Campus de São Carlos

**UM SISTEMA DE AUXÍLIO AO PROJETISTA DE
PEÇAS FUNDIDAS ATRAVÉS DA INTEGRAÇÃO
ENTRE A TÉCNICA DFM E PROJETO
AUXILIADO POR COMPUTADOR**

ALEX SANDER CHAVES DA SILVA

Orientador: Prof. Assoc. Eduardo Vila Gonçalves Filho

UNIVERSIDADE DE SÃO PAULO

**ESCOLA DE ENGENHARIA
DE SÃO CARLOS**

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA MECÂNICA

**UM SISTEMA DE AUXÍLIO AO PROJETISTA DE PEÇAS FUNDIDAS
ATRAVÉS DA INTEGRAÇÃO ENTRE A TÉCNICA DFM E PROJETO
AUXILIADO POR COMPUTADOR**

Alex Sander Chaves da Silva

Tese apresentada à Escola de Engenharia de São Carlos, da
Universidade de São Paulo, como parte dos requisitos para a
obtenção do título de Doutor em Engenharia Mecânica.

DEDALUS - Acervo - EESC

31100036986

ORIENTADOR: Prof. Dr. Eduardo Vila Gonçalves Filho

Serviço de Pós-Graduação EESC/USP
EXEMPLAR REVISADO
Data de entrada no Serviço: 05/07/02
Ass.:

São Carlos

2002

✓

Class.	TESE-EESC
Cott.	5639
Tombo	T0123/02
Sysno	1451598

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

S586s Silva, Alex Sander Chaves da
Um sistema de auxílio ao projetista de peças fundidas através da integração entre a técnica DFM e projeto auxiliado por computador / Alex Sander Chaves da Silva. -- São Carlos, 2002.

Tese (Doutorado) -- Escola de Engenharia de São Carlos-Universidade de São Paulo, 2002.
Área: Engenharia Mecânica.
Orientador: Prof. Dr. Eduardo Vila Gonçalves Filho.

1. Projeto voltado para a manufatura. 2. Engenharia simultânea. 3. Projeto auxiliado por computador. I. Título.

FOLHA DE JULGAMENTO

Candidato: Engenheiro **ALEX SANDER CHAVES DA SILVA**

Tese defendida e julgada em 14-05-2002 perante a Comissão Julgadora:

Prof. Assoc. **EDUARDO VILA GONÇALVES FILHO** (Orientador)
(Escola de Engenharia de São Carlos/USP)

Aprovado

Prof. Dr. **JONAS DE CARVALHO**
(Escola de Engenharia de São Carlos/USP)

APROVADO

Prof. Dr. **OVIDIO RICHARD CRNKOVIC**
(Escola de Engenharia de São Carlos/USP)

Aprovado

Prof. Dr. **JOSE CARLOS DE TOLEDO**
(Universidade Federal de São Carlos/UFSCar)

Aprovado

Prof. Dr. **DURVAL UCHOAS BRAGA**
(Fundação de Ensino Superior de São João Del Rei/FUNREI)

Aprovado.

Prof. Associado **ARTHUR JOSÉ VIEIRA PORTO**
Coordenador do Programa de Pós-Graduação
em Engenharia Mecânica

Profa. Assoc. **MARIA DO CARMO CALIJURI**
Presidente da Comissão de Pós-Graduação

*Há Homens que lutam um dia e são bons;
Há Homens que lutam muitos dias e são melhores;
Há Homens que lutam muitos anos e são excelentes;
Há Homens que lutam a vida inteira, estes são os imprescindíveis.*

Aos meus pais Ligzamon e Sirene, à Vânia e, em especial, aos meus filhos Alex Gabriel e Alex Rafael que souberam suportar a minha ausência em momentos tão importantes de suas vidas.

AGRADECIMENTOS

Ao Prof. Dr. Eduardo Vila Gonçalves Filho, pela orientação e pelos conhecimentos passados durante o desenvolvimento deste trabalho;

Ao Prof. Dr. Ovídio Richard Crnkovic, pela amizade, pelo apoio na parte teórica e prática em fundição;

Ao Prof. Dr Jonas de Carvalho, pela orientação no início deste trabalho;

À Vânia Regina Velloso Silva que sempre me apoiou para que eu completasse mais esta etapa importante de minha vida.

Aos meus pais pelo incentivo.

Ao colega Nilo Rodrigues pela sua importante contribuição no desenvolvimento deste trabalho.

Aos colegas Volnei e Neilor pela força e suporte.

Às secretárias do departamento Ana Paula e Beth pelo apoio e amizade.

Ao pessoal da secretaria, Maragno, Cristina, Margareth, Vilma pelo apoio e confiança.

À todos os professores do Departamento de Engenharia Mecânica, pelas orientações passadas nas aulas, em especial aos professores: Vila e Jonas.

Ao CNPQ, pela bolsa concedida para a conclusão desta tese.

E principalmente a DEUS que me iluminou e me deu forças para chegar ao fim de mais uma jornada feliz.

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO	1
1.1 GENERALIDADES	1
1.2 JUSTIFICATIVA E DEFINIÇÃO DO PROBLEMA	2
1.3 OBJETIVO DA PESQUISA	4
1.4 CONTEÚDO DA TESE	4
CAPÍTULO – 2 - ALGUMAS TÉCNICAS EMPREGADAS NA MELHORIA DO PROJETO E PROCESSO DE FABRICAÇÃO DE UM PRODUTO	5
2.1 ENGENHARIA SIMULTÂNEA	5
2.2 PROJETO VOLTADO PARA A MANUFATURA (DFM)	7
2.3 PROJETO VOLTADO PARA A MONTAGEM (DFA)	9
2.4 PROJETO VOLTADO PARA PARA A FUNDIÇÃO (DFC)	13
2.5 ALGUMAS FERRAMENTAS COMPUTACIONAIS PARA AUXILIAR O PROJETISTA NO DESENVOLVIMENTO DO PROJETO DO PRODUTO	19
2.5.1 USO DO CAD	20
2.5.2 O USO DO CAE	25
2.5.3 SOFTWARE PARA SIMULAÇÃO DO PROCESSO DE FUNDIÇÃO	26
2.6 FEATURES	27
2.7 BASE DE CONHECIMENTO OU BASE DE DADOS	32
CAPÍTULO – 3 - PROJETO DE PEÇAS FUNDIDAS VISANDO UMA FABRICAÇÃO ECONÔMICA	36
3.1 PROCESSO DE FUNDIÇÃO EM AREIA	36

3.2 ORIENTAÇÕES DE PROJETO E DE PROCESSO PARA O PROCESSO DE FUNDIÇÃO EM AREIA	49
3.2.1 <i>CONSIDERAÇÕES SOBRE A GEOMETRIA DA PEÇA</i>	49
3.2.2 <i>CONSIDERAÇÕES SOBRE O PROJETO DO FUNDIDO</i>	53
3.3 GUIA DE AUXÍLIO AO PROJETISTA NO PROJETO DE PEÇAS FUNDIDAS EM AREIA	64
 CAPÍTULO 4 - SISTEMA DE AUXÍLIO AO PROJETISTA NO PROJETO DE PEÇAS FUNDIDAS (SEMF)	 70
4.1 VISÃO GERAL DO SISTEMA	70
4.2 DETALHAMENTO DA CONCEPÇÃO DO SISTEMA	72
4.2.1 <i>DESENHAR PEÇA</i>	73
4.2.2 <i>DETERMINAÇÃO DO TRAIL ATUAL</i>	73
4.2.3 <i>LEITURA DO TRAIL E RECONHECIMENTO DE FEATURES</i>	74
4.2.4 <i>ÁRVORE PRINCIPAL DE DADOS (APD) E LISTA DE REFERÊNCIA (LR)</i>	76
4.2.5 <i>BASE DE CONHECIMENTO DO SEMF (REGRAS)</i>	81
4.2.6 <i>VERIFICAR FEATURE</i>	86
 CAPÍTULO 5 – SISTEMA IMPLANTADO E EXEMPLO DE APLICAÇÃO	 91
5.1 SISTEMA DE AUXÍLIO AO PROJETISTA DE PEÇAS FUNDIDAS – SEMF	91
5.2 EXEMPLO DE APLICAÇÃO	99
 CAPÍTULO 6 – CONCLUSÕES E PROPOSTAS PARA TRABALHOS FUTUROS	 135
 REFERÊNCIAS BIBLIOGRÁFICAS	 138
 APÊNDICE 1- IMPLEMENTAÇÃO DO PROGRAMA SEMF	
 APÊNDICE 2 - TABELA COM EXEMPLO DE UM TRAIL	

LISTA DE FIGURAS

Figura 3.1 - Processo de fundição em areia.....	36
Figura 3.2 - Forma em L (mesma espessura) para o processo de fundição em areia.....	50
Figura 3.3 - Uniformidade da espessura na seção.....	51
Figura 3.4 - Raios de borda recomendados para reentrâncias e ressaltos.....	52
Figura 3.5 - Raios de extremidade recomendado para nervuras.....	52
Figura 3.6 - Seções adjacentes.....	53
Figura 3.7 - Modos de evitar machos.....	59
Figura 3.8 - Orientações para furo.....	60
Figura 3.9 - Ângulo de saída.....	61
Figura 3.10 - Concordância em Prolongamento.....	62
Figura 3.11 - Espessura Mínima/ Máxima.....	63
Figura 3.12 - Acesso às orientações sobre a geometria da peça.....	66
Figura 3.13 - Acesso às orientações sobre o fundido.....	67
Figura 4.1 - Fluxograma do SEMF.....	72
Figura 4.2 - Algoritmo para localizar último <i>trail</i>	74
Figura 4.3 - Exemplo de informação: sólido com um furo e um chanfro.....	75
Figura 4.4 - Exemplo de como as informações de um projeto estão dispostas na APD/LR.....	77
Figura 4.5 - Unidade de informação (UI) da APD na forma de diagrama de blocos.....	78
Figura 4.6 - Esquema do nó na APD.....	80
Figura 4.7 - Desenho feito no PRO-ENGINEER para um exemplo de checagem no SEMF.....	87
Figura 4.8 - Como encontra as espessuras de parede através da cotas e <i>datum plane</i>	87
Figura 4.9 - Determinação da espessura de parede.....	88
Figura 4.10 - Esquema mostrando as dimensões para cálculo da regra espessura mínima de parede.....	89
Figura 5.1 - Área de trabalho do SEMF.....	91
Figura 5.2 - Botão “carregar” da área de trabalho do SEMF.....	92
Figura 5.3 - Botão “carregar” com a opção “escolher” selecionada.....	92

Figura 5.4 - Caixa de diálogo onde o usuário procura um <i>trail</i> específico.....	93
Figura 5.5 - Janela do guia.....	93
Figura 5.6 - Parte da área de trabalho do SEMF que contém os menus principais. O usuário clicou em “ARQUIVO”.....	94
Figura 5.7 - Área de trabalho do SEMF, quando se clica no botão “Arquivo” e seleciona a opção textos/regras.....	95
Figura 5.8 - Editor de texto para as regras.....	96
Figura 5.9 - Editor de preferência do SEMF.....	96
Figura 5.10 - Menu “ <i>pull down</i> ” que aparece quando se clica em “CAD/CAE”.....	97
Figura 5.11 - Área de trabalho do SEMF com um texto de não violações nos resultados de checagem.....	98
Figura 5.12 - Botão “cancela” e botão “todos” da área de trabalho do SEMF.....	98
Figura 5.13 - Local onde o usuário escolhe entre ver violações ou recomendações.....	99
Figura 5.14 - Peça a ser construída e checada.....	99
Figura 5.15 - Área de trabalho do PRO-ENGINEER.....	99
Figura 5.16 - Parte da área de trabalho do PRO-ENGINEER e seleção da opção “ <i>new</i> ”.....	100
Figura 5.17 - Janela onde o usuário pode escolher o tipo e o subtipo de desenho.....	100
Figura 5.18 - Janela do PRO-ENGINEER após o início de um desenho.....	101
Figura 5.19 - Opção <i>Feature</i> selecionada no <i>menu manager</i>	101
Figura 5.20 - Parte do <i>menu manager</i> mostrando a opção “ <i>Create</i> ”.....	102
Figura 5.21 - Parte do <i>menu manager</i> mostrando a opção “ <i>Datum</i> ” selecionada.....	102
Figura 5.22 - Parte do <i>menu manager</i> mostrando as opções do menu “ <i>Datum</i> ”.....	103
Figura 5.23 - Menu mostrando as opções de planos.....	103
Figura 5.24 - Os 3 planos ortogonais chamados de “ <i>Datum Planes</i> ”.....	104
Figura 5.25 - Menu <i>manager</i> mostrando a opção “ <i>Feat_Create</i> ”.....	104
Figura 5.26 - Menu <i>manager</i> mostrando a opção “ <i>Solid</i> ”.....	105
Figura 5.27 - Menu <i>manager</i> mostrando a opção “ <i>Protrusion</i> ”.....	105
Figura 5.28 - Menu <i>manager</i> mostrando as opções “ <i>Extrude</i> , <i>Solid</i> e <i>Done</i> ”.....	106
Figura 5.29 - Menu <i>manager</i> mostrando opções de desenho sólido.....	106
Figura 5.30 - Menu onde o PRO-ENGINEER pede que o usuário escolha o plano de desenho.....	107

Figura 5.31 - A seta no <i>datum plane</i> indica o sentido positivo de elevação do desenho.....	107
Figura 5.32 - Janela do menu <i>manager</i> mostrando a opção “ <i>Flip</i> ” que modifica o sentido do desenho.....	108
Figura 5.33 - Menu <i>manager</i> mostrando as opções de referência para o desenho.....	108
Figura 5.34 - Parte da área de trabalho do PRO-ENGINEER mostrando os 3 planos ortogonais.....	109
Figura 5.35 - Linha pontilhada criada depois que o usuário selecionou o plano horizontal.....	110
Figura 5.36 - Opção <i>Done Sel</i> selecionada no menu <i>manager</i>	111
Figura 5.37 - Janela do menu <i>manager</i> onde o usuário pode escolher o tipo de rascunho que vai fazer. Para o exemplo em particular o usuário deve clicar em <i>rectangle</i>	111
Figura 5.38 - Retângulo rascunhado pelo usuário.....	112
Figura 5.39 - Parte do menu <i>manager</i> mostrando a opção <i>modify</i>	112
Figura 5.40 - Área de trabalho do PRO-ENGINEER. A seta mostra a caixa de texto que o usuário deve utilizar para digitar o novo valor da cota.....	113
Figura 5.41 - Rascunho do retângulo. A cota horizontal de referência está em destaque..	114
Figura 5.42 - Rascunho do retângulo. A cota vertical de referência está em destaque.....	114
Figura 5.43 - Rascunho do retângulo. A cota horizontal está em destaque.....	115
Figura 5.44 - Rascunho do retângulo. A cota vertical está em destaque.....	116
Figura 5.45 - Parte do menu <i>manager</i> mostrando o comando “ <i>Blind</i> ”.....	117
Figura 5.46 - A área de trabalho do PRO-ENGINEER. A seta mostra o campo onde o usuário deve digitar a cota da altura.....	117
Figura 5.47 - Mostra a janela onde o usuário precisa clicar em OK para finalizar a construção do sólido.....	118
Figura 5.48 - Área de trabalho do PRO-ENGINEER com o sólido pronto.....	118
Figura 5.49 - Área de trabalho do SEMF.....	119
Figura 5.50 - SEMF exibindo a descrição do prisma sólido do desenho.....	120
Figura 5.51 - Resultado da checagem do prisma sólido sem violação.....	120
Figura 5.52 - Comando <i>hole</i> selecionado no menu <i>manager</i>	121

