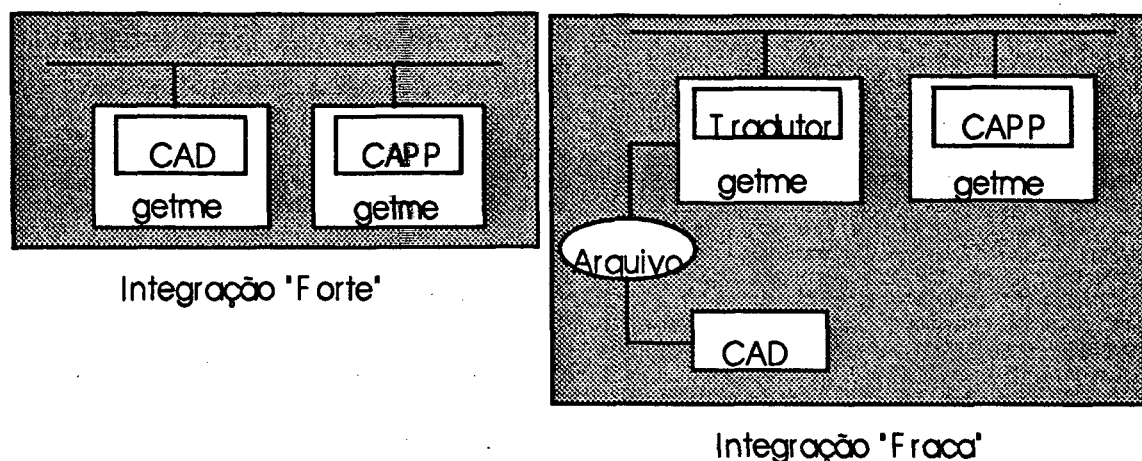


Figura 4.3 Exemplo de integração de ferramentas

4.4 Informações redundantes — "Opambae"

O Sistema de Manufatura abrange um largo espectro de informações mas, do ponto de vista de integração para Engenharia Concorrente, estas podem ser analisadas a partir das ferramentas que as manipulam. Neste sentido, as informações se concentram em torno da definição de um produto para todas as atividades desenvolvidas durante seu ciclo de vida. Como discutido em itens anteriores, uma série destas informações são utilizadas concomitantemente por mais de uma ferramenta do ambiente. Por esse motivo, a redundância deve ser manipulada de tal forma que, para todas as bases locais de ferramentas, o conteúdo da informação seja o mesmo. É importante observar que considera-se como uma informação redundante, toda aquela manipulada por mais de uma ferramenta, seja para definição ou consulta.

Na abordagem utilizada neste trabalho, o Sistema de Manufatura foi modelado em uma base de conhecimento (ver seção 3.4.3) de forma genérica, englobando, macroscopicamente, os aspectos de interesse da Engenharia Concorrente. Sua principal função é servir como elemento disseminador das informações, tanto para aquelas redundantes, recebendo modificações e atualizando as cópias, ou como fonte de consulta para o ambiente como um todo.

O objeto "Opambae" tem as mesmas características dos demais objetos do ambiente, ou seja, é manipulado através da troca de mensagens. O modelo representado nesta base de conhecimento é igualmente orientado a objetos, sendo discutido no capítulo 5.

O controle das redundâncias baseia-se no princípio que, ao alterar um conteúdo, a ferramenta deve fazê-lo também no objeto "Opambae" e este se encarregará de atualizar as demais cópias da informação. A figura 4.4 apresenta essa abordagem.

¹⁰ Segundo definições do projeto ESPRIT [CAM92], integração "Forte" ocorre quando existe uma aplicação que recebe informações diretamente, via troca de mensagens. Integração "Fraca" é realizada através de arquivos, sendo portanto passível de interpretação.

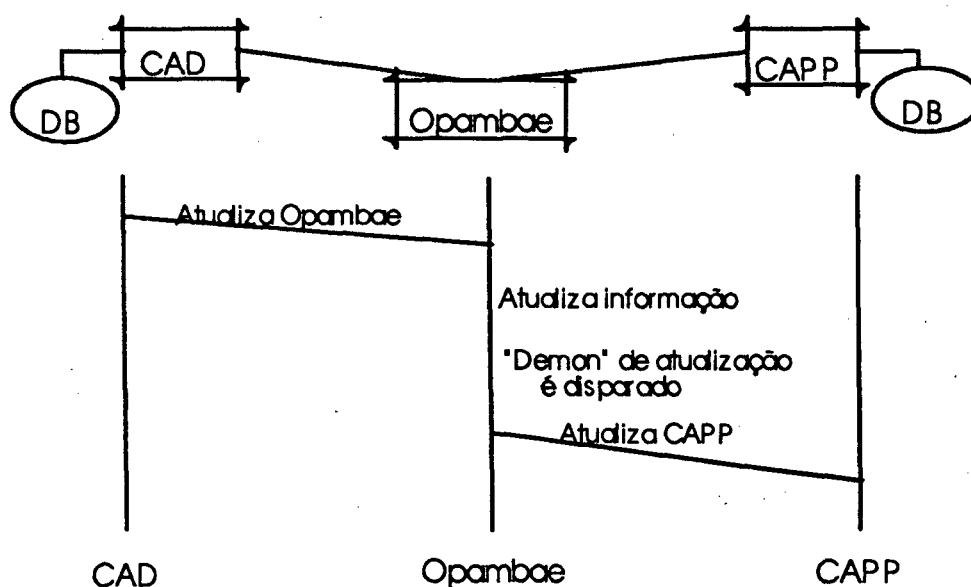


Figura 4.4 Atualização de informação redundante

Algumas considerações importantes a respeito da atualização:

- a) o tratamento de colisões de acesso deve ser manipulado pelo objeto Getme. Enquanto uma ferramenta estiver manipulando uma informação, somente acessos para leitura devem ser permitidos. Essa situação deve se manter até que todas as cópias tenham sido atualizadas;
- b) a implementação do serviço de atualização é feita através de "demons" (seção 3.4.5) nos atributos com redundância.

4.5 "Interface" com o usuário

A definição de uma "interface" padrão para todas as ferramentas que compõem o ambiente é encorajada, embora, a medida em que se utilizem ferramentas de diferentes fornecedores, isto seja uma tarefa difícil. Ainda assim, o objeto "Interface" tem um papel de destaque neste ambiente, pois, a partir dele, todas as interações de Engenharia Concorrente são realizadas.

Sua definição carece de estudo mais aprofundado, mas é possível definir algumas de suas características:

- a) as interações com o usuário a nível de pesquisa de informações, assim como a análise de um produto devem ser padronizadas, pois, a partir de qualquer das ferramentas, é possível acionar o mesmo serviço.
- b) informações sobre ferramentas, por exemplo, necessitam de um misto de informações geométricas e descritivas para serem bem entendidas. Por esse motivo, a interface de consulta de informações deve ser bastante abrangente em termos de facilidades de apresentação de resultados;
- c) é comum em ambientes de Engenharia Concorrente uma decisão, em qualquer área, depender da informação de diversas outras áreas. No sentido de facilitar o trabalho destes grupos, o ambiente deve prover ferramentas tipo quadro-negro, onde informações possam ser adicionadas por diversas pessoas a medida que o problema vai sendo analisado;
- d) o ambiente como um todo deve ser consistente, isso é, um procedimento de interface deve ser o mesmo para todas as ferramentas, facilitando assim seu uso;
- e) as ferramentas de correio eletrônico, assim como pequenas reuniões via terminal, aumentam em muito a eficiência do ambiente de trabalho e facilitam a tomada de decisões.

A tecnologia de multimídia hoje existente permite prever uma série de novas facilidades para a integração da manufatura. Seu uso tornará a transmissão de informações mais eficiente em todos os níveis do Sistema de Manufatura, já que, através da manipulação de imagem e som, tarefas como montagem, manutenção, preparação de máquinas ou pré-setagem poderão ser mais facilmente desenvolvidas. Estas facilidades associadas a hipertexto facilitam o aprendizado e transferência de conhecimento.

4.6 Modelo de ferramentas — CAX

As conhecidas ferramentas de auxílio às atividades de manufatura (CAD, CAPP, CAM, etc.) devem ser modificadas de forma a satisfazer as necessidades de um ambiente de Engenharia Concorrente. Como já descrito, cada ferramenta, além de realizar as tarefas usuais, deve manter uma base de informações pertinente a sua especialidade.

A função da base de informações é permitir consultas sobre assuntos que sejam de interesse em outras áreas. Por exemplo, o CAPP deve dispor de uma base de informações acerca das ferramentas disponíveis e suas características, assim como a relação entre processos de fabricação e máquinas e ferramentas. Essas informações podem ser muito úteis quando da definição de um detalhe da peça.

A ferramenta deve suportar análise de uma peça a respeito de sua especialidade. Ou seja, qual a possibilidade de uma peça ser realizada considerando as facilidades e conhecimento disponível a respeito.

4.7 Plataforma para a implementação

O modelo deve ser implementado sobre uma plataforma de rede que ofereça portabilidade. O protocolo TCP-IP tem se mostrado um padrão de fato, estando implementado para uma vasta gama de "hardwares" e sistemas operacionais utilizados no ambiente de manufatura. A partir dos serviços de RPC (Remote Procedure Call - Chamada Remota de Procedimentos), utilizando "sockets", é possível a implementação do protocolo, a ser definido, que suportará a troca de mensagens.

Conforme já discutido na seção 3.4, um Sistema Gerenciador de Base de Dados (DBMS) tradicional não é adequado para manipular informações de engenharia. Estes DBMS foram projetados para aplicações comerciais, não sendo eficientes para modelar as informações geométricas e estruturas de dados com tamanho variável, facilmente encontráveis na modelagem das informações do Sistema de Manufatura. Além disso, uma aplicação deste tipo necessita de uma semântica particularmente rica e facilidades para mudanças na estrutura da base.

A representação das informações deve ser executada em um sistema que permita o uso de conceitos de abstração, como classificação/generalização, associação e agregação. Além disso, métodos, "demons" e regras são também importantes uma vez que cada ferramenta de auxílio deve realizar inferências. Por prover esses recursos, o objeto "Opambae" foi desenvolvido sobre um Sistema Gerenciador de Base de Conhecimento orientado a objetos chamado KRISYS (seção 3.4.1), desenvolvido na Universidade de Kaiserslautern, Alemanha.

A implementação do objeto "Interface" merece um estudo aprofundado em técnicas de padronização de "interfaces", assim como conceitos de hipertexto e multimídia.

Capítulo 5

MODELAGEM DA BASE DE INFORMAÇÕES

The next generation of CAD/CAM systems must incorporate feature-based product development methods and shared databases. Olling - Chrysler Corporation. [OLL91]

A apresentação da modelagem da base de informações segue a ordem lógica deste trabalho. No capítulo 2 foram discutidas as informações trocadas no Sistema de Manufatura. Naquele momento foi utilizado IDEF₀ para representá-las, permitindo identificar os pontos em torno dos quais as informações se concentram. A partir da análise das necessidades de auxílio à Engenharia Concorrente, no capítulo 4 foi apresentada uma proposta de ambiente computacional que auxilia no controle e manipulação destas informações. Neste capítulo, é definido o modelo de representação das informações para este ambiente.

O modelo descrito a seguir foi implementado sobre um Sistema Gerenciador de Base de Conhecimento (KBMS) chamado Krisys. Este sistema tem suas principais características descritas na seção 3.4.1. No decorrer deste capítulo uma série de exemplos serão descritos utilizando conceitos empregados pelo sistema Krisys. Para auxiliar o entendimento, no anexo 2 é apresentado uma descrição dos princípios básicos de modelagem e dos gráficos utilizados para representar as relações entre os objetos.

5.1 Modelagem das informações no Sistema de Manufatura

Sistemas de computadores constituem um modelo do mundo real sobre o qual estão contidas informações específicas, pertinentes a uma aplicação particular. Na tentativa de modelar o mundo real, utiliza-se alguma forma de abstração para que seja possível organizar as informações da forma desejada. A abstração permite que alguns detalhes de um problema sejam desconsiderados em função de outros, cuja visão será mais elucidativa para a aplicação desejada. O uso de ferramentas de apoio como IDEF₀ possibilita, através de critérios bem definidos, identificar as informações de interesse, assim como a relação entre elas. De forma a representar as informações, é utilizado o paradigma de orientação a objetos, descrito na seção 3.2.

Na seção 4.2, apresenta-se uma análise do fluxo de informações no Sistema de Manufatura. Nesta o produto é considerado como elemento básico, sobre o qual se aplicam as transformações. Partindo do mesmo pressuposto, define-se o comportamento da informação a ser modelada.

Um produto é definido por sua utilidade. A especificação de produto, antes de mais nada, define uma função, sendo que a partir daí as características acessórias são apontadas. Definido o ponto base, são examinadas as condições para que o produto atinja os objetivos, considerando seus requisitos de qualidade, processos de fabricação disponíveis, etc.

Sintetizando estas características, chega-se às seguintes classes de informações no Sistema de Manufatura (figura 5.1):

- a) funcional: informações relativas a projeto, onde estão descritas as características de forma e funcionalidade do produto;
- b) tecnológica: contém dados tecnológicos em geral, como por exemplo, tolerância, acabamento superficial, material da peça, etc.;
- c) facilidades: contém informações sobre os recursos do chão de fábrica para a produção;
- d) enlace: informações que unem as demais gerando uma nova informação.

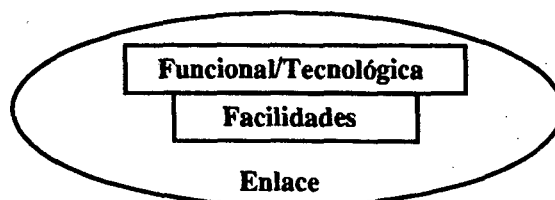


Figura 5.1 Classes de informação

A representação das informações é feita conforme o paradigma de orientação a objetos, sendo que seus principais objetos estão descritos na figura 5.2. O objeto Opambae é a raiz de todos os objetos da modelagem. A partir de seus métodos é controlada a instanciação¹¹ de objetos na base. Sua descrição encontra-se na seção 5.5.

Como visto na seção 3.2, objetos são descritos através de seus atributos e métodos. A descrição da modelagem, apresentada neste capítulo, priorizará a relação entre os objetos em detrimento de sua descrição interna. Esta pode ser consultada na listagem dos principais objetos contida no anexo 1.

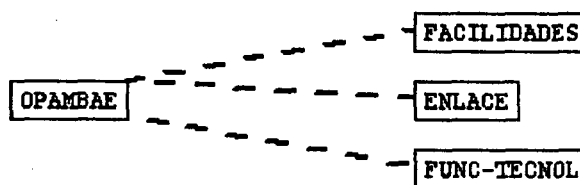


Figura 5.2 Objetos que representam as informações

5.2 Informações funcionais e tecnológicas

A partir das informações funcionais e tecnológicas é possível identificar as características e possibilidades de uso de uma peça. Por motivos óbvios, a representação de informações, seja ela qual for, deve ser feita de tal forma que seu uso seja otimizado, isto é, um número mínimo de objetos deve representar o máximo de informação. Assim, não é conveniente armazenar informações sobre uma peça como um elemento único, não reusável. Optou-se então por representá-la separando-a em suas funções localizadas.

A figura 5.3 apresenta o projeto simplificado de um eixo para polia esticadora de um sistema de transmissão por correia plana. Nesta peça é possível identificar dois elementos funcionais básicos. O conjunto de apoios de rolamento é usado para absorver esforços radiais e o assento da polia tem a função de montagem e ajuste. Cada função macro identificada na peça é chamada grupo.

¹¹ Instanciar um objeto significa criar uma instância de uma classe. Como visto na seção 3.2, uma instância possui valores associados a seus atributos.

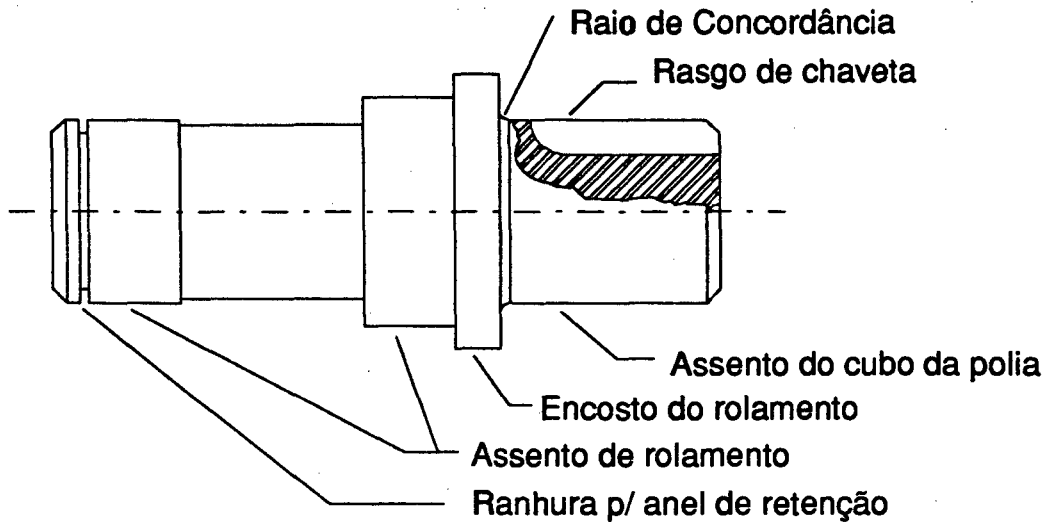


Figura 5.3 Eixo para polia esticadora

5.2.1 Grupos

Grupo é definido como um elemento construtivo que possui uma funcionalidade característica em um domínio de aplicação conhecido. Na figura 5.3 são identificados dois "grupos" principais:

- 1) composto pelos assentos de rolamento, ranhura para anel de retenção e encosto do rolamento. O conjunto tem a função, já descrita, de suportar esforços;
- 2) os elementos de apoio do cubo da polia, o rasgo de chaveta e o chanfro possuem a função de suportar a polia esticadora.

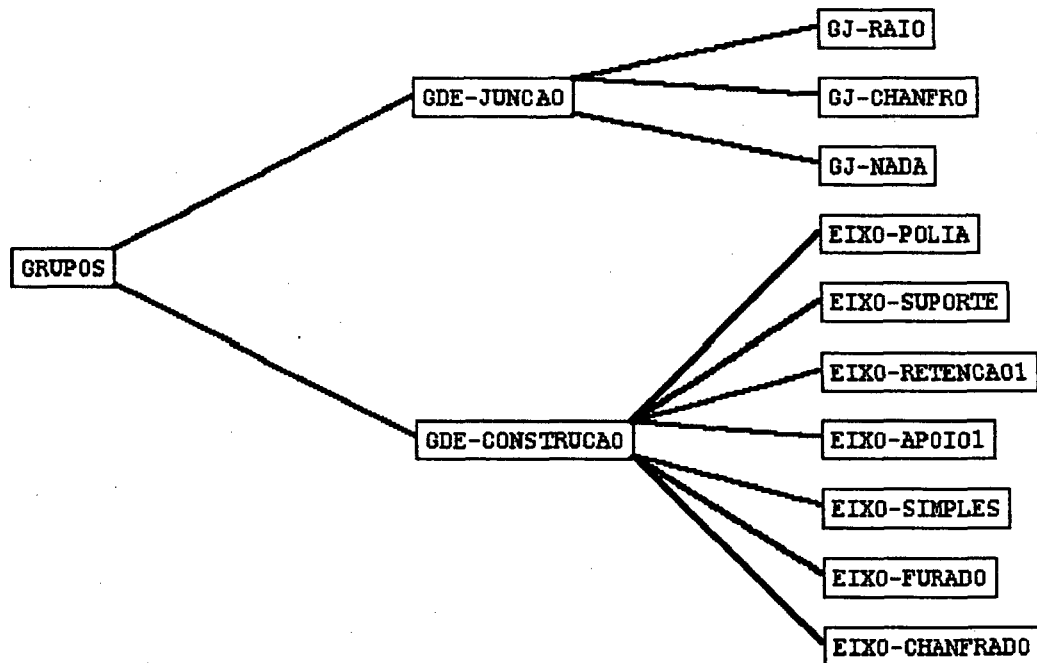


Figura 5.4 Classes de grupos

A união de dois grupos também deve ser considerada durante a representação de uma peça. Considerando o projeto da polia esticadora, a junção é feita através de um raio de concordância visando a redução de tensões. Este objeto não faz parte de nenhum dos dois grupos uma vez que tem uma função complementar a dos dois elementos. Por esse motivo, a união dos grupos deve ser modelada separadamente.

Os grupos são classificados conforme a figura 5.4. Os "de construção" são definidos a partir da função de projeto, expressando uma necessidade funcional conhecida e preferencialmente repetitiva em um domínio de produtos restrito. Os "de junção", por sua vez, são utilizados para identificar o local da união e representar um elemento que une as peças, quando isto for necessário.

A definição de um Grupo parte da cultura de projeto. Da análise destas necessidades é construída uma biblioteca com os grupos definidos. Os procedimentos de alteração desta biblioteca de grupos devem seguir normas bem definidas para que informações não sejam perdidas ou mal utilizadas.

O exemplo da figura 5.3 foi construído utilizando as subclasses do objeto GRUPOS. Os grupos de construção EIXO-POLIA e EIXO-SUPORTE, além do grupo de junção GJ-RAIO, pertencentes a biblioteca, foram instanciados para a representação da peça. Estes objetos carregam consigo informações que auxiliam o projetista na hora de compor a peça. O EIXO-SUPORTE, por exemplo, é um grupo específico para utilizar com rolamentos como apoio para uma carga em balanço, portanto, possui como atributo uma tolerância de concentricidade entre os dois assentos de rolamento. Ao instanciar um grupo EIXO-POLIA o projetista deve poder dimensionar a chaveta em função do esforço que o eixo deve transmitir à polia. Estes procedimentos de cálculo estão representados no objeto grupo. Os objetos estão descritos no anexo 1, sendo que a implementação da representação é apresentada na seção 5.5.

Um grupo é constituído por um conjunto de "features". Sua estrutura está baseada na agregação de informações de um elemento básico — primitiva — formador do corpo do grupo, com outros sobrepostos — detalhes — que conferem uma função específica ao conjunto. A união das informações contidas em cada "feature" caracteriza um elemento construtivo completo. Através dele estão disponíveis informações de diversas áreas do processo produtivo. A figura 5.5 aponta estes objetos no exemplo da polia esticadora.

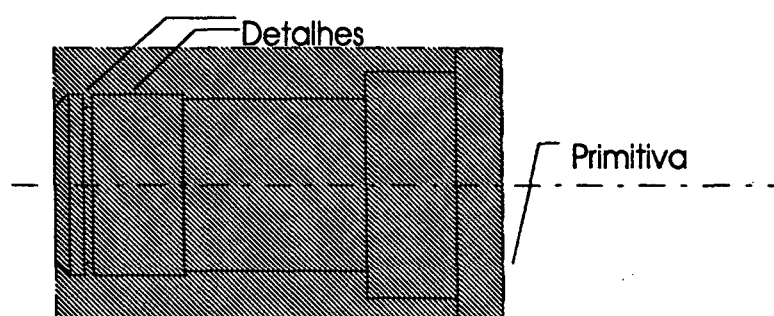


Figura 5.5 Primitiva de um grupo de construção

Utilizando-se os conceitos de orientação a objetos, modelou-se a transferência de características comuns, representadas a nível de grupo para as "features". Essa prática torna possível a disseminação de informações hierarquicamente. Por exemplo, após definido o material do qual uma peça é composta, todos os grupos e "features" que a compõe receberão automaticamente estes valores.

Por definição, a "feature" primitiva de um grupo engloba totalmente as "features" detalhe (figura 5.5). Esta característica é importante para o dimensionamento externo da peça como um todo, simplificando a análise de manufatura.

O elemento peça não é representado como uma informação funcional, uma vez que engloba várias funcionalidades diferentes, recebendo cada função diretamente das "features" e grupos. As peças, assim como todas as informações que unem funcionalidade, fazem parte da classe Enlace. Na seção 5.4 é apresentada a modelagem para o objeto produto.

A utilização de Tecnologia de "Features", como elemento básico para a representação da informação sobre o produto, é fundamental para a integração do Sistema de Manufatura. Esta prática gera um elemento de ligação entre informações que, em princípio, são heterogêneas, como por exemplo processos de inspeção e ferramentas de usinagem para um determinado processo de fabricação. Na seção 3.1 é apresentada uma longa discussão sobre as diferentes abordagens utilizadas na implementação desta tecnologia. Como visto naquela ocasião, o conceito de "features" ainda não é um consenso na comunidade científica.

5.2.2 "Yet another feature definition"¹²

No capítulo 4 são definidos requisitos de representação das informações no Sistema de Manufatura. Partindo deste princípio, propõe-se o uso de Tecnologia de "Features" como uma "chave de acesso" para as informações relativas ao produto. Ou seja, através deste elemento básico uma vasta gama de informações do ciclo de vida de um produto torna-se disponível aos usuários do AVEC, anteriormente descrito.

Como relatado na seção 3.1, ao longo do tempo o conceito de "features" evoluiu de forma a permitir a agregação das informações necessárias à integração do processo de manufatura. Por este motivo existem hoje controvérsias a respeito da forma de utilizar "features". Um caso já clássico é a diferença de abordagem entre reconhecimento de "features" e projeto com "features". A primeira corrente reza que se deve projetar a partir de um CAD convencional e após extrair as "features". A segunda afirma que todo o desenvolvimento do projeto deve ser feito a partir de uma biblioteca de "features". Outra celeuma é a questão da criação de novas "features" "on-the-fly", ou seja, se o projetista necessita de uma nova "feature", inexistente na biblioteca, deve ser permitida sua criação ou, pelo contrário, o trabalho deve ser realizado apenas com o que existe na biblioteca.

A literatura existente é bastante conflitante em uma série de questionamentos resumidos a seguir:

- a) as "features" são um conjunto finito de elementos?
- b) projeto por "features" ou reconhecimento de "features"?
- c) a união de duas "features" resulta em uma nova "feature"?
- d) é possível definir uma "feature" a qualquer tempo?
- e) uma "feature" é única para um mesmo elemento funcional, ou existem vários tipos (forma, produção, forjamento, tolerância, etc.)? Ou ainda, uma "feature" de projeto pode ser utilizada para planejamento de processos?
- f) "features" são cascas ou tem volume?
- g) uma "feature" pode ser tratada como um objeto?

As diferentes abordagens para o uso de "features" no Sistema de Manufatura visam, basicamente, modelar as informações de produto de forma integrada. A dificuldade está em fazer convergir as diferenças existentes entre o projeto (composição de funcionalidade) e a produção¹³ (decomposição de material). Com relação ao projeto, as "features" tem a função de representar funcionalidade através de sua forma e informações encapsuladas. Em termos de produção, o objetivo principal é armazenar informações relativas a execução da peça, como: definição das operações, seqüenciamento, escolha das ferramentas, etc. A principal diferença está na forma de manipular a informação. Ao utilizar uma "feature" única para todas as atividades do Sistema de Manufatura, esta deve servir de "chave de acesso" para todo o espectro de informações. Ou, caso contrário, se não existe relação entre "features" em projeto e em produção, então os dois elementos devem ser integrados sem perda de informações. Na seção 3.1.4 é apresentada uma revisão bibliográfica sobre o assunto.

A observação do esquema de produção na indústria faz crer que a funcionalidade é uma definição proveniente das atividades de projeto, sendo menos importante para as demais áreas do processo produtivo. Ou seja, a nível de produção a funcionalidade é um aspecto complementar, pois na geração do processo de fabricação são necessárias informações sobre a forma além das informações tecnológicas embutidas em cada elemento construtivo¹⁴. Conclusão análoga é possível para as atividades de engenharia de qualidade. A figura 5.6 descreve estas implicações.

¹² "Yet Another Compiler-Compiler" — YACC é um "software" clássico no ambiente UNIX, desenvolvido por Ken Thompson, e utilizado no desenvolvimento deste trabalho.

¹³ Por produção entenda-se a parte do ciclo de vida do produto após o projeto e análise de engenharia, englobando planejamento, execução e controle.

¹⁴ Por elemento construtivo entenda-se uma parte do produto dotada de função. No sentido hierárquico utilizado neste trabalho, este pode ser tanto uma peça como um grupo ou uma "feature".

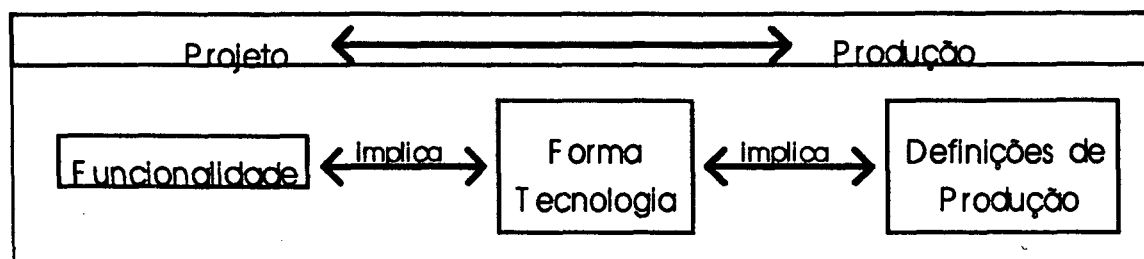


Figura 5.6 Relação entre "features" em projeto e produção

Uma "feature" existe para todo o ciclo de vida do produto, uma vez que é definido durante o projeto e utilizada nas atividades subsequentes. A questão está na maneira como cada fase do Sistema de Manufatura filtra as informações de um mesmo elemento. Uma situação similar é encontrada quando diversas pessoas, com diferentes formações, analisam uma mesma questão. Cada uma destas concluirá diferentemente, partindo de sua cultura própria. O elemento é um só, suas características são as mesmas, mas a importância de cada uma delas é relativa, dependendo do ponto-de-vista adotado.

Partindo deste princípio, a "chave de acesso" para as informações do ciclo de vida de um produto deve ser definida pelas informações de funcionalidade, sendo passível de análise considerando diferentes pontos de vista. Na prática, uma única "feature" possui todas as informações necessárias. O acesso a esta informação é realizado por consultas a partir das características referenciais para cada atividade do Sistema de Manufatura. A figura 5.7 representa esta abordagem. Nesta cada usuário (x, y, z) "olhando" para uma mesma "feature" através de um filtro próprio, encontra os atributos que lhe interessa. O reconhecimento destas características é realizado durante a atividade de identificação e classificação de "features".

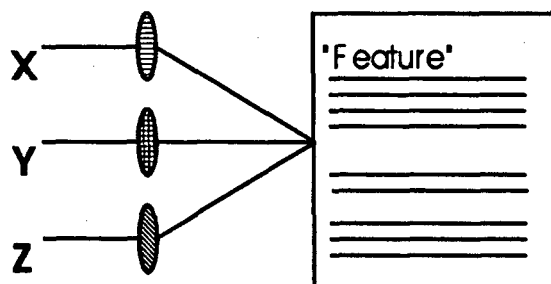


Figura 5.7 Uma "feature" vista sob vários aspectos

Voltando ao exemplo da figura 5.3, cada elemento construtivo pertencente a um grupo é uma "feature", portanto assentos de rolamento, ranhura para anel de retenção, encosto do rolamento, apoio do cubo da polia e rasgo de chaveta possuem funções específicas que as caracterizam. Por exemplo, um assento de rolamento, além de ser um cilindro, caracteriza-se por possuir tolerâncias, acabamento superficial e material, todos previamente especificados por normas apropriadas. Esse conjunto de informações compõe uma "feature". Como se pode notar, uma "feature" contém informações relativas ao próprio elemento, não conhecendo relações com as demais "features". Como detalhado na seção 5.4.1, este enfoque é utilizado em todos os níveis de representação do produto.

