

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELETRICA
DEPARTAMENTO DE ENGENHARIA DA COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL
AGOSTO DE 1989

Este Exemplar corresponde à redação
final da tese defendida por Rogério Almeida
Barra e aprovado pela Comissão julgadora em 26/9/89

Jaime Augusto

MODELO DE PRODUTO COMO INTEGRADOR EM
SISTEMAS DE MANUFATURA

POR : Rogério Almeida Barra

ORIENTADOR: Prof. Dr. Manuel de Jesus Mendes

M. J. Mendes, Manuel de Jesus

Tese apresentada à Faculdade de Engenharia
Elétrica-UNICAMP como parte dos requisitos
exigidos para a obtenção do título de MESTRE
EM CIÊNCIAS.

À Lê, minha companheira.

A G R A D E C I M E N T O S

Ao Dr. Mendes pela orientação deste trabalho.

A Marcos Rodrigues pelas discussões técnicas.

À Célia pela datilografia.

A Kito pela ajuda nos desenhos.

Ao CTI pelo apoio.

R E S U M O

Os sistemas de manufatura da década de noventa se caracterizarão por uma utilização maciça de computadores, visando a automação das tarefas envolvidas no ciclo de projeto e manufatura de produtos. Neste sentido, as empresas têm feito grandes investimentos em sistemas CAD, CAM, máquinas CNC, robôs, etc. O conceito de Manufatura Integrada por Computador-CIM surge da necessidade de que estes sistemas cooperem entre si, a fim de que a empresa opere em um nível máximo de eficiência. Para isto, deve ser possível transferir, entre sistemas, dados que tenham um significado comum a eles, de tal forma que possam ser utilizados pelo sistema que os recebe. Uma parte significativa destes dados a serem trocados diz respeito a produtos. Neste contexto, este trabalho tem por objetivos definir que requisitos um modelo de produto deve satisfazer a fim de prover às aplicações em um sistema CIM todas as informações que elas necessitem, verificar os esforços de padronização deste modelo e a relevância destes no que diz respeito à implantação de um sistema que possibilite um efetivo compartilhamento de informações sobre produtos entre seus componentes. Como resultado final é proposta uma arquitetura para este sistema.

A B S T R A C T

Manufacturing systems of the nineties will be distinguished by massive computer use, aiming at automation of tasks along the product design and manufacturing cycle. Thus, companies have invested heavily on CAD/CAM systems, CNC machines, robots, etc. The concept of Computer Integrated Manufacturing-CIM arises from the necessity that these systems cooperate, so that the company operates at a maximum efficiency level. For this, it must be possible transfer, among these systems, data that have a meaning common to them, so that they can be used by the system receiving them. Significant part of the data to be exchanged relates to products. In this context, this work intends to define which requirements a product model must satisfy in order to provide to applications within a CIM system all information they need, verify the efforts to standardize this model and their relevance concerning the implementation of a system that enables an effective product information sharing among its components. As a final result an architecture for this system is proposed.

I N D I C E

CAPITULO 1 - INTRODUÇÃO.....	1
CAPITULO 2 - INTEGRAÇÃO DE SISTEMAS DE MANUFATURA.....	7
2.1. Modelos de Produto em CIM.....	15
CAPITULO 3 - MODELAGEM DE PRODUTO.....	18
3.1. Aspectos Gerais de Modelos de Produto.....	20
3.2. Abordagem Formal para Modelagem de Produto.....	23
3.2.1. Propriedades de Esquemas de Representação.....	25
3.3. Tipos de Informação e Esquemas de Representação em Mode- lo de Produto.....	29
3.3.1. Geometria e Topologia.....	29
3.3.1.1. Modelos Gráficos.....	30
3.3.1.2. Modelos de Superfície.....	33
3.3.1.2.1. Representação de Curvas e Su- perfícies.....	37
3.3.1.3. Modelos Sólidos.....	43
3.3.1.3.1. Modelos por Decomposição.....	55
3.3.1.3.2. Modelos Construtivos.....	59
3.3.1.3.3. Modelos por Fronteira.....	65
3.3.2. Informações Tecnológicas.....	75
3.4. Conclusão.....	80
CAPITULO 4 - PADRONIZAÇÃO EM TROCA DE DADOS DE DEFINIÇÃO DE PRODUTOS.....	81
4.1. Evolução Histórica dos Padrões.....	83
4.1.1. Desenvolvimento de IGES.....	84
4.1.2. Padrões de Âmbito Nacional.....	86
4.1.3. Esforços Atuais de Padronização.....	89
4.2. IGES.....	92
4.2.1. Formato de Dados e Estrutura do Arquivo IGES.....	94
4.2.1.1. Formato ASCII.....	95
4.2.1.2. Estrutura do Arquivo IGES.....	95
4.2.2. Entidades Geométricas.....	103

4.2.3.	Entidades de Anotação.....	106
4.2.4.	Entidades Estruturais.....	108
4.2.4.1.	Associação.....	108
4.2.4.2.	Subfiguras.....	114
4.2.4.3.	Propriedades.....	116
4.2.4.4.	Entidades para Composição de Desenhos de Engenharia.....	116
4.2.4.5.	Definição de Entidades pelo Usuário.....	118
4.2.4.6.	Referências Externas.....	120
4.3.	STEP.....	120
4.3.1.	A Linguagem Express.....	122
4.3.2.	Esquema Conceitual em STEP.....	128
4.3.3.	Formato do Arquivo Neutro STEP.....	131
4.3.4.	Níveis de Implementação em STEP.....	134
4.4.	Análise Comparativa dos Padrões IGES e STEP.....	139
CAPITULO 5 - SOLUÇÕES DE INTEGRAÇÃO COM REDE DE COMUNICAÇÃO..		143
5.1.	O Projeto TOP.....	144
5.2.	Troca de Dados de Produto Via IGES Usando Redes TOP.....	149
5.3.	Compartilhamento de Dados Via STEP Usando Redes TOP.....	152
5.3.1.	Implementação STEP Nível 1.....	153
5.3.2.	Implementação STEP Nível 2.....	154
5.3.3.	Implementação STEP Nível 3.....	156
CAPITULO 6 - CONCLUSÕES.....		164
BIBLIOGRAFIA.....		168

LISTA DE FIGURAS

FIGURA 1.1	- Fluxo de dados em um sistema de manufatura automatizado.....	3
FIGURA 2.1	- Arquitetura de um sistema de manufatura informatizado.....	8
FIGURA 2.2	- Troca de dados por formato neutro.....	11
FIGURA 2.3	- Arquitetura da base de dados distribuídas.....	13
FIGURA 2.4	- Modelo de produto em sistemas CIM.....	16
FIGURA 3.1	- Exemplo de desenho de engenharia.....	21
FIGURA 3.2	- Visão de modelamento em três níveis.....	23
FIGURA 3.3	- Definição de esquema de representação.....	24
FIGURA 3.4	- Consistência entre representações.....	28
FIGURA 3.5	- Desenho fisicamente irrealizável.....	31
FIGURA 3.6	- Modelo de arame ambíguo.....	32
FIGURA 3.7	- Modelo de arame de objeto não realizável fisicamente.....	32
FIGURA 3.8	- Modelo de arame de objeto com superfície curva..	32
FIGURA 3.9	- Superfície regradada.....	34
FIGURA 3.10	- Cilindro Tabulado.....	34
FIGURA 3.11	- Mapeamento entre representações geométricas....	35
FIGURA 3.12	- Modelo de superfície ambíguo.....	36
FIGURA 3.13	- Modelo de superfície fisicamente irrealizável..	37
FIGURA 3.14	- Interpolação por spline cúbica.....	39
FIGURA 3.15	- Curva de Bezier.....	40
FIGURA 3.16	- Spline básica cúbica.....	41
FIGURA 3.17	- Superfície de Bezier.....	42
FIGURA 3.18	- Esfera representada por B-splines racionais....	43
FIGURA 3.19	- Sólido não regular.....	46
FIGURA 3.20	- Sólidos cujas superfícies não são variedades topológicas.....	48
FIGURA 3.21	- Modelo plano de um cilindro.....	49
FIGURA 3.22	- Modelo plano de uma pirâmide.....	50
FIGURA 3.23	- Superfície aberta.....	51
FIGURA 3.24	- Superfície não orientada.....	52
FIGURA 3.25	- Determinação de orientabilidade de uma superfície.....	53

FIGURA 3.26	- Superfície que intercepta a si mesma.....	53
FIGURA 3.27	- Sólidos que atendem a fórmula de Euler-Poincaré	54
FIGURA 3.28	- Instâncias de uma primitiva.....	56
FIGURA 3.29	- Enumeração exaustiva de um sólido.....	57
FIGURA 3.30	- Representação por árvore octal.....	58
FIGURA 3.31	- Modelo por semi-espaços de um cilindro.....	60
FIGURA 3.32	- Modelo CSG de um sólido.....	61
FIGURA 3.33	- Coleções de sólidos primitivos para modelos CSG	62
FIGURA 3.34	- Interseção regularizada entre conjuntos.....	63
FIGURA 3.35	- Relacionamentos entre entidades topológicas....	66
FIGURA 3.36	- Sólidos com auto-laço e multigrafo.....	67
FIGURA 3.37	- Sólido a ser modelado por fronteira.....	68
FIGURA 3.38	- Estrutura de dados para modelo B-REP.....	69
FIGURA 3.39	- Modelos planos dos operadores de Euler.....	72
FIGURA 3.40	- Operadores DFCLH e CFDLH.....	73
FIGURA 3.41	- Definição de um sólido por operadores de Euler.	74
FIGURA 3.42	- Grafo de adjacência de faces para uma peça.....	77
FIGURA 3.43	- Grafos de adjacência de faces para formas ca- racterísticas.....	78
FIGURA 4.1	- Uso de tradutores diretos para transferência de dados entre sistemas CAD.....	82
FIGURA 4.2	- Transferência de dados usando formato neutro....	83
FIGURA 4.3	- Arquitetura PDDI.....	88
FIGURA 4.4	- Relacionamento entre sistemas CAD e CAD*I.....	90
FIGURA 4.5	- Evolução dos padrões para troca de dados de pro- dutos.....	92
FIGURA 4.6	- Conjunto de entidades representado em IGES.....	94
FIGURA 4.7	- Estrutura do arquivo IGES.....	96
FIGURA 4.8	- Exemplo de seção inicial.....	97
FIGURA 4.9	- Entrada diretório para uma entidade.....	99
FIGURA 4.10	- Representação de um arco e um ponto em IGES....	103
FIGURA 4.11	- Entidade curva composta.....	104
FIGURA 4.12	- Superfície spline paramétrica.....	104
FIGURA 4.13	- Entidade flash.....	104
FIGURA 4.14	- Entidade cota.....	107
FIGURA 4.15	- Entidade rótulo geral.....	107
FIGURA 4.16	- Entidade nota de indicador.....	107
FIGURA 4.17	- Entidade símbolo geral.....	107

FIGURA 4.18 - Entidade área seccionada.....	107
FIGURA 4.19 - Entidade dimensão angular.....	107
FIGURA 4.20 - Entidade dimensão de diâmetro.....	107
FIGURA 4.21 - Entidade dimensão de raio.....	107
FIGURA 4.22 - Entidade dimensão linear.....	108
FIGURA 4.23 - Esquema de subfigura em IGES.....	114
FIGURA 4.24 - Exemplo de utilização da entidade desenho.....	118
FIGURA 4.25 - Volume de visualização.....	117
FIGURA 4.26 - Instâncias de entidades definidas pelo usuário.	118
FIGURA 4.27 - Triângulo definido por macro.....	119
FIGURA 4.28 - Exemplo de referência externa.....	121
FIGURA 4.29 - Modelo de três camadas em STEP.....	122
FIGURA 4.30 - Escopo dos elementos em Express.....	124
FIGURA 5.1 - O modelo OSI/ISO.....	145
FIGURA 5.2 - Blocos construtivos em Top 3.0.....	148
FIGURA 5.3 - Troca de dados de produtos via IGES em redes TOP	149
FIGURA 5.4 - Implementação de pré e pós-processadores IGES...	150
FIGURA 5.5 - Arquitetura de pré e pós-processadores STEP.....	154
FIGURA 5.6 - Troca de dados de produtos em implementação ní- vel 2 - STEP.....	155
FIGURA 5.7 - Arquitetura para uma base de dados distribuída heterogênea.....	157
FIGURA 5.8 - Modelo de interação em RDA.....	159
FIGURA 5.9 - Diagrama temporal do protocolo bifásico em CCR..	160
FIGURA 5.10 - Arquitetura de sistema para compartilhamento de dados de produtos.....	161

LISTA DE TABELAS

TABELA 3.1 - Correspondência entre elementos topológicos e geométricos.....	47
TABELA 3.2 - Estrutura-asa para o sólido da Fig. 3.37.....	70
TABELA 3.3 - Características de vários tipos de modelos.....	80
TABELA 4.1 - Parâmetros de seção global.....	98
TABELA 4.2 - Entidades geométricas em IGES.....	105
TABELA 4.3 - Entidades de anotação em IGES.....	106
TABELA 4.4 - Parâmetros para definição de associação.....	110
TABELA 4.5 - Parâmetros para instância de associação.....	111
TABELA 4.6 - Definição de associação em grupo.....	112
TABELA 4.7 - Instância de associação de grupo.....	112
TABELA 4.8 - Definição de índice para referência de arquivos externos.....	113
TABELA 4.9 - Instância de índice para referência de arquivos externos.....	113
TABELA 4.10 - Entidades do esquema de subfigura em IGES.....	115
TABELA 4.11 - Tipos de informação suportados por IGES e STEP.	141
TABELA 5.1 - Conjunto mínimo de entidades IGES especificado em TOP.....	152

CAPITULO 1

INTRODUÇÃO

A necessidade de se fazer presente em um mercado internacional cada vez mais competitivo, em que são exigidos produtos a baixos custos, alta qualidade e que sejam entregues ao mercado em curto prazo, faz com que as empresas sejam forçadas a reconsiderar a forma como operam seus sistemas de manufatura /CON 85/. A crescente complexidade dos produtos e processos de fabricação trouxe consigo a necessidade de especialização de tarefas, a fim de que fosse mantida a eficiência do sistema produtivo. O problema é que, ao mesmo tempo, esta especialização torna complexos os fluxos de informação e material do sistema, exigindo recursos cada vez maiores para seu gerenciamento. A Manufatura Integrada visa minimizar os custos e tempo associados a este gerenciamento, o que é de grande importância para o aumento da eficiência da empresa, uma vez que esta tarefa não adiciona valor algum ao produto /ING 85/.

A análise dos elementos do sistema de manufatura de um ponto de vista dos objetivos da empresa como um todo e o reconhecimento dos aspectos comuns a estes elementos são passos essenciais para que, com um mínimo de sobrecarga administrativa, os materiais e informações corretos estejam nos lugares corretos, nos instantes corretos, ou seja, para que a empresa opere de forma integrada. Em última análise, integração é o efeito de ter-se em comum os aspectos apropriados de diversas atividades da empresa, o que permite que esta passe a ser encarada como uma única entidade, portanto, operando coerentemente e com maior produtividade /BEL 87/. Face às considerações acima será analisada a seguir a importância da utilização de computadores para a consecução destes objetivos.

Utilização de Computadores em Sistemas de Manufatura

A utilização de computadores em uma empresa pode ser considerada sob dois enfoques:

a) como uma ferramenta para a automação de funções do sistema produtivo, desde a concepção até a entrega de um produto. A Fig.

1.1 apresenta um possível diagrama funcional de um sistema de manufatura automatizado. A partir deste serão aqui analisadas as funções e fluxos de dados relevantes a este trabalho, considerando-se que sistemas computacionais sejam utilizados em cada uma das funções:

- Projeto Auxiliado por Computador (CAD):

uma vez definidos os requisitos básicos de um produto, os sistemas CAD são utilizados para a definição da geometria, tolerâncias e demais parâmetros tecnológicos de um produto;

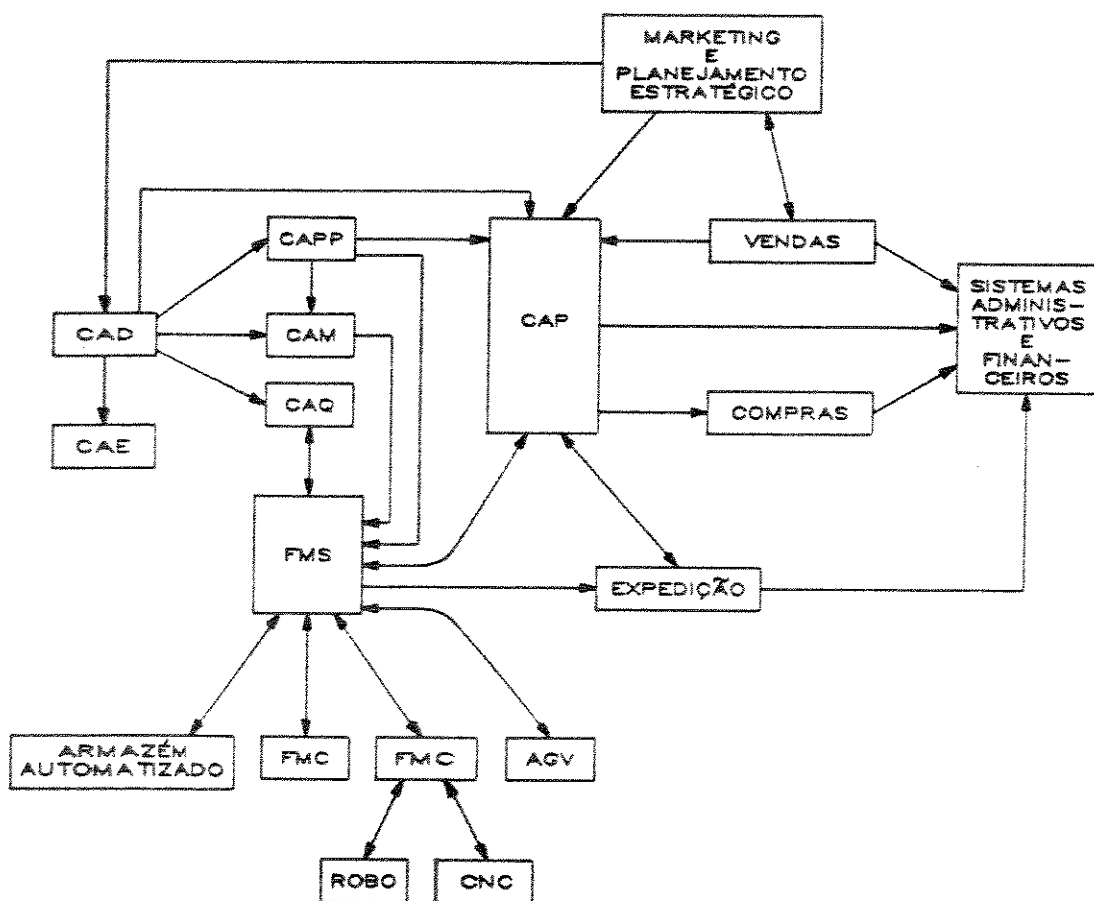


Fig. 1.1 - Fluxo de dados em um sistema de manufatura automatizado

- Engenharia Auxiliada por Computador (CAE):
são sistemas voltados à análise de engenharia como, por exemplo, análise estrutural e térmica por elementos finitos. Para tanto, utilizam informações geométricas e constantes que definem o material de que é feito o produto;
- Planejamento de Processos Auxiliado por Computador (CAPP):
através do acesso a dados que definem o produto e do conhecimento dos processos de fabricação disponíveis define máquinas, ferramentas, condições de usinagem e o roteiro para a fabricação de um produto;
- Manufatura Auxiliada por Computador (CAM):
realiza as funções relativas à engenharia de produção, tais como projeto de ferramental, fixações e gabaritos e gera programas para os equipamentos automatizados do chão de fábrica, tais como máquinas de comando numérico e robôs. Utiliza para isso informações sobre o produto (geométricas e tecnológicas) e informações sobre processos de fabricação geradas pelo sistema CAPP;
- Controle de Qualidade Auxiliado por Computador (CAQ):
a partir de informações sobre o produto gera um plano de inspeção, o que pode incluir programas para máquinas de medição controladas por computador (CAT). Além disso, processa informações provenientes do chão de fábrica sobre qualidade;
- Planejamento e Controle de Produção Auxiliados por Computador (CAP):
a partir de informações que recebe de vendas gera um plano de produção de produtos acabados (plano mestre). Através do acesso à estrutura do produto em termos de seus componentes e ao inventário destes, determina ordens de fabricação ou requisições de compra de componentes. Utiliza, também, informações sobre processos de fabricação

dos produtos para alocar centros de trabalho (FMS) onde são realizadas as operações de manufatura. Definidos os centros, a estes são enviadas as ordens de produção. Além disso, realiza as funções de controle de produção, utilizando dados provenientes do chão de fábrica;

- **Sistemas Flexíveis de Manufatura (FMS):**

recebe ordens de fabricação do sistema CAP. Para sua execução utiliza o plano de fabricação, plano de inspeção e programas de comando numérico e robô definidos para um produto. Realiza o planejamento de produção de curto prazo. Além disso, gerencia e controla os níveis mais baixos do chão de fábrica (células flexíveis de manufatura (FMC) e sistemas automatizados de transporte (AGV)).

Estes sistemas têm sido empregados pelas empresas visando ganhos de produtividade em suas funções específicas, melhoria da qualidade dos produtos e sistemas de manufatura mais flexíveis. Porém, a questão do gerenciamento do fluxo de informação não é solucionada simplesmente pela implantação de sistemas computacionais orientados à automação industrial. Pelo contrário, este fluxo se torna mais complexo devido à maior eficiência local das funções, portanto, gerando e requisitando informações em uma taxa maior. Sendo assim, pode ser observado um outro enfoque para a utilização de computadores em sistemas de manufatura.

b) como uma ferramenta para integração das funções do sistema produtivo. Desta forma, a Manufatura Integrada por Computador-CIM pode ser vista como a tecnologia que viabiliza a cooperação entre as diversas funções do ciclo de projeto e manufatura de produtos visando maximizar a eficiência da empresa. A utilização de computadores permite o gerenciamento efetivo da grande quantidade de informações que circula em uma empresa.

Uma vez considerados os aspectos gerais do sistema de manufatura, sobre o qual se estará tratando neste trabalho, no capítulo dois serão analisadas estratégias para a cooperação entre as funções deste sistema quanto à satisfação do objetivo de integração. Além disto, será caracterizado o papel a ser desempenhado pelo modelo de produto em um sistema CIM. No capítulo três, através de uma abordagem formal, serão estabelecidos requisitos a serem impostos ao modelo de produto, a fim de que este possa ser utilizado como fonte de informação para as várias aplicações em CIM. Serão também apresentados esquemas de representação para as informações que definem um produto (geometria, topologia, tolerâncias, etc.). O capítulo quatro tratará dos padrões para modelo de produto, enfocando, principalmente, IGES, que tem sido adotado pelo mercado, e STEP, que se constitui em um esforço de padronização a nível internacional, procurando mostrar como este deverá suprir as deficiências de IGES como mecanismo de troca de informações sobre produtos entre os componentes de um sistema CIM. Finalmente, no capítulo cinco serão propostas arquiteturas de sistemas, que fazem uso de padrões para integração de seus componentes, tanto a nível da rede de comunicação de dados quanto da estrutura de armazenamento de informação.

CAPITULO 2

INTEGRAÇÃO DE SISTEMAS DE MANUFATURA

Como mostra a Fig. 2.1, a configuração de uma empresa como geradora e usuária de informações resulta em uma arquitetura de sistema que é hierárquica, distribuída e heterogênea /GIB 88/, uma vez que normalmente os sistemas são provenientes de vários fabricantes. Devido a isto e ao fato de os sistemas serem projetados para operar de forma autônoma, sem levar em consideração os outros sistemas, tem-se observado a tendência de formação de "ilhas de automação" /LAM 86/. Sendo assim, apesar da geração e do processamento de informações serem otimizados, o fluxo daquelas permanece em sua forma convencional, utilizando saída em papel de um sistema para reentrada de dados em outro, o que, além de ineficiente, pode gerar erros. Além disso, as bases de dados dos sistemas podem conter dados conflitantes ou redundantes e em formatos incompatíveis, o que implica em dispendar recursos para gerenciar esta questão. Idealmente, uma informação só deve ser gerada uma vez, sendo prontamente acessível a qualquer sistema que dela faça uso /RAL 87/. No caso de alguma redundância de dados ser conveniente, este fato deve ser conhecido pelos sistemas a fim de que seja garantida a integridade da informação.

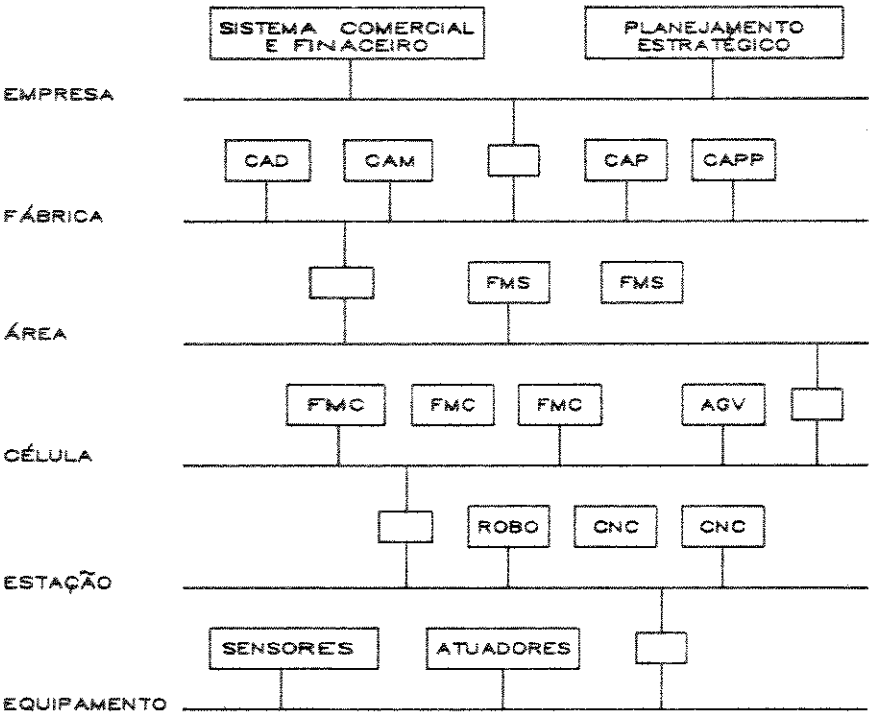


Fig. 2.1 - Arquitetura de um sistema de manufatura informatizado

Em um sistema integrado, as diversas funções que o compõem cooperam entre si visando atingir os objetivos globais da empresa. Para que isto aconteça, ou seja, para que CIM ocorra, as seguintes condições devem ser satisfeitas /AME 85/:

- todas as funções de processamento e funções de gerenciamento relacionadas devem ser expressas por dados;
- estes dados devem estar em uma forma que possa ser gerada, transformada, movida e armazenada por computadores;
- os dados são movidos livremente entre funções no sistema ao longo do ciclo de projeto e manufatura de um produto.

Uma primeira implicação dos requisitos acima diz respeito à comunicação de dados. Deve ser possível transferir dados de uma função do sistema CIM para outra. A utilização de redes de computadores vem de encontro a esta necessidade. A rede deve prover às aplicações os serviços que assegurem uma confiável transmissão e recepção de dados, de tal forma que sejam transparentes àquelas os procedimentos necessários à consecução deste objetivo. Os requisitos impostos ao sistema de comunicação em termos de tempos de espera permitidos, frequência e comprimentos de mensagens, variarão conforme o nível envolvido na arquitetura do sistema CIM (ver Fig. 2.1) /MEN 89/.

A transferência de dados, embora necessária, não é suficiente para garantir a integração das funções do sistema de manufatura. É preciso haver um acordo entre os sistemas envolvidos a nível de informação, aqui definida como o significado que os dados carregam, para que uma aplicação em um sistema possa operar corretamente sobre estes. Como foi observado na introdução deste capítulo, a integração resulta da correta identificação dos aspectos comuns às partes envolvidas. Portanto, fazem-se necessários modelos de informação que definam completamente e de forma não-ambígua a semântica dos elementos comuns aos sistemas sendo integrados. A existência destes modelos significa que, no que diz respeito aos dados compartilhados pelas

aplicações, a semântica deixa de ser definida por estas e passa a ser definida pelos modelos de informação. Estes se traduzirão em dicionários de dados, em que não deve haver sinônimos (diferentes dados com igual significado) nem homônimos (dados com múltiplos significados). Além disso, as informações devem ser modeladas em múltiplos níveis de abstração, uma vez que um mesmo conjunto de dados pode ser visto de formas diferentes por aplicações diferentes.

Uma vez verificados os níveis envolvidos na integração de sistemas computacionais, serão analisadas abordagens que podem ser empregadas visando a integração de um sistema de manufatura cujo contexto é definido pelas Figuras 1.1 e 2.1.

Partindo-se do pressuposto que os sistemas estejam interconectados através de redes de computadores que garantam a transferência de dados entre eles, pode-se buscar a cooperação das seguintes maneiras /GIB 88/:

- Utilização de tradutores diretos:

Os aspectos comuns a cada dois sistemas são analisados e, a partir disto, desenvolvidas interfaces entre eles. Desta forma, considerando-se que as interfaces devem ser bidirecionais, para N sistemas pode ser necessário o desenvolvimento de até $N(N-1)$ interfaces, o que, logicamente, implica em altos custos. Além disto, é preciso considerar a dinâmica de um sistema de manufatura. O surgimento de novas ferramentas voltadas à automação industrial pode tornar conveniente a reconfiguração do sistema, conseqüentemente, demandando o desenvolvimento de novas interfaces. Mesmo no caso de alterações de versão de um sistema computacional, todas as interfaces a ele associadas precisam ser atualizadas.

- Troca de dados usando formatos neutros:

Nesta abordagem são definidos modelos independentes das aplicações para as informações a serem compartilhadas por estas.

Sendo assim, para cada aplicação precisam ser desenvolvidas duas interfaces, uma que mapeia do modelo da aplicação para o modelo neutro e outra que faz a operação inversa, portanto, não sendo influenciados outros sistemas quando um deles é modificado ou um novo sistema é introduzido. Como mostra a Fig. 2.2, para que um sistema A acesse informações contidas no sistema B, este deve gerar um arquivo neutro que é transferido para A e, então, convertido no formato deste. Se este modelo neutro de informações capturar toda a semântica compartilhada pelos sistemas, o sistema A poderá operar corretamente sobre as informações recebidas de B.

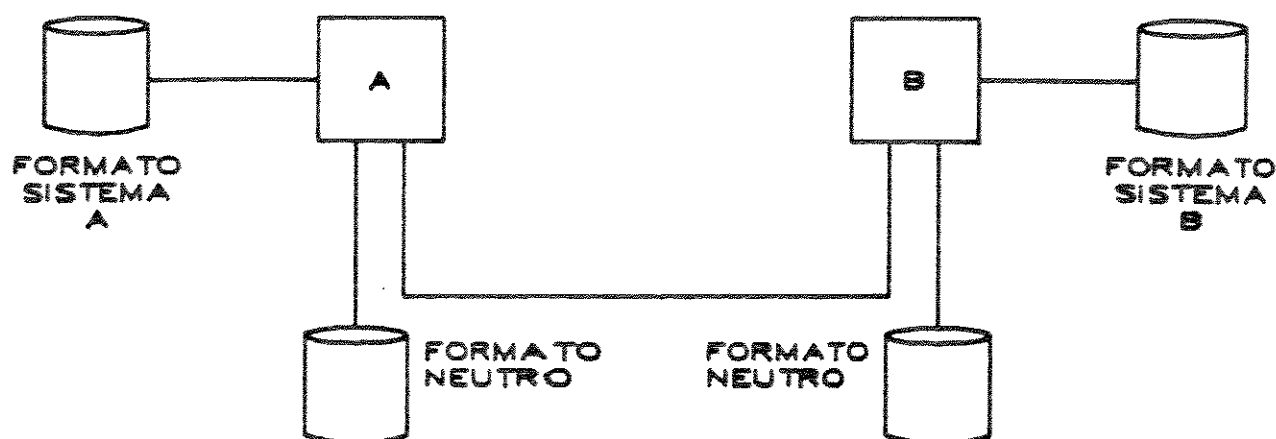


Fig. 2.2 - Troca de dados por formato neutro

Um problema desta abordagem é que os mesmos dados existem em vários lugares na empresa. Como muitas atividades são dinâmicas, iterativas e interrelacionadas, e esta redundância não é gerenciada (centralmente), estes dados podem ser modificados de forma inconsistente. Além disso, como a definição das informações requeridas é função do sistema que recebe e a definição das informações a serem transmitidas é função do sistema que envia, o arquivo neutro poderá conter informações que, embora conhecidas pelo modelo neutro, não sejam relevantes para o sistema que recebe. Da mesma forma, informações que seriam relevantes podem não ter sido convertidas para o arquivo neutro

pelo sistema que envia. Em síntese, a questão é que a transferência de informações não é baseada em requisições feitas pelo sistema que as demandam.

- Integração por base de dados distribuída:

Em um sistema integrado, a informação correta se encontra prontamente disponível a qualquer função que a solicite. No caso de sistemas de manufatura, as seguintes características devem ser consideradas quando se busca atingir o objetivo acima citado /BRA 87/:

- dados que devem ser compartilhados por sistemas se encontram distribuídos ao longo destes;
- a priori as aplicações executadas nos sistemas não conhecem a localização dos dados;
- frequentemente é justificável, devido ao desempenho requerido às aplicações, que haja redundância de dados;
- não são pré-determinados os instantes em que as informações serão requeridas.

O quadro que se configura a partir destas características requer um sistema administrador de informações que gerencie todo o acesso aos dados distribuídos /RUI 88/. A semântica dos elementos comuns aos sistemas sendo integrados, ou seja, os modelos de informação, deve ser conhecida por este sistema. Em termos computacionais, isto se traduz em um sistema gerenciador de base de dados distribuída que suporte um modelo de dados que capture a semântica definida pelos modelos de informação. Além disto, este sistema deve ser responsável pela consistência, integridade e segurança dos dados. Como mostra a Fig. 2.3, podem ser distinguidas três camadas na arquitetura da base de dados distribuída. O nível externo global define a parte da base de dados vista por uma aplicação, uma vez que, no caso geral, esta estará interessada em um subconjunto do modelo de informações. O nível conceitual global modela a totalidade das informações

compartilhadas pelas aplicações. O nível interno representa o particionamento dos dados do nível conceitual entre os sistemas sendo integrados. Este nível lida com a forma como os dados são realmente armazenados.

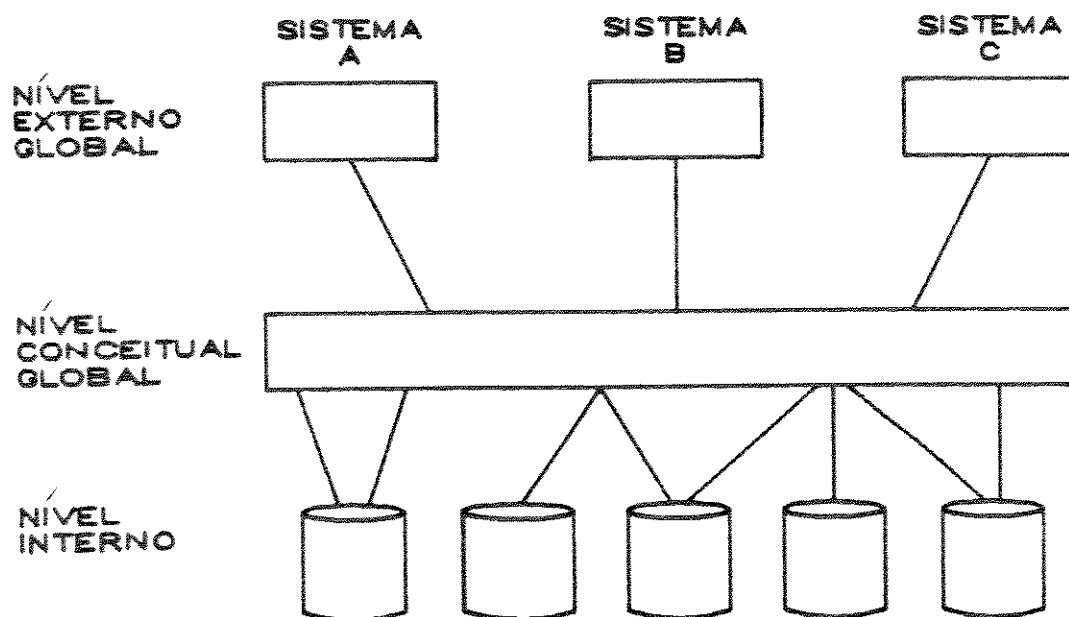


Fig. 2.3 - Arquitetura da base de dados distribuída

A heterogeneidade dos sistemas implica que, para a implementação desta arquitetura, o sistema de base de dados distribuída deve prover uma linguagem neutra de definição de dados, que permita definir cada uma das camadas e os mapeamentos entre elas. Deve também prover uma linguagem neutra de manipulação de dados que permita a uma aplicação acessar os dados de forma transparente, ou seja, sem conhecer como estes são armazenados em cada um dos sistemas.

Esta abordagem, por gerenciar efetivamente o fluxo de informações entre as funções do sistema de manufatura, é a mais adequada no que concerne ao objetivo de integração discutido neste capítulo.

Um último ponto a ser discutido diz respeito à importância dos esforços de padronização para a viabilização econômica do processo de integração de sistemas. Embora soluções específicas possam ser desenvolvidas para cada sistema de manufatura, esta abordagem não atende o requisito de flexibilidade do sistema. Este requisito pode ser visto sob dois aspectos: dado um conjunto de produtos, o sistema produtivo deve poder atender rapidamente a mudanças nas ordens de produção, permitindo, assim, uma melhor resposta às variações nas condições de mercado, uma vez que se viabiliza a produção em pequenos lotes. Além disto, o sistema deve ser reconfigurável, de modo a atender a mudanças na linha de produtos, evolução tecnológica dos sistemas computacionais e novos processos de fabricação. O fato que a modificação ou introdução de um sistema afetará todos aqueles com os quais interage implica em altos custos de manutenção do sistema integrado, podendo tornar indesejáveis alterações, portanto, não permitindo que a empresa se utilize dos sistemas mais adequados à sua configuração. Considerado este aspecto dinâmico dos sistemas de manufatura, evidencia-se a necessidade que os sistemas computacionais estejam aptos, a priori, a interagir uns com os outros, ou seja, são desejáveis sistemas abertos. Como, em geral, estes sistemas são heterogêneos, a consecução deste objetivo envolve, necessariamente, a adoção de padrões. No que diz respeito à transferência de dados, os projetos MAP (Manufacturing Automation Protocol) e TOP (Technical and Office Protocol) definem protocolos para interação de sistemas heterogêneos nos ambientes de chão de fábrica e escritórios, respectivamente. Da mesma forma, como será visto neste trabalho, estão sendo empreendidos esforços em âmbito internacional que visam definir modelos para as informações relevantes aos sistemas de manufatura. Uma vez analisados os aspectos gerais de integração de sistemas de manufatura, a próxima seção introduzirá os modelos que são objeto do presente trabalho.

2.1. MODELOS DE PRODUTO EM CIM

Conforme foi observado na seção anterior, para que vários sistemas cooperem entre si para a solução de tarefas, fazem-se necessários modelos de informação que descrevam os aspectos comuns a estes sistemas. A análise das funções e fluxo de dados em um sistema de manufatura feita na introdução deste trabalho mostra que um sistema CIM deve, entre outros, incluir os seguintes modelos /KIM 84b/:

- modelos de produto: representam toda a informação sobre produtos que é utilizada em sistemas de manufatura;
- modelos de processos: descrevem os recursos disponíveis para a manufatura de produtos, tais como máquinas, ferramentas, linhas de produção e processos de fabricação;
- modelos de produção: descrevem as informações necessárias ao gerenciamento e controle de produção, tais como ordens de fabricação e inventário.

As informações que um modelo de produto deve conter podem ser divididas em dois tipos /ELG 85/:

- informações de definição do produto: toda a informação geométrica e tecnológica que em um sistema de manufatura convencional é representada por desenhos de engenharia. O presente trabalho se concentrará em modelos deste tipo de informação;
 - informações de suporte ao produto: são informações sobre produtos que são geradas a partir de informações de definição. Entre outras, podem ser citadas: roteiro de fabricação do produto, programas para máquinas de comando numérico, resultados de análise de engenharia (por exemplo, elementos finitos), e plano de inspeção.
- Um modelo de produto genérico deve estabelecer os relacionamentos entre as informações de suporte e aquelas

de definição, de tal forma que, quando a definição for modificada, fiquem explícitas que informações de suporte precisam ser atualizadas.

A Fig. 2.4 apresenta uma visão lógica de modelos de produto, aqui restritos às informações de definição, no contexto do sistema de manufatura definido pela Fig. 1.1 /BJO 84/. Para que estes modelos atuem como elemento de integração entre os sistemas da Fig. 2.4, devem ser definidos de forma que superem as seguintes dificuldades /WIL 87b/:

- os mesmos dados podem ter significados diferentes para sistemas diferentes;
- a mesma informação pode ser representada por dados diferentes em sistemas diferentes;
- os sistemas operam em diferentes níveis de abstração dos produtos.

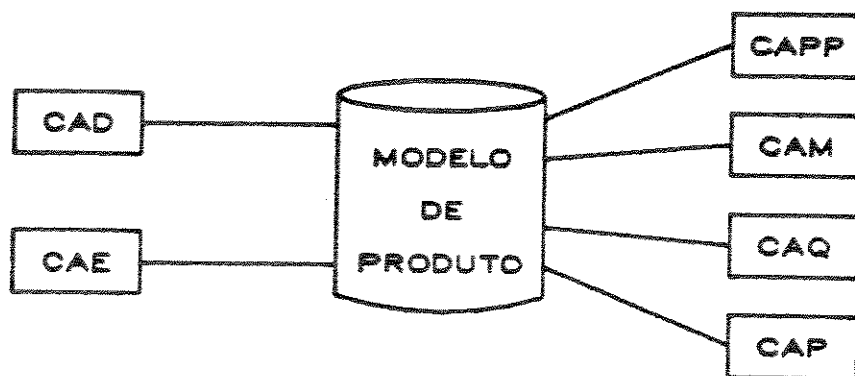


Fig. 2.4 - Modelo de produto em sistemas CIM

Os problemas acima podem ser verificados, por exemplo, na troca de dados de produtos entre um sistema CAD e um sistema CAPP. O usuário do sistema CAD define um produto, introduzindo informações geométricas, tolerâncias e outras informações tecnológicas. Tipicamente, tolerâncias são representadas por

símbolos padronizados que se traduzem em primitivas básicas como textos, linhas e arcos. Porém, o relacionamento destas tolerâncias com os elementos geométricos a que se referem, não é definido. Esta forma de representação é adequada à geração de desenhos que são interpretados por especialistas. Porém, para que o sistema CAPP possa definir automaticamente os processos de fabricação do produto, deve ter acesso às informações sobre relacionamentos entre tolerâncias e elementos geométricos, as quais não podem ser capturadas das primitivas básicas que recebeu do sistema CAD, uma vez que estes dados podem ter para ele um significado diferente daquele pretendido pelo sistema CAD. Sendo assim, o usuário do sistema CAPP terá que interpretar e editar os dados, de maneira que a informação entrada no sistema CAD possa ser recuperada, ou seja, a mesma informação é inserida no sistema como um todo mais de uma vez, com todas as desvantagens já citadas. Outro aspecto a ser considerado diz respeito ao nível de abstração das informações com que lidam os dois sistemas. Embora projetistas pensem em termos de formas características (furos, rebaixos, chanfros) quando inserem informações no sistema CAD, este as representa em termos de elementos geométricos, ou seja, informação de mais baixo nível. Por outro lado, existe uma forte associação entre formas características e processos de fabricação, fato este que é utilizado pelo sistema CAPP para definir os processos. Como anteriormente, esta informação terá que ser regenerada pelo usuário do sistema CAPP que recebeu os dados do sistema CAD.

A discussão acima mostra que mecanismos de transferência de dados de produtos adequados a sistemas CIM devem ser baseados em modelos que definam precisamente a semântica comum às aplicações no que diz respeito a produtos. A análise das informações que compõem o modelo de produto é objeto do próximo capítulo.

CAPITULO 3

MODELAGEM DE PRODUTO

Um modelo é uma representação de entidades físicas ou abstratas, a qual permite uma mais fácil observação de propriedades destas /FOL 84/. Para tanto, o modelo deve descrever as características importantes do objeto ou processo em questão, em um nível compatível com as respostas que dele se espera obter, ou seja, deve ser visto como uma abstração do objeto ou processo real. O sucesso de um modelador depende, basicamente, da capacidade deste em definir os elementos significativos do objeto ou processo sendo modelado e seus relacionamentos.

Modelos são criados a fim de evitar o custo, perigo, inconveniência ou impossibilidade de se manipular o objeto ou processo real. É frequentemente mais fácil simular, testar e prever o comportamento destes através de um modelo que testar, medir ou experimentar com o objeto real /MOR 85/. Além disso, o modelo pode ser utilizado como um meio de transmissão de informações entre várias funções que o utilizem.