Figura 5.53 - Menu <i>manager</i> com as opções de furo.....	121
Figura 5.54 - Área de trabalho do PRO-ENGINEER durante a criação de um furo.....	122
Figura 5.55 - Caixa de texto onde o usuário deve digitar o valor do diâmetro.....	123
Figura 5.56 - Janela mostrando botão OK que conclui a construção do furo.....	123
Figura 5.57 - Sólido + furo.....	124
Figura 5.58 - Parte da área de trabalho do SEMF mostrando o reconhecimento do furo cilíndrico.....	124
Figura 5.59 - Sólido com 5 furos.....	125
Figura 5.60 - Janela do SEMF mostrando descrição das <i>features</i>	125
Figura 5.61 - Parte do menu <i>manager</i> com o comando <i>create</i> destacado.....	126
Figura 5.62 - Parte do menu <i>manager</i> com as opções de sólido.....	126
Figura 5.63 - Rascunho do <i>cut</i>	127
Figura 5.64 - Janela que pergunta se o usuário deseja alinhar determinada <i>feature</i> com outra.....	128
Figura 5.65 - Peça com o corte.....	128
Figura 5.66 - Saída do SEMF para a peça desenhada na figura 5.65.....	128
Figura 5.67 - A seta azul mostra onde o usuário deve clicar.....	129
Figura 5.68 - Cotas do cilindro.....	130
Figura 5.69 - Peça de demonstração com o cilindro terminado.....	130
Figura 5.70 - Parte da tela do SEMF mostrando o reconhecimento do cilindro sólido.....	131
Figura 5.71 - Janela do PRO-ENGINEER mostrando a peça pronta.....	131
Figura 5.72 - Parte da janela do SEMF mostrando todas as <i>features</i> da peça desenhada na figura 5.71.....	132
Figura 5.73 - Saída do SEMF, quando o usuário aciona a regra (espessura mínima de parede).....	132

LISTA DE TABELAS

Tabela 2.1 - Ferramentas disponíveis para projetos com metodologia DFMA (<i>Design For Manufacture e Assembly</i>).....	10
Tabela 2.2 - Exemplo de parte de um arquivo <i>trail</i>	24
Tabela 2.3 - Ferramentas para o desenvolvimento integrado do produto.....	28
Tabela 2.4 - Distinção entre uma base de dados de uma base de conhecimento.....	32
Tabela 3.1- Ângulo de saída em função do material do modelo x método de socagem....	54
Tabela 3.2 - Tipo de Metal x Contração Linear.....	56
Tabela 3.3 - Orientações para usinagem de acabamento no modelo.....	57
Tabela 3.4 - Limitações de machos.....	59
Tabela 4.1 - Procedimento de reconhecimento das <i>features</i>	76
Tabela 4.2 - Tabela de decisão.....	82
Tabela 4.3 - Opções de verificação de <i>features</i> no SEMF.....	86

LISTA DE ABREVIATURAS E SIGLAS

APD	- Árvore Principal de Dados
CAD	- Computer Aided Design (Projeto Auxiliado por Computador)
CAE	- Computer Aided Engineering (Engenharia Auxiliada por Computador)
CAM	- Computer Aided Manufacturing (Manufatura Auxiliada por Computador)
CAPP	- Computer Aided Process Planning (Planejamento do Processo Auxiliado por Computador)
CAQC	- Computer Aided Quality Control (Controle de Qualidade Auxiliado por Computador)
CNC	- Comando Numérico Computadorizado
DFA	- Design For Assembly (Projeto Voltado a Montagem)
DFC	- Design For Casting (Projeto Voltado para o Fundido)
DFM	- Design For Manufacture (Projeto Voltado a Manufatura)
DFMA	- Design For Manufacture And Assembly (Projeto Voltado a Manufatura e Montagem)
ES	- Engenharia Simultânea
FEM/MEF	- Finite Element Method (Método de Elementos Finitos)
FMEA	- Failure Modes And Effects Analysis (Análise dos Modos de Falha e seus Efeitos)
LOM	- Laminated Object Modelling (Modelagem por Laminação)
LR	- Lista de Referência
PRO - E	- PRO- ENGINEER
RP	- Rapid Prototyping (Prototipagem Rápida)
SEMF	- Sistema de Auxílio ao Projetista no Projeto de Peça Fundidas
SLS	- Selective Laser Sintering (Sinterização Seletiva a Laser)
UI	- Unidade de Informação

RESUMO

SILVA, A. S. C. (2002). *Um Sistema de Auxílio ao Projetista de Peças Fundidas através da Integração entre a Técnica DFM e Projeto Auxiliado por Computador*. São Carlos, 2002. 146 p. Tese (Doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo.

Um sistema composto por orientações de projeto e de processo, por um software CAD, mais especificamente, pelo PRO-ENGINEER (PRO-E) e um sistema de auxílio ao voltado para a manufatura (SEMF) foi desenvolvido como forma de auxílio ao engenheiro projetista no desenvolvimento do projeto de peças fundidas em areia. Este sistema integrado é capaz de analisar um desenho feito em sistema CAD (PRO-E), de um modelo de uma peça a ser fundida e fornecer um diagnóstico sobre possíveis violações quanto à sua manufatura. Isto é feito através de uma base de conhecimento que reúne informações de como minimizar possíveis defeitos com relação à geometria da peça e ao processo de fundição. O desenho deve seguir um procedimento adequado para que as *features* criadas no PRO-E possam ser reconhecidas pelo sistema, para um possível diagnóstico de problemas existentes na geometria da peça e/ou no processo de fundição.

Palavras-chaves: Projeto Voltado para a Manufatura; Engenharia Simultânea; Projeto Auxiliado por Computador.

ABSTRACT

SILVA, A. S. C. (2002). *Um Sistema de Auxílio ao Projetista de Peças Fundidas através da Integração entre a Técnica DFM e Projeto Auxiliado por Computador*. São Carlos, 2002. 146 p. Tese (Doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo.

A computer system based on design and process guidelines and a CAD system has been developed in order to aid an engineer designer to develop cast parts free from defects. The proposed system can analyze and detect, automatically, problems that may happen during the casting process of a part. This is done through a built-in Knowledge Base that gather information on how to avoid defects related to part geometry and casting process. The parts must be designed using the CAD system, PRO-E, and the drawing of the part, must be done following a certain procedure, otherwise the system will not be able to check the part.

Keywords: Design for Manufacturing, Concurrent Engineering, Computer Aided Design.

CAPÍTULO 1 - INTRODUÇÃO

1.1 GENERALIDADES

Devido à competição em um mercado globalizado, a indústria necessita de métodos e meios para que seus produtos apresentem custo final mais baixo. Por isso a manufatura e a montagem têm merecido atenção especial no projeto do produto [Kusiak, 1997].

Com o decréscimo do tempo de ciclo de vida do produto, o projeto do mesmo passou a ter influência maior no seu custo final. Pesquisas mostram que 70 a 80 % do custo final do produto é determinado durante a fase de projeto [Kusiak, 1993]. Por isso, ao projetar um produto é necessário que o projetista tenha um conhecimento multidisciplinar em áreas diversas tais como: materiais, dimensionamento, fabricação, montagem, controle da qualidade, confiabilidade, mercadologia, manutenção, dentre outras, para que ele possa tomar as decisões corretas com relação ao projeto do produto. Dificilmente, o projetista detém um conhecimento profundo em cada uma destas áreas.

Atualmente, através do uso do conceito de Engenharia Simultânea, onde um grupo de especialistas das diferentes áreas formam uma equipe responsável pelo projeto completo do produto, todo o conhecimento necessário pode ser reunido. Esta equipe, capaz de aplicar conhecimentos específicos e necessários no desenvolvimento do produto, pode evitar erros que poderiam ser cometidos no processo tradicional de projeto de produtos.

O processo de fundição de peças é um dos processos mais antigos utilizados na fabricação de produtos (máquinas agrícolas, máquinas ferramentas, etc.).

Existem muitas informações sobre este processo, porém estas informações estão espalhadas. Um importante passo seria reunir todas as informações necessárias para auxiliar o projetista nas recomendações de projeto pertinentes ao processo de fundição,

de modo a desenvolver o projeto de uma peça, livre de defeitos, com a máxima qualidade e com menor custo.

Esta reunião de informações relativas ao processo e a geometria do produto permite a formação de uma base de conhecimento que pode ser de grande auxílio ao projetista.

Por outro lado, o projeto auxiliado por computador (CAD) tem sido extensivamente utilizado no projeto do produto. O CAD é de fundamental importância no desenvolvimento de produtos da manufatura moderna, a qual exige que o projeto seja ágil e interaja rápida e flexivelmente com as demais atividades de manufatura para que o produto final tenha qualidade ótima e atinja o mercado no menor tempo possível. A grande vantagem da aplicação de sistemas CAD é a economia de tempo do projetista em atividades repetitivas, reservando-o para atividades criativas e que agreguem maior valor ao produto. Um sistema CAD que é muito utilizado é o PRO-ENGINEER (PRO-E), porque utiliza o conceito de *feature* e possui vários módulos que permitem o desenho e a simulação, como por exemplo, da abertura da matriz de uma peça fundida no processo de fundição sob pressão. O uso do conceito de *feature* facilita a integração com outros softwares para um possível reconhecimento de problemas encontrados nas *features* na fase de projeto.

1.2 JUSTIFICATIVA E DEFINIÇÃO DO PROBLEMA

No processo tradicional de desenvolvimento de produtos, o produto é concebido, os componentes são desenhados e, em seguida, são desenvolvidos os planos de fabricação desses componentes. Estas fases são seqüenciais e a integração entre elas é geralmente pequena. Uma consequência desse tipo de procedimento é que eventuais erros de projeto muitas vezes só são descobertos quando o produto já saiu do departamento de projetos. A correção dos erros sempre acarreta prejuízos à empresa e eles aumentam à medida em que a sua detecção se dá mais longe da fase inicial de projeto. Com o emprego da técnica de Engenharia Simultânea (ES) possíveis erros são minimizados.

Apesar das evidentes vantagens da Engenharia Simultânea, algumas dificuldades podem ser encontradas:

- Uma delas é que nem sempre os especialistas estão disponíveis quando a sua presença e orientação se faz necessária, o que pode acarretar atraso no desenvolvimento do produto.
- Uma outra dificuldade, esta mais específica de empresas de pequeno porte, é que o equipe de especialistas nem sempre pode ser formado porque, às vezes, nem mesmo existe um número significativo de pessoas especialistas na empresa necessárias para a formação de uma equipe de projeto.
- Um dos conhecimentos fundamentais que o engenheiro projetista necessita ter é o conhecimento de processos de fabricação, além de uma visão ampla sobre a interdependência entre materiais e processos na hora de projetar e escolher os mais adequados. Para obter informações sobre processos de fabricação, muitas vezes o projetista tem de recorrer a várias fontes, como por exemplo especialista, bancos de dados, livros e etc.
- Problemas da integração, seja entre pessoas ou entre software e hardware.

Isto tem levado à necessidade de realizar pesquisas na área de fabricação e montagem, com o objetivo de se desenvolver sistemas de auxílio ao projetista de modo que o produto possa ser projetado com mínima quantidade de problemas.

Algumas justificativas para o desenvolvimento do trabalho são:

- 1) Dificuldade de implementação da ES;
- 2) Problema da integração em geral (pessoas);
- 3) Dificuldade de integração entre softwares;
- 4) Necessidade de competição no mercado (*time-to-market*);
- 5) Melhorar a qualidade do projeto e reduzir custos;
- 6) O PRO-E não fornece nenhum auxílio ao projetista, caso de peças que serão fabricadas através do processo de fundição em areia.

1.3 OBJETIVO DA PESQUISA

O propósito deste trabalho de pesquisa é desenvolver um programa computacional que seja capaz de detectar possíveis falhas no projeto de uma peça fundida pelo processo de fundição em areia ainda na fase de desenho. O auxílio ao Engenheiro Projetista é obtido através das recomendações de projeto e de manufatura para uma peça fundida.

O programa deve ser capaz de analisar uma peça, desenhada em um sistema CAD (PRO-E), e baseado em um conjunto de regras, específicas para o processo de fundição em areia, detectar possíveis violações quanto a aspectos relacionados com a geometria da peça e a manufatura do fundido .

1.4 CONTEÚDO DA TESE

Esta tese foi desenvolvida em seis capítulos. No primeiro capítulo apresenta-se a necessidade desta pesquisa, sua justificativa, e também o conteúdo dos demais capítulos.

O segundo capítulo mostra uma revisão bibliográfica sobre as técnicas empregadas na melhoria do projeto e processo de fabricação de um produto .

No terceiro capítulo são apresentadas as orientações para a geometria da peça e para o fundido. Estas orientações visam uma fabricação fácil, precisa e econômica.

No quarto capítulo é apresentado a metodologia de implantação do sistema que facilitará o acesso às orientações de projeto desejada.

No quinto capítulo são apresentados o sistema e um exemplo de aplicação.

No sexto capítulo são apresentadas a conclusão do trabalho, bem como algumas propostas para trabalhos futuros.

