

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
(Bacharelado)

**CUSTOMIZAÇÃO PARA ACRESCENTAR ATRIBUTOS EM
SÓLIDOS DO SOLIDWORKS.**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA
COMPUTAÇÃO — BACHARELADO

DELICIO VILL WEINFURTER

BLUMENAU, JUNHO/2003

2003/1-15

CUSTOMIZAÇÃO PARA ACRESCENTAR ATRIBUTOS EM SÓLIDOS DO SOLIDWORKS.

DELICIO VILL WEINFURTER

ESTE TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Paulo César Rodacki Gomes — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

Prof. Paulo César Rodacki Gomes

Prof. Jomi Fred Hübner

Prof. Mauro Marcelo Mattos

DEDICATÓRIA

Dedico este trabalho a minha companheira Loreni, que me apoiou durante esta escalada, a qual com certeza subimos juntos.

AGRADECIMENTOS

A minha companheira Loreni, pela compreensão, e pelo amor que ela tem me dado durante esta caminhada, o que me incentivou muito nos momentos mais difíceis.

Ao professor Paulo César Rodacki Gomes, pela orientação, compreensão e amizade durante a execução desta monografia.

Aos professores do Departamento de Computação, donde dentre muitos que conheci durante as aulas, ainda hoje mantenho a amizade.

Ao professor Marcel Hugo, pela amizade e pela ajuda durante a definição da especificação do protótipo.

A vida, pela qual tive a oportunidade de vivenciar a construção deste sonho, o que para muitos ainda é um sonho, mas para mim é pura realidade.

Aos meus amigos, que fiz durante este percurso, pois muitos jamais serão esquecidos pela convivência nesta Universidade.

Pelos bons e maus momentos que passei durante este percurso, os quais conseguiram me mostrar o quanto posso ser forte, e ir sempre em frente, enfrentando os problemas que aparecem.

RESUMO

Este trabalho de conclusão de curso trata da adaptação de ferramentas de CAD (*Computer Aided Design*) para modelagem 3D, atividade usualmente chamada de customização. São apresentadas técnicas de modelagem 3D e uma metodologia e ferramentas de desenvolvimento de aplicações (customizações) para o software SolidWorks. É proposta a implementação de um sistema para cadastramento semi-automático de atributos não geométricos em sólidos gerados dentro do SolidWorks, para posterior geração automática de legendas de formatos de desenho técnico. A viabilidade da proposta é apresentada através da implementação de um protótipo de software implementado em C++ utilizando a API fornecida com o SolidWorks.

ABSTRACT

This work deals with 3D CAD (Computer Aided Design) software customization. Techniques for 3D modeling are presented along with a methodology and tools for development of customized applications for SolidWorks 3D CAD software. The development of a software prototype for semi-automatic attribute insertions into SolidWorks' solids is proposed. The attributes are then used for automatic generation of technical drawings annotation text. The proposal's viability is presented through the implementation of a prototype in C++ along with SolidWorks' development API.

LISTA DE FIGURAS

FIGURA 2-1: EXEMPLO DE CAD 2D	18
FIGURA 2-2: EXEMPLO DE MODELO 3D.....	20
FIGURA 2-3: EXEMPLO DE CAM.....	21
FIGURA 2-4: TELA DE UM SOFTWARE CAE	23
FIGURA 2-5: SISTEMA PDM NUMA ORGANIZAÇÃO EMPRESARIAL.....	27
FIGURA 3-1: FACES LIMITANTES	30
FIGURA 3-2: REPRESENTAÇÃO POR WIREFRAME	31
FIGURA 3-3: REPRESENTADO POR ENUMERAÇÃO EXUSTIVA.....	32
FIGURA 3-4: DECOMPOSIÇÃO DE CÉLULAS.....	32
FIGURA 3-5: REPRESENTAÇÃO POR OCTREE.....	33
FIGURA 3-6: EXEMPLO DE SWEEP TRANSLACIONAL.....	34
FIGURA 3-7: EXEMPLO DE SWEEP CÔNICO	34
FIGURA 3-8: EXEMPLO DE SWEEP ROTACIONAL	35
FIGURA 3-9: EXEMPLO DE SWEEP HELICOIDAL	35
FIGURA 3-10: ÁRVORE CSG.....	36
FIGURA 3-11: EXEMPLO DE PRIMITIVAS.....	37
FIGURA 4-1: INTERFACE DO SOLIDWORKS 2003.....	39
FIGURA 4-2: EXEMPLO DE CONSTRUÇÃO DE MÓVEIS.....	40
FIGURA 4-3: TELA DO COSMOSWORKS INTEGRADA COM O SOLIDWORKS	42
FIGURA 4-4: INTERFACE DO GIBBSCAM	43
FIGURA 4-5: TOOLBAR MACRO	44
FIGURA 4-6: SOLIDWORKS API OBJETOS	46
FIGURA 5-1: DIAGRAMA DE CASOS DE USO	50
FIGURA 5-2: DIAGRAMA DE CLASSES	51

FIGURA 5-3: DIAGRAMA SE SEQÜÊNCIA	53
FIGURA 5-4: EXEMPLO DE CONFIGURAÇÃO	58
FIGURA 5-5: EXEMPLO DE LINK NA LEGENDA DO FORMATO	60
FIGURA 5-6: TELA DO PROTÓTIPO	61
FIGURA 5-7: FORMATO DESENHO TÉCNICO COM ATRIBUTOS.....	62

LISTA DE QUADROS

QUADRO 4-1: ARQUIVOS RECONHECIDOS PELO SOLIDWORKS 2003.....	41
QUADRO 4-2: EXEMPLO DE CÓDIGO FONTE CRIADO AUTOMATICAMENTE	45
QUADRO 5-1: MÉTODO QUANTIDADE DE ACABAMENTO.....	52
QUADRO 5-2: MÉTODO QUANTIDADE DE MATERIAL	52
QUADRO 5-3: FONTE PARA SE CONECTAR AO SOLIDWORKS.....	54
QUADRO 5-4: CLASSE PECA	55
QUADRO 5-5: CLASSE MATERIAL	56
QUADRO 5-6: CLASSE ACABAMENTO	56
QUADRO 5-7: TESTE PARA VERIFICAR SE É UM DRAWING.....	57
QUADRO 5-8: CAPTURANDO CONFIGURAÇÃO ATUAL.....	57
QUADRO 5-9: APAGANDO ATRIBUTOS.....	58
QUADRO 5-10: ACRESCENTANDO ATRIBUTOS	59

SUMÁRIO

1	INTRODUÇÃO	12
1.1	MOTIVAÇÃO.....	14
1.2	OBJETIVOS.....	14
1.3	ESTRUTURA DO TRABALHO	15
2	SISTEMAS CAD.....	16
2.1	COMPUTER AIDED DESIGN (CAD).....	16
2.1.1	CAD 2D	18
2.1.2	CAD 3D	19
2.2	COMPUTER AIDED MANUFACTURING (CAM).....	20
2.3	COMPUTER AIDED ENGINEERING (CAE).....	22
2.4	COMPUTER AIDED PROCESS PLANING (CAPP)	23
2.5	COMPUTER INTEGRATED MANUFACTURING (CIM).....	25
2.6	PRODUCT DATA MANAGEMENT (PDM).....	26
3	MODELAGEM GEOMÉTRICA	30
3.1	REPRESENTAÇÃO DE LIMITES	30
3.2	REPRESENTAÇÃO POR WIREFRAME	30
3.3	REPRESENTAÇÃO POR DECOMPOSIÇÃO VOLUMÉTRICA.....	31
3.3.1	ENUMERAÇÃO EXAUSTIVA.....	31
3.3.2	DECOMPOSIÇÃO DE CÉLULAS.....	32
3.3.3	REPRESENTAÇÃO POR OCTREE.....	32
3.4	SWEEP.....	33
3.4.1	SWEEP TRANSLACIONAL	33
3.4.2	SWEEP TRANSLACIONAL CÔNICO.....	34
3.4.3	SWEEP ROTACIONAL.....	34

	11
3.4.4 SWEEP HELICOIDAL	35
3.5 GEOMETRIA SÓLIDA CONSTRUTIVA - CSG	36
3.6 INSTANCIAMENTO DE PRIMITIVAS	36
4 SOFTWARE 3D SOLIDWORKS.....	38
4.1 A EMPRESA SOLIDWORKS CORPARATION.....	38
4.2 O SOFTWARE SOLIDWORKS 2003	38
4.3 COSMOS/WORKS (CAE)	42
4.4 SMARTEAM (PDM).....	42
4.5 GIBBSCAM (CAM)	43
4.6 CUSTOMIZAÇÃO DO SOLIDWORKS	44
5 DESENVOLVIMENTO DO TRABALHO	48
5.1 REQUISITOS DO PROTÓTIPO.....	48
5.2 ESPECIFICAÇÃO DO PROTÓTIPO	50
5.3 IMPLEMENTAÇÃO	53
5.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS.....	53
5.3.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO.....	60
5.4 RESULTADOS E DISCUSSÃO	63
6 CONCLUSÕES	64
6.1 LIMITAÇÕES.....	65
6.2 EXTENSÕES	65
REFERÊNCIAS BIBLIOGRÁFICAS	66

1 INTRODUÇÃO

Com o crescimento dos recursos para Computação Gráfica, principalmente para a área de modelagem de sólidos, as empresas cada vez mais estão utilizando estes recursos para se tornarem competitivas no mercado, não só as grandes empresas, mas também as médias e pequenas. Hoje é possível identificar várias marcas de softwares para modelagem de sólidos, os mais comuns são: Inventor (*Autodesk*), SolidWorks (*Dassault Systemes S.A*), Mechanical Desktop (*Autodesk*), Unigraphics (*Electronic Data Systems*), Catia (*IBM*), Solid Edge (*Electronic Data Systems*) etc.

A modelagem geométrica consiste em um conjunto de métodos que visam descrever a forma e as características geométricas de um objeto. Ela provê uma descrição ou modelo, muito mais analítico, matemático e abstrato que o real. Cria-se um modelo porque ele é mais conveniente e econômico que o objeto ou o processo real. Através desse modelo fica mais fácil e prático analisá-lo e testá-lo, como por exemplo: a aerodinâmica de automóveis e aviões. Esta modelagem é o ingrediente básico de sistemas *Computer Aided Design (CAD)/Computer Aided Manufacturing (CAM)*, computadores gráficos, arte por computador, visão por computador, simulação, robótica e animação. Avanços em quaisquer destes campos dependem de quão bem definidos estão os modelos geométricos criados (CASACURTA, 1999).

Conforme Silva (2002) a modelagem sólida paramétrica permite que se crie modelos de produtos, com dimensões variacionais. As dimensões podem ser ligadas através de expressões. Ligações bidirecionais entre o modelo e o esquema de dimensionamento permitem a regeneração automática de modelos depois de mudanças nas dimensões e atualização automática das dimensões relacionadas.

Com o crescimento destas ferramentas no mercado, também houve uma queda dos preços das mesmas, o que tornou possível cada vez mais a troca de softwares CAD 2D por modeladores 3D, onde se possa ter acesso às tecnologias CAD/CAM/*Computer Aided Engineering (CAE)* e *Product Data Management (PDM)* em um único software.

Mesmo que estas ferramentas possuam inúmeros recursos de modelagem de sólidos, muitas empresas necessitam de outros recursos para dar maior produtividade em seus projetos, pois muitos processos rotineiros ainda geram desgaste braçal dos projetistas. Para estes casos surgiram empresas especializadas em fazer customizações através de *Application*

Procedural Interface (API) das próprias ferramentas 3D, que podem vir incorporadas ao software ou não, neste caso o fornecedor do modelador de sólidos informa em qual linguagem de programação pode-se fazer tais customizações.

O termo “Customização” faz parte do jargão da computação gráfica. Customizar significa direcionar um aplicativo, como o SolidWorks, a desempenhar tarefas bem específicas, para as quais ele não foi originalmente projetado, facilitando o trabalho do usuário final.

Este trabalho aborda os sistemas CAD, customizações destes sistemas, e atua especificamente na área de PDM, pois neste trabalho é utilizado uma API de uma das ferramenta CAD 3D (SolidWorks), para incorporar “atributos” ao modelo gerado por esta ferramenta. Isto vem a atender uma necessidade do setor de Projetos Mecânicos de uma empresa de automação, onde os projetistas precisam incorporar informações ao modelo 3D para posteriormente recuperá-las, sendo que, posteriormente estas informações poderão ser utilizadas por toda Engenharia do grupo desta empresa. Assim, espera-se que qualquer projetista que utilize estes modelos 3D poderá facilmente obter características mais específicas de partes do modelo tais como material, cor, atributos estruturais e parâmetros de engenharia, bem como códigos de peças etc. A empresa necessita customização específica para o preenchimento automático das informações do modelo 3D nas legendas dos formatos de desenhos técnicos. Atualmente estas informações são preenchidas manualmente pelos projetistas, sendo portanto um processo muito pouco produtivo e com pouco grau de padronização, pois os projetistas tendem a não uniformizar as informações da legenda do desenho. Este trabalho também propõe a construção uma base de dados de atributos conforme as convenções e padrões a serem adotados posteriormente pelos projetistas em conformidade com as necessidades da empresa. Este trabalho prevê o uso da ferramenta CAD 3D SolidWorks.

Segundo Cunha (2000) muitos dos sistemas CAD/CAM/CAE, considerados de alto nível ou de aplicação final (*high-end applications*), possuem o seu próprio modelador geométrico embutido dentro do sistema. Assim como o ACIS (SPATIAL CORPORATION, 1962), o Parasolid também é um núcleo de funções de modelagem geométrica, que pode ser usado em muitos sistemas de alto nível ou de aplicação média (*mid-range application*). Atualmente, o Parasolid é usado como componente integrante de vários sistemas de CAD/CAM/CAE desenvolvidos pelos principais fornecedores de programas para engenharia. Os principais desenvolvedores de sistemas que já aderiram ao Parasolid são: *Parametric*

Technology, SolidWorks Corporation (uma subsidiária da *Dassault Systemes SA*), *ANSYS, Mechanical Dynamics, MacNeal-Schwendler, Bentley Systems etc.* Além disso, o Parasolid é usado internamente em grandes companhias, como uma ferramenta de desenvolvimento de aplicações, como por exemplo: *General Motors, Mitsubishi Motors, Boeing etc.*

O SolidWorks possui núcleo Parasolid (*Electronic Data Systems*) (EDS, 1962). Para implementação do protótipo foi utilizada a linguagem de programação C++ em ambiente de programação Microsoft Visual C++, com a respectiva API do SolidWorks.

1.1 MOTIVAÇÃO

A cada dia que passa, as empresas, não importando seu tamanho e poder, estão buscando cada vez mais os recursos da computação gráfica em ferramentas CAD, para aumentarem a sua produtividade e competitividade no mercado. Estas ferramentas nem sempre dão a produtividade necessária a determinadas aplicações, pois determinadas empresas requerem recursos específicos, nestes casos estas ferramentas CAD podem ser customizadas, gerando maior produtividade para aquela necessidade específica. O conhecimento do autor em modelagem de sólidos com algumas ferramentas CAD, e conhecimento adquirido durante a graduação em Ciências da Computação torna interessante o desenvolvimento deste trabalho, pois o resultado deste trabalho pode vir a ser continuado ou utilizado na área de engenharia (área de atuação do autor).

1.2 OBJETIVOS

O objetivo deste trabalho é criar um protótipo de um programa para acrescentar atributos de identificação (nome, código do produto etc) e atributos de forma (cor, tratamento de superfície etc) nos modelos 3D, gerados pelo software de modelagem 3D SolidWorks, para posteriormente serem recuperados automaticamente em uma legenda de um formato de desenho técnico.

Os objetivos específicos do trabalho são:

- a) construir um protótipo de programa para acrescentar atributos nos modelos 3D do SolidWorks;
- b) utilizar uma API do SolidWorks no ambiente Visual C++ para desenvolvimento do programa;
- c) criar uma base de dados de teste contendo características de cada tipo de atributo

- disponível para ser ligado ao modelo 3D;
- d) no SolidWorks, criar um arquivo *Template* (modelo do formato padrão), que reconheça os atributos do modelo 3D que será inserido e que preencha automaticamente a legenda do mesmo.

1.3 ESTRUTURA DO TRABALHO

O segundo capítulo descreve o significado dos sistemas CAD, CAM, CAE, CIM, CAPP e PDM. O terceiro capítulo apresenta conceitos relativos a modelagem geométrica. No quarto capítulo é apresentado o software SolidWorks escolhido para ser customizado. O quinto capítulo apresenta a especificação, construção, implementação e funcionamento do protótipo. No sexto capítulo tem-se as conclusões, limitações e extensões.

2 SISTEMAS CAD

Neste capítulo serão abordados tópicos referentes a sistemas CAD, sendo mais específico, aborda sobre o significado de algumas siglas ligadas ao CAD, que são encontradas no mercado.

2.1 COMPUTER AIDED DESIGN (CAD)

A sigla CAD ou CADD significa *Computer Aided Designe* (Projeto Auxiliado por Computador) ou ainda é encontrado *Computer Aided Drafting* (Desenho auxiliado por Computador). Quando o CAD é aplicado a áreas específicas são encontrados termos como CAAD (*Computer Aided Architectural Design*) ou CAID (*Computer Aided Industrial Designe*). Um sistema CAD pode ser entendido como um conjunto de ferramentas para a criação, manipulação e alteração interativa de projetos e desenhos. No CAD os desenhos são construídos a partir de uma série de entidades gráficas básicas, as quais são denominadas de entidades primitivas como linhas, círculos, arcos, polígonos, textos etc (KLEIN, 1992).