Modelos têm sido largamente utilizados em diversas ciências. Modelos físicos de formas tridimensionais tais como construções, navios, carros e aviões normalmente compartilham as dimensões relativas e a aparência dos objetos reais, porém, em escala reduzida. Como exemplo de aplicação de modelos físicos pode ser citado o estudo de propriedades aerodinâmicas de um avião através da obtenção de parâmetros em túnel de vento para um modelo do avião em escala compatível com o tamanho do túnel. Modelos moleculares usados em química compartilham o arranjo relativo dos vários átomos das moléculas entre si, porém, não levando em consideração outras propriedades. Modelos matemáticos representam aspectos do comportamento de fenômenos associados a vários campos da ciência e engenharia em termos de dados numéricos e equações. Entre outros podem ser citados modelos voltados à astronomia (onde modelos de nascimento, morte e interação de estrelas permitem estudar teorias que, através de observação direta, seriam impraticáveis devido ao excessivo tempo envolvido), à sociologia (onde a manipulação direta de grupos de pessoas para estudo poderia causar problemas éticos), sistemas ecológicos e económicos (onde as condições são difíceis de controlar). A

própria evolução da matemática deve-se, em grande parte, à sua utilização como ferramenta para modelar fenômenos de outras ciências.

A validação de um modelo pode se dar através da comparação entre o comportamento previsto por este e aquele realmente observado [BLI 84]. Caso haja discrepância, o modelo deve ser ajustado, permitindo, assim, que este seja utilizado de forma confiável. Porém, há casos (por exemplo sistemas ecológicos) em que são requeridos períodos excessivamente longos de observação para a coleta de dados que permitam a verificação dos resultados previstos, existindo, portanto, incertezas inerentes ao modelo.

O desenvolvimento de computadores teve forte impacto sobre a expansão do uso e utilidade de técnicas de modelagem. Através de modelos computacionais, a simulação de sistemas mais complexos e de maior porte, cujo tratamento é impraticável sem a utilização do poder de processamento de computadores, se tornou viável, estendendo amplamente a gama de aplicações de técnicas de modelagem.

Modelos computacionais consistem em dados e procedimentos. É uma questão central para sua construção a decisão entre o que é armazenado explicitamente e o que é computado. Para isso, são requeridos o conhecimento do problema sendo modelado e das propriedades da técnica de modelagem utilizada. Uma vez feitas estas considerações iniciais sobre modelos, no tópico seguinte serão caracterizados modelos de produto.

3.1. ASPECTOS GERAIS DE MODELOS DE PRODUTO

Desenhos de engenharia têm sido largamente utilizados pelas empresas como meio de representação de informações relativas ao projeto e manufatura de produtos. Sendo assim, eles podem ser vistos como modelos de produtos. Através de um conjunto de vistas bidimensionais, auxiliado por símbolos padronizados para representar dimensões, tolerâncias, acabamento superficial,

material (Fig. 3.1), os especialistas são capazes de responder questões a respeito de produtos. Um fato a ser ressaltado neste tipo de modelo é a sua generalidade, ou seja, a capacidade de um mesmo modelo ser utilizado para extrair informações para tarefas diversas. Em contraposição temos, por exemplo, modelos matemáticos, em que cada novo tipo de problema requer um novo modelo. Esta característica genérica de desenhos de engenharia faz com que eles atuem como meio de comunicação entre as funções envolvidas no projeto e manufatura de um produto.

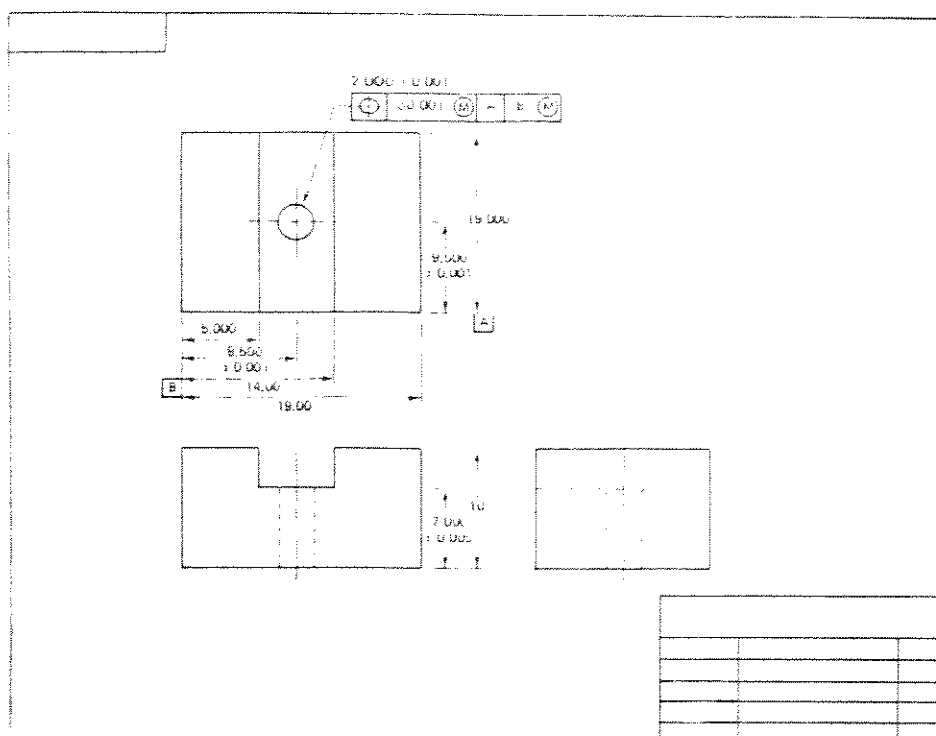


Fig. 3.1 - Exemplo de desenho de engenharia

Desenhos de engenharia estão inseridos em um ambiente caracterizado pela manipulação de informações por pessoas, sendo, portanto, inerentemente redundantes e incompletos. Frequentemente ambigüidades têm que ser resolvidas pelos especialistas que os utilizam. Por outro lado, em um ambiente de manufatura integrada por computador, caracterizado por aplicações automatizadas, as informações são manipuladas por programas de computador, fazendo-se necessário um modelo de produto completo e não-

ambíguo, de tal forma que as mais variadas questões a respeito de produtos possam ser respondidas por programas de aplicação. Dentre estas, podem ser citadas:

- Qual é a aparência do produto (representação em dispositivos gráficos)?
- Quais são suas propriedades geométricas (área, volume, momentos de inércia)?
- As partes constituintes do produto interferem umas com as outras numa montagem?
- O produto é capaz de resistir estruturalmente à aplicação de uma determinada carga?
- Como pode o produto ser fabricado, uma vez conhecidos os processos de fabricação disponíveis?
- Podem ser automaticamente gerados programas para máquinas de comando numérico, robôs, máquinas de medição?

O modelo de produto adequado ao ambiente CIM deve manter a generalidade dos desenhos de engenharia, ou seja, um mesmo modelo deve atender aos requisitos impostos por diversas aplicações e ser capaz de modelar diferentes produtos. Métodos essencialmente idênticos devem ser utilizados para modelar automóveis, máquinas, móveis ou eletrodomésticos. Para que vários programas de aplicação possam acessar e modificar dados do modelo e utilizá-los sem a necessidade de interpretação ou edição por parte do usuário, o modelo do produto deve conter a semântica da representação, ou seja, deve ser um modelo de informação (ver Cap. 2). A busca da definição das informações que constituem um modelo adequado à completa caracterização de um produto e sua representação computacional, com o propósito de suprir as necessidades das aplicações CIM, será objeto das seções seguintes deste capítulo.

3.2. ABORDAGEM FORMAL PARA MODELAGEM DE PRODUTO

A fim de estabelecer de forma rigorosa propriedades de esquemas de representação, aborda-se o problema de modelagem em três níveis (Fig. 3.2) /MAN 88/, /REQ 80/:

- **Objetos Físicos:**

através de modelos deseja-se tratar aspectos do mundo real. Obviamente, a complexidade de um produto não permite que todos os seus aspectos sejam representados em um computador.

- **Objetos Matemáticos:**

para que se possa modelar produtos em um computador é necessário adotar uma adequada idealização destes. Esta deve ter uma conexão com o mundo real de uma ordem tal que possibilite extrapolar para os objetos físicos resultados obtidos para as idealizações. Ao mesmo tempo, a classe de objetos matemáticos utilizada deve ser suficientemente simples para que possa ser representada computacionalmente.

- **Representações:**

uma vez estabelecida a classe de objetos matemáticos de interesse, o último nível de modelagem consiste em encontrar uma forma de representar computacionalmente estes objetos.

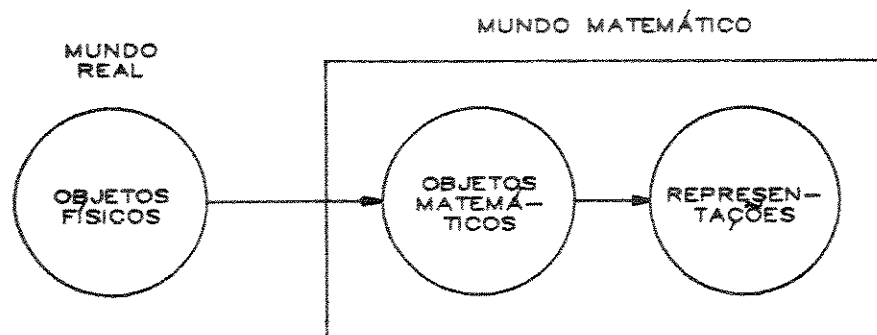


Fig. 3.2 - Visão de modelagem em três níveis

Representações sintaticamente corretas são estruturas finitas de símbolos de um alfabeto, construídas de acordo com regras sintáticas /REQ 80/. A coleção de todas as representações sintaticamente corretas é denominada espaço de representação (aqui denotado por R , Fig. 3.3). A semântica da representação é definida através da associação de entidades de um espaço de modelagem matemático (denotado por M com representações. Um esquema de representação é uma relação $S : M \rightarrow R$ e é caracterizado por um domínio D constituído pelos elementos de M que têm uma imagem na relação S e pelo conjunto-imagem V .

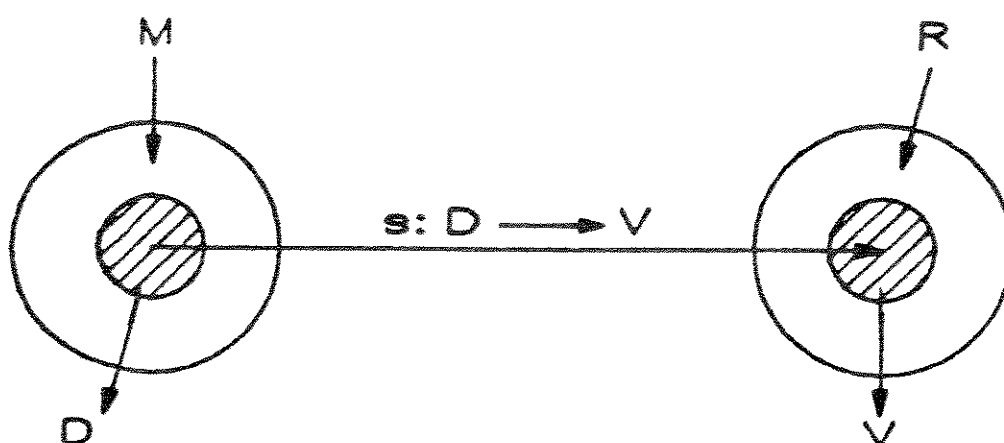


Fig. 3.3 - Definição de esquema de representação

Fazendo uso da definição acima, podem ser expressos rigorosamente validade, ambigüidade e unicidade de esquemas de representação:

- qualquer representação $r \in V$ é dita válida, uma vez que ela é sintática e semanticamente correta, ou seja, pertence a R e tem elementos correspondentes no domínio D ;
- uma representação é dita não-ambígua se ela modela um único objeto através da relação S , ou seja, ela é imagem de um único objeto de M . Formalmente, podemos dizer que a representação r é não-ambígua se

$$\langle m, r \rangle \in S \wedge \langle m', r \rangle \in S \Rightarrow m \equiv m'$$

onde m e m' são elementos do domínio D . Um esquema de representação é não-ambíguo se todas as suas representações válidas são não-ambíguas;

- uma representação é dita única se os objetos a ela associados através da relação S não admitem outra representação, ou, formalmente:

$$\langle m, r \rangle \in S \wedge \langle m, r' \rangle \in S \Rightarrow r \equiv r'$$

Um esquema de representação é único se todas as suas representações válidas são únicas.

3.2.1. Propriedades de Esquemas de Representação

A busca do espaço de modelagem, espaço de representação e do esquema de representação suportado por eles, de tal forma que sejam expressas as características relevantes de produtos, demanda propriedades formais para avaliação dos esquemas de representação. São elas:

- Domínio:

Descreve o poder de representação de um esquema, ou seja, o conjunto de objetos que ele é capaz de modelar. Logicamente, a gama de produtos que se pretende modelar ditará os requisitos que o espaço de modelagem deverá atender, conseqüentemente impactando a complexidade da representação. Além disso, deve ser considerado quão precisamente os produtos podem ser modelados.

- Validade:

A garantia de integridade do esquema de representação exige que todas as representações sintaticamente admissíveis sejam válidas, estando, portanto, associadas a algum objeto do domínio do espaço de modelagem. Um esquema com esta propriedade é denominado

sintaticamente válido, uma vez que ele garante a integridade da representação através de regras sintáticas.

- Abrangência:

É uma medida da capacidade do esquema de representação em responder a uma ampla gama de questões a respeito de produtos. Deve ser vista no contexto das aplicações que fazem uso do esquema de representação. Esquemas não-ambíguos contêm suficiente informação para distinguir um objeto de outro no espaço de modelagem e se constituem, portanto, no contexto do domínio do espaço de modelagem definido, em esquemas informativamente completos, atendendo, assim, aos requisitos das mais diversas aplicações.

- Unicidade:

Está relacionada ao problema de verificação da igualdade de objetos sendo modelados, uma vez conhecidas suas representações. Em esquemas de representação únicos esta verificação pode ser feita através de um algoritmo que compara as representações sintaticamente. Conforme será mostrado posteriormente, a questão da unicidade tem grande impacto sobre a complexidade das aplicações voltadas à manufatura de produtos.

Podem ainda ser citadas, por sua relevância em questões práticas, algumas propriedades informais de esquemas de representação:

- Concisão:

Refere-se à quantidade de espaço de armazenamento em computador requerida para representar um objeto em um dado esquema de representação, sendo, portanto, modelos concisos adequados ao armazenamento e transmissão de informações. Além disso, a busca de esquemas de representação concisos implicará, normalmente, em modelos com pouca redundância de dados, o que implica em maior facilidade de validação destes, pois menos restrições têm que ser

satisfeitas. Frequentemente a concisão é inversamente proporcional à precisão de representação. Outro ponto a ser considerado é que, muitas vezes, alguma forma de redundância é conveniente a fim de melhorar o desempenho computacional das aplicações.

- Facilidade de criação:

É uma medida de quão amigável ao usuário é o esquema em termos de criação de representações válidas. Normalmente, esta questão terá maior impacto sobre o projeto do modelador, uma vez que ao usuário devem, através de um subsistema de entrada, ser providas funções que omitam detalhes do esquema de representação. Dependendo deste, as funções terão que implementar a conversão entre o esquema apresentado ao usuário e aquele realmente utilizado.

- Eficácia:

A facilidade com que podem ser projetados algoritmos que visam responder a questões diversas a respeito de produtos é uma questão que deve ser cuidadosamente analisada quando se propõe um esquema de representação. Uma vez definidas as aplicações que farão uso do modelo de produto, deve ser verificada a complexidade computacional imposta por cada uma delas relativa ao esquema utilizado. Muitas vezes fez-se necessária a coexistência de múltiplos esquemas de representação devido à impossibilidade de um único esquema atender a todas as aplicações.

A presença de múltiplas representações introduz redundância de dados, fazendo-se necessária a garantia de consistência entre representações, de tal maneira que representações do mesmo objeto não sejam contraditórias. Formalmente (Fig. 3.4), duas representações r em R e r' em R' são consistentes se há pelo menos um objeto m de M tendo representações r e r' . Conforme a Fig. 3.4 indica, o objeto m pode ter várias representações em R e R' e outros objetos de M podem estar associados a r e r' , correspondendo, respectivamente, a representações que não são únicas e são ambíguas.

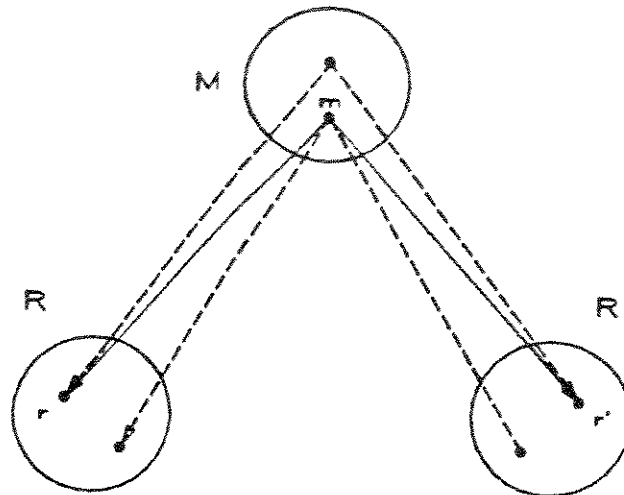


Fig. 3.4 - Consistência entre representações

Observando as propriedades formais definidas nesta seção no contexto das aplicações típicas encontradas em um ambiente CIM, podem ser definidos os seguintes requisitos para o modelo de produto adequado a este ambiente:

- deve ser completo, apresentando um domínio suficientemente amplo de tal forma que todos os produtos a serem manipulados possam ser representados;
- o modelador deve ser responsável pela validação dos modelos, uma vez que este será utilizado por programas de aplicação, tais como planejamento de processos generativo e geração automática de programas de comando numérico. Mesmo em aplicações que se caracterizam por um alto grau de interatividade, como a própria construção do modelo de produto, a complexidade deste torna inviável sua validação pelo usuário. Portanto, algoritmos devem ser implementados para verificar a validade da representação depois que esta é construída ou a validade deve ser forçada através de restrições às operações feitas sobre o modelo;

- deve ser não-ambíguo, a fim de fornecer informações suficientes às diversas aplicações em CIM;
- unicidade é desejável, principalmente no que diz respeito às aplicações de manufatura, em que padrões de representação podem ser utilizados para tomar decisões, como é o caso do reconhecimento de formas características (furos, chanfros) em um produto. Conforme será visto, unicidade só é encontrada em esquemas de representação em que o domínio é bastante limitado.

Uma vez estabelecidos os requisitos a serem satisfeitos, na próxima seção serão analisados modelos para representação de informações de produtos.

3.3. TIPOS DE INFORMAÇÃO E ESQUEMAS DE REPRESENTAÇÃO EM MODELO DE PRODUTO

A análise das funções envolvidas no ciclo de projeto e manufatura de produtos e da forma como são utilizados desenhos de engenharia mostra que as seguintes informações são necessárias à completa definição de produtos /AME 87/, /MAR 87/, /WOO 88/, /SHA 88/: geometria, topologia, tolerâncias, formas características, materiais e informações administrativas. A questão de representação de informações administrativas, tais como nome do produto, número, códigos de fornecedores de componentes, bem como de outras informações de suporte ao produto, tais como roteiro de fabricação, programas de comando numérico, plano de controle de qualidade, apesar da relevância, não será aqui tratada, concentrando-se a análise nos demais tipos de informação.

3.3.1. Geometria e Topologia

A questão da representação da geometria, o ramo da matemática que lida com formas e relações espaciais, é central para o projeto e manufatura de produtos, uma vez que esta

constitui a principal fonte de informação a respeito destes /REQ 82/. Inicialmente, artesãos utilizavam modelos físicos para transportar informação geométrica. Com o advento da produção em massa e conseqüente especialização do trabalho, passou-se a utilizar desenhos de engenharia para representar esta informação. Com o desenvolvimento de periféricos gráficos para computadores, viabilizaram-se sistemas computacionais para o tratamento de informações geométricas. O contínuo desenvolvimento de técnicas voltadas à computação gráfica e sistemas CAD/CAM (Projeto Auxiliado por Computador / Manufatura Auxiliada por Computador) tem sido fator preponderante para a evolução dos métodos de modelagem geométrico. A seguir serão apresentados esquemas de representação de informação geométrica, sendo analisados do ponto de vista dos requisitos estabelecidos na seção no anterior. Conforme será visto, estes diferem basicamente quanto à extensão com que atendem aqueles requisitos, o que refletirá diretamente sobre a sua aplicabilidade ao ambiente CIM.

3.3.1.1. Modelos Gráficos

Um modelo gráfico, também conhecido como modelo de arame, é uma coleção de segmentos de curvas e linhas que representam as arestas e vértices de um objeto sendo modelado /PET 86/.

Os primeiros sistemas CAD foram concebidos como uma ferramenta de auxílio à geração de desenhos de engenharia, visando a melhoria da qualidade destes e um aumento de produtividade, principalmente no que se refere a alteração de desenhos, uma vez que outro desenho pode ser tomado como ponto de partida. Portanto, o método utilizado nestes sistemas para representar informação geométrica de um produto é uma simples transposição do método convencional. Nestes sistemas, mudanças feitas em uma vista de um objeto não são propagadas automaticamente para as outras vistas porque o sistema desconhece que as várias vistas são projeções de um único objeto. Sendo assim, como mostra a Fig. 3.5, é perfeitamente possível criar

representações sintaticamente válidas, ou seja, que são sequências de segmentos de curvas ou linhas, mas que, claramente, correspondem a objetos irrealizáveis fisicamente, ficando totalmente a cargo do usuário a validação do modelo.

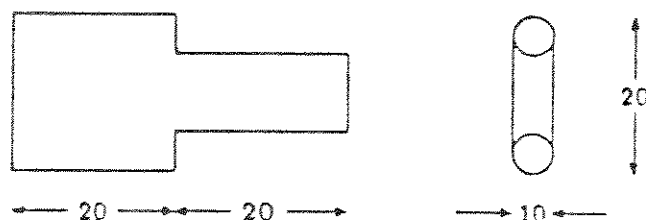


Fig. 3.5 - Desenho fisicamente irrealizável

A etapa seguinte consistiu no desenvolvimento de sistemas que armazenam uma única representação tridimensional de um objeto, sendo utilizadas como primitivas segmentos de curvas no espaço tridimensional. Desta forma, todas as vistas bidimensionais, sejam projeções paralelas ou perspectivas, podem ser geradas por meio de algoritmos facilmente implementáveis. Porém, como estes modelos só representam informações a respeito de arestas e vértices, o comportamento do objeto ao longo de suas faces é desconhecido, impondo restrições à sua utilização como fonte de informação para aplicações. Como mostra a Fig. 3.6, modelos de arame são ambíguos. No caso, por exemplo, de uma aplicação voltada ao cálculo do volume do objeto é imprevisível o resultado a ser alcançado. Além disso, como pode ser observado na Fig. 3.7, podem ser construídos modelos de objetos não realizáveis fisicamente. No caso de objetos com superfícies curvas surge ainda outro problema relativo à definição das arestas que compõem o objeto, uma vez que algumas delas são usadas como artifício para visualização, sendo dependentes da posição de observação deste. Tal situação é mostrada na Fig. 3.8. Todas estas questões estão ligadas ao fato de que modelos de arame são voltados à descrição do desenho de um objeto e, não, do objeto.

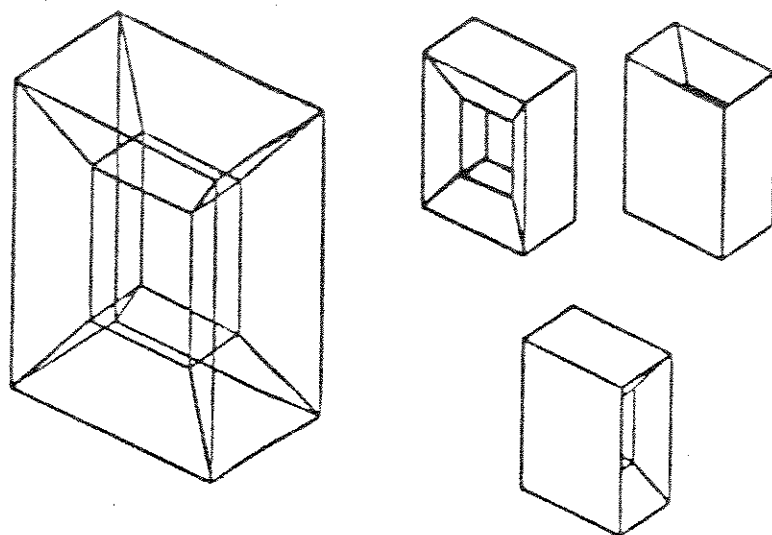


Fig. 3.6 - Modelo de arame ambíguo

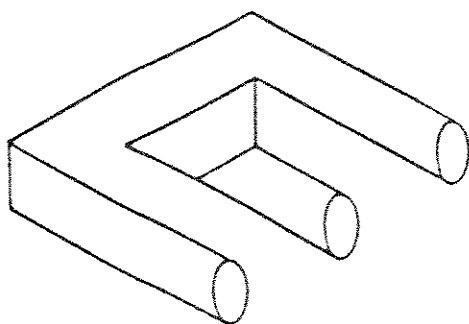


Fig. 3.7 - Modelo de arame de objeto não realizável fisicamente

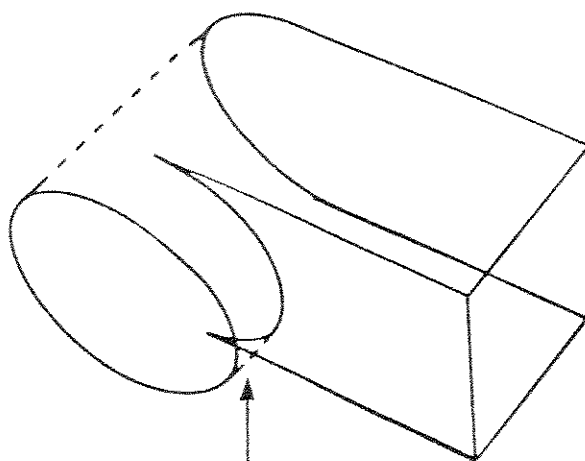


Fig. 3.8 - Modelo de arame de objeto com superfície curva

Logicamente as deficiências apontadas acima afetarão diretamente a gama de aplicações que podem fazer uso de modelos de arame Borgerson /BOR 80/ mostra os problemas relativos à utilização de modelos de arame em diversas aplicações automatizadas voltadas a projeto, análise, documentação e manufatura de produtos. Porém, há aplicações em que modelos de arame são adequados, como é o caso de problemas eminentemente bi-dimensionais (por exemplo, geração de programas de comando numérico para tornos) e aqueles em que se está interessado somente no envelope espacial ocupado por um objeto (por exemplo, uma primeira abordagem para verificação de interferência entre objetos).

Têm sido realizados trabalhos que visam a construção de modelos não-ambíguos a partir de modelos de arame. Do exposto anteriormente infere-se que este problema não tem solução única, sendo os algoritmos voltados à enumeração de todos os possíveis modelos obtidos a partir de um dado modelo de arame, ficando a escolha a cargo do usuário /MAR 80/. Outro algoritmo /HOJ 88/ vai além, eliminando soluções que contenham inconsistências, através da verificação de conflitos de caráter geométrico e topológico, utilizando para isto heurísticas.

Uma análise dos requisitos impostos nos modelos de produtos adequados ao ambiente CIM mostra que modelos de arame não os atendem. Estes são ambíguos, não-únicos e não há mecanismo algum para validação automática dos modelos. Quanto ao domínio não se verificam restrições, desde que se disponha de métodos adequados para representação de curvas. Estes serão analisados na próxima seção, juntamente com a representação de superfícies.

3.3.1.2. Modelos de Superfície

O desenvolvimento de técnicas que visam incorporar informações sobre superfícies a modelos de arame foi grandemente impulsionado pelas necessidades das indústrias automobilística, aeroespacial e naval, especialmente com o intuito de substituir

técnicas convencionais de fabricação de ferramental utilizado na manufatura de componentes que contém superfícies de dupla curvatura, tais como asas de aeronaves e tetos de automóveis. Uma vez definidas matematicamente as superfícies, podem ser gerados programas de comando numérico através de programa de aplicação implementado com esta finalidade, não se fazendo mais necessários os modelos de resina até então utilizados /BER 85/.

Um modelo de superfície geralmente começa com um modelo de arame cujas arestas são utilizadas para definir superfícies. Entre os tipos de superfícies suportados por sistemas CAD/CAM encontram-se:

- planos;
- superfícies quadráticas (esfera, cilindro, elipsóides, cones, parabolóides e hiperbolóides). Podem ser definidas analiticamente por meio da equação geral quadrática /FAU 79/;
- superfícies regradas, em que duas curvas são conectadas por linhas retas (Fig. 3.9);
- cilindro tabulado: superfície obtida movendo-se uma linha reta ao longo de uma curva plana de modo que a linha se mantenha paralela a uma outra linha de referência (Fig. 3.10);

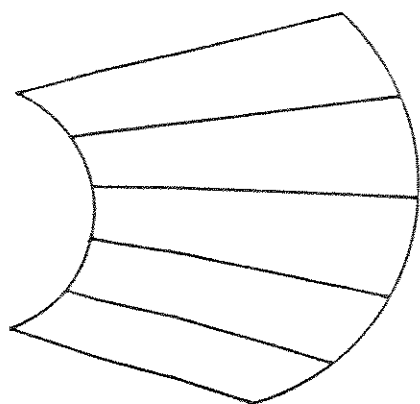


Fig. 3.9 - Superfície regrada

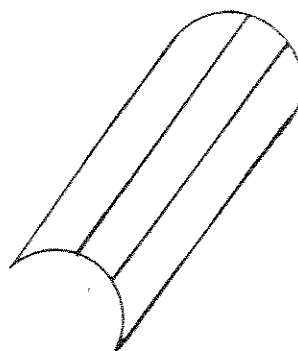


Fig. 3.10 - Cilindro Tabulado

- superfícies de revolução: definidas pela rotação de uma curva plana (aberta ou fechada) em torno de um eixo;
- superfícies de dupla curvatura (ver 3.3.1.2.1).

No caso de objetos em que todas as faces são planas, a representação explícita de informações sobre os planos, retas e pontos que os definem é redundante, uma vez que conhecida a representação geométrica de um deles, os outros dois podem ser obtidos. Esta situação é mostrada esquematicamente na Fig. 3.11, onde se observa que um ponto é a interseção de duas retas coplanares ou três planos, uma reta é a interseção de dois planos ou liga dois pontos e um plano é definido por duas retas não reversas ou três pontos. Sendo assim, neste caso, a questão de utilizar modelos de superfícies ou não é ditada pela conveniência frente às aplicações que farão uso do modelo. Porém, qualquer representação puramente geométrica de um objeto só garante que o esquema de representação será não-ambíguo se o domínio for restrito a objetos convexos /BAE 79/. Os sete planos da Fig. 3.12 podem definir qualquer dos dois objetos mostrados.

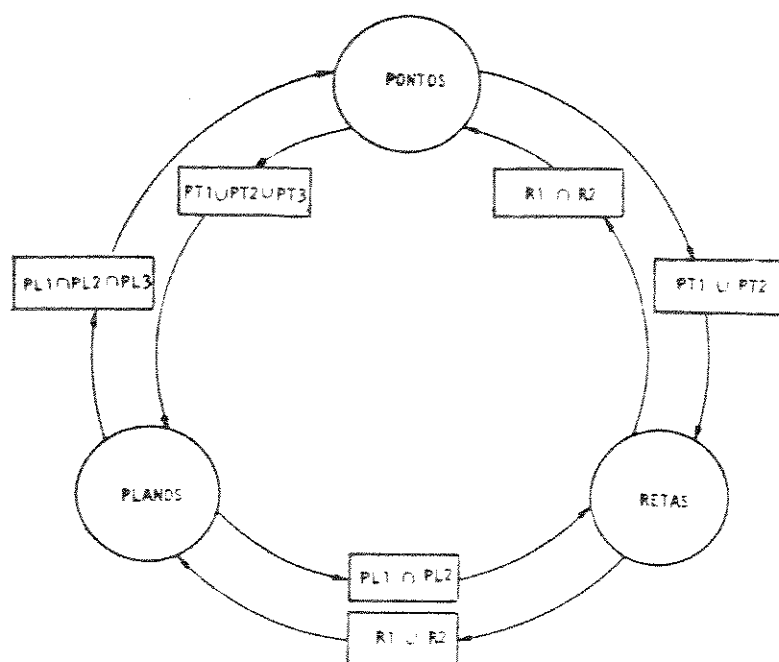


Fig. 3.11 - Mapeamento entre representações geométricas

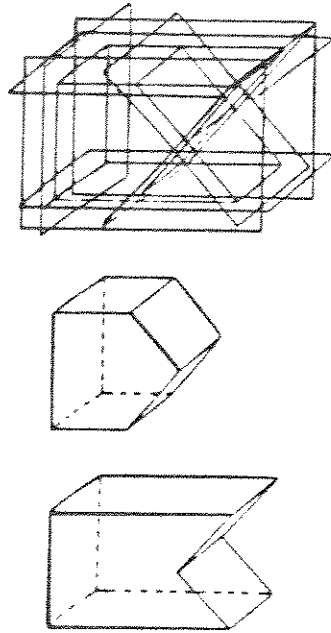


Fig. 3.12 - Modelo de superfície ambíguo

Pode-se concluir, então, que a representação de um objeto como um conjunto não estruturado de superfícies, ou seja, um modelo de superfície, não atende a um dos requisitos básicos impostos ao modelo de produto que se busca. Conforme será visto adiante o requisito de não-ambigüidade exigirá que, além dos elementos geométricos, um esquema de representação contenha os relacionamentos de adjacência entre estes, ou seja, sua estrutura topológica. Além disso, não são únicos pois a superfície total de um dado objeto pode ser dividida em superfícies elementares de várias maneiras. Outra questão diz respeito à validação de uma representação. Como no caso de modelos de arame, representações sintaticamente corretas podem corresponder a objetos sem significado físico, como é o caso da Fig. 3.13 em que as superfícies se interpenetram de tal maneira que tornam sua interpretação impossível, como um objeto realizável fisicamente. Embora seja possível projetar algoritmos que verifiquem a integridade geométrica e topológica de modelos de superfície, a ausência de informação de adjacência entre os elementos geométricos faz com que o esforço computacional exigido torne

esta verificação impraticável. Quanto ao domínio, este é limitado pela gama de superfícies que o sistema de modelagem suporta. A próxima seção apresentará métodos para a representação de superfícies de dupla curvatura.

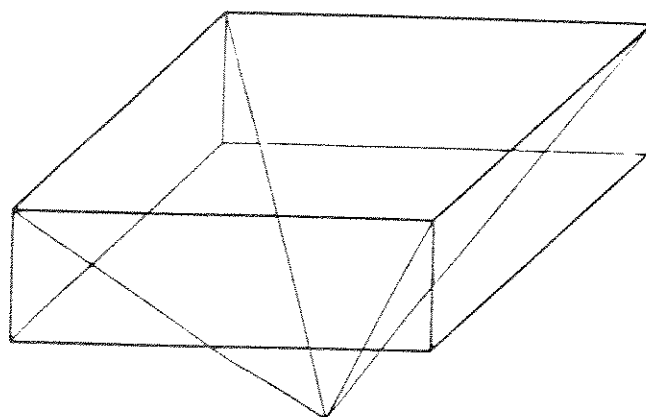


Fig. 3.13 - Modelo de superfície fisicamente irrealizável

3.3.1.2.1. Representação de Curvas e Superfícies

Além das áreas já citadas (indústria automobilística, aeroespacial e naval), a questão da representação de superfícies de dupla curvatura, também chamadas superfícies de forma livre, é central para o projeto de peças fundidas, forjadas e de moldes de injeção de peças plásticas. Logicamente, a necessidade de uma representação precisa de uma superfície também é ditada pela aplicação que a utilizará. Para geração de uma malha para análise por elementos finitos, uma aproximação da superfície por faces planas é suficiente. Uma vez que normalmente os métodos de representação de superfícies são extensões daqueles utilizados para representar curvas não definidas analiticamente, estes serão inicialmente abordados. A presente descrição se concentrará nos aspectos qualitativos das representações.

Uma primeira abordagem para o problema de ajustar uma curva passando por $N+1$ pontos é a interpolação por polinômios de

Lagrange, resultando em um polinômio de grau N . Porém, como o grau do polinômio interpolante está acoplado ao número de pontos, à medida que este aumenta surge a tendência de haver oscilações indesejáveis na curva, uma vez que existem $N-1$ possíveis máximos ou mínimos na curva, pois este é o grau do polinômio correspondente à primeira derivada /BOH 84/. Uma possível solução para este problema é interpolar polinômios de menor grau entre cada dois pontos da curva. Neste caso, surge outro problema que é a descontinuidade da primeira derivada, ou seja, tangência, o que é uma característica indesejável em curvas que representam elementos de projeto mecânico, pois estas descontinuidades induzem possíveis pontos de falha mecânica. A interpolação por polinômios de Hermite garante a continuidade de tangência utilizando polinômios de grau baixo. No caso de polinômios cúbicos, para cada ponto impõe-se a posição e o valor da primeira derivada. Portanto, para cada segmento da curva os quatro coeficientes do polinômio de terceiro grau são definidos por dois valores de posição e dois valores de derivada. O problema de ordem prática com este método é que normalmente não se dispõe de valores de primeira derivada nos pontos /FAU 79/.

A utilização de splines constitui um método de interpolação que supera as deficiências apontadas acima. Dado um conjunto de pontos para o qual se deseja ajustar uma curva e uma vez definido o grau K dos polinômios a serem interpolados, uma spline é definida como uma sucessão de polinômios de grau K , onde cada polinômio está definido entre dois pontos do conjunto dado. Além disso, impõe-se que a curva apresente continuidade das $K-1$ primeiras derivadas em toda a sua extensão /BAR 84/. A Fig. 3.14 mostra o caso de uma spline cúbica ajustada a cinco pontos coplanares. Entre cada par $[x_i, x_{i+1}]$ está definido um polinômio de grau três. A principal vantagem de splines é que precisam ser conhecidas apenas as posições dos pontos e ao mesmo tempo a continuidade das derivadas é garantida.

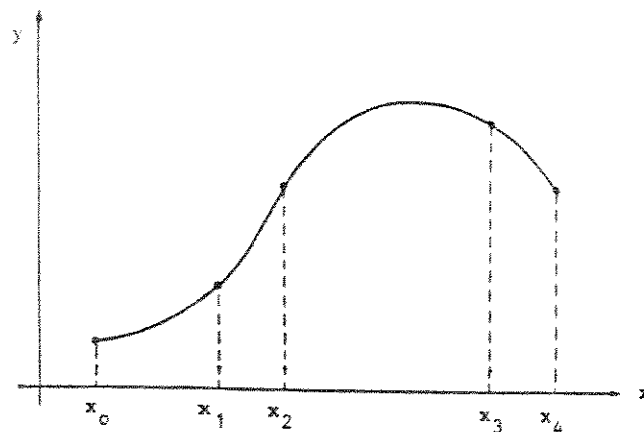


Fig. 3.14 - Interpolação por spline cúbica

Técnicas de interpolação apresentam algumas desvantagens quando considerado o problema do projeto de curvas em uma aplicação CAD, onde ao usuário não deve ser requerido conhecimento matemático. A dificuldade em controlar, de maneira intuitiva, a mudança de forma causada por uma alteração de um dos pontos interpolados de uma spline pode conduzir a perturbações indesejáveis, tanto na vizinhança do ponto quanto remotamente /MOR 85/. Visando definir curvas em um formato matemático mais adequado à utilização por projetistas, Bezier /BEG 74/, /FAU 79/, /MOR 85/ estabeleceu que cada ponto da curva deve ser escrito em função de um parâmetro u como:

$$\vec{P}(u) = \sum_{i=0}^n \vec{P}_i B_{i,n}(u), \quad u \in [0,1]$$

com

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}$$

Os vetores $\vec{P}_i$, chamados pontos de controle, representam os $n+1$ vértices do polígono característico. Este apresenta as seguintes propriedades:

- a curva passa pelo primeiro e último ponto do polígono;

- a tangente em $\vec{P}_0$ é dada por $\vec{P}_1 - \vec{P}_0$ e a tangente em $\vec{P}_n$ por $\vec{P}_n - \vec{P}_{n-1}$;
- a curva está contida no envelope definido pelo polígono.

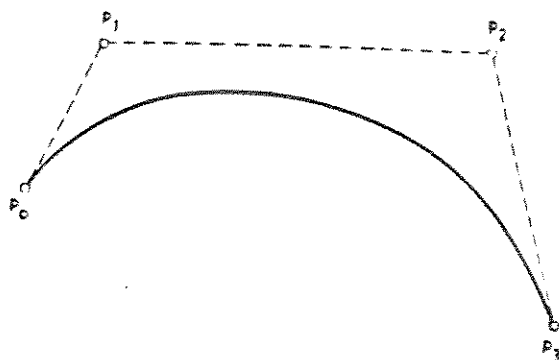


Fig. 3.15 - Curva de Bezier

Uma vez que o polígono característico aproxima a forma da curva, este pode ser tomado como ponto de partida, ajustando-se a curva através de alterações dos pontos de controle. Como pode ser observado nas equações anteriores, a curva é representada por um somatório de polinômios de grau máximo n . Cada polinômio tem valor diferente de zero sobre todo o domínio, ou seja, para $0 \leq u \leq 1$, o que significa que, alterando-se um ponto de controle, toda a curva será alterada. Além disso, o grau dos polinômios está acoplado ao número de pontos de controle, causando os problemas já citados quando este número aumenta.

B-spline /GOR 74/, /FAU 79/, /MOR 85/, /ROG 76/ é uma forma de representação que não apresenta as restrições acima expostas. Em B-splines temos:

$$\vec{P}(u) = \sum_{i=0}^n \vec{P}_i N_{i,k}(u)$$

onde k é a ordem dos polinômios a serem interpolados. As funções $N_{i,k}(u)$ são definidas pela seguinte recorrência:

$$N_{i,k}(u) = \frac{(u-x_i) N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u) N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}}$$

e

$$N_{i,1} = 1 \quad \text{se} \quad x_i \leq u < x_{i+1};$$

$$N_{i,1} = 0 \quad \text{caso contrário}$$

onde os x 's, chamados nós, relacionam a variável paramétrica u aos pontos de controle $\vec{P}_i$. Estas funções têm as seguintes propriedades:

- como mostra a Fig. 3.16, elas só apresentam valores não nulos no intervalo $[x_{i-k}, x_i]$. Nesta figura está representada a função $N_{i,4}(u)$, ou seja, de grau 3;
- pode-se provar que qualquer spline de ordem menor ou igual a k pode ser escrita como uma combinação linear destas funções [BAR 84]. Por isso, as funções $N_{i,k}(u)$ são chamadas splines básicas.

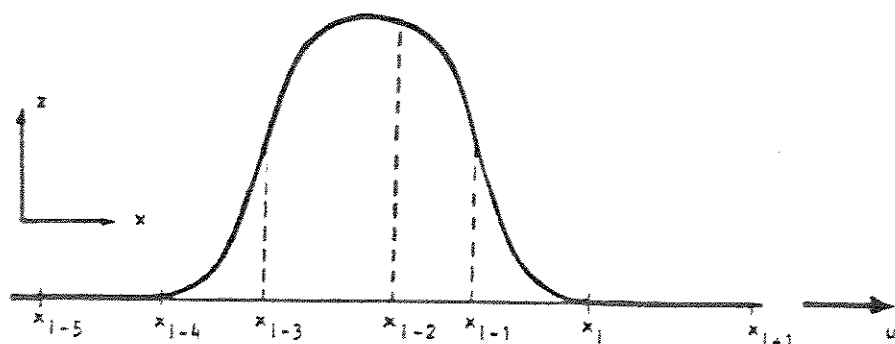


Fig. 3.16 - Spline básica cúbica

Destas propriedades decorre o fato de, em representação por B-splines, ter-se controle local, ou seja, a alteração de um ponto de controle surtirá efeito sobre uma região limitada da curva. Além disso, o aumento do número de pontos de controle não implica em aumento da ordem do polinômio interpolante. Outra característica importante é que, como foi descrito para splines, garante-se continuidade das $k-2$ primeiras derivadas. Finalmente, todas as propriedades citadas para curvas de Bezier são válidas para curvas B-spline, pois estas são uma generalização daquelas /FAU 79/.

Uma generalização de B-spline, chamada B-spline racional /PIE 87/, /TIL 83/, em que são atribuídos pesos aos pontos de controle, apresenta a importante propriedade que qualquer curva cônica pode ser representada exatamente /KNA 84/. Como todas as propriedades de B-splines são mantidas, em um modelador baseado em B-splines racionais podem ser representadas retas, cônicas e curvas não definidas analiticamente.

Os métodos de representação de superfícies são extensões dos aqui apresentados para curvas, sendo as superfícies definidas através do produto cartesiano de duas direções paramétricas. A Fig. 3.17 mostra uma superfície de Bezier e o poliedro formado por seus pontos de controle.