CAPÍTULO – 2 - ALGUMAS TÉCNICAS EMPREGADAS NA MELHORIA DO PROJETO E PROCESSO DE FABRICAÇÃO DE UM PRODUTO

Muitos pesquisadores têm contribuído para uma melhor aplicação dos conhecimentos de processos de fabricação no desenvolvimento de produtos visando um projeto otimizado através de regras, considerações gerais, guias de projeto, programas de simulação, etc. Este capítulo mostra o que tem sido desenvolvido nos últimos anos com relação à Engenharia Simultânea e suas ferramentas como DFM (*Design For Manufacture*), o DFA (*Design For Assembly*) e programas computacionais como forma de auxiliar o projetista no desenvolvimento do projeto do produto.

2.1 ENGENHARIA SIMULTÂNEA

A Engenharia Simultânea (ES) é a maneira de conduzir a atividade de projeto para que as várias atividades relacionadas à progressão do projeto sejam integradas e realizadas sempre que possível, em paralelo ao invés de seqüencialmente. Mais especificamente, “Engenharia Simultânea é a consideração, durante a fase de projeto, dos fatores associados ao ciclo de vida do produto. Eles incluem fabricação, montagem, teste, manutenção, custo e qualidade” (O’ Grady & Young citado por Naveiro, 1997). A chave para a implantação da Engenharia Simultânea é alcançar o mais cedo possível a integração do conhecimento prático da empresa na atividade de projetar um produto. A Engenharia Simultânea é universalmente aceita como pertencente ao processo de desenvolvimento de produtos, e seu uso pode proporcionar economias ao se evitar os custos adicionais decorrente de mudanças de engenharia que aparecem a jusante da progressão do projeto (Naveiro, 1995). Existem duas abordagens básicas para implementar a prática da Engenharia Simultânea: equipe de projeto e sistemas

computacionais (Parsaei & Sullivan citado por Naveiro, 1997). Na realidade, essas duas abordagens não são incompatíveis; ao contrário, o desempenho do grupo de projeto é melhorado pelo auxílio computacional, possibilitando aos projetistas considerarem vários aspectos do ciclo de vida do produto durante o desenvolvimento do projeto. A abordagem do time de projeto consiste em juntar um grupo de especialistas de diversos setores que contribuem para realização do projeto do produto de forma a viabilizar a criação coletiva.

A principal característica adotada nessa abordagem é a superposição de tarefas, executando ao mesmo tempo atividades que antes seriam realizadas de modo seqüencial. A implementação desse princípio se verifica pela liberação de documentação parcial do projeto para os membros da equipe que trabalham a jusante, de forma a permitir que eles façam avaliações preliminares do projeto desde as etapas iniciais. Segundo Whitney, [1994] esse procedimento de superposição de etapas é largamente utilizado na indústria automobilística japonesa e sua implementação se dá de uma maneira formal, sendo que cada participante da equipe tem consciência do conteúdo de informação a ser liberado antecipadamente. De fato, a identificação das tarefas, bem como sua identificação como constituintes do todo define as fronteiras entre as tarefas e, em consequência, a relação entre os participantes.

A informação parcial liberada faz parte do discurso do projeto e é definida pela reconciliação entre os universos de especialização dos participantes e pela concordância na denominação das peças do produto e na definição de suas funções (Lutters, 2000).

Young (2000), apresenta o projeto do produto num ambiente de ES. Este ambiente interdisciplinar utiliza procedimentos, regras, planos, métodos de análises (montagem), seleção e otimiza o projeto do produto.

Existem dois princípios básicos que norteiam a implementação da ES que são:

- 1) no processo de desenvolvimento do produto as atividades são desenvolvidas rapidamente e são rigorosamente verificadas;
- 2) durante o projeto indivíduos que desempenham diferentes funções na empresa estão presentes várias funções são executadas simultaneamente.

A prática da Engenharia Simultânea possibilita a detecção precoce de problemas durante a evolução do projeto.

Segundo Gonçalves Filho (1994), o conceito de Engenharia Simultânea está ligado ao trabalho em times de especialistas que contribuem, com o projetista, durante o desenvolvimento do projeto do produto, com informações de processos, montagem, inspeção, etc., projetando assim o produto com o menor número de problemas.

É importante usar o conceito de Engenharia Simultânea porque 70% do custo do produto é determinado durante a fase de projeto. Isto significa que decisões adequadas e corretas durante a fase de projeto levam a um produto fácil de fabricar e com menor tempo entre o projeto e a fabricação.

Dessa forma, a Engenharia Simultânea pode ser vista como parte da filosofia de negócios de uma empresa, objetivando o aumento da qualidade, diminuição de custos do projeto e do tempo de desenvolvimento de um produto.

Os benefícios do uso da Engenharia Simultânea, por sua vez, são muito gratificantes para a empresa. Obtém-se sensíveis melhorias do setor de desenvolvimento de produtos, uma vez que as atividades de trabalho em grupos proporcionam melhorias na comunicação interdepartamental, diminuindo a falta de informação na geração de um novo produto.

Assim, a Engenharia Simultânea mostra-se como uma metodologia muito interessante do ponto de vista da empresa, pois supera barreiras, faz a união entre pessoas e gera produtos cada vez melhores e mais competitivos [Chiusoli, 1996].

2.2 PROJETO VOLTADO PARA A MANUFATURA (DFM)

Segundo Bralla (1996), o DFM é definido como uma técnica baseada no conhecimento que enfoca uma série de orientações, princípios, recomendações ou regras para o projeto de um produto de forma a facilitar sua fabricação.

A técnica de DFM, visa ainda a redução dos custos de manufatura e aumento da qualidade do produto. Esta redução de custo é obtida na fase de projeto. Algumas formas de reduzir custos na fase de projeto são: simplificação das operações de usinagem,

reduzir o número de machos no processo de fundição em areia e prever uma folga suficiente para saída da ferramenta (usinagem), etc.

Segundo Meerkamn (2001), o enorme potencial para economia de custos e de tempo obtido pelo reconhecimento prematuro das diversas fases do ciclo de criação e de vida do produto, permitiu estabelecer uma multiplicidade de “estados de concordâncias” (projeto voltado para a manufatura e montagem). A totalidade destas “concordâncias” é resumida sob o conceito de *Design for X* (abreviado como DFX) e o X, no caso, representa uma reserva de lugar para uma fase do ciclo (como a manufatura) ou uma propriedade específica (como os custos).

Quanto mais cedo forem aplicados os conceitos do projeto voltado para a manufatura, melhores serão os resultados finais do projeto, ou seja, um menor número de problemas aparecerá no processo de manufatura.

As técnicas DFM tornaram-se uma poderosa ferramenta para atender às necessidades acima citadas, permitindo que as empresas otimizem o projeto de seu produto.

A tabela 2.1 apresenta algumas das técnicas que podem ser aplicadas, juntamente com as áreas onde elas podem mais contribuir em um projeto de um produto.

Lau (2000), diz que o primeiro passo no estudo do DFM é estimar os custos de manufatura. Esforços devem ser feitos no sentido de reduzir custos de componentes, custos de montagem e custos proveniente do aspecto dimensional inadequado. Uma das formas de reduzir os custos provenientes do aspecto dimensional é o uso do CAD (PRO-E) em um ambiente parametrizável (*features*) ainda na fase inicial de projeto.

Ferreira (2001), comenta que o DFM é entendido como o método de compreensão das interações do projeto do produto com o sistema de fabricação, no intuito de aprimorar a qualidade e de identificar alternativas que facilitem seus processos de fabricação.

2.3 PROJETO VOLTADO PARA A MONTAGEM (DFA)

O DFA é uma ferramenta muito importante na escolha do melhor projeto do produto. Boothroyd e Dewhurst (1983), deram uma importante contribuição para o desenvolvimento da técnica DFA. O sistema desenvolvido por eles permite calcular a eficiência de projeto de modo a permitir uma comparação qualitativa de dois projetos alternativos de um mesmo produto. Segundo eles, dois passos devem ser seguidos para aplicar esta técnica:

Passo 1: verificar se a peça é uma candidata à eliminação, ou seja se há possibilidade de combinação com outras peças na montagem;

Passo 2: verificar se é possível estimar o tempo para agarrar, manipular e inserir a peça;

Uma vez de posse destes dados é possível obter o tempo total de montagem e comparar com o tempo ideal de montagem de um bom projeto. Esta técnica serve também como auxílio ao engenheiro de projetos em casos de reprojeto.

O emprego da técnica do DFA força o projetista a examinar o projeto do produto com relação à eficiência de montagem.

Segundo Redford (1994), os princípios do Design for Assembly (DFA), como qualquer metodologia de ciclo de vida, ajudam a implementar os avanços tecnológicos no projeto de produtos competitivos. Estes mesmos princípios, podem ser usados como parâmetros para avaliar produtos, processos e serviços, tanto da pesquisa de mercado (*benchmarking*) quanto na escolha de alternativas de solução. Entretanto, os métodos de avaliação da montagem desenvolvidos são mais empregados durante o projeto preliminar, ao término do projeto conceitual. Isto quando são logo empregados: tradicionalmente, o projeto só é avaliado e conferido cuidadosamente ao fim do projeto detalhado.

Uma montagem deve ser usada somente quando resultados e custos desejáveis puderem ser melhor alcançados com um agrupamento de partes do que com uma parte individual mais complexa. Considerações de projeto relativas à montagem total devem enfatizar primeiro a simplificação do produto e projeto, as restrições humanas e mecânicas, e a sequência de montagem (Bralla, 1986).

Os princípios de DFA estão relacionados à minimização do custo da montagem dentro das restrições impostas pela necessidade de atender encaixe, forma e função da montagem. Estas restrições podem ser, por exemplo, humanas, mecânicas ou referentes à sequência de montagem e desmontagem.

Tabela 2.1 - Ferramentas disponíveis para projetos com metodologia DFMA (*Design For Manufacture e Assembly*) (Gonçalves Filho, 2001)

<i>Atividades de DFMA</i>				
<i>Ferramentas DFMA</i>	Conceito de otimização	Simplicidade	Esclarecer a conformidade do processo	Função de otimização da produção.
Axiomas de projeto	X	X	X	X
Guilines (DFM)	X	X		
Projeto para Montagem (DFA)		X		
Método de avaliação da montagem		X		
Avaliação das mínimas peças	X	X		
Método Taguchi	X			X
Avaliação da robustez	X	X	X	X
Regras de DFM		X	X	
Kit de ferramentas de projeto			X	
DFM auxiliado por computador	X	X	X	X
Tecnologia de Grupo	X	X	X	X
FMEA	X			X
Análise de valores				X

Nada impede que princípios diversos aos do DFMA sejam utilizados para criar um produto ótimo. As considerações para uma otimização do produto são extremamente abrangentes e incluem:

- ♦ a avaliação da capacidade de produção atual, recursos técnicos e meios de produção disponíveis ou possíveis de serem adquiridos pela empresa ou fornecidos por terceiros, sob a forma de prestação de serviços;
- ♦ definição da melhor montagem com a operação inversa da melhor desmontagem;

- ♦ os fatores ergonômicos e o projeto a fim de criar um produto mais elegante;
- ♦ a influência do planejamento dos processos de produção (fabricação e montagem);
- ♦ a desmontagem sem impossibilitar ou inviabilizar a remontagem;
- ♦ operações simples e padronizadas (a mesma ferramenta serve para várias etapas);
- ♦ forma de distribuição, retirada e/ou reciclagem;
- ♦ a embalagem compatível com o transporte e a distribuição;
- ♦ o impacto ambiental provocado pelo produto;

O projetista deve ainda entender a interface entre o ser humano e o projeto do equipamento. Lembrar também que regras e diretrizes podem ser dadas, mas o verdadeiro saber do seu uso vem da aplicação destas na prática.

O DFA NO PROCESSO DE PROJETO

A fase do projeto conceitual pode ser descrita nas etapas seguintes:

- 1- levantamento das necessidades dos clientes e estabelecimento dos requisitos de projeto;
- 2- decomposição ou estruturação funcional;
- 3- seleção de concepções alternativas com princípios de solução gerados a partir das funções, e
- 4- avaliação e escolha das concepções alternativas viáveis.

Este início do processo de projeto caracteriza-se por trabalhar com conceitos e requer um alto grau de abstração. Por não ser uma tarefa simples, fica claro que quanto maior for a compreensão dos conceitos envolvidos por parte da equipe de projeto, maior será a eficácia do esforço empregado. Catálogos de exemplos, com princípios e diretrizes, e listas de controle (*Check-lists*) são as ferramentas de maior aplicação e eficiência nesta fase conceitual.

A importância do DFA, ou de qualquer metodologia nas etapas do projeto conceitual, deve-se ao fato de que as decisões tomadas nestas etapas têm o maior efeito nos custos de um produto pelo menor investimento. Adicionalmente, estas decisões são responsáveis pela determinação de aspectos relacionados à funcionalidade, geometria e propriedades do produto; isto é, define-se o desempenho e a competitividade do produto por todo o seu ciclo de vida. Principalmente no aspecto de qualidade, que não pode ser construída num produto a não ser que seja projetada dentro dele; e no da atratividade econômica, já que o produto deve mostrar-se economicamente viável para todos os envolvidos, desde o fornecedor de matéria-prima até o recuperador. Ratifica-se que a aplicação de DFA exige que os projetistas vejam o projeto como um todo e procurem uma solução que se preste a uma montagem eficiente melhorando, também, a manutenção, a confiabilidade e qualidade do produto final.

Segundo Alber (2000), a técnica DFA tem reduzido significativamente os custos de montagem. Um sistema inteligente baseado em análise computacional voltado para a técnica DFA foi desenvolvido para reduzir os custos de montagem do produto.

Alber (2000), diz que 75% do custo de montagem é determinado no desenvolvimento e na montagem do produto. Os produtos que são projetados visando o processo de montagem podem ser montados mais economicamente. Esse pesquisador, desenvolveu um programa que contém dois catálogos que descrevem as regras e exemplos de diversas companhias para um melhor projeto de montagem do produto.

O programa é dividido em duas partes: as regras e os exemplos. Os exemplos contém um conjunto de regras na qual descrevem a idéia geral das orientações de projeto no processo de montagem. O programa mostra exemplos de peças antes e depois do processo de reprojeção e todas as trocas efetuadas. O software é programado em Microsoft - VISUAL BASIC.

Segundo Don (2000), existem ferramentas e metodologias que são capazes de auxiliar o engenheiro projetista no processo de desenvolvimento do produto. A tabela 2.3, da página 26 ilustra algumas dessas ferramentas. Por exemplo, pode-se considerar como ferramenta chave o software de Boothroyd & Dewhurst Inc. Este software, como

dito anteriormente, é capaz de calcular o custo do processo de montagem de um produto, bem como o tempo para sua montagem e ainda dar orientações para cada peça que constitui o produto final (junção de itens, eliminação etc.). Outras metodologias de suporte ao desenvolvimento do produto são: *Configuration Management*; *Contract Manufacture and Assembly*; *Design of Experiments*; *Failure Modes and Effects Analysis*; *Software Development life Cycle*.