Segundo Turnell (1999) um software CAD pode ser entendido como uma aplicação da informática com o objetivo de facilitar e agilizar a construção de desenhos em diversas áreas, tais como: aeronáutica, mecânica, automobilística, eletrônica e outras.

A rigor, um desenho feito em um sistema de CAD não é necessariamente feito com velocidade maior do que se fosse feito à mão, embora ocorra uma economia considerável de tempo em casos de modificações no mesmo. Entretanto, têm-se várias outras vantagens que acabam por levar ao uso destes softwares como, maior precisão, recursos para modificação, edição e cópia do desenho. Informações como material, acabamento, são denominados atributos que podem ser associadas a elementos do desenho, permitindo uma melhor documentação do projeto.

Conforme Voisenet (1988), reduzir o tempo de desenho numa companhia é de primordial importância. Considera-se a parte de desenho de projeto como o gargalo de uma garrafa. Desenhistas técnicos tradicionais usam aproximadamente dois terços de seu tempo para planejar ou traçar. Somente um terço é gasto para todas as outras funções profissionais combinadas, incluindo o desenho. A implantação do CAD muda isso. Alterações nos projetos e desenhos podem ser efetuadas muito mais rapidamente. Isto resulta em rápido tempo de

reciclagem. Conseqüentemente, o projeto flui melhor através da companhia. O gargalo representado pelos projetos tradicionais tende a ser eliminado.

Um programa de CAD não é apenas uma ferramenta de desenho, mas sim uma ferramenta para gerar um projeto utilizando um computador. Por meio dele é permitido ao projetista visualizar várias perspectivas do desenho, de forma a avaliar o equilíbrio estético do projeto em estudo. Entre os recursos incluídos na maioria dos aplicativos, pode-se citar:

- a) Criação de bibliotecas de objetos parametrizados, onde classes de objetos genéricos são definidas (por ex., portas), e ao se desejar incluir um objeto específico é necessário informar apenas os parâmetros (por ex., tipo da porta);
- b) Flexibilidade, isto é, opções de personalização do ambiente de trabalho através da alteração de menus, criação de macros e comandos, e mesmo da utilização de linguagens de programação para criação de aplicativos dedicados;
- c) Integração com outros aplicativos, como bancos de dados, com a importação/exportação de informações, entre outras.

O ponto chave dos projetos CAD é um banco de dados que pode ser usado pelos outros departamentos da empresa. Cada detalhe relevante de uma peça individual é armazenado no banco de dados CAD, de modo que outros departamentos podem acessar estes detalhes para seu próprio uso. E, o desenho, após a sua geração e armazenamento pode ser transformado em um arquivo de dados geométricos, podendo, com o auxílio de informações tecnológicas, gerar a programação para que uma determinada máquina produza este produto.

Os softwares CAD foram inicialmente aplicados na indústria aeronáutica e, com o seu sucesso nessa área, foram disseminados para outras áreas, tais como: indústrias de calçados, de vestuário, mecânicas, automobilística, eletrônica, naval e outras.

Pode-se dizer que os softwares de CAD se dividem em dois grupos: 2D e 3D. Os denominados 2D usados para trabalho no plano, conhecidos como sistemas de desenho. Já os denominados 3D permitem trabalhos em três dimensões, conhecidos como sistemas de projeto.

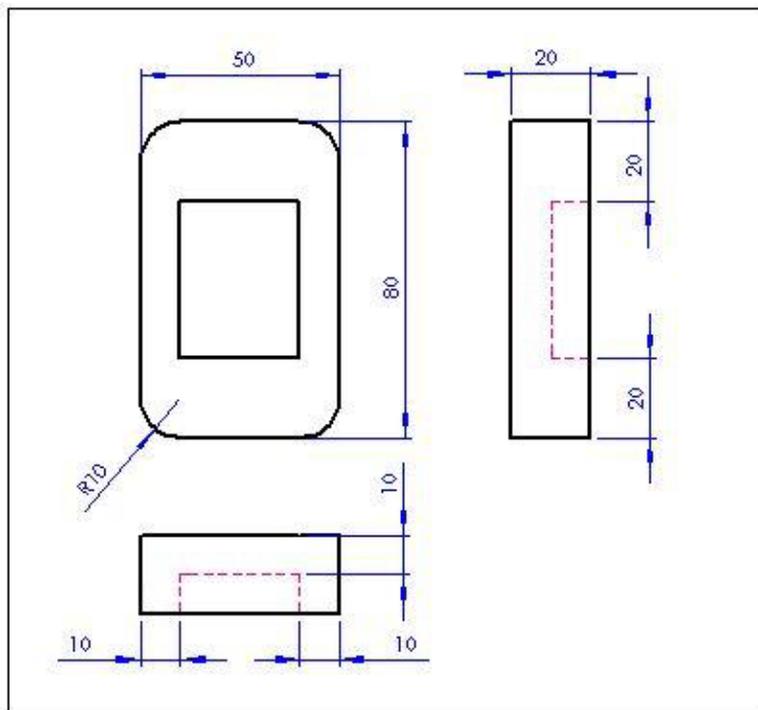
2.1.1 CAD 2D

O CAD 2D é um tipo de aplicativo onde a criação e a manipulação dos desenhos em duas dimensões acontecem sobre um sistema de coordenadas, com a utilização de entidades geométricas tais como: ponto, linha, círculo, elipse, polígono e outros elementos oferecidos pelos aplicativos.

Também é possível manipular os elementos desenhados alterando suas propriedades, variando escala, posicionamento, rotação e executando cópias simples ou múltiplas. Em nível de visualização pode-se controlar a representação do desenho, alterando valores de aproximação ou afastamento, tendo assim o aumento ou redução do desenho na tela. Este recurso é conhecido como *zoom*.

Existe ainda nos aplicativos bidimensionais a possibilidade de definição de cores, tipos de linhas, cotagem dos elementos e a criação de textos e símbolos. Um exemplo de modelo gerado em um aplicativo CAD 2D é apresentado na fig. 2-1.

FIGURA 2-1: EXEMPLO DE CAD 2D



Segundo Turnell (1999) ainda hoje, apesar do desenvolvimento de sistemas 3D, a representação em 2D é melhor para realizar as atividades de documentação e detalhamento final de um projeto. Pois durante o processo de fabricação da peça ou outro objeto, o desenho

2D é a representação física em forma de documento, contendo informações sobre a peça, que acompanha a mesma até seu ciclo final de fabricação.

2.1.2 CAD 3D

Estes sistemas permitem definir e manipular geometrias tridimensionais, dando ao desenho a noção de volume. Esta geração de desenho, é denominada modelagem geométrica, e permite a obtenção de representações visuais diversas, isto é, pode-se visualizar a geometria modelada de qualquer ponto de observação.

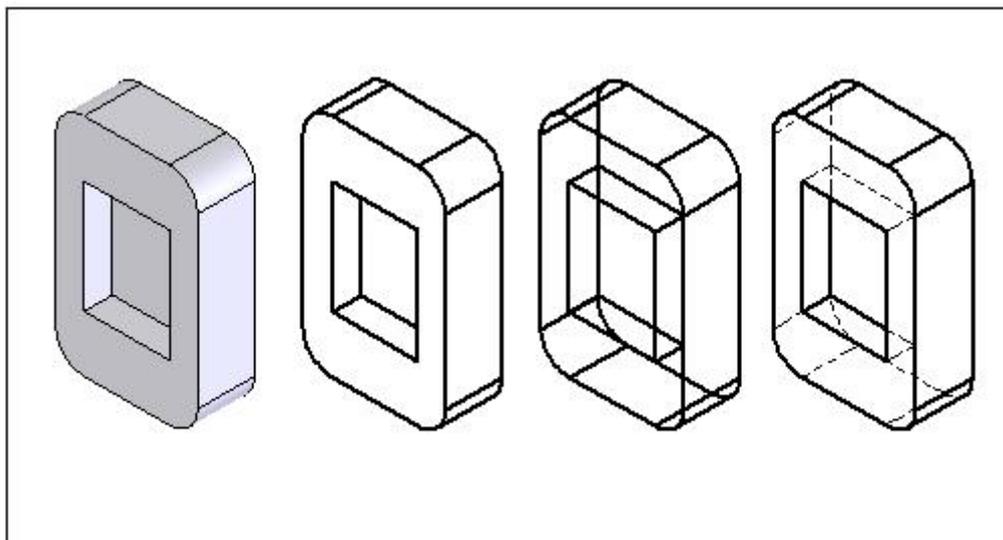
Segundo Proença (1995) existem três tipos de representação para modelagem geométrica:

- a) *Wireframe* (Modelagem de Arame): permite ampla criação de modelos, onde o desenho é visualizado no espaço por uniões de linhas definindo limites e dando a sensação tridimensional.
- b) Modelagem de Superfície: essa técnica se realiza com base na definição das superfícies do objeto, determinando assim o seu volume. A sua construção pode ser feita a partir de superfícies simples, esféricas, cúbicas e aquelas geradas por curvas francesas, além de técnicas de varredura de perfis.
- c) Modelagem de Sólidos: essa é a técnica mais avançada de representação de um objeto tridimensional. Baseia-se na delimitação do volume ocupado pelo objeto no espaço.

A partir de um conjunto de formas sólidas elementares tais como cubo, cilindro, esfera, pirâmide, e outras mais sofisticadas geradas a partir de perfis complexos e operações booleanas, como união, intersecção e subtração de primitivas, é possível chegar a objetos de múltiplas formas.

Estas representações permitem a remoção das linhas ocultas e visualização com sombreamento “*shading*”, conferindo maior realismo à imagem. Ainda é possível com o uso da modelagem de sólidos, extrair propriedades do tipo: centro de gravidade, volume, eixo etc (PROENÇA, 1995). Na fig. 2-2 é apresentado um exemplo de modelo 3D.

FIGURA 2-2: EXEMPLO DE MODELO 3D



2.2 COMPUTER AIDED MANUFACTURING (CAM)

Conforme Stemmer (1996), a utilização do computador na fabricação iniciou-se pela necessidade de assegurar uma simplificação do processo de execução de fitas perfuradas para máquinas de controle numérico e na necessidade de desenvolver processos, que já eliminassem de saída, a maioria dos erros que ocorrem na preparação manual.

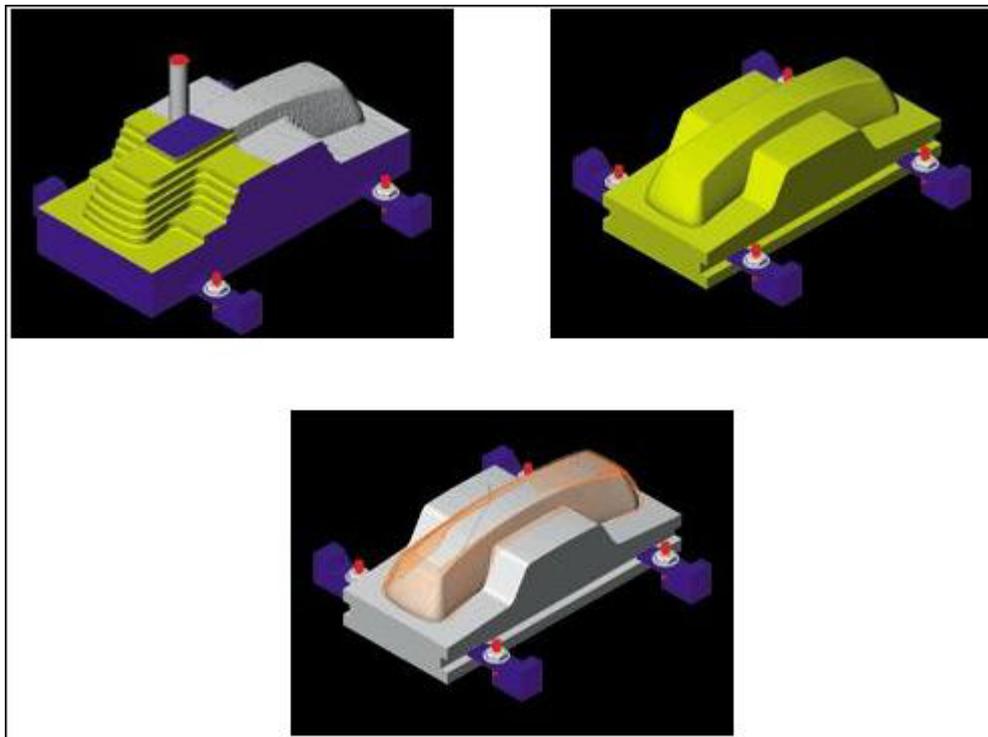
O CAM pode ser definido como o uso de sistemas computacionais para planejar, gerenciar e controlar as operações de uma planta de fabricação através de uma interface direta ou indireta com os recursos de produção da planta. A sua função primária é a preparação e remessa de programas para as máquinas automatizadas, centros de usinagem, equipamentos transporte de materiais e outros equipamentos de apoio, para com isso ocorrer a produção de um determinado item (TURNELL, 1999).

Com a evolução da complexidade das formas usinadas, tornou-se imprescindível a automação, planejamento e otimização deste processo. Em um sistema integrado, o modelo desenvolvido é submetido à análise de determinação da forma mais adequada de usinagem, desde a escolha do equipamento, ferramental e estratégia do processo. Os recursos incluem a definição do caminho da ferramenta (percurso que a ferramenta percorre durante a usinagem), magazine de ferramentas (ferramentas de usinagem disponíveis para uma determinada máquina), suporte a operações de fresamento, torneamento e perfuração.

Segundo Turnell (1999), um modelo tridimensional gerado em CAD ou no próprio software de CAM é usado como referência para geração da programação dos equipamentos envolvidos no processamento desse item. Após o operador estabelecer uma série de parâmetros para o software, este faz a computação necessária para a geração dos programas para as máquinas de usinagem mecânica, que farão a fabricação de peças reais.

A simulação visual do caminho da ferramenta e remoção de material é uma característica importante, pois permite ao engenheiro uma análise visual do processo, eliminando dúvidas e possibilitando a melhor escolha dentre as possíveis. A fig. 2-3 apresenta um exemplo de simulação do CAM, com três etapas da simulação de usinagem.

FIGURA 2-3: EXEMPLO DE CAM



FONTE: GibbsCam (1982)

Segundo Voisinet (1988), a combinação CAD e CAM teve o efeito de aumentar radicalmente a produtividade e a precisão. Quando o CAD é usado para preparar desenho de um produto, as instruções para produção ou preparação daquele desenho são enviadas diretamente para a fábrica.

É interessante notar que, em toda a bibliografia consultada para a realização deste trabalho, o CAM é sempre muito relacionado ao CAD, pois o objetivo sempre é passar do

projeto à manufatura da maneira mais rápida e fácil possível. Desse modo, muitas vezes eles são citados de forma conjunta (sistemas CAD/CAM). A relação também é óbvia uma vez que as atividades CAM quase sempre necessitam de alguma informação CAD como entrada.

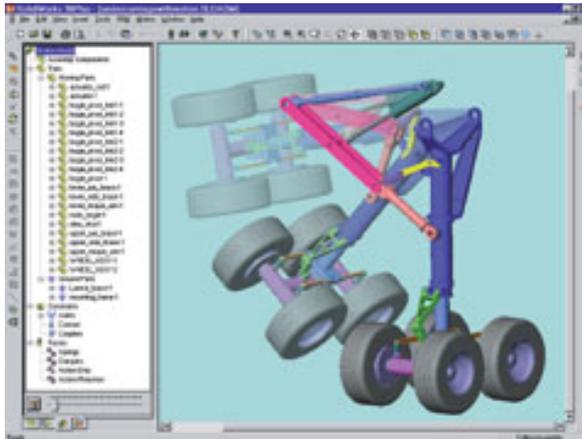
2.3 COMPUTER AIDED ENGINEERING (CAE)

O CAE é uma filosofia que busca a minimização do tempo de lançamentos de novos produtos e a redução dos custos de projeto. Ele utiliza uma abordagem de engenharia preceptiva, que se propõe a resolver problemas ainda na fase de projeto, tentando assim reduzir o número de correções na fase de implementação (que são correções de alto custo) (TURNELL, 1999).

Aqui são utilizados modelos computacionais dos produtos, ao invés de protótipos, para a realização dos testes e análises. Com isso, tem-se uma redução no tempo e custo do projeto, o aprimoramento e possíveis correções deste podem ser realizados mais rapidamente e com melhor qualidade, pois pode-se utilizar o computador para auxiliar na modelagem e análise. Um erro de projeto pode ser facilmente corrigido e rapidamente testado em um sistema computadorizado, enquanto que, se for utilizado um protótipo, muitas vezes será necessário à confecção de um novo protótipo, o que demandará novos custos e um tempo considerável para a confecção.

Tendo sido desenvolvido um projeto em CAD, mesmo que sem o detalhamento final, é possível realizar várias análises no modelo, dentre as quais pode-se destacar: cálculo de esforços, cálculos de transferência de calor, uso de equações diferenciais para descrever o comportamento dinâmico do sistema sendo projetado, simulação de mecanismo e outros. A fig. 2-4 ilustra a tela de um software de simulação em CAE.

FIGURA 2-4: TELA DE UM SOFTWARE CAE



FONTE: CosmosWorks (1982).

O conceito de CAE pode ser empregado nas mais diversas áreas da engenharia, tais como: estruturas, projetos mecânicos, análise de circuitos elétricos, microeletrônica, mecânica de fluidos, eletromagnetismo e outros. Em muitos casos, a utilização do CAE faz-se indispensável, devido à grande dificuldade no uso da abordagem tradicional. De qualquer forma, a sua utilização na indústria brasileira está se disseminando rapidamente, principalmente devido aos evidentes benefícios desta tecnologia (PROENÇA, 1995).