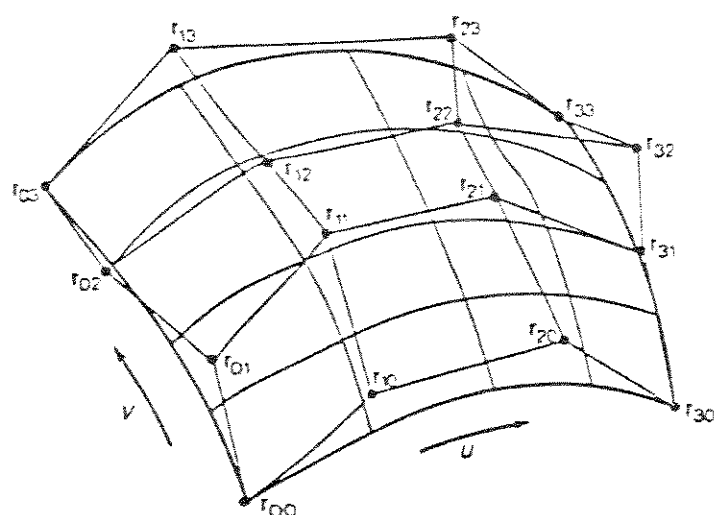


Fig. 3.17- Superfície de Bezier

A Fig. 3.18 apresenta uma esfera e o poliedro de controle correspondente em uma representação por B-splines racionais. Da mesma forma, no método de representação por elementos de superfície de Coons /COO 67/ se verificam restrições análogas àquelas citadas para polinômios de Hermite.

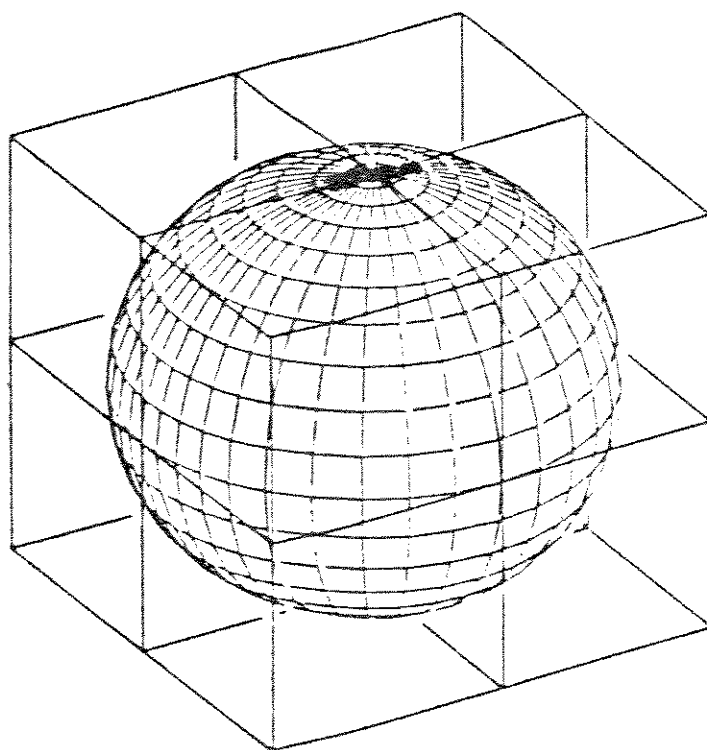


Fig. 3.18- Esfera representada por B-splines racionais

3.3.1.3. Modelos Sólidos

Modelar sólidos consiste em representar um conjunto infinito de pontos em um espaço de armazenamento finito de um computador. Como foi visto, o ambiente CIM requer modelos completos, não-ambíguos e cuja integridade seja garantida automaticamente. Para tanto, deve-se caracterizar rigorosamente a classe de objetos matemáticos (ver Fig. 3.2) que corresponde a sólidos fisicamente realizáveis. Isto pode ser feito através de duas abordagens /MAN 88/:

- considerando um sólido como um subconjunto do espaço Euclidiano tridimensional E^3 e usando conceitos de topologia de conjunto de pontos para caracterizar as propriedades deste;
- representando um sólido através da superfície que o limita e usando conceitos de topologia algébrica para caracterizar suas propriedades.

A fim de suportar o desenvolvimento posterior, serão apresentados a seguir conceitos básicos de topologia. Seja E^N um espaço Euclidiano N-dimensional. Pontos de E^N são denotados por X, Y e conjuntos de pontos por $A, B, \dots$. A distância entre dois pontos é dada por $d(X, Y)$. Sendo assim, temos as seguintes definições:

DEFINIÇÃO 3.1: A vizinhança V de um ponto X , $N(X, V)$ é definida como o conjunto de pontos Y tais que $d(X, Y) < V$. Portanto, em E^2 as vizinhanças são discos abertos e em E^3 esferas abertas.

DEFINIÇÃO 3.2: Um ponto X é dito estar próximo do conjunto A se qualquer vizinhança de X contém um ponto de A .

DEFINIÇÃO 3.3: Uma transformação contínua de um conjunto D para um conjunto R é uma função f com domínio D e imagem R tal que para qualquer ponto $X \in D$ e conjunto $A \subset D$, se X é próximo de A , então $f(X)$ é próximo do conjunto $f(A) = \{f(Y) \mid Y \in A\}$. Informalmente, estas transformações correspondem a deformações que podem ser feitas em um objeto de material flexível, como comprimir, estender, sem que haja porém ruptura do material. Nestas transformações a geometria do sólido pode mudar, mas a proximidade entre os pontos que o compõem é mantida.

DEFINIÇÃO 3.4: Uma transformação topológica é uma transformação contínua que tem uma inversa contínua. Dois conjuntos são ditos topologicamente equivalentes se existe uma transformação

topológica entre eles. Neste ponto, podemos definir topologia como o estudo de propriedades invariantes sob transformações topológicas.

DEFINIÇÃO 3.5: Um conjunto é fechado se ele contém todos os seus pontos próximos. O fechamento de um conjunto A , denotado por $c(A)$ é a união de A e dos pontos próximos a A .

DEFINIÇÃO 3.6: Um conjunto é limitado se ele está contido em uma vizinhança. Conjuntos fechados e limitados são ditos compactos.

DEFINIÇÃO 3.7: O interior de um conjunto A , denotado por $i(A)$, é o conjunto de pontos pertencentes a A e que não são próximos do complemento de A .

DEFINIÇÃO 3.8: A fronteira de um conjunto A , denotada por $b(A)$, é o conjunto de pontos que são próximos a A e ao complemento de A .

A partir destes conceitos podem ser estabelecidas as propriedades dos sólidos de interesse em modelos de produto:

- o sólido deve ser um subconjunto fechado e limitado do espaço Euclidiano E^3 ;
- o sólido deve ser rígido, ou seja, deve permanecer invariante em transformações rígidas, tais como translação e rotação;
- o sólido deve ser tridimensionalmente homogêneo. Esta condição implica que sólidos não podem ter pontos, linhas ou superfícies isoladas. Formalmente, podemos estabelecer que a regularização de um conjunto de pontos A é definida por $r(A) = c(i(A))$, onde fechamento e interior foram definidos anteriormente. Conjuntos que satisfazem $r(A) = A$ são ditos regulares. A Fig. 3.19 apresenta um conjunto não regular;

- um sólido deve ser finitamente descritível, ou seja, deve haver algum aspecto do modelo do sólido que seja finito, a fim de que este possa ser representado em computador. Esta condição exclui sólidos cuja superfície possa ser representada por funções que não admitam limite em um ponto.

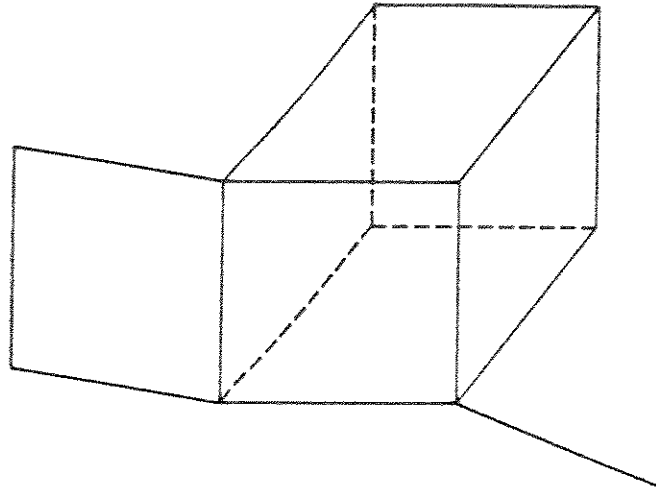


Fig. 3.19 - Sólido não regular

Conjuntos que atendem estes requisitos, chamados r-con, constituem o espaço de modelagem dos esquemas de representação de sólidos segundo a primeira abordagem citada.

A segunda abordagem visa caracterizar o sólido fisicamente realizável através de propriedades da superfície que o limita. Para isto, esta é representada por uma união de faces, limitadas por arestas que são limitadas por vértices. A estes elementos topológicos são associadas informações geométricas. A correspondência entre elementos topológicos e geométricos é sintetizada na Tabela 3.1. A descrição topológica do sólido se dá através dos relacionamentos de adjacência entre suas faces, arestas e vértices /MAL 85/. No contexto de representação de um sólido por sua fronteira, a topologia se refere a estes relacionamentos. No caso, por exemplo, de um cubo, a geometria se refere ao posicionamento dos planos que o formam e suas

interseções, enquanto a topologia define que cada uma de suas seis faces contém quatro arestas e vértices. De modo geral, um cubo define um particionamento particular de uma superfície de um sólido topologicamente equivalente a uma esfera, enquanto sua geometria define uma distorção particular desta, uma vez que, com a mesma topologia, poder-se-ia ter, por exemplo, um cubóide de faces curvas. Uma vez feitas estas considerações iniciais, busca-se a seguir a caracterização das superfícies que correspondem a sólidos fisicamente realizáveis.

TOPOLOGIA	GEOMETRIA
VERTICE	PONTO
ARESTA	CURVA
FACE	SUPERFICIE

TABELA 3.1 - Correspondência entre elementos topológicos e geométricos

Uma superfície é um subconjunto de E^3 que é essencialmente bidimensional, pois cada ponto desta enxerga a região adjacente como se fosse bidimensional. Esta bidimensionalidade inerente de superfícies significa que as propriedades destas podem ser analisadas através de modelos bidimensionais.

Uma noção intuitiva de superfícies de sólidos é que elas são fechadas. Formalmente, podemos definir: uma variedade topológica bidimensional M é um conjunto onde cada ponto tem uma vizinhança topologicamente equivalente a um disco aberto de E^2 . Por outro lado, os sólidos de interesse foram caracterizados como r -con's. Dado um r -con A , se uma variedade M e a fronteira de A , $b(A)$, são topologicamente equivalentes, diz-se que A é uma realização de M em E^3 [MAN 88]. As variedades realizáveis definem a classe de objetos matemáticos para a abordagem por fronteira. Porém, este espaço de modelagem não corresponde a

todos os sólidos fisicamente realizáveis, uma vez que nem todos os r-con's são realizações de alguma variedade topológica. A Fig. 3.20 apresenta exemplos de sólidos cuja superfície não é uma variedade topológica. No caso da figura (a), por exemplo, a vizinhança do ponto de contato entre os dois tetraedros consiste em dois discos, portanto não satisfazendo a definição de variedade topológica bidimensional. Casos como estes são tratados considerando-se o sólido como união de outros cujas superfícies são variedades topológicas.

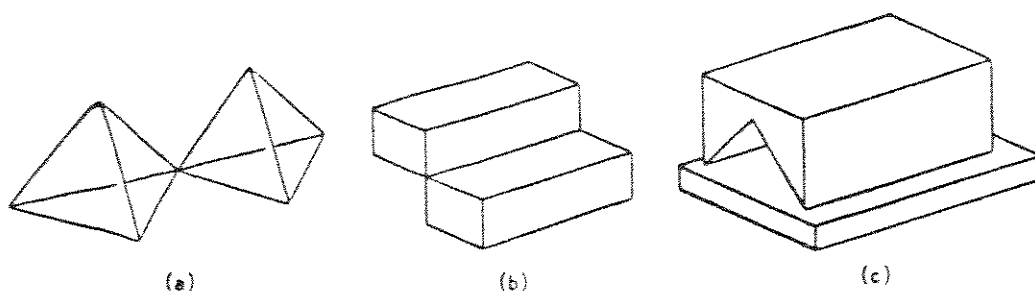


Fig. 3.20 - Sólidos cujas superfícies não são variedades topológicas

Conforme foi observado anteriormente, superfícies podem ser representadas por modelos bidimensionais que correspondem a variedades topológicas. Estes modelos devem ser definidos de tal forma que se possa reconhecer, através deles, quais são as variedades topológicas bidimensionais realizáveis. Com esta finalidade são aqui introduzidos modelos planos.

A Fig. 3.21 apresenta o modelo plano de um cilindro. Neste modelo a superfície cilíndrica é obtida pela junção das arestas rotuladas da figura (a). A fim de que os mesmos conceitos topológicos se apliquem a todos os pontos do modelo, definem-se vizinhanças, de forma diferente da apresentada anteriormente, para os pontos das arestas rotuladas. Para estes, a vizinhança é dada pela união de dois semicírculos do mesmo raio, cujos centros são o ponto e seu simétrico, na outra aresta rotulada. Sendo assim, pontos simétricos em arestas rotuladas têm a mesma

vizinhança (ver Fig. 3.21), permitindo, portanto, que sejam tratados como um único ponto. As definições a seguir formalizarão modelos planos.

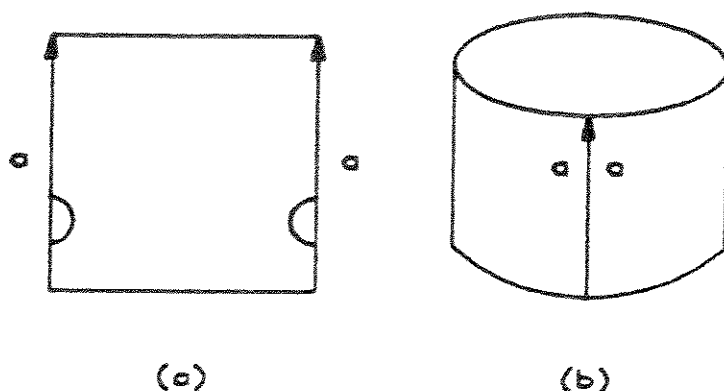


Fig. 3.21 - Modelo plano de um cilindro

DEFINIÇÃO 3.9: Sejam F um conjunto de faces e $a_1, a_2, \dots$ arestas destas faces. Estas arestas são ditas identificadas através dos seguintes procedimentos:

- a cada aresta é associada uma orientação de um vértice a outro, sendo a aresta topologicamente equivalente ao intervalo unitário, com o vértice inicial correspondendo ao valor 0 e o final ao valor 1;
- os pontos sobre as arestas que correspondam ao mesmo valor no intervalo unitário são tratados como um único ponto;
- vizinhanças são definidas como os círculos inteiramente contidos em uma única face e para os pontos sobre as arestas pela união de semicírculos de mesmo raio.

Através desta definição, arestas identificadas podem ser tratadas como se fossem uma única aresta. Na Fig. 3.21 as arestas rotuladas representam identificação.

DEFINIÇÃO 3.10: Sejam F um conjunto de faces e $v_1, v_2, \dots$ vértices destas faces. Estes vértices são ditos identificados através dos seguintes procedimentos:

- os vértices são tratados como um único ponto;
- vizinhanças são definidas como os círculos inteiramente contidos em uma única face e no caso de vértices como a união de setores circulares de mesmo raio com centros nos vértices.

Se os vértices $v_1, v_2, \dots$ são identificados, as respectivas faces $f_1, f_2, \dots$ são ditas identificadas por esta coleção de vértices.

DEFINIÇÃO 3.11: Um modelo plano é um grafo planar dirigido (V, A, F) com um número finito de vértices $V = \{v_1, v_2, \dots\}$, arestas $A = \{a_1, a_2, \dots\}$ e faces $F = \{f_1, f_2, \dots\}$ limitadas por arestas e vértices. Cada face do grafo tem uma orientação definida em torno de suas arestas. Arestas e vértices do grafo são rotulados. Se coleções de arestas e vértices têm o mesmo rótulo, são consideradas identificadas de acordo com as definições acima.

O modelo plano de uma pirâmide é mostrado na Fig. 3.22. A figura à direita apresenta sua forma condensada, em que as identificações implicitamente definidas são suprimidas. Neste modelo observa-se, por exemplo, que as vizinhanças de pontos

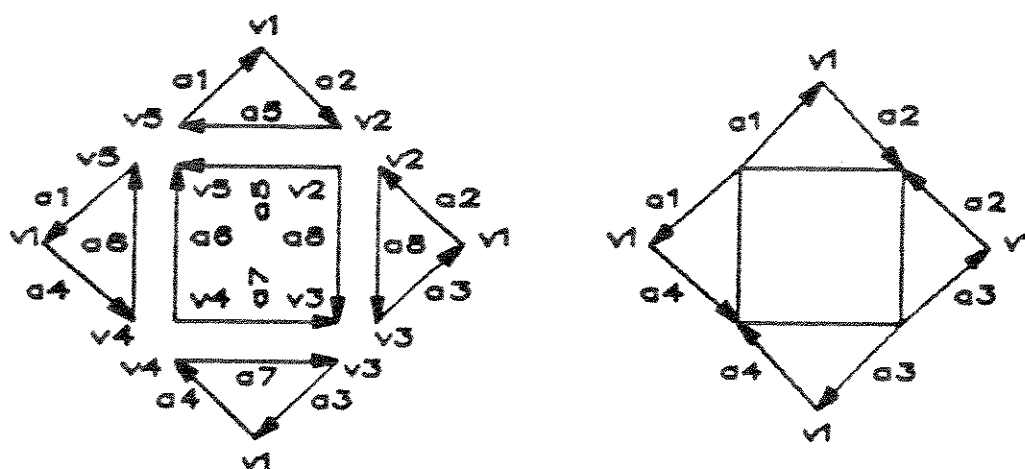


Fig. 3.22 - Modelo plano de uma pirâmide

sobre a aresta a_1 consistem em dois semicírculos e a vizinhança do vértice v_1 consiste em quatro setores circulares.

A limitação a grafos planares, em que as arestas não se cruzam e não ser em vértices, se deve ao fato que modelos planos devem ser mapeáveis em superfícies /SZW 84/.

Uma vez definidos modelos planos, podem ser estabelecidas condições para que estes representem superfícies bem formadas, que correspondam a sólidos fisicamente realizáveis. São elas /BAE 79/, /EAS 79/:

- A superfície deve ser fechada.

Superfícies abertas como aquela da Fig. 3.23 podem ser modeladas definindo-se faces virtuais, embora isto gere problemas, pois as aplicações que usam o modelo teriam que tratar estas faces de forma diferenciada. Em termos de modelos planos, esta condição significa que cada aresta deve estar identificada com uma única outra aresta. Além disso, para cada coleção de vértices identificados, as faces identificadas por esta coleção devem formar um ciclo tal que cada par consecutivo de faces no ciclo compartilhe uma aresta adjacente a um vértice da coleção. Em outras palavras, a vizinhança de uma coleção de vértices identificados deve ser um círculo. Observando-se a Fig. 3.22, verifica-se que esta condição é obedecida no vértice v_1 , pois sua vizinhança é formada por quatro setores circulares, sendo a vizinhança da coleção identificada circular.

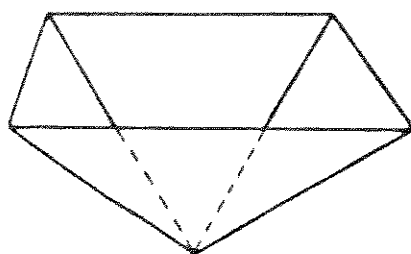


Fig. 3.23 - Superfície aberta

- A superfície deve ser orientada.

Uma superfície orientada é aquela em que podem ser distinguidos dois lados: um exterior e outro interior. A garrafa de Klein mostrada na Fig. 3.24, junto com seu modelo plano, é um exemplo de superfície não orientada. Embora seja fechada, esta superfície não é realizável fisicamente pois intercepta a si mesma. A verificação desta condição em modelos planos é feita através da regra de Moebius. Um modelo plano é orientado se as direções das faces podem ser escolhidas de tal modo que, para cada par de arestas identificadas, uma aresta ocorra em sua orientação positiva na direção escolhida para sua face, e a outra em sua orientação negativa. Cabe lembrar que a orientação de uma aresta

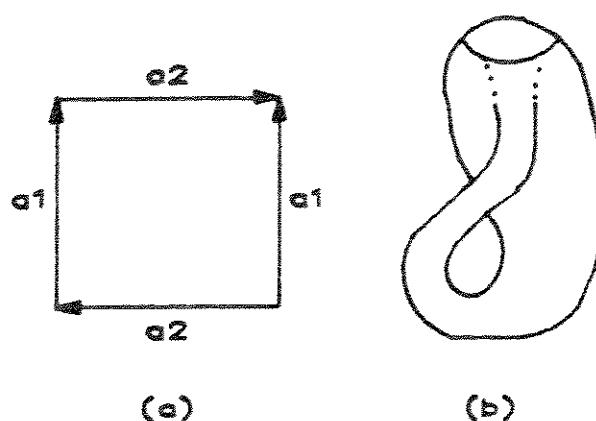


Fig. 3.24 - Superfície não orientada

é dada pela sequência de seus dois vértices. A Fig. 3.25 mostra um modelo plano em que todas as faces, exceto a externa, são orientadas no sentido horário. Como se observa, cada aresta ocorre uma vez em sua orientação positiva e outra na negativa, tratando-se, portanto, de um modelo plano orientado.

As duas condições acima garantem a integridade topológica de uma superfície. Porém, mesmo modelos em que as condições topológicas são satisfeitas podem não ser realizáveis fisicamente. Devido a isso, às condições anteriores deve ser adicionado um requisito de caráter geométrico, que é: a

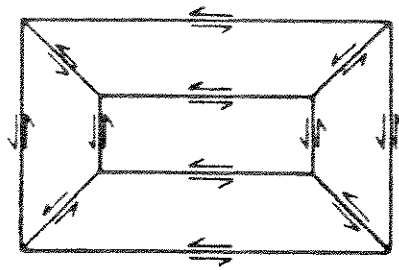


Fig. 3.25 - Determinação de orientabilidade de uma superfície

superfície não deve interceptar a si mesma. A Fig. 3.26 mostra uma situação em que, apesar da topologia ser válida, a posição do vértice superior faz com que a condição acima não seja atendida.

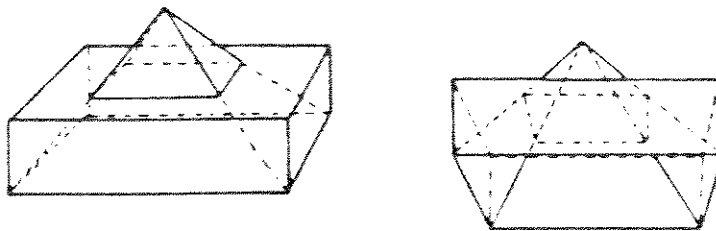


Fig. 3.26 - Superfície que intercepta a si mesma

Um resultado da topologia algébrica de grande relevância prática é a chamada lei de Euler [BAE 79/, [WIL 85b/, [MAN 88/. O matemático Euler mostrou que, para qualquer subdivisão em faces, arestas e vértices de uma superfície que limite um sólido topologicamente equivalente a uma esfera vale a equação:

$$V - A + F = 2$$

Uma generalização do resultado anterior mostra que para qualquer superfície representada por um modelo plano realizável a expressão $V-A+F$ é uma constante que depende da conectividade da superfície, da conectividade de cada uma de suas faces e do número de furos no sólido representado pela superfície. Esta

constante, chamada característica de Euler, é um invariante topológico de um sólido. A equação geral que define a dependência desta constante, conhecida como fórmula de Euler-Poincaré é dada por:

$$V - A + F = 2(S-H) + R$$

onde S é o número de superfícies desconectadas do sólido, H é o número de furos através do sólido e R é o número de furos em faces. Esta equação estende a lei de Euler para três classes de sólidos, além daqueles topologicamente equivalentes à esfera. A primeira, mostrada na Fig. 3.27(a) inclui sólidos com cavidades internas. A segunda, Fig. 3.27(b), inclui sólidos cujas faces não são simplesmente conectadas, ou seja, faces que contêm mais de uma fronteira. A terceira, Fig. 3.27(c), inclui sólidos com furos passantes, ou seja, cujo genus seja maior que zero. A fórmula de Euler-Poincaré pode ser usada para invalidar topologicamente um modelo, uma vez que ela não seja atendida. Porém, só se pode afirmar que um modelo é topologicamente válido se forem também atendidas as condições dele representar uma superfície fechada e orientada.

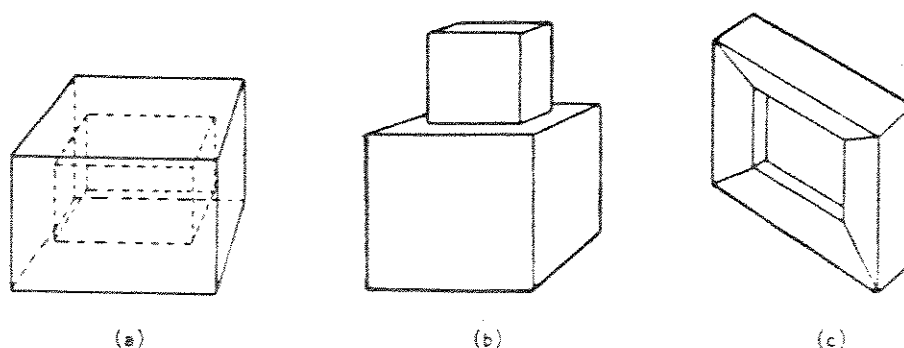


Fig. 3.27 - Sólidos que atendem a fórmula de Euler-Poincaré

Uma vez definidas as propriedades dos sólidos de interesse em modelagem de produto, serão apresentados a seguir esquemas de representação que se baseiam nas abordagens aqui desenvolvidas. Estes esquemas serão analisados quanto à estrutura

de dados que representam os modelos e o conjunto de operadores que atuam sobre estas. Os operadores devem compatibilizar dois objetivos inerentemente contraditórios, pois ao mesmo tempo devem permitir um amplo domínio de sólidos representáveis e garantir a integridade das representações. Como o modelo é utilizado por aplicações automatizadas, esta questão é resolvida restringindo-se as operações possíveis sobre os modelos, ao invés de construí-los e então, verificar sua integridade. Serão também analisadas as propriedades formais de cada esquema de representação.

3.3.1.3.1. Modelos por Decomposição

Modelos por decomposição representam o conjunto de pontos de um sólido como a união de sólidos primitivos disjuntos que são instâncias de uma coleção pré-fixada. Um caso particular deste tipo de modelo é a geração de sólidos como instâncias de primitivas definidas parametricamente. A Fig. 3.28 apresenta um exemplo. Através da especificação de valores para a , b , t , l , h , r_o , r_i e de uma matriz de transformação para posicioná-lo, o sólido é definido. Este tipo de modelo pode ser utilizado como forma de representação da geometria do produto, em tecnologia de grupo, em que famílias de peças parametrizadas são formadas a fim de que se tire proveito de sua similaridade na execução das funções de projeto e manufatura de produtos. Logicamente, o domínio deste esquema de representação é restrito às formas previamente definidas. Além disso, alguns objetos não têm uma parametrização natural como aquele da Fig. 3.28. A validade da representação pode ser facilmente verificada, confrontando-se os valores dos parâmetros a restrições previamente estabelecidas. No caso da Fig. 3.28 pode-se exigir, por exemplo, que $a > 2t$ e $r_o \geq 2r_i$. Uma vez satisfeitas as restrições, pode-se afirmar ser o sólido um r-con. Este esquema é não-ambíguo, pois a atribuição de valores aos parâmetros determina o sólido. Porém, não é único, pois pode haver mais de uma parametrização para um mesmo sólido. A principal desvantagem de sua utilização é que não podem ser

projetados algoritmos genéricos para as aplicações, tendo que ser tratada cada primitiva de forma diferenciada.

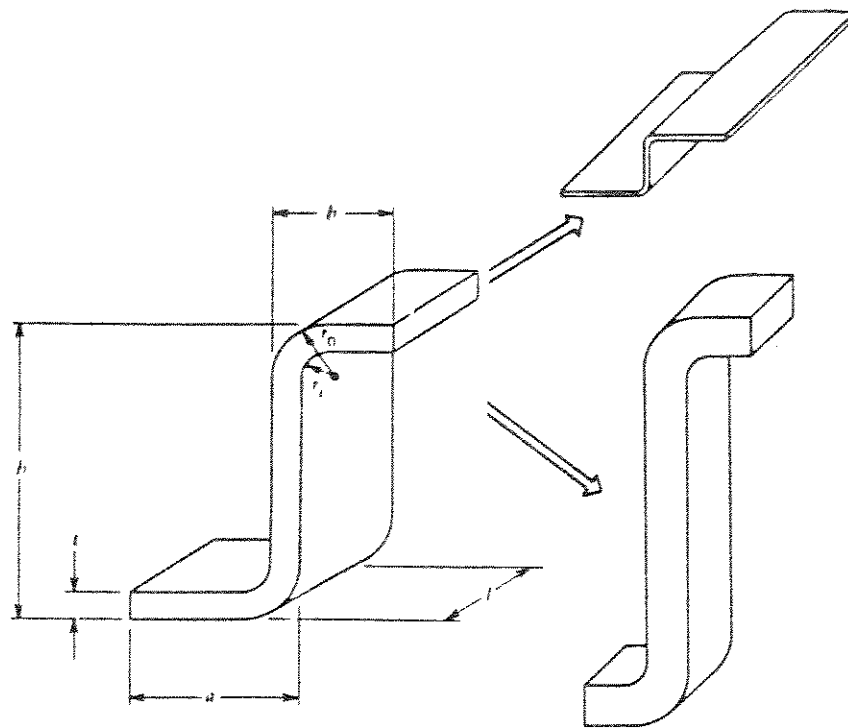


Fig. 3.28 - Instâncias de uma primitiva

Outro tipo de modelo por decomposição é a chamada enumeração exaustiva. Neste esquema o sólido é representado pela união de cubos de tamanho e orientação fixos que estão contidos, total ou parcialmente, no sólido. Cada cubo pode ser representado por uma coordenada, por exemplo, seu centróide. Enquanto em aplicações para processamento digital de imagens estes modelos podem ser criados por técnicas de varredura, no âmbito de modelos de produtos são criados através de conversões de outras formas de representação e, uma vez que se tenha a representação de dois sólidos, através de operações booleanas entre estes. O principal problema deste esquema de representação é que ele somente aproxima um sólido. A obtenção de uma representação com resolução aceitável para aplicações como geração de programas de comando numérico para sólidos com faces curvas implica em utilização de

um grande espaço de armazenamento em computador. Este esquema é não-ambíguo e único. Sua principal vantagem diz respeito aos algoritmos para aplicações que fazem uso do modelo. Pode-se, por exemplo, calcular facilmente o volume de um sólido. A Fig. 3.29 apresenta um exemplo de enumeração exaustiva de um sólido.

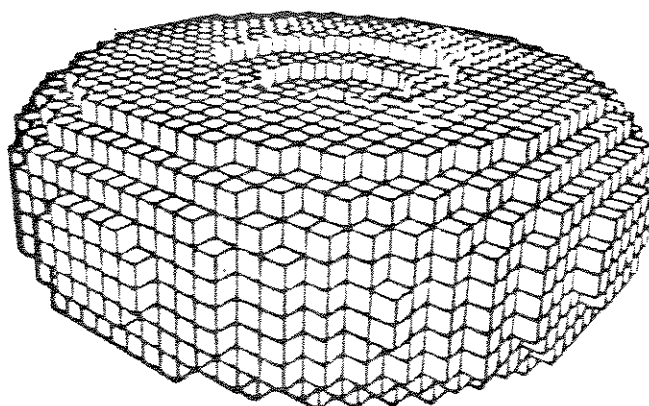


Fig. 3.29 - Enumeração exaustiva de um sólido

A fim de evitar o excessivo espaço de armazenamento exigido pelo esquema por enumeração exaustiva, pode ser utilizado um esquema em que o sólido passa a ser dividido não mais em cubos do mesmo tamanho, mas, sim, através de subdivisão adaptativa por cubos de tamanho variável. Um esquema deste tipo visa eliminar redundância de informação, pois a probabilidade de um cubo estar na fronteira do sólido é muito menor do que não estar. Um exemplo clássico de esquema por subdivisão adaptativa é a representação por árvores octais /YAM 84/, /MEA 82/. Neste esquema é feita uma subdivisão recursiva de um espaço de interesse em octantes até que uma resolução pré-definida seja alcançada. Este procedimento é mostrado na Fig. 3.30. Como se observa, o sólido é representado por uma árvore, em que cada nó tem oito ramos. Cada nó consiste em um código e oito ponteiros para os nós de nível imediatamente inferior. Este código dirá se o octante correspondente está totalmente, parcialmente ou não está contido no sólido. Nos casos do código indicar "totalmente contido" ou

"não está contido" o nó é uma folha da árvore e os oito ponteiros são nulos. Caso contrário, o nó é subdividido, com os ponteiros apontando para os nós inferiores.

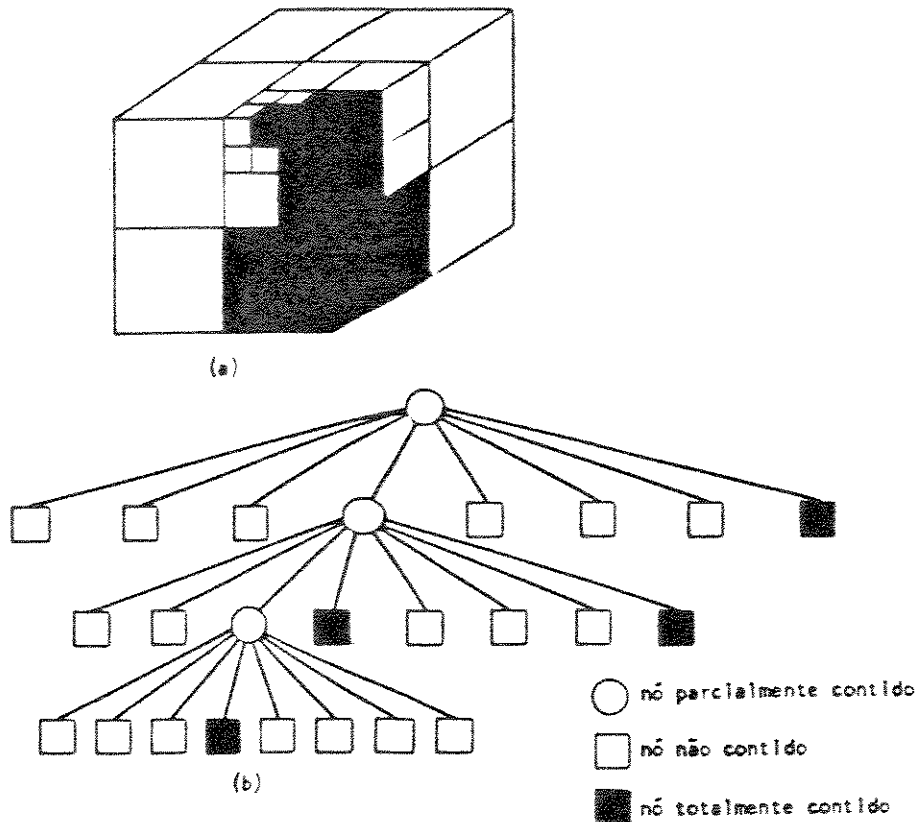


Fig. 3.30 - Representação por árvore octal

Da mesma forma que esquemas por enumeração exaustiva, árvores octais são criadas a partir de conversão de outras representações e dadas duas árvores correspondentes a sólidos, um terceiro pode ser obtido através de operações booleanas entre eles. Quanto às propriedades formais, temos:

- árvores octais representam aproximadamente um sólido. Portanto, seu domínio é limitado pelo espaço de armazenamento necessário à representação de um sólido com uma dada precisão, o que é ditado pelas aplicações que farão uso do modelo. Normalmente, este espaço é bastante grande;

- a validade é garantida, pois qualquer árvore octal representa um r-con;
- o esquema é não-ambíguo e único.

Uma última observação a respeito de árvores octais diz respeito a que operações sobre estas se prestam à implementação em hardware, uma vez que só envolvem aritmética inteira.

3.3.1.3.2. Modelos Construtivos

Modelos construtivos representam o conjunto infinito de pontos que constitui um sólido como uma combinação de conjuntos de pontos primitivos. A combinação é feita através de operações booleanas de união, interseção e diferença entre os conjuntos primitivos. Estes são definidos como instâncias de tipos de conjuntos pré-definidos.

Uma primeira abordagem para esquemas de representação construtiva são os chamados modelos por semi-espacos. Nestes, os conjuntos primitivos são definidos por superfícies que dividem o espaço em dois subconjuntos, sendo um deles tomado como o semi-espaco de interesse para representação do sólido. As superfícies utilizadas para definir os semi-espacos têm, normalmente, uma representação analítica simples, tal como plano, superfície cilíndrica, esférica, cônica. A Fig. 3.31 mostra a representação de um cilindro por semi-espacos. Este é descrito como resultado da interseção entre os semi-espacos definidos por dois planos e uma superfície cilíndrica. Cada semi-espaco é definido associando-se uma posição e orientação ao tipo de superfície que o define.

O domínio de modelos por semi-espacos é especificado pelos tipos de superfícies primitivas disponíveis e pelos operadores booleanos que se podem aplicar sobre os semi-espacos. A validade da representação só pode ser garantida se houver auxílio do usuário para definição do semi-espaco de interesse ou se for

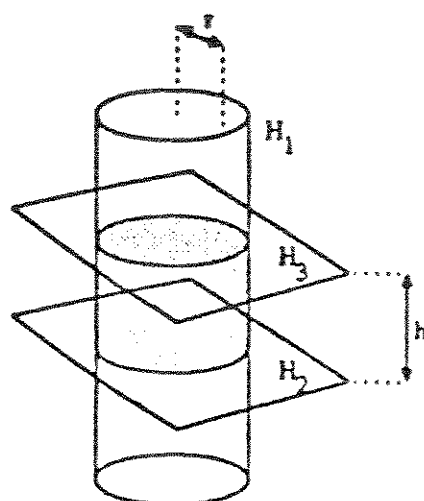


Fig. 3.31 - Modelo por semi-espacos de um cilindro

definido a priori um envelope espacial para o sólido. Caso contrário, podem ser criados sólidos não-limitados. Modelos por semi-espacos são não-ambíguos, pois qualquer combinação válida de semi-espacos determina um sólido. Não são únicos, o que se verifica lembrando-se que são descritos através de operações booleanas entre conjuntos e portanto sujeitos às propriedades destes (comutatividade, associatividade).

A abordagem de maior interesse prático para modelos construtivos é a chamada geometria construtiva de sólidos - CSG. Nos modelos CSG, os conjuntos primitivos são instâncias de sólidos parametrizados. Através de operações booleanas (união, interseção, diferença) entre sólidos primitivos e transformações rígidas (translação, rotação) é definido um novo sólido. A Fig. 3.32 mostra o modelo CSG de um sólido construído a partir de duas primitivas: um paralelepípedo π_1 e um cilindro π_2 . As transformações T_i incluem a atribuição de parâmetros aos sólidos primitivos e seu posicionamento e orientação.

Modelos CSG são representados por árvores binárias. A sintaxe da representação pode ser escrita em notação de Backus-Naur como:

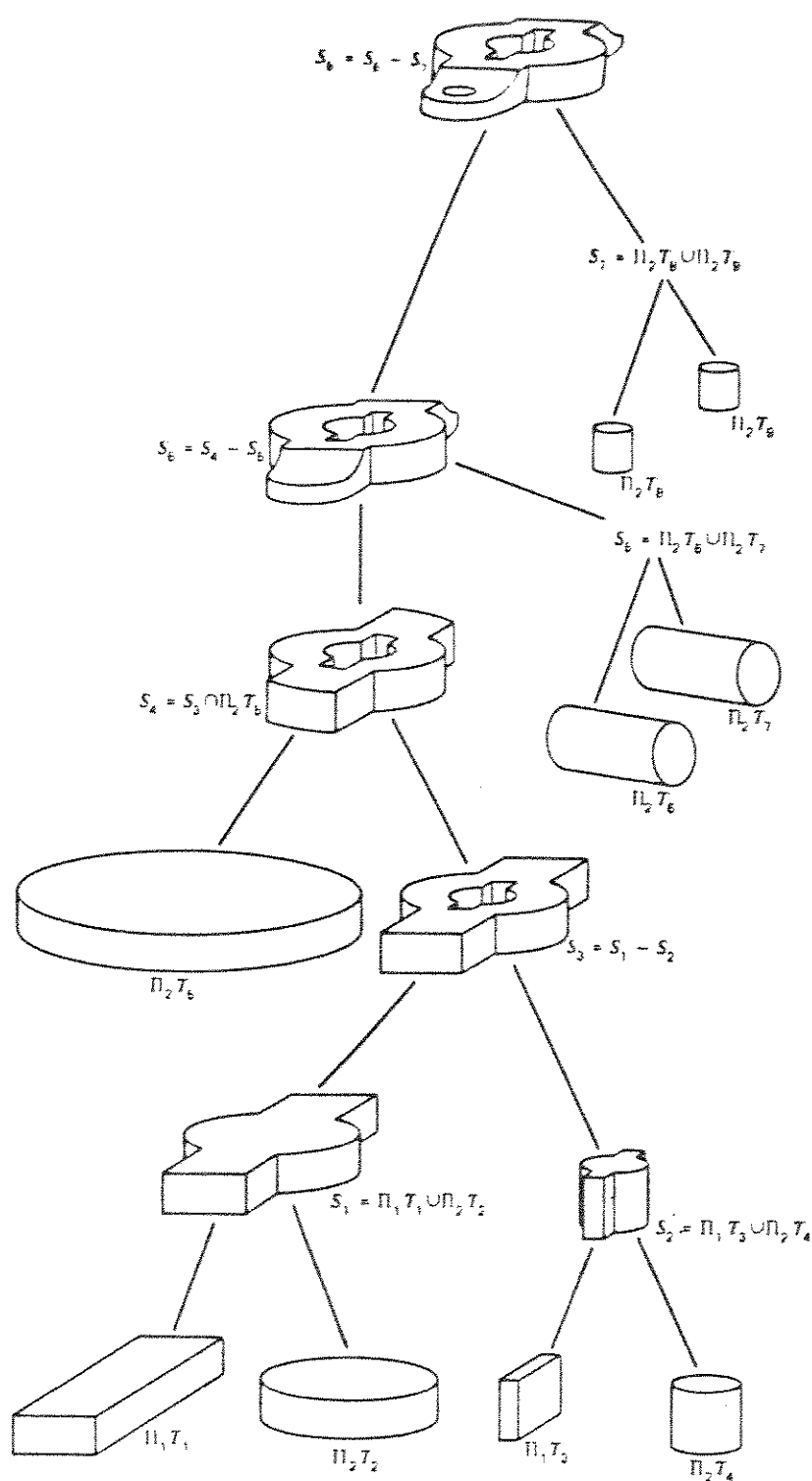


Fig. 3.32 - Modelo CSG de um sólido

$\langle \text{árvore CSG} \rangle :: \langle \text{primitiva} \rangle | \langle \text{árvore CSG} \rangle \langle \text{operação booleana} \rangle$
 $\langle \text{árvore CSG} \rangle | \langle \text{árvore CSG} \rangle \langle \text{transformação rígida} \rangle \langle \text{parâmetros da transformação} \rangle$

Portanto, nós da árvore são operadores booleanos ou transformações rígidas e folhas são instâncias de primitivas ou parâmetros para uma transformação.

Conforme discutido no item 3.3.1.3 o espaço de modelagem, quando se considera um sólido como um subconjunto do espaço Euclidiano tridimensional, é dado por *r-con's*, ou seja, conjuntos limitados e regulares. Portanto, os sólidos primitivos devem também ser *r-con's*. A Fig. 3.33 mostra duas possíveis coleções de sólidos primitivos. Uma vez que estes são representados internamente como combinações de semi-espacos, o domínio de um esquema de representação CSG é ditado pela coleção de semi-espacos que ele utiliza. Sendo assim, as duas coleções da Fig. 3.33 têm o mesmo domínio, pois são utilizados somente semi-espacos planares e cilíndricos.

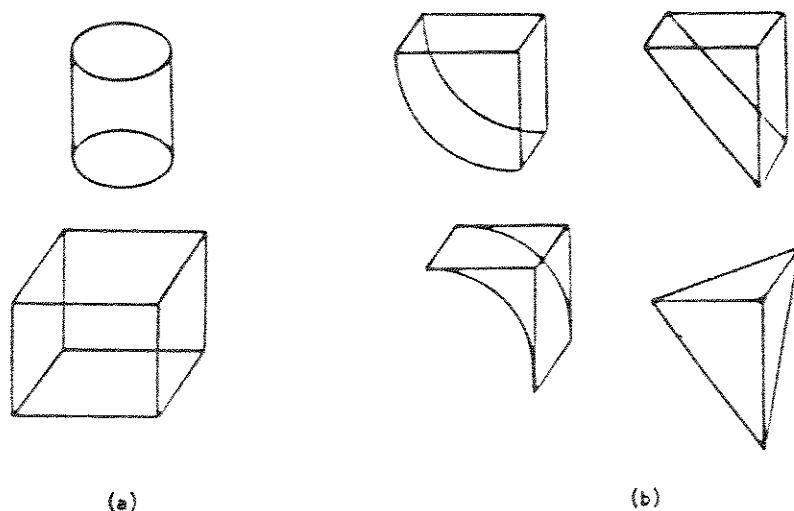


Fig. 3.33 - Coleções de sólidos primitivos para modelos CSG

A questão da validade de modelos CSG impõe restrições aos operadores booleanos. Dado que a coleção de sólidos primitivos é válida, para que os modelos CSG sejam válidos, exigem-se

operadores que produzam como resultado um r-con. Porém, como mostra a Fig. 3.34 (o raciocínio para o caso de sólidos é análogo), operações booleanas sobre r-con's podem resultar em conjuntos não dimensionalmente homogêneos, ou seja, não regulares. Sendo assim, fazem-se necessários operadores booleanos regularizados que eliminem este problema [TIL 80a]. Uma vez que as operações sejam feitas através de operadores regularizados, gaante-se a validade do modelo. Modelos CSG são não-ambíguos, pois cada árvore modela um único sólido. Não são únicos, pelos mesmos motivos citados para modelos por semi-espacos.