A criação de uma "feature" implica em grande quantidade de informações distribuídas em diversas áreas. Conseqüentemente, uma nova "feature" deve ser precedida por um estudo de suas características que englobe todo o ciclo de vida dos produtos para os quais representará informações. Todo este processo é lento e complexo, tornando sua prática desaconselhada. Alternativamente, dado um domínio conhecido de produtos, é possível identificar as "features" características que permitem a modelagem do conjunto. Esta abordagem tem a vantagem de gerar padronização nos elementos construtivos, incrementando a produtividade e a qualidade.

Todos os elementos construtivos existentes possuem volume, sendo compostos por um conjunto de superfícies. Da mesma forma, uma "feature" é composta por superfícies que representam este volume. Na figura 5.8, a "feature" furo é decomposta nas superfícies que a compõe. É importante observar que uma superfície não tem somente a função de dar corpo à "feature", permitindo também representar informações

do tipo tolerâncias de restrição, acabamentos diferenciados, etc. Na seção 5.2.4 a modelagem para superfícies é discutida com mais detalhes.

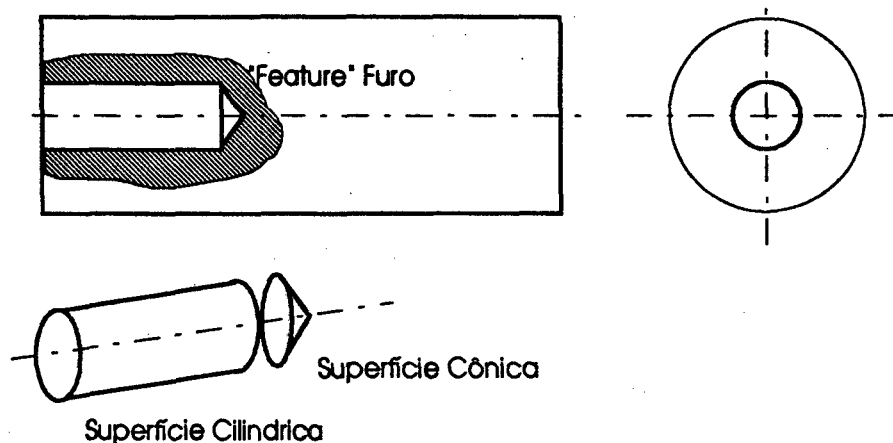


Figura 5.8 "Feature" decomposta em superfícies

Dentro do contexto de representação do conhecimento, "features" podem ser definidas sob o paradigma de orientação a objetos, de forma a simplificar sua manipulação. Dessa forma "features" seriam objetos formalizados por uma lista de atributos, métodos e hierarquia herdada. Como já foi dito, as "features" são pré-definidas, sendo então armazenadas em uma biblioteca a partir da qual são instanciadas. Ou seja, o projeto de uma peça é desenvolvido a partir da união em Grupos de modelos de "features" existentes em uma biblioteca. Como decorrência desta abordagem as "features" são parametrizadas.

5.2.3 Identificação e classificação das "features"

A partir da definição de "features" apresentada na seção anterior, conclui-se que a representação de informações sobre produtos parte da combinação de objetos pertencentes a um conjunto. Portanto, a identificação de "features" é fortemente dependente dos produtos escolhidos como modelo. Nesta seção, mais que apresentar um conjunto de "features", procura-se definir uma estratégia para identificação e classificação.

A partir de um universo de produtos, a identificação de "features" deve tomar como base duas perguntas relacionadas: a quem interessam as "features" e quais as características que devem ser consideradas. Suas respostas levarão a considerações sobre a definição da "feature" e quais seus atributos, definindo assim os critérios de acesso para cada atividade do Sistema de Manufatura.

As "features" devem ser abordadas numa visão hierárquica. Num nível macroscópico as "features" genéricas são identificadas a partir de uma análise dos desenhos das peças. Nesta fase, uma abordagem de formação de famílias de peças auxilia sobremaneira a identificação de elementos repetitivos. Os critérios para a identificação destes elementos são definidos considerando as características do domínio escolhido.

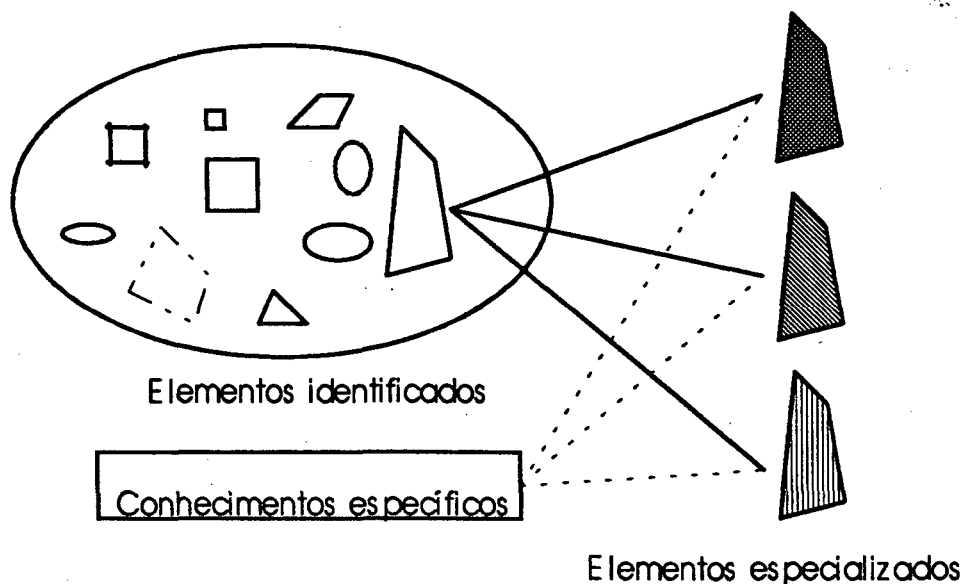


Figura 5.9 Especialização de objeto

O conjunto de "features" genéricas é a base para a identificação de "features" funcionais. Para tanto, devem ser analisadas as ocorrências de cada elemento, identificando as características que lhes atribuem a função. A especificação do objeto é feita a partir dos conhecimentos dos projetistas que a definiram. A figura 5.9 representa simplificada esta abordagem.

A modelagem das "features" tomou como base a representação de qualquer tipo de peça, a partir da representação apresentada pela Deere para a CAM-I [BUT86]. Sobre este modelo, detalhou-se as peças rotacionais quasi-simétricas definidas por Hernandez et al. em [HER91a, HER91b, GUT90, ROS90]. Todos estes modelos representam "features" genéricas, a partir das quais o exemplo de polia esticadora foi modelado. Para este exemplo, foram definidas subclasses funcionais para "features" genéricas.

As "features" se estruturam em classes que definem o domínio de aplicação no qual são utilizadas. Estas classes são:

- a) laminares;
- b) rotacionais quasi-simétricas de revolução;
- c) não rotacionais.

Para cada uma destas classes são definidas as "features" primitivas e os detalhes que vão se sobrepor a estas para formar o grupo. A figura 5.10 apresenta esta classificação.

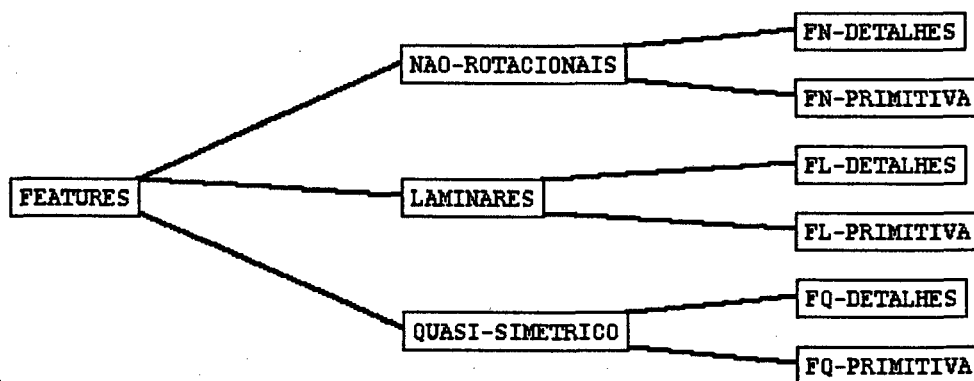


Figura 5.10 Classificação das "features"

As peças fabricadas a partir de chapas são representadas utilizando as subclasses do objeto laminares. A figura 5.11 apresenta a classificação destas "features". Neste exemplo simples é possível

observar os elementos construtivos genéricos de uma peça. Estes representam as "features" genéricas, não existindo qualquer funcionalidade agregada a elas.

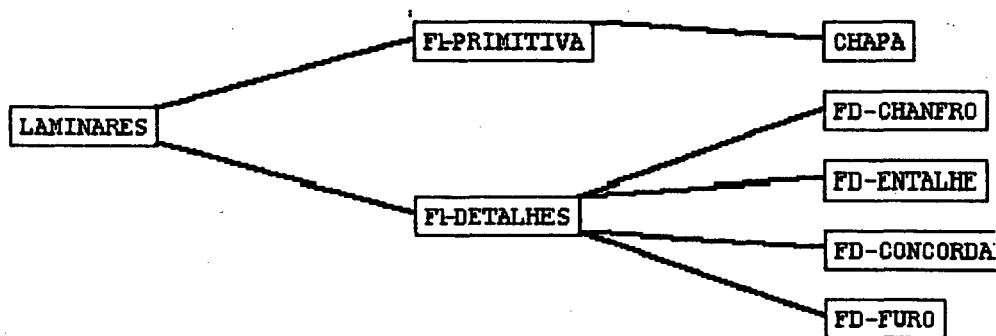


Figura 5.11 "Features" Laminares

As "features" quasi-simétricas de revolução são mais complexas, exigindo na representação uma série de subclasses que, a medida em que o objeto é especializado, vão agregando mais informações. A figura 5.12 apresenta esta classificação. As "features" são classificadas de acordo com sua posição no grupo, ou seja, interna ou externa. Cada uma destas é especializada em regulares e irregulares dependendo se abrange simetricamente a primitiva ou é localizada em uma área. Essa classificação é necessária para que os atributos, relativos a cada subclasse, sejam agregados a "feature" em etapas, permitindo uma modelagem mais clara dos objetos. Os atributos de cada subclasse estão listados no anexo 1.

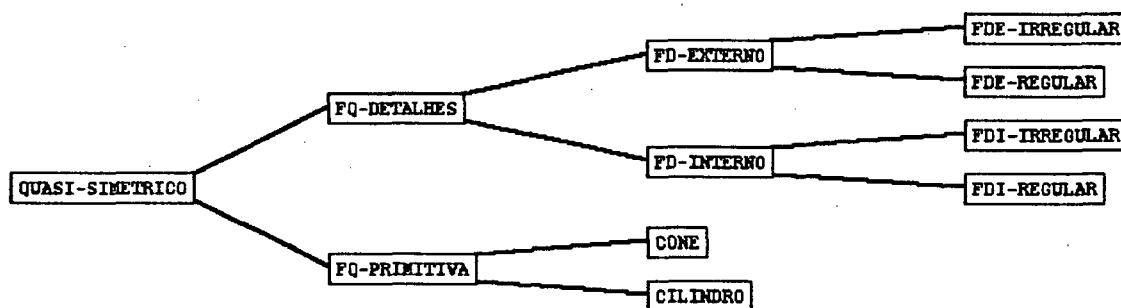


Figura 5.12 Classificação das "features" quasi-simétricas

Analisando a modelagem do objeto "feature" na base de informações (item 4.3 do anexo 1), observa-se a agregação de novos atributos a medida que esta é especializada, de forma que vão sendo somadas características ao objeto. Neste objeto as características que uma "feature" qualquer deve possuir estão representadas. Por exemplo: material, acabamento superficial, localização em relação ao grupo e tolerâncias de restrição entre superfícies. Neste nível as características são descritas de forma bastante abrangente. Sobre o material é possível afirmar somente que este é uma agregação de outro objeto, sua cardinalidade é 1 e nada mais. A medida que o grau de detalhamento vai aumentando, novas características vão sendo agregadas, assim como restringem-se mais os valores possíveis para cada atributo.

A figura 5.13 mostra a "feature" ranhura externa, com as dimensões que a caracterizam. Algumas dimensões, como a altura da ranhura são calculadas a partir dos diâmetros interno e externo. Os cálculos são definidos como subclasse do objeto cálculo e detalhados na seção 5.2.6.

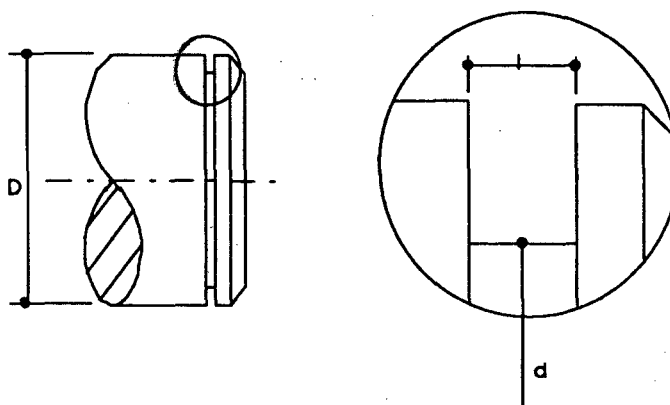
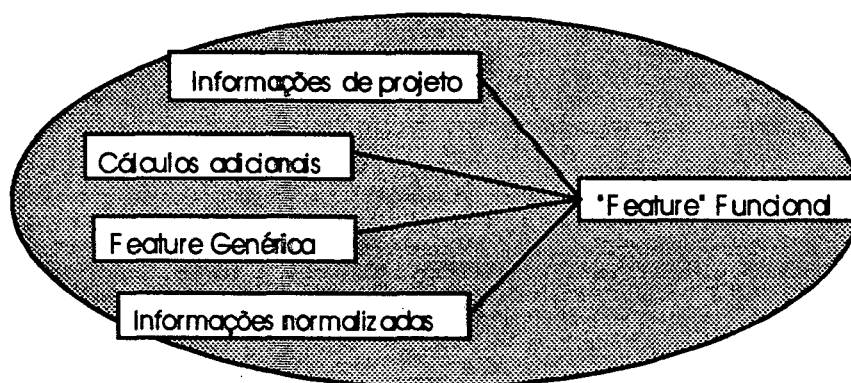


Figura 5.13 "Feature" ranhura externa

Dentro do contexto exposto anteriormente, uma "feature" ranhura externa é genérica uma vez que não é possível determinar sua função a partir de sua forma e informações herdadas. No sentido de atribuir-lhe uma função, uma série de outras características devem ser acrescentadas àquelas de uso comum.

Partindo de uma ranhura externa é possível definir, por exemplo, dois elementos de máquina: ranhura para colocação de anel de retenção e ranhura para anel de vedação. Ambos tem forma similar, mas seus detalhes construtivos são diversos. Para defini-los é necessário agregar às informações que definem uma simples ranhura uma série de conhecimentos de projeto. Por exemplo, para anel de retenção é necessário definir a tolerância dimensional no comprimento sendo que esta varia conforme o tipo do anel utilizado. Por sua vez, a "feature" para anel de vedação necessita de outras características como: inclinação nas paredes da ranhura, tolerância dimensional nos diâmetros e especificação para a rugosidade superficial interna a ranhura. Todas estas informações são conhecimento de projeto, proveniente de normalização do uso de elementos de máquina.

Conforme pode ser observado comparando os três objetos relativos a ranhuras, listados no anexo 1, a modelagem destes é realizada pelo acréscimo de informações, inserindo novos atributos nos objetos mais especializados ou restringindo aqueles definidos na superclasse. A figura 5.14 apresenta a construção de uma "feature" funcional a partir de uma genérica, agregando os conhecimentos discutidos acima.



5.14 Construção de uma "feature" funcional

A classificação das "features" no modelo é um espelho de sua definição. As "features" funcionais são subclasses das "features" genéricas que lhes dão forma (figura 5.15). Na realidade, a única distinção existente está no grau de especialização de cada "feature" o que proporciona bastante flexibilidade na modelagem na criação dos grupos e peças. Da mesma forma nada impede que uma subclasse de uma "feature" funcional seja criada, agregando a ela mais informações específicas.

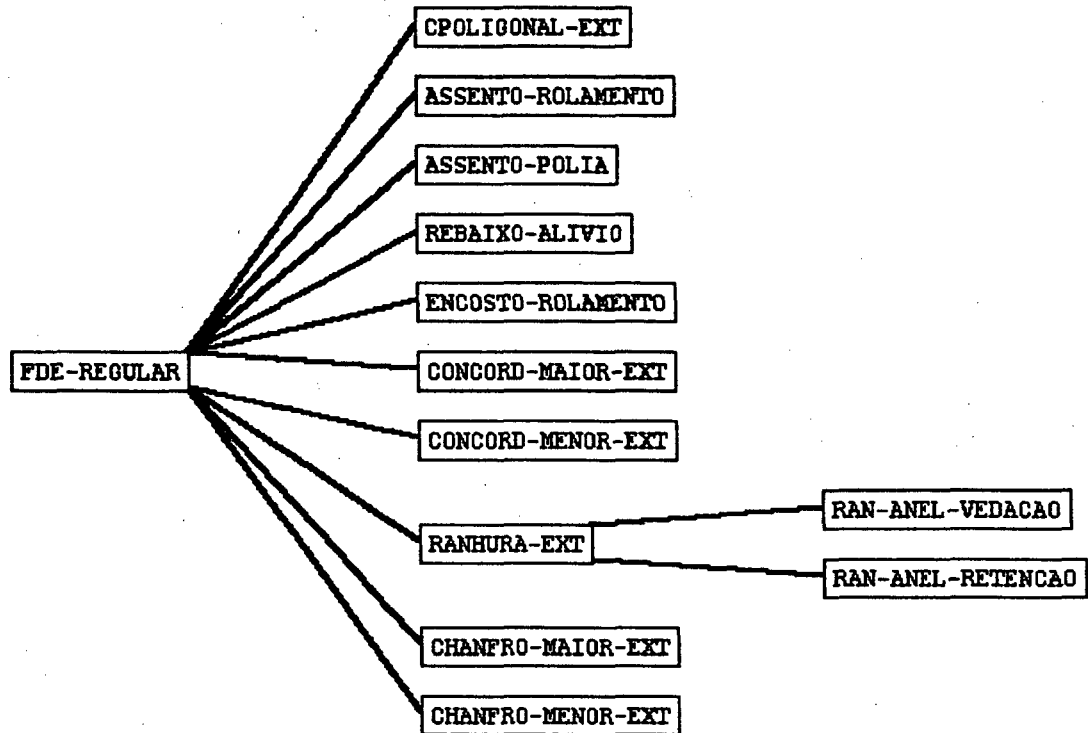
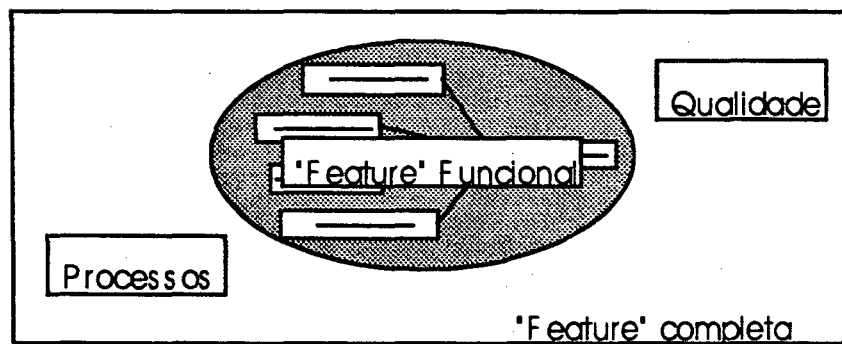


Figura 5.15 Classificação de "features" detalhe externo regular

Uma vez definida a "feature" funcional, é possível partir para a identificação de suas demais características, provenientes de outras atividades do Sistema de Manufatura (figura 5.16), como por exemplo, informações de processos de fabricação e engenharia de qualidade. Deve-se ter o cuidado de respeitar as restrições hierárquicas definidas anteriormente, ou seja, uma "feature" armazena informações que dizem respeito a ela própria. Relações entre elementos construtivos de um mesmo nível hierárquico são tratadas por seu superior.



5.16 Construção de uma "feature"

No momento em que é realizada a escolha de uma "feature" (instanciação) para uso em projeto, as restrições sobre cada característica aumentam. Estas restrições são informações provenientes de tabelas padronizadas, catálogos de fornecedores, normas, etc. utilizadas pelo projetista na definição do elemento construtivo. Esta facilidade é descrita na seção 5.2.5

5.2.4 Elementos agregados

Os objetos básicos da classe funcional e tecnológica agregam informações complementares, que auxiliam na sua definição. Em geral, estes complementos são igualmente complexos, devendo portanto ser representados como objetos. A união destas informações se dá a partir de uma relação de agregação.

Superfícies

Conforme descrito anteriormente, as superfícies conformam as "features", representando suas informações geométricas. Além disso, mantém valores de material e acabamento superficial, de forma a possibilitar seu detalhamento internamente às "features". Tanto material como acabamento são definidos para a peça e repassados para os níveis hierárquicos inferiores. Este processo permite que os valores sejam definidos somente uma vez e, se necessário, são alterados para um determinado elemento construtivo específico sem que isto reflita sobre os demais.

A representação de tolerâncias de restrição internas a uma "feature" (por exemplo, o paralelismo entre as duas faces de uma ranhura) é realizado utilizando as superfícies como referência. As tolerâncias de forma de uma "feature" são igualmente representadas pelas superfícies.

Da mesma forma que as "features" as superfícies devem ser identificadas especificamente para um domínio. O conjunto de superfícies deve ser definido de tal forma que na construção de uma "feature" mais de uma superfície seja utilizada. Assim, é possível representar o máximo de informações internas à "feature", aumentando o poder de representação do modelo.

As superfícies para as "features" quasi-simétricas são apresentadas na figura 5.16. Uma vez que superfícies são cascas não é necessário classificá-las em internas e externas, sendo possível utilizá-las em qualquer posição. É importante observar que as superfícies para representação das "features" chavetas não seguem a regra definida anteriormente e utilizam somente uma superfície dedicada. Isto ocorre devido à complexidade de suas formas.

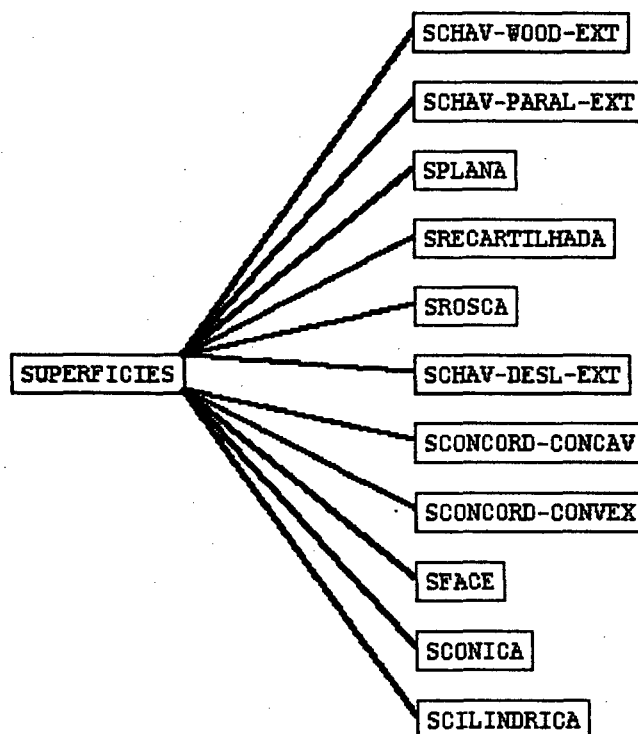


Figura 5.16 Classificação das superfícies.

As "features" utilizam um conjunto constante e pré-definido de superfícies para modelá-las, dimensionando-as a partir de suas próprias medidas e tolerâncias. A figura 5.17 mostra a relação entre "features", descritas na seção 5.2.2, e as superfícies que as compõem. A localização das superfícies em relação à "feature" é descrito na seção 5.4.2.

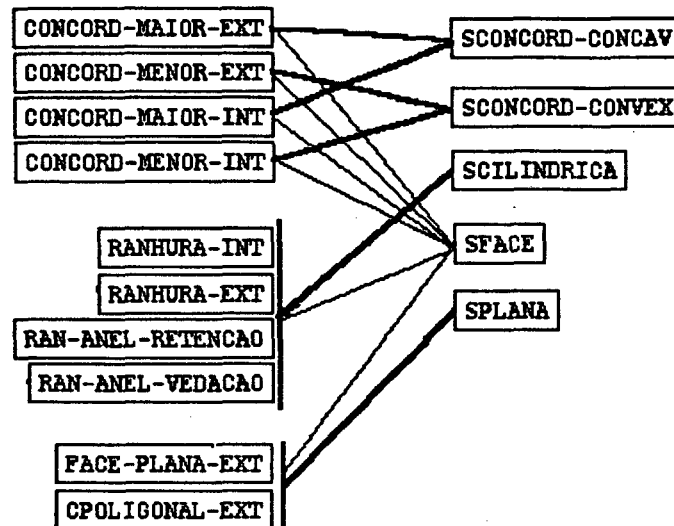


Figura 5.17 Relação "feature"/superfície

A utilização de um conjunto pré-definido de superfícies para uma "feature" implica em intersecções entre superfícies, fato este que não é tratado no modelo. Por exemplo, a "feature" ranhura interna, figura 5.18, é composta por uma superfície cilíndrica e duas faces, resultando em um cilindro fechado. A intersecção deste com a superfície cilíndrica do furo resulta na forma final da ranhura.

Uma vez que o modelo não trata as intersecções, as ferramentas que as manipularem devem possuir conhecimento para tal. Por exemplo, ao ser visualizada, em um sistema CAD, as partes de superfícies que sobram da intersecção devem ser eliminadas. Da mesma forma o CAPP deve reconhecer estas intersecções ao identificar as superfícies e definir a localização das operações. Este problema deve ser encarado do ponto de vista de CSG¹⁵ e operações de intersecção.

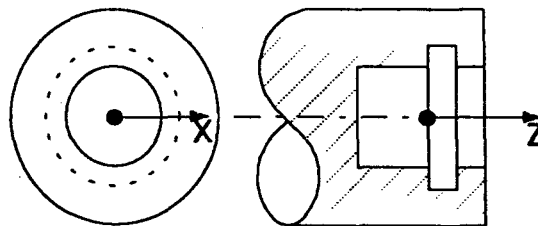


Figura 5.18 Superfícies na "feature" ranhura interna

As superfícies são identificadas sobre o domínio das "features" que devem construir. Os princípios de modelagem são os mesmos em relação a grupos e "features".

Tolerâncias

A construção de um produto mecânico pressupõe confiabilidade tanto nas medidas como no posicionamento das partes que o compõe. As tolerâncias realizam esta função garantindo a relação correta entre cada elemento construtivo do conjunto. Estas são divididas de acordo com sua função:

- as tolerâncias dimensionais definem quanto uma medida pode estar fora de seu valor nominal;
- as tolerâncias de forma especificam as diferenças entre a superfície real da peça e a forma geométrica teórica;

¹⁵ CSG - "Constructive Solid Modeling" é uma técnica de modelagem de peças em CAD, na qual, cada objeto é descrito por um conjunto de combinações entre primitivas sólidas. As combinações são representadas por operações booleanas.

c) tolerâncias de restrição relacionam dois elementos construtivos diferentes, entre os quais deve ser mantida uma posição ou orientação relativa.

Na modelagem, as tolerâncias foram classificadas de acordo com sua abrangência (figura 5.19). Ou seja, as tolerâncias de restrição, que relacionam dois objetos diferentes e as tolerâncias relativas a um único objeto.

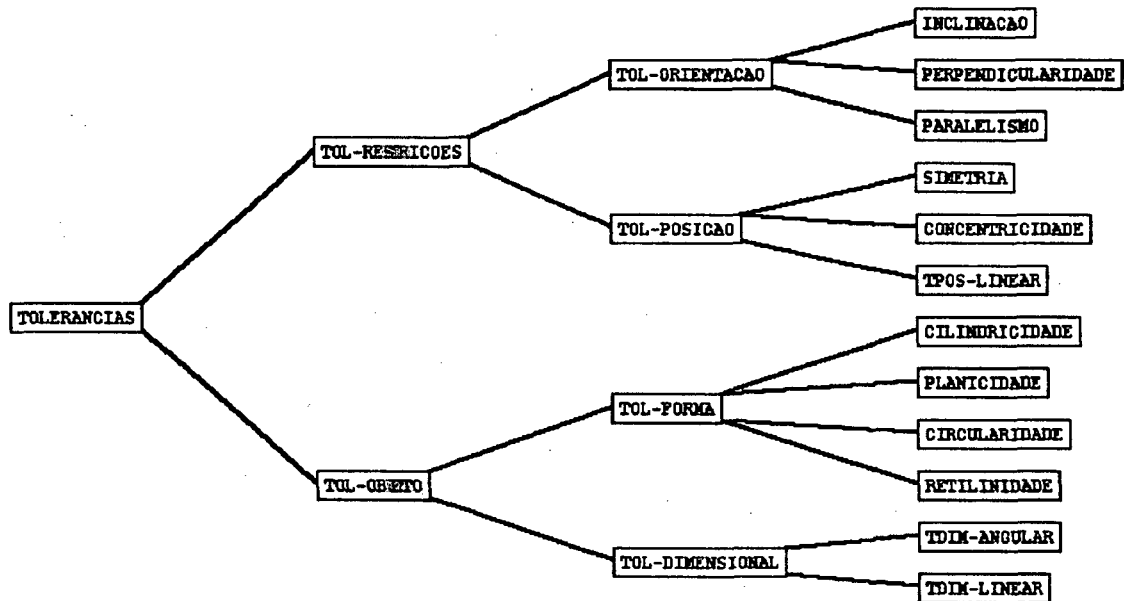


Figura 5.19 Classificação de tolerâncias

As tolerâncias dimensionais são agregadas diretamente à dimensão que restringem. Na figura 5.20 é possível observar que a "feature" primitiva — cilindro163 possui um diâmetro definido pelo objeto ddiametro167 e uma tolerância dimensional representada por tol-dimensional168. A implementação destas relações esta listada no item 4.4 do anexo 1.

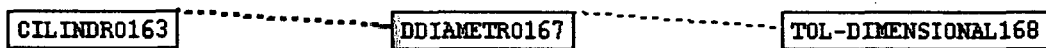


Figura 5.20 Agregação de tolerância dimensional

Como descrito no item anterior, as tolerâncias de forma são representadas diretamente sobre as superfícies. A figura 5.21 apresenta a estrutura de agregação das informações, sendo que scilindrica169 é uma superfície cilíndrica que compõe a "feature" primitiva cilindro163 e possui tolerância de cilíndricidade. O esquema de agregação e os atributos que armazenam os valores estão listados no item 4.4 do anexo 1.



Figura 5.21

As tolerâncias de restrição são sempre definidas no nível hierárquico superior. Na figura 5.22, a concentricidade entre os grupos eixo-simples é representada a nível de peça. Uma vez que neste tipo de restrição dois objetos são relacionados, é importante representar qual é a referência para que o outro seja comparado. Esta relação é representada por dois atributos no objeto tolerância de restrição: obj-referencia e obj-referenciado. A implementação desta modelagem está listada no item 4.4 do anexo 1.