Como pode ser visto, através da técnica DFA consegue-se reduzir o tempo de montagem bem como unir componentes de modo a ter um custo de montagem menor.

Segundo Ferreira (2001), DFA é o método que permite avaliar as condições de montagem de um produto a partir de seu projeto. Esta análise torna-se necessária antes da adaptação do maquinário à produção do produto, pois a alteração ou eliminação de alguma das partes pode acarretar na eliminação de estações de trabalho, operações e fixações, além de reduzir os riscos do projeto fugir aos padrões de qualidade estabelecidos.

2.4 PROJETO VOLTADO PARA FUNDIÇÃO

O projeto voltado para à fundição pode ser enquadrado dentro da metodologia DFM.

Através do projeto voltado para a manufatura de uma peça fundida se consegue evitar defeitos na peça depois de pronta e possíveis custos de reprojetos. Com a intenção de auxiliar o projetista, diversos pesquisadores têm contribuído com informações, regras, recomendações, etc., para se projetar adequadamente uma peça fundida.

Smalley (1975), mostra que o projeto têm influência marcante na prática da fabricação de um fundido. Ele apresenta algumas regras básicas de projeto, que devem ser seguidas pelo projetista, para maior facilidade de fabricação das peças fundidas no processo de fundição em areia. As regras são específicas para a relação forma/tamanho da peça, em virtude dos problemas causados por esta relação. Através da geometria da peça é possível minimizar diversos defeitos que poderiam ocorrer no processo de fundição.

Segundo Doyle (1985), na fundição em areia, os defeitos como vazios e trincas são causados por uma alimentação incorreta nas seções mais grossas (espessas) das peças, ou seja, nos pontos quentes. Para evitar tais defeitos, causados por projeto pouco eficiente, deve-se ter uma boa uniformidade das seções.

Stemmer, (1979), diz que os pontos mais importantes no projeto de uma peça fundida são:

- escolha do material mais adequado;
- concordância da forma com o tipo de solicitação (tração, flexão, etc.);
- escolha adequada do processo de fundição;
- conformação levando em conta as peculiaridades da fundição;
- como evitar (bolhas, escórias e porosidades superficiais);
- como evitar as rupturas a quente e as falhas a frio;
- atenuamento das tensões residuais/envelhecimento;
- normas gerais de projeto para atenuar as tensões internas e as deformações;
- conformação da peça fundida tendo em vista o custo mínimo do modelo, fácil moldagem e fácil limpeza da peça;
- tolerância dimensional de peças fundidas;
- ângulo de saída;
- prevenção de rechupes (forma geométrica);
- sobremetal de usinagem.

Todas estas informações são bem detalhadas por ele, como forma de auxiliar o projetista no desenvolvimento do projeto de uma peça fundida. Sua principal contribuição está relacionada com informações de uma fácil usinagem e montagem.

Niebel apud Bralla (1986), mostra que para se projetar uma peça a ser obtida com o processo de fundição em molde permanente (por gravidade), o projetista deve ter todas as informações do processo com relação forma interna e externa, o tipo de macho requerido e a espessura mínima de parede.

Maclaren apud Bralla (1986), diz que algumas informações importantes sobre a fundição sob pressão devem ser dadas ao projetista, para que uma peça possa desempenhar bem suas funções, livre de possíveis defeitos. Estas informações importantes estão relacionadas com: o processo a ser utilizado, vantagens e desvantagens, aplicações, quantidade econômica de produção, material adequado, considerações e/ou recomendações de projeto. Dessa forma, a partir do conhecimento do processo de fundição, ele pode projetar bem uma determinada peça.

Knysak apud Ettore (1978), dá algumas recomendações para o projeto de uma peça fundida no processo de fundição sob pressão. Ele faz um estudo relacionado com a forma da peça, visto que o custo da peça injetada depende da dificuldade de construção do molde, e também da posição da linha de divisão do molde. São também mostrados alguns exemplos para o projeto de peças fundidas sob pressão como forma de atingir a eficiência máxima e econômica no projeto da peça.

Ruiz & Koenigsberger (1970), mostram que a uniformidade de resfriamento é uma das condições que garantem a estrutura cristalina e, por consequência, a qualidade das peças fundidas. Por isto, é preciso que os projetistas tenham muita atenção ao projetar uma peça, devendo sempre que possível, evitar: cantos vivos, variações bruscas de forma ou de espessura de paredes ou ainda grandes acúmulos de material. Segundo ele, simples considerações podem reduzir consideravelmente os custos de projeto de uma peça.

Provenza (1976), apresenta algumas regras de projeto de como evitar defeitos em peças fundidas. As regras se referem a: como evitar tensões indesejáveis na peça, forma de evitar bolhas nos diferentes tipos de peças; como fazer a subdivisão das peças complicadas de modo a tornar sua fabricação mais fácil; os moldes devem merecer boa atenção porque são eles que tornam o trabalho do moldador mais fácil e mais barato.

Kotschi apud Metals Handbook (1986), mostra que o primeiro e mais importante aspecto do projeto voltado para fundição é a solidificação e, por isto, o projetista precisa ter profundos conhecimentos sobre a solidificação. A solidificação envolve três resultados de contração no processo de resfriamento que são: contração líquido para

líquido; contração líquido para sólido; contração sólido para sólido; levando-se em conta informações referentes a esse fenômeno tem-se um projeto voltado para fundição mais eficiente e com menores chances da peça apresentar defeitos. O segundo fator que o projetista deve considerar para reduzir custos, é a complexidade da confecção do molde, que está relacionado com a exigência excessiva do número de machos.

O Cast Metals Handbook (1957), mostra o processo de fundição sob pressão, onde ele apresenta o projeto do produto voltado para a manufatura de uma peça fundida. O projeto do produto e o projeto da matriz estão intimamente relacionados. Segundo ele a característica mais importante de um bom projeto é a uniformidade da espessura da parede, porque é através da uniformidade que se consegue obter um tempo de solidificação igual em todas as regiões da peça. Ele ainda diz que os defeitos que ocorrem no processo de fundição são causados por três fontes básicas que são: problemas mecânicos no molde; problemas metalúrgicos na liga fundida; interação fluxo de calor e fluxo do fluido e sistemas de canais e massalotes.

O'meara apud Metals Handbook (1986), descreve as três categorias do processo de moldagem em areia (seu maior enfoque é nas categorias de: molde de areia ligada, molde de areia não ligada e também o menos utilizado que é do processo que usa o ligante à base de resina) e também apresenta informações sobre os equipamentos de moldagem em areia e do seu processamento.

West & Thomas apud Metals Handbook (1986), tratam do processo de fundição em molde permanente, onde é mostrado o projeto do molde em função do tipo de operação, que pode ser manual ou em uma máquina automática. Os autores fazem diversas considerações sobre: rebaixos; seções espessas e separadas; ejetor do fundido; número de fundidos por molde; solidificação progressiva adequada dentro do molde. Eles também abordam a seleção do material do molde e do macho. Segundo eles os quatro principais fatores que afetam a seleção do material do molde permanente e dos machos são: a temperatura do metal a ser fundido; o tamanho da peça a ser fundida; o número de fundidos por molde e o custo do material do molde.

Outro ponto apresentado é a influência do projeto do molde na sua vida útil, onde são mostradas as principais variáveis que afetam a sua durabilidade. E também que variações na espessura de parede do molde podem causar tensões excessivas durante o processo de aquecimento e resfriamento, tornando-se uma das causas da falha prematura do molde que é uma possível trinca (fissura).

Segundo eles, é no projeto voltado para fundição em molde permanente que se consegue uma produção de qualidade aceitável e com custos mais baixos. Para que isto aconteça existem muitas considerações envolvidas e aplicadas, e que devem ser seguidas na medida do possível. Em adição a estas considerações gerais, alguns aspectos de projeto são particularmente aplicados para abaixar os custos de produção de peças fabricadas em molde permanente, livres de defeitos. Os defeitos que podem ocorrer na fundição em molde permanente são listados por eles, e estão relacionados com: porosidades; escórias; inclusões não metálicas; peças com enchimento incompleto; junta fria; distorções e trincas. As peças mais sujeitas a todos esses tipos de defeitos são as fundidas com ligas de alumínio. As peças fundidas com ligas de magnésio podem ter alguns destes defeitos, além do mais, com ligas de magnésio é mais provável de se ter a fragilidade (quebradiço) a quente. Peças fundidas em ligas de cobre são também suscetíveis à maioria dos defeitos comuns encontrados para alumínio e magnésio, porque a alta temperatura do metal é um problema que pode acarretar um aumento da trinca superficial devido à expansão ou contração durante o resfriamento da cavidade do molde.

Outro ponto abordado é do custo total da fundição em molde permanente sendo que estes custos incluem: o custo do metal; custo de mão de obra; combustível; abastecimento; manutenção dos moldes e outros equipamento; e inspeção. Outras avaliações que podem ser feitas do custo são: custo do método automático versus manual, custo versus qualidade e custos da fundição em areia versus custos da fundição em molde permanente.

Marek (1952), apresenta o que é fundamental na produção e no projeto de fundidos. No projeto voltado para a moldagem (confeção do molde), deve-se ter procedimentos adequados para a construção dos modelos e da caixa de macho, de modo

a assegurar excelente qualidade e máximo de economia na obtenção do fundido. Ele enfatiza o projeto como forma de se eliminar defeitos e apresenta alguns fatores que minimizam os defeitos em peças fundidas. Estes fatores estão relacionados com as características inerentes ao material; as características gerais do rechupe; o rechupe volumétrico; a linha isotérmica; solidificação direcional; interseções ou junções de partes uniformemente espessas; pontos quentes formados por nervuras (saliências), letras em alto relevo, rechupe no meio da parede e linha de centro do rechupe. Segundo ele, os métodos de fundição para se controlar o rechupe volumétrico são o uso de massalotes adequados ou checar a contração do alimentador. Esta checagem pode ser feita através: da alimentação (canais), do vazamento (velocidade), das tensões (fluido), etc.

A maior preocupação de Back (1983), é criar regras gerais para os mais diferentes tipos de processo de fabricação (fundição, estampagem e soldagem) como forma de se efetuar um bom projeto. No caso específico de fundição ele menciona os pontos que se deve observar no projeto e na manufatura das peças fundidas. Ele diz que existem várias maneiras de se evitar possíveis custos causados por projetos ineficientes. As várias maneiras de evitar custos são: a simplificação dos moldes; o uso de moldes bipartidos; evitar o uso de machos sempre que possível; prever o processo de retirada da peça do molde; tomar o cuidado com o apoio e a centragem dos machos; evitar mudanças bruscas na espessura de parede. Através destas regras consegue-se minimizar a dificuldade de manufatura e de montagem de uma peça fundida.

Gaspervich (1991) diz que peças pequenas de forma complexa podem ser moldadas através da injeção de metal, mas é preciso que o engenheiro ao projetar as peças e o moldes assegure um processo adequado do escoamento do material e também uma boa uniformidade da espessura da parede.

Ganter (1993) apresenta as técnicas para minimizar as *features* (características) na confecção de modelos e machos de fundição. Suas técnicas para extração automática das *features* no projeto de machos se baseiam na representação de um modelo sólido a partir de três classes de *features* dos machos que são: vazio interno; furos em superfícies múltiplas e simples; perfuração com limite.

O projeto voltado para moldagem econômica apresenta as seguintes recomendações: trocar a divisão irregular pela divisão reta; reprojetar para se eliminar o macho; reprojetar a peça para se evitar um molde com três partes; usar o macho para evitar um molde com três partes; analisar o efeito da quantidade sobre o custo de produção; escolher o plano de divisão; analisar a viabilidade da construção fundida-soldada; incorporar algumas *features* nas peças fundidas; criar algumas *features* para ajudar no acabamento e manuseio das peças fundidas; checar a evolução do projeto; verificar o uso dos modelos.

2.5 ALGUMAS FERRAMENTAS COMPUTACIONAIS PARA AUXILIAR O PROJETISTA NO DESENVOLVIMENTO DO PROJETO DO PRODUTO

O desenvolvimento dos computadores nas últimas décadas tem trazido muitas mudanças em diversas áreas do conhecimento. Com o emprego do computador, diversas técnicas surgiram e ainda estão surgindo com o intuito de auxiliar o projetista em suas tarefas. Nas áreas tecnológicas, o computador vem sendo maciçamente utilizado nas tarefas, como por exemplo, de projeto e produção de produtos.

Nos dias de hoje, a Engenharia Auxiliada por Computador (Computer Aided Engineering - CAE), a Fabricação Auxiliada por Computador (Computer Aided Manufacturing - CAM), o Projeto Auxiliado por Computador (Computer Aided Design - CAD), o Planejamento do Processo Auxiliado por Computador (Computer Aided Process Planing - CAPP) e o Controle de Qualidade Auxiliado por computador (Computer Aided Quality Control - CAQC) são bastante utilizados no mundo todo. No Brasil estes conceitos estão, cada vez mais, sendo utilizados nas indústrias de vários setores, devido principalmente ao crescente desenvolvimento dos computadores aliados à redução de custos.

Como se sabe, há toda uma gama de tarefas que, no entanto, exigem muito conhecimento especializado, que a maioria das pessoas não possui. Essas tarefas só podem ser realizadas por especialistas que acumulam o conhecimento exigido. Programas

que realizem essas tarefas são muito úteis, pois normalmente há falta de especialistas humanos qualificados.

2.5.1 USO DO CAD

O CAD (Projeto Auxiliado Por Computador), envolve qualquer tipo de atividade de projeto em engenharia, sobre o qual se faz uso interativo do computador, a fim de desenvolver, analisar, otimizar ou modificar o produto.

Segundo Schützer (1988), o primeiro software de auxílio à engenharia foi o CAD, desenvolvido para otimizar a geração de desenhos, uma vez que esta etapa consumia uma grande parcela do tempo de desenvolvimento do projeto.

Existem várias razões fundamentais que justificam a implementação de um sistema CAD: aumentar a produtividade do projetista; melhorar a qualidade do projeto; melhorar a comunicação durante o desenvolvimento e execução de um projeto; criar uma base de dados para a manufatura. Os sistemas CAD já alcançaram um estágio muito avançado, exemplificado pelo modelamento sólido baseado em *features* e as integrações a outros sistemas de suporte à engenharia: CAE, CAM e CAPP.