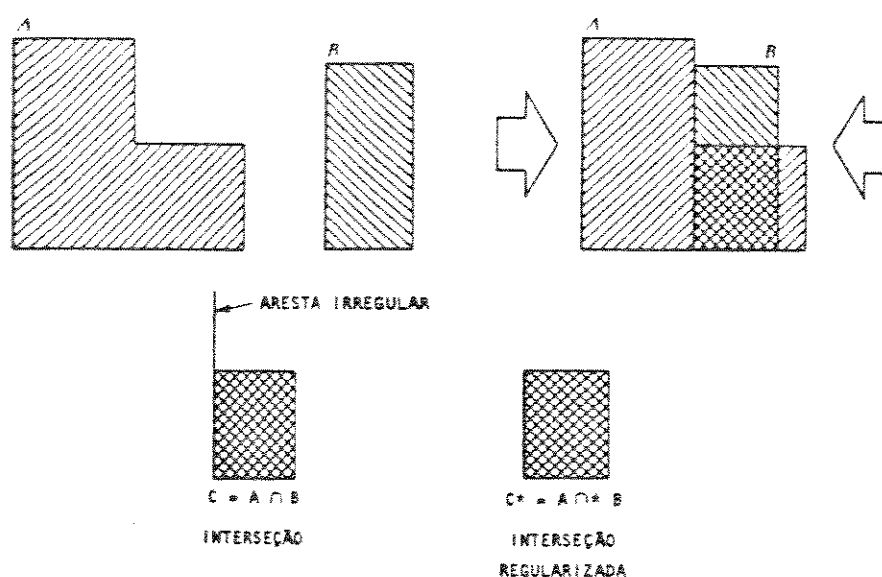


Fig. 3.34 - Interseção regularizada entre conjuntos

Modelos CSG definem a geometria do sólido implicitamente, sendo por isso chamados de modelos procedurais ou não avaliados. Aplicações podem fazer uso da estrutura em árvore para avaliação de propriedades geométricas do sólido, dividindo o problema em duas partes, resolvendo recursivamente cada parte e combinando as soluções parciais para obter a solução global. A recursão termina quando o problema é subdividido em partes que permitam uma solução direta [MAN 88]. Um procedimento deste tipo pode, por exemplo, ser utilizado em um algoritmo para avaliação de

fronteira, em que são geradas as faces, arestas e vértices do sólido. Visando a solução deste e outros problemas, em [TIL 80b] aborda-se, de forma rigorosa, a questão de classificação de conjuntos quanto à pertinência, ou seja, dado um conjunto candidato C e um conjunto de referência R, achar os subconjuntos de C que pertencem ao interior, à fronteira e ao exterior de R. Muitos problemas geométricos têm esta característica. Entre eles podem ser citados:

- inclusão de pontos: dados um sólido e um ponto, está este dentro, fora ou sobre a fronteira do sólido;
- recorte de linhas contra polígonos;
- interseção de polígonos;
- interferência entre sólidos.

A importância da abordagem genérica que Tilove propõe para as questões acima é que esta leva em consideração o requisito de regularidade dos conjuntos.

Uma utilização de esquemas de representação CSG de grande importância em modelagem de produto é a especificação da estrutura do produto. Através da árvore CSG podem ser representadas informações para montagem de um produto em termos de seus componentes.

Modelos CSG têm sido, juntamente com modelos por fronteira, os esquemas de representação de sólidos mais amplamente utilizados [MYE 82]. Brown [BRO 82] e Boyse [BOY 82] apresentam modeladores geométricos em que o esquema de representação primário é CSG.

3.3.1.3.3. Modelos por Fronteira

Modelos por fronteira, conhecidos como modelos B-REP, descrevem um sólido pela superfície que o limita. Esta é representada por uma coleção de faces, que por sua vez são representadas pelo laço de arestas que as limitam: arestas são limitadas por vértices. Uma vez que as faces podem não ser simplesmente conectadas, ou seja, conter furos, estas poderão ser também limitadas por laços internos. Os relacionamentos de adjacência entre estes elementos topológicos e a associação de informação geométrica (superfícies, curvas e pontos) a eles definem um sólido.

Conforme foi observado no item 3.3.1.3, a superfície que envolve um sólido pode ser analisada por meio de modelos planos. Nesta seção também foram estabelecidas condições que modelos planos devem satisfazer para que possam ser mapeados em superfícies que envolvem sólidos fisicamente realizáveis: os modelos planos devem representar superfícies fechadas e orientadas. O requisito da superfície não interceptar a si mesma não pode ser analisado por meio de modelos planos, uma vez que se trata de uma restrição de caráter geométrico e não topológico. Finalmente, verificou-se que a fórmula de Euler-Poincaré deve ser satisfeita. Portanto, o problema a ser resolvido na implementação de modelos B-REP é encontrar estruturas de dados que representem modelos planos de modo não-ambíguo e de tal forma que as condições acima sejam satisfeitas. Por outro lado, deve ser encontrado um conjunto de operadores que crie e manipule a estrutura de dados mantendo a sua integridade. Através deste procedimento, garante-se que todas as representações criadas são topologicamente válidas, o que é conveniente ao ambiente CIM, em que o modelo será utilizado por aplicações automatizadas, não se podendo, a priori, contar com o usuário para a validação.

Uma vez que a estrutura de dados representa um modelo plano, ela deve conter a estrutura do grafo e a estrutura de identificação das arestas e vértices. Conforme 3.3.1.3 a condição de que a superfície deve ser fechada implica numa estrutura de

identificação em que cada aresta é identificada com uma única outra aresta e as faces identificadas em cada vértice formam um ciclo de tal forma que a vizinhança do vértice seja um círculo. Além disso, a condição da superfície ser orientada implica que cada aresta deve ocorrer uma vez no sentido positivo e outra no negativo. Estas propriedades devem ser exploradas ao se projetar a estrutura de dados. Outra questão diz respeito à suficiência topológica da representação. Como mostra a Fig. 3.35, dadas as três entidades topológicas (faces, arestas e vértices), são possíveis nove relacionamentos de adjacência [WEI 85]. Uma vez

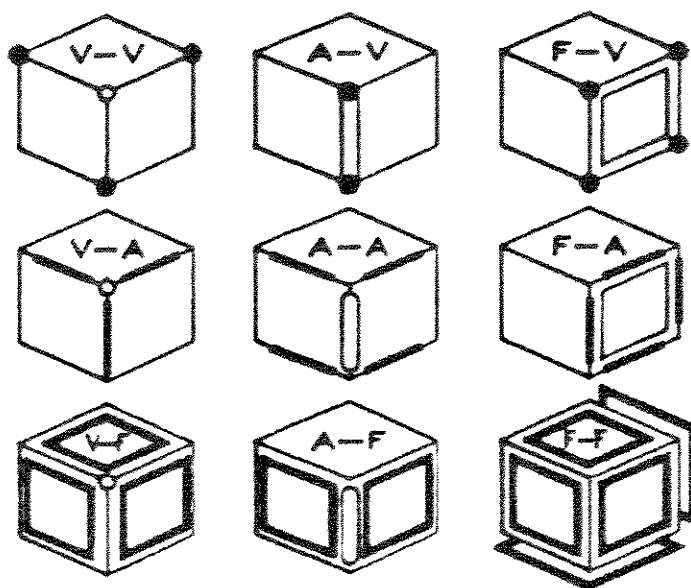


Fig. 3.35 - Relacionamentos entre entidades topológicas

que se exige que o esquema de representação seja não-ambíguo, a estrutura de dados deve ser tal que se possa extrair dela qualquer dos nove relacionamentos. Isto não significa que todos eles devam ser armazenados. A suficiência topológica dos relacionamentos depende do domínio assumido para o esquema de representação. Se este for restrito a poliedros com faces planas, V-V ou F-V (ver Fig. 3.35) são suficientes, o que significa que todos os outros oito relacionamentos podem ser obtidos a partir de um dos dois. Porém, se o domínio admitir faces cujas

superfícies são curvas, os grafos se tornam mais complexos, pois, como mostra a Fig. 3.36, podem existir auto-laços (uma aresta liga um vértice a si mesmo) e multigrafos (mais de uma aresta liga dois vértices). Para o caso geral, em que são permitidas superfícies curvas, somente V-A, F-A e A-A são topologicamente suficientes /WEI 85/. O relacionamento A-V, que corresponde a modelos de arame, não é suficiente em domínio algum, como pode ser visto na Fig. 3.6. Além disso, como a cada entidade topológica é associada informação geométrica (o que não é necessário em sólidos com faces planas), um mínimo de dois relacionamentos contendo as três entidades deve estar presente. O armazenamento de relacionamentos de adjacência, além do suficiente, é frequentemente justificado pela melhoria de desempenho das aplicações que esta redundância proporciona. No caso de geração de programas de comando numérico, por exemplo, é de grande valia o armazenamento do relacionamento F-F.

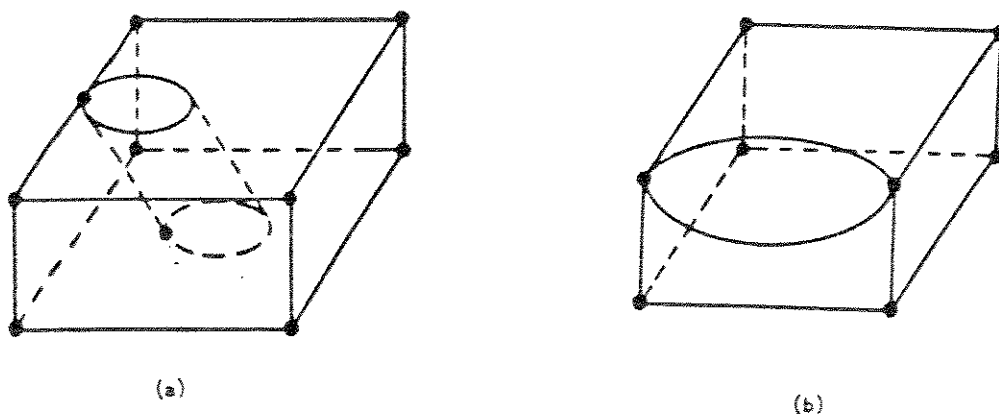
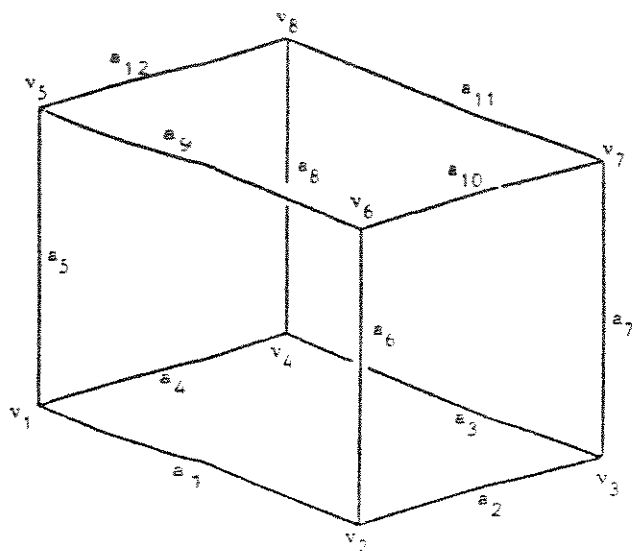


Fig. 3.36 - Sólidos com auto-laço e multigrafo

Delineados os parâmetros a serem considerados no projeto de uma estrutura de dados para modelos B-REP, será agora apresentado um exemplo que satisfaz os requisitos impostos. Para ilustrar uma instância da estrutura de dados, será utilizado o paralelepípedo mostrado na Fig. 3.37 junto com a definição de suas faces. A estrutura, conhecida como estrutura-asa /MAN 88/, /WEI 85/, será apresentada em pseudo-código e esquematicamente.



FACE	VERTICES
f ₁	v ₁ v ₂ v ₃ v ₄
f ₂	v ₅ v ₂ v ₁ v ₅
f ₃	v ₇ v ₃ v ₂ v ₆
f ₄	v ₈ v ₄ v ₃ v ₇
f ₅	v ₅ v ₁ v ₄ v ₈
f ₆	v ₈ v ₇ v ₆ v ₅

Fig. 3.37 - Sólido a ser modelado por fronteira

No pseudo-código * significa ponteiro. A Fig. 3.38 define a estrutura de dados. Esta, na forma apresentada, é adequada a sólidos com faces planas, sendo, por isto, desnecessário associar informação geométrica a faces e arestas. Uma aresta é orientada de seu vértice inicial para o vértice final. Quanto às faces convencionam-se positivo o sentido horário quando visto de fora do sólido. A estrutura "sólido" contém um ponteiro para uma face de uma lista duplamente ligada de faces. A estrutura "face" contém ponteiros para a próxima face e a face anterior na lista e um ponteiro para o laço de arestas externo da face. A estrutura "laço" contém um ponteiro para o próximo laço em uma lista ligada. Nesta lista o primeiro nó, apontado pela estrutura face, é o laço exterior e os nós subsequentes são laços interiores usados no caso de faces com furos. Além disso, a estrutura laço contém um ponteiro para uma aresta do laço. A estrutura aresta contém ponteiros para os vértices inicial e final da aresta (vert_ini, vert_fim), ponteiros para arestas posterior e anterior na face onde a aresta ocorre em sua orientação positiva (post_are_hor, prev_are_hor), ponteiros para arestas posterior e prévia na face onde a aresta ocorre em sua orientação negativa (post_are_antihor, prev_are_antihor) e ponteiros para as faces

```

estrutura sólido
  lista_de_faces: #face
fim
estrutura face
  lista_de_laços: #laço
  prox_face, ant_face: #face
fim
estrutura laço
  prox_laço: #laço
  lista_de_arestas: #aresta
fim
estrutura aresta
  vert_ini, vert_fim: #Vértice
  post_are_hor, prev_are_hor: #aresta
  post_are_antihor,
    prev_are_antihor: #aresta
  face_hor, face_antihor: #face
fim
estrutura vértice
  x,y,z: real
fim

```

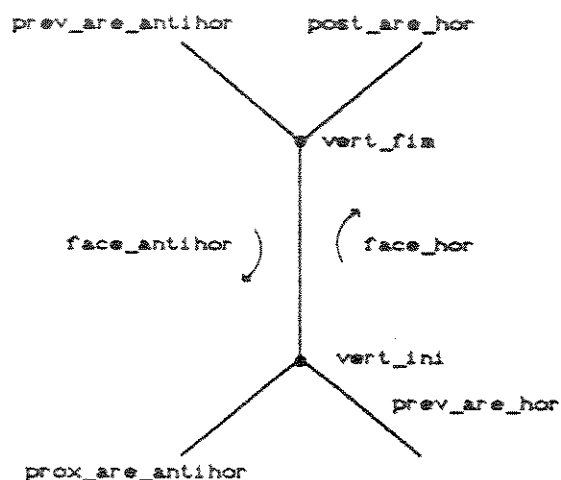


Fig. 3.38 - Estrutura de dados para modelo B-REP

onde a aresta aparece positiva e negativamente (face_hor, face_antihor). Finalmente, a estrutura vértice contém as coordenadas que definem geometricamente um vértice. A Tabela 3.2 apresenta a estrutura-asa para o sólido da Fig. 3.37. Nela é omitida a informação sobre laços, pois o sólido não tem faces com laços interiores.

Observando a estrutura de dados proposta frente aos requisitos anteriormente estabelecidos conclui-se que:

- ela sempre corresponde a superfícies fechadas, pois cada aresta ocorre em duas faces;
- ela sempre corresponde a superfícies orientadas, pois cada aresta ocorre uma vez em sua orientação positiva e outra na negativa;
- ela é topologicamente suficiente, pois armazena o relacionamento de adjacência A-A.

ARESTA	VERT_INI	VERT_FIM	FACE_HOR	FACE_ANTI_HOR	POST_ARE_HOR	PREV_ARE_HOR	POST_ARE_ANTI_HOR	PREV_ARE_ANTI_HOR
A ₁	V ₁	V ₂	F ₁	F ₂	A ₂	A ₄	A ₅	A ₆
A ₂	V ₂	V ₃	F ₁	F ₃	A ₃	A ₁	A ₆	A ₇
A ₃	V ₃	V ₄	F ₁	F ₄	A ₄	A ₂	A ₇	A ₈
A ₄	V ₄	V ₁	F ₁	F ₅	A ₁	A ₃	A ₈	A ₅
A ₅	V ₁	V ₅	F ₂	F ₅	A ₉	A ₁	A ₄	A ₁₂
A ₆	V ₂	V ₆	F ₃	F ₂	A ₁₀	A ₂	A ₁	A ₉
A ₇	V ₃	V ₇	F ₄	F ₃	A ₁₁	A ₃	A ₂	A ₁₀
A ₈	V ₄	V ₈	F ₅	F ₄	A ₁₂	A ₄	A ₃	A ₁₁
A ₉	V ₅	V ₆	F ₂	F ₆	A ₆	A ₅	A ₁₂	A ₁₀
A ₁₀	V ₆	V ₇	F ₃	F ₆	A ₇	A ₆	A ₉	A ₁₁
A ₁₁	V ₇	V ₈	F ₄	F ₆	A ₈	A ₇	A ₁₀	A ₁₂
A ₁₂	V ₈	V ₅	F ₅	F ₆	A ₅	A ₈	A ₁₁	A ₉

FACE HOR	PRIMEIRA ARESTA
F ₁	A ₁
F ₂	A ₉
F ₃	A ₆
F ₄	A ₇
F ₅	A ₁₂
F ₆	A ₈

TABELA 3.2 - Estrutura-asa para o sólido da Fig. 3.37

Quanto aos operadores, exige-se que estes só criem instâncias válidas da estrutura de dados. Devido à complexidade desta não é conveniente que aplicações as manipulem diretamente, sendo, portanto, desejável a existência de um conjunto de operadores que tornem a estrutura de dados transparente à aplicação e ao mesmo tempo sejam suficientemente flexíveis de modo que possam ser descritos todos os modelos planos fisicamente realizáveis. Estes requisitos são satisfeitos pelos chamados operadores de Euler /MAN 82b/, /EAS 79/, pois sua funcionalidade pode ser derivada a partir da fórmula de Euler-Poincaré,

$$V - A + F = 2(S-H) + R$$

que, conforme explicado anteriormente, todos os sólidos realizáveis obedecem. Esta equação pode ser interpretada como um espaço discreto de seis dimensões em que os sólidos realizáveis são definidos por um hiperplano. Portanto, os operadores devem ser tais que o novo ponto $P = (V \ A \ F \ S \ H \ R)$ que é resultado da

operação esteja contido neste hiperplano. Mantyla /MAN 88/ mostra que existe um conjunto de cinco operadores que satisfazem esta condição e ao mesmo tempo podem criar todos os modelos planos realizáveis. Além disso, mostra que todas as instâncias de modelos planos criadas por estes operadores correspondem a superfícies fechadas e orientadas. Portanto, este conjunto de operadores, junto com os cinco operadores inversos que destroem um modelo plano, é uma ferramenta adequada à definição e manipulação de sólidos em modelos B-REP. Os cinco operadores e seus inversos são (ver modelos planos correspondentes na Fig. 3.39, em que o círculo tracejado representa a face externa do modelo plano):

- CVFS : cria vértice, face e sólido
- DVFS : destroi vértice, face e sólido
- CAV : cria aresta e vértice
- DAV : destroi aresta e vértice
- CAF : cria aresta e face
- DAF : destroi aresta e face
- DACL : destroi aresta e cria laço interno
- CADL : cria aresta e destroi laço interno
- DFCLH : destroi face e cria laço interno e furo
- CFDLH : cria face e destroi laço interno e furo

A Fig. 3.39(f) mostra o caso em que é criado um laço interno sem arestas. Os operadores DFCLH e CFDLH são mostrados na Fig. 3.40. Sua representação por modelos planos é prejudicada pelo fato de que estes operadores alteram propriedades topológicas globais do sólido, como conectividade e o genus topológico (número de furos passantes). No caso da Fig. 3.40(a) o operador transforma a face f_2 em um laço interno da face f_1 e cria um furo. Em 3.40(b) os dois sólidos passam a ser considerados um único. Note-se que nos dois casos a expressão $(S-H)$ diminui uma unidade, sendo a fórmula de Euler-Poincaré obedecida, uma vez que o número de laços internos aumenta uma unidade. As vantagens de utilização de operadores de Euler podem ser assim resumidas:

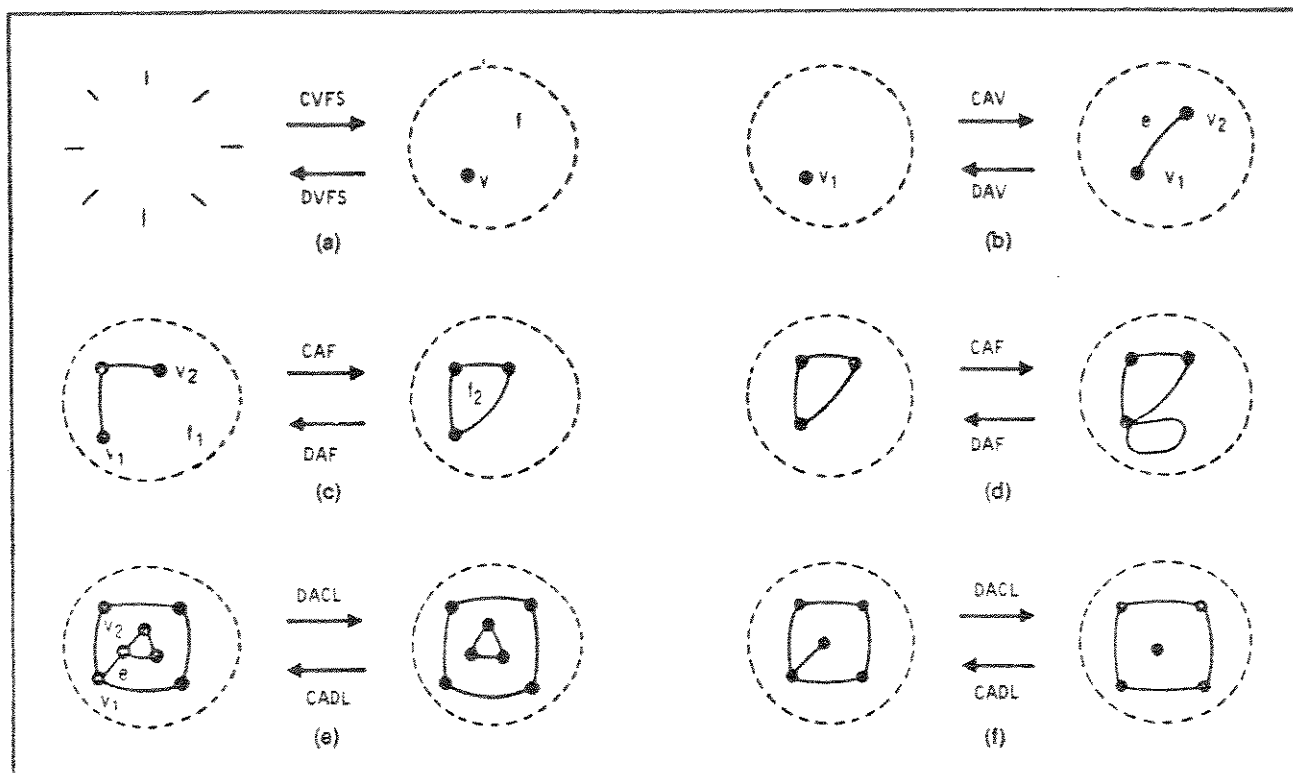


Fig. 3.39 - Modelos planos dos operadores de Euler

- garantem a integridade topológica da representação. A integridade geométrica tem que ser tratada em outro nível do modelador;
- fazem com que as aplicações fiquem independentes da estrutura de dados utilizada para representar modelos planos facilitando o desenvolvimento destas, pois todo o acesso à estrutura de dados é implementado uma única vez e compartilhado pelas aplicações;
- modelos B-REP podem ser armazenados em bancos de dados como sequências de operadores de Euler /MAN 82a/.

Encerrando a discussão sobre operadores de Euler, a Fig. 3.41 apresenta a definição de um cubo com um furo através destes operadores.

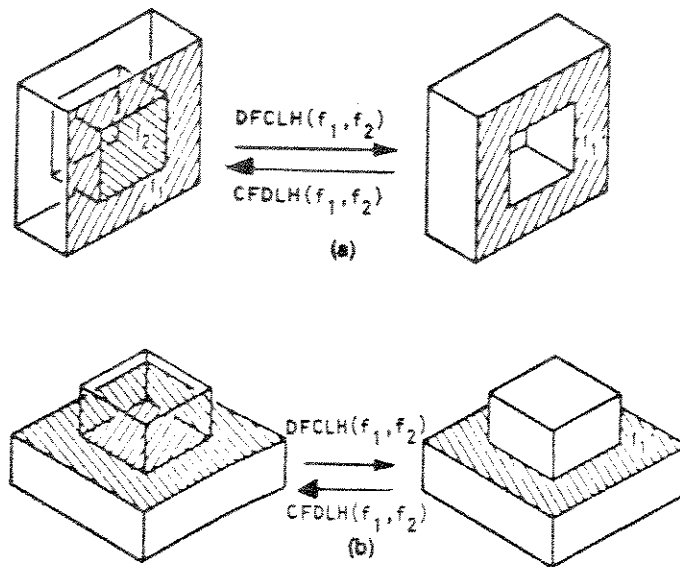


Fig. 3.40 - Operadores DFCLH e CFDLH

Quanto às propriedades formais de modelos B-REP, podemos concluir:

- o domínio do esquema de representação depende das superfícies que podem ser utilizadas. Não há restrição teórica alguma, como se verifica em modelos CSG, à representação de sólidos que contenham superfícies de dupla curvatura. Kimura /KIM 84a/ apresenta um modelador geométrico que utiliza esquema de representação B-REP e cujo domínio inclui sólidos com superfícies de dupla curvatura;
- são não-ambíguos, como foi mostrado anteriormente;
- não são únicos, pois um mesmo sólido pode ser subdividido em faces de diferentes maneiras;
- a integridade topológica da representação é garantida pelo uso de operadores de Euler. A garantia de integridade geométrica exige testes cuja carga computacional pode ser grande (verificação de interseção entre faces).

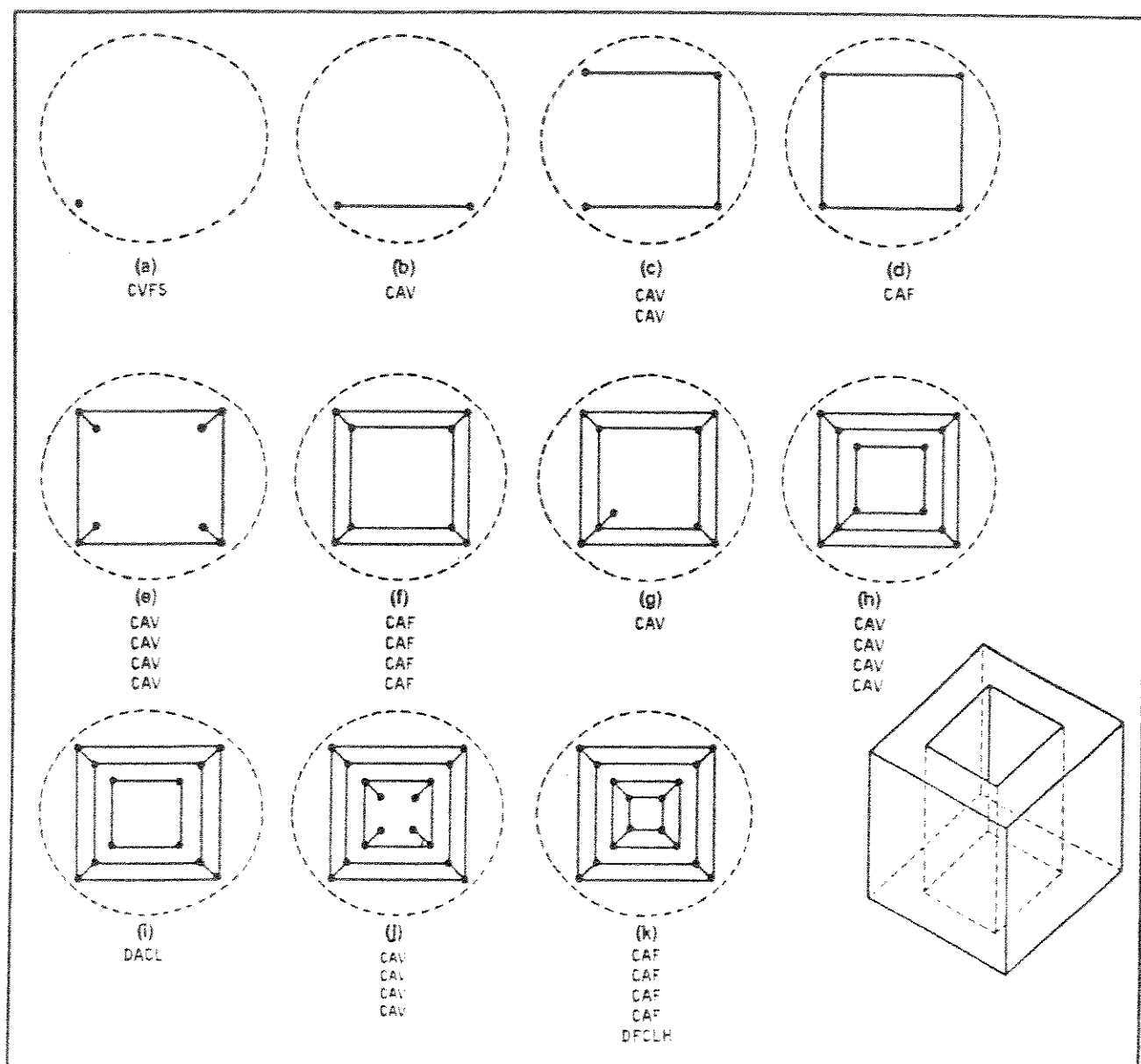


Fig. 3.41 - Definição de um sólido por operadores de Euler

Além da ausência de restrição quanto ao domínio, modelos B-REP apresentam, em relação aos modelos CSG, a vantagem de que as informações geométricas estão prontamente disponíveis. Devido a isso, estes modelos têm sido utilizados como esquema primário de representação de inúmeros modeladores, tanto no âmbito acadêmico quanto comercial. Braid /BRA 79/ e Hillyard /HIL

82/ apresentam sistemas baseados neste tipo de modelo. Porém, se modelos B-REP são menos concisos que os modelos CSG e, além disso, os operadores de Euler não são uma forma adequada de interação com o usuário, exigindo uma interface que não é baseada nos conceitos utilizados para representação interna no modelador. Estas questões e uma comparação dos esquemas de representação frente a várias aplicações, mostram que nenhum modelo é o mais indicado para todas elas. Sendo assim, vários sistemas de modelagem geométrica utilizam mais de um esquema de representação. A manutenção da integridade entre as representações leva ao problema de conversão entre elas. Requicha /REQ 83/ analisa esta questão, mostrando quais modelos podem ser convertidos em outros sem perda de informação. Caracterizadas as formas de representação de geometria, a próxima seção abordará outras informações relevantes para a descrição de um produto.

3.3.2. Informações Tecnológicas

Uma inspeção dos problemas a serem solucionadas, desde a fase de concepção até a entrega de um produto, mostra que os modelos sólidos, embora necessários, não são suficientes como fonte de informação sobre produtos em um ambiente CIM. Isto se deve principalmente a dois fatores /SHA 88/:

- definição incompleta do produto: somente a geometria nominal do produto é definida. Informações relevantes à manufatura, tais como tolerâncias, acabamento superficial e materiais não são representadas;
- baixo nível de abstração na definição do produto: como foi citado no item 2.1, a definição do produto em termos de entidades geométricas e topológicas é incompatível com as funções de manufatura, em que as informações são tratadas de forma agregada, em termos de entidades relacionadas a processos de manufatura, tais como furos, rebaixos e chanfros. A aplicação de planejamento de processos

auxiliado por computador (CAPP) lida com a informação neste nível de abstração e não linhas, círculos e seus relacionamentos de adjacência.

As questões levantadas acima implicam que, para a completa definição de um produto, as seguintes informações de caráter tecnológico devem ser incluídas no modelo:

- Formas Características:

São as entidades com que as funções de manufatura lidam (furos, cavidades,...). A definição das formas características presentes em um produto permite a geração automática do plano de fabricação do produto, tanto no que diz respeito à definição dos processos quanto ao seu sequenciamento, viabilizando, assim, a integração entre projeto e manufatura /PRA 84/, /PAT 88/. Três abordagens têm sido utilizadas para associar formas características a modelos sólidos:

- reconhecimento das formas características pelo usuário: interativamente o usuário aponta as superfícies que compõem uma forma característica. Chang /CHA 85/ apresenta um sistema de auxílio ao planejamento de processos que utiliza esta abordagem para obtenção das formas características;
- reconhecimento e extração automática de formas características: nesta abordagem um programa examina a estrutura de dados de um modelo sólido, procurando reconhecer padrões que permitam deduzir que se trata de uma determinada forma característica. Logicamente, este programa depende do esquema de representação de sólidos utilizado. Lee /LEE 87/ propõe um algoritmo para extração de formas características de modelos CSG. Estes, apesar de manterem a informação geométrica em um nível de abstração mais alto que modelos B-REP, não representam, necessariamente, formas características, o que fica evidente lembrando-se que modelos CSG não são únicos, ou seja, mais de uma

Árvore pode representar o mesmo sólido. O algoritmo de Lee reconstrói a árvore CSG, procurando agrupar primitivas espacialmente adjacentes em sub-árvores, através das quais serão reconhecidas as formas características. Além disso, para que haja uma relação direta com processos de manufatura são eliminadas as operações de interseção, uma vez que cada uma pode ser substituída por duas operações de diferença entre primitivas. Joshi /JOS 88/ propõe inferir formas características a partir de grafos de adjacências de faces obtidas de modelos B-REP. Nestes grafos nós representam faces e arcos representam arestas que unem as faces. A cada aresta é associado um número que indica se as faces que a compartilham formam uma superfície côncava ou convexa. A Fig. 3.42 apresenta uma peça e seu grafo de adjacência de faces. Por outro lado,

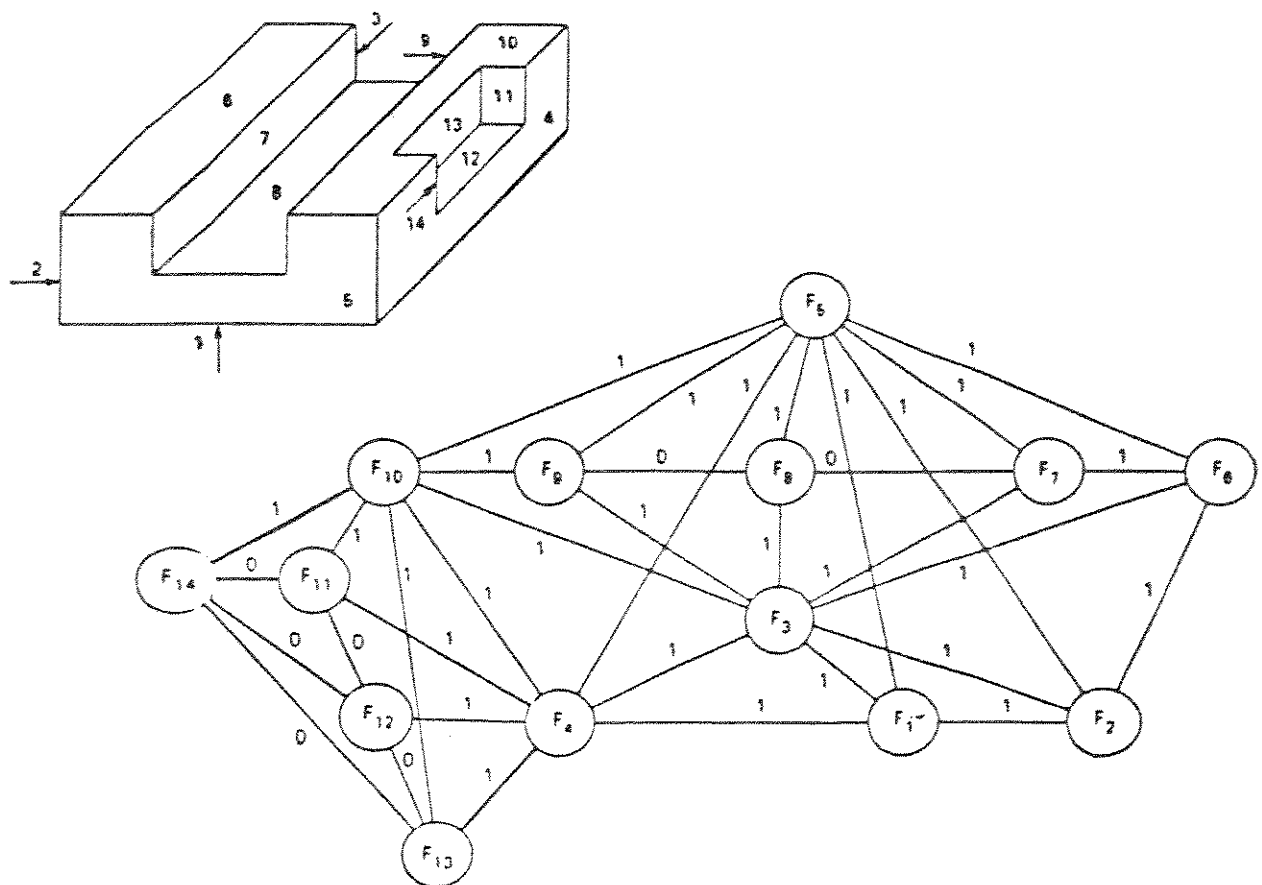


Fig. 3.42 - Grafo de adjacência de faces para uma peça

são definidos grafos análogos para as formas características. A Fig. 3.43 mostra alguns exemplos.

Sendo assim, o algoritmo de extração consiste em encontrar no grafo da peça os subgrafos correspondentes a formas características. Como este problema demanda uma carga computacional muito grande, Joshi propõe a utilização de heurísticas para sua solução. Henderson /HEN 84/ propõe um sistema especialista para o reconhecimento e extração de formas características em modelos B-REP;

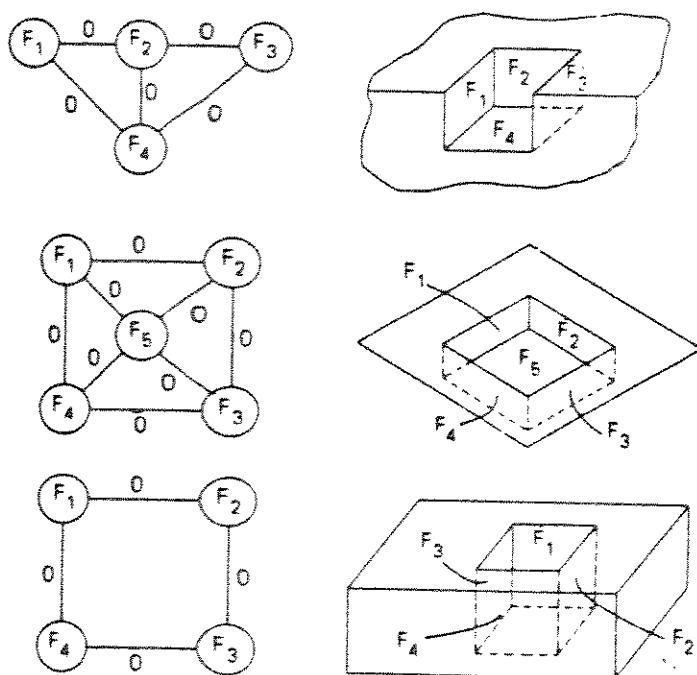


Fig. 3.43 - Grafos de adjacência de faces para formas características

- modelagem baseada em formas características: nesta abordagem o modelo é construído a partir de formas características armazenadas em uma biblioteca. Uma vez que os projetistas pensam em termos de informação de alto nível, a representação da informação de produtos somente em termos de entidades topológicas e geométricas implica em esforço redundante para retomar a informação fornecida

pelo projetista. Modelando-se o produto em termos de formas características, a informação de alto nível é retida no sistema. Desta forma, obtém-se um modelo de produto em vários níveis de abstração, adequado à gama de aplicações típicas de um ambiente CIM.

- Tolerâncias:

Os modelos geométricos lidam com formas idealizadas de produtos. Porém, nas situações reais de manufatura, estão envolvidos desvios da forma nominal de um produto. Sendo assim, faz-se necessário um modelo para representação dos vários tipos de tolerâncias (forma, tamanho, orientação e posição), normalmente encontrados em desenhos de engenharia, de tal forma que estas possam ser utilizadas pelas aplicações voltadas à manufatura /TRU 88/. Roy /ROY 88/ propõe a utilização de um modelo híbrido CSG/B-REP para representação de tolerâncias, uma vez que estão envolvidos, desde relacionamentos entre entidades de baixo nível, como vértice-vértice, vértice-aresta, face-face até relacionamentos entre primitivas, para as quais se utiliza o modelo CSG. Além disso, é possível associar atributos a elementos geométricos, como, por exemplo, acabamento superficial.

- Materiais:

O modelo de produto deve prover meios para especificação do material com que é feito o produto ou cada parte dele. Normalmente as aplicações voltadas à análise do produto requerem a especificação de vários coeficientes associados ao material. No caso de análise estrutural por elementos finitos, uma matriz de rigidez é requerida.

3.4. CONCLUSÃO

Encerrando este capítulo, a Tabela 3.3 sintetiza as diferenças entre os vários tipos de modelos geométricos apresentados e os modelos de produto adequados ao ambiente CIM. No próximo capítulo serão apresentados esforços que visam padronizar modelos que contenham as informações aqui inspecionadas.

	GEOMETRIA	TOPOLOGIA	TOLERANCIAS	FORMAS CARACTER.
MODELO DE ARAME BI- DIMENSIONAL	CURVAS 2D	VISUAL	TEXTO	VISUAL
MODELO DE ARAME TRI- DIMENSIONAL	CURVAS 3D	AMBIGUO	TEXTO	VISUAL
MODELO DE SUPERFICIE	SUPERFICIES BEZIER, B-SPLINE	AMBIGUO	TEXTO	VISUAL
MODELO SOLIDO	TODOS OS SOLIDOS RE- PRESENTAVEIS	NAO- AMBIGUO	TEXTO	VISUAL
MODELO DE PRODUTO	TODOS OS SOLIDOS	NAO- AMBIGUO	MODELO DE TOLERANCIAS	MODELO DE FORMAS CARACTER.

TABELA 3.3 - Características de vários tipos de modelos

CAPITULO 4

PADRONIZAÇÃO EM TROCA DE DADOS DE DEFINIÇÃO DE PRODUTOS

Na década de setenta, com a proliferação de sistemas CAD (Projeto Auxiliado por Computador), usuários e fabricantes começaram a sentir a necessidade de estabelecer padrões para troca de informações entre estes sistemas. Até então, a solução usual era o desenvolvimento de tradutores específicos, ou seja, para cada par de sistemas tinham que ser desenvolvidos dois tradutores. A Fig. 4.1 mostra que para o caso de três sistemas são necessários seis tradutores e que a inclusão de um quarto sistema requer o desenvolvimento de mais seis. Além disso, o

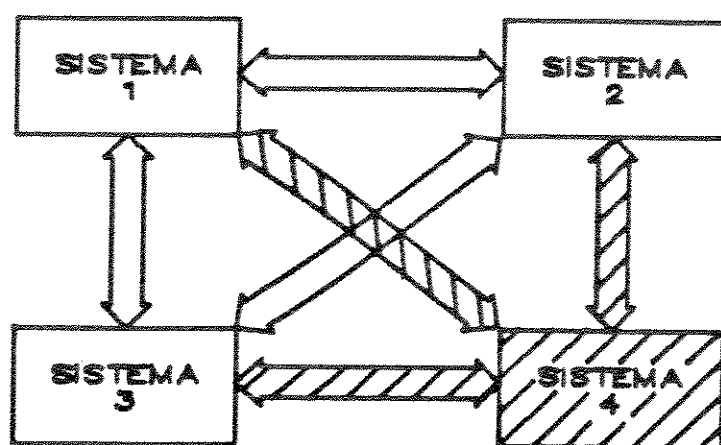


Fig. 4.1 - Uso de tradutores diretos para transferência de dados entre sistemas CAD

desenvolvimento destes tradutores exige conhecimento das estruturas de informação dos sistemas envolvidos, as quais, muitas vezes, não são acessíveis a outros fabricantes. Uma vez reconhecidos estes problemas, passou-se a buscar soluções que empregassem o conceito de formato neutro. Desta forma, como mostra a Fig. 4.2, para cada sistema devem ser desenvolvidos um pré-processador, que transforma informação do sistema para o formato neutro, e um pós-processador, que transforma do formato neutro para o do sistema (ver Cap. 2).