Figura 5.22

5.2.5 Normas

As "features" são modelos que, para uso prático, devem ser preenchidas com valores em seus atributos. Muitas vezes, as "features" funcionais são elementos construtivos padronizados, sendo que seus atributos são pré-definidos em tabelas e normas.

Em geral, somente "features" funcionais são normalizadas, pois estas são definidas a partir do uso de um elemento construtivo. Esta característica favorece seu uso, estimulando a padronização dos projetos realizados. As normas trazem consigo uma série de vantagens:

- a) reduzir custos através da compra de peças de uso comum como parafusos, anéis e chavetas;
- b) facilitar a montagem das peças utilizando construções já testadas;

A modelagem das normas (figura 5.23) no AVEC está baseada no princípio de que normas para elementos construtivos diferentes não possuem semelhança. Por este motivo, a relação entre os objetos é de subconjunto e não de subclasse. Neste segundo caso haveria transferência de características através do processo de herança.

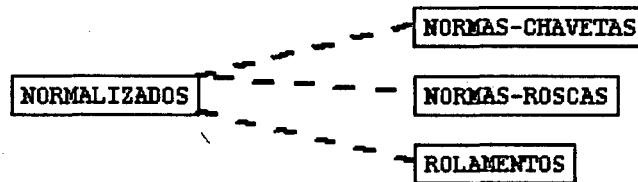


Figura 5.23 Modelagem das normas

Uma "feature" funcional é instanciada contendo apenas o modelo a ser utilizado na construção do produto. A partir daí, deve ser designado um valor a seus atributos para que a "feature" possa ser inserida no grupo ao qual pertence. Durante o processo de definição dos valores de cada atributo deve ser estimulado o uso de dados padronizados. A normalização de uma "feature" não insere novos atributos, apenas restringe seus valores. Isto ocorre através da definição de um espectro de valores para cada atributo, além de relações entre eles.

Cada subconjunto de normalizados é representado de uma maneira própria, segundo a forma como atua sobre os elementos construtivos. As tabelas para rolamentos¹⁶ relacionam o diâmetro interno do rolamento com o "assento" no qual este será inserido e seu "encosto". Basicamente, a normalização atinge tolerâncias dimensionais (para as medidas de diâmetro) e de cilíndricidade com relação a forma do "assento". Além disso, entre o "encosto do rolamento" e o "assento" é definida uma tolerância de perpendicularidade. Cada uma destas áreas de atuação da norma é uma subclasse do objeto rolamentos (figura 5.24).

Uma vez que a norma não segue uma relação matemática, os rolamentos devem ser representados um a um, como objetos. Desta forma o rolamento classe 61807, é um objeto do sistema, com atributos que

¹⁶ Catálogo Geral da SKF, Edição 1992, na página 235.

armazenam os valores para os quais ele está normalizado. Suas subclasses, dedicadas a cada elemento de interesse da norma, armazenam métodos que irão atuar sobre a "feature" a ser padronizada.

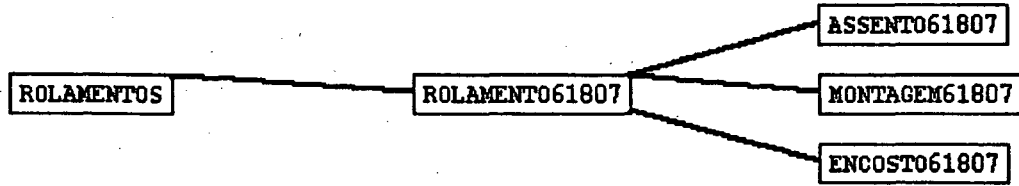


Figura 5.24 Modelo de norma para rolamentos

Por motivos de implementação, para que um aspecto possa ser especializado, é necessário que haja uma relação de subclasse entre eles. Assim, (figura 5.25) a normalização de "assentos de rolamento" (assento061807), é subclasse tanto da norma quanto do objeto que deve normalizar. Os procedimentos de normalização são representados como métodos e disparados por um "demon" no atributo sobre o qual atuará. Este procedimento é representado pelo atributo diâmetro da figura 5.25.

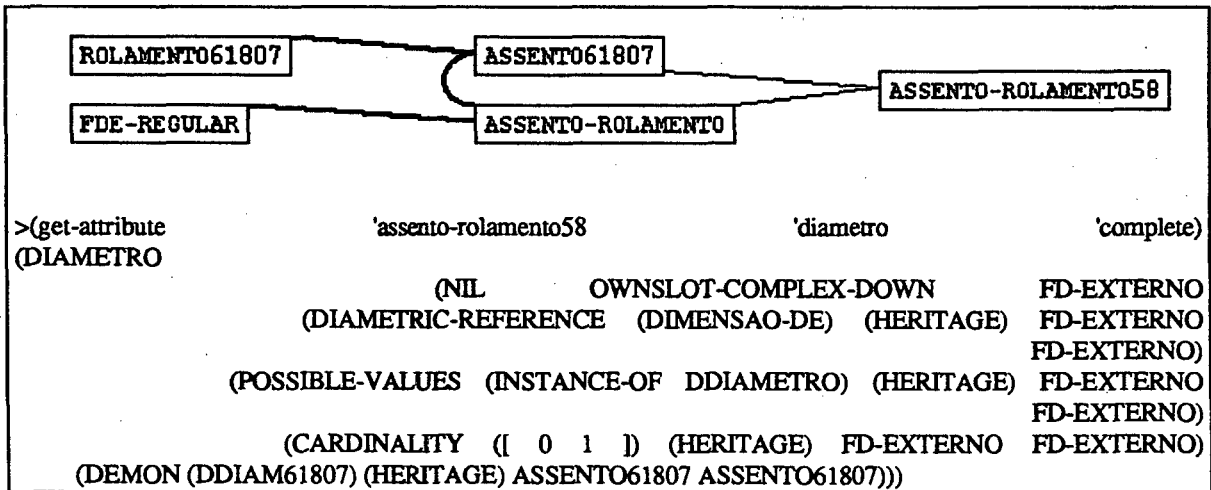
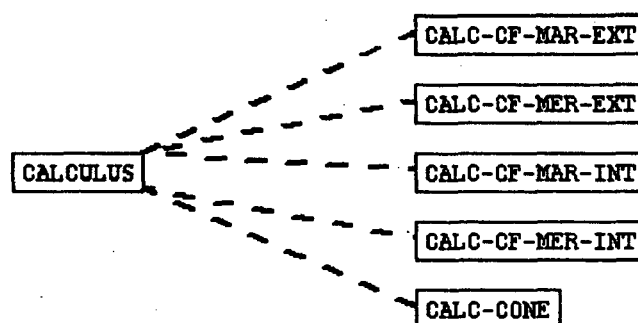


Figura 5.25 Normalização de uma instância de rolamento

Por fim, a normalização dos valores na instância (objeto assento-rolamento 58) é implementada pela múltipla instanciação da "feature". Ou seja, a "feature" é ao mesmo tempo instância de "feature" e da normalização correspondente, herdando, desta forma, características de ambos os objetos.

5.2.6 Cálculos

Qualquer elemento construtivo é definido por um conjunto de atributos primitivos, a partir dos quais outros são derivados por meio de cálculos sobre os primeiros. No sentido de diferenciar as duas classes de atributos, os primitivos são modelados no próprio objeto, enquanto que os derivados estão em um subconjunto do objeto "calculus" (figura 5.26).



5.26 Objetos cálculo

Os procedimentos de modelagem são similares aos utilizados para normas. No entanto, no modelo para cálculos, a classe "feature" é subclasse de duas outras classes (diferente de normas onde a união ocorria somente na instância), herdando o somatório dos atributos, tanto os primitivos como os derivados. Este procedimento difere das normas pois aqui se sabe, de antemão, quais objetos serão conectados.



Figura 5.27 Utilização dos atributos de cálculo

5.3 Informações de Facilidades

As informações sobre facilidades descrevem o chão de fábrica em termos de recursos disponíveis para a fabricação. Para tanto estão divididas em três áreas específicas:

- a) máquinas-ferramentas;
- b) ferramentas;
- c) sistemas de fixação.

A modelagem das informações de facilidades apresentada aqui é baseado no trabalho realizado por Hernandez et al. em [HER91a, HER91b, GUT90, ROS90]. Nesta seção é discutida a implementação proposta, no sentido de servir como referência para futuros estudos visando sua perfeita integração às aplicações que delas fizerem uso.

Na abordagem utilizada, as máquinas-ferramentas são o centro do modelo, sendo que as relações entre uma máquina-ferramenta/ferramentas/sistemas de fixação são descritas a partir delas. Esta proposta é apropriada para o AVEC, uma vez que algumas atividades do Sistema de Manufatura, notadamente o planejamento de processos e o escalonamento da produção, fazem uso destas relações na geração de seus serviços.

A classificação dos objetos segue o exemplo já descrito para "features", no qual, informações vão sendo agregadas a medida que se tornam mais específicos na sua utilização. No anexo 1, item 2.1 é listado um exemplo de especialização para ferramentas.

A figura 5.28 apresenta a classificação para máquinas-ferramentas.

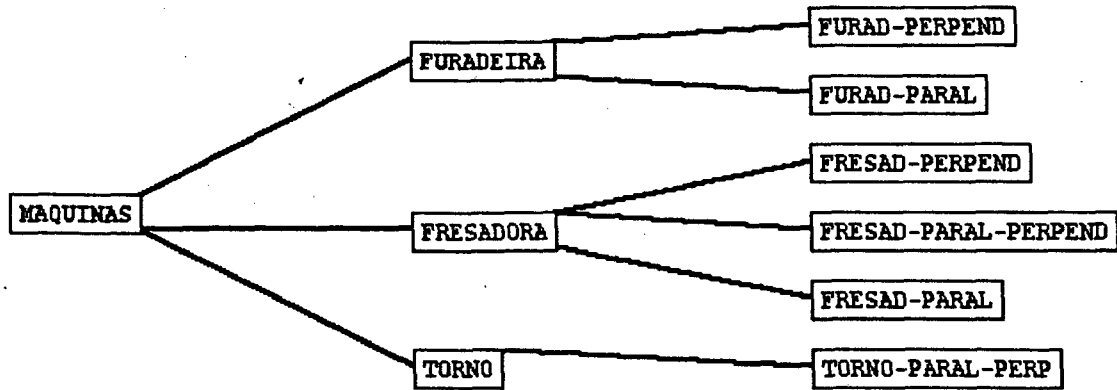


Figura 5.28 Classificação de máquinas-ferramentas

Como dito anteriormente, as ferramentas são classificadas de acordo com seu uso nas máquinas. A figura 5.29 apresenta esta classificação.

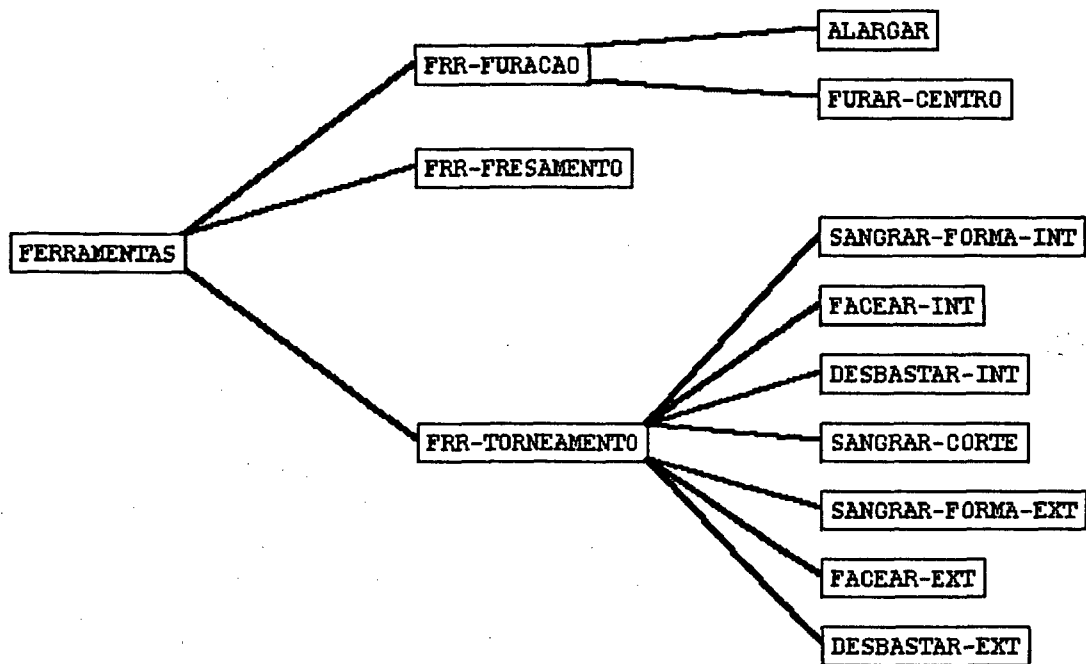


Figura 5.29 Modelagem das ferramentas

A fixação de uma peça para ser usinada é uma tarefa que requer uma série de cuidados. As informações relativas ao tratamento desta atividade devem estar representadas nos objetos do modelo, portanto o sistema de fixação é representado como a união de informações de dispositivos de fixação e superfícies nas quais estes se encaixam. Estas informações são representadas através de relações de agregação.

A seguir, as figuras 5.30 e 5.31, apresentam a modelagem dos sistemas e dispositivos de fixação.

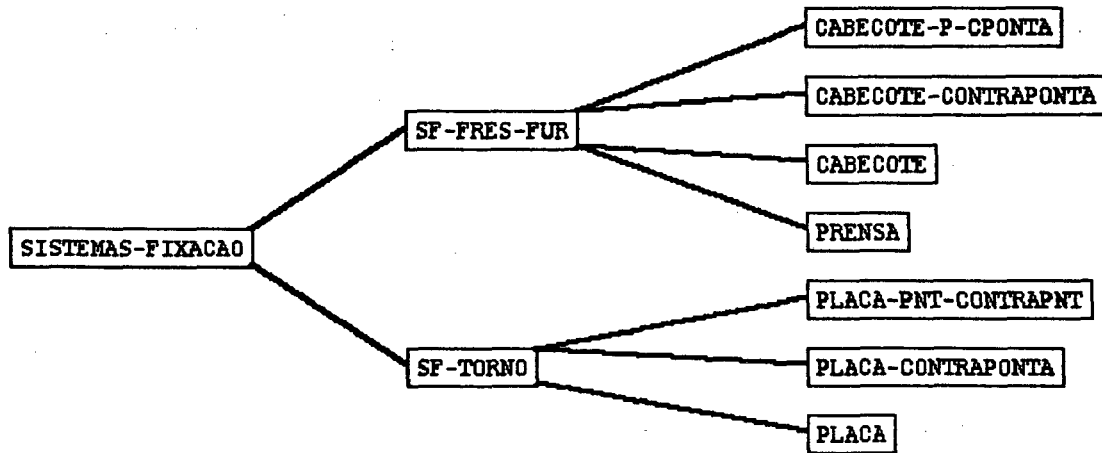


Figura 5.30 Modelagem dos sistemas de fixação

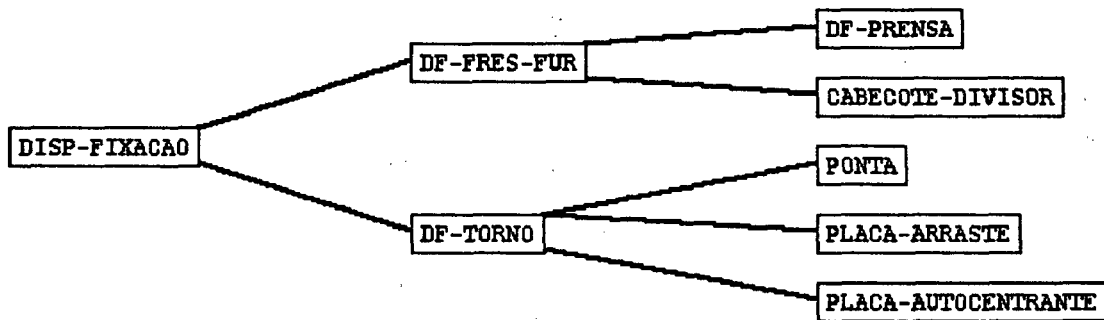


Figura 5.31 Modelagem dos dispositivos de fixação

5.4 Informações de enlace

Como discutido na seção 5.1, as informações de enlace suportam as relações entre funcionalidade de grupos e "features", tecnologia e facilidades. Para tanto, um amplo espectro de informações deve ser modelado. Por isto, os objetos partem de um nível bastante abstrato, sendo refinados até elementos básicos. A figura 5.32 apresenta os elementos iniciais do conjunto de informações de enlace.

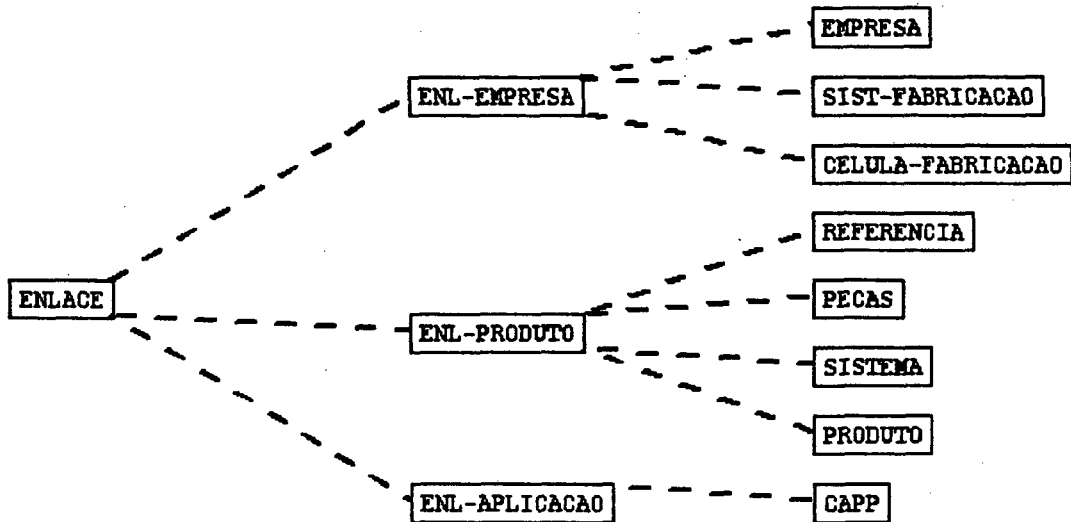


Figura 5.32 Modelagem das informações de enlace

A identificação dos objetos que descrevem o enlace inicia com a observação do fluxo de informações a nível de Sistema de Manufatura (seção 2.1.3). Neste, é possível identificar informações corporativas como, análises de mercado, autorização de serviços e gerenciamento de recursos financeiros, ou seja, informações complexas distribuídas em uma série de atividades diferentes. O ponto comum entre elas é a característica de controle da empresa como um todo, sendo o ponto de partida para as definições de manufatura.

Por outro lado, a definição de produto, parte das decisões gerenciais para as demais funções da empresa. Estas informações se constituem, em última análise, em elementos construtivos que, no conjunto, formam o produto.

As informações de controle sobre as ferramentas, que atuam no auxílio ao desenvolvimento das funções de manufatura, representam outro conjunto de informações. Neste, devem ser representadas as informações comuns entre CAD, CAPP, CAI, etc.

A seguir cada subconjunto é descrito, sendo dada ênfase para as informações de modelagem do produto.

5.4.1 O modelo do produto

No contexto deste trabalho, um produto, é uma conjunção de funcionalidades localizadas, resultando em um novo elemento, com características próprias. Sob este ponto-de-vista, seus atributos devem representar as funções de mais alto nível hierárquico do produto. Desta forma cada elemento construtivo se encarrega de representar seu próprio nível, evitando redundâncias na representação das informações. A especificação dos níveis de abstração segue o trabalho realizado por Rosa et al. [ROS92, ROS93].

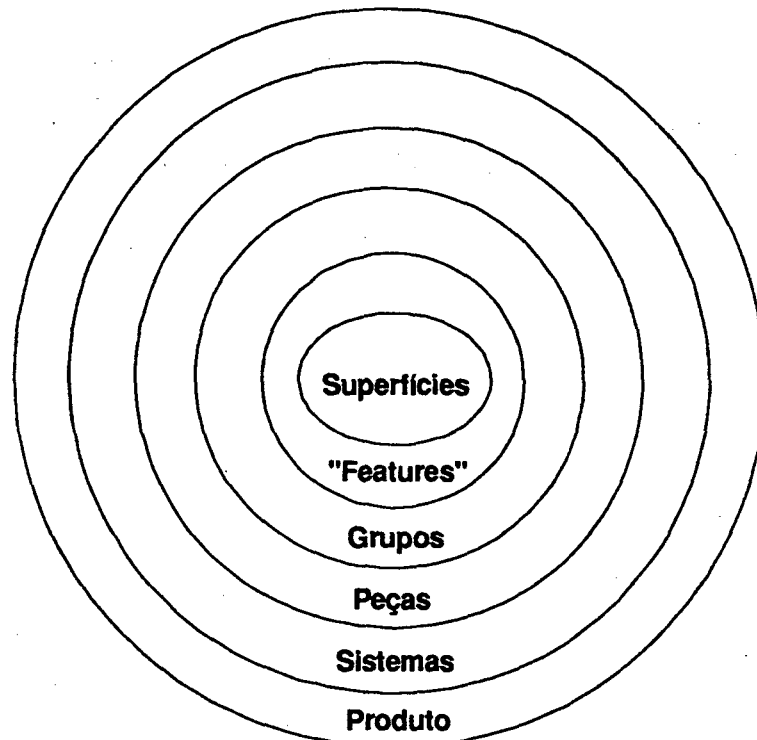


Figura 5.33 Hierarquia de Produto

Durante a definição de um produto, pode ser necessária a utilização repetida do mesmo elemento construtivo. Isto é possível no modelo proposto, uma vez que a localização é uma informação agregada ao objeto. Uma vantagem desta abordagem é a possibilidade de armazenamento de elementos construtivos comuns, a nível de projeto, e utilização sem modificações em outro ponto do projeto. Um exemplo de aplicação desta facilidade pode ser o uso de elementos normalizados. Essa facilidade é descrita na seção 5.4.2.

A seguir cada parte do produto é descrita, sendo que a modelagem de um exemplo de produto é apresentada na seção 5.5.

Produtos

Um produto é o elemento final, sobre o qual não será agregado nenhum outro elemento construtivo. Ou seja, é um elemento de convergência de informações proveniente dos níveis hierárquicos inferiores. Neste nível são modeladas as informações que descrevem: identificação, estado atual e código do produto. Quanto mais específico for o objeto para um determinado domínio de produtos, mais atributos este pode representar. Assim sendo, é aconselhável que o objeto produto seja especializado, definindo novas subclasses.

Sistemas e sub-sistema

Um conjunto de sistemas define um produto. Por isso, um sistema é definido de acordo com as necessidades de projeto, portanto pode ser dividido em sistemas menores, de forma a facilitar sua compreensão. A modelagem de sistemas prevê a possibilidade da especialização de sistemas a partir da relação de agregação. Desta forma é possível construir uma árvore de objetos (figura 5.34).

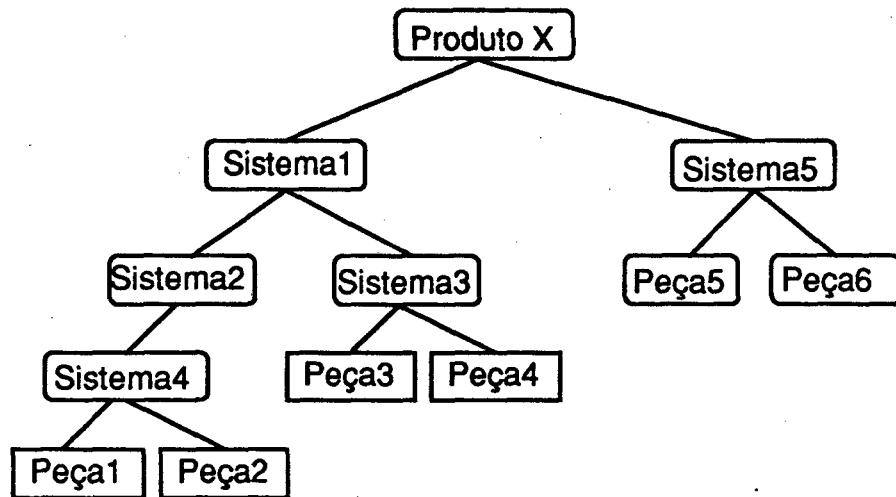


Figura 5.34 Construção de um produto

Um sistema pode agregar outros sistemas ou peças de forma excludente (figura 5.35). Esta restrição é garantida através de um método que verifica as agregações no objeto. A implementação está listada no anexo1, item 3.1.

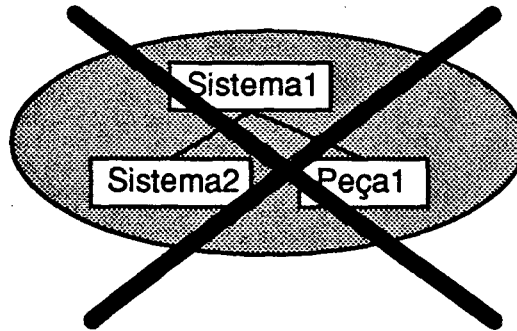


Figura 5.35 Agregação de sistemas e peças

Peças

Uma peça é o menor elemento construtivo a nível de usinagem, ou seja, nenhuma operação de montagem é realizada para produzi-la. Assim, qualquer elemento que contenha montagem é considerado um sistema. O objeto peças permite representar informações como por exemplo: material, acabamento superficial e tolerâncias de restrição entre grupos. A descrição completa do objeto está listada no anexo 1, item 3.2.

O material e o acabamento superficial têm um tratamento diferenciado dos demais atributos, uma vez que possuem métodos que disseminam estas informações para os níveis inferiores. Desta forma é possível definir um valor padrão para a peça como um todo e modificá-lo somente nos locais necessários. A implementação desta facilidade é apresentada no anexo 1, item 3.2.

Como mencionado anteriormente, as peças são formadas por grupos, mantendo assim um forte vínculo com as informações funcionais. No objeto peças são definidos métodos para a geração automática dos objetos que a compõem. Ou seja, uma vez definidos grupos pertencentes a uma peça, estes serão automaticamente instanciados, gerando as "features" e superfícies. A abordagem e os métodos utilizados são discutidos na seção 5.5.3.

5.4.2 Sistema de referência

Como descrito anteriormente o produto é representado hierarquicamente através de seus elementos construtivos (sistema, peça, grupo, "feature", superfície), sendo que cada um destes possui um sistema de coordenadas local, posicionado em relação ao nível anterior. Por conseguinte, o produto é o único localizado em relação ao sistema global de coordenadas. O sistema de coordenadas utilizado é o cartesiano, com 3 dimensões.

A localização de um elemento do sistema é feita em função do seu deslocamento (X,Y,Z) em relação ao sistema local do nível anterior. Além disso, pela rotação $(\alpha_x, \alpha_y, \alpha_z)$ em torno de seus eixos locais de coordenadas. A figura 5.36 apresenta um exemplo em três dimensões onde é possível observar o deslocamento de cada entidade em relação ao sistema de coordenadas local. A localização da elipse por exemplo é: $(X1,Y1,0)$ e $(0,0,90)$ em relação ao sistema local de coordenadas $S1$. Sendo possível transportá-lo para o sistema global — SG .

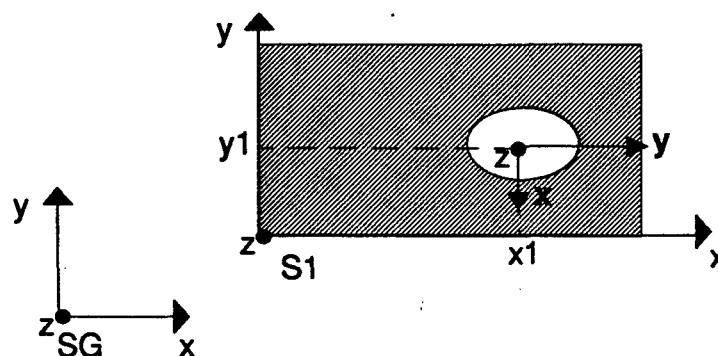


Figura 5.36 Sistemas de coordenadas locais

Qualquer elemento construtivo possui um ponto de referência definido, a partir do qual se fará seu posicionamento. Este ponto é variável, dependendo da forma do elemento e a facilidade de representação. Nos níveis de produto, sistema e peça este é definido pelo projetista, de acordo com as necessidades de projeto. Mas, para grupos, "features" e superfícies, os pontos são pré-definidos. Os elementos construtivos destinados às peças simétricas de revolução têm seu ponto de referência localizado conforme a figura 5.37. Ou seja, no eixo da peça, sobre seu maior diâmetro (se o diâmetro for constante, é considerado o lado esquerdo).

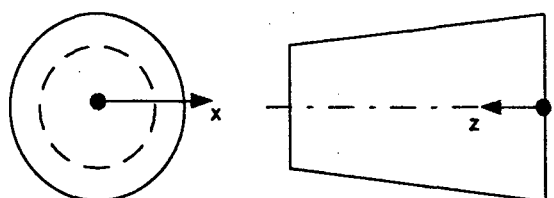


Figura 5.37 Ponto de referência para peças de revolução

Para os elementos construtivos não simétricos os casos são definidos um a um. Na figura 5.38 é apresentada a definição para um caso de "features" irregulares.

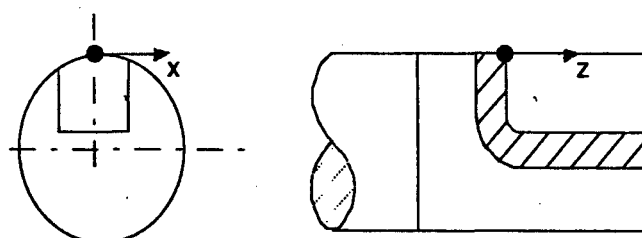


Figura 5.38 Ponto de referência para "features" irregulares

Quando um elemento construtivo é instanciado, sua localização é na origem de seu sistema de coordenadas local, sem rotações sobre eixo algum. O deslocamento deste elemento para sua posição final é realizado através da multiplicação da posição atual do objeto pela matriz de transformação (figura 5.39), que executa as operações de translação e rotação concomitantemente.

$$MT = \begin{vmatrix} Cy * Cz & Cy * Sz & -Sy & X \\ (Sx * Sy * Cz) - (Cz * Sz) & (Sx * Sy * Sz) + (Cx * Cz) & Sx * Cy & Y \\ (Cx * Sy * Cz) + (Sx * Sz) & (Cx * Sy * Sz) - (Sx * Cz) & Cx * Cy & Z \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

C = cosseno S = seno
(x, y, z) = valor das rotações sobre os eixos
(X,Y,Z) = valor das translações

Figura 5.39 Matriz de transformação para os elementos construtivos

A implementação deste método é apresentada no anexo 1 item 3.3.