Ao se escolher uma ferramenta CAE deve-se observar a sua capacidade de integração com o sistema CAD a ser utilizado, principalmente em relação ao formato dos arquivos. Hoje em dia, o CAE já se integra facilmente com o CAD, já existem soluções integradas CAE/CAD no mercado. Um exemplo disto é o CosmosWorks (COSMOSWORKS, 1982) que é totalmente integrado ao software CAD SolidWorks.

2.4 COMPUTER AIDED PROCESS PLANING (CAPP)

Segundo Proença (1995), o CAPP (planejamento de processo auxiliado por computador) tem por objetivo o direcionamento do fluxo de uma determinada peça através do departamento de produção, e é uma atividade em que a sua informatização pode resultar em grandes benefícios.

A utilização dos planos de processo varia de empresa a empresa, podendo, em alguns casos, ser pouco mais do que um sugestão sobre quais máquinas devam ser utilizadas na fabricação de determinada peça, e em outros, podem ser folhas de operação altamente

elaboradas, as quais delineiam em grandes detalhes os métodos a serem usados no processo de fabricação.

O principal objetivo do planejamento do processo é selecionar e definir, em detalhes, as etapas de fabricação de um produto. Deste modo, as especificações do produto (resultantes das atividades de projeto) são transformadas em informações de processo de manufatura. O planejamento de processos pode simples sugestões de que máquinas utilizar, ou podem ser altamente elaborados a ponto de conter detalhes sobre os métodos a serem usados no processo de fabricação do item.

As folhas de processo geralmente seguem um padrão. Assim, futuras peças podem ser baseadas no planejamento feito em peças que antecederam a estas. São casos raros de modificação nos planos de fabricação, sendo assim as peças são podem reutilizar planos anteriores, garantindo maior agilidade no planejamento de fabricação das peças (TURNELL, 1999).

A informatização do planejamento do processo, o CAPP, auxilia no direcionamento do fluxo de uma determinada peça no departamento de produção. O CAPP representa uma ponte entre o projeto e a manufatura em sistema CAD/CAM.

Segundo Proença (1995) existem duas técnicas fundamentais utilizadas em sistemas CAPP: a primeira e chamada de Abordagem Variante, que é derivada do tradicional planejamento de processos. Em um sistema variante, a tecnologia de grupo e técnicas de classificação e codificação para catalogar as peças em família. As famílias são então agrupadas pelas suas características similares de fabricação, permitindo que um plano de processo padrão possa ser desenvolvido para cada família de peças. Assim, quando uma nova peça for introduzida, o plano padrão para aquela família particular é alterado, acomodando quaisquer variações requeridas na fabricação da nova peça. O plano de processo resultante se torna, dessa forma, uma variação do plano padrão, donde vem o nome variante. a segunda técnica utilizada é chamada Abordagem Generativa. Tal abordagem relaciona-se aos sistemas especialistas utilizados em inteligência artificial. No sistema generativo, a lógica e a experiência dos planejadores de processo são capturadas no computador e utilizadas em algoritmos, modelos matemáticos, árvore de decisão etc., para produzir planos ótimos. O computador, de fato, gera um plano de processo baseado nas características da peça.

É importante lembrar que a interface entre os sistemas CAPP e os de CAD e CAM, precisam de fato estar plenamente alcançadas, para que as informações contidas nos desenhos gerados pelo CAD sejam compreendidas no CAPP, enquanto que, por outro lado, os planos de processo gerados devem ser prontamente executados em sistemas CAM.

2.5 COMPUTER INTEGRATED MANUFACTURING (CIM)

Conforme Turnell (1999), um sistema flexível de manufatura pode ser integrado eletronicamente aos departamentos de engenharia (projeto auxiliado por computador), produção, e controle de estoque. Neste caso, um esboço auxiliado por computador pode, no fim das contas, gerar o código eletrônico (instruções) necessário para controlar uma máquina de controle numérico. Se esta máquina está conectada a outras e a equipamentos de manipulação de material como parte de um sistema flexível de manufatura, então o sistema completo é chamado de CIM (manufatura integrada por computador).

A CIM é um modelo do sistema integrado de produção. Ele representa uma forma de organização de um sistema de manufatura que passa pela integração organizacional do sistema realizada por meio informatizado.

A CIM está centrada na integração de vários métodos e técnicas que são, usualmente, aplicados independentemente em vários estágios de todo o processo de manufatura. O conceito CIM representa a tentativa de integrar todas as aplicações computacionais dentro de uma empresa em uma estrutura harmônica e coerente. As aplicações computacionais existentes para automatização da manufatura (CAD, CAE, CAM, CAPP) são, freqüentemente, ilhas de automação, incapazes de se comunicar umas com as outras pois não possuem um meio de troca de dados (não possuem um banco de dados comum e/ou não estão interligados em uma rede de comunicação).

Em princípio, um modelo CIM é uma representação esquemática de uma proposta de integração das funções de uma empresa. Essa representação deve conter as partes a serem integradas e apresentar um esboço da arquitetura final do sistema. O grau de detalhamento da representação pode variar, identificando desde macro funções de uma empresa (finanças, produção, etc.) até as atividades concretas (emissão de ordens, coleta de dados operacionais, etc.) (PROENÇA, 1999).

Para uma indústria aplicar o modelo CIM em seu sistema ela deve integrar fisicamente e logicamente todos os seus setores. A integração física é realizada por meio da automação de seus sistemas e da interligação de todos os setores da indústria por meio de uma rede de comunicação de dados. Já a integração lógica é realizada pelo agrupamento dos dados do processo por meio de banco de dados compartilhados entre os diversos setores da indústria. Com isso os dados gerados no departamento de engenharia (projeto) podem ser facilmente enviados para o setor de produção para a fabricação de um determinado item (o que pode ocorrer de forma totalmente automatizada).

Segundo Turnell (1999), devido aos problemas tecnológicos e de capital relacionados com as redes de comunicações, sistemas CIM totalmente operacionais ainda são raros.

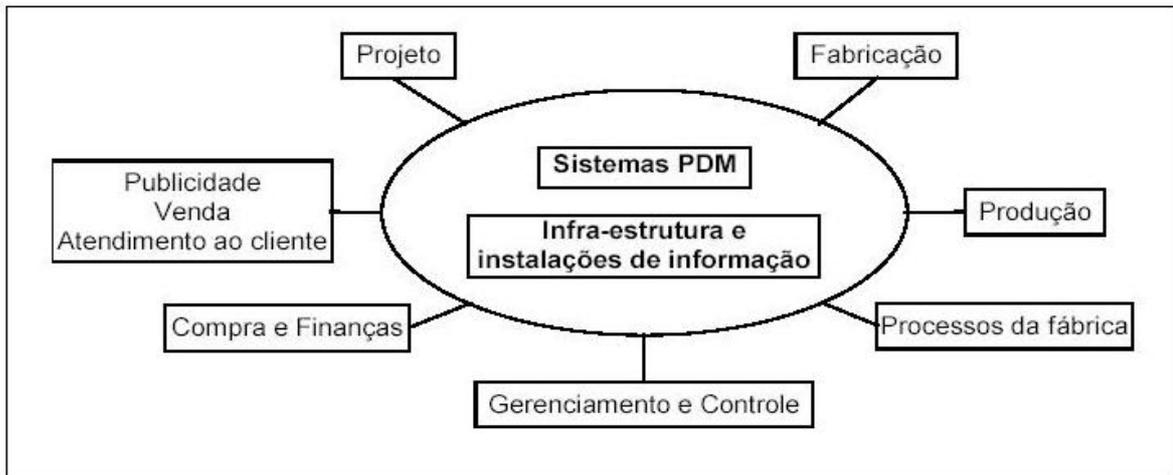
2.6 PRODUCT DATA MANAGEMENT (PDM)

PDM é um acrônimo das palavras, em inglês, Product Data Management, que quer dizer "Gerenciamento de Dados do Produto". PDM compreende conjuntos de técnicas e ferramentas para ajudar o gerenciamento dos dados e dos processos de desenvolvimento de produtos. Os sistemas PDM guardam a trilha de dados e informação requeridos para projetar, fabricar ou construir, e também auxiliar o suporte e a posterior manutenção do produto (CUNHA, 2000).

Os sistemas PDM são uma tecnologia relativamente nova, embora o conceito tenha recebido vários disfarces, ou se modificado com várias roupagens por um longo tempo. Os primeiros sistemas computadorizados disponíveis foram introduzidos no início da década de 80. O surgimento de tais sistemas é justificado pela necessidade e a oportunidade vislumbrada pelas empresas implementadoras de sistemas CAD em desenvolver sistemas que gerenciassem o grande volume de informações gerado. Por outro lado, empresas que comercializavam sistemas de gerenciamento de documentos e imagens, também identificaram esta necessidade ao verificar que seus aplicativos não atendiam todas as necessidades do processo de desenvolvimento de produtos.

Numa organização empresarial, os sistemas PDM atuam nos diversos departamentos e setores, conforme mostra a fig. 2-5.

FIGURA 2-5: SISTEMA PDM NUMA ORGANIZAÇÃO EMPRESARIAL



FONTE: Cunha (2000)

No início da década de noventa surgiram os primeiros sistemas PDM. Estes incorporavam ao recurso de gerenciamento de documentos eletrônicos dos *Electronic Data Management System* (EDMS), a possibilidade de gerenciar dados de produto. Isto significava atuar em rede com os sistemas CAD, compartilhando-os através de bancos de dados. Essa pode ser considerada a 1ª geração dos sistemas PDM (CUNHA, 200).

Os sistemas CAD/CAM/CAE e CIM, com a facilidade de rapidamente gerar e alterar dados do produto, têm exigido que, com muito esforço, os sistemas convencionais gerenciem dados, e também auxiliem no processo de criação e utilização dos dados numa forma ordenada e eficiente. Dessa forma, os usuários podem facilmente criar quantidades consideráveis de informação relacionada ao produto. E para complicar ainda mais a questão, esta informação está freqüentemente armazenada em diferentes formatos dentro de uma variedade de sistemas, e se encontra em diferentes e dispersos computadores. Esses podem estar armazenados em diferentes tipos de mídia.

Em suma, qualquer informação necessária em toda parte do ciclo de vida do produto pode ser gerenciado por um sistema PDM, deixando os dados corretos acessíveis a todas as pessoas e sistemas que tenham necessidade de usá-los. Um sistema PDM não está limitado a gerenciar somente os ciclos de projeto, mas de acordo com as necessidades do usuário, pode gerenciar a concepção do produto, o projeto detalhado, prototipagem e testes, fabricação, operação, e manutenção.

Os sistemas PDM proporcionam um significativo ganho de produtividade quando eles são usados por um grupo ou equipe de trabalho. Todavia, um impacto muito maior é sentido, ou fica mais evidente, quando eles são aplicados a uma organização empresarial e industrial de grande porte.

A utilização de sistemas PDM auxilia a implementação da Engenharia Simultânea (uma alteração do produto em uma das áreas da empresa reflete nas outras), pois estes sistemas fornecem meios para integração e para o gerenciamento das informações do produto e do processo de desenvolvimento de produtos conforme as necessidades da empresa (CIMDATA, 1999).

Atualmente, o desenvolvimento de produto deve estar integrado com todas as atividades, áreas e informações da empresa. Para essa necessidade, surgiram os sistemas PDM ditos de 2ª geração. Tais sistemas atendem grande parte das necessidades de engenharia, gerenciando os modelos CAD e aplicando o conceito *Digital Mock-Up* “maquete eletrônica” (DMU) . Os recursos de coordenação dos fluxos de atividades de decisão, aprovação e ordenamento de tarefas, e modificações realizadas sobre os modelos. Além disso, permitem uma integração com os sistemas computacionais de *Materials Requirements Planning* (MRP) através de relacionamentos com listas de materiais, atendendo as áreas de planejamento, fabricação, suprimentos e financeiras (CUNHA, 2000).

Devido à sua aplicação num amplo conjunto de usuários e áreas, e à sua facilidade para controlar o fluxo de informações, os sistemas PDM servem como instrumentos para implementar as práticas da Engenharia Simultânea. Os benefícios advindos da implementação da Engenharia Simultânea vão além da engenharia de projeto, propagando-se para economia de custos de fabricação, redução do tempo de lançamento de novos produtos no mercado e aumento da qualidade do produto, além dos benefícios de redução do tempo de projeto de engenharia.

Os sistemas PDM integram e gerenciam processos, aplicações, e informação que definem o produto por meio de diversos sistemas e mídias. A qualidade de um produto é função de um desenvolvimento, uma fabricação e uma distribuição eficiente e bem administrada. Os sistemas PDM ajudam atingir esses objetivos.

As mudanças nos produtos são inevitáveis, mas elas podem tornar-se excessivamente caras e descontínuas se ocorrem muito tarde dentro do ciclo de vida do produto, tais como na

produção ou fabricação, ou mais tarde ainda. As mudanças poderiam ocorrer bem mais cedo no ciclo de vida do produto se as equipes de projeto fossem encorajadas a compartilhar dados e documentos num ambiente suportado por um sistema PDM. Este é o caminho para melhores produtos, custos reduzidos, economia de tempo, menos sobras e menos re-projetos.

A classificação permite que peças similares ou padronizadas, processos, e outras informações de projeto possam ser agrupadas por atributos comuns e recuperadas para uso em produtos. Isto conduz a uma maior padronização, re-projeto reduzido, economia em compras e fabricação, e estoques reduzidos (CUNHA, 2000).

Segundo Barbosa (1996), em um sistema com recursos PDM as características não geométricas de partes do modelo são quantificadas ou qualificadas, bem como incorporadas ao modelo por objetos chamados “atributos” (*attribute*), aos quais podem ser associados “valor” (*value*) correspondente à sua representação.

As funções de classificação dos sistemas PDM fornecem recursos muito mais eficientes de procura de peças similares ou padronizadas, do que os recursos encontrados em catálogos e outros sistemas manuais. Quando engenheiros e projetistas são capazes de encontrar peças similares ou padronizadas facilmente, eles provavelmente as reusam ao invés de projetar partindo do nada ou de rascunhos. As peças podem ser agrupadas e encontradas por vários atributos, tais como: códigos funcionais da peça; classificação da forma, material, data de possíveis versões da peça, revisão, projetista, nome do projeto, proprietário, data de criação etc. Estes e outros atributos podem estar associados a dados do produto que têm relação com as peças. Peças e projetos que correspondem a atributos específicos, faixas de valores, e combinações lógicas de atributos, podem ser recuperados. Isto permite que os usuários facilmente obtenham dados de produtos, que sejam idênticos ou similares ao do projeto requerido.

3 MODELAGEM GEOMÉTRICA

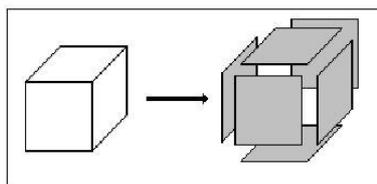
Neste capítulo serão abordados tópicos sobre modelagem geométrica, técnicas de representação do objeto real em meio digital, dentro de ferramentas CAD. Os tópicos serão abordados para dar uma noção básica sobre modelagem geométrica.

3.1 REPRESENTAÇÃO DE LIMITES

Os modelos *Boundary Representation* (B-Rep) representam um objeto sólido dividindo-o em faces convenientemente relacionadas, compostas por superfícies fechadas e orientadas, e que possuam uma representação matemática computacional compacta (normalmente são superfícies planas, quadráticas, toroidais, ou paramétricas). Em seguida à fragmentação do objeto em faces, estas são quebradas em curvas que representam as arestas de cada face constituinte do objeto. As arestas, por sua vez são quebradas em vértices (CUNHA, 2000).

O modelo por representação de limites define um sólido indiretamente através da representação das suas superfícies limitantes. São baseados em uma visão orientada para superfícies, em outras palavras, eles representam um objeto sólido através da subdivisão deste objeto em faces. A fig. 3-1 representa um exemplo de faces limitantes.

FIGURA 3-1: FACES LIMITANTES



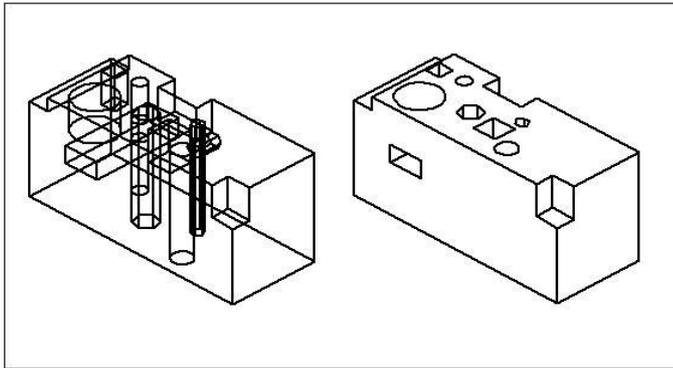
FONTE: Casacurta (1999)

3.2 REPRESENTAÇÃO POR WIREFRAME

Na representação estrutura de arame ou *wireframe*, os modelos geométricos são definidos por uma série de linhas e curvas que representam as arestas ou contornos físicos do objeto.

A representação por *wireframe* é a mais comum encontrada nos softwares CAD. Nesta representação as arestas dos objetos são mostradas como linhas. Em objetos com superfícies curvas, são adicionadas linhas de contorno (GIESECKE, 2002). A fig. 3-2 mostra a representação por wireframe.