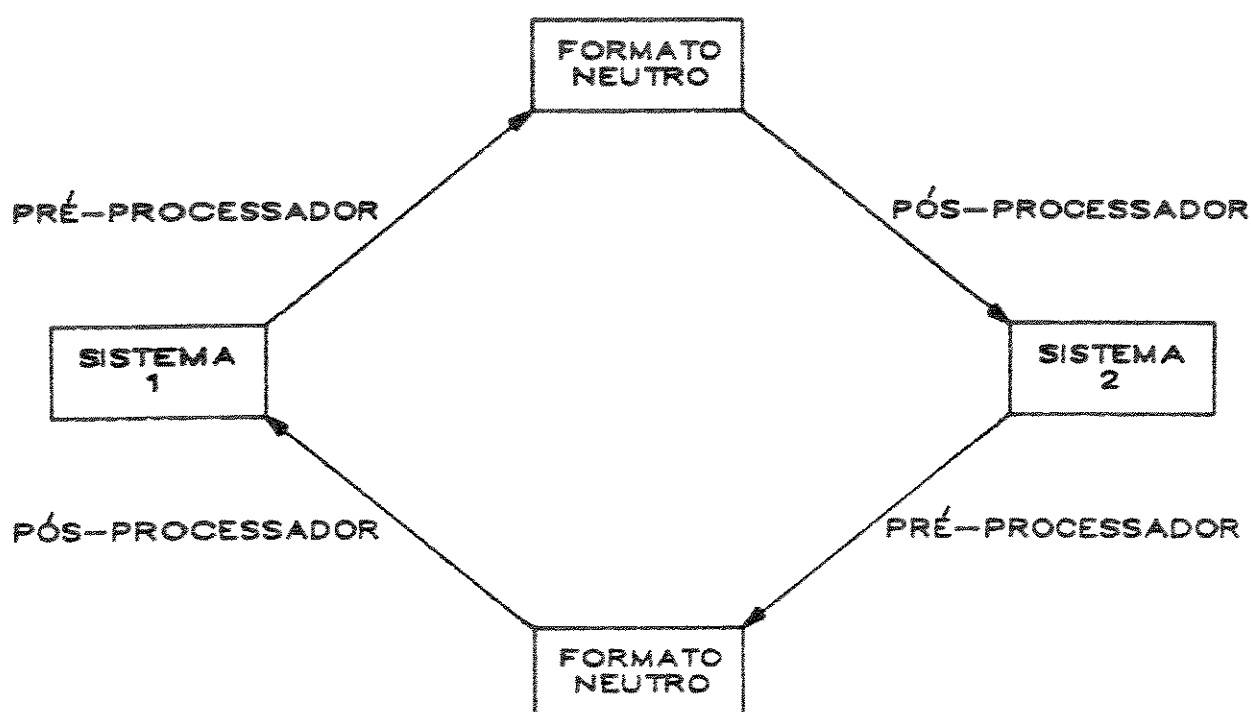


Fig. 4.2 - Transferência de dados usando formato neutro

Nos dias de hoje, com os requisitos impostos pela Manufatura Integrada por Computador (CIM), o problema se apresenta não mais, unicamente, como a troca de dados entre sistemas CAD, mas, sim, como a transferência de informações que definem completamente um produto entre os diversos sistemas presentes em um ambiente CIM. Conforme será visto em seções posteriores, a tendência atual de padronização objetiva a solução deste problema global.

4.1. EVOLUÇÃO HISTÓRICA DOS PADRÕES

Surpreendentemente, as primeiras especificações para troca de dados entre sistemas CAD foram relativas a modelos sólidos, quando seria de se esperar que inicialmente surgissem padrões que possibilitassem troca de dados entre sistemas voltados à geração de desenhos de engenharia /WIL 87a/. Em 1979, a entidade CAM-I (Computer Aided Manufacturing - International) firmou um contrato

com a empresa McDonnell Douglas Automation (McAuto) para desenvolver uma especificação de arquivo para transferência de dados entre sistemas baseados em modelos sólidos. Esta especificação cobriu tanto modelos construtivos (CSG), quanto aqueles representados por fronteiras (B-REP). Pouco depois, foi firmado um contrato entre CAM-I e a empresa Shape Data para desenvolvimento de uma interface procedural que permitisse acesso a estruturas de dados de sistemas de modelagem de sólidos. Esta interface consiste em aproximadamente 150 subrotinas que cobrem modelos construtivos, por fronteiras e modelos simplificados de representação por fronteiras usados para descrever placas de metal e sistemas de tabulação industrial. Este conjunto de rotinas ficou conhecido como AIS (Application Interface Specification) /WIL 85a/. O trabalho foi completado em 1980.

4.1.1. Desenvolvimento de IGES

Simultaneamente aos esforços descritos anteriormente, foi criado um comitê na ANSI (American National Standards Institute) para desenvolver um padrão para troca de dados entre sistemas voltados à geração de desenhos de engenharia. Paralelamente, com o mesmo objetivo, sob a coordenação da NBS (National Bureau of Standards), iniciou-se o desenvolvimento de uma especificação de arquivo neutro chamada IGES (Initial Graphics Exchange Specification). A primeira versão de IGES, publicada em janeiro de 1980, inclui dados para representação de curvas, superfícies, dimensões e anotações tipicamente encontradas em desenhos de engenharia. Além disso, podem ser representadas associações lógicas entre entidades, entidades voltadas à visualização e utilizada uma linguagem para definição de macros. O documento especifica transferência de dados por meio de arquivos ASCII com registros de 80 caracteres /LIE 85/. Em 1981 o comitê ANSI Y14.26 resolveu adotar IGES como parte do padrão para troca de dados de produtos. Este padrão se divide em cinco partes, sendo as quatro primeiras baseadas em IGES e a quinta na especificação feita pela McAuto para sólidos.

Com a rápida adoção, pelo mercado, de IGES como um padrão de fato, foi criado na NBS um comitê permanente para manutenção, difusão e adequação do padrão à evolução dos sistemas CAD/CAM. Como resultado deste trabalho, foi publicada em 1982 a versão 2.0 de IGES. Esta teve como principal preocupação estender IGES a outras áreas de aplicação, uma vez que a primeira versão era inteiramente voltada a aplicações mecânicas. Sendo assim, foram introduzidas entidades voltadas à representação de modelos de elementos finitos e placas de circuito impresso, tanto em seus aspectos lógicos (esquemático), quanto físicos. Outras adições importantes foram as seguintes /OWE 87/:

- criação de duas novas entidades geométricas: curva B-Spline racional e superfície B-Spline racional;
- especificação de um formato binário para o arquivo neutro.

Em abril de 1986 foi publicada a versão 3.0 de IGES. Nesta versão são introduzidas entidades para atender às aplicações em projetos industriais e AEC (Arquitetura, Engenharia e Construções). Além disso, é especificado um mecanismo para referência a arquivos externos, a fim de que possam ser utilizadas bibliotecas de símbolos ou componentes. São também feitas extensões à linguagem de definição de macros a fim de que possam ser representados modelos parametrizados. Com a intenção de reduzir o tamanho dos arquivos IGES é especificado o formato ASCII comprimido.

Modelos Sólidos em IGES

Em 1981, após constatar que a especificação feita pela McAuto para transferência de dados entre sistemas baseados em modelos sólidos não atendia a todos os requisitos, a entidade CAM-I decidiu revê-la. Nesta revisão, além do conteúdo de várias entidades, foi alterado o formato do arquivo para transferência, a fim de que este fosse compatível com aquele utilizado na versão corrente de IGES (versão 1.0). Deste trabalho surgiu um documento

denominado XBF ("Experimental Boundary File"), que foi submetido a uma revisão em 1982 /WIL 85a/.

Ao mesmo tempo em que se procedia à especificação de XBF, o comitê da NBS responsável por IGES empreendia esforços para inclusão de entidades voltadas à representação de modelos sólidos em IGES. Após um período de cooperação entre este comitê e CAM-I, surgiu uma especificação para modelos construtivos (CSG), por fronteiras (B-REP) e montagens. Esta foi denominada IGES-ESP ("Experimental Solids Proposal"). Testes foram realizados para transferência entre sistemas baseados em modelos construtivos, os quais apresentaram resultados que comprovaram a adequação da especificação para estes modelos. No caso de representação por fronteiras não se tem notícias de testes. A versão 4.0 de IGES, publicada em junho de 1988, apresenta como principal modificação a inclusão das partes de ESP relativas a modelos construtivos e montagens (estrutura do produto).

4.1.2. PADRÕES DE AMBITO NACIONAL

Embora IGES tenha sido desenvolvido nos Estados Unidos e seja um padrão nacional, uma vez que em 1987 a ANSI adotou a versão 3.0, tem sido utilizado em larga escala em todo o mundo para transferência de dados entre sistemas CAD. Porém, em âmbito nacional, nos Estados Unidos, Alemanha e França, foram desenvolvidas outras especificações com o intuito de sanar deficiências que a utilização de IGES demonstrou. Dentre estas especificações, as mais relevantes são /OWE 87/, /ENC 86/:

- SET (Standard d'Exchange et de Transfert):

Foi desenvolvido pela empresa francesa Aerospatiale, principalmente, por causa do tamanho excessivo dos arquivos IGES e porque esta especificação não atendia a todos os requisitos impostos pelos sistemas CAD por ela utilizados. SET utiliza o mesmo modelo de dados de IGES, porém seu formato é diferente, permitindo compartilhamento de dados entre registros e redução

correspondente no tamanho dos arquivos. O primeiro documento contendo a especificação foi publicado em 1983, sendo revisto em 1984. SET tem sido utilizado, principalmente, pela indústria aeronáutica europeia e pela indústria automobilística francesa, tendo sido adotado como um padrão nacional francês.

- VDA-FS (Verband der Automobilindustrieflächenschnittstelle):

É usado para transferir dados de superfície de dupla curvatura entre sistemas utilizados por fabricantes de veículos alemães. Esta especificação visa corrigir deficiências de IGES quanto à transferência de dados de superfícies, principalmente, relativas a problemas numéricos. As cinco entidades geométricas que usa podem ser diretamente mapeadas em IGES. VDA-FS foi publicada como padrão nacional alemão (norma DIN).

- VDA-IS:

Define uma série de subconjuntos de IGES que são usados para transferir dados entre sistemas utilizados por membros da Associação Alemã de Fabricantes de Veículos.

- VDA-PS:

Também utilizado na Alemanha, é um formato neutro para descrição da geometria de peças padronizadas.

- PDDI (Product Definition Data Interface):

Projeto patrocinado pela USAF (Força Aérea dos Estados Unidos), sendo executado pela empresa McDonnell Aircraft (McAir). Este projeto teve como objetivo definir e implementar uma interface de dados entre as funções de projeto e manufatura /ELG 85/. Esta interface é baseada em um modelo de dados para a completa definição de um produto. O projeto, iniciado em 1982, consistiu em duas fases principais. A primeira foi verificar a

utilidade da versão 1.0 de IGES para transferência de dados entre sistemas CAD e foi completada em 1983. A segunda fase tinha por objetivo determinar as informações requeridas pelas funções de manufatura para, junto àquelas de projeto, definir o produto. Concluiu-se serem necessárias representações de geometria, topologia, tolerâncias, formas características e informação de controle do produto.

Além do formato para troca de dados, o projeto desenvolveu uma especificação de interface para base de dados de tal forma que pudesse ser acessada por aplicações. A Fig. 4.3 mostra a arquitetura do protótipo PDDI. O módulo de acesso ao modelo permite que aplicações operem sobre o modelo de definição do produto, criando, removendo ou associando entidades. Foram também implementados pré e pós-processadores para tradução entre o formato de troca de dados e o modelo do produto.

O projeto PDDI teve grande importância, por ser a primeira especificação visando a completa definição de um produto.

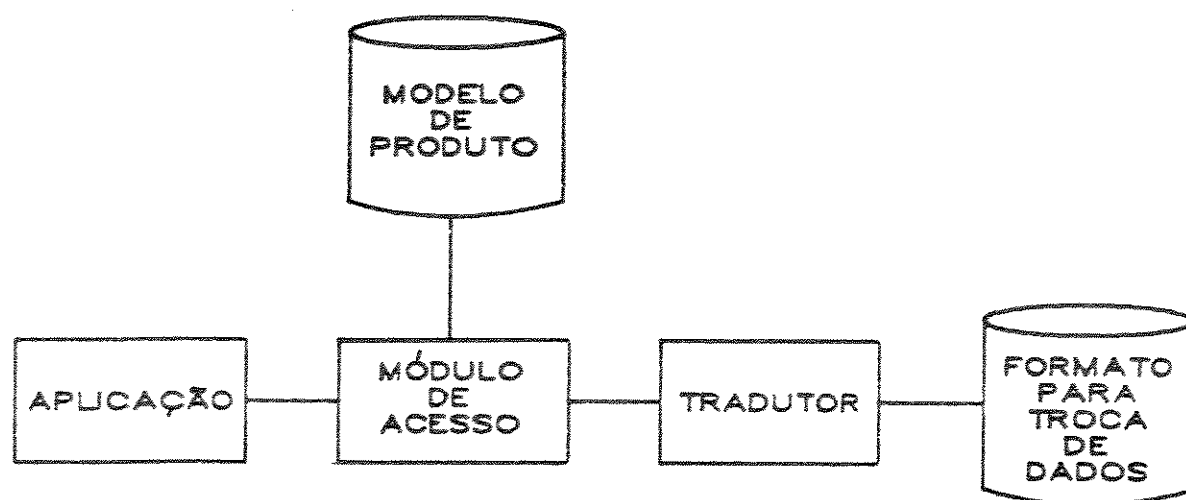


Fig. 4.3 - Arquitetura PDDI

- EDIF (Electronic Design Interchange Format):

Foi desenvolvido por 6 companhias americanas do ramo de eletrônica, independentemente de todos os outros padrões, visando a troca de dados entre sistemas CAD voltados a projetos eletrônicos.

4.1.3. Esforços Atuais de Padronização

A necessidade de suprir informações requeridas por aplicações automatizadas, características de um ambiente CIM, tais como geração automática de programas CNC e planejamento de processos generativo, implica em modelos de dados que representem não só informações geométricas, mas, também, todas aquelas necessárias à manufatura de um produto /SMI 87/. Sendo assim, atualmente, buscam-se padrões para transferência de modelos de produto entre os diversos sistemas componentes de um ambiente CIM.

Neste contexto, em 1983, foi instituído no âmbito da ISO (International Standards Organization) o subcomitê 4 do comitê 184 (Automação Industrial) com o objetivo de especificar um padrão único internacional para troca de dados de modelos de produto. O padrão a ser proposto foi denominado STEP (Standard for the Exchange of Product Model Data). O trabalho tem se desenvolvido tomando-se como base, inicialmente, IGES, SET, PDDI e VDA-FS. Posteriormente, as duas seguintes especificações tornaram-se principais referências para STEP:

- CAD*I:

É um projeto empreendido no âmbito do projeto ESPRIT ("European Strategic Program for Research in Information Technology"), cujo objetivo é desenvolver interfaces padronizadas que permitam a troca de informações entre sistemas CAD/CAE/CAM /BEY 88/, /SCH 87b/, /TRO 88/. Estão envolvidas doze instituições de seis países europeus. Como resultado deste trabalho foi

publicada em 1987 uma especificação de arquivo neutro para troca de informações geométricas entre os sistemas /SCH 87a/. Esta inclui modelos de arame, superfície e sólidos, sendo para estes especificados modelos construtivos e por fronteira.

Como mostra a Fig. 4.4, podem ser distinguidos três níveis para o relacionamento entre a especificação CAD*I e a base de dados de um sistema CAD qualquer. Para o usuário, interfaces realizam a tradução entre a representação dos dados do produto em um sistema e um arquivo sequencial. Em um segundo nível, o documento CAD*I especifica formalmente a sintaxe do arquivo neutro usando notação BNF, a semântica do arquivo neutro através de uma máquina de estados finitos para o pós-processador e um esquema de referência para as bases de dados de sistemas CAD. Para a especificação formal deste esquema foi desenvolvida a linguagem HDLS ("High Level Data Specification Language"). A nível de implementação, o arquivo neutro deve estar em conformidade com a gramática definida, o pós-processador deve estar em conformidade com a máquina de estados e as informações a serem transferidas via arquivo neutro devem poder ser mapeadas para o esquema de referência especificado.

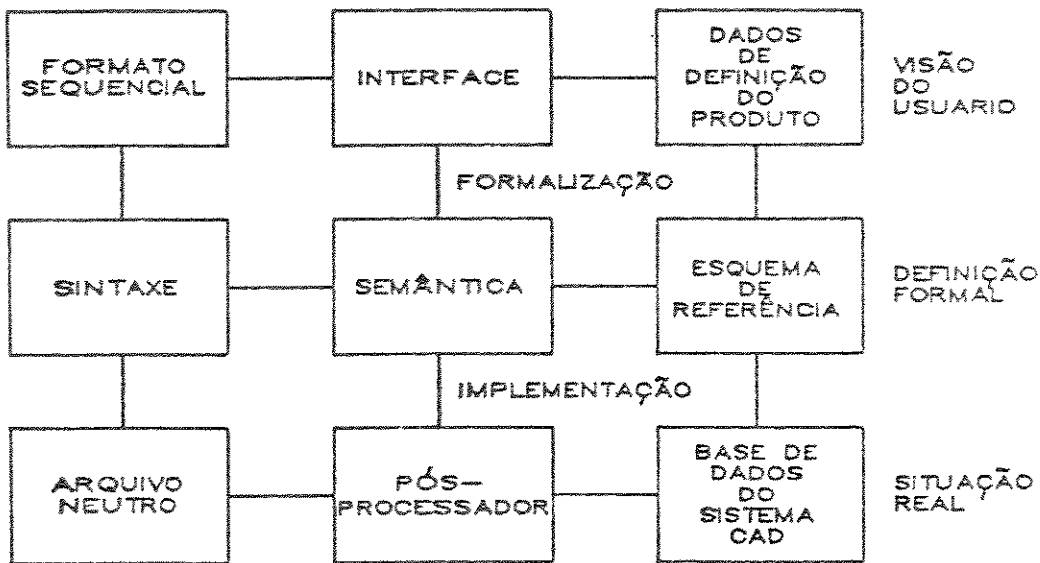


Fig. 4.4 - Relacionamento entre sistemas CAD e CAD*I

Além da máquina de estados que define regras para interpretação do arquivo neutro pelo pós-processador, a semântica é especificada informalmente sob outros dois aspectos:

- A nível da estrutura de dados do esquema de referência, são definidos os efeitos que operações elementares (criação, eliminação, identificação, etc.) causam sobre estas estruturas. Desta forma, pode-se validar o comportamento operacional de um modelo transferido observando-se os efeitos que estas operações causarão sobre ele. Este deverá se comportar da mesma forma que o modelo no sistema que envia, quando sujeito às mesmas operações. Sendo assim, garante-se que o modelo poderá continuar a ser utilizado pelo sistema que o recebe.
- Texto descritivo e representação gráfica definem a semântica relativa a cada entidade do modelo de referência.

- PDES (Product Data Exchange Specification):

Em 1984, uma vez constatado que IGES não supria as necessidades impostas por CIM, o comitê IGES da NBS iniciou um projeto com o objetivo de desenvolver e usar uma metodologia rigorosa para a criação de uma especificação para troca de dados de modelos de produto. Os conceitos utilizados neste projeto têm influenciado decisivamente o desenvolvimento de STEP. Tais conceitos serão tratados adiante neste trabalho (ver item 4.3).

A Fig. 4.5 sintetiza a evolução dos padrões para troca de dados de produtos e ilustra a influência de cada um no desenvolvimento de STEP. As seções a seguir detalharão os padrões de maior importância.

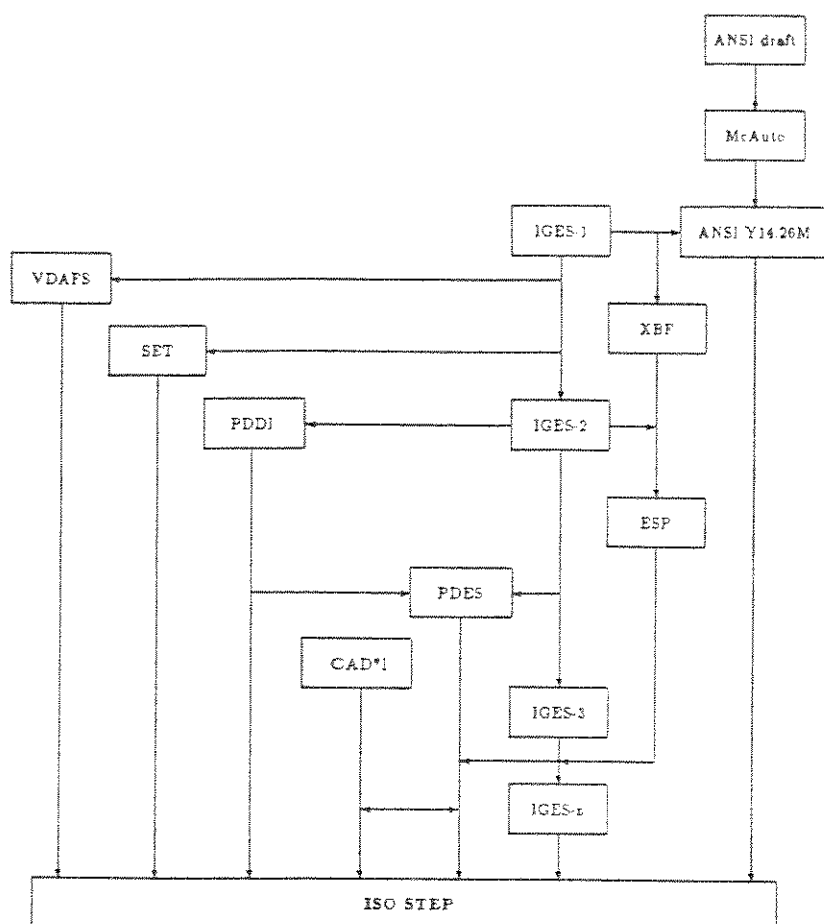


Fig. 4.5 - Evolução dos padrões para troca de dados de produtos

4.2. IGES

IGES (Initial Graphics Exchange Specification) tem por objetivo definir o formato e estrutura para representação digital de dados para definição de um produto. Como tal, tem sido o padrão mais amplamente utilizado para transferência de dados entre as funções de projeto e manufatura de produtos que fazem uso de computadores (sistemas CAD/CAM). IGES emprega o conceito de formato neutro (ver Fig. 4.2), sendo, portanto, requeridos, para cada sistema, um pré e um pós-processador para os mapeamentos entre a estrutura e formato do sistema e aquele especificado em IGES e vice-versa.

Nesta seção serão apresentados o formato e estruturas de dados que constam na versão 3.0, de abril de 1986 /IGE 86/. O desenvolvimento das sucessivas versões de IGES se dá de tal forma a garantir a compatibilidade com versões anteriores.

A versão 3.0 possibilita troca de dados entre sistemas voltados a, entre outras, as seguintes aplicações:

- sistemas para geração de desenhos de engenharia ("drafting systems");
- sistemas tridimensionais que utilizam modelos de arame e superfícies regradas e paramétricas;
- sistemas que utilizam modelos paramétricos;
- sistemas de modelagem por elementos finitos;
- sistemas para geração de programas para máquinas de comando numérico e para medição de coordenadas;
- sistemas para projeto de produtos eletro-eletrônicos;
- sistemas para projeto de instalações industriais (diagramas de processo e instrumentação).

Para suportar estas aplicações, IGES utiliza, como unidade de informação básica, entidades. Todos os dados de definição de produtos são expressos como uma lista de entidades. Como mostra a Fig. 4.6, as entidades podem ser divididas em geométricas e não-geométricas, sendo estas subdivididas em entidades de anotação e estruturais. Entidades geométricas representam a forma de um produto, incluindo, para isso, pontos, curvas e superfícies. Entidades de anotação são utilizadas para dimensionamento e detalhamento tipicamente encontrados em desenhos de engenharia. Entidades desta classe incluem, entre outras, dimensões, texto, definição de vistas, linhas de chamada e hachuras. As entidades estruturais permitem, além da associação de propriedades a outras entidades, a definição de esquemas para associação entre elas. A definição de uma associação pode residir em outro arquivo IGES.

Nesta classe está também compreendido um mecanismo para definição de entidades pelo usuário utilizando macros.



Fig. 4.6 - Conjunto de entidades representado em IGES

Cada entidade é representada em um formato independente da aplicação. Porém, como será observado posteriormente, várias entidades foram definidas a fim de que se tornasse possível a representação de modelos de dados referentes a aplicações específicas. Entidades voltadas a modelamento por elementos finitos são um exemplo do exposto acima. Além disso, a cada entidade é associado um número de forma, o que permite diferentes interpretações para uma mesma entidade.

4.2.1. Formato de Dados e Estrutura do Arquivo IGES

O modelo lógico definido pelas entidades em IGES precisa ser mapeado em um formato para troca de dados. Estão previstos na versão 3.0 dois formatos: ASCII e binário. Neste trabalho será apresentado o formato ASCII e a estrutura de arquivo correspondente. O formato binário utiliza a mesma estrutura de arquivo, diferindo, unicamente, na codificação dos dados.

4.2.1.1. Formato ASCII

Estão definidos em IGES dois tipos de formato ASCII:

- formato com registro fixo de 80 caracteres;
- formato ASCII comprimido.

A especificação define a representação de cinco tipos de constantes:

- inteiros;
- reais (ponto flutuante);
- cadeias de caracteres;
- ponteiros: várias entidades incluem entre seus parâmetros ponteiros para outras entidades. Em um arquivo IGES estes ponteiros são representados pelo número de sequência no arquivo em que é definida a entidade referenciada;
- comandos de linguagem: é especificada uma linguagem para definição de macros. Portanto, faz-se necessária uma forma de representação para os comandos desta linguagem (por exemplo, SET, LET). Além disso, são definidas regras para formação e interpretação de dados em formato livre, tais como aquelas relativas a separação de parâmetros em uma entidade e separação de entidades em uma seção do arquivo.

4.2.1.2. Estrutura do Arquivo IGES

Um arquivo IGES no formato ASCII com registro fixo é composto de seis seções na seguinte ordem (Fig. 4.7):

SECAO INDICADOR
SECAO INICIAL
SECAO GLOBAL
SECAO ENTRADA DE DIRETORIO
SECAO PARAMETROS
SECAO FINAL

Fig. 4.7 - Estrutura do arquivo IGES

A mesma estrutura é utilizada no formato binário. No formato ASCII comprimido as seções entrada de diretório e parâmetros são combinadas em uma única seção de dados, a fim de reduzir o tamanho dos arquivos gerados.

Nesta seção será detalhada a estrutura para o formato com registro de 80 caracteres, pois este, além de ser o mais utilizado, é o que consta na especificação TOP (ver Capítulo 5). Será utilizado o termo coluna para a posição do caracter na linha. Todas as linhas do arquivo conterão na coluna 73 uma letra que identifica a seção. Estas letras são as seguintes:

seção indicador - B ou C
seção inicial - S
seção global - G
seção entrada de diretório - D
seção parâmetros - P
seção final - T.

As colunas 74 a 80 são utilizadas para representar o número de sequência da linha em cada seção. Estes números, além de ordenarem os registros em um arquivo IGES, são usados para conectar fisicamente registros relacionados da seção entrada de diretório e da seção parâmetros. Além disso, todos os relacionamentos lógicos (ponteiros) entre entidades em IGES são expressos

através do número de sequência do registro da seção entrada de diretório da entidade referenciada /GRA 87/.

Em cada seção de um arquivo são representadas diferentes informações:

a) Seção Indicador:

É utilizada, quando presente, para indicar que o arquivo está em formato binário ou ASCII comprimido. No caso de formato com registro fixo não é utilizada.

b) Seção Inicial:

É utilizada para uma descrição não formatada do arquivo. Pode ser utilizada para indicar características gerais do arquivo, tais como, tipo de informação que contém, sua distribuição por níveis, etc. A Fig. 4.8 mostra um exemplo de seção inicial de um arquivo IGES.

1	72	73	80
NESTA SECAO PODE SER INSERIDO QUALQUER TEXTO USANDO ASCII	S0000001		
	S0000002		
	:		
	S000000N		
FIM DA SECAO INICIAL			

Fig. 4.8 - Exemplo de seção inicial

c) Seção Global:

Contém informação que descreve o pré- processador que gerou o arquivo e aquela requerida pelo pós- processador para manipular o arquivo IGES. No total, são 24 parâmetros na ordem e com o conteúdo mostrado na Tabela 4.1.

INDICE	TIPO DE DADO	DESCRIÇÃO
1	Cadeia de caracteres	Caracter delimitador de parâmetros
2	Cadeia de caracteres	Caracter delimitador de conjunto de parâmetros de entidade
3	Cadeia de caracteres	Identificação do produto no sistema que gerou o arquivo
4	Cadeia de caracteres	Nome do arquivo
5	Cadeia de caracteres	Código de identificação do sistema gerador
6	Cadeia de caracteres	Versão do pré-processador
7	Inteiro	Número de bits p/ representação de inteiros
8	Inteiro	Máxima potência de dez representável precisão simples
9	Inteiro	Número de dígitos significativos em precisão simples
10	Inteiro	Máxima potência de dez representável precisão dupla
11	Inteiro	Número dígitos significativos em precisão dupla
12	Cadeia de caracteres	Identificação do produto no sistema receptor
13	Real	Escala do espaço global
14	Inteiro	Unidade de medida
15	Cadeia de caracteres	Nome padronizado da unidade de medida
15	Inteiro	Número máximo de gradações de espessura de linha
17	Real	Tamanho da espessura máxima de linha
18	Cadeia de caracteres	Data e hora da geração do arquivo
19	Real	Resolução do arquivo
20	Real	Máxima coordenada
21	Cadeia de caracteres	Nome do autor
22	Cadeia de caracteres	Organização responsável
23	Inteiro	Número da versão de IGES
24	Inteiro	Padrão utilizado para notação de engenharia

TABELA 4.1 - Parâmetros de seção global

d) Seção Entrada de Diretório:

Contém uma entrada de diretório para cada entidade no arquivo. Cada entrada de diretório é representada por dois registros de 80 caracteres. Em cada registro são definidos 10 campos de 8 caracteres. Neles são especificados os atributos da entidade que constam na Fig. 4.9 e serão descritos a seguir.

1	8	9	16	17	24	25	32	33	40	41	48	49	56	57	64	65	72	73	80	
	NÚMERO DE TIPO DA ENTIDADE		PARAMETROS		ESTRUTURA		PADRÃO DE FONTE DE LINHA		NÍVEL		VISTA		MATRIZ DE TRANSFORMAÇÃO		ASSOCIAÇÃO DE EXIBIÇÃO DE RÓTULO		NÚMERO DE ESTADO		NÚMERO DE SEQUENCIA	
	#		++	#, ++			#, ++		0, ++		0, ++		0, ++		++		#		D#	
	1		2	3			4		5		6		7		8		9		10	
	NÚMERO DE TIPO DA ENTIDADE		ESPESSURA DE LINHA		COR		CONTADOR DE LINHAS DE PARAMETROS		NÚMERO DE FORMA		NÃO USADO		NÃO USADO		RÓTULO DE ENTIDADE		SUBSCRITO DE ENTIDADE		NÚMERO DE SEQUENCIA	
	#		#	#, ++			#		#								#		D#+1	
	11		12	13			14		15		16		17		18		19			

- INTEIRO
++ - PONTEIRO
#, ++ - INTEIRO OU PONTEIRO (PONTEIRO TEM SINAL NEGATIVO)
0 - ZERO

Fig. 4.0 - Entrada diretório para uma entidade

- Campos 1 e 11: são redundantes. Identificam o tipo de entidade descrito nesta entrada de diretório. A cada tipo de entidade em IGES é associado um único número.
- Campo 2: ponteiro para o primeiro registro da entidade na seção parâmetros.
- Campo 3: ponteiro para a entrada de diretório da entidade que especifica o esquema para a entidade considerada. Nem todas as entidades utilizam este campo.
- Campo 4: indica o padrão de linha a ser usado para visualização da entidade. Entre outras, podem ser utilizadas linhas sólidas e tracejadas.
- Campo 5: identifica o nível a ser associado à entidade para representação gráfica. Os sistemas CAD normalmente associam um nível às entidades para possibilitar operações seletivas sobre estas.
- Campo 6: ponteiro para indicar a visibilidade da entidade em uma vista, várias vistas ou que ela é visível em todas as vistas. Neste último caso o campo conterá o valor zero.
- Campo 7: todas as entidades são descritas em um sistema de coordenadas local. Para obter sua representação em coordenadas globais utiliza-se uma matriz de transformação. Este campo contém um ponteiro para uma entidade matriz de transformação ou zero no caso desta ser a matriz identidade.
- Campo 8: contém um ponteiro para a entrada de diretório de uma entidade que define parâmetros para a visualização do rótulo da entidade em questão.
- Campo 9: este campo é subdividido em quatro itens, cada um contendo dois caracteres. Os itens representam as seguintes informações:

- primeiro item: define se a entidade será visível ou não em um dispositivo de saída do sistema receptor;
 - segundo item: indica se a entidade é referenciada ou não por outras entidades do arquivo. Se for referenciada é definido o tipo de relacionamento existente (fisicamente dependente, logicamente dependente, ou ambos);
 - terceiro item: define o uso da entidade (geométrica, anotação, estrutural);
 - quarto item: indica se atributos como fonte de linha, nível, vista, são aplicados a entidades fisicamente subordinadas à entidade em questão.
-
- Campos 10 e 20: número que especifica a posição desta entrada de diretório na seção. A coluna 73 conterá a letra D.
 - Campo 12: indica a espessura da linha utilizada para visualização da entidade.
 - Campo 13: indica número de cor pré-definida ou um ponteiro para uma entidade que define uma nova cor.
 - Campo 14: indica o número de registro utilizado na seção parâmetros para descrição da entidade.
 - Campo 15: é utilizado para diferir várias interpretações de uma mesma entidade.
 - Campo 18: especifica um nome para a entidade no contexto de uma aplicação.
 - Campo 19: indica um número para a entidade no contexto de uma aplicação.

Uma restrição imposta nesta seção é que a definição de uma entidade deve preceder suas instâncias.

e) Seção Parâmetros:

Esta seção contém os parâmetros que definem as entidades, tais como, coordenadas, dimensões, ponteiros para outras entidades. Para cada entidade os dados são escritos em formato livre, utilizando o delimitador de parâmetros especificado na seção global. O primeiro parâmetro é sempre o número de tipo da entidade. As colunas 66 a 72 de cada registro nesta seção representam o número de sequência do primeiro registro na seção entrada de diretório da entidade para a qual os parâmetros estão sendo representados. A coluna 73 contém sempre a letra P, sendo as colunas 74 a 80 utilizadas para representar o número de sequência do registro na seção.

De grande importância para o modelo lógico que IGES define, visando sua utilização como meio de transferência de dados entre diversas aplicações, são dois grupos de parâmetros que aparecem após aqueles específicos para cada entidade. O primeiro grupo pode conter ponteiros para instâncias de associações, notas gerais (textos) e/ou entidades que definem gabaritos de textos. O segundo grupo pode conter ponteiros para entidades que especifica propriedades a serem associadas à entidade sendo definida. O número de ponteiros para cada grupo é indicado antes destes, podendo não haver ponteiros para um ou ambos os grupos.

f) Seção Final:

Esta seção contém somente um registro. Este é o último do arquivo, contendo 10 campos de 8 colunas. Os campos 1, 2, 3 e 4 contém um caracter representando o tipo de seção e o número de registros utilizados em cada uma das seções. Os campos 5 a 9 não são utilizados. A coluna 73 contém a letra T e as colunas 74 a 80 o número de sequência, ou seja, o número 1.

Para encerrar o tópico relativo a formato de dados e estrutura de arquivo, a Fig. 4.10 mostra a representação de um arco circular e um ponto nas seções entrada de diretório e parâmetros /GRA 87/.

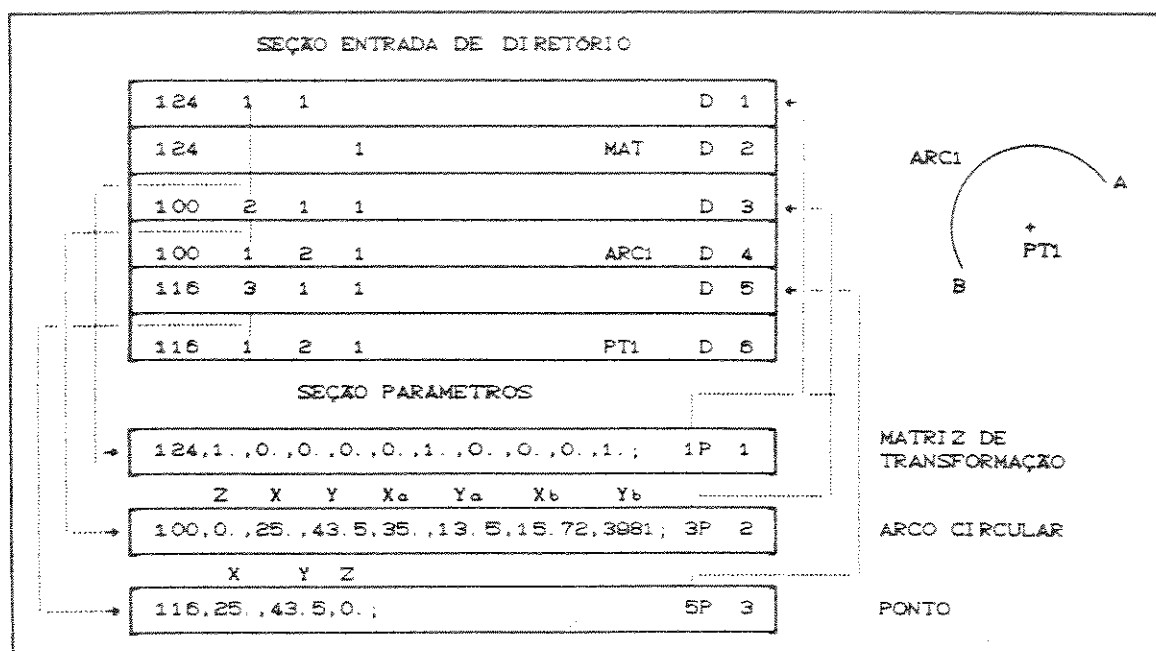


Fig. 4.10 - Representação de um arco e um ponto em IGES

4.2.2. Entidades Geométricas

Nesta seção serão apresentadas as entidades que constituem o modelo geométrico IGES.

A cada entidade geométrica está associado um sistema de coordenadas local (espaço local), em que esta é definida. Por outro lado, existe um sistema de coordenadas, denominado global (espaço global), que é único para todo o modelo. A transformação do sistema local para o global é feita através da matriz de transformação que é definida no campo 7 da entrada de diretório da entidade. No caso da entidade já ser definida no espaço global, este campo conterá o valor zero. Convenciona-se que ambos os sistemas de coordenadas são cartesianos e destrógiros.

A matriz de transformação é composta de uma matriz de rotação e um vetor de translação, de tal forma que a transformação entre os sistemas de coordenadas se dá pela seguinte equação:

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} X_{\text{LOCAL}} \\ Y_{\text{LOCAL}} \\ Z_{\text{LOCAL}} \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} X_{\text{GLOBAL}} \\ Y_{\text{GLOBAL}} \\ Z_{\text{GLOBAL}} \end{bmatrix}$$

MATRIZ DE ROTACAO
VECTOR DE TRANSLACAO

Outra convenção feita é que o sentido de rotação positivo é o anti-horário quando visto da direção positiva do eixo z.

Os números de tipo de entidade de 100 a 199 são reservados para entidades geométricas. A Tabela 4.2 apresenta as entidades, seu número de tipo e uma descrição resumida e/ou observações pertinentes.

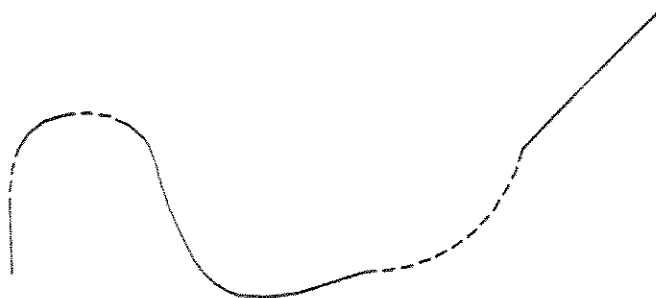


Fig. 4.11 - Entidade curva composta

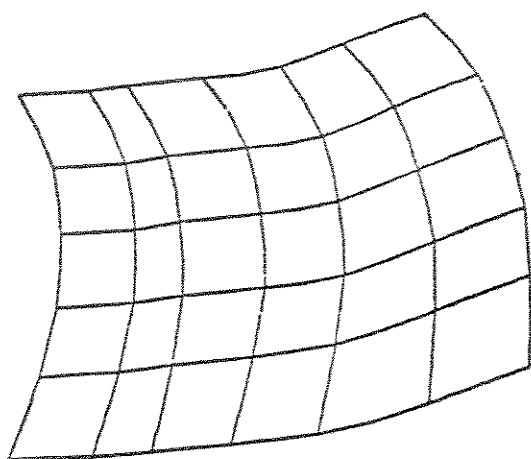


Fig.4.12 - Superfície spline paramétrica

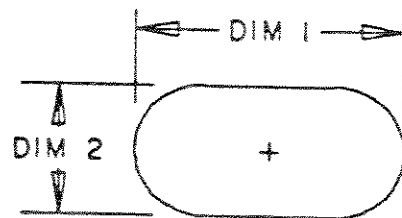


Fig. 4.13 - Entidade flash

ENTIDADE	NÚMERO DE TIPO	DESCRIÇÃO
Arco Circular	100	Arco definido no plano X-Y do sistema de coordenada local
Curva Composta	102	Curva conectada constituída de curvas elementares dos tipos: ponto, linha, arco circular, arco cônico, etc. (ver Fig. 4.11). A seção parâmetros contém o número de entidades elementares e um ponteiro para a entrada de diretório de cada entidade
Arco Cônico	104	Elipse, parábola ou hipérbole. Número de forma diferencia entre elas.
Dados Numerosos	106	Não representa forma geométrica. É usada para armazenar conjuntos de pontos na forma de pares, triplas ou sêxtuplas
Plano	108	Definido pelos coeficientes A,B,C,D da equação $Ax+By+Cz = D$
Linha	110	Definida por dois pontos.
Curva Spline Paramétrica	112	Definida por uma sequência de segmentos polinomiais paramétricos
Superfície Spline Paramétrica	114	Extensão da entidade anterior para o caso de duas direções paramétricas (ver Fig. 4.12).
Ponto	116	Pode ser especificado um ponteiro para uma entidade utilizada para visualizar o ponto.
Superfície Regrada	118	Formada movendo-se uma linha que conecta pontos da mesma coordenada paramétrica entre duas curvas (ver Fig. 3.9).
Superfície de Revolução	120	Definida pela rotação de uma curva geratriz em torno de um eixo
Cilindro Tabulado	122	Superfície formada movendo-se uma linha ao longo de uma curva (ver Fig. 3.10)
Matriz de Transformação	124	Ver introdução deste item.
Flash	125	Localiza uma instância de uma área fechada. É utilizada para fotoplotagem de circuitos eletrônicos (ver Fig. 4.13).
Curva B-spline Racional	126	Ver item 3.3.1.2.
Superfície B-spline Racional	128	Ver item 3.3.1.2.
Curva offset	130	Contém os dados necessários à determinação de uma curva que mantém uma distância em relação a outra que é dada por uma função. Distância constante ou variando linearmente são casos particulares
Superfície offset	140	Extensão da entidade anterior ao caso de superfície. A função é restrita a distância constante entre as superfícies.
Nó	134	Utilizada em malha de elementos finitos.
Elemento Finito	135	Utilizada em malha de elementos finitos.
Rotação e Deslocamento Nodal	138	Utilizada para representar resultados de pós-processamento de elementos finitos.
Ponto de Conexão	132	Descreve um ponto de conexão para zero, uma ou mais entidades. É utilizada na representação de circuitos eletro-eletrônicos e diagramas de instrumentação e processo.
Curva em uma Superfície Paramétrica	142	Associa uma curva a uma superfície. A curva precisa estar contida na superfície.
Superfície Paramétrica Limitada	144	Permite a definição de regiões em uma superfície paramétrica.

TABELA 4.2 Entidades geométricas em IGES

4.2.3. Entidades de Anotação

A maior parte das entidades de anotação utiliza, para sua definição, ponteiros para outras entidades. As entidades de anotação podem ser definidas em coordenadas do espaço local ou em coordenadas bidimensionais associadas com uma entidade desenho (número de tipo 404) que especifica parâmetros para a representação de um desenho de engenharia. A Tabela 4.3 apresenta as entidades, seu número de tipo e uma descrição resumida e/ou observações pertinentes. Além destas, a entidade geométrica dados numerosos é utilizada para representar linhas de chamada e linhas de centro.

ENTIDADE	NÚMERO DE TIPO	DESCRIÇÃO
Nota Geral	212	Consiste em uma ou mais cadeias de caracteres (texto). Para cada cadeia de caracteres são especificados, na seção parâmetros, seu conteúdo, um ponto de referência para posicionamento, tamanho do texto e número do fonte de caracteres usado.
Cota	214	Consiste em um ou mais segmentos de linha, sendo o primeiro segmento iniciado com uma seta. No caso de dimensões angulares os segmentos de linhas são substituídos por segmentos de arcos circulares (ver Fig. 4.14). O número de forma define o padrão para a seta a ser usada.
Rótulo Geral	210	Consiste em uma nota geral e uma ou mais cotas. Seus parâmetros são ponteiros para entidades dos tipos descritos acima (ver Fig. 4.15).
Nota de Indicador	208	Define um padrão para indicador utilizado em desenhos de engenharia. Pode ter cotas associadas (ver Fig. 4.16).
Símbolo Geral	228	Consiste em uma cota geral, uma ou mais entidades geométricas e zero ou mais cotas associadas. Permite a definição de símbolos utilizados em desenhos de engenharia (ver Fig. 4.17).
Área Seccionada	230	Usada para representar seções transversais hachuradas. Seus parâmetros contêm um ponteiro para uma curva fechada e o padrão para hachura (ver Fig. 4.18).
Dimensão Angular	202	Representa ângulos de forma normalmente encontrada em desenhos de engenharia (ver Fig. 4.19).
Dimensão de Diâmetro	206	Idem para diâmetros (ver Fig. 4.20).
Dimensão de Raio	222	Idem para raios (ver Fig. 4.21).
Dimensão Linear	216	Idem para segmentos retos (ver Fig. 4.22).
Dimensão Ordenada	218	Indica uma sequência de dimensões com base comum.
Dimensão Ponto	220	Representa em um desenho um dado associado a um ponto.