Como citado anteriormente, a localização de peça e sistema é feita de tal forma que uma mesma instância do objeto possa ser utilizada em mais de um local sem alterações em sua estrutura interna. Esta facilidade é modelada utilizando um objeto auxiliar para suportar a informação de localização. A figura 5.40 apresenta a implementação desta facilidade. No exemplo, um produto (produto167) é composto, inicialmente, por dois sistemas (sistema170 e sistema173). A localização dos sistemas em relação ao produto é realizada pelos objetos localiza-sistema (localiza-sistema171 e localiza-sistema168), que por fim agregam as matrizes de transformação (mtransf172 e mtransf169). O objeto polia-esticadora é uma peça pertencente ao sistema170. Por motivos de implementação sua localização é feita a partir de um objeto de classe diferente, localiza peça (localiza-peca174).

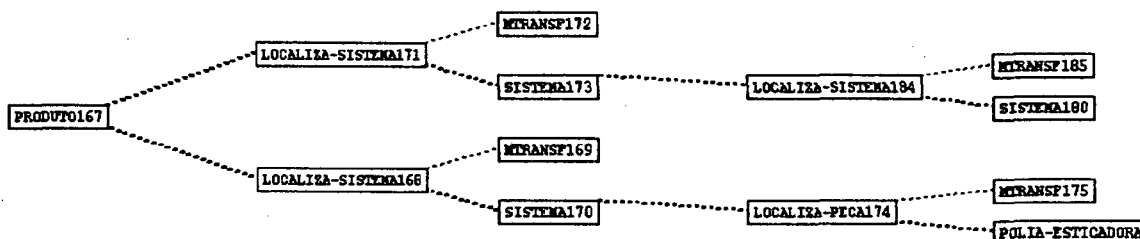


Figura 5.40 Representação da localização de sistemas e peças

Na figura, é possível notar ainda a ligação de sistemas em série, localizados através do objeto localiza-sistema. A modelagem permite ainda que uma mesma peça seja utilizada por dois sistemas ao mesmo tempo, como é representado na figura 5.41.

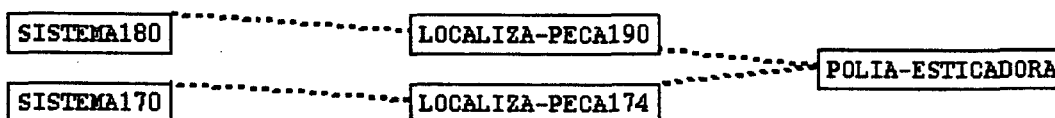


Figura 5.41 Reutilização de peças

Não está previsto no modelo a possibilidade de grupos e "features" serem reaproveitados da mesma forma que peças e sistemas. Os grupos e "features" são elementos pertencentes a bibliotecas e instanciados novamente a cada vez, de forma que seria muito oneroso ao sistema uma consulta sobre todos os elementos

instanciados a procura de elementos similares para serem reutilizados. A nível de peça e sistema a reutilização é responsabilidade do usuário do sistema facilitando a modelagem.

5.4.3 O modelo de empresa

Um modelo para empresa deve priorizar a integração das informações gerenciais relevantes para a Engenharia Concorrente. Embora não faça parte do escopo deste trabalho representar as informações a este nível, foi modelado o objeto inicial, a partir do qual devem ser inseridas novas informações.

Fazem parte das informações de enlace de empresa o leiaute do chão de fábrica, unindo as informações representadas na classe de facilidades. Estas informações são utilizadas na geração do planejamento da produção.

5.4.4 O modelo de aplicações

Como descrito no capítulo 4, uma série de informações relativas a integração das ferramentas de auxílio ao desenvolvimento das atividades no Sistema de Manufatura devem ser representadas no objeto "opambae". A partir do objeto enlace de aplicação estas informações podem ser representadas.

A seguir a modelagem de informações de aplicação é exemplificada através das atividades de planejamento de processos.

Modelo para CAPP

No anexo 4 é apresentada uma pequena introdução ao planejamento de processos sendo descrito o sistema IAGE, um CAPP generativo-iterativo, que serve de base para este exemplo. No modelo representado, cada "feature" agrega informações relativas aos processos utilizados na sua produção. Uma vez que uma "feature" manipula informações relativas, exclusivamente, a si mesma, neste nível podem ser representadas apenas informações localizadas. A união destas é função do conhecimento representado no sistema CAPP.

A figura 5.42 apresenta a classificação das operações de fabricação de torneamento. As informações relacionando o processo as "features" passíveis de serem produzidas são representadas por meio de aspectos e métodos.

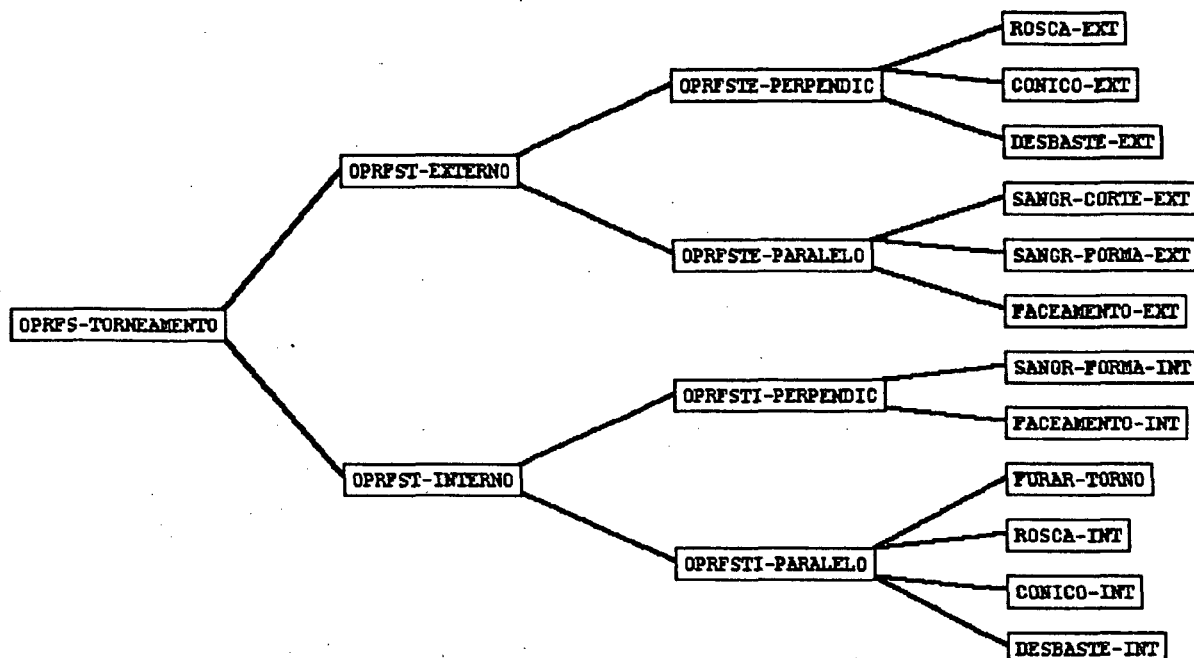


Figura 5.42 Classificação das operações de torneamento

A figura 5.43, apresenta a relação entre duas "features" e suas respectivas operações de fabricação. Nota-se a diferença existente entre uma ranhura para anel de retenção e anel de vedação. No primeiro, as paredes da ranhura não apresentam inclinação sendo possível utilizar apenas uma operação de sangramento de corte. No caso da vedação, é necessária uma inclinação nas paredes, exigindo portanto um sangramento de forma posterior.

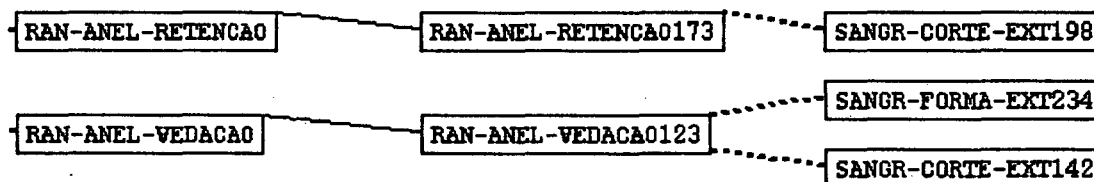


Figura 5.43 Relação entre "features" e operações de fabricação

Um sistema CAPP, construído no ambiente AVEC, deve reunir informações que possibilitem organizar aquelas representadas nas "features" e assim gerar o plano de processos.

5.5 Considerações sobre a implementação

5.5.1 Porque o sistema Krisys

Conforme descrito na seção 3.4.4, o Krisys possui potentes mecanismos de representação de informações baseados em orientação a objetos. Essa facilidade é fundamental para que a complexidade de relações existentes no Sistema de Manufatura sejam modeladas.

As relações entre os objetos do Sistema de Manufatura possuem características que as diferenciam da maioria das aplicações. Isto é, objetos complexos se relacionam fortemente sob diversos aspectos. A representação destes objetos necessita de modelos que possibilitem relações de abstração (seção 3.4.5) de

forma a modelar as relações e restrições a que são submetidos. Além disso, o sistema deve suportar independência de dados e troca de mensagens.

O sistema Krisys suporta relações de abstração (classificação/generalização, agregação e associação) que permitem a modelagem das relações entre as informações e, além disso, permite representar, através dos aspectos, a maior parte das necessidades de restrição entre os objetos modelados. Ainda assim, quando não for possível fazê-lo através dos aspectos pré-definidos, os "demons" são suficientemente potentes para sua implementação.

No capítulo 6 são tecidas considerações sobre problemas e sugestões para desenvolvimentos futuros no Sistema Krisys.

5.5.2 Implementação da modelagem das informações

A modelagem da base de informações foi discutida ao longo deste capítulo, sendo que restam ser comentados alguns pontos relativos a sua implementação e utilização. Os objetos possuem, intrinsecamente, procedimentos — métodos — que atuam sobre seus atributos gerando novas informações. Para a utilização eficiente do modelo proposto, uma série de métodos realizam manipulações automáticas, prevendo as necessidades que ferramentas (CAD, CAPP, CAI, etc.) teriam ao fazer parte do AVEC. Portanto, estes são modelados especificamente para cada objeto onde se façam necessários. Nesta seção é apresentada a modelagem dinâmica utilizada na criação de uma peça no sistema.

O objetivo destes métodos é instanciar os objetos necessários à modelagem de uma peça, partindo de uma mensagem enviada por um usuário do AVEC. Ou seja, a partir da instanciação da peça são oferecidos serviços que realizam a criação e conexão dos objetos que modelam as informações complementares, conforme visto neste capítulo.

Qualquer objeto da base de informações é criado solicitando-se ao objeto "opambae" (objeto antecessor de todos os outros) a instanciação de uma classe determinada. A partir do pedido, o "opambae" cria o objeto e envia uma mensagem solicitando sua inicialização. Ao se auto-inicializar, um objeto dispara processo encadeado de criação dos objetos agregados. Desta forma todos os objetos necessários à representação da peça são inicializados. A figura 5.44 descreve a troca de mensagens entre os objetos. Note que no Krisys os procedimentos são tratados sequencialmente, logo o primeiro objeto a ser instanciado será o último a terminar sua inicialização.

O método New-obj, pertencente ao objeto "opambae", suporta a geração da identificação dos objetos (única na base de informações). Estas são formadas pelo nome da classe da qual o objeto é instância acrescido de um número sequencial (e.g. cilindro3, assento-rolamento150).

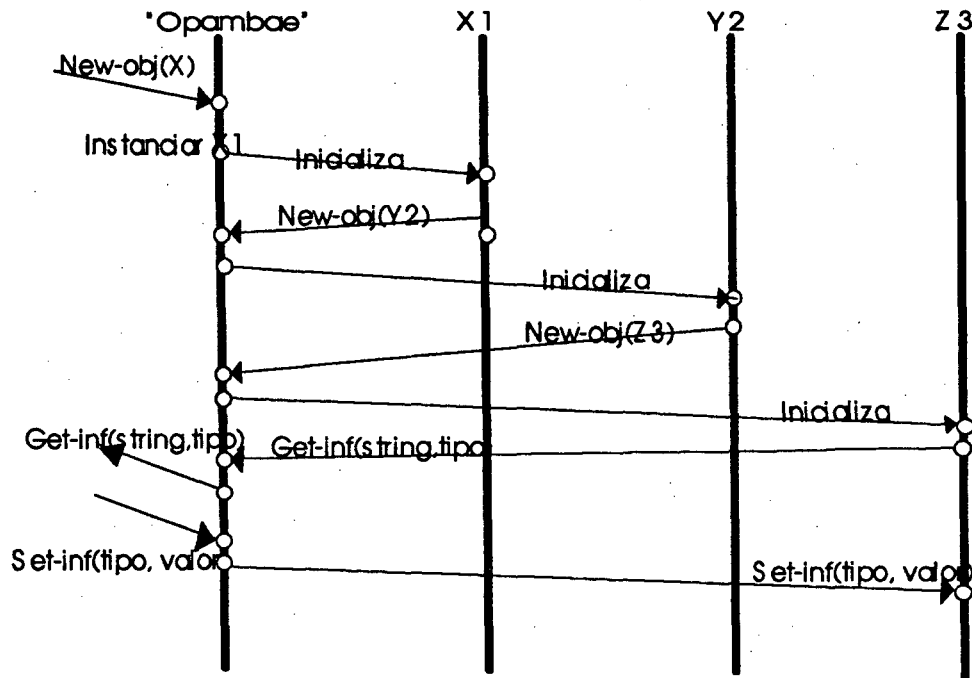


Figura 5.44 Geração encadeada de objetos

Durante o processo de geração dos objetos são enviadas mensagens ("Get-inf") ao "opambae" solicitando os valores para os atributos que estão sendo instanciados. O tratamento destas mensagens deve ser resolvido pelo objeto Getme (seção 4.3), uma vez que o pedido de geração do objeto partiu de alguma ferramenta integrante do AVEC.

Durante a criação dos objetos o sistema gerenciador está sempre verificando restrições semânticas pré-estabelecidas. Por exemplo, quanto às tolerâncias de forma permitidas para uma superfície. Ao ser solicitada uma tolerância de cilindricidade para uma superfície cônica, haverá uma quebra nas restrições, e a ação não terá efeito. É possível ainda verificar a cardinalidade do atributo, ou seja, uma "feature" só pode estar localizada em uma única posição, isto implica em apenas uma matriz de transformação agregada ao atributo local. Este caso é modelado através do aspecto "cardinality". A seguir são descritas algumas questões relevantes a nível de modelagem da base de informações.

Detalhes de implementação

- O método "inicializa" do objeto grupo dispara a instanciação da "feature" primitiva e de cada "feature" detalhe que o compõe. Uma vez que as "features" detalhe de um grupo podem ser em qualquer número, com repetidas instâncias da mesma classe, foi necessário criar uma estratégia para representá-las. Na definição da classe grupo-XX (uma especialização de grupo) é definido um atributo do tipo "ownslot"¹⁷ que contém o nome de outros atributos do mesmo tipo, onde está contida a descrição detalhada de quais e quantas "features" devem ser instanciadas (figura 5.45).

- A inicialização das "features" é feita a partir de um único método definido no objeto "features". Este possui conhecimento suficiente para diferenciar suas subclasses e inferir quais atributos devem ser inicializados durante a instanciação. O método está listado no anexo 1, item 4.3.

¹⁷ Ownslot e instanceslot são tipos de atributo. Um atributo ownslot, pode conter valores, não sendo, portanto herdado pelas classes que derivam daquele onde o atributo foi definido, enquanto que um instanceslot não pode conter valor, sendo repassado por herança para as subclasses. Numa instância de uma classe, todos os atributos são ownslots.

• As tolerâncias de posição e de restrição são definidas através de um método próprio que instancia as restrições, conectando-as aos objetos. O mesmo método é utilizado para peças, grupos e "features" (anexo1, itens 3.1, 4.2 e 4.3). Esta implementação caracteriza o polimorfismo (seção 3.2) dos objetos.

```
(DETALHES
  ((CHAVETAS-DESL-EXT CHANFROS-MENOR-EXT) OWNSLOT-TERMINAL
   EIXO-POLIA NIL))
(CHAVETAS-DESL-EXT
  (NIL INSTANCESLOT-COMPLEX-DOWN EIXO-POLIA
   (DIAMETRIC-REFERENCE (DETALHES-DE) (DETALHES-DE) EIXO-POLIA
    EIXO-POLIA)
   (CARDINALITY ([ 0 1 ]) ([0 1]) EIXO-POLIA EIXO-POLIA)
   (POSSIBLE-VALUES (INSTANCE-OF CHAVETA-DESL-EXT)
    (INSTANCE-OF CHAVETA-DESL-EXT) EIXO-POLIA EIXO-POLIA)))
(CHANFROS-MENOR-EXT
  (DIAMETRIC-REFERENCE (DETALHES-DE) (DETALHES-DE) EIXO-POLIA
   EIXO-POLIA)
  (CARDINALITY ([ 0 1 ]) ([0 1]) EIXO-POLIA EIXO-POLIA)
  (POSSIBLE-VALUES (INSTANCE-OF CHANFRO-MENOR-EXT)
   (INSTANCE-OF CHANFRO-MENOR-EXT) EIXO-POLIA EIXO-POLIA)))
```

Figura 5.45 Atributos do objeto grupo

• A "feature" furo pode conter uma superfície cônica no final do furo. Este cone é definido pelo valor da conicidade que caracteriza a "feature". Toda vez que este valor for atualizado é verificado se o cone ainda existe (quando a conicidade é zero, o final do furo é plano). Para tanto existe um "demon" associado ao atributo conicidade que se encarrega de atualizar a superfície se necessário. A figura 5.46 apresenta a implementação do "demon".

```
(CONICIDADE-FUNDO
  (NIL INSTANCESLOT-TERMINAL FURO-CILINDRICO
   (DEMON (D-CONIC-FURO-CIL) (D-CONIC-FURO-CIL) FURO-CILINDRICO
    FURO-CILINDRICO)
   (POSSIBLE-VALUES (REAL) (REAL) FURO-CILINDRICO
    FURO-CILINDRICO)
   (CARDINALITY ([ 0 1 ]) ([0 1]) FURO-CILINDRICO
    FURO-CILINDRICO)))

>(get-schema 'd-conic-furo-cil 'complete)

(D-CONIC-FURO-CIL (SUBCLASS-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (SUBSET-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (HAS-SUBCLASSES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (INSTANCE-OF ((PUT) OWNSLOT-TERMINAL GLOBAL NIL))
  (COMMENT (("created 1/3/1993 by krisys") OWNSLOT-TERMINAL GLOBAL
   NIL))
  (PUT-AFTER
   ((LAMBDA (SCHEMA ATTRIBUTE OLD NEW)
    (PRINT "d-conic-furo-cil")
    (IF (EQUAL (CAR NEW) 0.0)
     (LET ((FACE (SEND-MSG 'OPAMBAE 'NEW-OBJ-VAZIO '(SFACE))))
      (CONNECT FACE 'SUPERFICIE-DE SCHEMA))
     (LET ((CONICA (SEND-MSG 'OPAMBAE 'NEW-OBJ-VAZIO
      '(SCONICA))))
```

```

(CONNECT CONICA 'SUPERFICIE-DE SCHEMA)))
OWNMETHOD PUT
(COMMENT (ARGUMENTS SCHEMANAME ATTRIBUTENAME VALUE NEW-VALUE)
(HERITAGE) PUT PUT)))
(PUT-BEFORE
(NIL OWNMETHOD PUT
(COMMENT (ARGUMENTS SCHEMANAME ATTRIBUTENAME VALUE
NEW-VALUE)
(HERITAGE) PUT PUT)))
(USED-IN ((FURO-CILINDRICO FURO-CILINDRICO) OWNSLOT-TERMINAL DEMONS
NIL)))

```

Figura 5.46 Método para conicidade do furo

5.5.3 Exemplo de modelagem

A seguir é apresentada a modelagem da peça utilizada como exemplo na seção 5.2 — eixo para uma polia esticadora (figura 5.47). O anexo 3, contém a listagem descrevendo a interação entre os métodos que constroem a base o usuário desta. Nesta seção alguns modelos discutidos anteriormente, são exemplificados para a peça em análise.

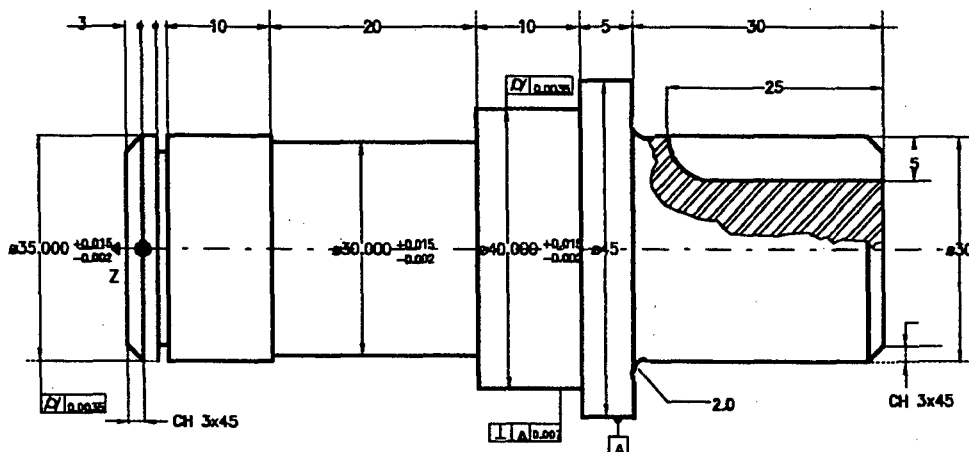


Figura 5.47 Polia esticadora

A listagem do anexo 3 demonstra o procedimento de criação de uma "feature" assento-rolamento, pertencente ao grupo eixo-suporte da peça. Através de procedimentos similares, todos os objetos que representam a peça vão sendo instanciados consecutivamente. A figura 5.48 apresenta o modelo simplificado, resultante da geração da peça.

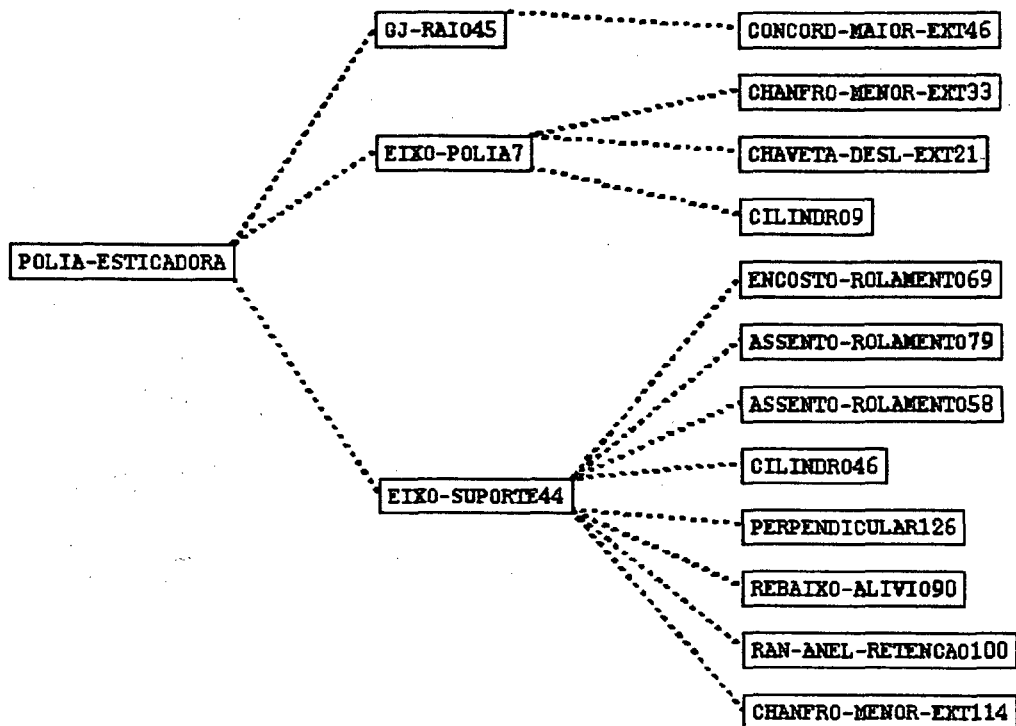


Figura 5.48 Objetos que representam a polia esticadora

O chanfro no extremo esquerdo da peça (figura 5.47) está posicionado a 3 mm do ponto de referência do grupo, e rotacionado de 180° sobre seu eixo Y. A figura 5.49 apresenta a estrutura de definição da "feature". Como discutido na seção 5.2.4, os objetos que representam material e acabamento (material212 e acabamento214) são agregados pela "feature", assim como por todas as superfícies. No momento que for necessário alterar o valor a nível de superfície, um novo objeto é instanciado.

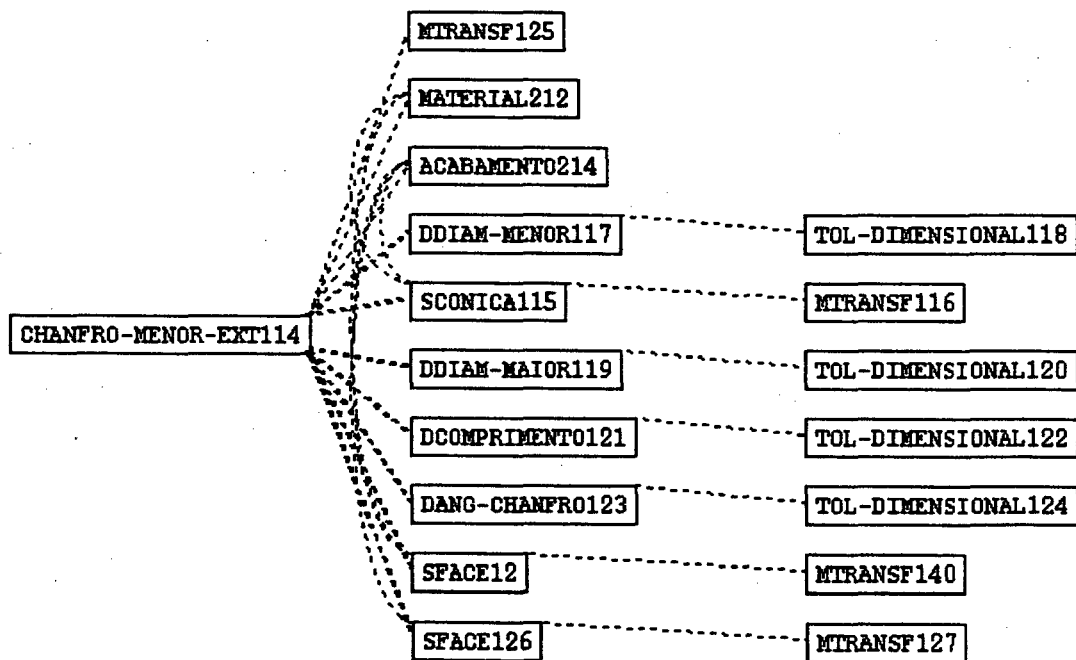


Figura 5.49 Estrutura da "feature" chanfro menor externo

A localização desta "feature", em relação ao grupo, é particularmente interessante, pois, devido a definição do ponto de referência (seção 5.4.2), a "feature" deve ser rotacionada de 180° em relação ao eixo Y, ao ser posicionada. Os atributos da matriz de rotação que modela esta situação estão representados na figura 5.50.

```
(MATRIZ ((#(1.0 0.0 0.0 0.0 0.0 -1.0 1.2246063538223773E-16 0.0 0.0
-1.2246063538223773E-16 -1.0 3.0 0.0 0.0 0.0 1.0))
OWNSLOT-TERMINAL MTRANSF
(CARDINALITY ([ 0 1 ]) (HERITAGE) MTRANSF MTRANSF)))
(VALORES ((0.0 0.0 3.0 180.0 0.0 0.0) OWNSLOT-TERMINAL MTRANSF
(CARDINALITY ([ 0 6 ]) (HERITAGE) MTRANSF MTRANSF)
(POSSIBLE-VALUES (REAL) (HERITAGE) MTRANSF MTRANSF)))
(MATRIZ-DE
((CHANFRO-MENOR-EXT114) OWNSLOT-PRIMITIVO-UP MTRANSF
(DIAMETRIC-REFERENCE (LOCAL) (HERITAGE) MTRANSF MTRANSF)))
```

Figura 5.50 Atributos da matriz de transformação

No desenho da figura 5.45 está representada uma tolerância de perpendicularidade entre as "features" assento de rolamento e encosto do rolamento (assento-rolamento58 e encosto-rolamento69). A representação desta restrição é representada pelo objeto da classe perpendicularidade, representado na figura 5.51. É importante notar, pelos atributos descritos na figura 5.52, que se mantém a relação de referência indicada no desenho. O atributo "obj-referencia" agrega a "feature" encosto de rolamento, representando que a restrição tomará esta como referência.



Figura 5.51 Representação de tolerância de perpendicularidade

```
((RESTRICAO-DE
((EIXO-SUPORTE44) OWNSLOT-COMPLEX-UP TOL-RESTRICOES
(DIAMETRIC-REFERENCE (TOL-RESTRICOES) (HERITAGE) TOL-RESTRICOES
TOL-RESTRICOES)))
(OBJ-REFERENCIA
((ASSENTO-ROLAMENTO58) OWNSLOT-COMPLEX-DOWN TOL-RESTRICOES
(DIAMETRIC-REFERENCE (REFERENCIA-DE) (HERITAGE) TOL-RESTRICOES
TOL-RESTRICOES)))
(OBJ-REFERENCIADO
((ENCOSTO-ROLAMENTO69) OWNSLOT-COMPLEX-DOWN TOL-RESTRICOES
(DIAMETRIC-REFERENCE (REFERENCIADO-DE) (HERITAGE) TOL-RESTRICOES
TOL-RESTRICOES)))
(VALOR ((0.00700000000000000001) OWNSLOT-TERMINAL TOL-ORIENTACAO
(POSSIBLE-VALUES (REAL) (HERITAGE) TOL-ORIENTACAO
TOL-ORIENTACAO)
(CARDINALITY ([ 0 1 ]) (HERITAGE) TOL-ORIENTACAO
TOL-ORIENTACAO))))
```

Figura 5.52 Atributos do objeto perpendicularidade

Capítulo 6

Krisys: resultados e sugestões

Order and simplification are the first steps toward the mastery of a subject. Thomas Mann

No desenvolvimento deste trabalho foi utilizado um Sistema Gerenciador de Bases de Conhecimento (KBMS) para suportar a modelagem das informações. O desenvolvimento do sistema Krisys — Knowledge Representation and Inference System — é responsabilidade do Departamento de Informática da Universidade de Kaiserslautern, Alemanha. Este capítulo tece considerações a respeito das facilidades de modelagem oferecidas cuja utilização foi descrita no capítulo 5. A título de sugestão são propostos incrementos na semântica de modelagem, de forma a facilitá-la. Uma descrição detalhada do sistema Krisys pode ser encontrada na seção 3.4.5 e nas referências bibliográficas.

6.1 Experiência de modelagem no Krisys

O KBMS Krisys utilizado na implementação do modelo de base de informações é a versão 2.7 do sistema. Na versão disponível na Universidade Federal de Santa Catarina este é construído sobre um Sistema Gerenciador de Bases de Dados (DBMS) convencional — INGRES, da Ingres System, versão universitária. Esta versão se caracteriza por ser um subconjunto da versão comercial, possuindo desempenho consideravelmente inferior. A plataforma de "hardware" utilizada é SUN-Sparc 2, com 16 Mbytes de memória RAM, rodando sobre o sistema operacional SUN-OS 4.0.1.

A representação das informações do Sistema de Manufatura no Krisys levou a utilização da quase totalidade dos recursos disponíveis no nível de engenharia¹⁸. Em geral, as facilidades oferecidas permitiram, de uma forma ou de outra, representar o relacionamento entre os objetos. Em alguns casos, quando não foi possível a modelagem direta da informação, esta se deu pela utilização de métodos e "demons".