FIGURA 3-2: REPRESENTAÇÃO POR WIREFRAME



3.3 REPRESENTAÇÃO POR DECOMPOSIÇÃO VOLUMÉTRICA

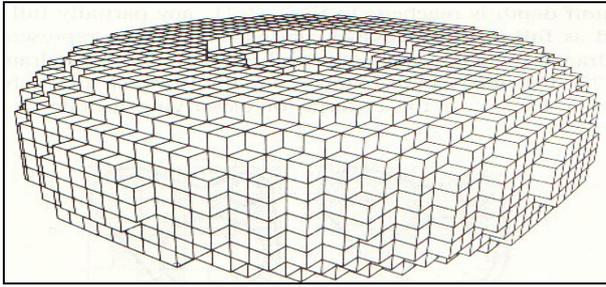
Este é um modelo por decomposição (*Spatial-Partitioning*), ou seja, os sólidos são descritos através da união de blocos básicos. Primitivas podem variar no tipo, tamanho, posição, parametrização e orientação, e a maneira como os blocos são combinados é que distingue cada variação deste esquema de representação (SILVA, 1999).

3.3.1 ENUMERAÇÃO EXAUSTIVA

A *spatial-occupancy enumeration* (enumeração de ocupação espacial ou enumeração exaustiva) é um caso especial da decomposição celular na qual o sólido é decomposto em células idênticas organizadas em uma grade regular e fixa.

Pode-se interpretar um sólido como um conjunto contíguo de pontos tridimensionais, porém, não é possível enumerar todos os pontos de um objeto. Partindo desta afirmação, pode-se facilmente representar um sólido enumerando-se pequenos sólidos básicos contidos (totalmente ou parcialmente) no sólido. Estes pequenos sólidos são muitas vezes denominados de *voxel* (volume *elements*) ou cubos (SILVA, 1999). A fig. 3-3 mostra um objeto representado por esta técnica.

FIGURA 3-3: REPRESENTADO POR ENUMERAÇÃO EXUSTIVA



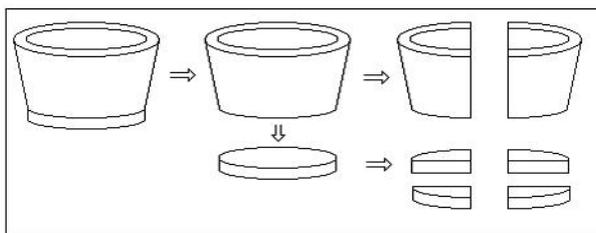
FONTE: Silva (1999)

3.3.2 DECOMPOSIÇÃO DE CÉLULAS

Outra técnica utilizada para resolver problemas da enumeração exaustiva, mas preservando as suas propriedades, utiliza outros tipos de elementos básicos além de cubos. Estes esquemas são denominados esquemas por decomposição de células. Cada sistema de decomposição de célula define um conjunto de células primitivas que são de característica paramétrica e muitas vezes curvas (SILVA, 1999).

Um sólido pode ser representado pela sua decomposição em células como é mostrado na fig. 3-4. Esquemas de ocupação espacial são um caso particular em que todas as células no esquema devem ser cúbicas e estar em uma rede espacial fixa.

FIGURA 3-4: DECOMPOSIÇÃO DE CÉLULAS



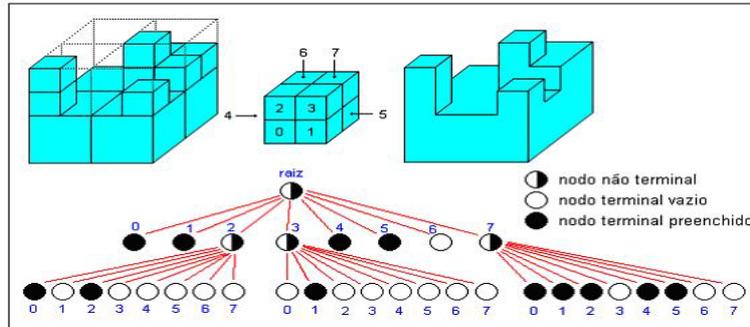
FONTE: Casacurta (1999)

3.3.3 REPRESENTAÇÃO POR OCTREE

Segundo Silva (1999), os pontos positivos da enumeração exaustiva contrastam com as grandes desvantagens desta técnica que são o enorme consumo de memória e a falta de exatidão. *Octree* é uma variante da enumeração exaustiva, concebida para otimizar o armazenamento. Ela é derivada da *quadtree*, um formato para codificar imagens.

Com base neste fato, o sistema de representação por *octree* utiliza uma subdivisão espacial recursiva de um espaço de interesse em seus oito octantes, de modo a formar uma árvore com oito nodos filhos ou ramos. A fig. 3-5 a seguir mostra a representação por *octree*.

FIGURA 3-5: REPRESENTAÇÃO POR OCTREE



FONTE: Cascurta (1999)

Conforme Silva (1999), no contexto da modelagem de sólidos, os objetos *octrees* são comumente construídos a partir de primitivas sólidas. Com relação a estas primitivas, cada nodo de um octante pode possuir os mesmos tipos de estado da *quadtree*: cheio, parcialmente cheio ou vazio.

Nota-se que, como na enumeração exaustiva, a técnica de representação *octree* é aproximativa, dando ao modelo apenas semelhança do objeto real. Mas também, esta representação pode ser considerada válida.

3.4 SWEEP

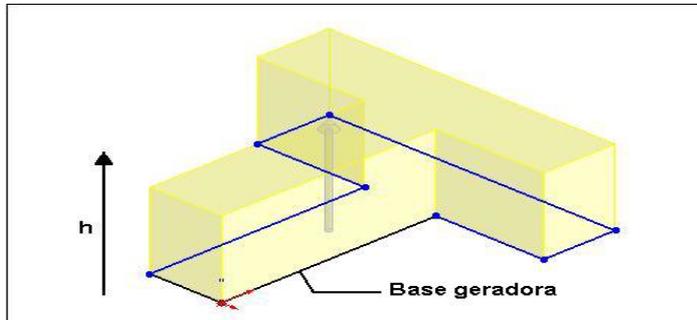
Arrastar um objeto ao longo de uma trajetória no espaço define um novo objeto, que pode ser chamado de um *sweep*. O tipo mais simples de *sweep* é definido por uma área 2D arrastada ao longo de um caminho linear normal ao plano da área para criar um volume.

3.4.1 SWEEP TRANSLACIONAL

Segundo Silva (1999), é o método de *sweep* translacional tradicional básico. Tendo-se um polígono de “n” vértices e uma altura de translação “h”, obtêm-se o polígono superior desse sólido simplesmente transladando o polígono gerador na direção perpendicular a base do polígono.

O *sweep* translacional em ferramentas CAD é conhecido como extrusão básica, sendo apenas necessário informar a base e altura do polígono em que se queira gerar. Um exemplo deste tipo de *sweep* é mostrado na fig. 3-6.

FIGURA 3-6: EXEMPLO DE SWEEP TRANSLACIONAL

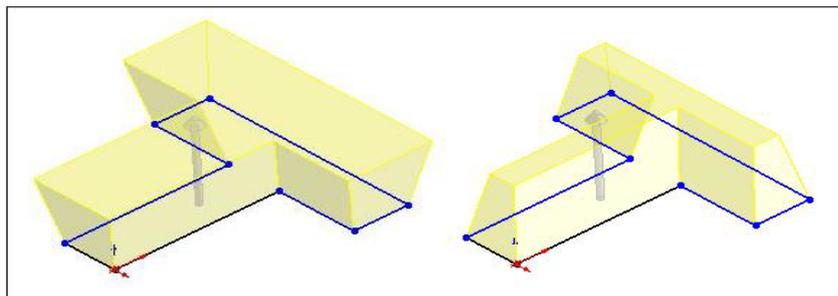


3.4.2 SWEEP TRANSLACIONAL CÔNICO

Uma simples extensão do *sweep* translacional, consiste de se alterar a dimensão da base formadora de forma que este deslocamento forme objetos de forma cônica assim como fazer um deslocamento de maneira não perpendicular ao plano em que está a base formadora (FOLEY, 1990).

Neste tipo de *sweep* que é semelhante ao translacional simples, é informado um novo parâmetro, onde em muitos softwares pode ser o um ângulo também conhecido como ponto de fuga, que pode aumentar ou diminuir a base geradora, conforme fig. 3-7.

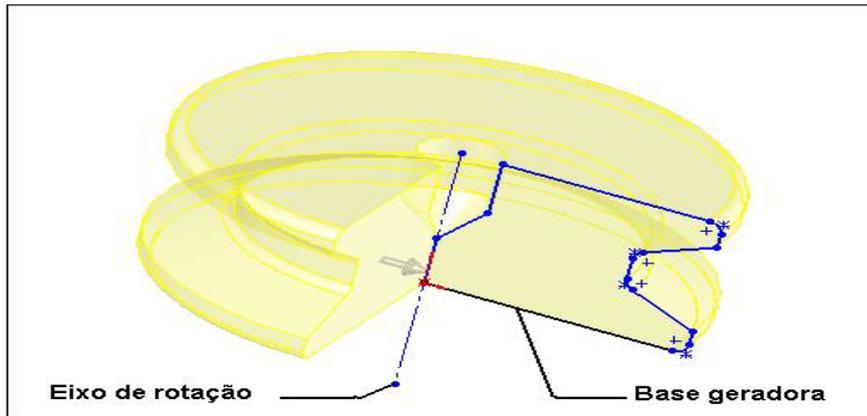
FIGURA 3-7: EXEMPLO DE SWEEP CÔNICO



3.4.3 SWEEP ROTACIONAL

O *sweep* rotacional pode ser definido como a rotação de um polígono gerador (base geradora) em torno de um eixo (eixo de rotação). Nesta técnica é possível construir sólidos com rotação igual ou menor que 360° , em torno do eixo de rotação. Na fig. 3-8, podemos ver um sólido gerado pela técnica *sweep* rotacional, onde a rotação definida foi 270° .

FIGURA 3-8: EXEMPLO DE SWEEP ROTACIONAL

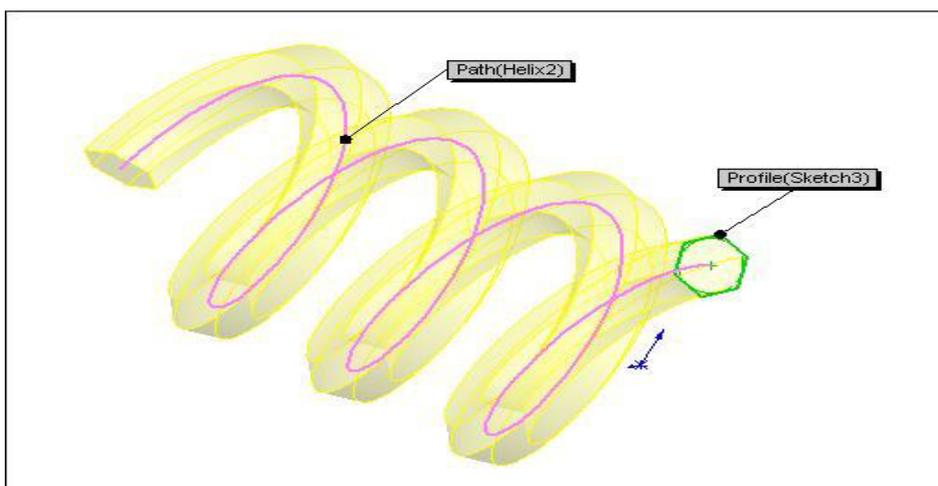


3.4.4 SWEEP HELICOIDAL

O *sweep* helicoidal é uma combinação dos tipos de *sweep*, o rotacional de um polígono fechado e translacional. Além de incremento de revolução também é feito um incremento de translação no mesmo sentido do eixo de revolução. Neste caso, o ângulo de revolução não está limitado a 360 graus, podendo executar várias voltas (SILVA, 1999).

Este tipo de *sweep* é muito utilizado para construção de molas, fiações elétricas etc. A fig. 3-9 mostra um exemplo de *sweep* helicoidal, sendo o *path* o caminho a ser seguido, e o *profile* o polígono gerador.

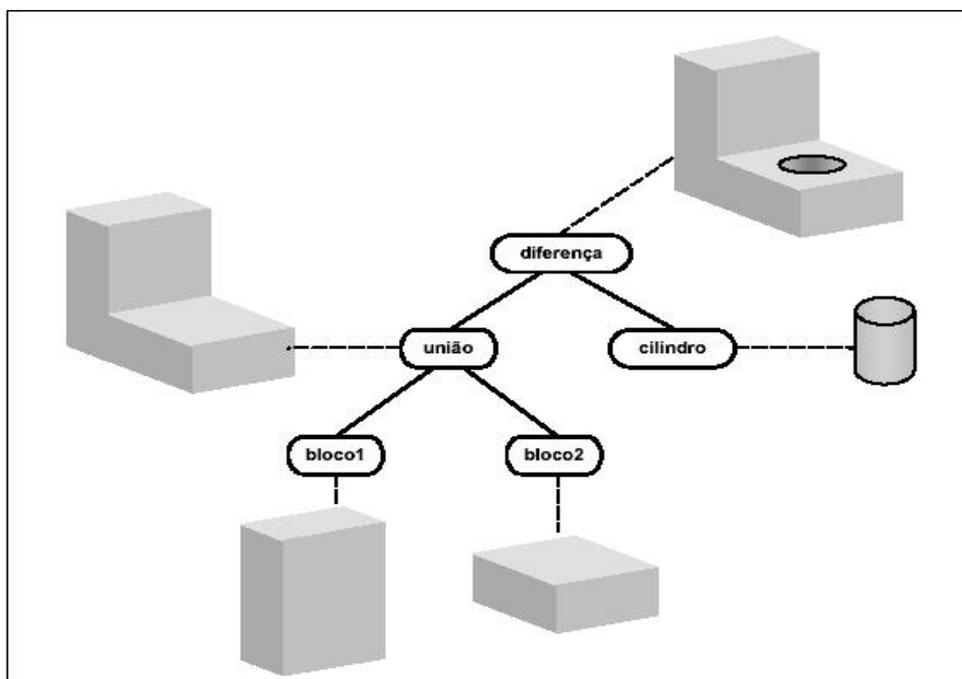
FIGURA 3-9: EXEMPLO DE SWEEP HELICOIDAL



3.5 GEOMETRIA SÓLIDA CONSTRUTIVA - CSG

Segundo Foley (1990), na técnica *Constructive Solid Geometry* (CSG) as primitivas simples são combinadas através de operadores booleanos normalizados, que são incluídos diretamente na representação. Um objeto é armazenado como uma árvore com operadores nos nodos internos, e primitivas simples nas folhas. Alguns nodos representam operadores booleanos, enquanto outros executam transformações como rotação, translação e escala. A representação CSG é uma árvore binária ordenada conforme fig. 3-10.

FIGURA 3-10: ÁRVORE CSG



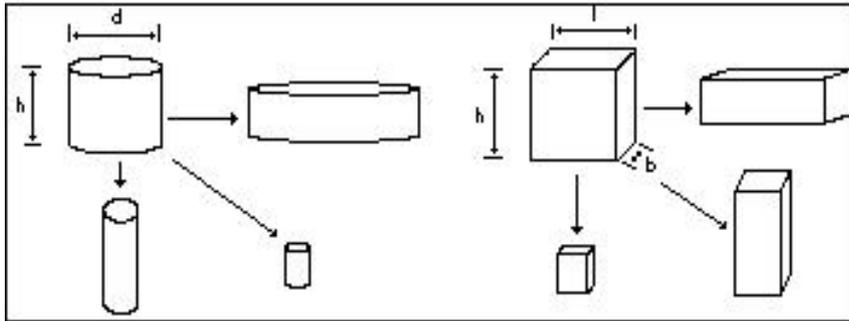
Após transformar e posicionar devidamente as primitivas no espaço, o sistema CSG utiliza as operações booleanas para combinar as diversas primitivas e criar o modelo final.

3.6 INSTANCIAMENTO DE PRIMITIVAS

Segundo Silva (1999), neste caso, o sistema de modelagem define um conjunto de primitivas sólidas 3D relevantes a uma área de aplicação. Estas primitivas são parametrizadas não apenas em termos de transformações, mas também em termos de outras propriedades. Por exemplo, uma primitiva de um objeto pode ser uma pirâmide regular com um número de faces definido pelo usuário. O instanciamento de primitivas é usado normalmente para objetos complexos, como engrenagens ou parafusos, que seriam difíceis de definir por outras técnicas, que pelas quais também não seria possível definir um bom nível de parâmetros.

Uma primitiva parametrizada pode ser vista como a definição de uma família de peças cujos membros variam em poucos parâmetros, que é um importante conceito de CAD conhecido como tecnologia de grupo. A fig. 3-11 representa o instanciamento de primitivas de objetos do tipo cilindro e de objetos do tipo bloco.

FIGURA 3-11: EXEMPLO DE PRIMITIVAS



4 SOFTWARE 3D SOLIDWORKS

Neste capítulo serão abordadas informações sobre o principal produto da empresa SolidWorks *Corporation*, também serão abordados alguns conceitos sobre as ferramentas disponíveis para modelagem de sólidos 3D no SolidWorks, em seguida será apresentada uma ferramenta CAE, PDM, CAM que trabalha integrada ao SolidWorks. Por fim, tem-se o tópico sobre customização do SolidWorks.