Nos dias atuais prancheta, lápis, papel e gastos com protótipos estão paulatinamente desaparecendo dos departamentos de projetos e do chão-de-fábrica de certas fundições. É que o CAD/CAM, adotado por elas, acabaram por mudar radicalmente as atividades de concepção e fabricação de ferramentais para fundição e dos próprios fundidos.

O uso destes sistemas em engenharia foi uma das primeiras aplicações, no início da década de 60, substituindo a prancheta do projetista pela tela do computador. A falta de uma visão tridimensional dificultava um pouco a vida do projetista, mas a facilidade de corrigir erros e a eliminação de papéis já davam sinais dos benefícios possíveis.

Quando mais tarde surgiram os primeiros sistemas com visualização em 3D, o engenheiro pôde transpor para a tela as idéias que concebia e passou a ter muito mais recursos para evitar defeitos e expor seus conceitos.

Outra inovação que surgiu durante a evolução dos sistemas foi a implantação dos modeladores sólidos, que abreviam o tempo de concepção de peças mecânicas de geometria complexa. Assim, com um número menor de comandos e a utilização de operadores booleanos, passou a ser possível obter um modelo com quantidade maior de informações úteis ao desenvolvimento de um produto.

Os modeladores mais recentes incluem técnicas de parametrização para a criação de protótipos eletrônicos, e neste caso o projetista tem total liberdade na fase de concepção do produto, sem preocupações com dimensões, que podem ser incluídas posteriormente e armazenadas na memória do computador. Com um modelo parametrizado qualquer modificação é facilmente assimilada, reduzindo ainda mais o tempo de desenvolvimento do produto.

a) DO CAD DIRETO AO FUNDIDO, COM PROTOTIPAGEM RÁPIDA

A prototipagem rápida foi comercialmente introduzida em 1987 com a apresentação do processo de estereolitografia, criada pela empresa norte americana 3D Systems. O material empregado para a geração dos protótipos era composto de diferentes resinas, fruto de um trabalho com a Ciba-Geigy. Atualmente é um dos métodos mais comuns de prototipagem rápida juntamente com o LOM (Laminated Object Modelling) e o SLS (Selective Laser Sintering), mas existem várias tecnologias de prototipagem rápida disponíveis no mercado (Commercial Systems, 1998)

A consolidação dos dados geométricos de modelo CAD é fundamental para o sucesso de um projeto. É aí que a tecnologia da prototipagem rápida apresenta-se como necessária (Yogui, 1998).

Segundo Gorni (2001), técnicas para desenvolver protótipos rapidamente foram introduzidas visando um melhor atendimento de prazos.

b) O SISTEMA CAD (PRO-ENGINEER)

Segundo a Parametric Technology Corporation (1998), este software é indicado para o projeto e modelamento de peças em 3D. O software disponibiliza uma ferramenta denominada Pro/Toolkit para personalização do PRO-ENGINEER, objetivando reduzir o

tempo da atividade de planejamento da manufatura e erros de projeto. O Pro/Toolkit, utilizando a linguagem C, permite adaptar o software às necessidades específicas do usuário. Por exemplo, no projeto de famílias de peças, os comandos podem ser agrupados, economizando tempo do projetista no desenho de uma peça da família.

O PRO-ENGINEER possui o módulo Pro/Casting; uma ferramenta de modelamento com capacidade de simular o projeto da peça (abertura da matriz) e estabelecer parâmetros para o processo de fundição sob pressão.

O Pro/Casting tem como objetivos:

- auxiliar o engenheiro no projeto e na montagem da matriz de fundição;
- simular a manufatura do fundido.

O processo básico de fundição envolve o vazamento do metal fundido num bloco que contém uma cavidade que se assemelha à peça a ser manufaturada. Quando simulado o processo de fundição no Pro/Casting, todas as operações são executadas no modelo de vazamento. A montagem do modelo de vazamento leva em consideração: o projeto do modelo, o bloco da matriz, os dispositivos, o macho de areia e a fundição.

O projeto do modelo, o bloco da matriz, e o macho de areia são os elementos básicos para a montagem do modelo de vazamento. O projeto do modelo é na realidade a peça original, criada no PRO-ENGINEER. Quando o projeto do modelo é montado dentro do Pro/Casting, o modelo é substituído por um modelo de referência. O bloco da matriz é uma peça do PRO-ENGINEER criada para representar o volume global da matriz fechada.

O modelo de referência pode ser usado para verificar quais são as superfícies que tem ângulos de saída suficientes para permitir uma fácil remoção da matriz. Se o ângulo de saída é requerido, o Pro/Casting possui recursos para adicionar o ângulo de saída nas superfícies.

O Pro/Casting permite avaliar o bloco da matriz, os componentes da matriz e analisar a sequência de abertura da matriz. Isto faz com que se consiga inspecionar o projeto da geometria do prototipo (bloco da matriz), de modo a separar o bloco da matriz em função do volume da matriz e das superfícies definidas.

Através do Pro/Casting é possível minimizar o número de defeitos causados por projetos inadequados da matriz, e também eliminar as possíveis dificuldades de manufatura, como por exemplo, não prever adequadamente ângulo de saída do modelo (processo de fundição sob pressão). Entretanto, o PRO-E não fornece nenhum auxílio ao projetista, caso de peças que serão fabricadas através do processo de fundição em areia.

O ARQUIVO *TRAIL*

O arquivo *trail* é criado pelo PRO-ENGINEER quando se desenha uma peça. Quando o PRO-ENGINEER é executado ele cria um arquivo onde vai armazenar todos os comandos, enviados pelo usuário, em forma de texto. O nome desse arquivo tem sua extensão indexada de acordo com o número de vezes que o PRO-ENGINEER for executado com uma determinada pasta de trabalho. As linhas desse arquivo se iniciam com um caractere que vai indicar a informação contida naquela linha.

O PRO-ENGINEER possui um mecanismo de segurança, que protege o usuário de alguma falha que o faça perder as informações não salvas do seu desenho. Esse mecanismo funciona da seguinte forma: quando o usuário “passa uma mensagem” ao PRO-ENGINEER o programa grava esta “mensagem” em um arquivo no seu diretório de trabalho. Essa mensagem pode ser um click de mouse ou um texto digitado ou qualquer outra informação que o usuário forneça ao PRO-ENGINEER através de sua área de trabalho. Esse arquivo tem o formato ASCII. O PRO-ENGINEER cria um novo arquivo cada vez que é executado. O problema é que o PRO-E é um programa fechado (arquivo fonte), ou seja, as linhas geradas no arquivo *trail* da peça que está sendo criada muitas vezes são de difícil compreensão (falta de conhecimento da visão 3D do PRO-E) o que dificulta um possível reconhecimento de uma dada *feature* que está sendo desenhada.

Esse arquivo recebe o nome de *trail file*, e mostra todo o caminho trilhado pelo usuário no projeto gráfico de sua peça. O PRO-ENGINEER pode carregar este arquivo e refazer tudo o que o usuário tinha feito ao criar aquele *trail*. É muito útil em caso de perda de trabalho devido a problemas na rede elétrica ou no equipamento.

Quando não existe nenhum arquivo *trail* no diretório de trabalho do PRO-ENGINEER ele cria um arquivo com o nome *trail.txt* quando é executado. Na próxima vez que se executa o PRO-ENGINEER ele cria o arquivo com o nome *trail.Txt*. 1. Na próxima seção do PRO-E ele cria o arquivo *trail.Txt*. 2 e assim por diante. Na tabela 2.2 é mostrado um exemplo desse arquivo. Esse arquivo *trail* foi criado ao se desenhar um paralelepípedo sólido com dois furos cilíndricos.

Tabela 2.2 - Exemplo de parte de um arquivo *trail*.

Na tabela abaixo existe um exemplo desse arquivo *trail* file version No. 1052

!Pro/ENGINEER TM 2000i (c) 1988-99 by Parametric Technology Corporation All Rights Reserved.

~ Activate 'main_dlg_cur' 'ProCmdModelNew.file'

~ Activate 'new' 'OK'

@ ok

~ Move 'newtree' 'newtree' \

2 140.233723 140.233723

~ Minimize 'newtree' 'newtree'

~ FocusOut 'newtree' 'AssyTree'

#FEATURE

#CREATE

#DATUM

#PLANE

#DEFAULT

!%CIDATUM PLANE has been created successfully.

(** 0.549107 0.457672 0 1 1)

:

:

:

:

:

:

:

Continua(Ver APÊNDICE 2)

Um exemplo completo do *trail*, bem como o desenho da peça, pode ser visto no APÊNDICE 2.

No arquivo *trail* cada linha representa um comando informado pelo usuário ao PRO-ENGINEER. Cada uma dessas linhas, inicia-se com um caractere que muda de acordo com a informação contida na linha. Por exemplo, as linhas que se iniciam com #, armazenam um comando acionado no menu principal do PRO-ENGINEER (#DONE).

As linhas iniciadas por dois asteriscos indicam um ponto onde o usuário clicou na área de trabalho do PRO-ENGINEER (** 0.549107 0.457672 0 1 1).

O arquivo *trail* será a base do sistema proposto neste trabalho de pesquisa.

2.5.2 O USO DO CAE

Segundo Almeida (1997), os sistemas CAE (Computer – Aided Engineering) utilizam para a resolução de problemas de engenharia, modelos matemáticos (método numéricos), como por exemplo, o Método dos Elementos Finitos (MEF), elementos de contorno, diferenças finitas e volumes finitos. Esses métodos auxiliaram nas resoluções de problemas do contínuo (concentração de tensões, análise em pistões e materiais compostos); determinação de autovalores (frequência naturais de máquinas ferramentas) e de propagação (fraturas e rupturas sob cargas dinâmicas).

Dentro da Engenharia Auxiliada por Computador o Método de dimensionamento baseado em Elementos Finitos é um dos métodos mais utilizados no auxílio ao desenvolvimento do produto (parte de cálculo).

Inicialmente o Método de Elementos Finitos foi utilizado como uma ferramenta que mostrava porque um produto falhou e que mudanças eram necessárias a fim de que se pudesse evitar novas falhas. Recentemente, este método passou a ser utilizado para confirmar se o produto irá ou não atender às necessidades de projeto, antes mesmo do produto ser fabricado. Atualmente o Método de Elementos Finitos é muito utilizado para a determinação de tensões. O software ANSYS usa o método dos elementos finitos (FEM). As vantagens de se utilizar o programa ANSYS são: formas complexas podem ser mais facilmente descritas usando-se FEM; procedimento automático de cálculos; solução de um sistema de equações; fornece informações sobre qualquer ponto do domínio, de modo a permitir a obtenção de resultados em pontos de difícil acesso aos métodos experimentais [Ansys manual procedures, 1995].

O ANSYS é também um software avançado de solidificação, porque não modela somente os “pontos quentes”, mas também defeitos como microporosidades e diferenças na concentração da liga.

A simulação pode ser realizada usando o elemento denominado SHELL57 que faz análise linear térmica.

Através do programa ANSYS pode-se analisar a transferência de calor transiente no processo de fundição. O objetivo é analisar o caminho de distribuição de temperatura metal (aço) fundido em contato com o molde (areia), durante o processo de solidificação, este processo pode ter uma duração de 3 horas.

2.5.3 SOFTWARE PARA SIMULAÇÃO DO PROCESSO DE FUNDIÇÃO

Em função da crescente pressão pela melhoria da qualidade e redução de despesas, muitas fundições têm utilizados pacotes de *software* avançados para reduzir ou eliminar a necessidade de protótipos que utiliza o método da “teste-erro-correção-novo teste”.

Estrin (1996), apresenta os vários *software* voltados para projetos de fundidos que foram lançados no mercado nos últimos tempos. Segundo ele, existe atualmente uma grande possibilidade de escolha de *software* para a simulação do processo de solidificação. Em função desta variedade de opções é necessário avaliar os *software* e escolher o mais apropriado às suas necessidades. Em geral os *software* podem ser agrupados por preço, plataforma de trabalho e performance. Em termos de performance de avaliação há várias subcategorias a serem observadas, incluindo capacidades de pré-processamento, cálculos, capacidades de pós-processamento, facilidade de uso, visualização e suporte.

Alguns dos *Software* mais conhecidos de modelagem matemática e simulação do processo de solidificação são: AFS Solidification System; CAP, AMESH; CFDS-FLOW3D, ASTEC; FIDAP7; FLOW 3D; MAVIS DIANA; PHOENICS-2; PROCAST; RAPID/CAST; SOLSTAR; SIMULOR; SWIFT; ANSYS e MAGMA.

Naveiro (1997), desenvolveu o software DICAST: um sistema inteligente para auxílio ao projeto de peças fundidas. Este software é capaz de realçar o processo decisório do projetista, dando-lhe assistência nas diretrizes básicas do processo e possibilita a avaliação da manufaturabilidade ao longo da progressão do projeto. Porém pode-se constatar que o software é limitado para peças criadas em 3D; sua tela principal não é amigável com o usuário, devido a dificuldade de interpretação dos dados; muito genérico por não especificar o processos de fundição a ser utilizado; não existe um processo de reconhecimento de *features*; não existe uma base de conhecimento, onde o usuário possa obter orientações de como reprojetar.

Segundo Zha (1998), a modelagem e a simulação do processo de fundição proporcionam um grande número de fatores positivos para o setor, que não estão associados diretamente com a metodologia. Primeiramente, elas dão uma melhor impressão para o cliente potencial do que a apresentação de projetos em papel. O cliente tem certeza de que, diferentemente dos cálculos feito no papel, o computador é capaz de dar sempre os mesmos resultados. Usada corretamente, a simulação pode aumentar a credibilidade técnica da fundição aos olhos do seu cliente. Muito foi publicado sobre a economia de custo alcançada, utilizando simulações por computador que incluem a melhoria da qualidade, o aumento do rendimento e de cotações mais precisas.

2.6 FEATURES

Segundo Gayretli (2000) a representação de uma *feature* (característica) é definida como uma forma genérica que traduz a informação do produto (características), sendo assim facilmente identificada na modelagem do componente.

Segundo Don (1998) existem várias ferramentas que usam o conceito de *feature* no desenvolvimento do produto, como mostra a tabela 2.3.

A metodologia do modelamento baseado em *features* é uma proposta relativamente recente para promover a integração entre os vários elos de um sistema de manufatura.

Tabela 2.3 – Ferramentas para o desenvolvimento integrado do produto (Don,1998).