Uma das características básicas do Krisys é a representação de sua estrutura de controle como objetos e atributos do sistema. Esta prática permite maior flexibilidade de modelagem, uma vez que se pode participar do controle dos objetos. A definição de aspectos em atributos pré-definidos ("instance-of", por exemplo) permite que um objeto dispare métodos automaticamente ao ser instanciado. Esta possibilidade facilita a modelagem da base como um todo.

Muitas dificuldades de modelagem foram superadas através do uso de atributos auxiliares de controle, internamente a uma classe. Em especial durante a instanciação em cadeia de objetos o uso de "ownslots" foi largamente utilizado.

¹⁸ O Krisys possui dois níveis de interação com o usuário: Nível de Engenharia e de Aplicação (seção 3.4.5).

Na modelagem das informações de mais alto nível, pouco especializadas, os objetos não possuem relação entre si, embora o contexto em que estão inseridas possua características próprias. Utilizando o conceito de conjunto é possível representar este tipo de relação.

Durante a geração de um conjunto de objetos encadeados é utilizado o conceito de transação, ou seja, uma operação só se torna efetiva na base de informações no momento que todas as operações pendentes são concluídas. Numa transação longa (por exemplo a geração de uma peça), facilidades como salvamento do estado dos objetos e "check-points", para retorno a determinadas situações, é importante.

Uma série de outros conceitos foram utilizados na representação das 550 classes de objetos que compõem o modelo. Da experiência apreendida, foram compilados alguns recursos semânticos que facilitariam sobremaneira a modelagem, Estas sugestões são descritas a seguir.

6.2 Krisys: sugestões

Durante a modelagem da base de informações para o Sistema de Manufatura necessitou-se de algumas facilidades de modelagem adicionais, visando permitir uma modelagem concisa e clara dos objetos. Em geral, foi possível representar a informação desejada, utilizando os recursos disponíveis com algum artifício de modelagem, mas com certeza a implementação seria mais poderosa se as facilidades descritas a seguir estivessem disponíveis.

"Demons" multivalorados

Um "demon" é um aspecto que vincula o disparo de um método à alteração do valor de um atributo. Neste sentido, é uma facilidade que permite modelar eficientemente procedimentos relacionados a atributos. A dificuldade está em disparar mais de um método para uma única alteração. O sistema só permite que o aspecto "demon" possua um método associado.

Desta forma a solução é criar um método que dispare outros métodos em seqüência. Considerando que um "demon" também é um objeto, representado no sistema, esta alternativa não permite uma clara visualização do que acontecerá quando a alteração for realizada, a não ser que o método seja listado e depurado.

"Diametric-reference"

A relação de agregação entre dois objetos é realizada a partir de atributos em cada um destes objetos. Estes atributos possuem um aspecto "diametric-reference" que armazena o nome do argumento de ligação no outro objeto. A necessidade de restringir o nome do atributo dificulta a modelagem da seguinte situação, bastante comum: um objeto possui dois atributos, uma dimensão diâmetro e outra comprimento. Ambas agregam o objeto dimensão que contém o valor das tolerâncias dimensionais, além do próprio valor da dimensão. de informações que sejam agregadas por diferentes objetos. A partir do modelo de representação atual isto não é possível, pois como se pode observar na figura 6.1, não seria possível definir o valor do "diametric-reference" de dimensão. Na implementação atual, utilizou-se um objeto para cada tipo diferente de dimensão, embora todos contenham a mesma estrutura.

```

SCHEMA_A:
(COMPIMENTO (NIL INSTANCESLOT-COMPLEX-DOWN SCHEMA_A
(Possible-values (INSTANCE-OF DIMENSION))
(DIAMETRIC-REFERENCE (DIMENSION-OF))))
(DIAMETRO (NIL INSTANCESLOT-COMPLEX-DOWN SCHEMA_A
(Possible-values (INSTANCE-OF DIMENSION))

```

```

(DIAMETRIC-REFERENCE (DIMENSION-OF))))
DIMENSION:
...
(DIMENSION-OF (NIL INSTANCESLOT-COMPLEX-UP DIMENSION
(DIAMETRIC-REFERENCE ( ????????)

```

Figura 6.1 Problema no uso de agregação

"Possible-values"+ "Cardinality"

A semântica do aspecto "possible-value" não possibilita representar uma cardinalidade associada a um valor possível para um atributo. Esta possibilidade resultaria em diversas simplificações na implementação realizada, principalmente para a agregação de elementos construtivos de níveis hierárquicos inferiores.

Tratamento de erros

Como já foi dito o Krisys permite modelar inúmeros casos de restrições, através dos aspectos dos atributos, mas a cada restrição quebrada, uma condição de erro é executada, interrompendo imediatamente o processo de instanciação em curso. Os resultados deste tratamento podem ser desastrosos, uma vez que, um erro durante uma transação elimina todas as alterações realizadas na base desde seu início. Como já citado anteriormente, alguma forma de salvamento de contexto em transações longas, é fundamental. Além disso, a quebra de restrições deveria retornar uma situação de erro possível de ser manipulada pelos métodos em execução.

Capítulo 7

Análise de resultados: conclusões

**Sonhos! Guarde-os bem,
até que teus pés pisem o chão.**

Neste capítulo são descritos e avaliados os resultados obtidos neste trabalho. Por fim, são sugeridas etapas futuras de desenvolvimento e tecidas conclusões.

7.1 Objetivo proposto

O objetivo principal do trabalho é o desenvolvimento de uma abordagem para a adequação do Sistema de Manufatura aos conceitos de Engenharia Concorrente. Para tanto, é proposta uma plataforma computacional que integra e descentraliza as informações relevantes nas diversas atividades que se desenvolvem no ciclo de vida de um produto. O desenvolvimento deste trabalho implica na manipulação de um largo espectro de conhecimentos. Dessa forma optou-se por especificar o sistema como um todo, implementando o modelo das informações a serem compartilhadas.

7.2 Resultados obtidos

A partir da análise do modelo de uma empresa metal-mecânica ideal [HAR84], foram identificados núcleos de informação no Sistema de Manufatura. Estes núcleos desenvolvem atividades e estabelecem um fluxo das informações entre eles. A análise identificou o elemento principal, sobre o qual se troca informações — o produto.

A especificação de uma plataforma computacional que auxilia na implantação de Engenharia Concorrente, baseou-se no fluxo de informações identificado. Portanto, o AVEC (capítulo 4) é o resultado da união de conhecimentos de diversas áreas computacionais com as necessidades provenientes da aplicação dos conceitos de CE no Sistema de Manufatura.

Uma vez que o produto é o elemento mais importante, e em torno do qual a maior parte das informações do Sistema de Manufatura gravita, foi necessário representá-lo de forma tal que fosse possível agregar tais informações. Assim, aquelas, provenientes das diversas atividades desenvolvidas durante o ciclo de vida do produto, estão unidas e disponíveis para consulta. A Tecnologia de "Features" é o elemento chave para a integração destas informações. A abordagem utilizada para "features" foi desenvolvida com base na literatura e adaptada, de forma apropriada, às necessidades do modelo.

A utilização de conceitos de Engenharia Concorrente traz inúmeros benefícios, discutidos na seção 2.2, em termos de eficiência, padronização e qualidade nas atividades de projeto. Através do AVEC, as vantagens se estendem para as demais atividades do Sistema de Manufatura, uma vez que as informações relevantes estão disponíveis por meio de uma chave de acesso comum.

Os conceitos que embasam o AVEC permitem que a implementação da plataforma ocorra em etapas, desde que se mantenha a independência entre os objetos. Esta característica é particularmente importante no momento da implantação, uma vez que permite um entrosamento gradual, tanto do usuário com o sistema, quanto entre as equipes de diferentes áreas da empresa que serão auxiliadas pelo AVEC. Neste sentido, uma série de atividades de treinamento devem ser realizadas com o pessoal envolvido.

Como dito anteriormente, a representação das informações é o elemento principal na implementação do AVEC. Por este motivo foi escolhido para ser o ponto de partida para a construção do ambiente como um todo. Nos capítulos 5 e 6 são tecidas considerações sobre a implementação propriamente dita. A utilização de orientação a objetos e Tecnologia de "Features" viabilizou a modelagem de informações bastante complexas, a partir de conceitos avançados de representação do conhecimento.

A utilização do sistema Krisys, além do intercâmbio com a Universidade de Kaiserslautern, viabilizou a implementação da modelagem utilizando os conceitos de orientação a objetos. No capítulo 6 são apresentadas sugestões para aumentar a capacidade de representação do sistema.

7.3 Análise dos resultados

Os resultados obtidos neste trabalho devem ser considerados um ponto de partida para o desenvolvimento de novos estudos. A partir desta especificação uma série de trabalhos devem ser desenvolvidos visando a implementação da plataforma como um todo. Neste sentido, na seção 7.4, são sugeridos desenvolvimentos que se adaptem ao conjunto.

O resultado obtido na modelagem da base de informações permitiu a representação de um produto por "features" com relação às informações de projeto e planejamento de processos. Embora outros trabalhos realizados [ECK88, TSA89, RAD91] atestem a validade do uso de uma "feature" única, contendo informações provenientes das diversas áreas do Sistema de Manufatura, novos estudos devem ser realizados visando definir uma quantidade maior de informações agregadas a "feature". Além disso, um sistema CAPP (em desenvolvimento), utilizando as mesmas "features" deve gerar um plano de processos sobre a mesma representação utilizada para projeto.

Atualmente, a base não possui qualquer facilidade para comunicação com aplicações externas, devendo este ser o passo seguinte a ser tomado. Por outro lado, ferramentas como um sistema CAPP generativo-iterativo podem ser desenvolvidas sobre a modelagem existente, uma vez que o KBMS utilizado modela regras de produção.

O uso de conceitos de sistemas abertos, distribuídos e heterogêneos, baseados no paradigma cliente-servidor, permite independência a nível de representação das informações em cada ferramenta (CAD, CAPP, CAI, etc.) do AVEC. Conforme descrito no capítulo 4, a implementação atual utiliza um KBMS, que emula os conceitos de sistemas distribuídos através da troca de mensagens entre os objetos. Essa abordagem permite a integração do modelo mesmo sem a existência de um HDBMS.

7.4 Sugestões de trabalhos futuros

Este trabalho prevê o prosseguimento dos estudos em três áreas básicas. Em primeiro lugar a implementação do AVEC deve ser encaminhada, executando trabalhos nas áreas de redes de computadores, protocolos para manufatura, sistemas especialistas e modelagem das informações restantes. Uma outra frente deve desenvolver trabalhos em cooperação com a indústria, visando validar a definição corrente de

"features" e expandir seu domínio. Por fim, estudos na área de HDBMS e agentes cooperantes devem ser iniciados visando futuras versões baseadas na efetiva utilização destes conceitos.

Conforme descrito na seção 4.3.1 são especificadas as necessidades para a definição de um protocolo para a troca de mensagens entre os objetos do AVEC. Estudos sobre MAP (Protocolo de Automação da Manufatura) e MMS (Especificação de Mensagens para a Manufatura) devem ser realizados no sentido de servir de base para a definição e implementação do suporte ao protocolo (objeto Getme).

As ferramentas que atuam no AVEC (CAD, CAPP, CAI, etc.) devem possuir facilidades para auxiliar a análise de um projeto com relação a suas especialidades (seção 4.6). Por este motivo sistemas especialistas devem ser utilizados na implementação. No mesmo sentido, um "software" de gerenciamento de ferramenta (GEFER, desenvolvido no GRUCON) deve ser redefinido de forma a dispor de informações sobre ferramentas e suas potencialidades de uso.

A interface do AVEC com o usuário é descrita na seção 4.5. Embora poucos estudos existam nesta área, esta facilidade é de fundamental importância em um ambiente de Engenharia Concorrente. Conceitos de padronização de interfaces e multimídia devem ser estudados visando definir o objeto Interface.

O modelamento de produtos baseado em Tecnologia de "Features" deve ser, através de aplicações reais, validado e expandido. Para tanto, deve ser considerado um domínio de peças proveniente da indústria. Desta forma, as informações necessárias para a definição das "features", como funcionalidade dos elementos construtivos, estarão disponíveis.

Conforme descrito no capítulo 4, o uso de sistemas especialistas, trocando informações entre si, constituem-se no desenvolvimento natural da abordagem proposta. Estudos na área de HDBMS, agentes cooperantes e sistemas especialistas, devem dar seguimento a este trabalho visando ampliar o espectro das informações representadas. Trabalhos nesta área podem ser desenvolvidos paralelamente à implementação das demais partes da versão atual do AVEC.

7.5 Conclusões

A Engenharia Concorrente, antes de mais nada, trata da integração e descentralização das informações. Uma vez que, estas circulam entre as pessoas, de pouco adianta a existência da tecnologia sem um trabalho de conscientização, intenso e abrangente, de seu uso e benefícios. Da mesma forma, a informação em si deve estar organizada e racionalizada, evitando assim, entre outras coisas, descontinuidade e incompatibilidade, que conduzem à perda de qualidade nas atividades.

A indústria, em geral, procura por soluções prontas, de fácil implantação e rápido retorno. Entretanto, a prática tem demonstrado que, no caso de tecnologias que utilizam conceitos tão abrangentes como Engenharia Concorrente, as soluções dificilmente estarão disponíveis para uso, exigindo portanto desenvolvimentos e adaptações adicionais. Além disso, a implantação de novas soluções vai depender muito de um trabalho conjunto, unindo treinamento de pessoal e sistemas flexíveis o suficiente para, não só se adaptarem às necessidades atuais da empresa, mas também, evoluir e acompanhar o surgimento de novas técnicas em busca de soluções para novos desafios que o futuro inevitavelmente trará. O AVEC aponta nesta direção.

As características de ambiente aberto, distribuído e heterogêneo atribuem, à proposta apresentada nesta dissertação, facilidades para, tendo como base a informação, ser configurável de acordo com a realidade da empresa. A flexibilidade que esta característica lhe atribue permite ainda que sua implantação seja gradual, acompanhando o crescimento do espírito de cooperação entre as áreas.

Atualmente, estão sendo desenvolvidos esforços de aprimoramento da proposta base, através de um acordo de cooperação com a indústria. Neste, são utilizados os conceitos básicos descritos ao longo deste trabalho.

O nível de automatização das tarefas tende a ser maior a medida que novas tecnologias de automação vão surgindo. O AVEC possui flexibilidade suficiente para seguir incorporando estes avanços, mas por mais automatizado que seja o sistema, a Engenharia Concorrente só trará resultados com a efetiva integração cooperativa entre as pessoas.

Capítulo 8

Referências Bibliográficas

[AGU91] Aguiar, M.W.C. Estratégias para a Implantação de Manufatura Integrada. Publicação Interna Fundação CERTI. Flórianópolis, Brasil. maio, 1991.

[AHM91] Ahmadi, J. Nayak, N. Ow, P.S. Wald, J. Key Technologies for CIM in the USA. Computer Applications in Production and Engineering Integration Aspects. Elsevier Science Publishers. North-Holland. 1991.

[BAR85] Barkocy, B.E. Zdeblinck, W.J. A Knowledge-based System for Machining Operation Planning. Internal Report. Metcut Research Associates Incorporated. Cinchinnati, USA. 1985.

[BED88] Bedworth, D. Li, R. A Framework for the Integration of CAD/CAPP. Computers and Industrial Engineering, Vol 14, Nro 4, 1988.

[BED91] Bedworth, D Henderson, M.R. Wolfe, P.M. Computer- Integrated Design and Manufacturing. McGraw-Hill, Inc. 1991. ISBN 0-07-004204-7.

[BIE90] Biegel, J. Gupta, U.G. Expert Systems in Manufacturing: Promisses and Perils. Computers in Industrial Engineering. Vol. 19. Nro. 1-4. 1990.

[BON87] Bond, A.H. Chang, J. Subplanning Methods in Automatic Process Planning. Journal of Manufacturing Systems. Internal Report. University of California. Los Angeles. 1987

[BON89] Bond, A.H. Chang, J. Subplanning Methods for Procces Planning. Computer in Engineering. Vol 1. 1989.

[BUT86] Butterfield, W.R. Green, M.K. Scott, D.C. Stoker, W.J. Part Features for Process Planning. CAM-I Report R-86-PPP-01, CAM-I, Arlington, Texas, November, 1986.

[CAM91] Camarinha-Matos, L.M. Osório, A.L. Lopes, L.S. Information Management System - tool specification. Universidade Nova de Lisboa. Departamento de Informática. Relatório Interno. 1991.

[CAM92] Camarinha-Matos, L.M. Osório, A.L. Lopes, L.S. Integração de Informação para arquiteturas CIM. Universidade Nova de Lisboa. Departamento de Informática. Relatório Interno. 1992.

[CAR92] Carrol, D. Bagert, D. DOOCIMS: the next generation of distributed Control Systems. Proceedings of the International Conference on Object-Oriented Manufacturing Systems. University of Calgary. Alberta. Canada. 1992.

[CAS85] Casanova, M.A. Moura, A.V. Princípios de Sistemas de Gerência de Bancos de Dados Distribuídos. Editora Campus. Rio de Janeiro. Brasil. 1985. ISBN 85-7001-263-2.

[CHA84] Chang, TC; Wysk, R.A. Integrating CAD and CAM through automated process planing. International Journal of Production Research. Vol 22 Nro 5. pag 877-894. 1984

[CHE89] Chen, C-S. A Form-Feature Oriented Coding Scheme. Computers in Industrial Engineering. Vol 17. pp 227-233. 1989.

[CHE92] Chen,C-S. Swift,F. Lee,S. Ege,R. Shen,Q. Development of a Feature-Based and Object-oriented Concurrent Engineering System. Proceedings of the International Conference on Object-Oriented Manufacturing-systems. University of Calgary, Alberta, Canada. 1992.

[CHR86] Chryssolouris,G. Wright,K. Knowledge-Based Systems in Manufacturing. Annals of CIRP. Vol. 35. Nro 2. 1988.

[CHU88] Chung,J.C. Cook,R.L. Patel,D. Simmons,M.K. Feature Based Geometry Construction for Modeling Reasoning. ASME Computers in Engineering Conference. San Francisco. august, 1988.

[CON88] Consalter,L. Bochs,L. Ferreira,A. Programação de Máquinas NC Utilizando Linguagem de Alto Nível Apoiada por Banco de Dados. VIII Seminário de Comando Numérico e Automação Industrial (SCNB-SOBRACON). agosto, 1988.

[COR91] Corbett,J. Woodward,J.A.J. A CAD Integrated Knowledge Based System for the Design of Die Case Component. Annals of CIRP, Vol. 40. Nro 1. 1991.

[COS92] César da Costa. Terceirização: novos rumos para a automação industrial. Automação e Indústria. Nro 43. novembro/dezembro, 1992.

[CUN88] Cunningham,J.J. Dixon,J.R. Design with Features: The Origin of the features. ASME Computers in Engineering Conference. San Francisco. august, 1988.

[DE89] DeBloch,S. Härder,T. Mattos,N.M. Mitschang,B. KRISYS: KBMS Suport for Better CAD Systems. Internal Report. University of Kaiserslautern. Kaiserslautern. Germany. 1989.

[DE91] DeBloch,S. Hübel,C. Mattos,N.M. Sutter,B. Handling Functional Constraints of Technical Modeling Systems in a KBMS Environment. Internal Report. University of Kaiserslautern. Kaiserslautern. Germany. 1991.

[DUL91] Dulli,S. Milanese,V. A Database Design for Graphical Models SIGPLAN Notices. Vol 25. Nro 12. 1991.

[ECK88] Eckersley,J.S. Features in Design and Manufacture Internal Report. Loughborough University of Technology. Loughborough. England. 1991.

[ELM90] Elmagarmid,J.K. Pu,C. Introduction to Heterogeneous Databases. ACM Computing Surveys. Vol 22. Nro 3. september, 1990.

[ELS88] Elshennawg,A. The Role of Inspection in Automated Manufacturing. Computers in Industrial Engineering. Vol 17. 1989.

[ERV86] Erve,A.H. Kals,H.J.J. XPLANE, a Generative Computer Aided Process Planning Systema for Part Manufacturing. Annals of the CIRP. Vol 35. Nro 1. 1986.

[ESP89] Esprit Consortium AMICE - Open System Architecture for CIM [CIM-OSA], Springer Verlag. Germany. 1989.

[FER91] Ferreira,A.C. Zendron,P.C. IAGE: Uma proposta de sistema CAPP orientado por "features". Simpósio sobre CAE/CAD/CAM. SOBRACON. São Paulo. Brasil. novembro, 1991.

[FUO91] Fuoco, T.H. Furtado,M.R. Ferramentarias brasileiras já usam e aprovam sistemas CAD/CAM. Máquinas e Metais. Nro 310. novembro, 1991.

[GAR84] Gardiner, K. Manufacturing as a System. WESTEC'84 Conference - Socyte of Manufacturing Engineers. Los Angeles. USA. march, 1984.

[GIN89] Gindy, N. A hierarchical Estructure for Form-feature. *International Journal of Production Research*. Vol 27. Nro 12. december 1989.

[GRA88] Graves, G.R. Yelamanchilit, B. Parks, C.M. An interface architecture for CAD/CAPP integration using knowledge-based systems and feature recognition algorithms. *International Journal in Computer Integrated Manufacturing*. Vol 1. Nro 2. 1990.

[GRO87] Groover, M.P. *Automation, Production Systems, and Computer Integrated Manufacturing*. Printice-Hall, Inc. USA. 1987. ISBN 0-13-054652-6

[GUT90] Gutiérrez, E. Rodríguez, M. Sistema de Apoyo a la Generacion de Gamas de Fabricacion en un Ambiente Integrado CAD/CAM. Internal Report. Universidad de los Andes. Facultad de Ingenieria. Bogotá. Colombia. 1990.

[HAM88] Ham, I. Computer-Aided Process Planning: The present and the Future. *Annals of the CIRP*. Vol 37. Nro 2. 1988.

[HAR84] Harrington, J. Understanding the Manufacturing Process - key to succesful CAD/CAM implementation. Marcel Dekker, Inc. USA. 1984. ISBN: 0-8247-7170-2

[HAY83] Hayes-Roth, F. Watermann, D.A. Lenat, D.B. *Building Expert Systems*. Addison-Wesley Publishing Company. Inc. Massachusetts, USA. 1983. ISBN 0-201-10686-8.

[HEN88] Henderson, M.R. Chang, G.J. FRAPP: Automated Feature Recognition and Process Planning from Solid Model Data. Internal Report. Arizona State University. Arizona. USA. 1988.

[HER91a] Hernandez, J.T. Rodrigues, M. Ferreira, A.C. Zendron, P.C.D. Programação de Processo de Fabricação para Peças Metal-Mecânicas. Relatório Final - CNPq. julho 1991.

[HER91b] Hernandez, J.T. Rodrigues, M. Ferreira, A.C. Zendron, P.C.D. IAGE: A Feature Oriented CAPP System. CAD Forum. august, 1991.

[HIT79] Hitomi, K. *Manufacturing Systems Engineering*. Taylor & Francis Ltd. London. UK. 1979.

[HOU91] Hou, T. Wang, B. Integration of a CAPP System and a FMS. *Computers and Industrial Engineering*, Vol 20. Nro 2. 1991.

[HUD89] Hudson, S.E. King, R. Cactis: A self-adaptative, concurrent implementation of an Object-oriented Database Management System. *ACM Transaction on Database Systems*. Vol 14. 1989.

[HYE91] Hyeon, H.J. Hamid, R.P. Wong, J.P. Concurrent Engineering: The manufacturing philosophy for the 90's. *Computers in Industrial Engineering*. Vol 21. Nros 1-4. 1991.

[IPE92] IPESI, redação Situação atual da pesquisa em Inteligência Artificial. IPESI. março, 1992.

[IWA88] Iwata, K. Applications of Expert Systems to Manufacturing in Japan. *The International Journal of Advanced manufacturing Technology*. Vol 3. Nro 3. 1988.

[JAC91] Jacobs, R; Mathieson, K; Muth, J; Hancock, T. A rule-based system to generate NC programs from CAD exchange files. *Computers and Industrial Engineering*, vol20. Nro 02. 1991.

[JAR86] Jared, G.E.M. Feature Recognition & Expert Systems for Operation Planning in NC Machining. Internal report. Dept. of Applied Computing and Mathematics. Cranfield Institute of Technology. Cranfield, Bedford. UK. 1986.

[JAR89] Jared, G.E.M. The Feature Recognition Battle - Latest from the front. Internal report. Dept. of Applied Computing and Mathematics. Cranfield Institute of Technology. Cranfield, Bedford. UK. 1989.

[JOS84] Joshi,S. Chang,T.C. Liu,C.R. An automated PP system structure based on AI. Internal Report. School of Industrial Engineering. Purdue University. Indiana. USA. 1984.

[JOS86] Joshi,S. Chang,T.C. Liu,C.R. Process planning formalization in an AI framework. Artificial Intelligence. Vol 1. Nro 1. 1986.

[JUR90] Juri,A.H. Saia,A. Pennington,A. Reasoning about machining operations using feature-based models. Int. J. Prod. Res. Vol 28. Nro. 1. 1990.

[KEM85] Kempf,K.G. Manufacturing and Artificial Intelligence. Robotics. Elsevier Science Publishers B.V. North-Holland. 1985.

[KIM90] Kim,W. Banerjee,J. Chou,H. Garza,J. Object-oriented database support for CAD. Computer-aided Design. Vol 22. Nro 8. 1990.

[KIN88] Kingsley,N. Bryan,G.C. CIM - the technology's not the real problem! Computer Aided Engineering Journal. february, 1988.

[KRA88] Kramer,T. A parser that converts a boundary representation into a features representation. Internal Report. National Institute of Standards and Technology and Research Associate, Catolic University Galthersburg, MD 20899. october, 1988

[LEE87] Lee,Y.C. Machine Understanding of CSG: Extraction and Unification of Manufacturing Features. IEEE CG&A. january 1987.

[LEM92] Lemos,A.S. Guia do Usuário do Krisys. Publicação Interna. Universidade Federal do Rio Grande do Sul. Porto Alegre. Brasil. 1992. Versão preliminar.

[LIT90] Litwin,W. Mark,L. Roussopoulos,N. Interoperability of Multiple Autonomous Databases. ACM Computing Survey. Vol 22. Nro 3. September 1990.

[LIT91] Li,R. Taur,B. Shijur,H. A Two-stage Feature-based Design System International Journal of Production Research. Vol. 29, Nro. 1. january 1991.

[LOR93] Lorini,F.J. Tecnologia de Grupo e Organização da Manufatura. Editora da UFSC. Florianópolis. Brasil. 1993.

[LUC92] Luckmann,C.L. Lensch,V. Pesquisa mostra qual a atual base instalada brasileira. Máquinas e Metais. Nro 313. fevereiro, 1992.

[MAT88] Mattos,N.M. A Multi-layered Prototype KBMS Supporting Knowledge Independence. Proc. Int. Computer Science Conference '88 - Artificial Intelligence Theory and Application. Hongkong. december, 1988.

[MAT89] Mattos,N.M. An approach to knowledge base management. Doctoral Thesis. University of Kaiserslautern. Kaiserslautern. Germany. 1989.

[MAT90] Mattos,N.M. KRISYS - a KBMS Supporting Development and Processing of Knowledge-based Applications in Workstation/Server Environments. Internal Report. University of Kaiserslautern. Kaiserslautern. Germany. 1990.

[MOR91] Moreira,N.P. Ferreira,A.C. Tecnologia de "Features" em CIM: Aplicações e Ferramentas. Simpósio sobre CAE/CAD/CAM. Sobracon. São Paulo. Brasil. novembro, 1991.

[MOR92a] Moreira,N.P. Ferreira,A.C. Ambiente Aberto Visando CIM: uma proposta de base. Anais do I Congresso Internacional de Computação Gráfica. Sobracon, São Paulo. Brasil. julho, 1992.

[MOR92b] Moreira,N.P. Ferreira,A.F. Production System's integration: a feature approach. VII International Symposium in Computer Applications. Antofagasta. Chile. novembro, 1992.

[MOU92] Moura,A. Descrição e os componentes da arquitetura cliente/servidor. Mundo Unix. Nro 41. agosto 1992.

[NIE88] Nierstrasz,O. A survey of Object-Oriented Concepts. in Object-Oriented Concepts, Applications and Databases, ed. Kin,W. Lochovsky,F. Addison-Wesley. MA. USA. 1988.

[OLL91] Olling,G.J. CIM Status and Directions in the USA. Computer Applications in Production and Engineering Integration Aspects. Elsevier Science Publishers. North-Holland. 1991.

[OPI71] Opitz, H. Wiendahl, P. Group technology and manufacturing systems for small and medium quantify production. Int. J. Prod. Res. Vol 9. No 1. 1971.

[OSO91] Osório, A.L. Camarinha-Matos, L.M. Connection between CIM-PLATO Tools and the Information Management System - a C language implementation specification. Universidade Nova de Lisboa. Departamento de Informática. Relatório Interno. 1992.

[PEL91] Pelletier,M.C. CIM - The integration of manufacturing and information systems. Computers and Industrial Engineering. Vol 21. Nro 1-4. 1991.

[PHA91] Pham,D. Martin,K. Khoo,L.A. Knowledge-based pre-processor generator for CMM. International Journal of Production Research, Vol 29. Nro 4. april 1991.

[PRA85] Pratt,M.J. Wilson,P.R. Requirements for Support of Features in a Solid Modeling System, Final Report R-85- ASPP-01, CAM-I. Arlington. Texas. 1985.

[RAB91] Rabelo,R.J. Camarinha-Matos,L.M. Escalonamento Dinâmico: Estado Actual e Tendências. Relatório Interno. Universidade Nova de Lisboa/Departamento de Informática. Monte da Caparica. Portugal. 1991.

[RAD91] Rados,G.J.V. Manufacturing Code Generation for Rotational Parts in a Feature Based Product Modelling Environment. Doctoral Thesis. Loughborough University of Technology. Loughborough. UK. 1991.

[RAM92] Ramamurthy, K.; King, WR. Computer Integrated Manufacturing: an exploratory study of key organizational barriers. OMEGA - Int. J. of the Manag. Soc. Vol 20. Nro 4. 1992.

[REI92] Reimann, M. D.; Huq,F. A Comparative Analisis Approach for Evaluating the Effect that Concurrent Engineering has on Product Life Cycle Cost. Flexible Automation and Information Management - FAIM. july, 1992.

[RIC88] Rich,E. Inteligência Artificial. Editora McGraw-Hill, Ltda. 1988. ISBN 0-07-45-364-2.

[RIH92] Rihl,A.P. Conexão de redes heterogêneas é destaque no Networld. Computerworld. Vol 1. Nro 2. outubro, 1992.

[ROD91] Rodrigues,M. Hernandez,J.T. Gutierrez,E. Ferreira,A.C. Zendron,P.C. IAGE: A feature oriented CAPP system. 6th International Conference on CAD/CAM, Robotics and Factory of the Future. London. august, 1991.

[ROL89] Roller,D. Design by Features: An approach to High Level Shape Manipulation. Computers in Industry. Vol 12. 1989.

[ROS90] Rosa,F.D. Gálvez,J.A. Porras,O. Diseño Asistido por Computador de Sistemas Mecanicos usando Modelage por Rasgos. Internal Report. Universidad de los Andes. Facultad de Ingenieria. Bogotá. Colombia. 1990.