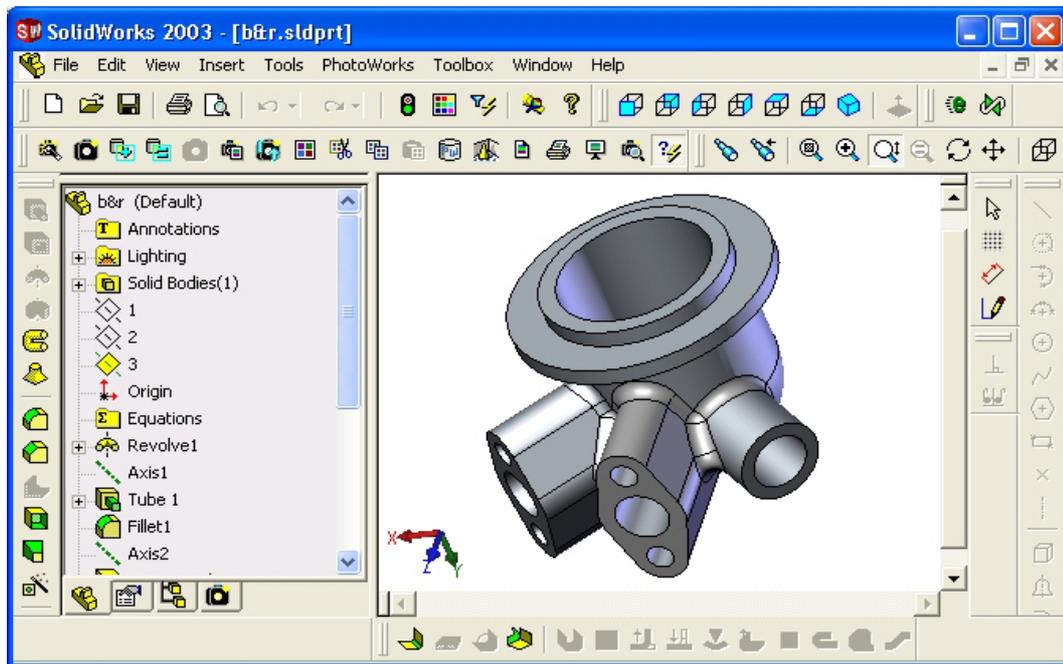
4.1 A EMPRESA SOLIDWORKS CORPORATION

A SolidWorks *Corporation* é uma empresa do grupo *Dassault Systemes SA*, desenvolve e comercializa soluções de software para projetos mecânicos. Fundada em 1993, a missão da SolidWorks é distribuir ferramentas de produtividade 3D a todos que trabalham nas áreas de projeto e manufatura. A SolidWorks distribui seus produtos em mais de 70 países, através de uma rede composta por mais de 300 revendas (SOLIDWORKS, 1995).

4.2 O SOFTWARE SOLIDWORKS 2003

O SolidWorks (fig. 4-1) é um software de CAD, que modela peças em 3D parametricamente. Permite, a partir do modelo 3D, gerar as vistas ortográficas automaticamente e também a montagem de conjuntos e dispositivos mecânicos. Portanto, trabalha com três ambientes gráficos distintos: o ambiente *part* (ambiente de modelagem), o ambiente *drawing* (ambiente de representação bidimensional) e o ambiente *assembly* (ambiente de montagem) (SOUZA, 2001).

FIGURA 4-1: INTERFACE DO SOLIDWORKS 2003



Em vários softwares atuais de modelagem, como o SolidWorks, existe um conjunto de comandos básicos que permitem que se faça a modelagem do sólido. Estes comandos podem, por sua vez, ser divididos em dois grupos:

- a) comandos básicos de construção: *Extrude*, *Revolve*, *Sweep*, *Loft* e *Rib*. Com exceção do *Rib*, que apenas acrescenta material, cada um destes comandos pode ser utilizado tanto para acrescentar material quanto para retirar material. Estes comandos necessitam de um ou mais, desenho ou esquema bidimensional (*sketch*) que é executado sobre um plano auxiliar, ou sobre uma superfície do próprio objeto;
- b) comandos básicos de edição: *Fillet*, *Chanfer*, *Shell*, *Draf*, e *Hole*. Os comandos básicos de edição, por sua vez, modificam o modelo sem a necessidade de um desenho auxiliar (*sketch*) e atuam diretamente no objeto.

Segundo RenderWorks (1998), o SolidWorks possibilita aos projetistas e engenheiros, transformarem suas idéias em projetos completos, através de ferramentas de fácil aprendizado. Destinado a áreas como:

- a) Máquinas: O SolidWorks é o software de CAD tridimensional que gera os menores arquivos, que ocupa o menor volume de disco na instalação e o que abre

arquivos em menor tempo. Conseqüentemente é o que exige os menores requisitos de hardware para trabalhos com grandes modelos.

- b) Chapas: O módulo para dobra e desenvolvimento de chapas vem integrado ao SolidWorks como ferramenta nativa.
- c) Moldes: O SolidWorks possui ferramentas específicas para geração de cavidades de moldes.
- d) Móveis: A indústria moveleira está se aproximando cada vez mais da mecânica em termos de precisão de montagens de componentes. O SolidWorks é um software CAD 3D para o desenvolvimento de móveis e possui recursos para o desenvolvimento de catálogos e manuais de montagem (fig. 4-2).
- e) Elétrica: O Software EmbassyWorks, é um produto para projeto de cabeamentos e fiações.
- f) *Design*: Com um poderoso modelador de sólidos com superfícies integradas, e recursos de renderização e animação, o SolidWorks é uma excelente ferramenta para designers.

FIGURA 4-2: EXEMPLO DE CONSTRUÇÃO DE MÓVEIS



FONTE: RenderWorks (1998)

O SolidWorks 2003 roda em DESKTOP com processadores classe Intel Pentium ou AMD Athlon com 128 MG de memória RAM. O sistema operacional pode ser Microsoft Windows XP Professional, Windows 2000 (recomendado), Windows NT, Windows Me ou Windows 98 Second Edition (SOLIDWORKS, 1995).

Conforme SolidWorks (1995), por definição o SolidWorks lê e grava vários tipos de arquivos conforme quadro 4-1.

QUADRO 4-1: ARQUIVOS RECONHECIDOS PELO SOLIDWORKS 2003

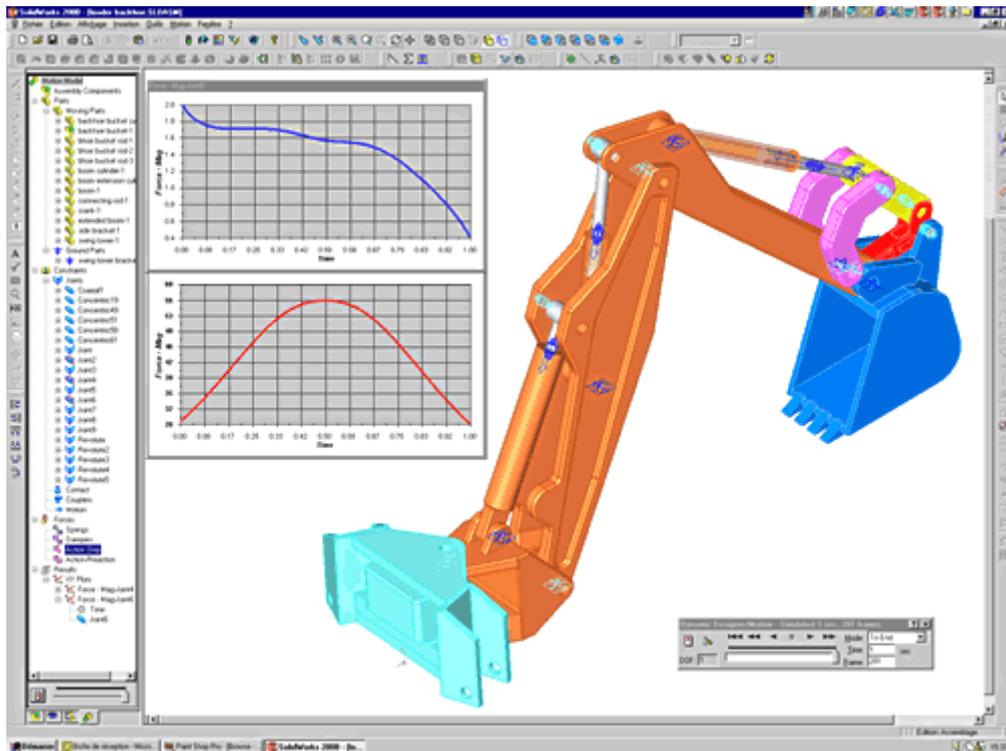
2D DXF/DWG	importa e exporta para AutoCAD até versão 2002
3D DXF	importa
STEP	importa e exporta até versão AP203, importa para versão AP214
IGES	importa e exporta
STL	importa e exporta
ACIS (*.SAT)	importa e exporta
Parasolid X_T files & X_B (Parasolid Binary)	importa e exporta
VDAFS (*.VDA)	importa e exporta
TIF	exporta
JPEG	exporta
Pro/ENGINEER	importa e exporta
Unigraphics	importa
Mechanical Desktop	importa
Autodesk Inventor Part	importa
Solid Edge Part	importa
CADKEY	importa
VRML	importa e exporta
RealityWave ZGL	exporta
CGR	exporta
HCG	exporta
HOOPS HSF	exporta

O SolidWorks pode ser customizado através de *Application Programming Interface* (API), com suporte para Visual Basic e Microsoft Visual C++ (SOLIDWOKS, 1995). Com este recurso , o SolidWorks pode ser customizado para realizar tarefas que ajudaram a dar mais produtividade, em situações específicas, dentro empresas.

4.3 COSMOS/WORKS (CAE)

É o sistema de CAE (fig. 4-3) escolhido pela maior parte dos usuários do SolidWorks sendo capaz de realizar análises estáticas e dinâmicas de tensões e deformações, análises de frequência, análises térmicas com grande facilidade sem ter a necessidade de transferir dados, pois a interface do Cosmos/Works é embutida no SolidWorks e trabalha com seus arquivos nativos (ASSESSOCON, 1987).

FIGURA 4-3: TELA DO COSMOSWORKS INTEGRADA COM O SOLIDWORKS



FONTE: Assessocon (1987)

4.4 SMARTTEAM (PDM)

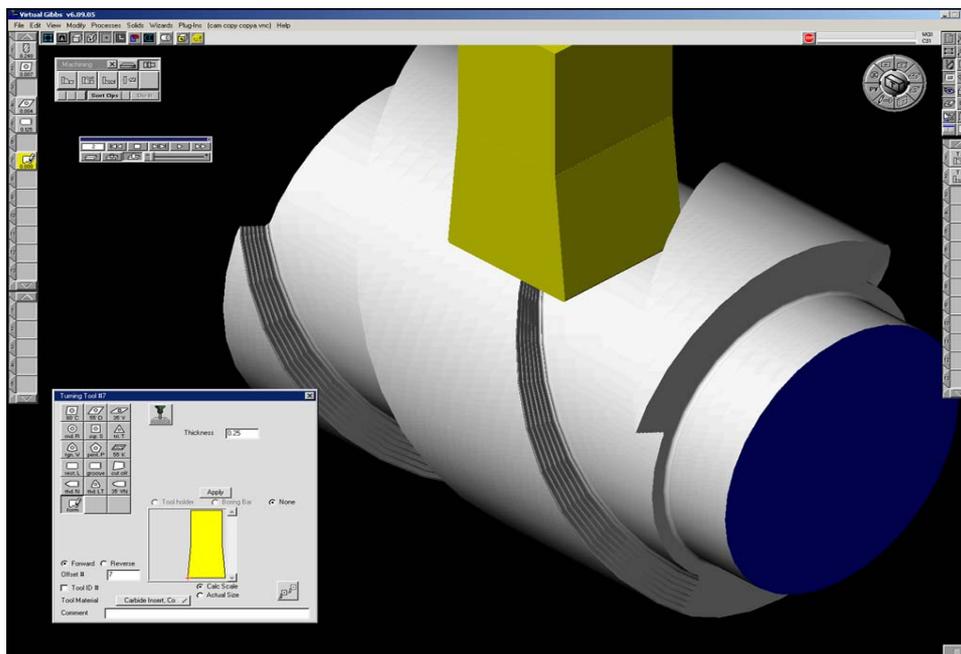
SmarTeam é a uma solução PDM totalmente aberta e a mais integrada do mercado ao SolidWorks, permitindo através de um menu na própria aplicação CAD o gerenciamento de configurações, peças, conjuntos e desenhos. Disponibilizando o mapeamento bidirecional com carimbos, blocos de revisão, "*custom properties*" etc. Possibilitando a qualquer momento a verificação de "quem está usando" determinado objeto, além de oferecer informações "onde é utilizado" e "composto de". Integrando-se ainda com vários sistemas CAD do mercado, o SmarTeam conta também com um poderoso visualizador embutido com recursos para anotações em desenho ("*markup*"), uma solução Web para acesso remoto via Internet/Intranet,

um módulo específico para controle e automação do fluxo de trabalho (workflow), integração a sistemas ERP, CRM e soluções colaborativas de e-business para comunicação com os parceiros da empresa (ASSESSOCON, 1987).

4.5 GIBBSCAM (CAM)

Segundo RenderWorks (1998), o GibbsCAM é um software de CAM baseado em PCs. Possui uma interface gráfica, interativa que permite ao usuário facilmente acessar funções de geometria, ferramentas estratégias de usinagem, simulação e pós-processamento. Com métodos flexíveis de programação o usuário pode criar rapidamente programas CNC otimizados para as suas máquinas. Assistentes tecnológicos integrados auxiliam na execução de tarefas comuns como determinação de matérias-prima ou criação de múltiplos processos de furação e mandrilamento com poucos cliques. A fig. 4-4 mostra a interface do software GibbsCam.

FIGURA 4-4: INTERFACE DO GIBBSCAM



FONTE: GibbsCAM (1982)

4.6 CUSTOMIZAÇÃO DO SOLIDWORKS

O software SolidWorks possui documentação com exemplos e informações sobre como customizar através API, na qual pode-se usar automatizar e personalizar o SolidWorks. A documentação da API pode ser baixada da página da empresa do software ou, se o usuário estiver com a versão SolidWorks 2003, já terá a documentação da API embutida no software. A documentação API contém centenas de funções que podem ser chamadas no Visual Basic (VB), Visual Basic for Application (VBA), C, C++, ou arquivos macro do SolidWorks. Estas funções provêm acesso direto à funcionalidade de SolidWorks, tais como, criar uma linha, criar modelos, inserir atributos ou informações ao modelo, verificar os parâmetros de uma superfície e outros.

Todo objeto disponível no SolidWorks API tem documentado suas propriedades e métodos associados. É bom lembrar que para o uso do API é necessário que o programador esteja familiarizado com o Visual Basic, C ou C++, pois os exemplos estão todos relacionados a estas linguagens.

O SolidWorks possui funções *macro* em sua interface para criação de programas para VBA. Onde, na *toolbar* fig. 4-5 os comandos da mesma em seqüência são: Run Macro, Stop macro, Record/Pause macro e Edit Macro. Então quando coloca-se a macro para gravar, as operações executadas com o SolidWorks geram o programa com as chamadas equivalentes às funções do API. Em seguida, pode-se abrir a macro criada automaticamente no VBA e editá-la se necessário, e com isto ganhar um bom tempo de programação.

FIGURA 4-5: TOOLBAR MACRO



A seguir (quadro 4-2) é apresentado o conteúdo de um arquivo fonte de um programa, criado automaticamente durante a construção de um sólido 3D, onde foi criado um modelo através de um *sweep* translacional simples, tendo como base geradora um retângulo sem dimensões definidas, e apenas foi informada a dimensão de translação, que é representada no programa como extrusão. Verificando o código fonte gerado automaticamente, apresentado

no quadro 4-2 nota-se o quando de programação pode ser automatizada, caso programador utilizar o mesmo como base para construir um novo programa. Apenas é necessário seguir utilizando o SolidWorks e gravando a macro, para que as funções equivalentes sejam criadas no código do programa. Neste caso é bom que o programador tenha conhecimentos sobre modelagem geométrica e também sobre ferramentas CAD de modelagem 3D.