Tipo	Ferramenta
CAD mecânico e CAD eletrônico	Pro/ENGINEER; PT Moldeler; Hyperlynx; MicroSym PSPICE; Synario; Cadence; System Architect; Cadre; Coewrite; Various Compilers e Various in Circuit Emulators.
Definição, Planejamento, Documentação e ferramentas de administração	QFD Capture; QFD Designer; Invention Machine (Triz); Exert Choice e Automated Kano Model.
Projeto Industrial e Humano	Alias; Pro/DESIGNER; Jack; Vellum e Rhino
Ferramentas de Análise de Projeto	DFMA; C-MOLD; ABAQUS; Pro – Mesh; ANSYS e DADS
Suporte de Modelagem e Manufatura	Witness; Micro Saint; Improvit; Pro/Manufacture e Mastercam.

Uma feature deve ser capaz :

- mapear uma forma genérica, ou seja, ela tem uma sintaxe. Para representar a forma genérica de um componente, a *feature* é um elemento de forma específica (isto é uma *feature* de forma);
- ter um significado específico no contexto de engenharia. Além de ser capaz de representar uma semântica;
- ser capaz de representar uma semântica.

Uma feature pode ser:

- geométrica: como por exemplo, chanfros;
- quantitativa: como por exemplo, distância entre furos;
- qualitativo: como por exemplo, rugosidade;
- funcional: como por exemplo, alinhamento.

Algumas definições de features

Segundo Nnaji (1990) são vários os pesquisadores que já se dedicaram à conceituação e definição de *features*.

As definições mais interessantes dizem que elas:

- “representam o significado de informações de engenharia da geometria de um componente ou conjunto”;
- “representam qualquer elemento geométrico ou funcional, ou propriedade de um objeto, útil no entendimento da função, comportamento ou performance daquele objeto”;
- “ são componentes em termos de entidades de alto nível e funcionalmente significantes, como furos, rasgo e canais”;
- “ são conjuntos de superfícies com especificações das relações de contornos entre elas que implicam a função definida em termos de engenharia de um objeto”

As features aplicadas à manufatura

São entidades geométricas definidas que, além dos parâmetros dimensionais (diâmetro, comprimento, largura, altura, coordenadas espaciais, etc.), podem conter outras informações diretamente incorporadas no objeto que se está modelando, como por exemplo, dados de engenharia.

- processo de manufatura;
- material;
- rugosidade;
- tratamento superficiais;
- parâmetros do processo de fabricação;
- tolerâncias dimensionais, de forma e posição.

Segundo *Shah* (1998), as técnicas de reconhecimento automático de *features* e projeto baseado em *features* podem ser usadas como auxílio ao usuário na identificação e incorporação das *features*.

a) Reconhecimento automático de *features*

Esta técnica, também conhecida como extração automática de *features*, tem por objetivo básico a identificação de elementos (furos, rasgos, etc.) ou parâmetros que foram utilizados no modelamento de uma determinada peça (acabada) a partir de *features* pré-definidas.

Assim, ao analisar uma peça modelada, o sistema partir de algoritmos com regras e procedimentos bem definidos, que “reconhece” se foi utilizada alguma *feature* pré-definida pelo sistema e “extraí” os dados necessários.

Como exemplo, essas *features* podem representar operações específicas de usinagem como: furos, rasgos, rebaixos.

b) Projeto baseado em *features*

Ao contrário do método descrito acima esta técnica é também conhecida como projeto através de *features*, utiliza o modelamento baseado em *features* a partir da concepção do modelo.

Normalmente esta técnica utiliza *features* pré-definidas e disponíveis numa biblioteca, que podem conter um número finito e pré-definido de *features* ou, ainda, permitir ao usuário incorporar novas *features* segundo a necessidade.

A expressão *form features* é utilizado como termo genérico para *design features*, *manufacturing features*, *function features*, etc.

Reconhecimento de *form features*

A peça é inicialmente construída com uso de sistemas convencionais. Usando um sistema complementar, onde serão extraídas as *manufacturing features* a partir de informações geométricas do sistema. Para que isto aconteça existem duas possibilidades:

- reconhecimento manual;
- reconhecimento automático.

a) Reconhecimento manual

O usuário irá interagir com o sistema que possibilita a identificação dos elementos que irão compor uma *feature* de fabricação.

b) Reconhecimento automático

O projetista irá interagir com uma interface baseada nesta nova semântica durante todas as fases do projeto e, portanto, cada uma das peças já estará sendo modelada com o uso destes elementos.

Representação de uma *feature*

As *features* podem ser representadas por volumes, que podem ser representados por superfícies, que podem ser representadas por planos, que podem ser representadas por arestas, e que finalmente podem ser representadas por vértices. As operações entre esses elementos podem tornar o projeto pré-definido e portanto o número de comandos inseridos para o modelamento passa a ser menor.

Isto faz com que o usuário possa rapidamente detalhar ou simplificar um modelo que venha a ser colocado num programa de análise de elementos finitos.

Alguns conceitos para o modelamento de *features*

O primeiro define *features* como entidades geométricas com aspectos funcionais e de fabricação, permitindo que suas funções sejam transformadas em métodos de fabricação através da geometria.

O segundo trata *features* como regiões ou partes da peça que possuem algum significado, para o processo de manufatura.

Segundo Chen, & Changghien (1998) o modelamento de *feature* é uma ferramenta com alto grau de tecnologia incorporada: as metodologias de modelamento baseado em *features*, também podem auxiliar o desenvolvimento de outras frentes de desenvolvimento tais como: engenharia simultânea; inteligência artificial; manufatura celular; ferramentas assistidas por recursos computacionais, como CAD, CAE, CAM,

CAPP, etc., pois constituem uma base tecnológica de informações consistentes sobre o conhecimento e das informações disponíveis no sistema.

Pode-se utilizar a metodologia do modelamento baseado em *form features*, como uma biblioteca que inclui *design features* e *manufacturing features*.

O projetista não precisa mais trabalhar com uma geometria de nível elementar porque, o uso de *features* auxilia o modelamento com uma nova semântica baseada em elementos, o que melhora a interface com o usuário e permite reduzir o tempo de projeto.

A vantagem do modelamento de peças baseado em *features*, com sistemas comerciais resultam globalmente em uma significativa redução do tempo de modelamento e geração do programa CNC. Essa metodologia possibilita a integração entre projeto, processo e a manufatura.

2.7 BASE DE CONHECIMENTO OU BASE DE DADOS

Segundo Costa (1996) a Base de Conhecimento é o componente vital de um Sistema Especialista. É através da Base de Conhecimento que se faz a aquisição do conhecimento sobre um determinado assunto de interesse.

Bases de conhecimento podem não parecer diferentes de bases de dados. Ambas são estruturas de dados, ou conjunto de estruturas de dados, nas quais a informação é armazenada e recuperada. A tabela 2.4, mostra o que diferencia uma base de conhecimento de uma base de dados.

Tabela 2.4 - Distinção entre uma base de dados de uma base de conhecimento (Costa, 1996).

Diferenciador	Base de Dados	Base de Conhecimento
Coletor	Escriturário	Especialista
Uso	Recuperação	Múltiplo
Tipo de Informação	Fatos	Alto Nível
Requisitos Teóricos	Teoria da Computação	Interpretação

Conforme o primeiro diferenciador, quando um especialista – alguém com conhecimento especializado em algum domínio específico – é necessário para coletar a informação, aquela informação é conhecimento e é armazenada numa base de conhecimento; contudo, quando um escriturário pode fazer – conhecimento especializado não é necessário – a informação é um dado e é armazenada numa base de dados.

O segundo diferenciador indica que bases de dados são usadas apenas para recuperar informação, enquanto bases de conhecimento possuem muitos usos: informar, diagnosticar, monitorar e assim por adiante.

O terceiro diferenciador é baseado no tipo de informação armazenada: fatos, tais como uma descrição de uma conta bancária, são dados; abstrações de níveis mais elevados – por exemplo, uma afirmação que baleias são animais – é conhecimento.

De acordo com o quarto diferenciador, as bases possuem requisitos teóricos diferentes. Uma base de conhecimento possui uma interpretação, ou seja, uma especificação de seu relacionamento com o mundo real. Por outro lado, uma base de dados é guarnecida com uma teoria computacional de forma que possa gerenciar o armazenamento físico das informações.

De certo modo, as bases de dados possuem uma semântica, mas é o usuário ou analista que realiza essa interpretação. Por exemplo, interpreta-se um registro de empregado na base de dados como uma referência a um empregado

CONCLUSÃO

Na pesquisa bibliográfica realizada neste trabalho encontrou-se um volume grande de informações de ferramentas de auxílio ao projetista, como relatado anteriormente, porém pode-se observar que estas ferramentas trabalham isoladamente.

Várias técnicas foram apresentadas tais como: Engenharia Simultânea, DFM, DFA, DFC, bem como Ferramentas Computacionais (CAD, CAE, Software de Solidificação), *Features*, Base de dados e de conhecimento.

No caso da Engenharia Simultânea, pode se observar que existem grandes vantagens no seu emprego no processo de desenvolvimento do produto, porém pode-se notar que existe uma dificuldade na obtenção das orientações devido à falta de disponibilidade do especialista.

As técnicas de DFM, DFA e DFC, são de muita valia para o projetista, porque detém conhecimento armazenado em anos de experiência, bem como utiliza a técnica de medir a eficiência do projeto. O problema é que o uso destas orientações podem causar uma demora no processo de desenvolvimento do projeto.

Ferramentas Computacionais (CAD, CAE, Software de Solidificação) são importantes porque o computador é capaz de dar sempre os mesmos resultados, melhorar a qualidade do projeto e tornar a vida do projetista mais fácil, principalmente nos processos repetitivos. O problema é que o sistema é muito rígido, ou seja, o projetista obtém os resultados em função do conhecimento que ele detém sobre o assunto. O que muitas vezes pode induzi-lo a erros.

Através de um conjunto de *Features* é possível representar uma peça e analisar separadamente cada uma delas quanto a possíveis violações de projeto. Porém é necessário o uso de um software (parametrizável), por exemplo o CAD (PRO-E).

Com uma Base de Conhecimento pode-se armazenar o conhecimento específico desejado. Também é necessário o uso de um software para aquisição e recuperação do conhecimento.

Como foi mencionado anteriormente existem várias formas de auxílio ao projetista, porém pode-se constatar que existem limitações e falta de integração. Seria bastante útil o desenvolvimento de um sistema integrado que pudesse utilizar as vantagens de cada um, de modo a facilitar a vida do projetista no processo de desenvolvimento do produto.

CAPÍTULO – 3 - PROJETO DE PEÇAS FUNDIDAS VISANDO UMA FABRICAÇÃO ECONÔMICA

Neste capítulo será apresentada uma breve introdução ao processo de fundição em areia, bem como exemplos de orientações de projeto visando uma manufatura de peças fundidas livres de defeitos. Estas orientações visam auxiliar o projetista tanto na manufatura como no projeto de uma peça fundida no processo de fundição em areia.

3.1 PROCESSO DE FUNDIÇÃO EM AREIA

O processo de fundição em areia consiste basicamente, em encher com metal líquido a cavidade de um molde (mistura de areia, aglutinante, água e outros ingredientes) cujas dimensões e formas correspondem às das peças a serem obtidas. Após a solidificação e resfriamento obtêm-se as peças com formas e dimensões, geralmente, quase definidas, pois, em muitos casos, as peças são usinadas antes de estarem em condições de utilização. A figura 3.1 ilustra um molde de areia com os seus respectivos elementos.

Figura 3.1 - Processo de fundição em areia [MED, 1988].

Como a fundição é um dos processos mais antigo utilizado na fabricação de um produto, o seu conhecimento é de fundamental importância para auxiliar o projetista no projeto de uma peça fundida. Este auxílio ao projetista pode ser obtido através das orientações relativas ao processo de fundição, de modo a permitir a formação de uma base de conhecimento. Através das orientações sobre fundição pode-se auxiliar o projetista nas recomendações de projeto pertinentes ao processo de fundição em areia, de modo a desenvolver o projeto de uma peça, livre de defeitos, com a máxima qualidade e com menor custo.

Existem vários processos de fundição, mas segundo Back (1983), os três principais são: fundição em molde permanente, fundição sob pressão e fundição em areia (Areia Natural, Areia Sintética e Areia Semi-Sintética). Esta classificação é usualmente feita em função do tipo de molde empregado [Kalpakjian, 1989].

Etapas de fabricação de peças fundidas

Para que uma peça fundida saia perfeita é preciso que haja uma interação entre projeto e processo. Esta interação se dá através do conhecimento das etapas de fabricação de uma peça fundida. A partir deste conhecimento o projetista minimizará o aparecimento de possíveis defeitos através da adequação do projeto ao processo de fundição. Essas etapas de fabricação são discutidas a seguir.

As etapas de fabricação de uma peça fundida são:

- escolha da liga;
- fabricação dos modelos (modelação);
- fabricação dos moldes (moldagem);
- fabricação dos machos;

Escolha da Liga

Quando os requisitos de projeto exigem condições adversas, o projetista precisa selecionar a liga para depois selecionar o processo de fabricação. As questões que o projetista deve analisar na escolha do material requerido em projeto são:

- 1 - O material possui requisitos de desempenho? As características do ambiente de trabalho são completamente definidas?
- 2 - Existe uma boa correlação entre o requisito de desempenho e as propriedades do material? São estes dois requisitos utilizados na avaliação do material requerido?
- 3 - As possíveis modificações das propriedades do material no decorrer do processo de fabricação estão sendo consideradas?
- 4 - A forma e configuração requerida do material é disponível e tem um preço aceitável?

Para selecionar o material, o projetista necessita dos requisitos de projeto e das seguintes informações sobre o material:

- suas propriedades (mecânica, física, etc.);
- sua fabricação (precisão necessária, acabamento superficial, etc.);
- forma final da peça fundida (barra, tubo, etc.);

Para selecionar a liga deve-se obter informações sobre os seguintes fatores:

- Condições de trabalho;
- Tipos de cargas (estáticas ou dinâmicas);
- Tipos de tensões e deformações sofridas em serviços;
- Propriedades de fundição:
 - capacidade de preencher o molde (Fluidez);
 - baixa contração volumétrica durante o resfriamento;
 - insensibilidade a tensões de fundição ou ao fenômeno de segregação na estrutura bruta de fusão.

Para analisar as propriedades de fundição, uma dificuldade que se pode ter é que as propriedades físicas dos líquidos de maior interesse na metalurgia são de difícil medição, sendo obtidos através de processos empíricos ou de medidas diretas. Fluidez ou contração são exemplos de fatores que podem trazer problemas na fundição das ligas [Kondic, 1973].