TABELA 4.3 - Entidades de anotação em IGES



Fig. 4.14 - Entidade cota

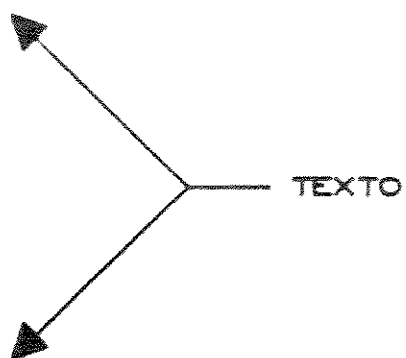


Fig. 4.15 - Entidade rótulo geral

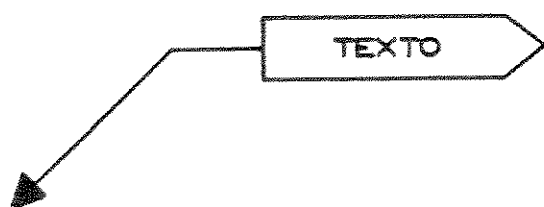


Fig. 4.16 - Entidade nota de indicador



Fig. 4.17 - Entidade símbolo geral

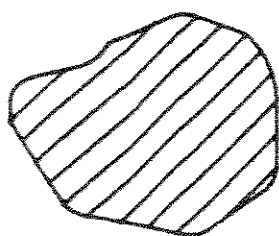


Fig. 4.18 - Entidade área seccionada

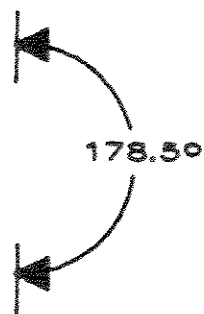


Fig. 4.19 - Entidade dimensão angular

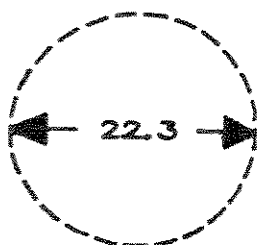


Fig. 4.20 - Entidade dimensão de diâmetro



Fig. 4.21 - Entidade dimensão de raio

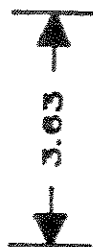


Fig. 4.22 - Entidade dimensão linear

4.2.4. Entidades Estruturais

Como foi citado anteriormente, IGES tem por objetivo permitir não só a representação dos dados necessários à geração de desenhos de engenharia (entidades geométricas e de anotação), mas, também, os modelos de dados utilizados nas diversas aplicações que se pretende suportar. As entidades estruturais visam suprir esta flexibilidade requerida quanto aos esquemas de representação, provendo mecanismos para estabelecer relacionamentos entre entidades, especificação de propriedades, acesso a entidades definidas em outros arquivos, definição de entidades pelo usuário e especificação de parâmetros para composição de vistas em desenhos de engenharia.

4.2.4.1. Associação

Entidades para associação permitem várias entidades serem logicamente relacionadas umas às outras. Para isso são especificadas duas entidades em IGES: uma para definição de associação e outra para instância de associação.

Entidade Definição de Associação:

- número de tipo da entidade: 302

- através desta entidade o pré-processador IGES define um esquema para o relacionamento lógico entre entidades. Este esquema define a sintaxe do relacionamento e, não, sua semântica. A definição do esquema de associação se baseia nos seguintes conceitos:

- classe: é uma coleção de itens, que podem ser valores ou ponteiros para outras entidades. O relacionamento se dá entre os itens de cada classe e, no caso de ser definida mais de uma classe pela entidade definição de associação, também, entre as várias classes. Para cada classe, os parâmetros da entidade contêm valores que indicam se os retro-apontadores das entidades apontadas pela associação sendo definida são requeridos ou não. Conforme citado anteriormente, no final dos parâmetros de cada entidade pode haver ponteiros para instâncias de associação. O referido parâmetro diz respeito a estes retro-apontadores. Além disso, há outro parâmetro que indica se a ordem das instâncias para cada classe é relevante ou não.
- entrada: é uma instância para cada classe. A entidade definição de associação fixa o número de classes e o número de itens por entrada para cada classe. Cada instância de associação terá um número variável de entradas por classe.

A definição de classes e itens para uma entidade definição de associação se dá através dos dados na seção parâmetros que constam na Tabela 4.4.

Os parâmetros 2 a $4+N1$ são repetidos para cada uma das K classes.

INDICE	NOME	TIPO	DESCRIZÃO
1	K	INTEIRO	NUMERO DE CLASSES
2	BP1	INTEIRO	1) RETRO-APONTADORES REQUERIDOS 2) RETRO-APONTADORES NAO REQUERIDOS
3	OR1	INTEIRO	1) CLASSE ORDENADA 2) CLASSE NAO ORDENADA
4	N1	INTEIRO	NUMERO DE ITENS POR ENTRADA
5	IT1(1)	INTEIRO	1) PONTEIRO PARA ENTRADA DE DIRETORIO DE UMA ENTIDADE 2) VALOR 3) PONTEIRO OU VALOR SE ≥ 0 - VALOR SE < 0 - PONTEIRO
4+N1	IT1(n1)	INTEIRO	MESMO ACIMA PARA O ULTIMO ITEM

TABELA 4.4 - Parâmetros para definição de associação

Entidade Instância de Associação:

- número de tipo da entidade: 402
- especifica os dados envolvidos em uma ocorrência particular de uma associação, cujo esquema é definido por uma entidade definição de associação. A forma geral dos parâmetros de uma instância de associação é especificada na Tabela 4.5 (os parâmetros K, N1...NK são definidos na definição de associação).

Para cada parâmetro o tipo é definido pelo parâmetro correspondente $ITn(m)$ da definição de associação.

INDICE	NOME	TIPO	DESCRIÇÃO
1 : :	NE1	INTEIRO	NUMERO DE ENTRADAS PARA A PRIMEIRA CLASSE
K	NEK	INTEIRO	NUMERO DE ENTRADAS PARA A K-ESIMA CLASSE
K+1	I(1,1,1) : :	VARIÁVEL	PRIMEIRO ITEM DA PRIMEIRA ENTRADA DA PRIMEIRA CLASSE
	I(1,1,N1) : :	VARIÁVEL	N1-ESIMO ITEM DA PRIMEIRA ENTRADA DE PRIMEIRA CLASSE
	I(1,NE1,1) : :	VARIÁVEL	PRIMEIRO ITEM DA NE1-ESIMA ENTRADA DA PRIMEIRA CLASSE
	I(1,NE1,N1) : :	VARIÁVEL	N1-ESIMO ITEM DA NE1-ESIMA ENTRADA DA PRIMEIRA CLASSE
	I(K,1,1) : :	VARIÁVEL	PRIMEIRO ITEM DA PRIMEIRA ENTRADA DA K-ESIMA CLASSE
	I(K,NEK,NK)	VARIÁVEL	NK-ESIMO ITEM DA NEK-ESIMA ENTRADA DA K-ESIMA CLASSE

TABELA 4.5 - Parâmetros para instância de associação

A identificação da associação a que se refere uma específica entidade instância de associação se dá através de seu número de forma na entrada de diretório. Se o número de forma estiver entre 5001 e 9999, é usada uma entidade definição de associação criada pelo pré-processador. Se estiver entre 1 e 5000, diz respeito a associações pré-definidas em IGES. A entidade instância de associação apresenta o mesmo comportamento para ambos os tipos de definição de associação. A diferença está no fato que um tipo é definido explicitamente no arquivo e outro implicitamente pelo documento IGES. Para ilustrar este tópico relativo a associações serão apresentadas algumas pré-definidas. Para cada uma será especificada a definição de associação de

forma similar à que seria necessária se ela fosse especificada explicitamente. Outros casos podem ser verificados em /IGE 86/.

1) Associação de Grupo:

- número de forma: 1
- permite que entidades sejam logicamente relacionadas.
- definição:

INDICE	VALOR	SIGNIFICADO
1	1	UMA CLASSE
2	1	RETRO-APONTADORES REQUERIDOS
3	2	NAO ORDENADA
4	1	UM ITEM POR ENTRADA
5	1	O ITEM É UM PONTEIRO

TABELA 4.6 - Definição de associação de grupo

- instância:

INDICE	NOME	TIPO	SIGNIFICADO
1	N	INTEIRO	NUMERO DE ENTRADAS
2	DE1	PONTEIRO	PONTEIRO PARA ENTIDADE 1
...			
N+1	1	PONTEIRO	PONTEIRO PARA ENTIDADE N

TABELA 4.7 - Instância de associação de grupo

2) Índice para referência de arquivos exteriores

- número de forma: 12
- aparece em um arquivo que contém definições de entidades referenciadas por arquivos externos. Consiste em uma lista de nomes simbólicos usados pelos arquivos que referenciam e ponteiros para as entidades referenciadas.
- definição:

INDICE	VALOR	SIGNIFICADO
1	1	UMA CLASSE
2	2	RETRO-APONTADORES REQUERIDOS
3	2	NAO ORDENADA
4	2	NUMERO DE ITENS EM UMA ENTRADA
5	2	PRIMEIRO ITEM É UM VALOR (NOME SIMBOLICO)
6	1	SEGUNDO ITEM É UM PONTEIRO

TABELA 4.8 - Definição de índice para referência de arquivos externos

- instância:

INDICE	NOME	TIPO	SIGNIFICADO
1	N	INTEIRO	NUMERO DE ENTRADAS
2	NOME1	CADEIA DE CARACTERES	NOME SIMBOLICO
3 : 2N	PTR1 NOMEN	PONTEIRO CADEIA DE CARACTERES	PONTEIRO PARA ENTIDADE NOME SIMBOLICO
2N+1	PTRN	PONTEIRO	PONTEIRO PARA ENTIDADE

TABELA 4.9 - Instância de índice para referência de arquivos externos

4.2.4.2. Subfiguras

A definição de subfiguras permite que a coleção de entidades seja manipulada como se fosse uma única entidade. Uma vez definida a subfigura, esta pode ser instanciada em várias localizações e orientações no espaço global. Subfiguras podem referenciar outras subfiguras, sendo o número de referências aninhadas representado pelo parâmetro profundidade da subfigura. A Fig. 4.23 mostra duas instâncias de uma subfigura, com profundidade N. A definição da subfigura consiste em duas

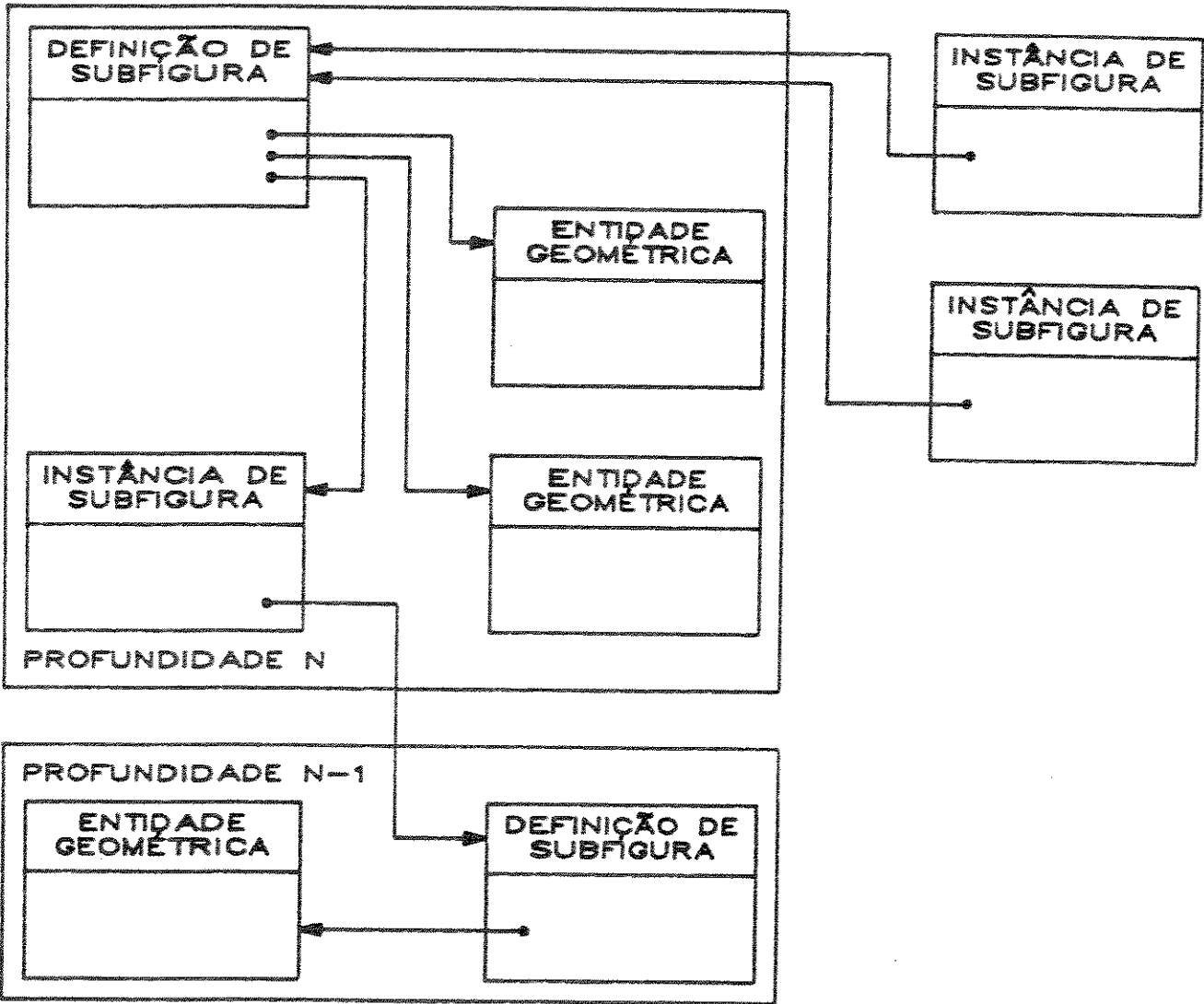


Fig. 4.23 - Esquema de subfigura em IGES

entidades geométricas e uma instância de subfigura. A definição de subfigura correspondente a esta instância tem profundidade N-1. A Tabela 4.10 apresenta as entidades que dizem respeito ao esquema de subfigura.

ENTIDADE	NUMERO DE TIPO	DESCRIÇÃO
DEFINICAO DE SUBFIGURA	308	CONTEM EM SEUS PARAMETROS A PROFUNDIDADE DA SUBFIGURA, UM NOME PARA ELA E PONTEIROS PARA AS ENTIDADES QUE A CONSTITUEM
INSTANCIA DE SUBFIGURA	408	CONTEM UM PONTEIRO PARA A ENTIDADE DEFINICAO DE SUBFIGURA CORRESPONDENTE E PARAMETROS PARA SUA LOCALIZACAO NO ESPACO GLOBAL
INSTANCIA DE MATRIZ RETANGULAR DE SUBFIGURA	412	PERMITE A DEFINICAO DE UMA MATRIZ EM QUE UMA SUBFIGURA SE REPETE EM UM DADO NUMERO DE LINHAS E COLUNAS
INSTANCIA DE MATRIZ CIRCULAR DE SUBFIGURA	414	PERMITE A DEFINICAO DE UMA MATRIZ EM QUE UMA SUBFIGURA SE REPETE CIRCULARMENTE EM TORNO DE UM PONTO
DEFINICAO DE SUBFIGURA DE REDE	320	DEFINE UMA SUBFIGURA ESPECIAL, CUJAS INSTANCIAS FAZEM PARTE DE REDES (CIRCUITOS, DIAGRAMAS DE INSTRUMENTACAO E PROCESSO). ALEM DOS PARAMETROS EXISTENTES NA ENTIDADE DEFINICAO DE SUBFIGURA, SAO ESPECIFICADOS PONTEIROS PARA ENTIDADES PONTO DE CONEXAO UTILIZADAS NA REDE
INSTANCIA DE SUBFIGURA DE REDE	420	CONTEM PARAMETROS PARA A LOCALIZACAO DA INSTANCIA DE SUBFIGURA DE REDE E UM PONTEIRO PARA A ENTIDADE QUE A DEFINE

TABELA 4.10 - Entidades do esquema de subfigura em IGES

4.2.4.3. Propriedades

É especificada uma entidade propriedade (número de tipo 406), que permite que dados numéricos ou textos sejam associados a entidades. Como foi observado anteriormente (item 4.2.1.2), as propriedades são referenciadas pelas entidades através de ponteiros que podem estar presentes no final dos parâmetros da entidade. Diversos números de forma da entidade propriedade estão associados a esquemas para suprir dados requeridos em diversas aplicações, tais como largura de pista (circuitos eletrônicos), matriz de rigidez (elementos finitos) e coeficientes térmicos.

4.2.4.4. Entidades para Composição de Desenhos de Engenharia

IGES especifica duas entidades utilizadas para composição de desenhos utilizando vistas em várias orientações (Fig. 4.24):

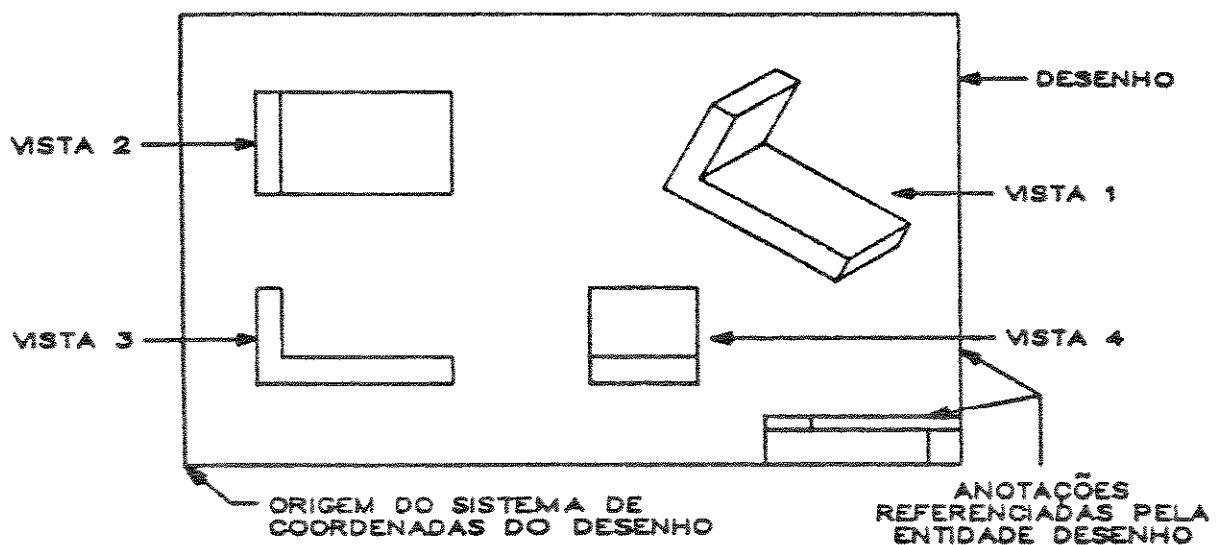


Fig. 4.24 - Exemplo de utilização da entidade desenho

Entidade Desenho:

- número de tipo da entidade: 404

- especifica um desenho como uma coleção de entidades de anotação e vistas (projeções de entidades geométricas em um plano de visualização segundo uma direção). Um mesmo arquivo IGES pode conter vários desenhos. A cada desenho é associado um sistema de coordenadas, no qual as entidades de anotação podem ser especificadas e que serve para localizar as vistas no desenho. Contém em seus parâmetros ponteiros para as entidades vista que o compõem, a localização de cada vista no desenho e ponteiros para entidades de anotação.

Entidade Vista:

- número de tipo da entidade: 410
- é usada para definir o plano de visualização em que uma figura será projetada. Isto é feito através da matriz de transformação que é referenciada na entrada de diretório da entidade. Além disso, contém em seus parâmetros ponteiros para planos que definem um volume de visualização para recorte ("clipping") (Fig. 4.25).

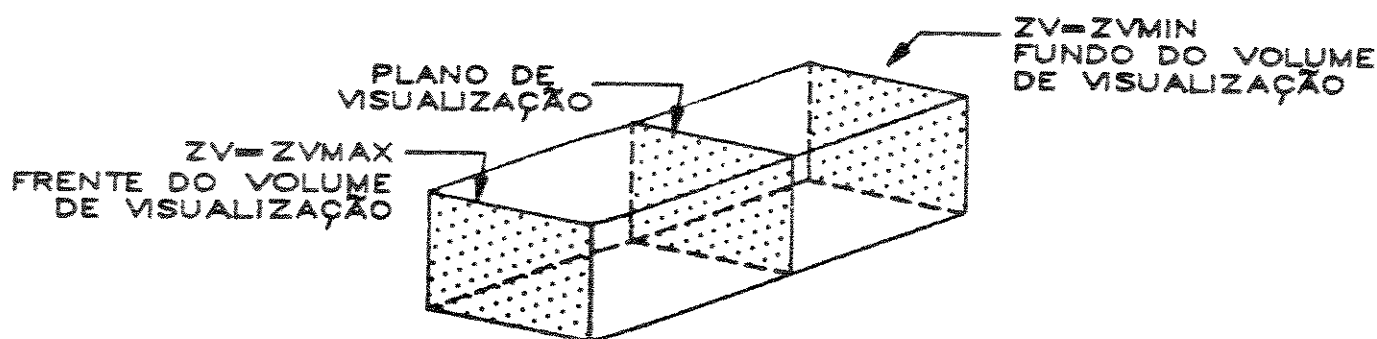


Fig. 4.25 - Volume de visualização

4.2.4.5. Definição de Entidades pelo Usuário

Além das entidades pré-especificadas, IGES provê um mecanismo para definição de novas entidades em termos de entidades básicas. Isto é feito através de uma linguagem para definição de macros. A entidade definição de macro (número de tipo 306) contém em seus parâmetros uma sequência de instruções de linguagem que definem a nova entidade e o número de tipo definido pelo usuário para a entidade sendo criada . Este será o número de tipo utilizado na entidade instância de macro, que corresponde a uma ocorrência particular da entidade definida. A Fig. 4.26 mostra duas instâncias de macro. O terceiro campo da entrada de diretório de uma instância contém um ponteiro para a definição de macro correspondente.

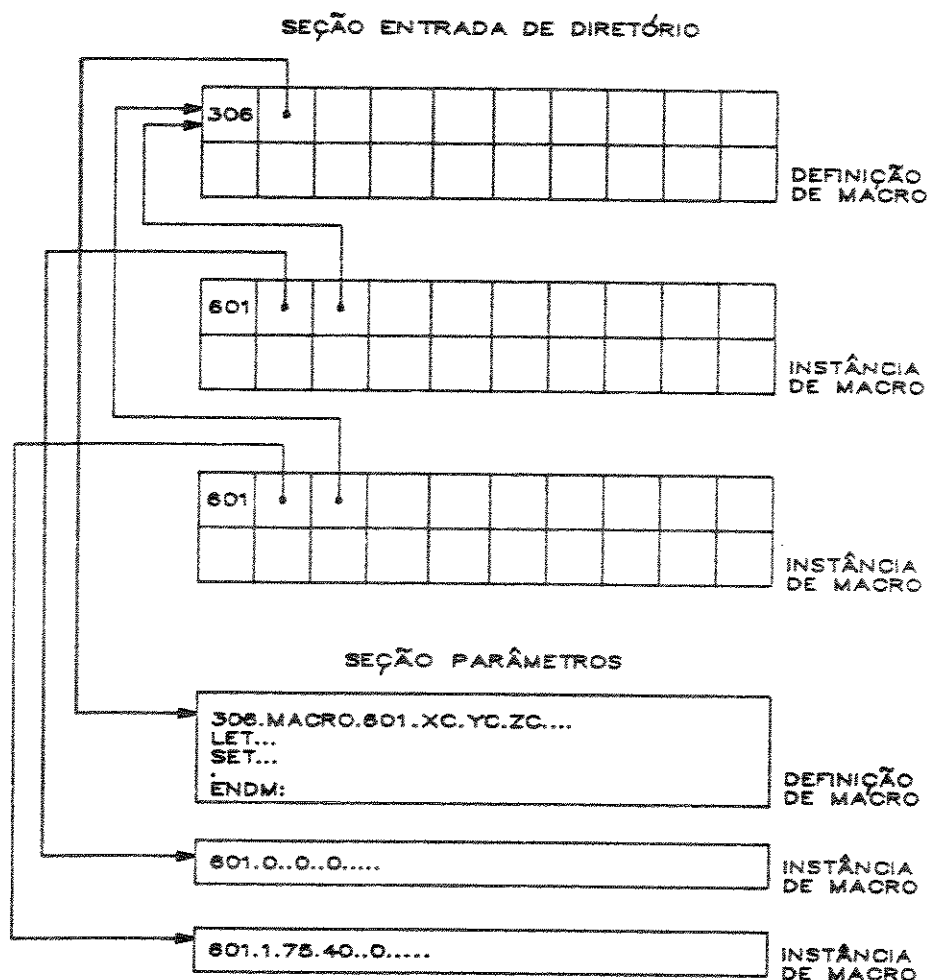


Fig. 4.26 - Instâncias de entidades definidas pelo usuário

A linguagem para definição de macros permite a utilização de constantes e variáveis de diversos tipos (real, inteiro, etc.). São também permitidas expressões e definidas funções de forma similar a uma biblioteca Fortran ($\sin(x)$, $\cos(x)$, $\log(x)$, etc.). Entre as instruções de linguagem encontram-se LET, SET, CONTINUE, IF, BREAK, etc.

Como exemplo da utilização de macros para a definição de entidades são apresentadas as entidades envolvidas na criação de um triângulo isósceles, conforme a Fig. 4.27.

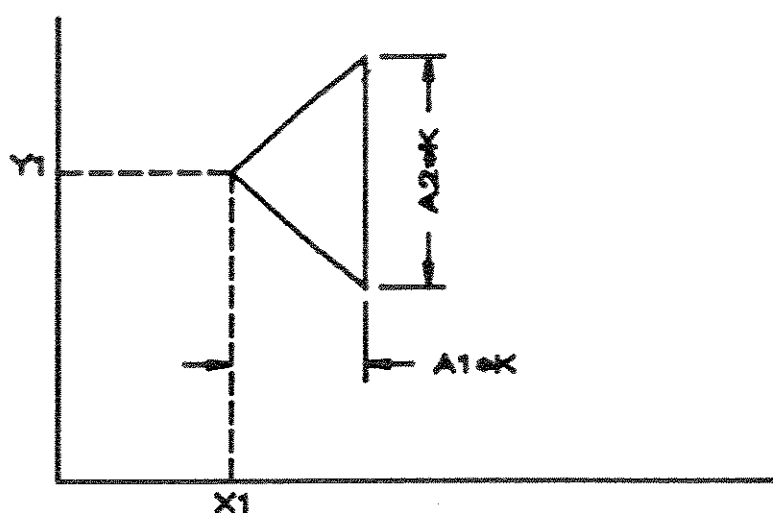


Fig. 4.27 - Triângulo definido por macro

A entidade definição de macro contém em sua seção parâmetros as seguintes instruções de linguagem:

```
306, MACRO, 621, X1, Y1, A1, A2, K;
LET Z=0;
SET # LINE1 = 110, X1, Y1, Z, X1+(K*A1), Y1+(K*A2/2), Z, 0, 0;
SET # LINE2 = 110, X1+(K*A1), Y1+(K*A2/2), Z, X1+(K*A1),
              Y1-(K*A2/2), Z, 0, 0;
SET # LINE3 = 110, X1+(K*A1), Y1-(K*A2/2), Z, X1, Y1, Z, 0, 0;
ENDM;
```

A entidade define para o triângulo o número de tipo 621, com os parâmetros X1, Y1, A1, A2, K. Portanto, uma entidade

instância de macro correspondente a uma ocorrência do triângulo deve apresentar como número de tipo 621 e ter em sua seção parâmetros um registro contendo, além do número de tipo (válido para qualquer entidade em IGES), cinco valores, por exemplo: 621, 0., 0., 17., 5., 1.

4.2.4.6. Referências Externas

Em IGES, referências podem ser feitas não só a entidades definidas no mesmo arquivo, mas, também àquelas especificadas em outro. Para isso, são utilizadas as seguintes entidades:

- entidade propriedade (406) com número de forma 12:
apresenta uma lista de todos os arquivos referenciados pelo arquivo que a contém.
- entidade referência externa (número de tipo 406):
faz a conexão entre a entidade que referencia em um arquivo e aquela que é referenciada em outro. Contém em seus parâmetros o nome do arquivo em que se encontra a entidade referenciada e o nome simbólico desta. Este nome simbólico será utilizado para, através da entidade instância de associação (número de tipo 402) com número de forma 12 (ver 4.2.4.1), encontrar a real localização da entidade referenciada. A Fig. 4.28 mostra as conexões para o caso de uma entidade instância de subfigura cuja definição se encontra em outro arquivo.

4.3. STEP

STEP ("Standard for the Exchange of Product Model Data") tem como objetivo estabelecer um padrão internacional para a completa representação das informações que definem um produto, de forma que seja viabilizada a efetiva transferência destas entre sistemas heterogêneos envolvidos no ciclo de projeto e manufatura

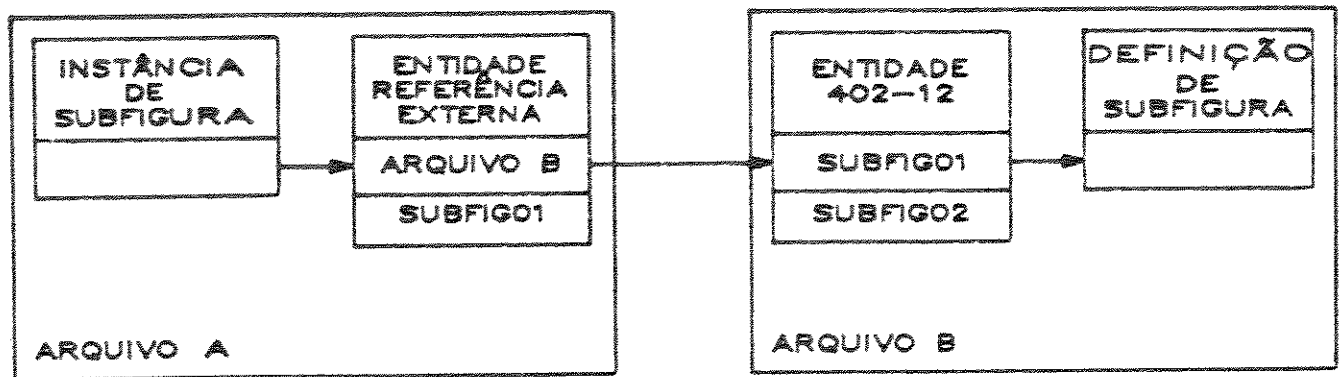


Fig. 4.28 - Exemplo de referência externa

de produtos. Para isso, foi constituído no âmbito da ISO, um grupo de trabalho (TC184/SC4/WG1). Este tomou como base o esforço americano de padronização PDES, havendo, a partir de então, uma fusão dos dois projetos. Em novembro de 1988 foi publicada a primeira versão da norma STEP /STE 88a/. Neste trabalho participaram especialistas dos principais usuários e fabricantes de sistemas CAD/CAM em todo o mundo.

Como mostra a Fig. 4.29, o modelo de informação definido em STEP apresenta uma arquitetura em três camadas: física, lógica e da aplicação /NOW 88/. Baseada nesta arquitetura, a metodologia para especificação do modelo de produto consiste em três etapas /WIL 87c/:

- desenvolver um modelo de informação para cada aplicação de interesse. Este deve capturar, de forma precisa, a semântica das aplicações (camada de aplicação);
- integrar os modelos das aplicações em um modelo de informações de produto, de tal forma que sejam eliminadas informações redundantes ou conflitantes. Desta forma, os modelos de referência das aplicações são mapeados em um esquema conceitual que especifica todas as informações que definem um produto (camada lógica);

- mapear o modelo conceitual para um ou mais formatos de troca de dados (camada física).

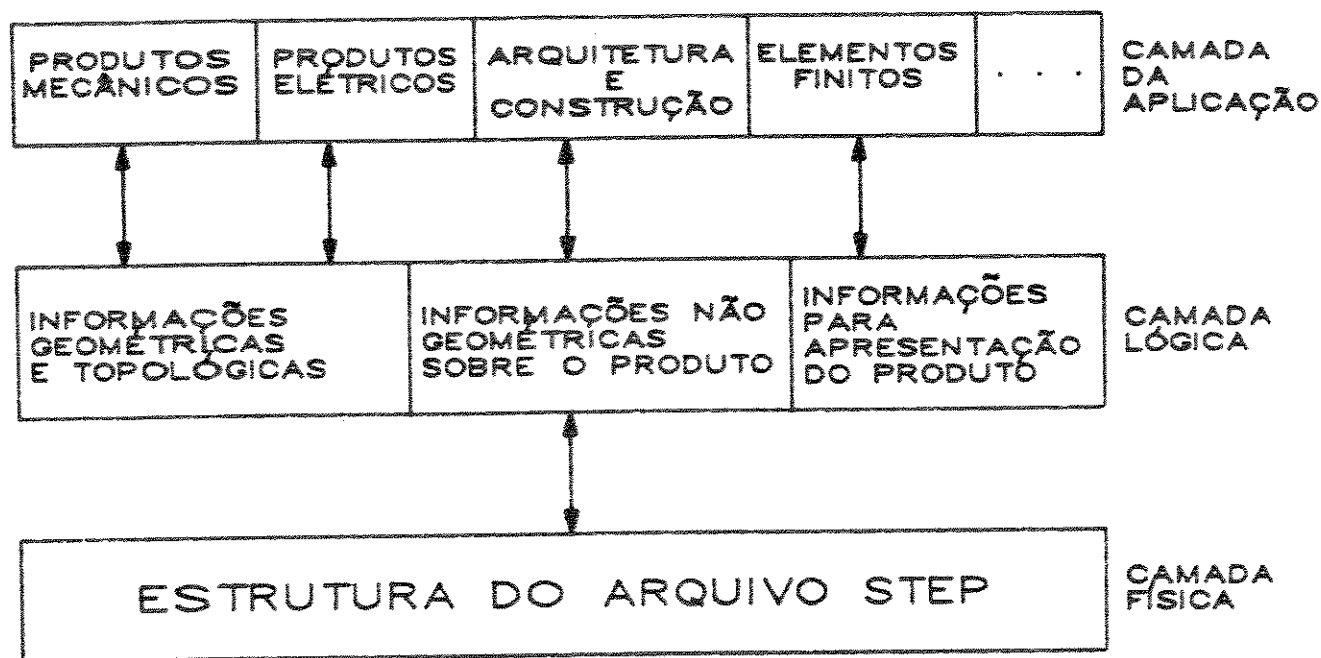


Fig. 4.29 - Modelo de três camadas em STEP

Para a especificação formal do modelo de informação nas camadas lógica e da aplicação, foi desenvolvida uma linguagem chamada Express. Da mesma forma, a sintaxe do arquivo neutro STEP é definida através de uma gramática formal.

4.3.1. A Linguagem Express

Express é uma linguagem que permite especificar precisamente as informações que modelam um subconjunto do mundo real (universo de discurso) em termos que pessoas possam entender e computadores possam processar /SCH 88/. A fim de atender aos requisitos impostos pelo objetivo de modelar informações sobre produtos, os seguintes aspectos foram considerados quanto ao projeto da linguagem:

- deve ser suficientemente poderosa, de tal forma que os complexos relacionamentos encontrados em modelos de produto possam ser expressos;
- deve permitir que os mesmos objetos possam ser modelados em múltiplos níveis de abstração;
- a semântica associada a um objeto sendo modelado deve ser independente do contexto em que ele se encontra.

A fim de capturar a semântica de um universo de discurso sendo descrito a linguagem provê a seguinte funcionalidade:

- Modelamento dos Objetos de Interesse:

Uma entidade, representando um objeto, é descrita por um conjunto de atributos. O nome de uma entidade descreve a função que esta representa no universo de discurso. A um atributo estão associados um nome e um tipo. O nome descreve a função do atributo na definição da entidade. O tipo define como o atributo é representado. Relacionamentos são descritos através de referência a uma entidade como atributo de outra.

- Definição das Restrições que devem ser Impostas aos Objetos:

Para que uma instância seja considerada válida, todas as restrições devem ser obedecidas. Estas são de dois tipos: locais e globais. Restrições locais aplicam-se aos valores dos atributos de uma entidade, seja individualmente ou combinados. Restrições globais aplicam-se às instâncias de um tipo de entidade.

- Definição das Operações em que os Objetos Participam:

O comportamento de entidades em um universo de discurso pode ser definido através de funções que se apliquem a ela.

Para satisfazer a estes requisitos a linguagem utiliza os seguintes elementos:

Esquema:

Descreve todos os elementos conceituais de um universo de discurso e seus relacionamentos. Inclui a declaração de tipos, entidades, funções e restrições. Para isto são definidos um conjunto de caracteres, operadores, constantes, funções e procedimentos pré-definidos. Qualquer elemento do esquema é criado através da declaração de um identificador (nome), ao qual está associado um escopo. O elemento só pode ser referenciado dentro do escopo em que foi definido. Os blocos que definem um escopo em Express são: esquema, entidade, função, procedimento e restrição. Na Fig. 4.30, a referência feita no atributo A1 é ilegal, pois a entidade E1 é desconhecida no escopo definido pelo esquema S3. Ela poderia ser referenciada a nível do esquema S1, pois este inclui os esquemas S2 e S3. As regras de escopo podem ser sobrepassadas pelo uso da diretiva ASSUME. Com a declaração ASSUME NOME_DOS_ESQUEMAS, qualquer referência não resolvida no escopo de um esquema será verificada nos esquemas sendo assumidos.

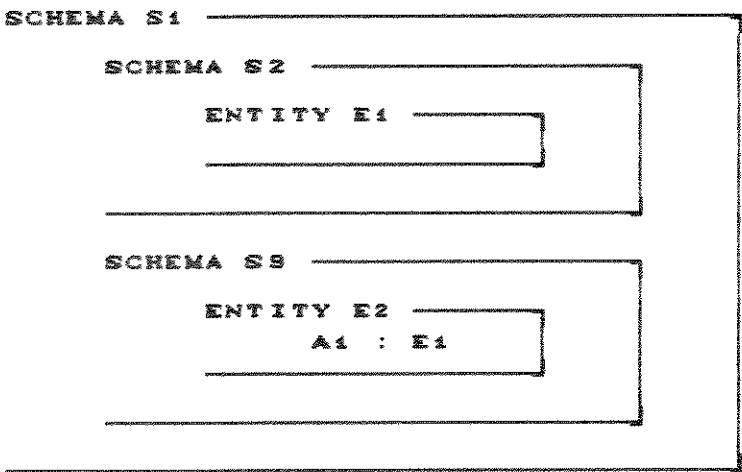


Fig. 4.30 - Escopo dos elementos em Express

Por outro lado, com a diretiva EXPORT um esquema define o que pode ser visto fora de seu escopo.

Tipo:

Identifica a forma de representação de um dado. Express prové as seguintes categorias de tipos:

- Tipos Básicos: inteiro, real, lógico e cadeia de caracteres. A cada tipo básico são associadas regras para representação dos literais correspondentes.
- Tipos Agregados: representam coleções de outros tipos. São definidos os tipos ARRAY, LIST e SET. O tipo LIST representa uma coleção ordenada de elementos. O tipo SET define uma coleção não-ordenada, em que todos os elementos têm valores diferentes entre si. A cardinalidade dos relacionamentos entre entidades pode ser estabelecida através de atributos cujo tipo é agregado.
- Tipo Entidade: uma entidade declarada em um esquema pode ser usada como tipo de um atributo, variável local de uma função ou parâmetro desta.
- Tipo Função: uma função pode ser usada como tipo para mostrar como atributos podem ser obtidos de outros atributos.
- Tipo Definido pelo Usuário: uma declaração TYPE define um novo tipo, o qual pode ser utilizado da mesma forma que os tipos pré-definidos pela linguagem.
- Tipo Enumeração: uma declaração ENUMERATION define um conjunto ordenado de valores representados por nomes. O seguinte exemplo mostra uma declaração de um tipo enumeração:
TYPE rotational_direction = ENUMERATION OF (clockwise,
counterclockwise);
END_TYPE.

- Tipo Seleção: uma declaração SELECT define os possíveis tipos que um atributo pode assumir, como no seguinte caso:

```
TYPE a = SELECT (REAL, INTEGER);  
END_TYPE;  
ENTITY b  
  c: a;  
END_ENTIFY;
```

O atributo c pode assumir um valor real ou inteiro. A ocorrência de um tipo é distinguida através de uma função pré-definida TYPEOF.

- Tipo Genérico: a declaração de um parâmetro de função como genérico permite que a ele seja associado qualquer tipo. A distinção é feita pela função TYPEOF.

Entidade:

Representa uma unidade de informação que pode ser trocada entre sistemas. A declaração de uma entidade cria um tipo cujo escopo é o esquema em que ela está inserida. Além disto, a entidade pode ser declarada como supertipo ou subtipo de uma ou mais outras entidades. Entidades que compartilham um supertipo têm características em comum. A estas definições aplica-se o mecanismo de herança. Os atributos da entidade supertipo são parte da definição de um subtipo. Uma entidade supertipo precisa mencionar as entidades que são seu subtipo. O relacionamento entre o supertipo e o subtipo pode ser de inclusão ou exclusão. Já o relacionamento subtipo-supertipo é sempre de inclusão, significando que a existência de um subtipo implica na existência de todos os seus supertipos.

Os atributos que definem uma entidade podem ser explícitos ou derivados. Qualquer instância da entidade conterá valores para seus atributos explícitos. Atributos derivados representam informação que é definida em função de atributos explícitos. Além disto, um atributo pode ser declarado opcional, significando que ele pode ser instanciado ou não. Seja o exemplo seguinte:

```

ENTITY unit_vector
    a,b: real,
    c: OPTIONAL real;
END_ENTITY;

```

No caso bidimensional o atributo c não é definido. Além disto, um atributo pode ser declarado único, o que significa que não podem haver instâncias iguais do atributo.

Restrições:

Restrições locais, ou seja, aquelas que se aplicam aos atributos de uma entidade individualmente ou de forma combinada, são definidas por declarações WHERE. Para o exemplo acima teríamos:

```

WHERE
    a**2 + b**2 + c**2 = 1.0;

```

Restrições globais, ou seja, que se aplicam ao modelo como um todo, são definidas por declarações RULE. O estado final de uma restrição global indica se houve ou não violação da regra, porém, não sendo tomada ação corretiva alguma. Um exemplo de restrição global é a imposição que deve existir um ponto na origem do sistema de coordenadas do modelo. Para isso, têm que ser verificadas as instâncias da entidade ponto no modelo.

Algoritmos:

Especificam, através de funções e procedimentos, o comportamento do modelo. Como em outra linguagem qualquer, traduzem-se em uma sequência de declarações, podendo ser definidas variáveis cujo escopo é local à função ou procedimento. Parâmetros formais definem a entrada para o algoritmo. Um procedimento opera sobre os parâmetros, podendo modificá-los. Uma função produz um resultado cujo tipo é especificado na declaração da função. Procedimentos e funções são pré-definidos em Express ou podem ser construídos, usando, para isso, declarações executáveis (IF_THEN_ELSE, REPEAT, CASE, etc.).

4.3.2. Esquema Conceitual em STEP

A nível da camada lógica da arquitetura do modelo de produto STEP é definido um modelo integrado de informações de produto (IPIM). Este descreve, além das informações comuns a várias aplicações (geometria, topologia, etc.), aquelas que dizem respeito especificamente a cada aplicação. Como foi visto na introdução desta seção, IPIM é o resultado de um processo de integração dos modelos de referência para as aplicações. Este processo teve como objetivo produzir um modelo de informação não-ambíguo, completo e com um mínimo de redundância. Usando-se a terminologia de bases de dados, IPIM representa um esquema conceitual, sendo definido formalmente utilizando-se a linguagem Express, portanto, de forma independente do formato dos dados na camada física.

IPIM é definido como o esquema que engloba todos os subesquemas que modelam as informações para definição do produto /STE 88b/. Sendo assim, todas as constantes, tipos, funções e procedimentos pré-definidos a nível do esquema IPIM são válidos em todos os subesquemas. Além destes elementos genéricos, o documento IPIM especifica subesquemas para a representação das seguintes informações:

- Geometria:

O esquema para geometria em STEP é definido no nível mais alto pela seguinte entidade:

```
ENTITY geometry
  SUPERTYPE OF (coordinate_system XOR
    point XOR
    vector XOR
    axis_placement XOR
    transformation XOR
    curve XOR
    surface);
```

Portanto, são definidos todos os elementos geométricos básicos que são utilizados por outros esquemas. Curvas incluem, entre outras, cónicas, B-spline racional, linha. Superfícies incluem,

entre outras, plano, superfícies quadráticas, B-spline racional, superfície de revolução. A título de ilustração, seja a definição de um círculo:

```
ENTITY CIRCLE
  SUBTYPE OF (CONIC)
    radius: REAL;
    position: axis2_placement;
  WHERE
    RADIUS > 0;
END_ENTITY;
```

AXIS2_PLACEMENT define o centro e o plano que contém o círculo.