[ROS92] Rosa, E. New Modeller. Conceitos para um ambiente integrado de desenvolvimento de produtos para sistemas CAE/CAD/CAM.. Internal Report. Universidade Federal de Santa Catarina. Departamento de Engenharia Mecânica. Florianópolis. Brasil. 1992.

[ROS93] Rosa, E. Faria, P. Moreira, N. Vieira, R. New-Modeller. Uma base para a implementação dos conceitos de Engenharia Simultânea em um ambiente computacional. Anais do Congresso Brasileiro de Engenharia Mecânica. Brasil 1993.

[ROZ90] Rozenfeld, H. Um sistema CAPP modular para indústrias pequenas e médias. Máquinas e Metais. março, 1990.

[RYA91] Ryan, B. AI's Identity Crisis. Byte. January 1991.

[SAH90] Sahay, A. Grivos, G. Parks, C.M.A. Methodology for Recognizing Features in Two-dimensional cylindrical part design. International Journal of Production Research. Vol 28. Nro. 8. August 1990.

[SAU92a] Sauv e, J. Fatores que motivam a implementa o da computa o distribu da. Mundo Unix. Nro 38. maio 1992.

[SAU92b] Sauv e, J. O que   computa o distribu da e por que precisamos dela? Mundo Unix. Nro 39. junho 1992.

[SAU92c] Sauv e, J. Vantagens e benef cios da computa o distribu da. Mundo Unix. Nro 40. julho 1992.

[SCH87] Schildt, H. Artificial Intelligence using C. Osborne McGraw Hill. Berkeley. California. 1987. ISBN 0-07-881255-0

[SCH90] Sch tzer, K. Hintz, A. Modelador Geom trico Orientado pela Fabrica o. Relatório Interno, traduzido. Institut f r Produktionstechnik und Spanende Werkzeugmaschinen. Darmstadt. BRD.

[SCH91] Schopbach, S.C. Key to Concurrent Engineering: Managing Product Data. Automation. Vol 38. Nro 8. August, 1991.

[SHA86] Shah, J.J. A Scheme for CAD-CAPP Integration Through Group Technology. GE Corporate R & D Report, Automation Systems Lab. August 1986.

[SHA88a] Shah, J. Rogers, M. Functional Requirements and Conceptual Design of the Feature-Based Modeling System. Computer-Aided Engineering Journal. February, 1988.

[SHA88b] Shah, J. Sreevalsan, P. Rogers, M. Billo, R. Mathew, A. Current Status of Features Technology. CAM-I Report R-88-GM-04.1. November, 1988.

[SHA88c] Shah, J. Rogers, M.T. Feature based modelling shell: design and implementation. Internal Report. Arizona State University. Department of Mechanical and Aerospace Engineering. Tempe, Arizona, USA. 1988.

[SHA88d] Shah, J. Bhatnagar, A. Hsiao, D. Feature Mapping and Application Shell. Internal Report. Arizona State University. Department of Mechanical and Aerospace Engineering. Tempe, Arizona, USA. 1988.

[SHA91] Shah, J. Assessment of Feature Technology. Computer Aided Design. Vol 23. Nro 5. June 1991.

[SSE90a] Ssemakula, M. Process Planning Systems in the CIM Environment. Computers and Industrial Engineering, Vol 19. 1990.

[SSE90b] Ssemakula,M. Satsangi,A. Applications of PDES to CAD/CAM Integration. Computers and Industrial Engineering, Vol18. Nro4. 1990.

[STA83] Stanley,S.M. Henderson,M.R. Anderson,D.C. Using syntatic pattern recognition to extract feature information from a solid geometric database. Computers mech. Enging. Vol 2. Nro 2. 1983.

[STE86] Stefik,M. Bobrow,D. Object-Oriented Programming: Themes and Variations. The AI Magazine. january 1986.

[SUN90] Sun Microsystems, Inc. Network Programming Guide. USA. 1990.

[TAN90] Tang,X.Q. Davies,B.J. Knowledge Representation for Inspection Planning in EXCAP. Proceedings of the 28th International Matader Conference. Manchester. UK. april 1990.

[TAR92] Taraman, K. Sem produtividade as empresas perdem suas fatias de mercado. Maquinas e Metais. Nro 321. outubro, 1992.

[TEN81] Tenenbaum, A.S. Computer Networks, Prentice-Hall Inc., Englewood Cliffs. New Jersey. 1981.

[TER92] Terpeny, J.P. Deisenroth, M.P. A Concurrent Engineering framework: three basic components. Flexible Automation and Information Management - FAIM. july, 1992.

[THO89] Thomas,D. What's in an Object? Byte. march, 1989.

[THO90] Thomas,G. Thompson,G. Chung,C. Barkmeyer,E. Carter,F. Templeton,M. Fox,S. Hartman,B. Heterogeneous Distributed Database Systems for Production Use. ACM Computing Survey. Vol 22. Nro 3. September, 1990.

[TSA87] Tsang,J.P. Planification Par Combinaison de Plans Application à la Génération de Gammes d'Usinage. These Docteur. Institute National Polytechnique de Grenoble. 1987.

[TSA89] Tsang,J.P. Brissaud,D. A feature-approch to process planning. International Engineering Conference and Exposition-ASME. California. USA. 1989.

[TSU89] Tsuzuki,M.S.G. Miyagi,P.E. Reconhecimento Automático de "Features" Baseado na Representação B-rep. Anais do X Congresso Brasileiro de Engenharia Mecânica. 1989.

[TUI93] Tuijnman,F. Afsarmanesh,H. Sharing Complex Objects in a Distributed PEER environment. 13th IEEE International Conference on Distributed Computing Systems. Pittsburgh, Pennsylvania. USA. May, 1993.

[UNG88] Unger,M.B. Ray,S.R.Feature-based Process Planning in the AMRF. Internal Report. National Bureau of Standards. Gaithersburg. Maryland. USA. 1988.

[WAG92] Wagner,R. Scegin,D.N. Development of Object Oriented Manufacturing Message Specification Application. Proccedings of the International Conference on Object-Oriented Manufacturing Systems. University of Calgary. Alberta. Canada. 1992.

[WAL89] Walton, A.D. A introdução da manufatura integrada por computador na Jaguar Cars. Boletim Sobracom. Nro 46. setembro/outubro 1989.

[WAN87] Wang,H. Wysk,R. Intelligent Reasoning for PP. Computer in Industry. Vol 8. 1987.

[WIN84] Winston,P.H. Artificial Intelligence. Addison-Wesley Publishing Company, Inc. Massachusetts, USA. 1984. ISBN 0-201-08259-4.

[WRI89] Wright,P.K. Bourne,D.A. **Manufacturing Intelligence**. Addison-wesley Publishing Company Inc. Massachusets, USA. 1989.

[WUW90] Wu,D. Wusk,R. **An Inference Structure for Control and Scheduling of Manufacturing System**. Computers and Industrial Engineering, Vol. 18. Nro. 3. 1990.

[YEO91] Yeo,S.H. Wong Y.S. Rahman, M. **Integrated knowledge-based machining system for rotational parts**. International Journal in Production Research. Vol. 29, Nro 7. 1991.

Anexo 1

Representação dos objetos

Neste anexo são apresentados a modelagem dos principais objetos do sistema, referenciados no capítulo 5.

Alguns esclarecimentos são necessários no sentido de facilitar a leitura deste anexo, assim os primeiros objetos são comentados servindo de explicação para os demais.

1 Objeto Opambae

- 1 - A primeira linha dos objetos declaram seu nome. Neste caso Opambae.
(OPAMBAE (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL)))
- 2 - As linhas a seguir descrevem o relacionamento entre este objeto e os demais existentes na base de conhecimento.
(SUBCLASS-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-SUBCLASSES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-SUBSETS
((FACILIDADES ENLACE FUNC-TECNOL) OWNSLOT-TERMINAL GLOBAL
NIL))
(SUBSET-OF ((SETS) OWNSLOT-TERMINAL GLOBAL NIL))
- 3 - Comentário padrão do Krisys.
(COMMENT ("created 1/3/1993 by krisys") OWNSLOT-TERMINAL
GLOBAL NIL))
- 4 - A seguir é declarado um método. Seu nome é new-obj, e recebe como parâmetro uma classe de objetos que deve ser instanciado.
(NEW-OBJ ((LAMBDA (TIPO)
(CHANGE-ATTRIBUTE-VALUE SELF 'ULT-OBJ
(LIST (+ 1
(CAR (GET-ATTRIBUTE SELF 'ULT-OBJ
'VALUE))))))
(LET ((OBJ (READ-FROM-STRING
(CONCATENATE 'STRING (STRING TIPO)
(WRITE-TO-STRING
(CAR
(GET-ATTRIBUTE 'OPAMBAE 'ULT-OBJ
'VALUE))))))
(CREATE-SCHEMA (LIST OBJ))
(CONNECT OBJ 'INSTANCE-OF TIPO)
(SEND-MSG OBJ 'INICIALIZA
OBJ))
OWNMETHOD OPAMBAE
(COMMENT (METODO QUE CRIA UM NOVO OBJETO PARA SER
CONECTADO)
(METODO QUE CRIA UM NOVO OBJETO PARA SER
CONECTADO)
OPAMBAE OPAMBAE)))
(ULT-OBJ ((23) OWNSLOT-TERMINAL OPAMBAE NIL))
NEW-OBJ-VAZIO
(LAMBDA (TIPO)
(CHANGE-ATTRIBUTE-VALUE SELF 'ULT-OBJ
(LIST (+ 1
(CAR (GET-ATTRIBUTE SELF 'ULT-OBJ 'VALUE))))))

```
(LET ((OBJ (READ-FROM-STRING
  (CONCATENATE 'STRING (STRING TIPO)
  (WRITE-TO-STRING
  (CAR
  (GET-ATTRIBUTE 'OPAMBAE 'ULT-OBJ
  'VALUE))))))
  (CREATE-SCHEMA (LIST OBJ)
  (CONNECT OBJ 'INSTANCE-OF TIPO)
  (SEND-MSG OBJ 'INICIALIZA-VAZIO)
  OBJ))
OWNMETHOD OPAMBAE NIL)))
```

2 Informações de Facilidades

2.1 Ferramentas

```
(FERRAMENTAS (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(SUBSET-OF ((FACILIDADES) OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-SUBSETS ((ELEMENTOS-FRR) OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-SUBCLASSES
  ((FRR-FURACAO FRR-FRESAMENTO FRR-TORNEAMENTO) OWNSLOT-TERMINAL
  GLOBAL NIL))
(SUBCLASS-OF ((CLASSES) OWNSLOT-TERMINAL GLOBAL NIL)))
(COMMENT ("created 1/3/1993 by krisys") OWNSLOT-TERMINAL GLOBAL
NIL))
```

- 5 - Os slots a seguir são definidos pelo usuário para caracterizar o objeto. A estrutura de um slot é a seguinte: (NOME (VALOR TIPO-SLOT OBJETO-ORIGEM ASPECTOS)). Sendo que tipo-slot significa que se for instanceslot não possui valor pois deverá servir de padrão para as instâncias. Se for ownslot tem valor associado e não será instanciado. Se um slot é do tipo terminal então não agrega nenhum outro objeto. Caso contrário, se for primitive ou complex é um slot de agregação.

IMPORTANTE: A origem do slot é aquela onde foi definido, sendo herdado por suas subclasses. Por exemplo, os slots a seguir são definidos no próprio objeto ferramentas.

```
(MATERIAL (NIL INSTANCESLOT-TERMINAL FERRAMENTAS NIL))
(FRR-OPERACAO (NIL INSTANCESLOT-TERMINAL FERRAMENTAS NIL))
(FRR-MAQUINA (NIL INSTANCESLOT-TERMINAL FERRAMENTAS NIL))
(QUANTIDADE (NIL INSTANCESLOT-TERMINAL FERRAMENTAS NIL))
(ID (NIL INSTANCESLOT-TERMINAL FERRAMENTAS NIL))
```

Exemplo: especialização de ferramentas para torneamento

```
(FRR-TORNEAMENTO (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(SUBSET-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(ID (NIL INSTANCESLOT-TERMINAL FERRAMENTAS NIL))
(QUANTIDADE (NIL INSTANCESLOT-TERMINAL FERRAMENTAS NIL))
(FRR-MAQUINA
  (NIL INSTANCESLOT-TERMINAL FERRAMENTAS
  (POSSIBLE-VALUES (INSTANCE-OF TORNO) (INSTANCE-OF TORNO)
  FRR-TORNEAMENTO FRR-TORNEAMENTO)))
(FRR-OPERACAO (NIL INSTANCESLOT-TERMINAL FERRAMENTAS NIL))
(MATERIAL (NIL INSTANCESLOT-TERMINAL FERRAMENTAS NIL))
(COMMENT ("created 1/3/1993 by krisys") OWNSLOT-TERMINAL GLOBAL
NIL))
(DIST-PNT-FRR
  (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
  (POSSIBLE-VALUES (REAL) (REAL) FRR-TORNEAMENTO
  FRR-TORNEAMENTO)
  (CARDINALITY (( 0 1 )) ((0 1)) FRR-TORNEAMENTO
  FRR-TORNEAMENTO)))
(COMPR-ARESTA
  (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
  (POSSIBLE-VALUES (REAL) (REAL) FRR-TORNEAMENTO
  FRR-TORNEAMENTO)
  (CARDINALITY (( 0 1 )) ((0 1)) FRR-TORNEAMENTO
  FRR-TORNEAMENTO)))
(ANG-INCL-ARESTA
  (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
  (POSSIBLE-VALUES (REAL) (REAL) FRR-TORNEAMENTO
  FRR-TORNEAMENTO)
  (CARDINALITY (( 0 1 )) ((0 1)) FRR-TORNEAMENTO
```

FRR-TORNEAMENTO)))
 (ANG-SAIDA
 (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
 (POSSIBLE-VALUES (REAL) (REAL) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)
 (CARDINALITY ((0 1)) ((0 1)) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)))
 (ANG-ESQUINA
 (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
 (POSSIBLE-VALUES (REAL) (REAL) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)
 (CARDINALITY ((0 1)) ((0 1)) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)))
 (ANG-DIR-SECUND
 (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
 (POSSIBLE-VALUES (REAL) (REAL) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)
 (CARDINALITY ((0 1)) ((0 1)) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)))
 (ANG-DIR-ARESTA
 (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
 (POSSIBLE-VALUES (REAL) (REAL) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)
 (CARDINALITY ((0 1)) ((0 1)) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)))
 (ANG-CHANFRO
 (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
 (POSSIBLE-VALUES (REAL) (REAL) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)
 (CARDINALITY ((0 1)) ((0 1)) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)))
 (RAIO-ESQUINA
 (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
 (POSSIBLE-VALUES (REAL) (REAL) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)
 (CARDINALITY ((0 1)) ((0 1)) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)))
 (LARGURA-PORTAFRR
 (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
 (POSSIBLE-VALUES (REAL) (REAL) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)
 (CARDINALITY ((0 1)) ((0 1)) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)))
 (ALTURA-PORTAFRR
 (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
 (POSSIBLE-VALUES (REAL) (REAL) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)
 (CARDINALITY ((0 1)) ((0 1)) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)))
 (COMPR-TOTAL
 (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
 (POSSIBLE-VALUES (REAL) (REAL) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)
 (CARDINALITY ((0 1)) ((0 1)) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)))
 (FORMA-PASTILHA
 (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
 (POSSIBLE-VALUES (ELEMENT-OF FORMA-PASTILHA)
 (ELEMENT-OF FORMA-PASTILHA) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)
 (CARDINALITY ((0 1)) ((0 1)) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)))
 (DIRECAO-CORTE
 (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
 (POSSIBLE-VALUES (ELEMENT-OF DIRECAO-CORTE)
 (ELEMENT-OF DIRECAO-CORTE) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)
 (CARDINALITY ((0 1)) ((0 1)) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)))
 (FORMA-PORTAFERR
 (NIL INSTANCESLOT-TERMINAL FRR-TORNEAMENTO
 (POSSIBLE-VALUES (ELEMENT-OF FORMA-PORTAFRR)
 (ELEMENT-OF FORMA-PORTAFRR) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)
 (CARDINALITY ((0 1)) ((0 1)) FRR-TORNEAMENTO
 FRR-TORNEAMENTO)))

```
(SUBCLASS-OF ((FERRAMENTAS) OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-SUBCLASSES
  ((SANGRAR-FORMA-INT FACEAR-INT DESBASTAR-INT SANGRAR-CORTE
    SANGRAR-FORMA-EXT FACEAR-EXT DESBASTAR-EXT)
    OWNSLOT-TERMINAL GLOBAL NIL)))
```

3 Informações de Enlace

3.2 Sistemas

```
>(get-userdefined-attributes 'sistema 'slots 'complete)
```

```
((SISTEMA-DE
  (NIL INSTANCESLOT-COMPLEX-UP SISTEMA
    (DIAMETRIC-REFERENCE (SISTEMA) (SISTEMA) SISTEMA SISTEMA)))
(PECAS (NIL INSTANCESLOT-COMPLEX-DOWN SISTEMA
  (DEMON (RESTR-PECA-SIS) (RESTR-PECA-SIS) SISTEMA SISTEMA)
  (DIAMETRIC-REFERENCE (LOC-PECA-DE) (LOC-PECA-DE) SISTEMA
    SISTEMA)
  (POSSIBLE-VALUES (INSTANCE-OF LOCALIZA-PECA)
    (INSTANCE-OF LOCALIZA-PECA) SISTEMA SISTEMA)))
(SISTEMAS
  (NIL INSTANCESLOT-COMPLEX-DOWN SISTEMA
    (DEMON (RESTR-SIS-PECA) (RESTR-SIS-PECA) SISTEMA SISTEMA)
    (DIAMETRIC-REFERENCE (LOC-SISTEMA-DE) (LOC-SISTEMA-DE)
      SISTEMA SISTEMA)
    (POSSIBLE-VALUES (INSTANCE-OF LOCALIZA-SISTEMA)
      (INSTANCE-OF LOCALIZA-SISTEMA) SISTEMA SISTEMA)))
(SISTEMA-ID (NIL INSTANCESLOT-TERMINAL SISTEMA NIL)))
```

```
>(get-schema 'restr-peca-sis 'complete)
```

6 - Os objetos a seguir são "demons", sendo que os métodos definidos em seus atributos serão disparados de acordo com a condição de disparo definida pelo nome do atributo, ou seja, o método definido em put-before, será disparado quando uma operação de inserção de um valor no "slot" estiver para ser realizada. Desta forma, antes que o valor seja colocado no slot o método é executado.

```
(RESTR-PECA-SIS (SUBCLASS-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (SUBSET-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (HAS-SUBCLASSES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (PUT-AFTER
    (NIL OWMETHOD PUT
      (COMMENT (ARGUMENTS SCHEMANAME ATTRIBUTENAME VALUE
        NEW-VALUE)
        (HERITAGE) PUT PUT)))
  (PUT-BEFORE
    ((LAMBDA (SCH ATTR OLD NEW)
      (IF ((GET-ATTRIBUTE SCH 'SISTEMAS 'VALUE)
        ((ERROR "Objeto ja utilizado para sistema"))) NIL))
    OWMETHOD PUT
    (COMMENT (ARGUMENTS SCHEMANAME ATTRIBUTENAME VALUE NEW-VALUE)
      (HERITAGE) PUT PUT)))
  (USED-IN ((SISTEMA) OWNSLOT-TERMINAL DEMONS NIL))
  (COMMENT ("created 20/1/1993 by krisys") OWNSLOT-TERMINAL GLOBAL))
(INSTANCE-OF ((PUT) OWNSLOT-TERMINAL GLOBAL NIL)))
```

```
>(get-schema 'restr-sis-peca)
```

```
(RESTR-SIS-PECA (SUBCLASS-OF (NIL OWNSLOT-TERMINAL GLOBAL))
  (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL))
  (SUBSET-OF (NIL OWNSLOT-TERMINAL GLOBAL))
  (HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL))
  (HAS-SUBCLASSES (NIL OWNSLOT-TERMINAL GLOBAL))
  (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL))
  (HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL))
  (PUT-AFTER (NIL OWMETHOD PUT))
  (PUT-BEFORE
    ((LAMBDA (SCH ATTR OLD NEW)
      (IF ((GET-ATTRIBUTE SCH 'PECAS 'VALUE)
        ((ERROR "Objeto ja utilizado para pecas"))) NIL))
    OWMETHOD PUT))
  (USED-IN ((SISTEMA) OWNSLOT-TERMINAL DEMONS))
  (COMMENT ("created 20/1/1993 by krisys") OWNSLOT-TERMINAL GLOBAL))
(INSTANCE-OF ((PUT) OWNSLOT-TERMINAL GLOBAL)))
```

3.2 Peças

```

(PECAS (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBCLASSES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (COMMENT ("created 1/3/1993 by krisys") OWNSLOT-TERMINAL GLOBAL
 NIL))
(ACABAMENTO
 (NIL INSTANCESLOT-PRIMITIVE-DOWN PECAS
7 - O aspecto demon indica que um método será disparado quando este slot tiver seu valor modificado. O código do
método assim como sua condição de disparo são representados no objeto D-ACAB-PECAS.
 (DEMON (D-ACAB-PECAS) (D-ACAB-PECAS) PECAS PECAS)
 (DIAMETRIC-REFERENCE (ACABAMENTO-DE) (ACABAMENTO-DE)
 PECAS PECAS)
 (POSSIBLE-VALUES (INSTANCE-OF ACABAMENTO)
 (INSTANCE-OF ACABAMENTO) PECAS PECAS)
 (CARDINALITY ((0 1) ((0 1) PECAS PECAS)))
(MATERIAL
 (NIL INSTANCESLOT-PRIMITIVE-DOWN PECAS
 (DEMON (D-MAT-PECAS) (D-MAT-PECAS) PECAS PECAS)
 (DIAMETRIC-REFERENCE (MATERIAL-DE) (MATERIAL-DE) PECAS
 PECAS)
 (POSSIBLE-VALUES (INSTANCE-OF MATERIAL)
 (INSTANCE-OF MATERIAL) PECAS PECAS)
 (CARDINALITY ((0 1) ((0 1) PECAS PECAS)))
9 - Os slots <nome> de tem um significado próprio nesta implementação. nestes estão registrados as ligações para agregação. O objeto que
está agregando este objeto é representado no valor do slot.
(REFERENCIADO-DE
 (NIL INSTANCESLOT-COMPLEX-UP PECAS
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIADO)
 (OBJ-REFERENCIADO) PECAS PECAS)))
(REFERENCIA-DE
 (NIL INSTANCESLOT-COMPLEX-UP PECAS
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIA) (OBJ-REFERENCIA)
 PECAS PECAS)))
(TOL-RESTRICOES
 (NIL INSTANCESLOT-COMPLEX-DOWN PECAS
 (DIAMETRIC-REFERENCE (RESTRICAO-DE) (RESTRICAO-DE)
 PECAS PECAS)))
(PECA-DE (NIL INSTANCESLOT-COMPLEX-UP PECAS
 (DIAMETRIC-REFERENCE (PECA) (PECA) PECAS PECAS)))
(GRUPOS (NIL INSTANCESLOT-COMPLEX-DOWN PECAS
 (DIAMETRIC-REFERENCE (GRUPO-DE) (GRUPO-DE) PECAS
 PECAS)
 (POSSIBLE-VALUES (INSTANCE-OF GRUPOS)
 (INSTANCE-OF GRUPOS) PECAS PECAS)))
(SITUACAO-ATUAL (NIL INSTANCESLOT-TERMINAL PECAS NIL))
(RESPONSAVEL (NIL INSTANCESLOT-TERMINAL PECAS NIL))
(DATAS (NIL INSTANCESLOT-PRIMITIVE-DOWN PECAS
 (DIAMETRIC-REFERENCE (DATA-DE) (DATA-DE) PECAS
 PECAS)
 (POSSIBLE-VALUES (INSTANCE-OF DATA)
 (INSTANCE-OF DATA) PECAS PECAS)))
(GTCODE (NIL INSTANCESLOT-TERMINAL PECAS
 (CARDINALITY ((0 1) ((0 1) PECAS PECAS)))
(CODIGO (NIL INSTANCESLOT-TERMINAL PECAS NIL))
(DESCRICAO (NIL INSTANCESLOT-TERMINAL PECAS NIL))
(PECA-ID (NIL INSTANCESLOT-TERMINAL PECAS NIL))
(SUBSET-OF ((ENL-PRODUTO) OWNSLOT-TERMINAL GLOBAL NIL))
(NEW-PECA
 ((LAMBDA (SISTEMA NOME)
 (LET ((LOCALIZA
 (SEND-MSG 'OPAMBAE 'NEW-OBJ '(LOCALIZA-PECA)))
 (PECA (SEND-MSG 'OPAMBAE 'NEW-OBJ '(PECAS))))
 (CONNECT LOCALIZA 'LOC-PECA-DE SISTEMA)
 (CONNECT PECA 'PECA-DE LOCALIZA)
 (CHANGE-ATTRIBUTE-VALUE PECA 'PECA-ID (LIST NOME))))
 OWNMETHOD PECAS NIL))
(INICIALIZA
 ((LAMBDA ()
 (PRINT "inicializa-pecas")
 (FORMAT T
 "~%Entre com a lista de tipos de grupos que formam a peca:~%"
 (LET ((GRUPOS (READ)))

```

```

(DOLIST (GRUPO GRUPOS)
(LET ((NOMEGRUPO
      (SEND-MSG 'OPAMBAE 'NEW-OBJ (LIST GRUPO))))
      (CONNECT NOMEGRUPO 'GRUPO-DE SELF)))
(LET ((MATERIAL (SEND-MSG 'OPAMBAE 'NEW-OBJ 'MATERIAL)))
      (ACABAMENTO
      (SEND-MSG 'OPAMBAE 'NEW-OBJ 'ACABAMENTO))))
(CONNECT MATERIAL 'MATERIAL-DE SELF)
(CONNECT ACABAMENTO 'ACABAMENTO-DE SELF))
INSTANCMETHOD PECAS NIL))
(TOL-RESTR
((LAMBDA (REFERENCIADA REFERENCIA TIPO)
  (PRINT "tol-restr-pecas")
  (LET ((TOL-RESTR
        (SEND-MSG 'OPAMBAE 'NEW-OBJ (LIST TIPO))))
        (CONNECT TOL-RESTR 'RESTRICAO-DE SELF)
        (CONNECT REFERENCIA 'REFERENCIA-DE TOL-RESTR)
        (CONNECT REFERENCIADA 'REFERENCIADO-DE TOL-RESTR)))
        INSTANCMETHOD PECAS NIL))
(HAS-INSTANCES
  ((PECAS142 PECAS98 PECAS87 PECAS65 PECAS28 PECAS13 PECAS5)
   OWNSLOT-TERMINAL GLOBAL NIL))
(SUBCLASS-OF ((CLASSES) OWNSLOT-TERMINAL GLOBAL NIL)))

```

3.3 Sistema de referências

Sistema de referência para os objetos

```

(LOCALIZACAO (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(COMMENT ("created 1/3/1993 by krisys") OWNSLOT-TERMINAL GLOBAL
  NIL))
(LOCAL (NIL INSTANCESLOT-PRIMITIVE-DOWN LOCALIZACAO
  (DIAMETRIC-REFERENCE (MATRIZ-DE) (MATRIZ-DE)
    LOCALIZACAO LOCALIZACAO)
  (POSSIBLE-VALUES (INSTANCE-OF MTRANSF)
    (INSTANCE-OF MTRANSF) LOCALIZACAO LOCALIZACAO)
  (CARDINALITY ((0 1)) ((0 1)) LOCALIZACAO LOCALIZACAO)))
(SUBSET-OF ((REFERENCIA) OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-SUBCLASSES
  ((LOCALIZA-PECA LOCALIZA-SISTEMA) OWNSLOT-TERMINAL GLOBAL NIL))
(SUBCLASS-OF ((CLASSES) OWNSLOT-TERMINAL GLOBAL NIL)))

```

Cálculos relativos a matriz de transformações

```

(MTRANSF (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(SUBCLASS-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-SUBCLASSES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(COMMENT ("created 1/3/1993 by krisys") OWNSLOT-TERMINAL
  GLOBAL NIL))
(MATRIZ-DE
  (NIL INSTANCESLOT-PRIMITIVE-UP MTRANSF
  (DIAMETRIC-REFERENCE (LOCAL) (LOCAL) MTRANSF MTRANSF)))
(VALORES (NIL INSTANCESLOT-TERMINAL MTRANSF
  (CARDINALITY ((0 6)) ((0 6)) MTRANSF MTRANSF)
  (POSSIBLE-VALUES (REAL) (REAL) MTRANSF MTRANSF)))
(MATRIZ (NIL INSTANCESLOT-TERMINAL MTRANSF
  (CARDINALITY ((0 1)) ((0 1)) MTRANSF MTRANSF)))
(SUBSET-OF ((REFERENCIA) OWNSLOT-TERMINAL GLOBAL NIL))
(INICIALIZA
  ((LAMBDA ()
    (PRINT "inicializa-mtransf")
    (FORMAT T
      "Entre com a lista dos valores de x,y,z e ax,sy,az~%")
    (LET ((LISTA (READ)))
      (CHANGE-ATTRIBUTE-VALUE SELF 'VALORES LISTA))
      (SEND-MSG SELF 'CALCULA-MATRIZ))
      INSTANCMETHOD MTRANSF NIL))
  (CALCULA-MATRIZ
  ((LAMBDA ()
    (LET ((LISTA (GET-ATTRIBUTE SELF 'VALORES 'VALUE)))