QUADRO 4-2: EXEMPLO DE CÓDIGO FONTE CRIADO AUTOMATICAMENTE

```

Dim swApp As Object
Dim Part As Object
Dim boolstatus As Boolean
Dim longstatus As Long
Dim Annotation As Object
Dim Gtol As Object
Dim DatumTag As Object
Dim FeatureData As Object
Dim Feature As Object
Dim Component As Object

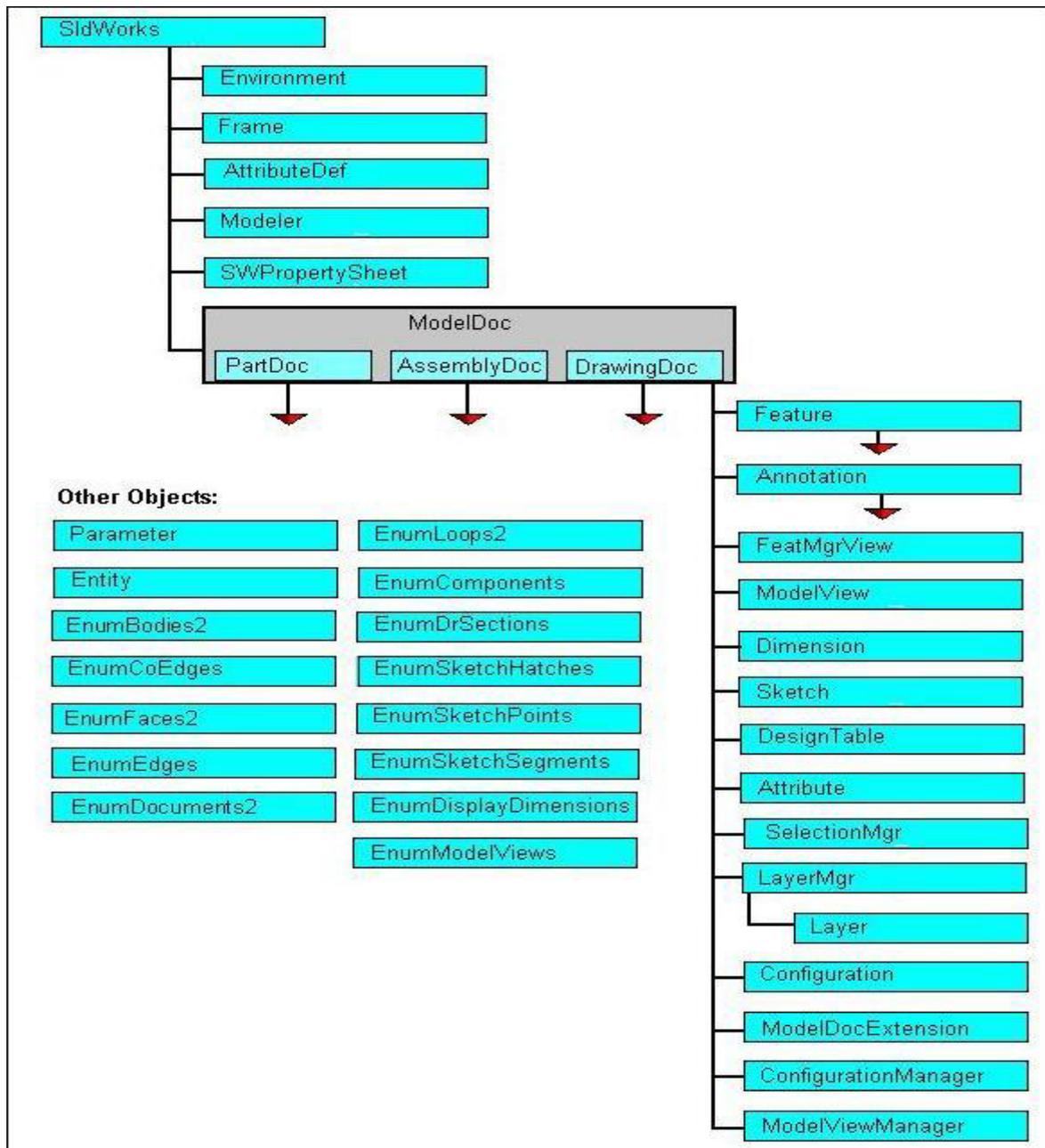
Sub main()
Set swApp = Application.SldWorks
Set Part = swApp.ActiveDoc
Part.InsertSketch2 True
Part.SketchRectangle -0.02034845938375, 0.02426162464986, 0, 0.02778347338936, _
- 0.02034845938375, 0, 1
Part.ClearSelection
boolstatus = Part.Extension.SelectByID("Line2", "SKETCHSEGMENT", 0 , 0, 0, False, _
0, Nothing)
boolstatus = Part.Extension.SelectByID("Line1", "SKETCHSEGMENT", 0 , 0, 0, True, _
0, Nothing)
boolstatus = Part.Extension.SelectByID("Line4", "SKETCHSEGMENT", 0 , 0, 0, True, _
0, Nothing)
boolstatus = Part.Extension.SelectByID("Line3", "SKETCHSEGMENT", 0 , 0, 0, True, _
0, Nothing)
Part.FeatureManager.FeatureExtrusion True, False, False, 0, 0, 0.01, 0.01, False, False, False, False,
0.01745329251994 , 0.01745329251994, False, False, False, False, 1, 1, 1
Part.SelectionManager.EnableContourSelection = 0
End Sub

```

Através do Visual C++ é possível gerar programas em arquivos DLL (*Dynamic Link Library*) que podem ser abertos pelo SolidWorks ou incorporadas ao ambiente, registrando esta DLL no Windows. Para este caso é necessário baixar um *Software Developer's Kit* (SDK) da página da SolidWorks Corporation que cria um gabarito (*template*) de construção dentro do Microsoft Visual C++. Também pode-se criar arquivos executáveis no ambiente Visual C++, baixando os fontes necessários para conexão com SolidWorks da página da empresa, e assim criar aplicativos MFC appWizard(.exe). Demais bibliotecas (*includes*) necessárias para compilar as aplicações se encontram no diretório de instalação do SolidWorks.

A seguir na fig. 4-6 é apresentado o diagrama de visão do API Objetos do SolidWorks 2003.

FIGURA 4-6: SOLIDWORKS API OBJETOS



FONTE: SolidWorks (1995)

Como se observa na fig. 4-6, o objeto API principal é o SldWorks que é o objeto de mais alto nível no SolidWorks API, ele provê e dirige o acesso a todos os outros objetos do SolidWorks API. No SolidWorks, há também três tipos de documentos principais: *Parts*, *Drawings* e *Assemblies* que pertencem ao objeto ModelDoc. Cada tipo de documento tem seu próprio API sendo para *Part* (PartDoc), para *Drawing* (DrawingDoc) e para *Assembly*

(*AssemblyDoc*) com suas próprias funções relacionadas. Por exemplo, a função de *AddComponent* existe somente no objeto de *AssemblyDoc*, porque é específica a adicionar componentes nos documentos de montagem(*assembly*). Então de uma forma geral, estes são os objetos principais do objeto API do SolidWorks.

O próximo capítulo apresenta com maiores detalhes a implementação do protótipo para acrescentar atributos nos sólidos do SolidWorks, que constitui um dos objetivos citados na seção 1.2.

5 DESENVOLVIMENTO DO TRABALHO

Neste capítulo serão apresentados os requisitos, especificação, implementação e operacionalidade do protótipo, identificando a necessidade real que ocorre em empresas de projeto e manufatura de artefatos mecânicos. Os processos de projeto podem ser otimizados através da customização do software 3D SolidWorks para acrescentar atributos nos sólidos criados por este software, para o preenchimento automático de legendas de desenho técnico, garantindo que as informações dos atributos contidas nos sólidos estejam sempre atualizados com a documentação do desenho técnico.

5.1 REQUISITOS DO PROTÓTIPO

A construção do protótipo visa aumentar a produtividade de um determinado setor de uma empresa de projetos, onde nota-se que os projetistas perdem muito tempo cadastrando informações (atributos) a modelos criados pelo software 3D SolidWorks. Estas informações cadastradas são utilizadas em etapas seguintes dos processos de projeto e manufatura, onde a necessidade atual, no momento é a de preenchimento automático de legendas de formatos de desenho técnico. Posteriormente, as informações cadastradas podem ser usadas em outras necessidades, como por exemplo, no CAE, para a simulação CAE, onde os dados podem ser recuperados de forma semi-automática antes da simulação, e assim aumentar a produtividade em se verificar informações referentes ao modelo, pois estas já estarão incorporadas aos próprios sólidos que compõem o modelo.

O protótipo deve auxiliar o projetista no cadastramento dos atributos, que são:

- a) projetista: nome do projetista responsável pelo desenho da peça (Modelo 3D);
- b) verificador: nome do responsável pela peça, caso o projetista estiver ausente durante a fabricação da peça, sendo que este também verifica a peça antes da sua fabricação, analisando se não haverá problemas de fabricação;
- c) tolerância: este atributo representa a tolerância numérica para precisão de medidas para manufatura da peça. Existe um valor padrão para este campo, sendo 0,5mm, mas o projetista pode alterá-lo conforme a necessidade da peça;

- d) escala: é a representação de quanto o desenho da peça é aumentado ou diminuído no documento que é enviado para fabricação, permanece um valor padrão, sendo %, pouco usual, mas é possível ser usado;
- e) denominação Superior: neste campo é informado o tipo de produto e sua particularidade;
- f) denominação Inferior: aqui o projetista informa o nome da peça que foi criada;
- g) tratamento: neste campo o projetista seleciona o tipo de tratamento que a peça deverá sofrer, como por exemplo “bicromatizar”, “fosfatizar” e “jatear”.

O atributo material é informado pelo projetista dentre vários tipos de materiais de estoque da empresa, sendo que estes materiais são divididos em quatro grupos, onde projetista tem uma tabela com a descrição do material, peso específico, comprimento, largura e espessura do material. Três destes quatro grupos são quantificados em kg (chapa em kg, alumínio em kg e barra em kg) e um dos tipos é quantificado por metros quadrados (chapa em m^2).

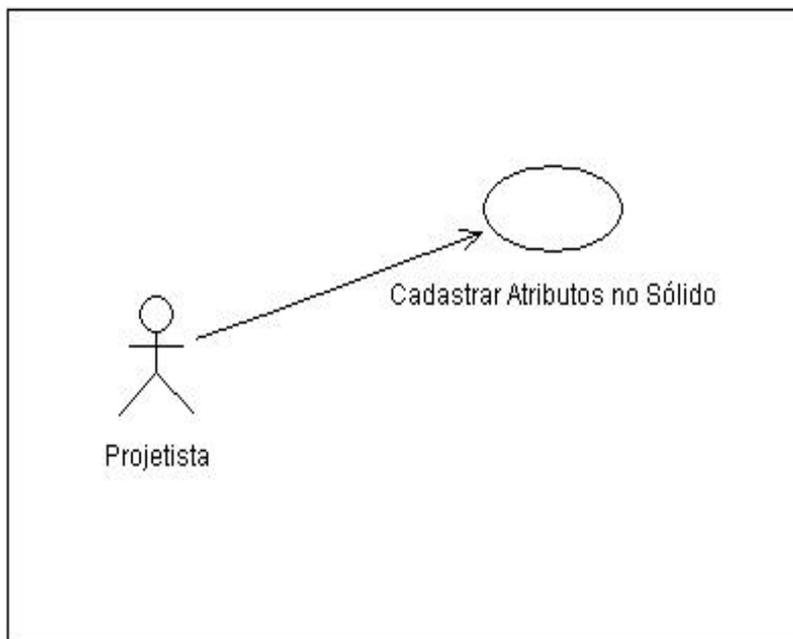
Também é necessário que o projetista calcule o peso do material. Para o cálculo do peso da chapa em Kg, é feita a multiplicação de seu peso específico pela área superior da peça desenvolvida (planificada). Para o cálculo do peso da barra em Kg, é feito seu peso específico vezes o volume, e, por fim, o cálculo da quantidade da chapa em m^2 é igual a área superior da peça. Caso o projetista necessite usar um material que não é padrão da empresa, ele não calcula o peso ou a quantidade de material, somente informa a descrição do material.

Para o atributo acabamento o projetista escolhe um dos cinco tipos padrões da empresa, que ele encontra também em uma tabela que contém a descrição do acabamento, fator de calculo da quantidade de acabamento, sendo quatro dos cinco tipos são tintas (Ral 7032, Ral 7022, Laranja Munsell, cinza Munsell N6,5) e um tipo estanho (estanhagem eletrolítica). O projetista também precisa calcular a quantidade de acabamento. Para tintas, é igual a área superior da peça vezes 0.4125, em Kg Para o cálculo da estanhagem, é igual a área total da peça em metros quadrados. Caso o projetista informe um tipo de acabamento que não é o padrão da empresa, não será calculada a quantidade de acabamento.

5.2 ESPECIFICAÇÃO DO PROTÓTIPO

Para a especificação do protótipo foi utilizada a ferramenta Rational Rose versão 4.0 Demo, utilizando a linguagem para modelagem de sistemas *Unified Modeling Language* (UML). A seguir, são apresentados os principais diagramas criados com auxílio da ferramenta Rational Rose. Na fig. 5-1 é apresentado o diagrama de casos de uso.

FIGURA 5-1: DIAGRAMA DE CASOS DE USO

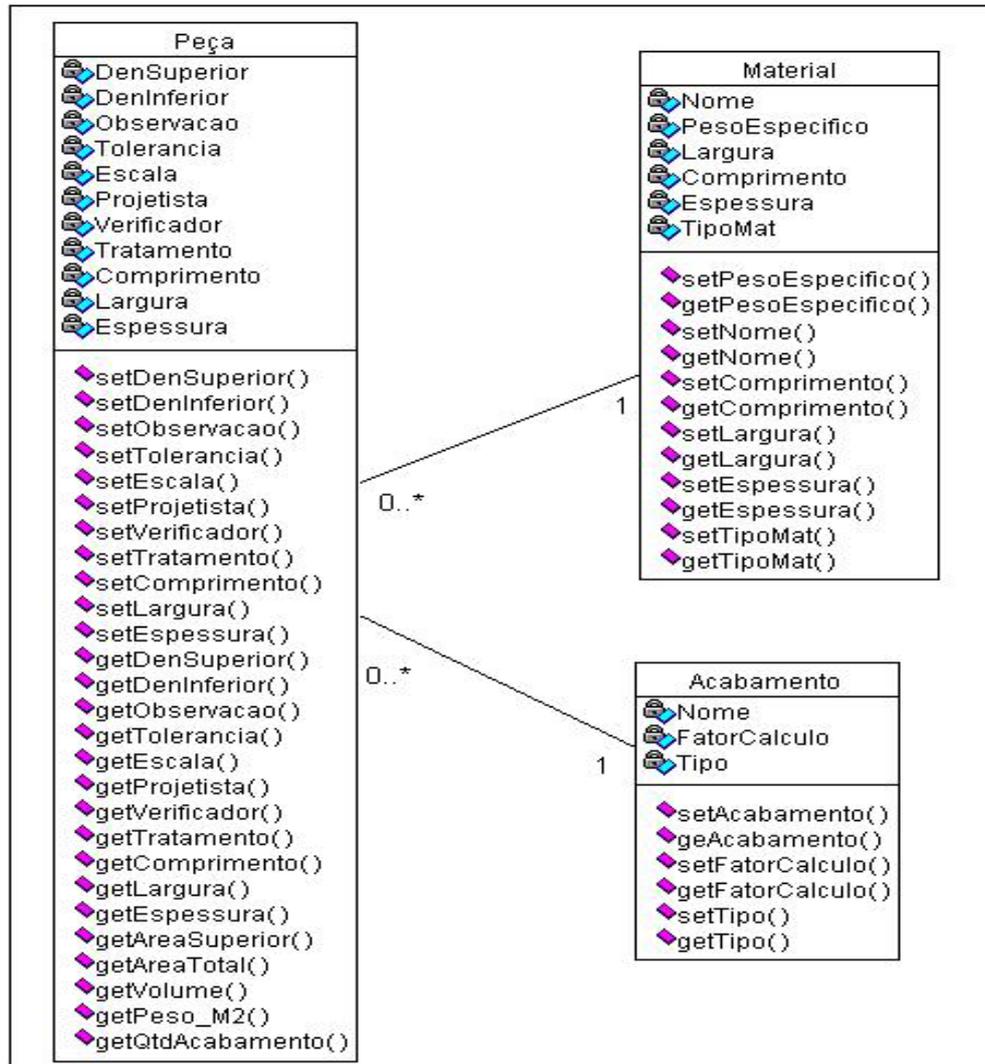


Descrição dos casos de uso:

- a) cadastrar atributos no sólido: Baseado na função da peça que está desenvolvendo, o projetista cadastra os atributos nesta peça (sólido 3D) em que acabou de modelar na ferramenta CAD 3D SolidWorks.
- b) projetista: Representa o projetista que desenvolveu o sólido, a partir da ferramenta CAD 3D SolidWorks.

Na fig. 5-2 é representado o diagrama de classes referente ao problema levantado pela empresa, para construção do protótipo.

FIGURA 5-2: DIAGRAMA DE CLASSES



A classe *peça* representa as informações gerais da peça, e é nesta classe que estão os métodos para calcular quantidade de acabamento (quadro 5-1) e a quantidade de material (quadro 5-2). Esta classe contém informações específicas da peça que o projetista está desenvolvendo.

QUADRO 5-1: MÉTODO QUANTIDADE DE ACABAMENTO

```

double Peca::getQtdAcabamento(double fatorCalc)
{
    if (fatorCalc == 1.0)
        return fatorCalc * getAreaTotal()/1000000; //acabamento para peca estanhada
                                                //transformado para metros
    else if(fatorCalc == 0.4125)
        return fatorCalc * getAreaSuperior()/1000000; //acabamento para peca pintada
                                                //transformado para metros

    else
        return 0;
}

```

QUADRO 5-2: MÉTODO QUANTIDADE DE MATERIAL

```

double Peca::getPeso_M2(double pesoEsp, CString tipoMat)
{
    double peso = 0;

    //material em metros quadrados - transformado para metros
    if (tipoMat == "chapam2")
        peso = getAreaSuperior()/1000000;

    //material em kg "barra" - transformado para metros
    else if (tipoMat == "barraKg")
        peso = (pesoEsp * getVolume())/1000000000;

    // material em Kg "chapa" ou "aluminio" - transformado para metros
    else if (tipoMat == "chapaKg" || tipoMat == "aluminioKg")
        peso = (pesoEsp * getAreaSuperior())/1000000;

    return peso;
}

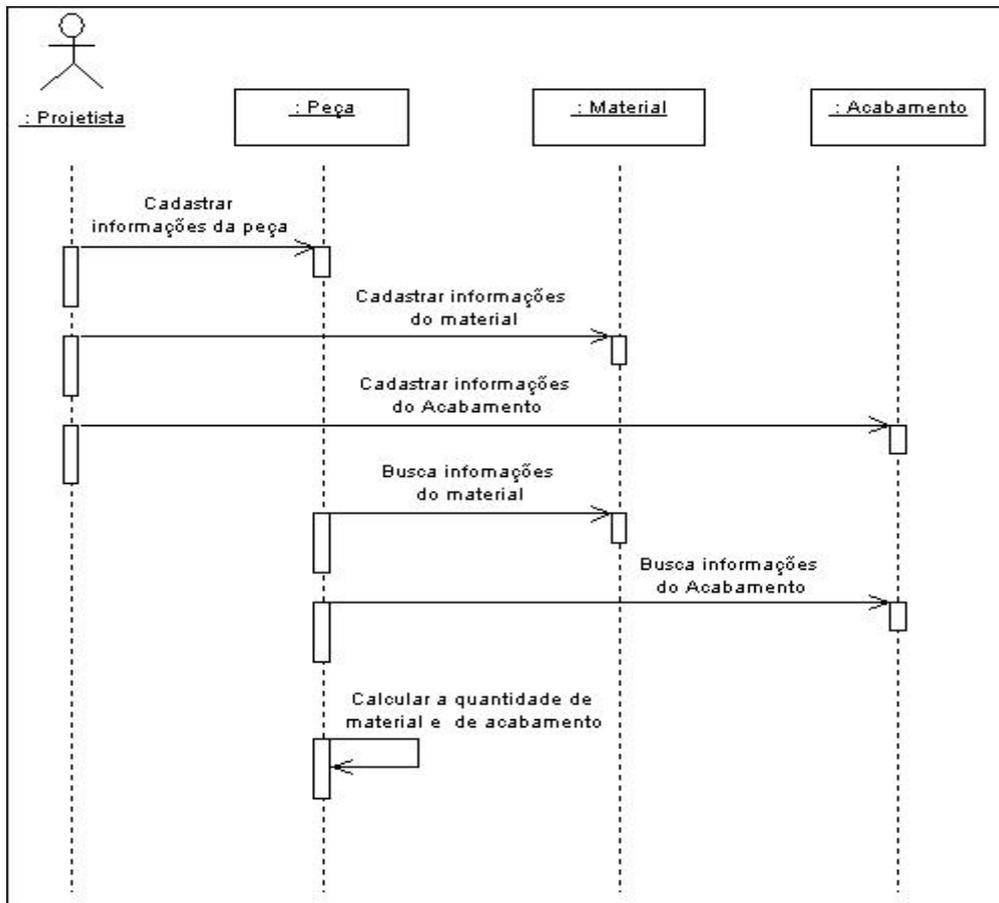
```

A classe *material* representa as informações sobre a matéria prima para a construção da peça, como: peso específico para calcular a quantidade de material, dimensões da peça em estoque, tipo de material e nome.