- Topologia:

O esquema para topologia tem por objetivo atender não só aos requisitos impostos por modelos sólidos, mas também a outras aplicações como esquemas lógicos de circuitos elétricos, etc. Sendo assim, as entidades topológicas são definidas independentemente de sua utilização, seja pelas aplicações ou por entidades de mais alto nível na estrutura topológica. Portanto, quaisquer restrições dependentes do contexto serão impostas ao nível da entidade que referencia uma entidade topológica. A estrutura topológica é definida hierarquicamente, com uma superfície definida por suas faces, e estas por seus laços. Um laço é especificado pelas arestas que o compõem, sendo estas limitadas por seus vértices. Caracteriza-se, desta forma, um relacionamento F-A (face-aresta, ver Fig. 3.35), o qual, conforme observado no item 3.3.1.3.3 é topologicamente suficiente. Funções implementam variações da fórmula de Euler-Poincaré (ver item 3.3.1.3) como restrições às entidades topológicas.

- Formas Características:

Este esquema define um modelo conceitual que permite a associação de elementos característicos (furos, roscas, rebaixos) a uma representação geométrica. São suportados, entre outros, elementos característicos para transição entre formas geométricas, deformação, depressão, rebaixos cujas faces não são

planas e padrões circulares ou retangulares de formas características.

- Tolerâncias:

É definido um esquema para representação dos desvios da forma nominal do produto, de tal forma que as entidades geométricas afetadas pela tolerância são referenciadas por uma entidade do modelo de tolerâncias. Um exemplo do modelo é a entidade concentricidade, cujos atributos especificam o desvio admissível e referências às áreas circulares correspondentes.

- Representação de Formas Geométricas:

Este esquema descreve, utilizando os esquemas acima apresentados, a forma de um produto. Para isto, são suportados modelos de arame, de superfície e sólidos. Estes incluem modelos construtivos (CSG) e por fronteira (B-REP), sendo, para estes, especificado um caso particular em que todas as faces são planas. Além disto, a entidade ASSEMBLY define um produto em termos de suas partes constituintes, sejam elas submontagens ou componentes elementares. Para cada um destes componentes, a entidade PART_MODEL contém atributos que definem a forma nominal, formas características associadas e tolerâncias que afetam o componente.

- Interface à Representação de Formas Geométricas:

Este esquema tem por objetivo prover entidades que permitam às aplicações referenciar formas geométricas independentemente da maneira como estas são representadas. Desta forma, pode torna-se transparente, por exemplo, a um esquema voltado à aplicação se uma forma tridimensional é representada por um modelo CSG ou B-REP.

- Material:

Neste esquema são definidas entidades que especificam coeficientes elásticos e térmicos para os materiais. Tais entidades são referenciadas por aplicações como é o caso de modelos por elementos finitos.

- Apresentação:

Provê as entidades que suportam a visualização das informações contidas no modelo de produto. Embora outros padrões possam ser utilizados para transmitir informações gráficas (por exemplo, CGM = "Computer Graphics Metafile"), a junção das informações que governam a aparência do modelo àquelas que o definem traz vantagens, uma vez que estas consistirão em um volume de informações menor do que aquele necessário à completa representação das informações gráficas por CGM /KLE 88/. São definidos fontes de caracteres, tabelas de estilo de linhas, tabelas de estilo de texto, hachuras, etc. Outras entidades especificam os parâmetros para projeção (paralela ou perspectiva) e recorte quando são visualizadas as informações geométricas.

Os esquemas acima descritos especificam os recursos básicos para a definição de um produto. Além destes, o documento IPIM contém esquemas, os quais referenciam os esquemas básicos, voltados às seguintes aplicações:

- geração de desenhos de engenharia (drafting);
- gerenciamento de configuração da estrutura do produto;
- arquitetura e construção;
- representação de produtos elétricos;
- modelagem por elementos finitos.

Um último esquema, ainda em fase de desenvolvimento, diz respeito à capacidade de referenciar entidades cuja definição se dá externamente ao arquivo que contém uma instância do modelo de produto.

4.3.3. Formato do Arquivo Neutro STEP

A sintaxe do arquivo neutro STEP é definida formalmente utilizando-se a Notação de Sintaxe de Wirth, a qual permite a especificação de uma gramática de forma não-ambígua /STE 88c/. O formato do arquivo STEP é sequencial, sendo utilizado para sua

definição o conjunto de caracteres ASCII. Cadeias destes caracteres formam palavras válidas no contexto do arquivo e que, portanto, podem ser reconhecidas por um analisador léxico. As seqüências válidas destas palavras, chamadas produções, definem a sintaxe do arquivo. As palavras válidas para o formato STEP incluem:

- separador de palavras (espaço ou comentário);
- delimitadores: caracteres especiais que servem para agrupar coleções de dados (por exemplo: fim da definição de uma entidade, início de uma seção do arquivo);
- palavras reservadas: são palavras com significado pré-determinado (por exemplo, "STEP;" indica o começo do arquivo);
- palavras-chave: indicam a ocorrência de uma entidade no arquivo neutro. São representadas pelos nomes das entidades definidas no esquema conceitual;
- tipos de dados: inteiro, real, cadeia de caracteres, identificador de entidade, referência a entidade, enumeração e lista.

Um identificador de entidade define uma posição que será utilizada pelo tipo referência a entidade nos casos em que um tipo entidade é associado a um atributo. Os identificadores não são baseados nas posições relativa da ocorrência da entidade no arquivo, o que significa que eles não têm que ser seqüencialmente ordenados. Uma enumeração representa algum valor pré-definido no esquema conceitual (por exemplo, .TRUE.). Listas permitem representar tipos estruturados do esquema conceitual (SET, LIST, ARRAY).

O arquivo neutro STEP consiste em duas seções, que seguem pela ordem a palavra reservada STEP:

- Seção Header:

Contém as informações que dizem respeito ao arquivo como um todo. Inclui, entre outros, o nome do arquivo, data de criação, versão de STEP utilizada e nível de implementação. É iniciada com a palavra reservada HEADER e terminada com ENDSEC.

- Seção DATA:

Define os dados de produto a serem transferidos, contendo ocorrências de entidades especificadas no esquema conceitual. Esta seção é iniciada pela palavra reservada DATA e terminada com ENDSEC. Uma restrição imposta é que ocorrências de entidades devem ser definidas no arquivo antes de serem referenciadas. Cada ocorrência apresenta a seguinte sintaxe:

```
IDENTIFICADOR_DE_ENTIDADE = NOME_DA_ENTIDADE [ESTRUTURA  
ESCOPO] (VALORES_DOS_ATRIBUTOS);
```

onde [] significa opcional. A estrutura escopo é iniciada pela palavra reservada SCOPE, seguida de ocorrências de entidades e terminada com ENDScope. Esta estrutura define uma coleção de entidades que só podem ser referenciadas como atributos de entidades que estão inseridas neste escopo, portanto, tornando-as invisíveis àquelas entidades em outro escopo. Um escopo pode ser sobrepassado pela declaração de entidades como exportáveis. Além disto, este mecanismo tem implicações quanto ao comportamento operacional do modelo. Qualquer operação a que uma entidade for submetida, será repassada a todas as entidades em seu escopo às quais esta operação for aplicável. Outra forma de estabelecer dependência entre entidades, a nível do arquivo neutro, é definir uma entidade como atributo de outra nos casos em que este atributo for do tipo entidade.

Uma vez declaradas as duas seções, a palavra reservada ENDSTEP especifica o fim do arquivo. O exemplo abaixo mostra um arquivo STEP em que estão definidos dois pontos, dois vértices e uma aresta:

```

STEP;
HEADER;
    parâmetros da seção HEADER
ENDSEC;
DATA;
$1=POINT (0.0, 0.0, 0.0);
$2=POINT (0.0, 0.0, 5.0);
$11=VERTEX (#1);
$12=VERTEX (#2);
$21=EDGE (#11, #12);
ENDSEC;
ENDSTEP;

```

onde \$ significa identificador de entidade e # significa referência a entidade. Uma última observação relativa a esta seção diz respeito ao mapeamento entre o esquema conceitual e o arquivo neutro. Uma vez que ambos são definidos formalmente, é possível definir regras que viabilizem que este mapeamento seja automático /STE 88d/. Conforme se pode observar na estrutura definida para o arquivo STEP, as declarações executáveis (funções, procedimentos e regras) não são mapeadas. Além disso, os atributos de entidades supertipo são incorporados aos seus subtipos, não sendo aquelas mapeadas.

4.3.4. Níveis de Implementação em STEP

A questão dos níveis de implementação de padrões para troca de informações de produto pode ser vista sob dois aspectos /STE 88e/:

- usuários desejam transferir entre sistemas as informações requeridas em uma dada área de aplicação. Sendo assim, é preciso estabelecer níveis de implementação que definam os subconjuntos do esquema conceitual que um sistema deve tratar para que esteja habilitado a trocar informações relativas a uma aplicação. No caso, por exemplo, de informações sobre a forma geométrica de um produto devem ser especificados os modelos que um sistema suporta (arame, superfície, sólidos). Uma vez definidos estes níveis de implementação, um sistema pode comunicar a

outro, a priori, o nível requerido para viabilizar a troca de informação entre eles;

- implementadores precisam analisar os sistemas face às várias estratégias para compartilhamento de informações entre eles (arquivo neutro, bases de dados - ver Cap. 2), objetivando, assim, encontrar a melhor forma de integrá-los. Para isso, a arquitetura destes sistemas deve ser confrontada com uma arquitetura-padrão, a fim de que sejam reconhecidos pontos comuns entre eles, que permitam tornar mais eficiente o compartilhamento de informações. A especificação da arquitetura-padrão consiste em definir os elementos que a compõem e a funcionalidade requerida em cada um deles.

No caso de STEP, estes elementos se traduzem em níveis de implementação, sendo requerido que implementações de um nível suportem também todos os níveis inferiores. Para a definição destes níveis foram utilizados os seguintes critérios:

- 1) O sistema pode ler/escrever em um arquivo neutro padrão?
- 2) O sistema pode gerar/acessar uma forma operacional (acessível através de chamadas de funções) padrão?
- 3) Qual é a granularidade dos dados que o sistema pode acessar?
 - pode consultar um único atributo de uma entidade
 - pode consultar múltiplos atributos de uma entidade
 - pode consultar uma única entidade
 - pode consultar múltiplas entidades
 - pode consultar um único modelo
 - pode consultar múltiplos modelos
- 4) Que formas padrão de consulta aos dados são suportados?
 - nenhuma forma padrão
 - chamadas de funções padronizadas

- linguagem de manipulação de dados interativa padronizada
 - linguagem de manipulação de dados por lotes padronizada
 - todas as formas anteriores
- 5) Que formas padrão de resposta às consultas são suportadas?
- nenhuma forma padrão
 - parâmetros de saída de funções padronizadas
 - saída padronizada em um arquivo
 - saída padronizada em um dispositivo
 - todas as formas anteriores
- 6) Que localizações físicas para os dados a implementação suporta?
- na memória local
 - em um disco local
 - em disco local e/ou remoto
- 7) De que forma se pode operar sobre o modelo de produto?
- através de consultas baseadas no modelo hierárquico
 - através de consultas baseadas no modelo rede
 - através de consultas baseadas no modelo relacional

Este critério define que estruturas de dados e operadores associados uma implementação suporta /DAT 82/.

- 8) São verificadas as restrições impostas no esquema conceitual?
- 9) Qual é o nível de automação do processo de compartilhamento de informações?
- não automatizado
 - parcialmente automatizado (um sistema notifica o outro da necessidade de trocar informações)
 - completamente automatizado

Este critério estabelece em que medida será necessária a intervenção do usuário para que os sistemas envolvidos cooperem entre si.

Baseado nestes critérios foram definidos quatro níveis de implementação, cuja funcionalidade é definida a seguir:

- Nível 1: Troca por Arquivo Neutro

- dados de produto são traduzidos de/para um formato de arquivo padrão usando pré e pós-processadores não padronizados (critério 1);
- os dados do produto no arquivo neutro são derivados de IPIM (Modelo Integrado de Informações de Produto);
- a granularidade dos dados no arquivo vai de múltiplos modelos a uma única entidade (critério 3);
- nenhuma forma padronizada de consulta é definida (critério 4);
- as restrições de integridade impostas ao modelo não são verificadas (critério 8);
- o nível de automação do processo de compartilhamento de informações não é definido (critério 9).

- Nível 2 : Troca por Forma Operacional

- dados de produto são traduzidos de/para uma forma operacional padrão;
- os dados na forma operacional são derivados de IPIM;
- a granularidade dos dados vai de um único modelo até uma única entidade (critério 3);
- precisa haver um mapeamento padrão para a forma operacional dos dados contidos no arquivo neutro;

- dados de produto na forma operacional devem poder ser acessados através de chamadas de funções padronizadas (critério 4);
- deve-se poder operar sobre a forma operacional através de consultas baseadas nos modelos relacional, rede e hierárquica (critério 7);
- as restrições de integridade impostas ao modelo não são verificadas (critério 8);
- o nível de automação do processo de compartilhamento de informações não é definido (critério 9).

- Nível 3: Troca por Base de Dados

- dados de produto são traduzidos de/para um sistema gerenciador de base de dados (DBMS);
- os dados de produto no DBMS são derivados de IPIM;
- a granularidade dos dados vai de múltiplos modelos a uma única entidade (critério 3);
- precisa haver um mapeamento padrão para o DBMS dos dados contidos no arquivo neutro;
- dados de produto no DBMS devem poder ser acessados através de chamadas de funções padronizadas ou linguagem de manipulação de dados padronizada (critério 4);
- deve-se poder operar sobre a base de dados através de consultas baseadas nos modelos relacional, rede e hierárquico (critério 7);
- as restrições de integridade impostas ao modelo não são verificadas (critério 8);
- o processo de compartilhamento de informações é completamente automático (critério 9).

- Nível 4: Troca por Base de Conhecimento

- é especificada a mesma funcionalidade do nível 3, com exceção do fato de que as restrições impostas ao modelo são verificadas;

Estabelecidos os níveis de implementação, para que se viabilize a troca de informações entre sistemas devem ser definidas as seguintes interfaces:

- formato de arquivo padrão (ver item 4.3.3);
- conjunto padronizado de rotinas para acesso à forma operacional e à base de dados;
- linguagem de manipulação de dados padronizada para acesso à base de dados.

As duas últimas interfaces estão em fase de discussão.

4.4. Análise Comparativa dos Padrões IGES e STEP

STEP tem por objetivo suprir as deficiências constatadas ao longo da utilização de IGES como mecanismo para transferência de informações de produtos entre os componentes de um sistema CIM. A diferença fundamental entre os dois padrões diz respeito à arquitetura em que são baseados. Na arquitetura de duas camadas de IGES, as informações em um sistema são mapeadas diretamente em um formato neutro definido pelo padrão. A arquitetura de três camadas de STEP (ver Fig. 4.29) inclui a definição pelo padrão de um esquema conceitual, especificado formalmente utilizando-se a linguagem Express. As informações contidas neste esquema conceitual são mapeadas em um formato neutro de arquivo, também definido pelo padrão. A existência deste esquema conceitual pode ser vista sob vários aspectos no que diz respeito à satisfação dos requisitos impostos por CIM:

- define um modelo de produto que, conforme visto (item 2.1), é um dos modelos de informação necessários em CIM. Diferentemente de IGES, que é voltado principalmente à transferência dos dados contidos em desenhos de engenharia, apropriados à interpretação por especialistas, o modelo definido por STEP tem por objetivo suprir as informações requeridas por aplicações automatizadas (sem a intervenção do usuário). Em outras palavras, pode-se dizer que, enquanto IGES visa a transferência de dados, STEP visa a transferência de informações, pois o esquema conceitual captura a semântica dos dados de produto compartilhados pelos sistemas componentes de CIM, deixando esta semântica de ser definida pelas aplicações que fazem uso das informações de produto. Desta forma, elimina-se o problema de transferir dados cujos significados são diferentes para aplicações diferentes e aquele de transferir uma informação que é representada por diferentes dados em diferentes sistemas;
- o esquema conceitual provê o modelo que constitui o nível conceitual global na arquitetura de um sistema de base de dados distribuída (ver Fig. 2.3). Sendo assim, o controle de integridade dos dados e o controle de concorrência de acesso a estes passa a ser responsabilidade do sistema gerenciador de base de dados distribuída, viabilizando, portanto, que as informações corretas estejam prontamente disponíveis às aplicações que as necessitem. Além disto, qualquer redundância de dados entre os sistemas é conhecida pelo gerenciador. Esta abordagem para compartilhamento de informações entre os sistemas, que corresponde aos níveis três e quatro de implementação definidos em STEP (ver item 4.3.4), favorece a obtenção de sistemas CIM reconfiguráveis, ou seja, em que os sistemas componentes podem ser modificados e reintegrados com um dispêndio mínimo de recursos. Por outro lado, a única estratégia suportada em IGES é troca por arquivo neutro (nível um).

A Tabela 4.11 sintetiza os tipos de informação para definição de produtos (ver Capítulo 3) suportados por IGES e STEP. Como se pode observar, STEP satisfaz os requisitos impostos à definição completa e não-ambígua de um produto. Um problema, relativo a estas informações, que ocorre em IGES é que não são definidos níveis de implementação que especifiquem que subconjuntos das informações uma determinada implementação suporta. Sendo assim, dois sistemas que declaram poder trocar informações via arquivo neutro IGES poderão não ter sucesso nesta transferência, pois podem implementar diferentes subconjuntos de entidades. Como foi visto no item 4.3.4, este problema será sanado em STEP. Outra questão que STEP procura resolver diz respeito à instabilidade numérica resultante da forma como são definidas as entidades geoméricas em IGES. Em STEP, ao contrário de IGES, em que as entidades são especificadas por valores numéricos, as entidades geométricas são definidas através de referência a entidades básicas como ponto e direção.

TIPO DE INFORMAÇÃO	IGES	STEP
MODELO DE ARAME	SIM	SIM
MODELO DE SUPERFICIE	SIM	SIM
MODELO SOLIDO CONSTRUTIVO	SIM (VERSAO 4.0)	SIM
MODELO SOLIDO POR FRONTEIRA	NAO	SIM
ESTRUTURA DO PRODUTO	NAO	SIM
INFORMACOES ADMINISTRATIVAS	NAO	SIM
MODELO DE TOLERANCIAS	NAO	SIM
MODELO DE FORMAS CARACTERISTICAS	NAO	SIM
MATERIAIS	SIM	SIM
APRESENTACAO	LIMITADO	SIM

TABELA 4.11 - Tipos de informação suportados por IGES e STEP

Quanto ao formato do arquivo neutro definido pelos padrões, podem ser feitas as seguintes observações:

- a definição formal da sintaxe do arquivo, como acontece em STEP, favorece a implementação de um analisador sintático;
- STEP procura corrigir uma das deficiências frequentemente apontadas em IGES, que é o excessivo tamanho dos arquivos;
- o fato de STEP exigir que não sejam permitidas referências a entidades ainda não definidas no arquivo minimiza o gerenciamento dos dados necessário na fase de pós-processamento do arquivo neutro.

CAPITULO 5

SOLUÇÕES DE INTEGRAÇÃO COM REDE DE COMUNICAÇÃO

Conforme observado no Capítulo 2, a adoção de padrões que permitam a interconexão de sistemas heterogêneos é fundamental à consecução do objetivo de integração destes. Reconhecendo esta questão, a General Motors iniciou em 1980 o projeto MAP ("Manufacturing Automation Protocol"), com o objetivo de especificar protocolos de comunicação a serem usados pelas redes de campo das plantas industriais /MEN 89/. Posteriormente, foi formado um grupo técnico para MAP no âmbito da SME ("Society of Manufacturing Engineers"). Em 1985, como resultado de esforços conjuntos de usuários e fabricantes no âmbito da NBS ("National Bureau of Standards"), foi definido um projeto que visa complementar a especificação MAP, atendendo aos requisitos de comunicação nos níveis técnicos e administrativos de uma empresa, sendo denominado TOP ("Technical and Office Protocol").

A arquitetura de rede escolhida para ambos os projetos é baseada no modelo de referência ISO/OSI ("Open Systems Interconnection"). Este, conforme mostra a Fig. 5.1, é um modelo composto de sete camadas, em que é definida a funcionalidade requerida em cada uma a fim de que os sistemas envolvidos sejam interoperáveis, ou seja, dados possam ser transferidos de um sistema a outro e sejam corretamente interpretados e utilizados pelos sistemas que os recebem /TOP 87/. Porém, por se tratar de um modelo abstrato, ISO/OSI não determina especificamente a funcionalidade em cada camada a ser implementada, cabendo aos projetos MAP/TOP essa definição. Onde possível, têm sido utilizados padrões definidos no âmbito da ISO, conhecidos como protocolos OSI. Como redes TOP provêm o suporte básico de comunicação necessário à transferência de dados de produtos, este será detalhado a seguir.

5.1. O PROJETO TOP

Entre as aplicações típicas que devem ser suportadas pela especificação TOP incluem-se /TOP 87/:

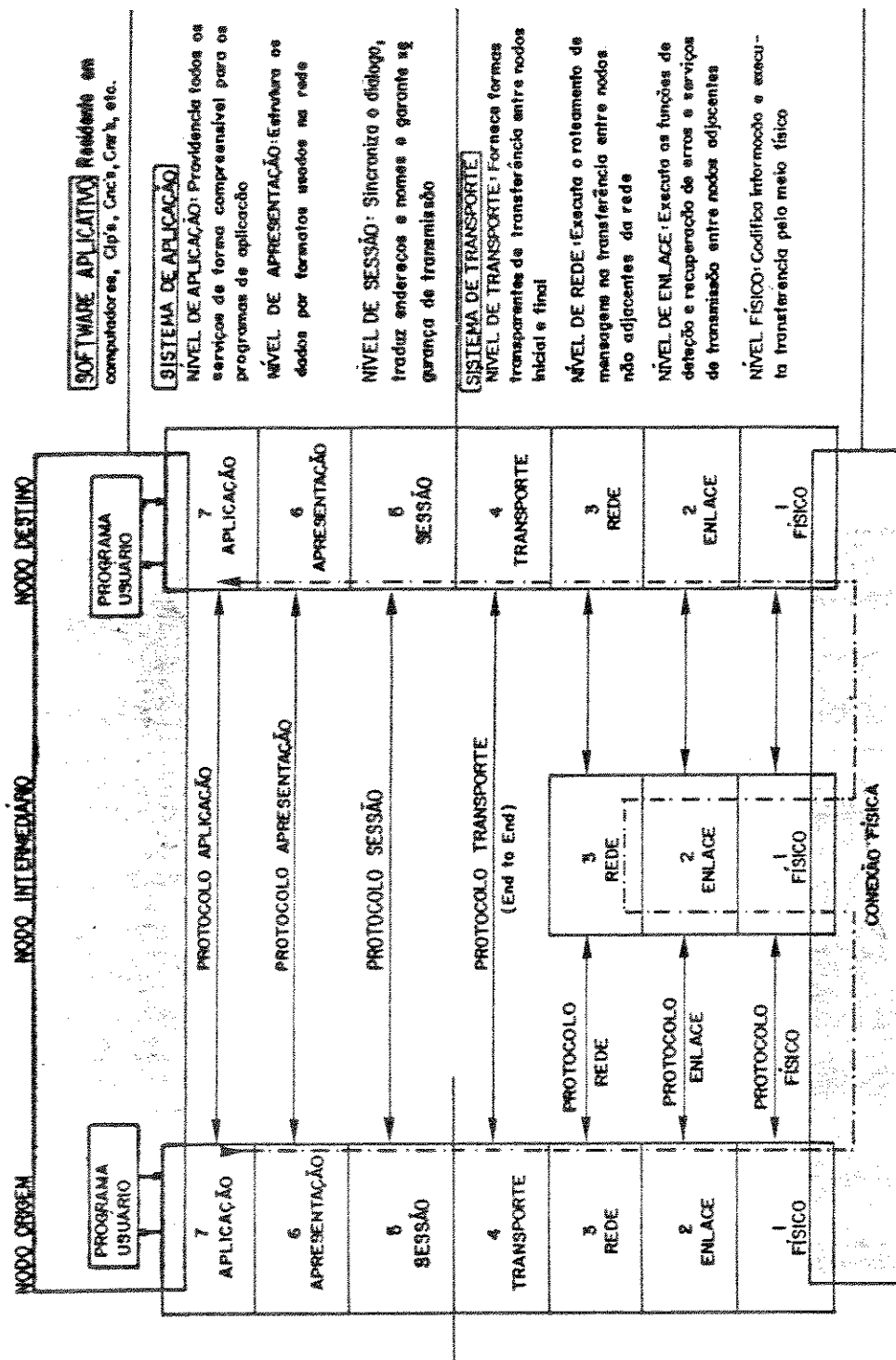


Fig. 5.1 - O modelo OSI/ISO

- correio eletrônico;
- processamento de textos;
- informação híbrida texto/gráficos;
- acesso a base de dados;
- transferência de arquivos;
- sistemas CAD/CAM distribuídos;
- troca de documentos;
- transações bancárias;
- sistemas de administração distribuídos;
- compartilhamento de recursos como impressoras e memórias de massa.

Para atender aos requisitos quanto à comunicação impostos por estas aplicações, as redes TOP utilizam três componentes básicos:

- sistemas finais: são aqueles que implementam as sete camadas de protocolos requeridas na especificação TOP;
- sistemas intermediários: são aqueles que implementam protocolos até a camada de rede. São usados para interconectar duas ou mais sub-redes.
- sub-redes: são coleções de equipamentos e meios físicos que interligam sistemas finais e intermediários a fim de comunicar dados.

No que concerne a um sistema final, dois aspectos devem ser considerados. O primeiro diz respeito ao tipo de sub-rede ao qual este será conectado. O segundo se refere aos serviços de comunicação que o sistema oferecerá a seus usuários. Para permitir a ligação entre a visão do usuário, que é orientada à função, e a visão do fabricante, que é orientada ao projeto, quanto aos aspectos mencionados acima, são definidos blocos construtivos para especificação. O uso destes blocos permite ao

fabricante, uma vez definidos os requisitos do usuário, relacioná-los à especificação TOP e determinar os protocolos, unidades funcionais e opções necessárias à implementação de um produto em conformidade com TOP. A Fig. 5.2 mostra os blocos construtivos definidos em TOP. Para cada bloco são especificados a função que exerce, um conjunto de parâmetros e opções baseados em padrões internacionais e regras que estabelecem restrições à associação de blocos entre si. Uma arquitetura completa de sistema final TOP incluirá:

- um ou mais dos quatro possíveis blocos que compõem o sistema de transporte (cobrindo as camadas 1, 2, 3 e 4). Estes blocos são:
 - bloco de acesso à sub-rede CSMA/CD;
 - bloco de acesso à sub-rede "Token-passing-ring";
 - bloco de acesso à sub-rede "Token-passing-bus";
 - bloco de acesso à sub-rede X.25 de comutação de pacotes;
- pelo menos um dos blocos do sistema da aplicação (camadas 5, 6 e 7). Estes blocos são:
 - correio eletrônico (MHS);
 - acesso a arquivos remotos (FTAM);
 - acesso a terminais remotos (VT);
 - serviços de diretório;
 - serviços de gerenciamento da rede;
- opcionalmente poderão ser incluídos blocos construtivos que definem a estrutura e formato dos dados de um arquivo remotamente acessível ou de uma mensagem a ser transferida por correio eletrônico. São definidos os seguintes formatos:
 - CGMIF ("Computer Graphics Metafile Interchange Format") para a troca de arquivos gráficos;
 - ODIF ("Office Document Interchange Format") para a troca de documentos contendo texto e gráficos;

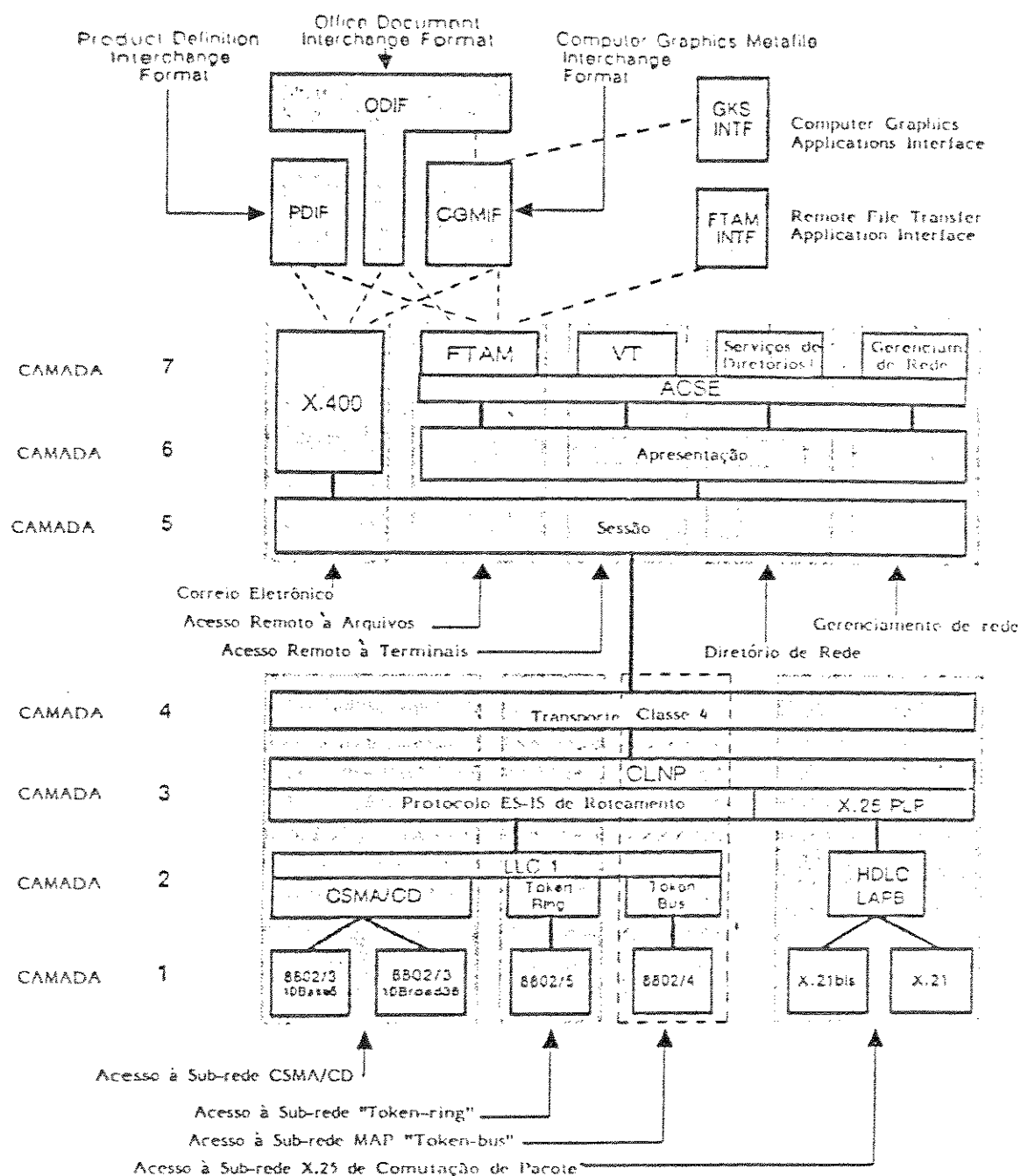


Fig. 5.2 - Blocos construtivos em Top 3.0

- PDIF ("Product Definition Interchange Format") para troca de dados de definição de produtos;
- poderá conter interfaces para os processos de aplicação dos usuários. São elas:
 - GKS-INTF ("Graphics Kernel System-Interface") para aplicações gráficas;
 - FTAM-INTF ("File Transfer, Access and Management-Interface") para o protocolo de aplicação FTAM.

5.2. TROCA DE DADOS DE PRODUTO VIA IGES USANDO REDES TOP

Conforme citado acima, um dos blocos construtivos definidos em TOP diz respeito à troca de dados para definição de produtos. O documento TOP versão 3.0 /TOP 87/ especifica IGES versão 3.0 /IGE 86/ como formato de arquivo neutro para transferência destes dados. Como mostra a Fig. 5.3, em que são omitidas as camadas 1 a 6 por serem transparentes ao processo de aplicação, a transferência de arquivos IGES pode se dar através dos protocolos

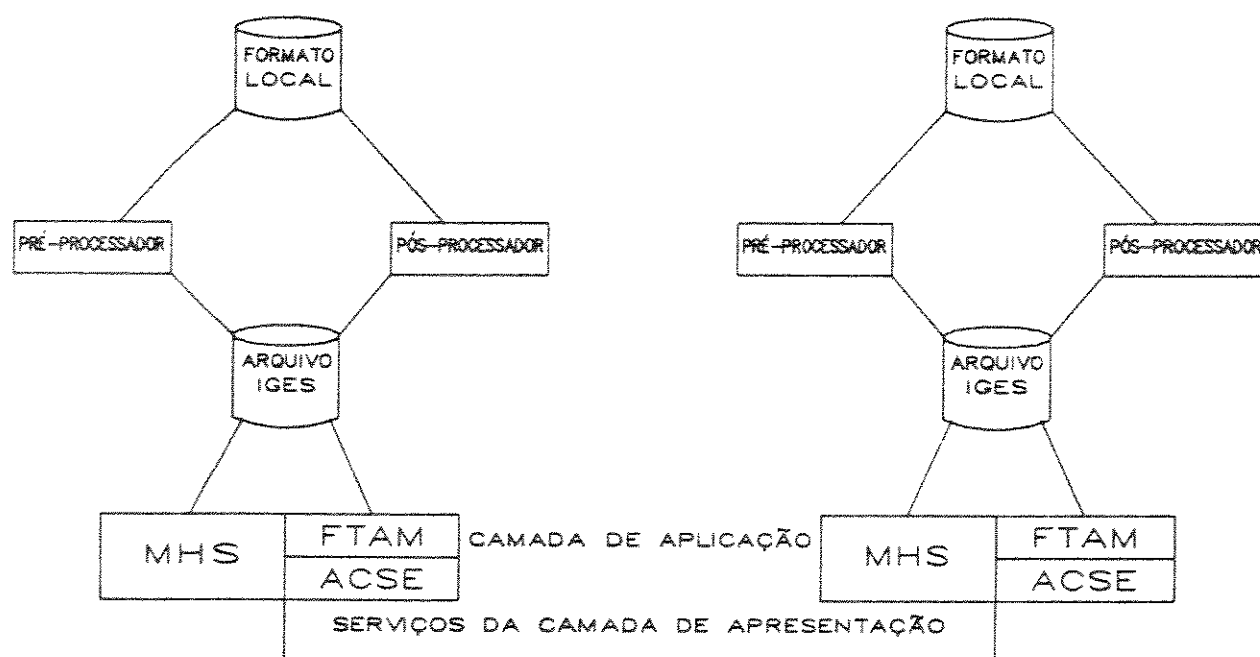


Fig. 5.3 - Troca de dados de produtos via IGES em redes TOP

da camada de aplicação MHS ou FTAM. Uma vez que dois sistemas desejem trocar dados de produto, o pré-processador gera um arquivo IGES que é transferido ao outro nó da rede e, então, pós-processado. Quanto à implementação de pré e pós-processadores, é conveniente separar as funções relacionadas ao sistema local daquelas que manipulam diretamente o arquivo IGES /WEI 84/. Desta forma, para cada nova implementação, somente as funções locais têm que ser escritas. A Fig. 5.4 apresenta um arquitetura para pré e pós-processadores que obedecem ao requisito acima. O programa editor deve ler/escrever um arquivo IGES e gerar/processar arquivos intermediários, os quais devem permitir um acesso eficiente por parte dos tradutores. Para isso, são gerados arquivos separados que contêm as seções parâmetros e entrada de diretório, sendo para esta eliminadas as informações redundantes ou que não se apliquem a uma determinada entidade. Além disto, o editor deve converter os dados do formato ASCII para o formato adequado ao ambiente operacional em que se encontram os sistemas. Os programas tradutores devem, além de gerar/processar os dados no formato local, prover ao usuário funções que permitam especificar como entidades em que não existe compatibilidade entre o sistema local e IGES devem ser processadas.

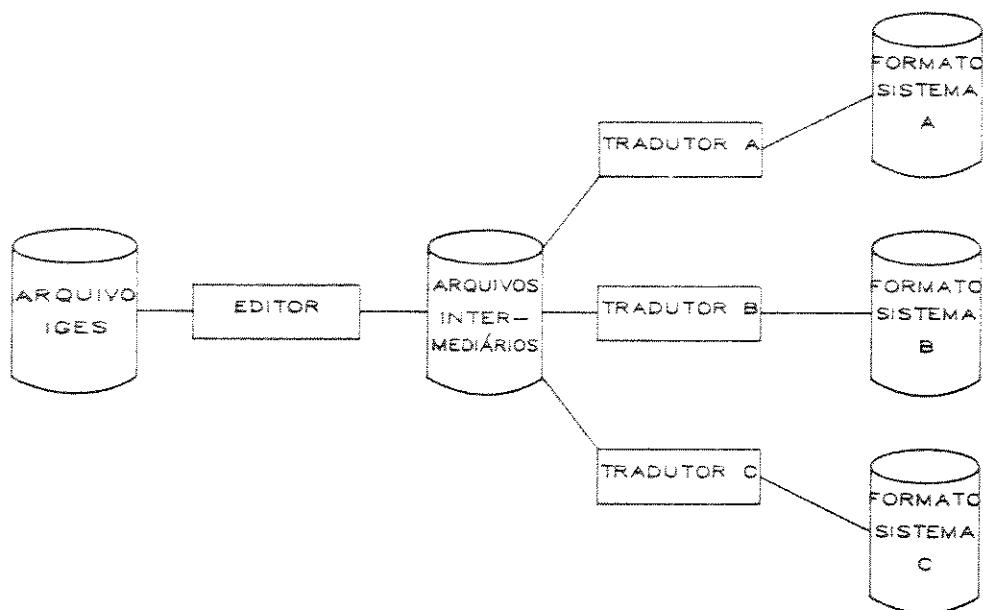


Fig. 5.4 - Implementação de pré e pós-processadores IGES

O perfil da aplicação IGES em TOP estabelece algumas restrições em relação à versão 3.0 de IGES, a fim de garantir a interoperabilidade entre os sistemas envolvidos. As principais são:

- deve ser usado o formato ASCII, não sendo permitidos os formatos binário e ASCII comprimido;
- os dados gerados por pré-processadores devem ir para arquivos em disco, não sendo permitidas fitas magnéticas como meio físico para armazenamento;
- devem ser seguidas instruções que constam no documento "IGES Recommended Practices Guide" /IGE 85/. Este visa resolver ambigüidades que a prática de implementação de pré e pós-processadores revelou;
- o mínimo conjunto de entidades que deve ser implementado precisa conter aquelas utilizadas em sistemas bidimensionais ou sistemas tridimensionais baseados em modelos de arame. Este conjunto é listado na Tabela 5.1. Outro fato a ser observado é que a especificação não suporta acesso remoto a arquivos IGES, mas, somente, transferência destes.

O documento da versão 3.0 de TOP prevê a incorporação do padrão STEP tão logo este atinja estabilidade suficiente. Conforme será visto na seção seguinte, a implementação dos diversos níveis previstos em STEP (ver item 4.3.4) em um ambiente distribuído exigirá um suporte adicional no que se refere aos protocolos da camada de aplicação.

NÚMERO DE TIPO DA ENTIDADE	NOME DA ENTIDADE
116	PONTO
110	LINHA
100	ARCO CIRCULAR
112	SPLINE PARAMETRICA
126	B-SPLINE RACIONAL
104	ARCO CONICO
124, FORM 0	MATRIZ DE TRANSFORMACAO
212	NOTA GERAL
202	DIMENSAO ANGULAR
206	DIMENSAO DIAMETRO
210	ROTULO GERAL
214	TERMINADOR
216	DIMENSAO LINEAR
222	DIMENSAO RADIAL
106, COPIOUS DATA FORM 40	LINHA DE CHAMADA

TABELA 5.1 - Conjunto mínimo de entidades IGES especificado em TOP

5.3. COMPARTILHAMENTO DE DADOS VIA STEP USANDO REDES TOP

A forma de cooperação entre sistemas que compartilham informações sobre produtos, fazendo uso do padrão STEP, depende do nível de implementação deste nos sistemas envolvidos e de um suporte de comunicação que garanta a funcionalidade necessária à interação entre sistemas heterogêneos. Como um nível de implementação deve também suportar os níveis inferiores, dois sistemas poderão interagir no nível mais baixo entre eles. Feitas estas considerações, serão apresentadas a seguir arquiteturas de sistemas que suportam os vários níveis de implementação STEP.

5.3.1. Implementação STEP Nível 1

No que concerne aos protocolos da camada de aplicação em redes TOP, todas as observações feitas para o caso de troca de dados via arquivo IGES valem para a troca por arquivo neutro STEP. Quanto à implementação de pré e pós-processadores, três aspectos da especificação do arquivo neutro STEP devem ser considerados: sua sintaxe é descrita formalmente, não são permitidas referências a entidades ainda não definidas no arquivo e são definidas regras para o mapeamento entre a estrutura de dados que o esquema conceitual STEP define e o arquivo neutro. A Fig. 5.5 apresenta uma arquitetura para pré e pós-processadores que exploram as características acima. No caso de pré-processadores, uma vez gerada uma estrutura de dados em conformidade com o esquema conceitual STEP pelas rotinas específicas para o sistema local, um conjunto genérico de rotinas que implementa o mapeamento acima descrito será responsável pela geração do arquivo neutro. A primeira função de pós-processamento consiste em um analisador léxico que lê palavras do arquivo neutro e verifica sua validade, passando-as ao analisador sintático. Este opera ao nível de declarações do arquivo neutro, verificando sua integridade e construindo uma árvore sintática que representa a declaração, usando, para isso, a gramática expressa em WSN ("Wirth Syntax Notation", ver 4.3.3). O analisador semântico acessa a árvore e gera uma estrutura de dados compatível com o esquema conceitual STEP. É sua função, também, verificar se os tipos das entidades referenciadas são aqueles previstos no esquema STEP. Como todas as entidades referenciadas precisam já ter sido definidas, o analisador semântico somente manterá uma tabela que associe identificadores de entidade no arquivo neutro com as ocorrências destas na estrutura de dados. A partir deste ponto, rotinas específicas para o sistema local fazem a conversão entre a estrutura de dados STEP e a estrutura local e geram o formato de dados local.