```

```

(LET ((M (MAKE-ARRAY 16) (X (FIRST LISTA))
      (Y (SECOND LISTA)) (Z (THIRD LISTA))
      (AX (/ (* (FOURTH LISTA) PI) 180))
      (AY (/ (* (FIFTH LISTA) PI) 180))
      (AZ (/ (* (SIXTH LISTA) PI) 180)))
      (MSET M 1 1 (* (COS AY) (COS AZ)))
      (MSET M 1 2 (* (COS AY) (SIN AZ)))
      (MSET M 1 3 (* (SIN AY) -1.0))
      (MSET M 1 4 X)
      (MSET M 2 1
        (- (* (SIN AX) (SIN AY)) (COS AZ))
          (* (COS AX) (SIN AZ))))
      (MSET M 2 2
        (+ (* (SIN AX) (SIN AY)) (SIN AZ))
          (* (COS AX) (COS AZ))))
      (MSET M 2 3 (* (SIN AX) (COS AY)))
      (MSET M 2 4 Y)
      (MSET M 3 1
        (+ (* (SIN AX) (SIN AY)) (COS AZ))
          (* (SIN AX) (SIN AZ))))
      (MSET M 3 2
        (- (* (COS AX) (SIN AY)) (SIN AZ))
          (* (SIN AX) (COS AZ))))
      (MSET M 3 3 (* (COS AX) (COS AY)))
      (MSET M 3 4 Z)
      (MSET M 4 1 0.0)
      (MSET M 4 2 0.0)
      (MSET M 4 3 0.0)
      (MSET M 4 4 1.0)
      (MSHOW M)
      (CHANGE-ATTRIBUTE-VALUE SELF 'MATRIZ (LIST M))))
(INSTANCEMETHOD MTRANSF NIL))
(INICIALIZA-VAZIO
 ((LAMBDA ()
  (PRINT "inicializa-vazio-mtransf")
  (CHANGE-ATTRIBUTE-VALUE SELF 'VALORES
    '(0.0 0.0 0.0 0.0 0.0 0.0))
  (SEND-MSG SELF 'CALCULA-MATRIZ))
  INSTANCEMETHOD MTRANSF NIL))
(LOCALIZA-PONTO
 ((LAMBDA (PX PY PZ)
  (PRINT "localiza-ponto-mtransf")
  (LET ((M (CAR (GET-ATTRIBUTE SELF 'MATRIZ 'VALUE)))
        (RESPOSTA (MAKE-ARRAY 3)))
    (DOTIMES (1 3)
      (LET ((IND (+ 1 1)))
        (SETF (AREF RESPOSTA I)
              (+ (* (MREF M IND 1) PX)
                 (* (MREF M IND 2) PY)
                 (* (MREF M IND 3) PZ)
                 (MREF M IND 4))))))
    RESPOSTA))
  INSTANCEMETHOD MTRANSF NIL))
(GETX ((LAMBDA ()
  (PRINT "getX-mtransf")
  (LIST (FIRST (GET-ATTRIBUTE SELF 'VALORES 'VALUE))))
  INSTANCEMETHOD MTRANSF NIL))
(GETY ((LAMBDA ()
  (PRINT "getY-mtransf")
  (LIST (SECOND (GET-ATTRIBUTE SELF 'VALORES 'VALUE))))
  INSTANCEMETHOD MTRANSF NIL))
(GETZ ((LAMBDA ()
  (PRINT "getZ-mtransf")
  (LIST (THIRD (GET-ATTRIBUTE SELF 'VALORES 'VALUE))))
  INSTANCEMETHOD MTRANSF NIL))
(GETAX ((LAMBDA ()
  (PRINT "getAX-mtransf")
  (LIST (FOURTH (GET-ATTRIBUTE SELF 'VALORES 'VALUE))))
  INSTANCEMETHOD MTRANSF NIL))
(GETAY ((LAMBDA ()
  (PRINT "getAY-mtransf")
  (LIST (FIFTH (GET-ATTRIBUTE SELF 'VALORES 'VALUE))))
  INSTANCEMETHOD MTRANSF NIL))
(GETAZ ((LAMBDA ()
  (PRINT "getAZ-mtransf")
  (LIST (SIXTH (GET-ATTRIBUTE SELF 'VALORES 'VALUE))))
  INSTANCEMETHOD MTRANSF NIL))

```

```

INSTANCEMETHOD MTRANSF NIL))
(SETX ((LAMBDA (VALOR)
  (PRINT "setX-mtransf")
  (CHANGE-ATTRIBUTE-VALUE SELF 'VALORES
    (RPLACE (GET-ATTRIBUTE SELF 'VALORES 'VALUE) 1
      VALOR))
  (SEND-MSG SELF 'CALCULA-MATRIZ))
INSTANCEMETHOD MTRANSF NIL))
(SETY ((LAMBDA (VALOR)
  (PRINT "setY-mtransf")
  (CHANGE-ATTRIBUTE-VALUE SELF 'VALORES
    (RPLACE (GET-ATTRIBUTE SELF 'VALORES 'VALUE) 2
      VALOR))
  (SEND-MSG SELF 'CALCULA-MATRIZ))
INSTANCEMETHOD MTRANSF NIL))
(SETZ ((LAMBDA (VALOR)
  (PRINT "setZ-mtransf")
  (CHANGE-ATTRIBUTE-VALUE SELF 'VALORES
    (RPLACE (GET-ATTRIBUTE SELF 'VALORES 'VALUE) 3
      VALOR))
  (SEND-MSG SELF 'CALCULA-MATRIZ))
INSTANCEMETHOD MTRANSF NIL))
(SETAX ((LAMBDA (VALOR)
  (PRINT "setAX-mtransf")
  (CHANGE-ATTRIBUTE-VALUE SELF 'VALORES
    (RPLACE (GET-ATTRIBUTE SELF 'VALORES 'VALUE) 4
      VALOR))
  (SEND-MSG SELF 'CALCULA-MATRIZ))
INSTANCEMETHOD MTRANSF NIL))
(SETAY ((LAMBDA (VALOR)
  (PRINT "setAY-mtransf")
  (CHANGE-ATTRIBUTE-VALUE SELF 'VALORES
    (RPLACE (GET-ATTRIBUTE SELF 'VALORES 'VALUE) 5
      VALOR))
  (SEND-MSG SELF 'CALCULA-MATRIZ))
INSTANCEMETHOD MTRANSF NIL))
(SETAZ ((LAMBDA (VALOR)
  (PRINT "setAZ-mtransf")
  (CHANGE-ATTRIBUTE-VALUE SELF 'VALORES
    (RPLACE (GET-ATTRIBUTE SELF 'VALORES 'VALUE) 6
      VALOR))
  (SEND-MSG SELF 'CALCULA-MATRIZ))
INSTANCEMETHOD MTRANSF NIL))
(HAS-INSTANCES
  ((MTRANSF223 MTRANSF214 MTRANSF211 MTRANSF209 MTRANSF203
    MTRANSF202 MTRANSF200 MTRANSF197 MTRANSF191
    MTRANSF190 MTRANSF187 MTRANSF178 MTRANSF175
    MTRANSF173 MTRANSF170 MTRANSF164 MTRANSF162
    MTRANSF160 MTRANSF151 MTRANSF146 MTRANSF144
    MTRANSF141 MTRANSF139 MTRANSF136 MTRANSF130
    MTRANSF129 MTRANSF126 MTRANSF115 MTRANSF112
    MTRANSF110 MTRANSF108 MTRANSF102 MTRANSF100 MTRANSF97
    MTRANSF95 MTRANSF91 MTRANSF89 MTRANSF86 MTRANSF84
    MTRANSF82 MTRANSF79 MTRANSF77 MTRANSF75 MTRANSF69
    MTRANSF67 MTRANSF64 MTRANSF59 MTRANSF57 MTRANSF55
    MTRANSF49 MTRANSF47 MTRANSF45 MTRANSF43 MTRANSF40
    MTRANSF38 MTRANSF32 MTRANSF30 MTRANSF27 MTRANSF25
    MTRANSF23 MTRANSF17 MTRANSF15 MTRANSF12 MTRANSF10
    MTRANSF9 MTRANSF7 MTRANSF4 MTRANSF2))

```

4 Informações Funcionais-tecnológicas

4.1 Normas

```

(NM-ASSENTO-ROL-61807 (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (SUBCLASS-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (HAS-SUBCLASSES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
  (HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))

```


(COMMENT ("created 1/3/1993 by krisys") OWNSLOT-TERMINAL GLOBAL NIL))

(SUBSET-OF ((NORMALIZADOS) OWNSLOT-TERMINAL GLOBAL NIL)))

4.2 Grupos

(GRUPOS (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (COMMENT ("created 1/3/1993 by krisys") OWNSLOT-TERMINAL GLOBAL NIL))
 (GRUPO-DE
 (NIL INSTANCESLOT-COMPLEX-UP GRUPOS
 (DIAMETRIC-REFERENCE (GRUPOS) (GRUPOS) GRUPOS GRUPOS)))
 (LOCAL (NIL INSTANCESLOT-PRIMITIVE-DOWN GRUPOS
 (DIAMETRIC-REFERENCE (MATRIZ-DE) (MATRIZ-DE) GRUPOS GRUPOS)
 (POSSIBLE-VALUES (INSTANCE-OF MTRANSF)
 (INSTANCE-OF MTRANSF) GRUPOS GRUPOS)
 (CARDINALITY ((0 1)) ((0 1)) GRUPOS GRUPOS)))
 (MATERIAL
 (NIL INSTANCESLOT-PRIMITIVE-DOWN GRUPOS
 (DEMON (D-MAT-GRUPOS) (D-MAT-GRUPOS) GRUPOS GRUPOS)
 (DIAMETRIC-REFERENCE (MATERIAL-DE) (MATERIAL-DE) GRUPOS GRUPOS)
 (POSSIBLE-VALUES (INSTANCE-OF MATERIAL)
 (INSTANCE-OF MATERIAL) GRUPOS GRUPOS)
 (CARDINALITY ((0 1)) ((0 1)) GRUPOS GRUPOS)))
 (ACABAMENTO
 (NIL INSTANCESLOT-PRIMITIVE-DOWN GRUPOS
 (DEMON (D-ACAB-GRUPOS) (D-ACAB-GRUPOS) GRUPOS GRUPOS)
 (DIAMETRIC-REFERENCE (ACABAMENTO-DE) (ACABAMENTO-DE) GRUPOS GRUPOS)
 (POSSIBLE-VALUES (INSTANCE-OF ACABAMENTO)
 (INSTANCE-OF ACABAMENTO) GRUPOS GRUPOS)
 (CARDINALITY ((0 1)) ((0 1)) GRUPOS GRUPOS)))
 (SUBSET-OF ((FUNC-TECNOL) OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBCLASSES
 ((GDE-JUNCAO GDE-CONSTRUCAO) OWNSLOT-TERMINAL GLOBAL NIL))
 (SUBCLASS-OF ((CLASSES) OWNSLOT-TERMINAL GLOBAL NIL)))

4.3 "Features"

(FEATURES (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (COMMENT ("created 1/3/1993 by krisys") OWNSLOT-TERMINAL GLOBAL NIL))
 (REFERENCIADO-DE
 (NIL INSTANCESLOT-COMPLEX-UP FEATURES
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIADO) (OBJ-REFERENCIADO) FEATURES FEATURES)))
 (REFERENCIA-DE
 (NIL INSTANCESLOT-COMPLEX-UP FEATURES
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIA) (OBJ-REFERENCIA) FEATURES FEATURES)))
 (LOCAL (NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
 (DIAMETRIC-REFERENCE (MATRIZ-DE) (MATRIZ-DE) FEATURES FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF MTRANSF)
 (INSTANCE-OF MTRANSF) FEATURES FEATURES)
 (CARDINALITY ((0 1)) ((0 1)) FEATURES FEATURES)))
 (MATERIAL
 (NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
 (DIAMETRIC-REFERENCE (MATERIAL-DE) (MATERIAL-DE) FEATURES FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF MATERIAL)
 (INSTANCE-OF MATERIAL) FEATURES FEATURES)
 (CARDINALITY ((0 1)) ((0 1)) FEATURES FEATURES)))
 (ACABAMENTO
 (NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
 (DIAMETRIC-REFERENCE (ACABAMENTO-DE) (ACABAMENTO-DE) FEATURES FEATURES)))

FEATURES FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF ACABAMENTO)
 (INSTANCE-OF ACABAMENTO) FEATURES FEATURES)
 (CARDINALITY ((0 1)) ((0 1)) FEATURES FEATURES)))
 (TOL-RESTRICOES
 (NIL INSTANCESLOT-COMPLEX-DOWN FEATURES
 (DIAMETRIC-REFERENCE (RESTRICAO-DE) (RESTRICAO-DE)
 FEATURES FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF TOL-RESTRICOES)
 (INSTANCE-OF TOL-RESTRICOES) FEATURES FEATURES)))
 (SUBSET-OF ((FUNC-TECNOL) OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBCLASSES
 ((FDETALHES FPRIMITIVA) OWNSLOT-TERMINAL GLOBAL NIL))
 (SUBCLASS-OF ((CLASSES) OWNSLOT-TERMINAL GLOBAL NIL))
 (TOL-RESTR
 ((LAMBDA (REFERENCIADA REFERENCIA TIPO)
 (PRINT "tol-restr-features")
 (LET ((TOL-RESTR (SEND-MSG 'OPAMBAE 'NEW-OBJ (LIST TIPO))))
 (CONNECT TOL-RESTR 'RESTRICAO-DE SELF)
 (CONNECT REFERENCIA 'REFERENCIA-DE TOL-RESTR)
 (CONNECT REFERENCIADA 'REFERENCIADO-DE TOL-RESTR)))
 INSTANCEMETHOD FEATURES NIL)))

"Feature": ranhura externa

(RANHURA-EXT (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (SUBSET-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (DETALHES-DE
 (NIL INSTANCESLOT-COMPLEX-UP FQ-DETALHES
 (CARDINALITY ((0 1)) (HERITAGE) FQ-DETALHES FQ-DETALHES)
 (DIAMETRIC-REFERENCE (RANHURAS-EXT) (RANHURAS-EXT)
 FQ-DETALHES RANHURA-EXT)
 (POSSIBLE-VALUES (INSTANCE-OF GRUPOS) (HERITAGE)
 FQ-DETALHES FQ-DETALHES)))
 (TOL-RESTRICOES
 (NIL INSTANCESLOT-COMPLEX-DOWN FEATURES
 (DIAMETRIC-REFERENCE (RESTRICAO-DE) (HERITAGE) FEATURES
 FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF TOL-RESTRICOES) (HERITAGE)
 FEATURES FEATURES)))
 (ACABAMENTO
 (NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
 (DEMON (D-ACAB-FDETALHES) (HERITAGE) FQ-DETALHES
 FQ-DETALHES)
 (DIAMETRIC-REFERENCE (ACABAMENTO-DE) (HERITAGE) FEATURES
 FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF ACABAMENTO) (HERITAGE)
 FEATURES FEATURES)
 (CARDINALITY ((0 1)) (HERITAGE) FEATURES FEATURES)))
 (MATERIAL
 (NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
 (DEMON (D-MAT-FDETALHES) (HERITAGE) FQ-DETALHES
 FQ-DETALHES)
 (DIAMETRIC-REFERENCE (MATERIAL-DE) (HERITAGE) FEATURES
 FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF MATERIAL) (HERITAGE)
 FEATURES FEATURES)
 (CARDINALITY ((0 1)) (HERITAGE) FEATURES FEATURES)))
 (LOCAL (NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
 (DIAMETRIC-REFERENCE (MATRIZ-DE) (HERITAGE) FEATURES
 FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF MTRANSF) (HERITAGE)
 FEATURES FEATURES)
 (CARDINALITY ((0 1)) (HERITAGE) FEATURES FEATURES)))
 (REFERENCIA-DE
 (NIL INSTANCESLOT-COMPLEX-UP FEATURES
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIA) (HERITAGE) FEATURES
 FEATURES)))
 (REFERENCIADO-DE
 (NIL INSTANCESLOT-COMPLEX-UP FEATURES
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIADO) (HERITAGE)
 FEATURES FEATURES)))
 (DIAMETRO

```

(NIL INSTANCESLOT-COMPLEX-DOWN FD-EXTERNO
 (DEMON (D-DIAMETR-FD-EXTERNO) (HERITAGE)FD-EXTERNO
  FD-EXTERNO)
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (HERITAGE) FD-EXTERNO
  FD-EXTERNO)
 (POSSIBLE-VALUES (INSTANCE-OF DDIAMETRO) (HERITAGE)
  FD-EXTERNO FD-EXTERNO)
 (CARDINALITY (( 0 1 )) (HERITAGE) FD-EXTERNO FD-EXTERNO)))
(COMMENT (" created 20/8/1993 by krisys") OWNSLOT-TERMINAL GLOBAL
 NIL))
(HAS-SUBCLASSES
 ((RAN-ANEL-VEDACAO RAN-ANEL-RETENCAO) OWNSLOT-TERMINAL GLOBAL
  NIL))
(ALTURA (NIL INSTANCESLOT-COMPLEX-DOWN RANHURA-EXT
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (DIMENSAO-DE)
  RANHURA-EXT RANHURA-EXT)
 (POSSIBLE-VALUES (INSTANCE-OF DALTURA)
  (INSTANCE-OF DALTURA) RANHURA-EXT RANHURA-EXT)
 (CARDINALITY (( 0 1 )) ((0 1)) RANHURA-EXT
  RANHURA-EXT)))
(COMPRIMENTO
 (NIL INSTANCESLOT-COMPLEX-DOWN RANHURA-EXT
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (DIMENSAO-DE)
  RANHURA-EXT RANHURA-EXT)
 (POSSIBLE-VALUES (INSTANCE-OF DCOMPRIMENTO)
  (INSTANCE-OF DCOMPRIMENTO) RANHURA-EXT RANHURA-EXT)
 (CARDINALITY (( 0 1 )) ((0 1)) RANHURA-EXT RANHURA-EXT)))
(DIAM-MAIOR
 (NIL INSTANCESLOT-COMPLEX-DOWN RANHURA-EXT
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (DIMENSAO-DE)
  RANHURA-EXT RANHURA-EXT)
 (POSSIBLE-VALUES (INSTANCE-OF DDIAM-MAIOR)
  (INSTANCE-OF DDIAM-MAIOR) RANHURA-EXT RANHURA-EXT)
 (CARDINALITY (( 0 1 )) ((0 1)) RANHURA-EXT RANHURA-EXT)))
(SUBCLASS-OF ((FDE-REGULAR) OWNSLOT-TERMINAL GLOBAL NIL))
(SUPERFICIES ((SDE-RANHURAS-EXT) OWNSLOT-TERMINAL RANHURA-EXT NIL))
(SDE-RANHURAS-EXT
 (NIL INSTANCESLOT-COMPLEX-DOWN RANHURA-EXT
 (DIAMETRIC-REFERENCE (SUPERFICIE-DE) (SUPERFICIE-DE)
  RANHURA-EXT RANHURA-EXT)
 (CARDINALITY (( 0 1 )) ((0 1)) RANHURA-EXT RANHURA-EXT)
 (POSSIBLE-VALUES (INSTANCE-OF SDE-RANHURA-EXT)
  (INSTANCE-OF SDE-RANHURA-EXT) RANHURA-EXT RANHURA-EXT)))
(INICIALIZA
 ((LAMBDA ()
 .....
 INSTANCEMETHOD FD-EXTERNO NIL))
(TOL-RESTR
 ((LAMBDA (REFERENCIADA REFERENCIA TIPO)
 (PRINT "tol-restr-features")
 (LET ((TOL-RESTR (SEND-MSG 'OPAMBAE 'NEW-OBJ (LIST TIPO))))
 (CONNECT TOL-RESTR 'RESTRICAO-DE SELF)
 (CONNECT REFERENCIA 'REFERENCIA-DE TOL-RESTR)
 (CONNECT REFERENCIADA 'REFERENCIADO-DE TOL-RESTR)))
 INSTANCEMETHOD FEATURES NIL)))

```

"Feature": Ranhura para anel de retenção

```

(RAN-ANEL-RETENCAO (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (SUBSET-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBCLASSES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (TOL-RESTR
 ((LAMBDA (REFERENCIADA REFERENCIA TIPO)
 (PRINT "tol-restr-features")
 (LET ((TOL-RESTR (SEND-MSG 'OPAMBAE 'NEW-OBJ (LIST TIPO))))
 (CONNECT TOL-RESTR 'RESTRICAO-DE SELF)
 (CONNECT REFERENCIA 'REFERENCIA-DE TOL-RESTR)
 (CONNECT REFERENCIADA 'REFERENCIADO-DE TOL-RESTR)))
 INSTANCEMETHOD FEATURES NIL))
(INICIALIZA
 ((LAMBDA ()
 .....
 INSTANCEMETHOD FD-EXTERNO NIL))
(SDE-RANHURAS-EXT

```

(NIL INSTANCESLOT-COMPLEX-DOWN RANHURA-EXT
 (DIAMETRIC-REFERENCE (SUPERFICIE-DE) (HERITAGE)
 RANHURA-EXT RANHURA-EXT)
 (CARDINALITY ((0 1)) (HERITAGE) RANHURA-EXT RANHURA-EXT)
 (POSSIBLE-VALUES (INSTANCE-OF SDE-RANHURA-EXT) (HERITAGE)
 RANHURA-EXT RANHURA-EXT)))
 (DIAM-MAIOR
 (NIL INSTANCESLOT-COMPLEX-DOWN RANHURA-EXT
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (HERITAGE) RANHURA-EXT
 RANHURA-EXT)
 (POSSIBLE-VALUES (INSTANCE-OF DDIAM-MAIOR) (HERITAGE)
 RANHURA-EXT RANHURA-EXT)
 (CARDINALITY ((0 1)) (HERITAGE) RANHURA-EXT RANHURA-EXT)))
 (COMPRIMENTO
 (NIL INSTANCESLOT-COMPLEX-DOWN RANHURA-EXT
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (HERITAGE) RANHURA-EXT
 RANHURA-EXT)
 (POSSIBLE-VALUES (INSTANCE-OF DCOMPRIMENTO) (HERITAGE)
 RANHURA-EXT RANHURA-EXT)
 (CARDINALITY ((0 1)) (HERITAGE) RANHURA-EXT RANHURA-EXT)))
 (ALTURA (NIL INSTANCESLOT-COMPLEX-DOWN RANHURA-EXT
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (HERITAGE)
 RANHURA-EXT RANHURA-EXT)
 (POSSIBLE-VALUES (INSTANCE-OF DALTURA) (HERITAGE)
 RANHURA-EXT RANHURA-EXT)
 (CARDINALITY ((0 1)) (HERITAGE) RANHURA-EXT
 RANHURA-EXT)))
 (DIAMETRO
 (NIL INSTANCESLOT-COMPLEX-DOWN FD-EXTERNO
 (DEMON (D-DIAMETR-FD-EXTERNO) (HERITAGE) FD-EXTERNO
 FD-EXTERNO)
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (HERITAGE) FD-EXTERNO
 FD-EXTERNO)
 (POSSIBLE-VALUES (INSTANCE-OF DDIAMETRO) (HERITAGE)
 FD-EXTERNO FD-EXTERNO)
 (CARDINALITY ((0 1)) (HERITAGE) FD-EXTERNO FD-EXTERNO)))
 (REFERENCIADO-DE
 (NIL INSTANCESLOT-COMPLEX-UP FEATURES
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIADO) (HERITAGE)
 FEATURES FEATURES)))
 (REFERENCIA-DE
 (NIL INSTANCESLOT-COMPLEX-UP FEATURES
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIA) (HERITAGE) FEATURES
 FEATURES)))
 (LOCAL (NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
 (DIAMETRIC-REFERENCE (MATRIZ-DE) (HERITAGE) FEATURES
 FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF MTRANSF) (HERITAGE)
 FEATURES FEATURES)
 (CARDINALITY ((0 1)) (HERITAGE) FEATURES FEATURES)))
 (MATERIAL
 (NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
 (DEMON (D-MAT-FDETALHES) (HERITAGE) FQ-DETALHES
 FQ-DETALHES)
 (DIAMETRIC-REFERENCE (MATERIAL-DE) (HERITAGE) FEATURES
 FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF MATERIAL) (HERITAGE)
 FEATURES FEATURES)
 (CARDINALITY ((0 1)) (HERITAGE) FEATURES FEATURES)))
 (ACABAMENTO
 (NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
 (DEMON (D-ACAB-FDETALHES) (HERITAGE) FQ-DETALHES
 FQ-DETALHES)
 (DIAMETRIC-REFERENCE (ACABAMENTO-DE) (HERITAGE) FEATURES
 FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF ACABAMENTO) (HERITAGE)
 FEATURES FEATURES)
 (CARDINALITY ((0 1)) (HERITAGE) FEATURES FEATURES)))
 (TOL-RESTRICOES
 (NIL INSTANCESLOT-COMPLEX-DOWN FEATURES
 (DIAMETRIC-REFERENCE (RESTRICAO-DE) (HERITAGE) FEATURES
 FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF TOL-RESTRICOES) (HERITAGE)
 FEATURES FEATURES)))
 (DETALHES-DE
 (NIL INSTANCESLOT-COMPLEX-UP FQ-DETALHES

(DIAMETRIC-REFERENCE (RANS-ANEL-RETENCAO)
 (RANS-ANEL-RETENCAO) FQ-DETALHES RAN-ANEL-RETENCAO)
 (CARDINALITY (| 0 1 |) (HERITAGE) FQ-DETALHES FQ-DETALHES)
 (POSSIBLE-VALUES (INSTANCE-OF GRUPOS) (HERITAGE)
 FQ-DETALHES FQ-DETALHES)))
 (COMMENT ("created 20/8/1993 by krisys") OWNSLOT-TERMINAL GLOBAL
 NIL))
 (SUBCLASS-OF ((RANHURA-EXT) OWNSLOT-TERMINAL GLOBAL NIL))
 (SUPERFICIES
 ((SDE-RANHURAS-EXT) OWNSLOT-TERMINAL RAN-ANEL-RETENCAO NIL))
 (HAS-INSTANCES
 ((RAN-ANEL-RETENCAO100) OWNSLOT-TERMINAL GLOBAL NIL)))

"Feature": Ranhura para anel de vedação

(RAN-ANEL-VEDACAO (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (SUBSET-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBCLASSES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (TOL-RESTR
 ((LAMBDA (REFERENCIADA REFERENCIA TIPO)
 (PRINT "tol-restr-features")
 (LET ((TOL-RESTR (SEND-MSG 'OPAMBAE 'NEW-OBJ (LIST TIPO))))
 (CONNECT TOL-RESTR 'RESTRICAO-DE SELF)
 (CONNECT REFERENCIA 'REFERENCIA-DE TOL-RESTR)
 (CONNECT REFERENCIADA 'REFERENCIADO-DE TOL-RESTR)))
 INSTANCEMETHOD FEATURES NIL))
 (INICIALIZA
 ((LAMBDA ()

 INSTANCEMETHOD FD-EXTERNO NIL))
 (SDE-RANHURAS-EXT
 (NIL INSTANCESLOT-COMPLEX-DOWN RANHURA-EXT
 (DIAMETRIC-REFERENCE (SUPERFICIE-DE) (HERITAGE)
 RANHURA-EXT RANHURA-EXT)
 (CARDINALITY (| 0 1 |) (HERITAGE) RANHURA-EXT RANHURA-EXT)
 (POSSIBLE-VALUES (INSTANCE-OF SDE-RANHURA-EXT) (HERITAGE)
 RANHURA-EXT RANHURA-EXT)))
 (DIAM-MAIOR
 (NIL INSTANCESLOT-COMPLEX-DOWN RANHURA-EXT
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (HERITAGE) RANHURA-EXT
 RANHURA-EXT)
 (POSSIBLE-VALUES (INSTANCE-OF DDIAM-MAIOR) (HERITAGE)
 RANHURA-EXT RANHURA-EXT)
 (CARDINALITY (| 0 1 |) (HERITAGE) RANHURA-EXT RANHURA-EXT)))
 (COMPRIMENTO
 (NIL INSTANCESLOT-COMPLEX-DOWN RANHURA-EXT
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (HERITAGE) RANHURA-EXT
 RANHURA-EXT)
 (POSSIBLE-VALUES (INSTANCE-OF DCOMPRIMENTO) (HERITAGE)
 RANHURA-EXT RANHURA-EXT)
 (CARDINALITY (| 0 1 |) (HERITAGE) RANHURA-EXT RANHURA-EXT)))
 (ALTURA (NIL INSTANCESLOT-COMPLEX-DOWN RANHURA-EXT
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (HERITAGE)
 RANHURA-EXT RANHURA-EXT)
 (POSSIBLE-VALUES (INSTANCE-OF DAL TURA) (HERITAGE)
 RANHURA-EXT RANHURA-EXT)
 (CARDINALITY (| 0 1 |) (HERITAGE) RANHURA-EXT
 RANHURA-EXT)))
 (DIAMETRO
 (NIL INSTANCESLOT-COMPLEX-DOWN FD-EXTERNO
 (DEMON (D-DIAMETR-FD-EXTERNO) (HERITAGE) FD-EXTERNO
 FD-EXTERNO)
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (HERITAGE) FD-EXTERNO
 FD-EXTERNO)
 (POSSIBLE-VALUES (INSTANCE-OF DDIAMETRO) (HERITAGE)
 FD-EXTERNO FD-EXTERNO)
 (CARDINALITY (| 0 1 |) (HERITAGE) FD-EXTERNO FD-EXTERNO)))
 (REFERENCIADO-DE
 (NIL INSTANCESLOT-COMPLEX-UP FEATURES
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIADO) (HERITAGE)
 FEATURES FEATURES)))
 (REFERENCIA-DE
 (NIL INSTANCESLOT-COMPLEX-UP FEATURES

```

(DIAMETRIC-REFERENCE (OBJ-REFERENCIA) (HERITAGE) FEATURES
FEATURES)))
(LOCAL (NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
(DIAMETRIC-REFERENCE (MATRIZ-DE) (HERITAGE) FEATURES
FEATURES)
(POSSIBLE-VALUES (INSTANCE-OF MTRANSF) (HERITAGE)
FEATURES FEATURES)
(CARDINALITY (( 0 1 )) (HERITAGE) FEATURES FEATURES)))
(MATERIAL
(NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
(DEMON (D-MAT-FDETALHES) (HERITAGE) FQ-DETALHES
FQ-DETALHES)
(DIAMETRIC-REFERENCE (MATERIAL-DE) (HERITAGE) FEATURES
FEATURES)
(POSSIBLE-VALUES (INSTANCE-OF MATERIAL) (HERITAGE)
FEATURES FEATURES)
(CARDINALITY (( 0 1 )) (HERITAGE) FEATURES FEATURES)))
(ACABAMENTO
(NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
(DEMON (D-ACAB-FDETALHES) (HERITAGE) FQ-DETALHES
FQ-DETALHES)
(DIAMETRIC-REFERENCE (ACABAMENTO-DE) (HERITAGE) FEATURES
FEATURES)
(POSSIBLE-VALUES (INSTANCE-OF ACABAMENTO) (HERITAGE)
FEATURES FEATURES)
(CARDINALITY (( 0 1 )) (HERITAGE) FEATURES FEATURES)))
(TOL-RESTRICOES
(NIL INSTANCESLOT-COMPLEX-DOWN FEATURES
(DIAMETRIC-REFERENCE (RESTRICAO-DE) (HERITAGE) FEATURES
FEATURES)
(POSSIBLE-VALUES (INSTANCE-OF TOL-RESTRICOES) (HERITAGE)
FEATURES FEATURES)))
(DETALHES-DE
(NIL INSTANCESLOT-COMPLEX-UP FQ-DETALHES
(CARDINALITY (( 0 1 )) (HERITAGE) FQ-DETALHES FQ-DETALHES)
(DIAMETRIC-REFERENCE (RANHURAS-EXT) (HERITAGE) FQ-DETALHES
RANHURA-EXT)
(POSSIBLE-VALUES (INSTANCE-OF GRUPOS) (HERITAGE)
FQ-DETALHES FQ-DETALHES)))
(COMMENT ("created 23/8/1993 by krisys") OWNSLOT-TERMINAL GLOBAL
NIL))
(ANGULO (NIL INSTANCESLOT-COMPLEX-DOWN RAN-ANEL-VEDACAO
(DIAMETRIC-REFERENCE (DIMENSAO-DE) (DIMENSAO-DE)
RAN-ANEL-VEDACAO RAN-ANEL-VEDACAO)
(POSSIBLE-VALUES (INSTANCE-OF DANGULO)
(INSTANCE-OF DANGULO) RAN-ANEL-VEDACAO
RAN-ANEL-VEDACAO)
(CARDINALITY (( 0 1 )) ((0 1) RAN-ANEL-VEDACAO
RAN-ANEL-VEDACAO)))
(SUBCLASS-OF ((RANHURA-EXT) OWNSLOT-TERMINAL GLOBAL NIL))
(SUPERFICIES
((SFACES SCILDRICAS) OWNSLOT-TERMINAL RAN-ANEL-VEDACAO NIL)))