A classe *acabamento* representa as informações sobre o acabamento que será utilizado na peça projetada, como: nome do acabamento, fator de cálculo para calcular a quantidade de acabamento e o tipo de acabamento.

A fig. 5-3 apresenta o diagrama de seqüência do protótipo para acrescentar atributos ao modelo criado pelo SolidWorks.

FIGURA 5-3: DIAGRAMA SE SEQÜÊNCIA



5.3 IMPLEMENTAÇÃO

Nesta seção serão descritas as ferramentas utilizadas para construção do protótipo, bem como as técnicas utilizadas para solucionar problemas encontrados durante a programação, qual foi a solução encontrada para acrescentar atributos nos modelos 3D do SolidWorks. Também será apresentada a operacionalidade do protótipo.

5.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

O protótipo foi implementado na linguagem C++ em ambiente de programação Microsoft Visual C++ versão 6.0. O protótipo foi construído utilizando a biblioteca *Microsoft Foundation Class* (MFC). O software 3D escolhido para ser customizado foi o SolidWorks 2003, com a respectiva API.

Para se conectar ao SolidWorks foi utilizado o código fonte (quadro 5-3) fornecido na página da empresa SolidWorks Corporation, no campo *API support*. Os demais arquivos-

fonte para compilar os programas no ambiente Visual C++ 6.0, estavam disponíveis nos diretórios de instalação do SolidWorks: “SolidWorks\samples\AppComm”, “SolidWorks\samples\ComuUserdll” e “SolidWorks\samples\userdll”.

QUADRO 5-3: FONTE PARA SE CONECTAR AO SOLIDWORKS

```

bool CAtributosApp::Connect()
{
    LPUNKNOWN          pUnk = NULL;
    LPDISPATCH        dSldWorks;
    CLSID              clsid;
    int                _convert = 0;
    bool               status = true;

    if (CLSIDFromProgID(OLESTR("SldWorks.Application"), &clsid) != NOERROR )
    {
        status = FALSE;
    }
    else if (GetActiveObject(clsid, NULL, &pUnk) != NOERROR)
    {
        status = false;
    }
    else if ( pUnk->QueryInterface(IID_IDispatch, (LPVOID*)&dSldWorks) != NOERROR )
    {
        pUnk->Release();
        status = false;
    }
    else
    {
        m_SldWorks.AttachDispatch(dSldWorks);
        pUnk->Release();
    }
    return status;
}
ISldWorks* CAtributosApp::GetSldWorks()
{
    return &theApp.m_SldWorks;
};

```

FONTE: SolidWorks (1995)

Como mencionado no tópico customização do SolidWoks, para Visual C++ poderia-se gerar arquivo “.exe” ou arquivo “.dll”. Pela facilidade de executar os teste no ambiente Visual C++, optou-se por gerar arquivo executável (.exe), pois quando criado arquivos “.dll” era necessário fechar o aplicativo SolidWorks e abri-lo novamente, carregar a dll e então executar os testes. Já com o arquivo “.exe” os testes puderam ser feitos a qualquer momento, apenas abria-se o aplicativo SolidWoks e executava-se o protótipo quando necessário.

Inicialmente criou-se as classes necessárias para implementação do protótipo conforme especificação vista anteriormente. Foi criada a classe “peça” (quadro 5-4) que possui os atributos gerais do problema da empresa. Nesta classe foram incluídos métodos para

capturar e devolver as informações referentes aos atributos gerais, também é nesta classe que está implementado os métodos para cálculo de quantidade de material e cálculo de quantidade de acabamento.

QUADRO 5-4: CLASSE PECA

```

class Peca
{
    CString denSuperior;
    CString denInferior;
    CString observacao;
    CString tolerancia;
    CString escala;
    CString projetista;
    CString verificador;
    CString tratamento;
    double comprimento;
    double largura;
    double espessura;
public:
    Peca();
    virtual ~Peca();
    setDenSuperior (CString m_DenSup);
    setDenInferior (CString m_DenInf);
    setObservacao (CString m_Obs);
    setTolerancia (CString m_Toler);
    setEscala (CString m_Escala);
    setProjetista (CString m_Projetista);
    setVerificador (CString m_Verificador);
    setTratamento (CString m_Tratamento);
    setComprimento (double m_Comprimento);
    setLargura (double m_Largura);
    setEspessura (double m_Espessura);
    CString getDenSuperior ();
    CString getDenInferior ();
    CString getObservacao ();
    CString getTolerancia ();
    CString getEscala ();
    CString getProjetista ();
    CString getVerificador ();
    CString getTratamento ();
    double getComprimento ();
    double getLargura ();
    double getEspessura ();
    double getAreaSuperior();
    double getAreaTotal();
    double getVolume();
    double getPeso_M2(double pesoEsp, CString tipoMat); //tanto para kg ou m2
    double getQtdAcabamento(double fatorCalc);
};

```

A classe “material” (quadro 5-5) é responsável por guardar informações sobre os materiais padrões da empresa que estão em estoque. Esta classe possui métodos para capturar e devolver informações tanto para o cálculo do peso do material, como para sua identificação.

QUADRO 5-5: CLASSE MATERIAL

```

class Material
{
    CString nome;
    double pesoEspecifico;
    double comprimento;
    double largura;
    double espessura;
    CString tipoMat;
public:
    Material();
    virtual ~Material();
    setNome (CString m_Nome);
    setPesoEspecifico(double m_PesEsp);
    setComprimento (double m_Comprimento);
    setLargura (double m_Largura);
    setEspessura (double m_Espessuara);
    setTipoMat (CString m_TipoMat);
    CString getNome();
    double getPesoEspecifico();
    double getComprimento();
    double getLargura();
    double getEspessura();
    CString getTipoMat();
};

```

E finalmente, a classe “acabamento” (quadro 5-6) é responsável por guardar as informações sobre o tipo de acabamento aplicado ao modelo criado no SolidWorks, estas informações, servem para o cálculo da quantidade de acabamento, e também para identificá-lo, na hora de incorporar os atributos do modelo 3D.

QUADRO 5-6: CLASSE ACABAMENTO

```

class Acabamento
{
    CString nome;
    double fatorCalculo; //fator usado para calcular a quantidade de acabamento
    CString tipo;
public:
    Acabamento();
    virtual ~Acabamento();
    setAcabamento (CString m_Nome);
    setFatorCalculo (double fator);
    setTipo (CString m_Tipo);
    CString getAcabamento ();
    double getFatorCalculo ();
    CString getTipo ();
};

```

A maioria das informações sobre a peça, o material e o acabamento são cadastrados pelo projetista em uma única tela do protótipo, sendo apenas calculado a quantidade de material e a quantidade de acabamento. As informações sobre o material e sobre o

acabamento serão buscadas em uma base de dados, de acordo com a opção de material e acabamento escolhido pelo projetista.

Para acrescentar os atributos ao modelo 3D, é necessário primeiro obter o documento ativo do SolidWorks e testá-lo (quadro 5-7), para verificar se não é um documento do tipo *drawing* (arquivo de documentação). Caso este documento estivesse ativo quando o projetista tentasse acrescentar os atributos, o protótipo abortaria. Então, caso o projetista tente aplicar os atributos em um *drawing*, o protótipo retorna uma mensagem para que o projetista deixe ativo um *part* (documento contendo apenas um modelo 3D), ou um *assembly* (documento com um ou mais de um modelo 3D), que são os documentos permitidos para execução da tarefa.

QUADRO 5-7: TESTE PARA VERIFICAR SE É UM DRAWING

```
//pega o modelo ativo do solidworks
dModelDoc = CAtributosApp::GetSldWorks()->GetActiveDoc();
if( dModelDoc != NULL)
{
    ModelDoc.AttachDispatch(dModelDoc);

    //Verifica se o arquivo ativo do SolidWorks não é um Drawing (documento desenho)
    if ((ModelDoc.GetType() != swDocPART) && (ModelDoc.GetType() != swDocASSEMBLY))
    {
        AfxMessageBox("*** Arquivo Ativo não é um Modelo ou uma Montagem ***");
        return false;
    }
}
```

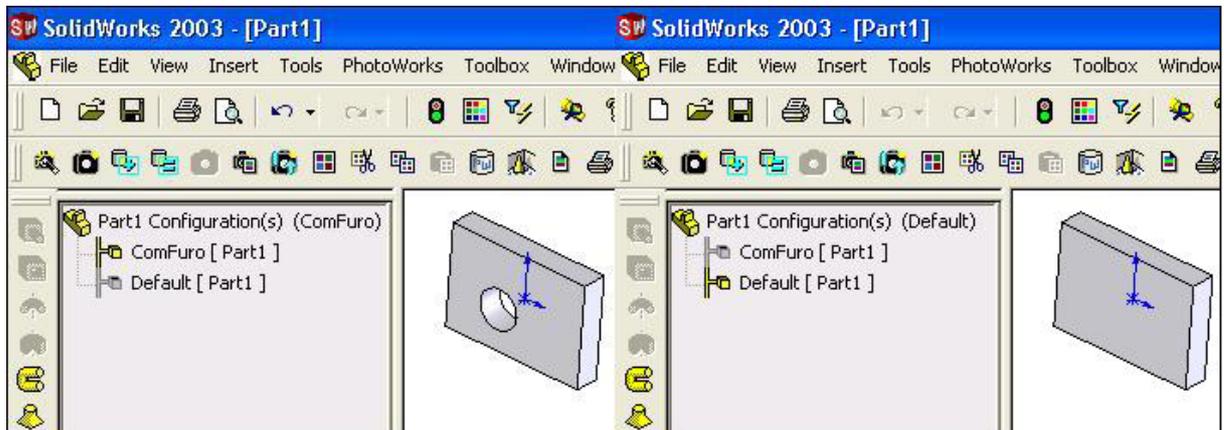
Em seguida há outro problema, onde, *part* ou *assembly*, podem possuir mais de uma configuração, ou seja, tendo como exemplo um modelo de um suporte, este serviria para vários disjuntores, sendo apenas necessário fazer pequenas alterações no suporte. Então o projetista não cria um outro modelo, apenas cria uma nova configuração de peça, e faz as alterações necessárias no mesmo documento. Assim foi necessário buscar (quadro 5-8) a configuração atual do documento ativo, para que nesta configuração fossem acrescentados os atributos, não interferindo nas demais configurações.

QUADRO 5-8: CAPTURANDO CONFIGURAÇÃO ATUAL

```
//seleciona o tipo de configuração atual do modelo
//sendo que é nesta configuração que será colocado os atributos
if ( (dConfiguration = ModelDoc.GetActiveConfiguration()) != NULL )
{
    Configuration.AttachDispatch(dConfiguration);
    config = Configuration.GetName();
}
```

Um exemplo de configurações de peças no SolidWorks, é representado na fig. 5-4, onde existe uma peça na configuração *Default* (criado automaticamente pelo SolidWorks) e outra na configuração *ComFuro* (adicionada pelo projetista).

FIGURA 5-4: EXEMPLO DE CONFIGURAÇÃO



Resolvidos os problemas acima, na seqüência, foram identificados os métodos necessários para acrescentar os atributos no modelo. Como não existia um método para editar atributos previamente existentes no modelo 3D, foi utilizado um outro método denominado de *DeleteCustomInfo2* (quadro 5-9), usado para apagar os atributos do modelo 3D. Este método só apaga os atributos com nomes iguais aos que serão usados pelo protótipo. No primeiro campo do método *DeleteCustomInfo2(configuração, nome do atributo)* é enviado o nome da configuração atual do modelo 3D, representado pela variável “config” no protótipo, e no segundo é enviado o nome do atributo que se queira apagar, no caso deste trabalho, serão os nomes dos atributos que posteriormente serão acrescentados no modelo 3D.

QUADRO 5-9: APAGANDO ATRIBUTOS

```
//Limpa os atributos do modelo, se existirem para poder inserir novos, caso contrario não funciona.
ModelDoc.DeleteCustomInfo2(config, "Material");
ModelDoc.DeleteCustomInfo2(config, "Tratamento");
ModelDoc.DeleteCustomInfo2(config, "Acabamento");
ModelDoc.DeleteCustomInfo2(config, "Observação");
ModelDoc.DeleteCustomInfo2(config, "Tolerancia");
ModelDoc.DeleteCustomInfo2(config, "Escala");
ModelDoc.DeleteCustomInfo2(config, "Qtd Acabamento");
ModelDoc.DeleteCustomInfo2(config, "Qtd Material");
ModelDoc.DeleteCustomInfo2(config, "Denominação Superior");
ModelDoc.DeleteCustomInfo2(config, "Denominação Inferior");
ModelDoc.DeleteCustomInfo2(config, "Projetista");
ModelDoc.DeleteCustomInfo2(config, "Verificador");
```

E finalmente, foi então trabalhado no método que acrescentaria os atributos escolhidos pelo projetista, sendo ele denominado de *AddCustomInfo3*. No método

AddCustomInfo3(configuração, nome do atributo, tipo de valor do atributo , valor do atributo) (quadro 5-10), o primeiro campo (configuração) é informado o nome da configuração atual do modelo 3D, representada pela variável “config” do protótipo, no segundo campo (nome do atributo) é informado o nome do atributo que se queira acrescentar no modelo 3D, no terceiro campo (tipo de valor do atributo) é definido o tipo de valor do atributo que será enviado, sendo estes valores podem ser entre: *swCustomInfoUnknown* (para tipo desconhecido), *swCustomInfoText* (para texto), *swCustomInfoDate* (para data), *swCustomInfoNumber* (para números) ou *swCustomInfoYesOrNo* (para resposta verdadeiro ou falso) e por último no quarto campo (valor do atributo), é enviado o valor que será acrescentado ao atributo correspondente. No caso do protótipo, serão acrescentados tipos de valores *swCustomInfoText*, que conterão apenas textos como valor do atributo.

QUADRO 5-10: ACRESCENTANDO ATRIBUTOS

```
//Adiciona os atributos ao modelo
ModelDoc.AddCustomInfo3(config,"Material",swCustomInfoText,T_Material.getNome());
ModelDoc.AddCustomInfo3(config,"Tratamento",swCustomInfoText,T_Peca.getTratamento());
ModelDoc.AddCustomInfo3(config,"Acabamento",swCustomInfoText,T_Acabamento.getAcabamento());
ModelDoc.AddCustomInfo3(config,"Observação",swCustomInfoText,T_Peca.getObservacao());
ModelDoc.AddCustomInfo3(config,"Tolerancia",swCustomInfoText,T_Peca.getTolerancia());
ModelDoc.AddCustomInfo3(config,"Escala",swCustomInfoText,T_Peca.getEscala());

//referente ao atributo quantidade de acabamento
valor = (float) T_Peca.getQtdAcabamento(T_Acabamento.getFatorCalculo());
    if (valor != 0)
        //verifica o tipo de acabamento para colocar a unidade de medida correta
        if (T_Acabamento.getTipo() == "tinta")
            strQtd.Format(_T("%.4f Kg"), valor);
        else
            strQtd.Format(_T("%.4f m2"), valor);
    else
        strQtd = "";
ModelDoc.AddCustomInfo3(config,"Qtd Acabamento",swCustomInfoText,strQtd);

//referente ao atributo quantidade de material
valor = (float)T_Peca.getPeso_M2 (T_Material.getPesoEspecifico() , T_Material.getTipoMat());
    if (valor != 0)
        //verifica o tipo de material para colocar a unidade correta
        if (T_Material.getTipoMat() != "chapam2")
            strQtd.Format(_T("%.4f Kg"), valor);
        else
            strQtd.Format(_T("%.4f m2"), valor);
    else
        strQtd = "";
ModelDoc.AddCustomInfo3(config,"Qtd Material",swCustomInfoText,strQtd);

ModelDoc.AddCustomInfo3 (config,"Denominação Superior",swCustomInfoText, T_Peca.getDenSuperior());
ModelDoc.AddCustomInfo3(config,"Denominação Inferior",swCustomInfoText, T_Peca.getDenInferior());
ModelDoc.AddCustomInfo3(config,"Projetista",swCustomInfoText,T_Peca.getProjetista());
ModelDoc.AddCustomInfo3(config,"Verificador",swCustomInfoText,T_Peca.getVerificador());
```

Para criação de um arquivo *template* (modelo de formato padrão), onde serão inseridos automaticamente os atributos cadastrados no modelo 3D, foi criado um formato através de linhas, dentro de um arquivo do tipo *drawing*. Para criar um link (ligação) com os atributos do modelo 3D criados a partir do protótipo, que serão inseridos neste formato, foi feita uma cópia de um link de atributos já criados automaticamente pelo SolidWorks, tais como: data de criação, item da peça e outros, e neste link apenas foi alterado o nome do atributo que ali será inserido (fig. 5-5). Por exemplo, o link “\$PRPSHEET:{SW-File Name(File Name)}” trás o nome do arquivo do SolidWorks, e já é um atributo padrão do software, então é feita uma cópia dele e alterado o nome do atributo que está entre colchetes, para o nome dos atributos que foram atribuídos ao modelo 3D.