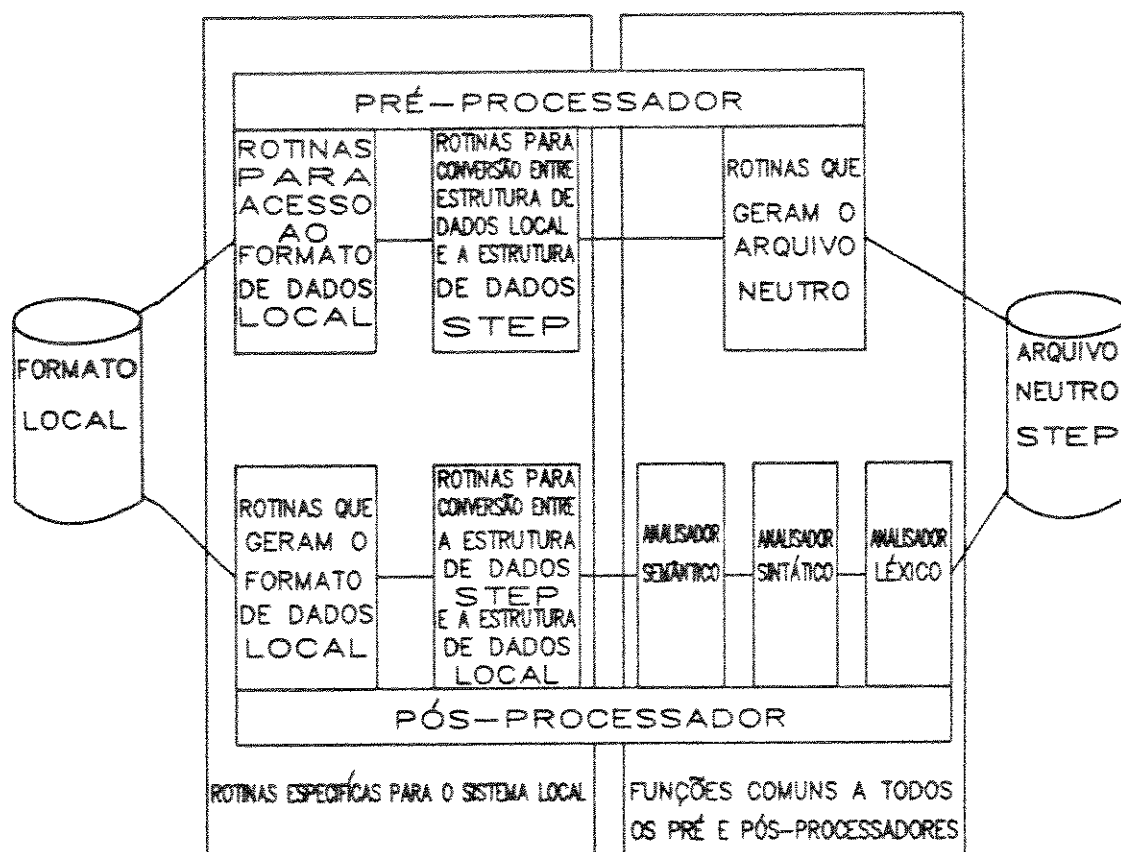


Fig. 5.5 - Arquitetura de pré e pós-processadores STEP

5.3.2. Implementação STEP Nível 2

Em uma implementação STEP nível 2 os dados de produto a serem trocados entre sistemas são traduzidos de/para uma forma operacional. A principal característica desta é o fato de poder ser acessada através de chamadas de funções padronizadas. A padronização deste conjunto de funções deve se dar em termos da funcionalidade requerida para cada uma delas e pela definição da sintaxe para uma ou mais linguagens de programação (este procedimento é análogo à definição de "bindings" para o padrão GKS). Basicamente estas funções devem permitir a uma aplicação criar, atualizar e consultar dados na forma operacional. Como mostra a Fig. 5.6 a interação entre uma aplicação e a forma operacional em um sistema distribuído pode ser modelada por um processo servidor e um processo cliente. Este, desejando receber

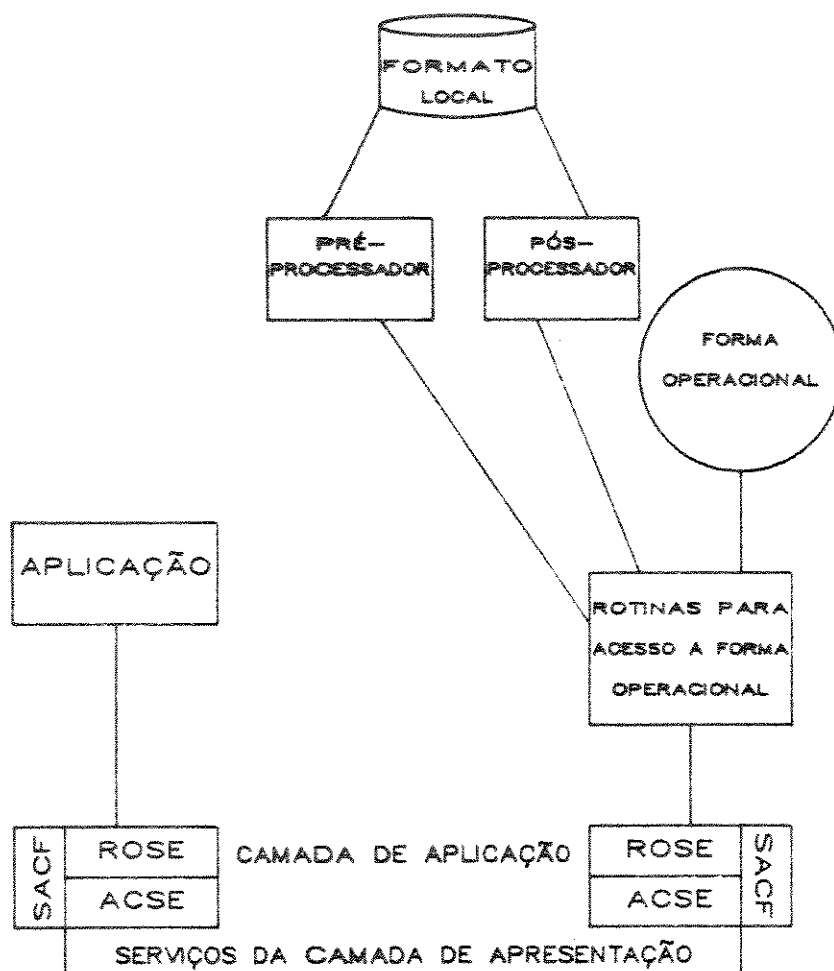


Fig. 5.6 - Troca de dados de produtos em implementação nível 2 - STEP

ou enviar informações de produtos, o fará através de chamadas de funções que são executadas remotamente. A funcionalidade requerida para essas chamadas de procedimentos remotos, ou seja, iniciar operações e receber resultados de um nó remoto, é provida pelo protocolo da camada de aplicação denominado ROSE ("Remote Operations Service Element") /PAP 87/, /HAL 88/. O protocolo ACSE é responsável pelo estabelecimento e liberação de associações entre entidades da camada de aplicação, enquanto a função de controle de associação (SACF) coordena as interações entre os protocolos de aplicação envolvidos em uma associação (no caso ROSE e ACSE) e mapeia as unidades de dados de protocolos de aplicação (A-PDU's) na camada de apresentação /BEL 88/. A

tradução entre o formato de dados local e a forma operacional é feita através de pré/pós-processadores que, para isso, utilizam o conjunto de funções padronizadas. A principal vantagem de implementações neste nível, em relação à troca por arquivo neutro, do ponto de vista de uma aplicação remota que deseja enviar ou receber dados de produtos, diz respeito ao fato de que esta tem que lidar somente com listas de parâmetros das funções e, não, com a sintaxe do arquivo neutro. Além disto, a aplicação pode requerer desde informações sobre uma única entidade até um modelo de produto inteiro /SHI 88/.

5.3.3. Implementação STEP Nível 3

Um sistema que suporte uma implementação nível 3 de STEP deve manter os dados de produto a serem compartilhados com outros sistemas em uma base de dados cujo esquema conceitual possa ser mapeado de/para o modelo integrado de informações de produto (IPIM) definido pelo padrão STEP. Um sistema distribuído em que todos os componentes suportem este nível de implementação possibilita a forma de cooperação mais efetiva entre os sistemas, em que informações corretas sobre produtos se encontram prontamente disponíveis a qualquer um deles (ver Cap. 2). Para isto, a interação entre os sistemas deve ser administrada por um sistema gerenciador de bases de dados distribuídas (SGBDD). Através deste, os sistemas gerenciadores de bases de dados locais (SGBD's locais) são interligados usando uma rede de comunicação de dados, de tal forma que qualquer aplicação/usuário em um nó da rede tenha uma visão integrada dos dados distribuídos, possibilitando o acesso a estes sem conhecer como são armazenados fisicamente e sua fragmentação e replicação ao longo do sistema distribuído /CAS 85/, /CER 84/. Além disto, cada sistema deve ter autonomia local, ou seja, cada SGBD local deve manter controle sobre seus próprios dados e acessos a dados locais devem ser executados localmente, sem que seja necessário consultar outros nós. Como o caso prático, em empresas voltadas à manufatura de produtos, envolve a integração de sistemas pré-existentes, caracteriza-se um SGBDD heterogêneo em que os modelos de dados

locais podem ser diferentes entre si. Para lidar com esta questão e evitar que um único passo defina o critério de distribuição de dados e a estratégia para seu armazenamento em cada nó, a Fig. 5.7 propõe uma arquitetura em cinco camadas para a base de dados distribuída /LAN 86/. Esquemas externos globais definem visões particulares que usuários/aplicações têm do banco de dados, o qual é definido a nível lógico pelo esquema conceitual global. No caso do problema de interesse deste trabalho, este é descrito pelo modelo integrado de informações de produto (IPIM) definido por STEP. Os esquemas internos globais especificam o particionamento e replicação dos dados ao longo dos nós do sistema distribuído. Estes são equivalentes aos esquemas conceituais locais, sendo, porém, descritos no mesmo modelo de

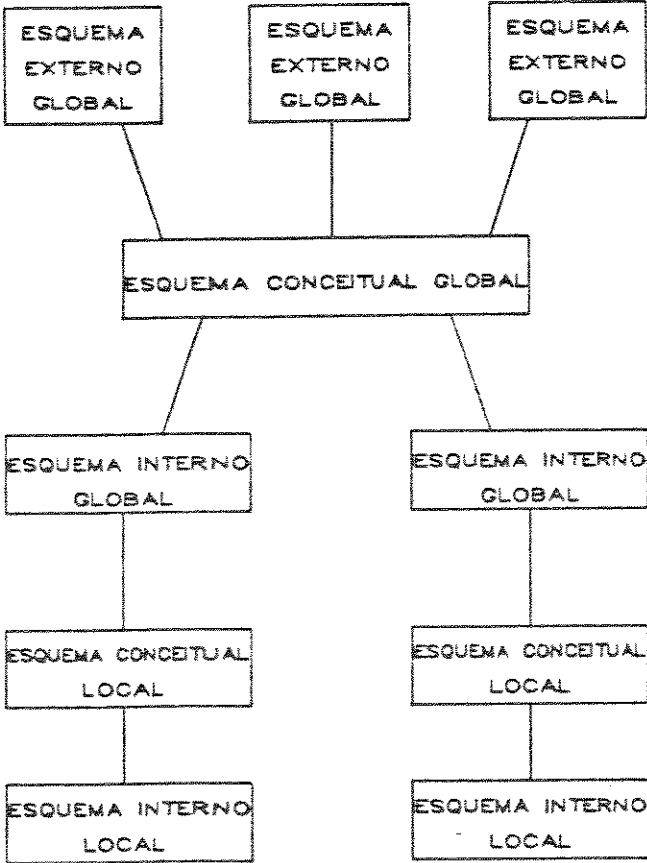


Fig. 5.7 - Arquitetura para uma base de dados distribuída heterogênea

dados que os esquemas externos globais e conceitual global. Os esquemas conceituais locais, descritos usando os modelos de dados locais, especificam a nível lógico os bancos de dados locais, enquanto os esquemas internos locais definem sua forma de armazenamento. Ambos serão manipulados somente pelo SGBD local. Para a descrição dos níveis globais e dos mapeamentos entre eles, o SGBDD deve prover uma linguagem neutra de definição de dados (NLDD). Para o acesso e entrada de dados por usuários/aplicações, faz-se também necessária uma linguagem neutra de manipulação de dados (NLMD). Esta deve ser não-procedural, a fim de que as aplicações precisem somente especificar que dados devem ser acessados e não como fazê-lo. Além disto, a linguagem deve ser orientada à manipulação de conjuntos e não registros individuais, pois isto implica na diminuição do tráfego de mensagens na rede. Outra questão fundamental a ser considerada no que diz respeito a ambas as linguagens (NLDD e NLMD) é que estas devem ser baseadas em um modelo de dados que possa ser mapeado para os modelos locais (normalmente relacional, rede ou hierárquico). Neste sentido, alguns trabalhos de pesquisa têm proposto a utilização de modelos semânticos de dados, em que as informações são descritas em um nível de abstração mais alto que nos modelos convencionais /STA 85/, /HAW 86/, /ALT 86/, /APP 85/, /CAR 87/. O modelo a ser utilizado para a definição de IPIM como esquema conceitual global deve, idealmente, ser capaz de capturar toda a semântica definida com a linguagem Express.

O suporte necessário à implementação do SGBDD, em termos de comunicação de dados, é provido pelo protocolo da camada de aplicação chamado RDA ("Remote Database Access"). Este fornece os serviços que possibilitam a uma aplicação acesso a uma base de dados em um nó remoto. Como mostra a Fig. 5.8, estes podem ser modelados pelos seguintes componentes:

- um processo cliente que deseja manipular dados que se encontram em uma base de dados remota;
- um processo servidor que traduz as consultas feitas em uma linguagem de manipulação de dados padrão em chamadas de procedimentos que são submetidos ao SGBD local. Além

disto, este processo retorna resultados ao processo cliente usando os serviços RDA;

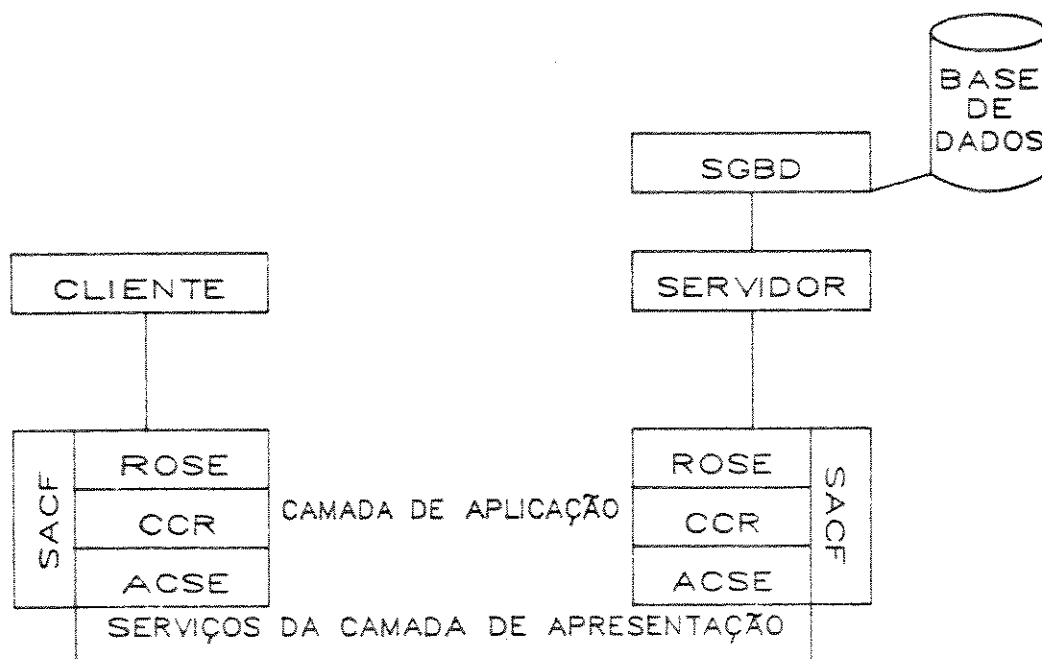


Fig. 5.8 - Modelo de interação em RDA

- entidades de aplicação RDA: Estas utilizam o protocolo ACSE (ver 5.3.2), o protocolo ROSE para as chamadas de procedimento remoto que as declarações na LMD padrão demandam e o protocolo CCR ("Commitment, Concurrency and Recovery") que gerencia transações distribuídas. Uma transação é uma coleção de comandos que deve ser processada como se fosse atômica, ou seja, ou todos os comandos são processados ou nenhum deles /CAS 85/. Além disto, uma transação sempre deve levar o banco de dados de um estado consistente a outro estado consistente, ou seja, transações concorrentes devem ser processadas como se cada uma delas fosse executada uma após a outra. É função do SGBD prover mecanismos para este controle de concorrência e para o controle de integridade associado à atomicidade de transações, cabendo ao protocolo CCR somente iniciar transações em um nó remoto e implementar procedimentos que

confirmem que o estado final de uma transação foi atingido ou que o banco de dados voltou a seu estado inicial. O diagrama da Fig. 5.9 mostra o protocolo bifásico utilizado em CCR com esta finalidade [HAL 88]. Na primeira fase, após as operações associadas a uma transação serem completadas, o processo cliente consulta o servidor para saber se este obteve sucesso ou não. Caso responda positivamente, o cliente solicita que sejam completadas as ações que levam o banco de dados no servidor ao estado final da transação. Este confirma ao cliente o procedimento. Caso responda negativamente, não responda, ou o cliente deseje que o banco de dados retorne ao estado estado inicial, este solicita ao servidor que providencie as ações necessárias para este retorno. Este protocolo permite, portanto, que o processo cliente gerencie transações que envolvem mais do que um processo servidor, o que é bastante comum em bases de dados distribuídas.

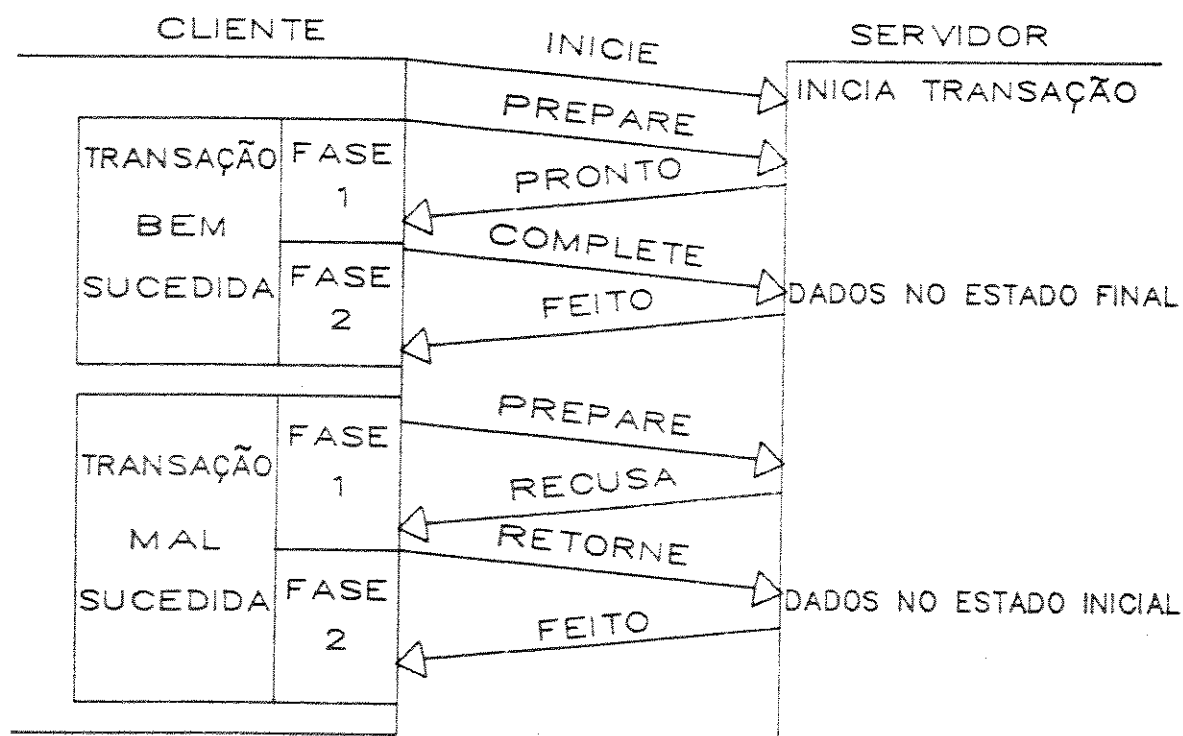


Fig. 5.9 - Diagrama temporal do protocolo bifásico em CCR

Considerando-se que cada nó da rede suporte uma implementação nível 3 de STEP, ou seja, mantenha uma base de dados em conformidade com o esquema conceitual STEP, a Fig. 5.10 apresenta a arquitetura de um sistema que integra efetivamente os dados de produtos fisicamente distribuídos ao longo dos nós. Este sistema deve manter um dicionário de dados global que descreve os níveis externo, conceitual e interno global e os mapeamentos entre eles. Este é criado usando-se a NDDL do sistema. Como qualquer outro banco de dados no sistema, o dicionário de dados pode estar distribuído entre os nós. Porém, como é acessado frequentemente, justifica-se sua replicação em cada um deles. As funções de cada um dos componentes do sistema são os seguintes:

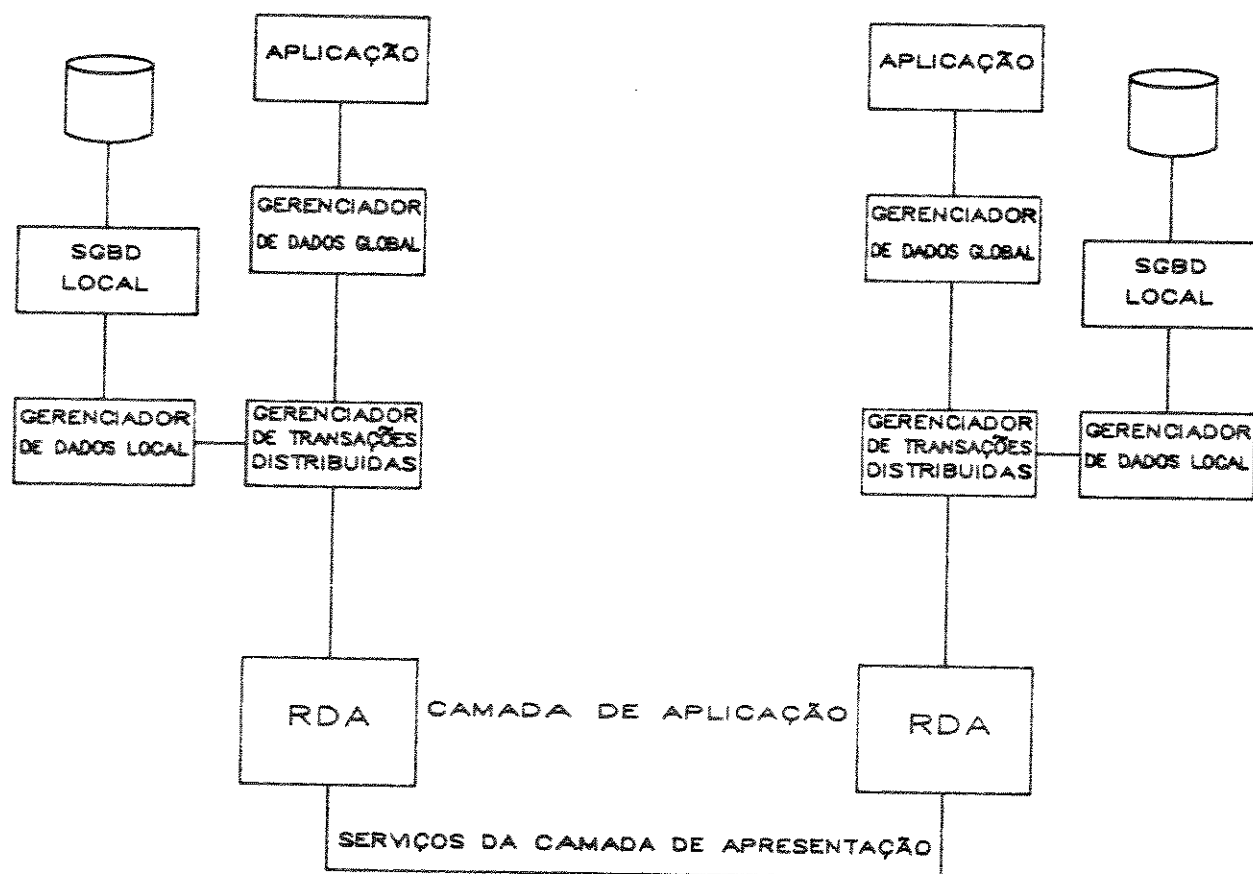


Fig. 5.10 - Arquitetura de sistema para compartilhamento de dados de produtos

- Gerenciador de Dados Global

- ao receber transações solicitadas por uma aplicação, as quais contêm comandos escritos na NDML do sistema em termos de um esquema externo global, acessa o dicionário de dados para realizar a tradução destes comandos em termos do esquema conceitual global;
- consultando o esquema conceitual global e os esquemas internos globais descritos no dicionário de dados, além do mapeamento entre eles, fragmenta o comando em sub-comandos referentes aos nós que serão acessados para a execução deste. Neste ponto, cada sub-comando estará escrito em termos do esquema interno local que lhe diz respeito;
- gera um plano de execução para os sub-comandos, o qual visa a otimização global do processamento do comando. Para isso considera, entre outros aspectos, quais sub-comandos devem preceder outros, quais podem ser executados em paralelo e qual o relacionamento entre os resultados intermediários que os sub-comandos geram. Por outro lado, já deve ter sido considerada na etapa de fragmentação a questão de replicação de dados, em que se pode escolher mais do que um nó da rede para o processamento de uma consulta. De uma forma geral, deve-se procurar minimizar o tráfego de dados na rede;
- agrega os resultados recebidos dos sub-comandos, representando-os em termos do esquema conceitual global. Além disto, realiza a tradução destes para o esquema externo global e entrega-os à aplicação;

- Gerenciador de Transações Distribuídas:

- monitora o início e término das atividades do sistema no nó em que se encontra;
- é responsável pelo controle de concorrência e pelo controle de integridade global do sistema. Para isso, os

gerenciadores utilizam o protocolo bifásico que RDA implementa;

- recebe comandos do gerenciador de dados global. Se o comando for dirigido ao mesmo nó, este é repassado ao gerenciador de dados local. Caso contrário, utiliza os serviços RDA para submeter o comando em um nó remoto, no qual o gerenciador de transações distribuídas acionará o gerenciador de dados local para execução;
- uma vez recebidos os resultados dos comandos submetidos, repassa-os ao gerenciador de dados global;

- Gerenciador de Dados Local

- deve gerar uma sequência ótima de comandos na linguagem de manipulação de dados do SGBD local que traduza a ação requerida no comando que recebeu, o qual é definido em termos do esquema interno global usando a NLMD do sistema;
- uma vez processado o comando, deve traduzir os resultados para o esquema interno global correspondente e entregá-los ao gerenciador de transações distribuídas.

CAPITULO 6

CONCLUSÕES

Os requisitos impostos ao modelo de produto necessário à consecução do objetivo de integração dos sistemas envolvidos no ciclo de projeto e manufatura de produtos podem ser assim sintetizados:

- 1) deve ser um modelo comum às aplicações que defina, em termos de estruturas, restrições e operações sobre os dados, as propriedades semânticas destes, de tal forma que todas as informações relativas a produtos a serem compartilhadas entre os sistemas tenham um significado preciso, possibilitando, assim, que um sistema opere sobre dados recebidos de outros;
- 2) deve ser capaz de representar toda a gama de produtos da empresa;
- 3) deve ser não-ambíguo, de tal forma que todas as informações requeridas pelas aplicações possam dele ser extraídas;
- 4) deve ser tal que todas as representações sintaticamente corretas correspondam a produtos fisicamente realizáveis, uma vez que devem ser suportadas aplicações sem a intervenção do usuário;
- 5) deve permitir que informações sejam modeladas em múltiplos níveis de abstração, correspondentes aos requisitos de diferentes aplicações.

A viabilização econômica de sistemas CIM exige que seus componentes sejam sistemas abertos, aptos, a priori, a cooperar uns com os outros. Como estes componentes são, normalmente, sistemas heterogêneos, evidencia-se a necessidade de padronizar um modelo de produto que satisfaça os requisitos acima. O esquema conceitual definido por STEP tem este objetivo, contemplando, porém, na presente versão, somente as informações para definição do produto. A linguagem Express, com que é descrito o esquema conceitual, provê os elementos necessários à satisfação do primeiro requisito. O segundo requisito é atendido pela

possibilidade de serem representados desde produtos com formas geométricas regulares até aqueles que apresentam superfícies de dupla curvatura. O suporte à representação de modelos sólidos e modelos de tolerâncias garante, conforme foi visto no capítulo três, a satisfação do terceiro e quarto requisitos. Além disto, modelos de formas características permitem representar informações em um nível de abstração compatível com as funções de manufatura, o que vem de encontro ao último requisito.

Uma questão ainda não resolvida em STEP é a definição de um esquema para representação de modelos paramétricos e entidades criadas pelo usuário através de macros. Além disto, novos modelos de referência para aplicações devem ser integrados ao esquema conceitual e a representação de informações de suporte ao produto deve ser considerada.

Como foi observado no capítulo 4, STEP prevê quatro níveis de implementação. O quarto nível, que envolve o mapeamento de regras de uma base de conhecimento em outra, não foi analisado neste trabalho, constituindo-se em um dos tópicos a serem considerados em sua continuidade. À exceção das regras para consistência dos dados, o sistema gerenciador de bases de dados distribuídas proposto no capítulo cinco, em que são integradas bases de dados locais que correspondem a implementações do nível três de STEP, provê o suporte computacional necessário ao efetivo compartilhamento de informações entre sistemas componentes de CIM. Claramente, a implementação deste sistema é um projeto ambicioso, dando lugar a vários temas de pesquisa, entre os quais podem ser citados:

- algoritmos para controle de integridade e controle de concorrência global do sistema;
- especificação de uma linguagem de manipulação de dados que opere sobre as estruturas de dados que podem ser definidas pela linguagem Express;

- mapeamento de comandos desta linguagem em comandos de linguagens de manipulação de dados baseadas em modelos convencionais (relacional, rede e hierárquico).

É de se esperar que a linguagem de manipulação de dados a ser padronizada em STEP leve em consideração as questões acima. A definição desta interface significa, em termos da arquitetura do sistema gerenciador de bases de dados distribuídas que integra os dados de produto, que se poderá requisitar de um fornecedor de sistema um gerenciador de dados local aderente ao padrão. Quanto aos aspectos relativos à comunicação de dados, a especificação de RDA como um dos protocolos da camada de aplicação em redes TOP garante os serviços necessários à interligação dos sistemas onde se encontram as bases de dados.

Implementações de STEP do nível dois exigirão a padronização de um conjunto de rotinas, a serem usadas pelas aplicações para operar sobre o modelo de produto. Para que estas possam acessar modelos em nós remotos, a camada de aplicação da rede TOP deve prover os serviços ROSE.

A troca de dados por arquivo neutro utilizando o padrão IGES é a estratégia de cooperação entre sistemas até hoje adotada, na prática. Em relação a esta, a troca por arquivo neutro STEP, que corresponde a implementação nível um, apresenta a vantagem de que os dados presentes no arquivo têm o mesmo significado para todos os sistemas, uma vez que este é definido formalmente pelo esquema conceitual STEP. Em ambos os casos, serviços do protocolo de aplicação FTAM podem ser usados para transferência de arquivos entre nós de redes TOP. Porém, implementações neste nível não se constituem em uma forma de integração do sistema de manufatura, uma vez que o usuário continua sendo responsável pela consistência dos dados a nível global.

BIBLIOGRAFIA

- /ALT 86/ Althoff, J. & Shumski, J. Integrated information support system, common data model subsystem. IN: Autofact 86, 14.9-14.13, 1986.

- /AME 85/ The interface challenge. American Machinist, 129(1): 95-102, Jan/85.

- /AME 87/ Database management: gateway to CIM. American Machinist, 131(10):81-88, Oct/87.

- /APP 85/ Appleton, D.S. The technology of data integration. Datamation, 31(21):106-116, Nov/85.

- /BAE 79/ Baer, A.; Eastman, C. & Henrion, M. Geometric modeling: a survey. Computer Aided Design, 11(5):253-272, Sep/79.

- /BAR 84/ Barsky, B.A. A description and evaluation of various 3-D models. IEEE Computer Graphics and Applications, 4(1):38-52, Jan/84.

- /BEL 87/ Below, L.J. The meaning of integration. IN: Proceedings of the 4th European Conference on Automated Manufacturing, Springer-Verlag, 345-351, 1987.

- /BEL 88/ Belisle, P. & Jansonn, H. OSI - what is next? IN: Proceedings of Enterprise Network Event, Baltimore, Eva, 4.1-4.27, 1988.

- /BER 85/ Bernat, A.R. Applying CAD/CAM to automotive body development. Computer Aided Design, Engineering and Drafting, Auerbach Publishers, 1985.

- /BEY 88/ Bey, I. & Gengenbach, U. The CAD*I interface for solid model exchange, Computer & Graphics, 12(2):181-190, 1988.

- /BEZ 74/ Bezier, P. Mathematical and practical possibilities of Unisurf. IN: Computer Aided Geometric Design, Academic Press, 127-152, 1974.
- /BJO 84/ Bjorke, O. Towards integrated manufacturing systems - manufacturing cells and their subsystems. Robotics and Computer Integrated Manufacturing, 1(1):3-19, 1984.
- /BLI 84/ Bliss, F.W. A survey of CAD/CAM modeling techniques. Computer Aided Design, Engineering and Drafting, Auerbach Publishers, 1984.
- /BOH 84/ Bohm, W.; Farin, G. & Kahmann, J. A survey of curve and surface methods in CAGD. Computer Aided Geometric Design, 1(1):1-60, Jul/84.
- /BOR 80/ Borgerson, B.R. & Johnson, R.H. Beyond CAD to Computer Aided Engineering. IN: Proceedings of the 3th World Computer Congress, Tôquio, Japão, Out/80.
- /BOY 82/ Boyse, J.W. & Gilchrist, J.E. GM Solid: interactive modeling for design and analysis of solids. IEEE Computer Graphics and Applications, 2(2):27-42, Mar/82.
- /BRA 79/ Braid, I.C. Notes on a geometric modeler, CAD Group document number 101, University of Cambridge, Jun/79.
- /BRA 87/ Bray, O.H. Data management for engineering: current capabilities, future directions. Computer Aided Engineering Journal, 4(2):20-24, Mar/87.
- /BRO 82/ Brown, C.M. PADL-2: a technical summary. IEEE Computer Graphics and Applications, 2(2):69-84, Mar/82.
- /CAR 87/ Cardenas, A.F. Heterogeneous distributed database management: the HD-DBMS. IN: Proceedings of the IEEE, 75(5):588-600, Mai/87.

- /CAS 85/ Casanova, M.A. & Moura, A.V. Princípios de sistemas de gerência de bancos de dados distribuídos, Editora Campus, 1985.

- /CER 84/ Ceri, S. & Pelagatti, G. Distributed databases principles and systems, McGraw-Hill, 1984.

- /CHA 85/ Chang, T.C. & Wysk, R.A. An introduction to automated process planning systems, Prentice Hall, 1985.

- /CON 85/ Conaway, J. What's in a name: plain talk about CIM. Computers in Mechanical Engineering, 4(3):23-31, Nov/85.

- /COO 67/ Coons, S.A. Surfaces for computer aided design of space forms, Project MAC, MIT, 1967.

- /DAT 82/ Date, C.J. An introduction to database systems, terceira edição, Addison-Wesley, 1982.

- /EAS 79/ Eastman, C.M. & Weiler, K. Geometric modeling using the Euler operators, Institute of Physical Planning Research, Report number 78, Carnegie-Mellon University, 1979.

- /ELG 85/ Elgabry, A.K. Communicating product definition and support data in a CAE/CAD/CAM environment. IN: Autofact 85, 9.11-9.26, 1985.

- /ENC 86/ Encarnação, J.L.; Schuster, R. & Voge, E. Product data interfaces in CAD/CAM applications, Springer-Verlag, 1986.

- /FOL 84/ Foley, J.D. & Dam, A.V. Fundamentals of interactive computer graphics, Addison-Wesley, 1984.

- /FAU 79/ Faux, I.D. & Pratt, M.J. Computational geometry for design and manufacture, John Wiley & Sons, 1979.

- /GIB 88/ Gibbons, A.J. A standards framework for the Computer Integrated Enterprise. CIM Review, 4(3):10-17, 1988.

- /GOR 74/ Gordon, W.J. & Riesenfeld, R.F. B-splines curves and surfaces. IN: Computer Aided Geometric Design, Academic Press, 95-126, 1974.

- /GRA 87/ Grabowski, H. & Glatz, R. IGES model comparison system: a tool for testing and validating IGES processors. IEEE Computer Graphics and Applications, 7(11):47-57, Nov/87.

- /HAL 88/ Halsall, F. Data communications, computer networks and OSI, Addison-Wesley, 1988.

- /HAW 86/ Hawk, M. IISS technical overview. IN: Autofact 86, 14.5-14.7, 1986.

- /HEN 84/ Henderson, M.R. & Anderson, D.C. Computer recognition and extraction of form features: a CAD/CAM link. Computers in Industry, 5(4), 329-339, Dez/84.

- /HIL 82/ Hillyard, R. The Build group of solid modelers. IEEE Computer Graphics and Applications, 2(2):43-52, Mar/82.

- /HOJ 88/ Hojnicky, J. & White, P.R. Converting CAD wireframe data to surfaced representations. Computers in Mechanical Engineering, 6(5):19-25, Mar/88.

- /IGE 85/ IGES recommended practices guide, National Bureau of Standards, Jun/85.

- /IGE 86/ Initial Graphics Exchange Specification, versão 3.0, National Bureau of Standards, Abr/86.

- /ING 85/ Ingersoll Engineers Integrated Manufacture, Springer-Verlag, 1985.

- /JOS 88/ Joshi, S. & Chang, T.C. Graph-based heuristics for recognition of machined features from a 3D solid model. Computer Aided Design, 20(2):58-66, Mar/88.

- /KIM 84a/ Kimura, F. Geomap-III: designing solids with free-form surfaces. IEEE Computer Graphics and Applications, 4(6):58-72, Jun/84.

- /KIM 84b/ Kimura, F.; Kawabe, S. & Sata, T. A study on product modeling for integration of CAD/CAM. Computers in Industry, 5(3):239-252, Sep/84.

- /KLE 88/ Klement, K. & Nowacki, H. Exchange of model presentation information between CAD systems. Computer & Graphics, 12(2):173-180, 1988.

- /KNA 84/ Knapp, L.C. Surface definition techniques for solid modeling. Computer Aided Design, Engineering and Drafting, Auerbach Publishers, 1984.

- /LAM 86/ Lambourne, E.B. Towards integration of computer aided design, manufacture and production management. Computer Aided Engineering Journal, 3(6):240-244, Dez/86.

- /LAN 86/ Landers, T. & Rosenberg, R.L. An overview of Multi-base. IN: Distributed Systems, vol. II: Distributed Data Base Systems, Artech House, 391-421, 1986.

- /LEE 87/ Lee, Y.C. & Fu, K.S. Machine understanding of CSG: extraction and unification of manufacturing features. IEEE Computer Graphics and Applications, 7(1):20-32, Jan/87.

- /LIE 85/ Liewald, M.H. Initial Graphics Exchange Specification: successess and evolution. Computer & Graphics, 9(1): 47-50, 1985.

- /MAL 85/ Malraison, P.J. The importance of topological information in CAD/CAM modeling. Computer Aided Design, Engineering and Drafting, Auerbach Publishers, 1985.

- /MAN 82a/ Mantyla, M. An inversion algorithm for geometric models. ACM Computer Graphics, 16(3):51-59, Jul/82.

- /MAN 82b/ Mantyla, M. & Sulonen, R. GWB: a solid modeler with Euler operators. IEEE Computer Graphics and Applications, 2(7):17-31, Set/82.

- /MAN 87/ Mantyla, M. Solid modeling: state of the art and the next steps. IN: MICAD 87, Paris, França, 1987.

- /MAN 88/ Mantyla, M. An introduction to solid modeling, Computer Science Press, 1988.

- /MAR 80/ Markowski, G. & Wesley, M.A. Fleshing out wire-frames. IBM Journal of Research and Development, 24(5): 582-597, 1980.

- /MAR 87/ Marks, P. What do solid models need? Machine Design, 59(5):139-143, Mar/87.

- /MAY 87/ Mayer, R.J. IGES: one answer to the problems of CAD database exchange. Byte, 12(6):209-214, Jun/87.

- /MEA 82/ Meagher, D. Geometric modeling using octree encoding. Computer Graphics and Image Processing, 19:129-147, Jun/82.

- /MEN 89/ Mendes, M.J. Comunicação fabril e o projeto MAP/TOP, quarta escola brasileiro-argentina de informática, Jan/89.

- /MOR 85/ Mortenson, M.E. Geometric modeling, John Wiley & Sons, 1985.

- /MYE 82/ Myers, W. An industrial perspective on solid modeling. IEEE Computer Graphics and Applications 2(2):86-97, Mar/82.
- /NOW 88/ Nowacki, H. Uebertragung von Produkt-beschreibungen ueber Rechnernetze. Informationstechnik, 30(5): 349-356, 198.
- /OWE 87/ Owen, J. & Bloor, M.S. Neutral formats for product data exchange: the current situation. Computer Aided Design, 19(8):436-443, Out/87.
- /PAP 87/ Pappe, S.; Effelsberg, W. & Hamersdorf, W. Database access in open systems. IN: Networking in open systems, Lecture Notes in Computer Science, 148-164, 1987.
- /PAS 88/ Pasquill, K.G. Practical uses of IGES. Computer Aided Design, 20(6):358-359, Ago/88.
- /PAT 87/ Patel, R.M. & Mchcod, A.J. Engineering feature description in mechanical engineering design. Computer Aided Engineering Journal, 5(5):180-183, Out/88.
- /PET 86/ Pettinati, F. & Rimoldi, A. Solid modeling using an intelligent workstation. IN: Anais do 6o. Seminário de Comando Numérico, São Paulo, Brasil, Ago/86.
- /PIE 87/ Piegl, L. & Tiller, W. Curve and surface constructions using rational B-splines. Computer Aided Design, 19(9): 485-498, Nov/87.
- /PRA 84/ Pratt, M.J. Solid modeling and the interface between design and manufacture. IEEE Computer Graphics and Applications, 4(7):52-59, Jul/84.
- /RAL 87/ Ralston, D. & Munton, T. Computer integrated manufacturing. Computer Aided Engineering Journal, 4(4): 167-175, Ago/87.

- /REQ 80/ Requicha, A.A.G. Representations for rigid solids: theory, methods and systems. ACM Computing Surveys, 12(4):437-464, Dez/80.
- /REQ 82/ Requicha, A.A.G. & Voelcker, H.B. Solid modeling: a historical summary and contemporary assesment. IEEE Computer Graphics and Applications, 2(2):9-24, Mar/82.
- /REQ 83/ Requicha, A.A.G. & Voelcker, H.B. Solid modeling: current status and research directions. IEEE Computer Graphics and Applications, 3(7):25-37, Out/83.
- /ROG 76/ Rogers, D.F. & Adams, J.A. Mathematical elements for computer graphics, McGraw-Hill, 1976.
- /ROY 88/ Roy, U. & Liu, C.R. Feature-based representational scheme of a solid modeler for providing dimensioning and tolerancing information. Robotics and Computer Integrated Manufacturing, 4(3):335-345, 1988.
- /RUI 88/ Rui, A. et al Automating information transfer in manufacturing systems. Computer Aided Engineering Journal, 5(3):113-120, Jun/88.
- /SCH 87a/ Schlechtendahl, E.G. Specification of a CAD*I neutral file for CAD geometry, versão 3.2, Springer-Verlag, 1987.
- /SCH 87b/ Schlechtendahl, E.G. CAD data transfer: the goals and achievements of project 322, CAD Interfaces (CAD*I). IN: ESPRIT 87, North-Holland, 1547-1569, 1987.
- /SCH 88/ Schenk, D. Information modeling language Express, ISO TC 184/SC4/WG1, Out/88.
- /SHA 88/ Shah, J.J. & Rogers, M.T. Functional requirements and conceptual design of feature-based modeling system. Computer Aided Engineering Journal, 5(1):9-15, Fev/88.

- /SHI 88/ Shi, C.H. & Barker, S.L. PDES and IISS: implementations issues and requirements. IN: Proceedings of Enterprise Network Event, Baltimore, EUA, 2.109-2.128, 1988.

- /SMI 87/ Smith, B.M. CAD/CAM data exchange leaps beyond IGES. The S. Klein Computer Graphics Review, 79-80, Winter, 1987.

- /STA 85/ Stankovic, J.A.; Ramamritham, K. & Kohler, W.H. A review of current research and critical issues in distributed system software. IN: IEEE Distributed Processing Technical Committee Newsletter, 7(1):14-47, Mar/85.

- /STE 88a/ STEP first working draft, ISO TC 184/SC4/WG1, Nov/88.

- /STE 88b/ ISO STEP baseline requirements document (IPIMD), ISO TC 184/SC4/WG1, documento no. 284, Out/88.

- /STE 88c/ The STEP file structure, ISO TC 184/SC4/WG1, documento no. 279, Set/88.

- /STE 88d/ Mapping from Express to physical file structure, ISO TC 184/SC4/WG1, documento no. 280, Set/88.

- /STE 88e/ PDES/STEP implementation levels, ISO TC184/SC4/WG1, documento no. 282, Out/88.

- /SZW 84/ Szwarcfiter, J.L. Grafos e algoritmos computacionais, Editora Campus, 1984.

- /TIL 80a/ Tilove, R.B. & Requicha, A.A.G. Closure of Boolean operations on geometric entities. Computer Aided Design, 12(5):219-220, Set/80.

- /TIL 80b/ Tilove, R.B. Set membership classification: a unified approach to geometric intersection problems. IEEE Transactions on Computers, C-29 (10):874-883, Out/80.
- /TIL 83/ Tiller, W. Rational B-splines for curve and surface representation. IEEE Computer Graphics and Applications, 3(6):61-69, Set/83.
- /TOP 87/ Technical and Office Protocols, versão 3.0, Society of Manufacturing Engineers, Abr/87.
- /TRO 88/ Trotsmann, E. et al CAD data exchange via neutral interface. IN: Proceedings of Enterprise Network Event, Baltimore, EUA, 2.95-2.107, 1988.
- /TRU 88/ Truslove, K.C. The implications of tolerancing for computer aided mechanical design. Computer Aided Engineering Journal, 5(2):79-85, Abr/88.
- /WEI 84/ Weissflog, U. Experience in design and implementation of an IGES translator. Computer & Graphics, 8(3):269-273, 1984.
- /WEI 85/ Weiler, K. Edge-based data structures for solid modeling in curved surface environments. IEEE Computer Graphics and Applications, 5(1):21-40, Jan/85.
- /WIL 85a/ Wilson, P.R. et al Interfaces for data transfer between solid modeling systems. IEEE Computer Graphics and Applications, 5(6):41-51, Jun/85.
- /WIL 85b/ Wilson, P.R. Euler formulas and geometric modeling. IEEE Computer Graphics and Applications, 5(8):24-36, Ago/85.
- /WIL 87a/ Wilson, P.R. A short history of CAD data transfer standards. IEEE Computer Graphics and Applications, 7(6):64-67, Jun/87.

- /WIL 87b/ Wilson, P.R. Information and/or data? IEEE Computer Graphics and Applications, 7(11):58-61, Nov/87.
- /WIL 87c/ Wilson, P.R. Information modeling. IEEE Computer Graphics and Applications, 7(12):65-67, Dez/87.
- /WOO 88/ Woodward, J. Shape models in computer integrated manufacture: a review. Computer Aided Engineering Journal, 5(3):103-112, Jun/88.
- /YAM 84/ Yamaguchi, K. et al Octree-related data structures and algorithms. IEEE Computer Graphics and Applications, 4(1):53-59, Jan/84.