```

"Feature": assento de rolamento

```

(ASSENTO-ROLAMENTO (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(SUBSET-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-SUBCLASSES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
(TOL-RESTR
((LAMBDA (REFERENCIADA REFERENCIA TIPO)
(PRINT "tol-restr-features")
(LET ((TOL-RESTR (SEND-MSG 'OPAMBAE 'NEW-OBJ (LIST TIPO))))
(CONNECT TOL-RESTR 'RESTRICAO-DE SELF)
(CONNECT REFERENCIA 'REFERENCIA-DE TOL-RESTR)
(CONNECT REFERENCIADA 'REFERENCIADO-DE TOL-RESTR)))
INSTANCEMETHOD FEATURES NIL))
(INICIALIZA
((LAMBDA ()
(PRINT "inicializa-feature-detalle-externo")
INSTANCEMETHOD FD-EXTERNO NIL))
(DETALHES-DE
(NIL INSTANCESLOT-COMPLEX-UP FDETALHES
(POSSIBLE-VALUES (INSTANCE-OF GRUPOS) (HERITAGE) FDETALHES

```

FDETALHES)
 (DIAMETRIC-REFERENCE (DUMMY) (HERITAGE) FDETALHES
 FDETALHES)
 (CARDINALITY ([0 1]) (HERITAGE) FDETALHES FDETALHES)))
 (REFERENCIADO-DE
 (NIL INSTANCESLOT-COMPLEX-UP FEATURES
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIADO) (HERITAGE)
 FEATURES FEATURES)))
 (REFERENCIA-DE
 (NIL INSTANCESLOT-COMPLEX-UP FEATURES
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIA) (HERITAGE) FEATURES
 FEATURES)))
 (LOCAL (NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
 (DIAMETRIC-REFERENCE (MATRIZ-DE) (HERITAGE) FEATURES
 FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF MTRANSF) (HERITAGE)
 FEATURES FEATURES)
 (CARDINALITY ([0 1]) (HERITAGE) FEATURES FEATURES)))
 (MATERIAL
 (NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
 (DEMON (D-MAT-FDETALHES) (HERITAGE) FDETALHES FDETALHES)
 (DIAMETRIC-REFERENCE (MATERIAL-DE) (HERITAGE) FEATURES
 FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF MATERIAL) (HERITAGE)
 FEATURES FEATURES)
 (CARDINALITY ([0 1]) (HERITAGE) FEATURES FEATURES)))
 (ACABAMENTO
 (NIL INSTANCESLOT-PRIMITIVE-DOWN FEATURES
 (DEMON (D-ACAB-FDETALHES) (HERITAGE) FDETALHES FDETALHES)
 (DIAMETRIC-REFERENCE (ACABAMENTO-DE) (HERITAGE) FEATURES
 FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF ACABAMENTO) (HERITAGE)
 FEATURES FEATURES)
 (CARDINALITY ([0 1]) (HERITAGE) FEATURES FEATURES)))
 (TOL-RESTRICOES
 (NIL INSTANCESLOT-COMPLEX-DOWN FEATURES
 (DIAMETRIC-REFERENCE (RESTRICAO-DE) (HERITAGE) FEATURES
 FEATURES)
 (POSSIBLE-VALUES (INSTANCE-OF TOL-RESTRICOES) (HERITAGE)
 FEATURES FEATURES)))
 (DIAMETRO
 (NIL INSTANCESLOT-COMPLEX-DOWN FD-EXTERNO
 (DEMON (D-DIAMETR-FD-EXTERNO) (HERITAGE) FD-EXTERNO
 FD-EXTERNO)
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (HERITAGE) FD-EXTERNO
 FD-EXTERNO)
 (POSSIBLE-VALUES (INSTANCE-OF DDIAMETRO) (HERITAGE)
 FD-EXTERNO FD-EXTERNO)
 (CARDINALITY ([0 1]) (HERITAGE) FD-EXTERNO FD-EXTERNO)))
 (COMMENT ("created 14/1/1993 by krisys") OWNSLOT-TERMINAL GLOBAL
 NIL))
 (SUPERFICIES ((SCILINDRICAS SFACES) OWNSLOT-TERMINAL ASSENTO-ROLAMENTO NIL))
 (SCILINDRICAS
 (NIL INSTANCESLOT-COMPLEX-DOWN ASSENTO-ROLAMENTO
 (DIAMETRIC-REFERENCE (SUPERFICIE-DE) (SUPERFICIE-DE)
 ASSENTO-ROLAMENTO ASSENTO-ROLAMENTO)
 (CARDINALITY ([0 1]) ([0 1]) ASSENTO-ROLAMENTO
 ASSENTO-ROLAMENTO)
 (POSSIBLE-VALUES (INSTANCE-OF SCILINDRICA)
 (INSTANCE-OF SCONICA) ASSENTO-ROLAMENTO
 ASSENTO-ROLAMENTO)))
 (SFACES
 (NIL INSTANCESLOT-COMPLEX-DOWN ASSENTO-ROLAMENTO
 (DIAMETRIC-REFERENCE (SUPERFICIE-DE) (SUPERFICIE-DE)
 ASSENTO-ROLAMENTO ASSENTO-ROLAMENTO)
 (CARDINALITY ([0 1]) ([0 1]) ASSENTO-ROLAMENTO
 ASSENTO-ROLAMENTO)
 (POSSIBLE-VALUES (INSTANCE-OF SFACES)
 (INSTANCE-OF SCONICA) ASSENTO-ROLAMENTO
 ASSENTO-ROLAMENTO)))
 (SUBCLASS-OF ((FDE-REGULAR) OWNSLOT-TERMINAL GLOBAL NIL)))

4.4 Elementos Agregados

Superficies

```

(SUPERFICIES (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (COMMENT ("created 1/3/1993 by krisys") OWNSLOT-TERMINAL GLOBAL
 NIL))
(REFERENCIADO-DE
 (NIL INSTANCESLOT-COMPLEX-UP SUPERFICIES
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIADO) (OBJ-REFERENCIADO)
 SUPERFICIES SUPERFICIES)))
(REFERENCIA-DE
 (NIL INSTANCESLOT-COMPLEX-UP SUPERFICIES
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIA) (OBJ-REFERENCIA)
 SUPERFICIES SUPERFICIES)))
(SUPERFICIE-DE
 (NIL INSTANCESLOT-COMPLEX-UP SUPERFICIES
 (DIAMETRIC-REFERENCE (DUMMY) (DUMMY) SUPERFICIES
 SUPERFICIES)
 (CARDINALITY ((0 1)) ((0 1)) SUPERFICIES SUPERFICIES)))
(LOCAL (NIL INSTANCESLOT-PRIMITIVE-DOWN SUPERFICIES
 (DIAMETRIC-REFERENCE (MATRIZ-DE) (MATRIZ-DE)
 SUPERFICIES SUPERFICIES)
 (POSSIBLE-VALUES (INSTANCE-OF MTRANSF)
 (INSTANCE-OF MTRANSF) SUPERFICIES SUPERFICIES)
 (CARDINALITY ((0 1)) ((0 1)) SUPERFICIES SUPERFICIES)))
(MATERIAL
 (NIL INSTANCESLOT-PRIMITIVE-DOWN SUPERFICIES
 (DEMON (D-TRIB-SUPERFICIE) (D-TRIB-SUPERFICIE)
 SUPERFICIES SUPERFICIES)
 (DIAMETRIC-REFERENCE (MATERIAL-DE) (MATERIAL-DE)
 SUPERFICIES SUPERFICIES)
 (POSSIBLE-VALUES (INSTANCE-OF MATERIAL)
 (INSTANCE-OF MATERIAL) SUPERFICIES SUPERFICIES)
 (CARDINALITY ((0 1)) ((0 1)) SUPERFICIES SUPERFICIES)))
(ACABAMENTO
 (NIL INSTANCESLOT-PRIMITIVE-DOWN SUPERFICIES
 (DEMON (D-TRIB-SUPERFICIE) (D-TRIB-SUPERFICIE)
 SUPERFICIES SUPERFICIES)
 (DIAMETRIC-REFERENCE (ACABAMENTO-DE) (ACABAMENTO-DE)
 SUPERFICIES SUPERFICIES)
 (POSSIBLE-VALUES (INSTANCE-OF ACABAMENTO)
 (INSTANCE-OF ACABAMENTO) SUPERFICIES SUPERFICIES)
 (CARDINALITY ((0 1)) ((0 1)) SUPERFICIES SUPERFICIES)))
(TOL-FORMA
 (NIL INSTANCESLOT-PRIMITIVE-DOWN SUPERFICIES
 (DIAMETRIC-REFERENCE (TFORMA-DE) (TFORMA-DE) SUPERFICIES
 SUPERFICIES)
 (POSSIBLE-VALUES (INSTANCE-OF TOL-FORMA)
 (INSTANCE-OF TOL-FORMA) SUPERFICIES SUPERFICIES)))
(SET-TOL-FORMA
 (LAMBDA( TIPO-TOLER VALOR-TOLER)
 (CHANGE-ATTRIBUTE-VALUE 'SELF 'TIPO-FORMA TIPO-TOLER)
 (CHANGE-ATTRIBUTE-VALUE 'SELF 'VALOR-FORMA VALOR-TOLER)
 INSTANCEMETHOD SUPERFICIES NIL))
(TIPO-FORMA
 (NIL INSTANCESLOT-TERMINAL SUPERFICIES NIL))
(VALOR-FORMA
 (NIL INSTANCESLOT-TERMINAL SUPERFICIES NIL))
(SUBSET-OF ((ELEMENTOS-AGREGADOS) OWNSLOT-TERMINAL GLOBAL NIL))
(HAS-SUBCLASSES
 ((SCHAV-WOOD-EXT SCHAV-DESL-EXT SCHAV-PARAL-EXT SCHAVETA-INT
 SPLANA SCPOLIGONAL SRECARILHADA SROSCA-EXT SROSCA-INT
 SDE-RANHURA-EXT SDE-RANHURA-INT SCONCORD-CONCAV
 SCONCORD-CONVEX SFACE SCONICA SCILINDRICA)
 OWNSLOT-TERMINAL GLOBAL NIL))
(SUBCLASS-OF ((CLASSES) OWNSLOT-TERMINAL GLOBAL NIL))
(INICIALIZA-VAZIO
 ((LAMBDA ()
 (PRINT "inicializa-superficies-vazio")
 (PRINT SELF)
 (LET ((MATRIZ (SEND-MSG 'OPAMBAE 'NEW-OBJ-VAZIO '(MTRANSF))))

```


(CONNECT MATRIZ 'MATRIZ-DE SELF'))
 INSTANCEMETHOD SUPERFICIES NIL))
 (INICIALIZA
 ((LAMBDA ()

 INSTANCEMETHOD SUPERFICIES NIL)))
 (SCILINDRICA (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (SUBSET-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBCLASSES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (TOL-FORMA
 (NIL INSTANCESLOT-PRIMITIVE-DOWN SUPERFICIES
 (DIAMETRIC-REFERENCE (TFORMA-DE) (HERITAGE) SUPERFICIES
 SUPERFICIES)
 (POSSIBLE-VALUES (INSTANCE-OF TOL-FORMA) (HERITAGE)
 SUPERFICIES SUPERFICIES)))
 (ACABAMENTO
 (NIL INSTANCESLOT-PRIMITIVE-DOWN SUPERFICIES
 (DEMON (D-ATRIB-SUPERFICIE) (HERITAGE) SUPERFICIES
 SUPERFICIES)
 (DIAMETRIC-REFERENCE (ACABAMENTO-DE) (HERITAGE)
 SUPERFICIES SUPERFICIES)
 (POSSIBLE-VALUES (INSTANCE-OF ACABAMENTO) (HERITAGE)
 SUPERFICIES SUPERFICIES)
 (CARDINALITY ((0 1)) (HERITAGE) SUPERFICIES SUPERFICIES)))
 (MATERIAL
 (NIL INSTANCESLOT-PRIMITIVE-DOWN SUPERFICIES
 (DEMON (D-ATRIB-SUPERFICIE) (HERITAGE) SUPERFICIES
 SUPERFICIES)
 (DIAMETRIC-REFERENCE (MATERIAL-DE) (HERITAGE) SUPERFICIES
 SUPERFICIES)
 (POSSIBLE-VALUES (INSTANCE-OF MATERIAL) (HERITAGE)
 SUPERFICIES SUPERFICIES)
 (CARDINALITY ((0 1)) (HERITAGE) SUPERFICIES SUPERFICIES)))
 (LOCAL (NIL INSTANCESLOT-PRIMITIVE-DOWN SUPERFICIES
 (DIAMETRIC-REFERENCE (MATRIZ-DE) (HERITAGE) SUPERFICIES
 SUPERFICIES)
 (POSSIBLE-VALUES (INSTANCE-OF MTRANSF) (HERITAGE)
 SUPERFICIES SUPERFICIES)
 (CARDINALITY ((0 1)) (HERITAGE) SUPERFICIES
 SUPERFICIES)))
 (SUPERFICIE-DE
 (NIL INSTANCESLOT-COMPLEX-UP SUPERFICIES
 (DIAMETRIC-REFERENCE (SCILINDRICAS) (SCILINDRICAS)
 SUPERFICIES SCILINDRICA)
 (CARDINALITY ((0 1)) (HERITAGE) SUPERFICIES SUPERFICIES)))
 (REFERENCIA-DE
 (NIL INSTANCESLOT-COMPLEX-UP SUPERFICIES
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIA) (HERITAGE)
 SUPERFICIES SUPERFICIES)))
 (REFERENCIADO-DE
 (NIL INSTANCESLOT-COMPLEX-UP SUPERFICIES
 (DIAMETRIC-REFERENCE (OBJ-REFERENCIADO) (HERITAGE)
 SUPERFICIES SUPERFICIES)))
 (COMMENT ("created 1/3/1993 by krisys") OWNSLOT-TERMINAL GLOBAL
 NIL))
 (COMPRIMENTO
 (NIL INSTANCESLOT-COMPLEX-DOWN SCILINDRICA
 (DEMON (D-ATRIB-SUPERFICIE) (D-ATRIB-SUPERFICIE)
 SCILINDRICA SCILINDRICA)
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (DIMENSAO-DE)
 SCILINDRICA SCILINDRICA)
 (POSSIBLE-VALUES (INSTANCE-OF DCOMPRIMENTO)
 (INSTANCE-OF DCOMPRIMENTO) SCILINDRICA SCILINDRICA)
 (CARDINALITY ((0 1)) ((0 1)) SCILINDRICA SCILINDRICA)))
 (DIAMETRO
 (NIL INSTANCESLOT-COMPLEX-DOWN SCILINDRICA
 (DEMON (D-ATRIB-SUPERFICIE) (D-ATRIB-SUPERFICIE)
 SCILINDRICA SCILINDRICA)
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (DIMENSAO-DE)
 SCILINDRICA SCILINDRICA)
 (POSSIBLE-VALUES (INSTANCE-OF DDIAMETRO)
 (INSTANCE-OF DDIAMETRO) SCILINDRICA SCILINDRICA)
 (CARDINALITY ((0 1)) ((0 1)) SCILINDRICA SCILINDRICA)))

```

(SUBCLASS-OF ((SUPERFICIES) OWNSLOT-TERMINAL GLOBAL NIL))
(INICIALIZA-VAZIO
  ((LAMBDA ()
    (PRINT "inicializa-superficies-vazio")
    (PRINT SELF)
    (LET ((MATRIZ (SEND-MSG 'OPAMBAE 'NEW-OBJ-VAZIO '(MTRANSF))))
      (CONNECT MATRIZ 'MATRIZ-DE SELF)))
    INSTANCEMETHOD SUPERFICIES NIL))
(HAS-INSTANCES
  ((SCILINDRICA539 SCILINDRICA189 SCILINDRICA169 SCILINDRICA138
    SCILINDRICA128 SCILINDRICA107 SCILINDRICA74 SCILINDRICA54
    SCILINDRICA37 SCILINDRICA22)
    OWNSLOT-TERMINAL GLOBAL NIL))
(INICIALIZA
  ((LAMBDA ()
    .....
    INSTANCEMETHOD SUPERFICIES NIL)))

```

Tolerâncias de restrição

>(get-attribute 'pecas28 'tol-restricoes 'complete)

```

(TOL-RESTRICOES
  ((CONCENTRICIDADE62) OWNSLOT-COMPLEX-DOWN PECAS
    (DIAMETRIC-REFERENCE (RESTRICAO-DE) (HERITAGE) PECAS PECAS)))

>(get-userdefined-attributes 'concentricidade62 'slots 'complete)
.....
((VALOR ((0.050000000000000003) OWNSLOT-TERMINAL CONCENTRICIDADE
  (POSSIBLE-VALUES (REAL) (HERITAGE) CONCENTRICIDADE
    CONCENTRICIDADE)
  (CARDINALITY (1 0 1)) (HERITAGE) CONCENTRICIDADE
    CONCENTRICIDADE)))
(RESTRICAO-DE
  ((PECAS28) OWNSLOT-COMPLEX-UP TOL-RESTRICOES
    (DIAMETRIC-REFERENCE (TOL-RESTRICOES) (HERITAGE) TOL-RESTRICOES
      TOL-RESTRICOES)))
(OBJ-REFERENCIA
  ((EIXO-SIMPLES29) OWNSLOT-COMPLEX-DOWN TOL-RESTRICOES
    (DIAMETRIC-REFERENCE (REFERENCIA-DE) (HERITAGE) TOL-RESTRICOES
      TOL-RESTRICOES)))
(OBJ-REFERENCIADO
  ((EIXO-SIMPLES46) OWNSLOT-COMPLEX-DOWN TOL-RESTRICOES
    (DIAMETRIC-REFERENCE (REFERENCIADO-DE) (HERITAGE) TOL-RESTRICOES
      TOL-RESTRICOES)))

>(get-userdefined-attributes 'eixo-simples29 'slots 'complete)
.....
(GRUPO-DE
  ((PECAS28) OWNSLOT-COMPLEX-UP GRUPOS
    (DIAMETRIC-REFERENCE (GRUPOS) (HERITAGE) GRUPOS GRUPOS)))
(REFERENCIADO-DE
  (NIL OWNSLOT-COMPLEX-UP GDE-CONSTRUCAO
    (DIAMETRIC-REFERENCE (OBJ-REFERENCIADO) (HERITAGE)
      GDE-CONSTRUCAO GDE-CONSTRUCAO)))
(REFERENCIA-DE
  ((CONCENTRICIDADE62) OWNSLOT-COMPLEX-UP GDE-CONSTRUCAO
    (DIAMETRIC-REFERENCE (OBJ-REFERENCIA) (HERITAGE) GDE-CONSTRUCAO
      GDE-CONSTRUCAO)))

>(get-userdefined-attributes 'eixo-simples46 'slots 'complete)
.....
(GRUPO-DE
  ((PECAS28) OWNSLOT-COMPLEX-UP GRUPOS
    (DIAMETRIC-REFERENCE (GRUPOS) (HERITAGE) GRUPOS GRUPOS)))
(REFERENCIADO-DE
  ((CONCENTRICIDADE62) OWNSLOT-COMPLEX-UP GDE-CONSTRUCAO
    (DIAMETRIC-REFERENCE (OBJ-REFERENCIADO) (HERITAGE) GDE-CONSTRUCAO
      GDE-CONSTRUCAO)))
(REFERENCIA-DE
  (NIL OWNSLOT-COMPLEX-UP GDE-CONSTRUCAO
    (DIAMETRIC-REFERENCE (OBJ-REFERENCIA) (HERITAGE)
      GDE-CONSTRUCAO GDE-CONSTRUCAO)))

```

Tolerância dimensional

>(get-attribute 'cilindro163 'diametro 'complete)

```

(DIAMETRO
  ((DIAMETRO167) OWNSLOT-COMPLEX-DOWN CILINDRO
   (DIAMETRIC-REFERENCE (DIMENSAO-DE) (HERITAGE) CILINDRO CILINDRO)
   (POSSIBLE-VALUES (INSTANCE-OF DDIAMETRO) (HERITAGE) CILINDRO
    CILINDRO)
   (CARDINALITY (( 0 1 )) (HERITAGE) CILINDRO CILINDRO)))

>(get-userdefined-attributes 'TOL-DIMENSIONAL168 'slots 'complete)

((VALOR-SUPERIOR
  ((0.1000000000000002) OWNSLOT-TERMINAL TOL-DIMENSIONAL
   (POSSIBLE-VALUES (REAL) (HERITAGE) TOL-DIMENSIONAL
    TOL-DIMENSIONAL)
   (CARDINALITY (( 0 1 )) (HERITAGE) TOL-DIMENSIONAL
    TOL-DIMENSIONAL)))
 (VALOR-INFERIOR
  ((0.1000000000000002) OWNSLOT-TERMINAL TOL-DIMENSIONAL
   (POSSIBLE-VALUES (REAL) (HERITAGE) TOL-DIMENSIONAL
    TOL-DIMENSIONAL)
   (CARDINALITY (( 0 1 )) (HERITAGE) TOL-DIMENSIONAL
    TOL-DIMENSIONAL)))
 (TDIM-DE ((DIAMETRO167) OWNSLOT-PRIMITIVE-UP TOL-DIMENSIONAL
  (DIAMETRIC-REFERENCE (TDIM) (HERITAGE) TOL-DIMENSIONAL
   TOL-DIMENSIONAL)))

>(get-userdefined-attributes 'ddiametro167 'slots 'complete)

((DIMENSAO-DE
  ((CILINDRO163) OWNSLOT-COMPLEX-UP DDIAMETRO
   (DIAMETRIC-REFERENCE (DIAMETRO) (HERITAGE) DDIAMETRO DDIAMETRO)))
 (VALOR ((100.0) OWNSLOT-TERMINAL DIMENSOES
  (POSSIBLE-VALUES (REAL) (HERITAGE) DIMENSOES DIMENSOES)
  (CARDINALITY (( 0 1 )) (HERITAGE) DIMENSOES DIMENSOES)))
 (TDIM ((TOL-DIMENSIONAL168) OWNSLOT-PRIMITIVE-DOWN DIMENSOES
  (DIAMETRIC-REFERENCE (TDIM-DE) (HERITAGE) DIMENSOES DIMENSOES)
  (POSSIBLE-VALUES (INSTANCE-OF TOL-DIMENSIONAL) (HERITAGE)
   DIMENSOES DIMENSOES))))

```

Tolerância de forma

```

(CILINDRICIDADE (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (SUBSET-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBCLASSES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (VALOR (NIL INSTANCESLOT-TERMINAL TOL-FORMA
  (POSSIBLE-VALUE (REAL) (HERITAGE) TOL-FORMA TOL-FORMA)
  (CARDINALITY (( 0 1 )) (HERITAGE) TOL-FORMA TOL-FORMA)))
 (TFORMA-DE
  (NIL INSTANCESLOT-PRIMITIVE-UP TOL-FORMA
   (DIAMETRIC-REFERENCE (TOL-FORMA) (HERITAGE) TOL-FORMA
    TOL-FORMA)))
 (COMMENT ("created 1/3/1993 by krisys") OWNSLOT-TERMINAL GLOBAL
  NIL))
 (SUBCLASS-OF ((TOL-FORMA) OWNSLOT-TERMINAL GLOBAL NIL))
 (INICIALIZA
  ((LAMBDA ()
   (PRINT "inicializa-tol-forma")
   (FORMAT T "~%Entre com valor da tolerancia de forma: ")
   (LET ((VALOR (READ)))
    (CHANGE-ATTRIBUTE-VALUE SELF 'VALOR (LIST VALOR))))
   INSTANCEMETHOD TOL-FORMA NIL))
 (HAS-INSTANCES ((CILINDRICIDADE171) OWNSLOT-TERMINAL GLOBAL NIL)))

```

4.5 Cálculos

```

(CALCULUS (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (SUBCLASS-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBCLASSES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (COMMENT ("created 1/3/1993 by krisys") OWNSLOT-TERMINAL GLOBAL
  NIL))
 (SUBSET-OF ((FUNC-TECNOL) OWNSLOT-TERMINAL GLOBAL NIL))

```

(HAS-SUBSETS
 ((CALC-CF-MAR-EXT CALC-CF-MER-EXT CALC-CF-MAR-INT
 CALC-CF-MER-INT CALC-CONE)
 OWNSLOT-TERMINAL GLOBAL NIL)))

Cálculos relativos ao chanfro maior externo

(CALC-CF-MAR-EXT (INSTANCE-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (ELEMENT-OF (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-INSTANCES (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-ELEMENTS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBSETS (NIL OWNSLOT-TERMINAL GLOBAL NIL))
 (COMMENT ("created 1/3/1993 by krisys") OWNSLOT-TERMINAL GLOBAL
 NIL))
 (CONICIDADE
 (NIL INSTANCESLOT-TERMINAL CALC-CF-MAR-EXT
 (POSSIBLE-VALUES (REAL) (REAL) CALC-CF-MAR-EXT
 CALC-CF-MAR-EXT)
 (CARDINALITY ((0 1)) ((0 1) CALC-CF-MAR-EXT
 CALC-CF-MAR-EXT)))
 (DIAM-MAIOR
 (NIL INSTANCESLOT-COMPLEX-DOWN CALC-CF-MAR-EXT
 (POSSIBLE-VALUES (INSTANCE-OF DDIAM-MAIOR)
 (INSTANCE-OF DDIAM-MAIOR) CALC-CF-MAR-EXT
 CALC-CF-MAR-EXT)
 (DIAMETRIC-REFERENCE (DIMENSAO-DE) (DIMENSAO-DE)
 CALC-CF-MAR-EXT CALC-CF-MAR-EXT)
 (CARDINALITY ((0 1)) ((0 1) CALC-CF-MAR-EXT
 CALC-CF-MAR-EXT)))
 (SUBSET-OF ((CALCULUS) OWNSLOT-TERMINAL GLOBAL NIL))
 (HAS-SUBCLASSES ((CHANFRO-MAIOR-EXT) OWNSLOT-TERMINAL GLOBAL NIL))
 (SUBCLASS-OF ((CLASSES) OWNSLOT-TERMINAL GLOBAL NIL)))

Anexo 2

Gráficos de objetos

Este anexo descreve a legenda utilizada nos gráficos que descrevem a hierarquia dos objetos no modelo apresentado no capítulo 5.

Na figura 1 são apresentados os tipos de relações de abstração e sua representação nos gráficos desenvolvidos pelo Krisys. A direção dos arcos é da ponta de uma caixa para o meio de outra. Então, sc-obj1 é uma subclasse de obj1, assim como inst-sc-obj2 é um objeto composto, parte de inst-sc-obj1.

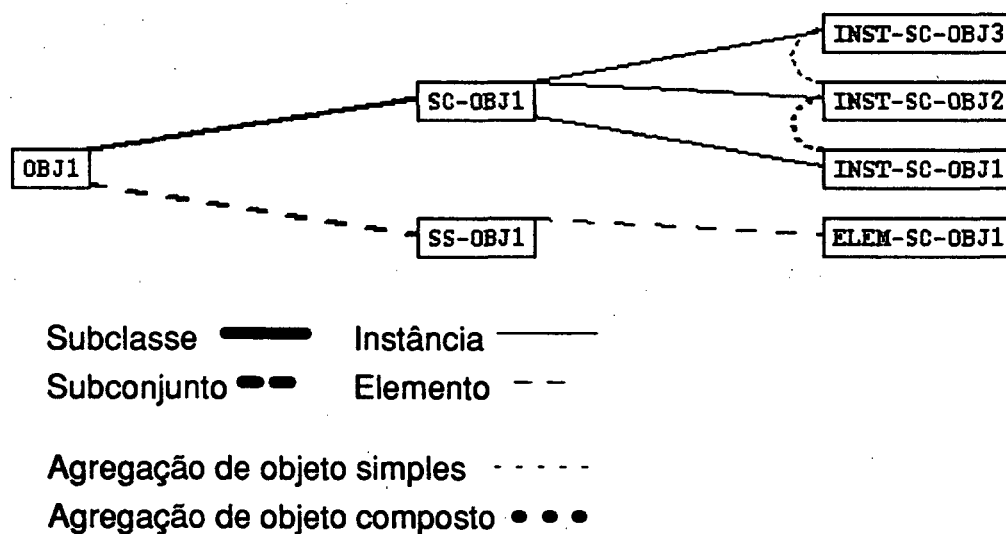


Figura 1 Relações de abstração no Krisys

Anexo 3

Geração do objeto "polia-esticadora"

Na versão atual da base de informações a interação com o usuário ocorre diretamente via o "prompt" do Krisys. Uma vez que a base estiver integrada ao AVEC, as mensagens provenientes dos objetos serão direcionadas ao Getme (capítulo 4). A listagem a seguir foi extraída da geração da peça polia-esticadora, utilizada como exemplo na seção 5.5. As informações digitadas pelo usuário estão em itálico de forma a facilitar sua identificação. Os comentários em negrito não fazem parte da interação com o usuário, são inseridos no sentido de explicar os procedimentos.

/* Usuário solicita a criação de uma peça chamada polia-esticadora. */
 >(send-msg 'pecas 'new-peca '(polia-esticadora))
/* Inicializa a Matriz de transformação para localizar o grupo em relação à peça. */
 Entre com posicao peca/sistema: (0.0 0.0 0.0 0.0 0.0 0.0)
/* Define os grupos que compõem a peça. */
 Entre com a lista de tipos de grupos que formam a peça: (*eixo-suporte gj-raio eixo-polia*)

Inicializa Grupo de Construção - EIXO-SUPORTE44
 Entre com posicao grupo/peca: (0.0 0.0 0.0 0.0 0.0 0.0)

Inicializa feature Primitiva do Grupo - CILINDRO47
 Entre com posicao feature/Grupo: (0.0 0.0 0.0 0.0 0.0 0.0)
/* O objeto cilindro pede ao usuário os valores para suas dimensões e tolerâncias. */
 Inicializa dimensoes da feature
 Entre com COMPRIMENTO: 54.0 Tolerancia dimensional superior: 0.2 Inferior: 0.2
 Entre com DIAMETRO: 45.0 Tolerancia dimensional superior: 0.2 Inferior: 0.2

Inicializa Superfície Cilíndrica da feature - SCILINDRICA52
 Entre com posicao superficie/feature: (0.0 0.0 0.0 0.0 0.0 0.0)
 Entre com o tipo da tolerancia de forma: nil

Inicializa Superfície Face da feature - SFACE54
 Entre com posicao superficie/feature: (0.0 0.0 0.0 0.0 0.0 0.0)
 Entre com o tipo de tolerancia de forma: nil

Inicializa Superfície Face da feature - SFACE56
 Entre com posicao superficie/feature: (0.0 0.0 54.0 0.0 0.0 0.0)
 Entre com o tipo de tolerancia de forma: nil

Inicializa feature detalhe externo - ASSENTO-ROLAMENTO58
/* No caso das "features" de detalhe, as superfícies são instanciadas em primeiro lugar pois, este é o primeiro atributo da lista de atributos da "feature" */

Inicializa Superfície cilíndrica da feature - SCILINDRICA59
 Entre com posicao superficie/feature: (0.0 0.0 0.0 0.0 0.0 0.0)
 Entre com o tipo de tolerancia de forma: cilíndricidade
 Entre com valor da tolerancia de forma: 0.0035

Inicializa Superfície face da feature - SFACE62
 Entre com posicao superficie/feature: (0.0 0.0 0.0 0.0 0.0 0.0)

Entre com o tipo de tolerancia de forma: nil

Inicializa Superficie face da feature - SFACE64

Entre com posicao superficie/feature: (0.0 0.0 10.0 0.0 0.0 0.0)

Entre com o tipo de tolerancia de forma: nil

Inicializa dimensoes da feature

Entre com COMPRIMENTO: 10.0 Tolerancia dimensional superior: 0.2 Inferior: 0.2

Entre com DIAMETRO: 45.000 Tolerancia dimensional superior: 0.015 Inferior: 0.002

Entre com posicao feature/Grupo: (0.0 0.0 39.0 0.0 0.0 0.0)