Uma vez respondido o questionamento, a seleção do material pode ser feita segundo as etapas abaixo discriminadas.

1- Análise e escolha da matéria-prima requerida:

A análise e seleção da matéria-prima requerida envolve os seguintes fatores:

a) como primeiro fator as solicitações mecânicas que o material vai sofrer (tração, compressão, flexão, torção, cisalhamento e abrasão);

b) as solicitações químicas que o material vai sofrer estão relacionadas com o meio de atuação, que muitas vezes pode ser agressivo, fazendo com que a peça sofra corrosão reduzindo assim a sua vida útil;

c) a temperatura com que a peça fundida vai trabalhar pode causar dois tipos de problemas: a fragilização do material quando a temperatura é baixa e a fluência do material quando a temperatura é elevada;

d) a condutividade elétrica, térmica ou magnética, pode muitas vezes influenciar de forma decisiva na escolha do material da peça em função das necessidades de projeto como, por exemplo, de dissipação de calor;

e) as propriedades mecânicas e físicas da peça sofrem alterações durante o resfriamento e sob as pressões exercidas sobre o metal durante o resfriamento, agindo de maneira decisiva sobre as propriedades da maioria dos metais;

f) o custo do material é um fator de extrema importância e pode ser minimizado com o aproveitamento das sobras e a escolha do material com propriedades mais ajustadas às necessidades;

g) a disponibilidade da matéria-prima é outro ponto de extrema importância no projeto.

2- Identificação e seleção da matéria-prima alternativa.

Seguem-se as mesmas recomendações do item 1 (análise da matéria-prima requerida).

3- Avaliação da matéria-prima que permanece.

4- Desenvolvimento dos dados de projeto para aplicação crítica (serviço).

Uma vez consideradas as etapas na seleção do material, o projetista é capaz de fazer a escolha adequada do material em função dos requisitos exigidos em projeto, fazendo com que a vida útil da peça fundida aumente.

A partir desses fatores, o projetista escolhe o material mais adequado e seleciona o processo de fundição em função da liga escolhida.

Modelação

A modelação consiste em confeccionar o modelo de fundição. Os modelos de fundição são usados para dar forma à peça fundida.

Na confecção do modelo, o projetista deve considerar o sobremetal para usinagem e a contração linear do metal [Pereira, 1991]. Os modelos de fundição podem ser projetados com várias características para adaptar à sua aplicação e necessidade econômica. O modelo de fundição para uma peça de formas simples e de baixa quantidade de produção é geralmente feito de madeira, pois este é mais barato. Já para o caso de alta produção são indicados os modelos feitos em alumínio [Kalpakjian, 1989].

Moldagem

A moldagem consiste em obter o molde para a obtenção da peça fundida. O molde é o recipiente que contém as cavidades com a forma da peça a ser fundida e no interior das quais o metal líquido será vazado. Os processos de fundição são escolhidos de acordo com o tipo de molde empregado. Os moldes empregados em fundição são subdivididos em consumível e permanente [Dieter, 1983].

No processo de fundição em areia, utiliza-se o molde de areia, que emprega o modelo permanente para dar forma à peça fundida. Na moldagem em areia o projetista

precisa ter alguns cuidados com relação ao molde. Para isto alguns requisitos devem ser preenchidos pelo molde:

- suportar a ação erosiva do metal líquido no momento do vazamento;
- apresentar resistência suficiente para suportar o peso do metal líquido;
- gerar a menor quantidade possível de gás, de modo a evitar erosão do molde e contaminação do metal;
- deve facilitar a fuga de gases gerados para a atmosfera.
- especificar a granulometria da areia em função do acabamento desejado.

Num molde de areia existem vários elementos a serem considerados para que a peça fundida saia perfeita no final da fundição. Por isto o projetista precisa conhecê-los bem de modo a projetá-los adequadamente [Kalpakjian, 1989]:

- areia;
- molde interior;
- macho (areia);
- cavidade do molde;
- caixa superior do molde;
- saída de gás
- massalote aberto;
- canal de descida;
- canal de distribuição;
- canal de ataque;
- linha da peça;
- caixa de moldar;
- bacia de vazamento;
- massalote não ligado a atmosfera;
- choque (Secção do sistema de alimentação que regula o fluxo de metal);

É através do controle dos elementos do molde que o projetista evita possíveis defeitos e até mesmo um possível reprojeto da peça fundida. Os outros fatores que determinam uma moldagem adequada no processo de fundição em areia são:

- a areia para macho;
- os dispositivos de canais;
- o equipamento adequado;
- a confecção dos machos [Bradaschia, 1980];
- a escolha da areia de moldagem [Pereira, 1991].

Fusão

A fusão é o processo pelo qual se obtém o metal na forma líquida. Existem vários tipos de equipamentos ou fornos construídos para a fusão dos metais e preparo das ligas [Chiaverini, 1977]. A fusão é um dos aspectos mais importantes na operação de fundição a que o projetista deve ter atenção, porque é onde se tem um efeito direto sobre a qualidade das peças fundidas.

A seleção do forno requerido merece cuidadosas considerações, porque é um dos fatores que podem influenciar significativamente no tempo gasto para fundir a peça e consequentemente no custo final.

Existe uma variedade de fornos para a fusão do metal puro e de ligas compostas de vários metais; deste modo, os fatores que o projetista precisa considerar para selecionar o forno adequado são:

- a) considerações econômicas, tais como custos das operações iniciais e custo de manutenção;
- b) verificar se a unidade (tipos fornos de fusão) é capaz de levar uma dada massa de metal, com composição química especificada, para a temperatura de vazamento numa velocidade de fusão desejada e com eficiência econômica [Resende, 1992];
- c) controle da atmosfera para se evitar a contaminação do metal;

- d) a capacidade e a velocidade da fusão requerida;
- e) considerações ambientais, tais como poluição do ar e ruído;
- f) suprimento de força e o custo do combustível;
- g) facilidade de superaquecimento do metal;
- h) tipo de material-carga que pode ser usado [Kalpakjian, 1989].

Os tipos de fornos mais utilizados são:

- forno cubilô - ar frio (coque);
- forno cubilô - ar quente (coque);
- forno a arco;
- forno de indução (bifásico elétrico);
- forno de indução (monofásico elétrico);
- forno de resistência com grafita (elétrico);
- forno a óleo.

Dentre os vários tipos de fornos o mais indicado para fusão de metais (ferrosos e não ferrosos) é o forno elétrico de indução. Em resumo, a escolha do forno adequado é feita em função do tipo de metal empregado. Para evitar uma possível contaminação do metal é necessário, que o metal, quando líquido seja de alta qualidade e com especificações adequadas. A qualidade do metal líquido depende:

- pureza dos materiais;
- extensão com que as condições de pré-tratamentos e de fusão possam ser determinadas visando excluir ou eliminar contaminações.

Pode-se adotar um bom número de operações específicas para auxiliar a otimização da qualidade do líquido. São elas:

- seleção cuidadosa do material da carga e pré-tratamento para minimizar a contaminação;
- utilização de técnicas cuidadosas de fluxação para remover gases dissolvidos e inclusões insolúveis;

- filtração e degaseificação ativa, para reduzir os contaminantes a um nível o mais baixo possível;
- fundição em atmosfera protetora e/ou utilização de cargas com formas que minimizem o contato com o ar [Campos Filho, 1978].

Vazamento (temperatura de superaquecimento x espessura da peça x tipo de liga)

O vazamento é o processo pelo qual se enche o molde com o metal líquido. Existem vários pontos por onde se pode vazar: pelo topo, pela base, pela divisão do molde, por degraus, por faca ou contínuo.

Considerações sobre o vazamento:

- sempre é importante que haja metal suficiente para encher completamente o molde, os canais e os massalotes;
- o vazamento pode ser feito continuamente e a uma velocidade uniforme até que o molde esteja cheio, a fim de impedir que a escória sedimente;
- a temperatura do metal deve ser alta o suficiente para que o fluido escoe fácil e rapidamente;
- no vazamento do ferro líquido a escória está sempre presente e deve ser mantida fora do molde, a fim de evitar regiões fracas na peça [Doyle, 1985].

No processo de vazamento pode haver alguns tipos de problemas como bolhas, escórias e porosidades superficiais. Para sanar tais defeitos é preciso que ao se projetar a forma da peça fundida, permita-se a livre ascensão dos referidos defeitos.

A seguir serão mostrados alguns tipos de defeitos e a forma de saná-los:

- bolha: é minimizada através do controle da umidade da areia (adequada) ou de um sistema de alimentação adequado com alta velocidade de fluxo;
- escórias: são minimizadas através da colocação de filtros ou de sistemas de decantação de escórias nos sistemas de canais;
- porosidades superficiais: são minimizadas deixando-se o metal resfriar no forno ou reduzindo o teor de umidade da areia.

Desmoldagem

A desmoldagem tem como finalidade separar a areia das peças solidificadas (quebra do bolo). Uma vez desmoldada, a peça sofre as operações de rebarbação, limpeza e inspeção antes de ser utilizada.

Métodos de retirada da peça fundida no processo de fundição em areia:

- 1- levantar a caixa superior, por meio de uma talha e bater nas mesmas com marreta ou martelo, método mais antigo. Em seguida sacudir a peça fundida sobre uma grade vibratória, através do qual passa a areia
- 2 - para remoção de machos pode-se usar também o processo de vibração [Bradaschia, 1980], nesse processo de fundição, o método de retirada da peça fundida não é limitado pela complexidade da peça quanto à superfície externa do molde;
- 3 - tambor rotativo (tamboreamento) – a areia e as peças entram no tambor por intermédio de uma calha vibratória a qual se encaixa perfeitamente na entrada do tambor. Uma vez dentro do tambor, os elementos abrasivos (estrelas) formam uma cama de proteção para as peças ao mesmo tempo em que limpam e retiram a areia das peças. À medida que as peças limpas saem do tambor, os pedaços de machos e de metais passam através de outra peneira, enquanto que os elementos abrasivos (estrelas) voltam diretamente para a câmara de entrada, para nova circulação (método mais atual) [Fundição & Matérias-Primas, 2000].

Para maior comodidade e segurança, ao projetar a forma da peça, o projetista precisa prever a retirada do molde, ou seja, é necessário prever o processo de desmoldagem da peça fundida.

Rebarbação

A rebarbação é a operação através da qual são removidos os canais, alimentadores e as rebarbas na superfície (interna e externa) das peças fundidas

[Chiaverini, 1977]. As rebarbas surgem mesmo que o molde esteja bem ajustado. Tais rebarbas aumentam pelo desgaste causado com o uso corrente do molde na obtenção da peça fundida.

A operação de rebarbação pode ser feita de dois modos:

- 1 - a peça pode ser completamente rebarbada através do estampo de corte;
- 2 - as rebarbas podem ser retiradas manualmente com auxílio de martelo de madeira, lima, lixa rasquete.

Cuidados na hora da rebarbação com o meio utilizado para sua remoção:

- 1) Através do estampo de corte dois defeitos podem surgir, devido ao ajuste da peça ao estampo:

- a) aparecimento de riscos, marcas;
- b) deformação devido ao esforço excessivo de corte ou extração da peça.

Solução para estes problemas:

- fazer ressaltos nos canais e bolsas de ar para servirem de guia para peça no estampo de corte;
- adequar a posição da ferramenta;
- mudar a posição da rebarba.

- 2) Através da rebarbação manual, os defeitos são geralmente causados pela falta de habilidade do operador. [Ettore, 1978]

Limpeza

A limpeza que uma peça fundida deve sofrer se divide em duas etapas: limpeza grosseira e limpeza da superfície.

- a) A limpeza grosseira é destinada à remoção de canais e de alimentadores, para isso utiliza-se a serra de fita ou disco de corte sendo a escolha feita em função do tipo de liga. A limpeza de canais pode ser feita utilizando-se o maçarico eletrodo de grafite para remoção dos mesmos. Já para o caso de canais grandes utiliza-se o corte por discos de cortes de óxido de alumínio ou serra de fita para o corte de canais. Em ligas de alumínio

e de magnésio utiliza-se a serra de fita, enquanto que em ligas de cobre e bronze utilizam-se serras de fita ou discos de corte de óxido de alumínio.

b) A limpeza da superfície destina-se à parte interna e à externa da peça, utilizam-se os processos de jateamento de granalha de aço, tamboreamento e o rebolo de retificação.

Inspeção

A inspeção na peça fundida tem dois objetivos: verificar se existem defeitos e corrigir os defeitos encontrados se possível.

a) Verificação de defeitos na peça fundida: a inspeção de uma peça fundida pode ser realizada por dois métodos: destrutivo e não destrutivo.

No método destrutivo a amostra fundida é seccionada, o metal cortado é examinado de modo a se observarem os defeitos internos e acuracidade dimensional.

Na inspeção não-destrutiva incluem-se os seguintes métodos:

- 1 - exame visual;
- 2 - são (sadio) e percussão (teste qualidade);
- 3 - impacto (teste);
- 4 - supersônico (teste);
- 5 - pressão (teste);
- 6 - penetrante;
- 7 - condução elétrica;
- 8 - partícula magnética;
- 9 - raio-X; e
- 10 - raio Gama [Marek, 1952];

b) Correção dos defeitos encontrados na peça: uma vez constatada a existência de defeitos na peça fundida, o projetista precisa de todos os parâmetros relevantes com relação ao processo e ao projeto, de forma a processar de novo a peça ou até mesmo fazer um novo projeto. Para tal o projetista precisa rever as informações dos processos de fundição bem como as recomendações de projeto para eliminar as falhas na peça fundida.

Tratamento Térmico

O material da peça fundida, antes de ser vazado no molde, passa pela operação de fusão, onde muitas vezes há variações de temperatura de vazamento. Estas mudanças de temperatura de vazamentos podem alterar as propriedades de alguns tipos de ligas, como por exemplo, o coquilhamento do ferro fundido cinzento ou nodular [Siegel, 1982].

Os ferros fundidos (cinzentos e nodulares) podem ser submetidos a diversos tratamentos térmicos de modo a se atingir os seguintes objetivos:

- a. Alívio de tensões provocadas pela solidificação;
- b. Obtenção de maior ductilidade e melhor usinabilidade;
- c. Melhoria das propriedades mecânicas de resistência;
- d. Decomposição de carbonetos, ou
- e. Endurecimento.

O objetivo tecnológico da maior parte dos tratamentos térmicos realizados em ferro fundido é o atendimento às especificações, tanto no que se refere à microestrutura como às propriedades mecânicas. Como exemplo deste fato, pode-se citar o emprego do recozimento na eliminação de carbonetos em peças finas, ou têmpera e revenimento para aumentar a dureza [Souza Santos, 1977].