FIGURA 5-5: EXEMPLO DE LINK NA LEGENDA DO FORMATO

\$PRPSHEET:{Projeto}	\$PRPSHEET:{Observação}	TOL: \$PRPSHEET{Tolerancia}
VERIF: \$PRPSHEET{Verificação}	TRATAM: \$PRPSHEET{Tratamento}	ESC: \$PRPSHEET{Escala}
LIE: \$PRPSHEET{Linha}	ACABAM: \$PRPSHEET{Acabamento}	\$PRPSHEET{Qty Acabamento}
\$PRPSHEET{SW-Short Date(Short Date)}	\$PRPSHEET{Material}	\$PRPSHEET{Qty Material} #c
	\$PRPSHEET{Denominação Superior}	\$PRPSHEET:{SW-File Name(File Name)}
	\$PRPSHEET{Denominação Inferior}	

Em seguida este arquivo foi salvo dentro da pasta *template* do SolidWorks com o nome de drawing A3N BCC.drwdot. Agora toda vez que o projetista usar o comando *new* do SolidWorks, será exibido um novo tipo de documento *drawing*, sendo que contém as ligações com os nomes dos atributos acrescentados nos Modelos 3D.

Estes foram os principais métodos e técnicas utilizadas para construção do protótipo, sendo que os métodos utilizados foram encontrados na documentação do SolidWorks API.

5.3.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO

O protótipo para acrescentar atributos nos sólidos gerados no software 3D SolidWorks, possui uma única interface (fig. 5-6), onde o projetista encontra as informações necessárias para selecionar os tipos de atributos necessários ao modelo 3D.

FIGURA 5-6: TELA DO PROTÓTIPO

Na parte superior esquerda, o projetista seleciona o tipo de material clicando no *radio button*, informa também o comprimento e a largura da peça desenvolvida (planificada). Neste momento o *combo box* material libera os materiais conforme está seleção, então é só selecionar *combo box* um dos materiais padrões que a empresa possui. Na parte central superior o projetista poderá escolher o tipo de tratamento padrão que a empresa possui clicando no *radio button* correspondente e está seleção já preenche o *edit box* Tratamento. Na parte superior direita, o projetista seleciona o tipo de acabamento padrão que a empresa possui, clicando no *radio button* correspondente ao acabamento desejado, também já é preenchido o *edit box* acabamento nesta seleção. Os campos dos *edit Box* tolerância e escala, já vêm preenchido com os valores mais usados pelos projetistas, neste caso os valores foram informados pela empresa. Mas se o projetista optar por algum outro valor, para material, acabamento, tratamento, escala e tolerância o mesmo é informado diretamente nos *edit box* correspondentes.

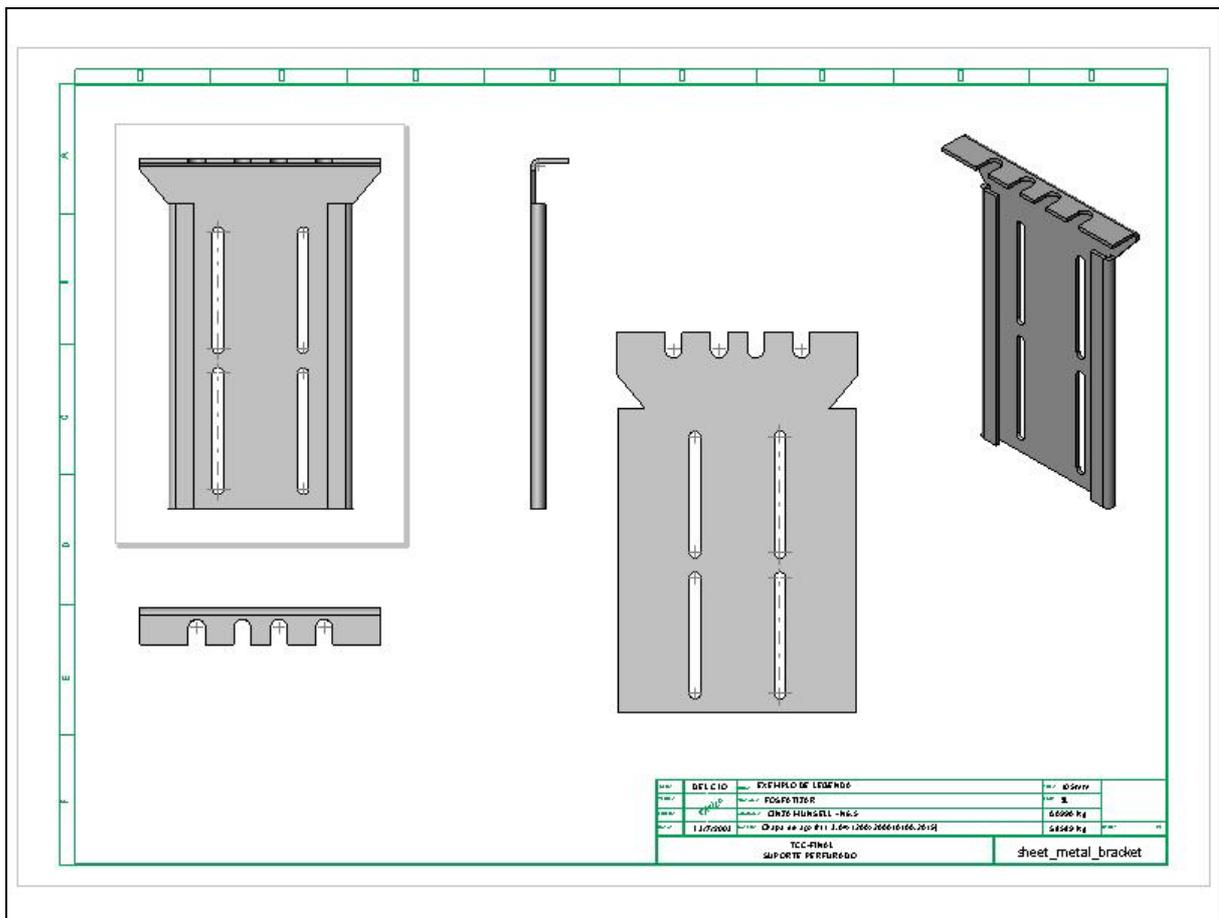
Para os *edit box*: projetista, verificador, observação, denominação superior e denominação inferior, o projetista preenche diretamente no *edit box* correspondente, informando o valor do campo.

Quando não existir material ou acabamento especial, a quantidade de material e quantidade de acabamento serão calculadas, e se o projetista optar por ver os valores nos *edit box* correspondentes, deverá então clicar no botão aplicar e o protótipo não se fecha, mostrando os valores no *edit box*, caso contrário o projetista pode clicar apenas no botão OK, e assim o protótipo se encerra, acrescentado os atributos no sólido ativo do SolidWorks. Caso o projetista não quiser completar a operação, poderá clicar no botão cancelar, e o protótipo se encerra sem cadastrar os atributos no sólido.

Vale lembrar, que os atributos serão cadastrados no modelo ativo e na configuração ativa deste modelo. O programa vai alertar com uma mensagem ao projetista se o mesmo tentar acrescentar atributos em um *drawing*, informando para que ative um *part* ou *assembly*.

A seguir na fig. 5-7 é apresentado como resultado, um desenho de formato técnico com os atributos cadastrados pelo protótipo.

FIGURA 5-7: FORMATO DESENHO TÉCNICO COM ATRIBUTOS



5.4 RESULTADOS E DISCUSSÃO

Foram executados alguns testes para verificar o funcionamento do protótipo. Nestes testes, foram criados alguns sólidos e o protótipo foi então utilizado para se fazer a inclusão de atributos aos sólidos. O funcionamento do protótipo foi compatível com o resultado esperado, gerando maior produtividade através da agilização no cadastramento de atributos dos sólidos criados no software SolidWorks.

6 CONCLUSÕES

Este trabalho apresentou o desenvolvimento de uma ferramenta customizada para inserção de atributos em sólidos no SolidWorks, utilizando a API fornecida pelo fabricante, em linguagem C++. No decorrer do desenvolvimento do trabalho, várias dificuldades surgiram, notadamente a falta de referências sobre desenvolvimento em C++ para a plataforma SolidWorks.

A escolha do tema de trabalho se mostrou muito desafiadora para o autor, pois trata-se de um trabalho pioneiro entre os demais na área de CAD na FURB. Não obstante pôde-se observar que a aplicabilidade do protótipo na prática foi comprovada, mesmo tratando-se de um protótipo que contempla ainda muito desenvolvimento futuro. A boa documentação do SolidWorks API foi essencial para que este trabalho fosse concluído, pois uma grande parte do tempo de desenvolvimento do trabalho foi gasto com testes para compreender a funcionalidade das bibliotecas do SolidWorks antes de iniciar a construção do protótipo. Esta foi a melhor forma de se adquirir experiência no desenvolvimento de aplicativos para o SolidWorks. Pode-se inclusive afirmar que o ponto chave deste trabalho foi alcançado, pois foi possível utilizar a API do SolidWorks para construir um protótipo para acrescentar atributos em sólidos 3D deste software, o que demonstra o início do futuro desenvolvimento de um sistema PDM, que será realizado no decorrer dos próximos meses.

Durante a execução deste trabalho houveram problemas na construção da base de dados para teste. Este era um dos objetivos específicos inicialmente propostos, e não foi concluído neste trabalho. Entretanto, isto não interferiu na conclusão do ponto chave deste trabalho, mesmo porque a empresa que irá utilizar este protótipo já possui um banco de dados diferente do que seria usado neste trabalho.

O protótipo foi testado durante um período de uma semana, na empresa onde se caracterizou a necessidade do mesmo. Em uma primeira análise, os resultados foram positivos, alcançando o resultado esperado e satisfazendo as primeiras expectativas dos projetistas usuários do protótipo.

Durante os testes, verificou-se a necessidade de acréscimo de novas funções ao protótipo, como por exemplo: edição de atributos que já foram cadastrados previamente. Esta funcionalidade não figurava entre os objetivos deste trabalho, mas a qual poderá ser

acrescentada ao protótipo com algumas modificações e representa também uma das sugestões para trabalhos futuros.

E finalmente vale lembrar que o conhecimento adquirido pelo autor em anos de trabalho com ferramentas CAD, foram essenciais durante a construção do protótipo. Profissionais da computação preparados para estas áreas de atuação são relativamente escassos, forçando as empresas a deslocarem profissionais da área de engenharia a obter cursos de programação, para desempenhar uma outra função essencial para empresa, que é a customização de ferramentas CAD.

6.1 LIMITAÇÕES

O protótipo utiliza a API do software SolidWorks 2003 de forma a customizá-lo para acrescentar atributos nos modelos 3D, mas o item que tratava da construção de uma base de dados para testes, que era um dos objetivos específicos, não foi concluída, ficando o protótipo limitado neste item.

Além disso, por ser destinado a um setor específico de uma empresa, o protótipo está direcionado ao trabalho daquele setor, não possibilitando utilizá-lo em outras áreas.

6.2 EXTENSÕES

Para uma possível extensão deste trabalho, sugere-se a construção de um programa que seja mais genérico quanto aos tipos de atributos contemplados, tais como outros tipos de materiais, e não ser tão específico quanto o protótipo construído neste trabalho. Também sugere-se um trabalho de pesquisa dentro das empresas para conhecer os sistemas PDM que elas possuem, pois cada vez mais o problema das empresas é relacionado ao gerenciamento dos dados de seus produtos, geralmente relacionados com os correspondentes modelos 3D.

Sugere-se também a construção de um programa para fazer modelagens geométricas paramétricas de alguns sólidos para o ambiente SolidWorks, pois a experiência do autor na área de projetos mecânicos atesta que existe a necessidade de maior agilidade em construir modelos 3D com mesmas características, onde somente mudam-se algumas dimensões para uma nova necessidade. Isto não acontece com um apenas um sólido, mas também em conjuntos, onde vários sólidos precisam ser alterados para uma nova necessidade ou requisito de modelagem.

REFERÊNCIAS BIBLIOGRÁFICAS

ASSESSOCON. **Comércio de equipamentos de informática e Softwares Ltda**, Sorocaba, [1987]. Disponível em:<<http://www.assessocon.com.br>>. Acesso em: 24 mai. 2003.

BARBOSA, Carlos Alberto Moreno. **Um modelo para a representação do aspecto evolutivo do processo de design**. 1996. 102 f. Dissertação de mestrado (Mestre em Engenharia Mecânica) – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.

CASACURTA, Alexandre. **Modelagem geométrica**, São Leopoldo, [1999]. Disponível em:<<http://www.inf.unisinos.br/~marcelow/ensino/grad/cg/modelagem/modelagem.html>>. Acesso em: 17 nov. 2002.

CIMDATA. **CIMdata**, [S.l.]. [1999?]. Disponível em:<<http://www.cimdata.com>>. Acesso em: 20 fev. 2003.

COSMOSWORKS. **Structural research & analysis corporation**, Los Angeles, [1982?]. Disponível em:<<http://www.cosmosm.com>>. Acesso em: 20 mai. 2003.

CUNHA, Raimundo Ricardo Matos da. **Estudo e desenvolvimento de metodologias na troca de dados em CAD/CAM**. 2000. 130 f. Tese de doutorado (Doutor em engenharia mecânica) – Departamento de engenharia mecânica, Universidade Federal de Santa Catarina, Florianópolis.

EDS. **Parasolid**, Texas, [1962?]. Disponível em :<<http://www.eds.com/products/plm/parasolid/index.shtml>>. Acesso em: 23 nov. 2002.

FOLEY, James D et al. **Computer graphics: principles and practice**. 2. ed. Washington:Addison-Wesley, 1990.

GIESECKE, Frederick E. et al. **Comunicação gráfica moderna**. Tradução Alexandre Kawano. Porto Alegre:Bookman, 2002.

GIBBSCAM. **Simple powerful**, Los Angeles, [1982?]. Disponível em :<<http://www.gibbscam.com>>. Acesso em: 20 mai. 2003.

KLEIN, Roberto. Conceitos básicos de CAD. In: Seminário internacional de Computação: Arquitetura e Urbanismo, 1., 1992, São Paulo. **Anais...** São Paulo: USP, 1992. p. 9-30.

PROENÇA, Adriano et al. **Manufatura integrada por computador: sistemas integrados de produção: estratégia, organização, tecnologia e recursos humanos**. 1. ed. Organizadores Luis Sergio Salles Costa; Heitor M. Caulliraux. Rio de Janeiro: Campus: Senai: COPPE/UFRJ, 1995. 450 p.

RENDERWORKS, RenderWorks, Joinville, [1998?]. Disponível em: <<http://www.renderworks.com.br>>. Acesso em: 10 ago. 2002.

SILVA, André Tavares da. **Integração de técnicas de modelagem com a VRML**, 1999. 95 f. Monografia de bacharel (Bacharel em Informática) – Centro de Ciências Exatas e Tecnológicas, Universidade do Vale do Rio dos Sinos, São Leopoldo.

SILVA, Isaac Newton Lima da. **Modeladores 3D**, Porto Alegre, [2002?]. Disponível em: <<http://www.em.pucrs.br/~isaac/aula2fab.pdf>>. Acesso em: 23 nov. 2002.

SOLIDWORKS. **SolidWorks**, Baker Avenue, [1995?]. Disponível em: <<http://www.solidworks.com>>. Acesso em: 20 ago. 2002.

SOUZA, Antônio Carlos de et al. **SolidWorks – guia prático para modelagem sólida**. Florianópolis: Visual Books Editora, 2001. 220 p.

STEMMER, Caspar E.; FERREIRA, Aureo C. **Características e programação de máquinas com comando numérico**. 1996. 63 f. Grupo de pesquisa e treinamento em comando numérico, Universidade federal de Santa Catarina, Florianópolis.

TURNELL, Maria de Fátima Queiroz Vieira. **Visão geral da automação industrial**, Campina Grande, [1999?]. Disponível em: <http://www.dee.ufpb.br/~fatima/Info_Ind/Tutoriais/Automacao/>. Acesso em: 10 out. 2002.

VOISINET, Donald D. **CADD - Projeto e desenho auxiliados por computador: introdução, conceitos e aplicações**. 1. ed. Tradução Ricardo Bertini Renzetti; Roberto Bertini Renzetti. Revisão Técnica Roberto Klein. São Paulo: McGraw-Hill, 1988. 450 p.