Terminadas as etapas de fabricação da peça, esta vai para oficina mecânica de usinagem para receber os últimos ajustes. Nesta fase a peça pode ou não sofrer acabamentos posteriores (niquelação e cromação), conforme necessidade. [Bradaschia, 1980].

3.2 ORIENTAÇÕES DE PROJETO E DE PROCESSO PARA O PROCESSO DE FUNDIÇÃO EM AREIA

Ao projetar uma peça fundida usando a metodologia da Engenharia Simultânea o projetista nem sempre tem facilidade em obter o conhecimento dos diversos especialistas nas diversas áreas que contribuem para o projeto de uma peça fundida, devido à indisponibilidade do especialista ou à falta do especialista.

Para que se obtenha hoje, informações confiáveis acerca destas diversas áreas de conhecimento é necessário que o projetista busque respostas junto a especialista em fundição, livros especializados e centros de pesquisa.

O problema é que nem sempre os cronogramas prevêem o tempo a ser despendido com esta pesquisa. Para auxiliar o projetista na obtenção destas informações de forma rápida e confiável este trabalho utiliza um guia DFM que contém um conjunto de conhecimentos que irão contribuir para se ter um bom projeto de peças fundidas.

O guia DFM, é constituído por regras ou orientações sistemáticas de boa prática de projeto, e que foram obtidas em anos de experiência em projeto e manufatura de peças fundidas. Estas orientações podem ser como especificar um raio de borda adequado, como especificar o ângulo de saída, como definir o sobremetal mínimo, etc. Neste item serão mostrados as recomendações referentes ao processo de fundição e também as recomendações referentes ao projeto para peças fundidas.

3.2.1 CONSIDERAÇÕES SOBRE A GEOMETRIA DA PEÇA

Nesta seção será apresentado uma pequena amostra do conhecimento ou informações que o projetista do produto deve dispor para um projeto adequado de uma peça a ser fundida no processo de fundição em areia. Estas informações sobre a geometria visam auxiliar o projetista nas recomendações de projeto para: espessura na seção; raios de borda em reentrâncias e ressalto; raios de extremidade em nervuras; seções adjacentes; seções afiladas; proporções das nervuras; saliências com ressalto;

ressaltos para assentamento de cabeça de prendedor e do acréscimo de material para reforço.

A espessura na seção e o raio de arredondamento influem nas propriedades físicas da peça fundida e no preenchimento completo das paredes do molde. O raio de arredondamento é estabelecido para junções arredondadas entre duas paredes de mesma espessura.

Através das informações sobre o modo correto de se especificar a concordância entre paredes (concordância em L) o projetista reduzirá as concentrações de tensões e o acúmulo de metal na parede da peça.

Um exemplo do conteúdo colocado no guia pode ser visto na figura 3.2, que mostra a concordância entre paredes para uma peça em forma em L a ser fundida no processo de fundição em areia. A figura 3.2 ilustra o valor do raio de concordância para evitar a concentração de tensão.

RECOMENDAÇÃO	ILUSTRAÇÃO
<p>Evite um raio de valor grande porque a massa de metal é aumentada e possíveis rechupes aparecerão.</p>	<p><i>Modo correto</i></p> <p>O raio de concordância para evitar a concentração de tensão deve ser:</p> $r = E \quad \text{ou} \quad \frac{r}{E} = 1$

Figura 3.2 - Forma em L (mesma espessura) para o processo de fundição em areia [SENAI,1987].

A uniformidade da espessura na seção é considerada o fator mais importante no projeto de fundidos. Na figura 3.3 está mostrado o modo recomendado e o não recomendado no projeto da seção de uma peça. O item “c” da figura 3.3, por exemplo, mostra como projetar a uniformidade da espessura da peça (recomendado) e como um projeto inadequado pode causar uma provável porosidade (não recomendado).

Figura 3.3 - Uniformidade da espessura na seção [MED, 1988].

A figura 3.4 apresenta recomendações com relação ao raio de borda. Os raios de borda não devem ser menores do que 1mm. No caso das reentrâncias e ressaltos muito

pequenos, apesar de existir um mínimo para o raio de borda que é de 0,4mm, recomenda-se usar o maior raio de borda possível.

Figura 3.4 - Raios de borda recomendados para reentrâncias e ressaltos [MED, 1988].

Quando uma peça fundida tiver arredondamento externos em nervuras, então os raios de extremidade correspondentes às diversas larguras de uma nervura, podem ser especificados como mostrado na figura 3.5.

	T - mm	R - mm
	até 6,4	$0,7 \pm \begin{matrix} 0,7 \\ 0,5 \end{matrix}$
	de 6,4 a 16,0	$1,5 \pm \begin{matrix} 1,5 \\ 0,7 \end{matrix}$
	de 16,0 a 25,4	$2,3 \pm 1,5$
	acima de 25,4	$3,0 \pm 1,5$

Figura 3.5 - Raios de extremidade recomendado para nervuras [MED, 1988].

Recomenda-se um raio de extremidade de 0,7 mm para reentrâncias e ressaltos. (figura 3.5). Quando a extremidade coincide com a linha de partição, o canto vivo é inevitável, sendo usualmente removido pelo fundidor.

No projeto que envolve seções adjacentes, sempre que possível o número de seções deve ser mínimo ou a espessura da parede interna deve ser 9/10 da espessura da parede externa, (figura 3.6). O item “a” da figura 3.6 mostra, por exemplo, que uma junção larga entre as paredes requer o uso de macho. O modo correto de projetar um a seção adjacente sem o uso de machos é ter uma menor largura da seção.

Figura 3.6 - Seções adjacentes [Metals Handbook da ASM].

3.2.2 CONSIDERAÇÕES SOBRE O PROJETO DO FUNDIDO

Além das orientações já apresentadas sobre a geometria de uma peça, o projetista necessita de outros tipos de orientações importantes sobre o projeto do fundido, como por exemplo: projetar o modelo, escolher o material do modelo, prever a retirada do modelo do molde, prever a contração da peça, etc.

Novamente, este tipo de orientação (figuras 3.7 a 3.11 e tabelas 3.1 a 3.4) se encontra espalhado na literatura. O guia de auxílio ao projetista de peças fundidas referido anteriormente está melhor detalhado no item 3.3 e fornece recomendações de

projeto para o fundido. As orientações sobre o projeto do fundido estão divididas em quatro grupos. O primeiro grupo trata das operações finais de obtenção da peça (fundido) que engloba informações sobre desmoldagem; limpeza; remoção de canais e massalotes e rebarbação.

Os modelos devem ser fabricados de madeira (cedro, peroba, pinho), plástico ou metal (latão, bronze, metal branco, ferro fundido, liga de alumínio). Usualmente o material mais utilizado na fabricação de um modelo é o alumínio [Marek, 1952],[Chiaverini, 1977].

A seleção do material do modelo deve ser feita primeiramente baseada nos seguintes fatores: aplicação; tamanho e peso; acuracidade dimensional; número de peças a ser fundida; usinabilidade; resistência mecânica; possibilidade de reparos; resistência a corrosão e ao inchamento; o processo de moldagem empregado [Kalpakjian, 1989] & [Marek, 1952].

Os modelos precisam ser facilmente removidos devido à fragilidade do molde de areia. É através do ângulo de saída do modelo que se consegue facilitar a retirada do molde. Ângulo de saída é uma inclinação dada à superfície do modelo paralela à direção de sua saída, para facilitar a retirada do modelo do molde.

A tabela 3.1 mostra o ângulo de saída para a superfície externa de um modelo para moldagem em areia.

Tabela 3.1- Ângulo de saída em função do material do modelo x método de socagem [Bralla, 1986].

	Material do modelo					
	Madeira		Alumínio		Ferrosos	
Método de Socagem	Nível de qualidade do modelo					
	Normal	Alto	Normal	Alto	Normal	Alto
A Mão	5°	3°	4°	3°	1°	
Automático			2°	1°	1.5°	0.5°
Compressão	3°	2°	3°	2°		

* O ângulo interno da superfície pode ser duas vezes o ângulo da superfície externa.

Sabe-se que o modelo deve ser maior, de modo a prever a contração linear da peça na fundição. Através das margens dimensionais recomendadas para o modelo, consegue-se prever a contração do metal ao se solidificar. O processo de contração ocorre em função da temperatura de vazamento, da espessura da peça, da forma e da composição do metal.

A tabela 3.2 apresenta o material da peça (metal) e a contração correspondente [Bralla,1986], [Resende, 1992] e [Beeley, 1972].

Ao projetar o modelo, é preciso prever o sobremetal para a operação de usinagem de acabamento. As margens dimensionais recomendadas nos modelos prevendo a usinagem de acabamento, são mostradas na tabela 3.3 .

Algumas Informações Básicas do Processo e de Projeto para os Principais Processos de Fundição (fundição em: areia, molde permanente e sob pressão).

A seguir serão mencionadas algumas informações básicas de processo válidas para os principais processos de fundição.

Informações gerais:

- o uso de machos deve ser sempre evitado para não haver formação de rebarbas nas peças, que são causadas pela penetração do metal no interior das junções entre moldes e machos. O custo envolvido na rebarbação e limpeza das peças formam parte considerável do custo total.
- evitar depressões profundas devido à dificuldade de limpar o seu interior.
- o empenamento que ocorre durante a solidificação pode ser diminuído evitando mudanças bruscas da seção transversal, em áreas planas e largas, deve-se usar nervuras (reforçadores).
- o empenamento que ocorre durante o tratamento térmico pode ser diminuído selecionando ligas que requeiram têmperas mais brandas.

Tabela 3.2 – Tipo de Metal x Contração Linear [Bralla, 1986], [Resende, 1992]
e [Beeley, 1972].

Metal	Contração %
Ferro fundido Cinzento	0.83 – 1.3
Ferro fundido Branco	2.1
Ferro fundido Maleável	0.78 – 1.0
Bronze Duro	1.0 – 1.6
Bronze/ Alumínio	2.1
Bronze/ Magnésio	2.1
Ligas de Magnésio	1.3
Ligas de Alumínio	1.3
Níquel	4.5
Estanho	2.3
Chumbo	3.5
Zinco	4.7
Cobre	4.2
Latão Amarelo	1.3 – 1.6
Aço Fundido	2,0

Tabela 3.3 - Orientações para usinagem de acabamento no modelo [Bralla, 1986].

		Sobremetal necessário (mm)	
Ligas Fundidas	Dimensões do Modelo (cm)	Orifício	Superfície
Ferro Fundido Cinzento	Até 15,2	3,16	2,38
	De 15,2 a 30,5	3,16	3,16
	De 30,5 a 50,8	4,76	4,0
	De 50,8 a 91,4	6,35	4,76
	De 91,4 a 152,4	7,94	4,76
	Acima de 304,8	Instrução especial	Instrução especial
Ferro Fundido Maleável	Até 15,3	1,59	1,59
	Acima de 91,44	Instrução especial	Instrução especial
Ferro Fundido Nodular	Até 15,2	2,3	6
	De 15,2 a 30,5	3	10
	De 30,5 a 50,8	5	19
	De 50,8 a 91,4	6	19
	De 91,4 a 152,4	8	25
	Acima de 304,8	Instrução especial	Instrução especial
Aço Fundido	Até 15,2	3,18	3,18
	De 15,2 a 30,5	6,35	4,76
	De 30,5 a 50,8	6,35	6,35
	De 50,8 a 91,4	7,14	6,35
	De 91,4 a 152,4	7,94	6,35
	Acima de 304,8	Instrução especial	Instrução especial
Não Ferrosos: Bronze, Latão, Alumínio	Até 7,6	1,59	1,59
	De 7,6 a 20,3	2,38	2,38
	De 20,3 a 30,5	2,38	3,18
	De 30,5 a 50,8	3,18	3,18
	De 50,8 a 91,4	3,18	4,0
	De 91,4 a 152,4	4,0	4,76
	Acima de 152,4	Instrução especial	Instrução especial

Informações de processo:

As informações de processo visam auxiliar o projetista no projeto da peça, de modo que sua fabricação seja fácil e precisa. Abaixo são mostradas algumas informações a respeito do processo:

- construir um pequeno modelo se a peça for complexa.
- determinar o processo mais viável e os processos opcionais (escolha do processo de fundição).
- escolher um material que não necessite ou que necessite de um mínimo de tratamento térmico desde que sua escolha seja que as propriedades requeridas possam, praticamente, ser obtidas diretamente da fundição.
- tratamento térmico causa distorção conforme sua severidade e o número de vezes que é aplicado.
- macho é geralmente uma parte separada do molde, que é fixada à cavidade do mesmo para formar peças com rebaixos, furos, cavidades, tanto internas como externas. A maioria dos machos são feitos com areia shell mold (resina), formando uma estrutura sólida e frágil, a qual se desmancha com a solidificação.
- deve-se sempre que possível, evitar o uso de machos, pois estes causam uma tolerância adicional, um custo mais alto e maior probabilidade de falhas de fundição. Alguns meios para evitar o uso de machos estão ilustrado na figura 3.7.
- não devem ser usados machos para furos de diâmetro menor que 6,4mm ou maior que a metade da espessura da peça. Veja as limitações dimensionais de machos na tabela 3.4.
- de preferência, os machos devem ter formas simples possível.

Figura 3.7 - Modos de evitar machos [MED, 1988].

Tabela 3.4 : Limitações de machos [Bralla, 1986].

Material Da Peça Fundida	Macho Suportado nas 2 Extremidades				Macho Suportado em 1 Extremidade			
	Comprimento		Diâmetro		Comprimento		Diâmetro	
	[mm]		[mm]		[mm]		[mm]	
	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN
Alumínio	2438	7	3048	7	1828	7	1219	7
Magnésio	1524	7	304	7	457	7	609	26
Aço	2133	7	1524	7	1219	26	304	7

- o ângulo de saída deve ser no máximo de $0,017 \text{ rad. (1}^\circ\text{)}$ tanto para superfícies internas como externas como ilustra a figura 3.8. Para modelos de grande profundidade com paredes finas, permite-se ângulos de saída de $0,035 \text{ rad. (2}^\circ\text{)}$. Estes valores apresentados para o ângulo de saída são para qualquer processo de fundição. Para a fundição em areia é melhor consultar a tabela 3.1 que trás valores com maior precisão.

- ângulos de saída naturais podem ser aproveitados pela escolha adequada da linha de partição. Isto dispensa a adição de superfícies de saída reduzindo assim o peso.

Figura 3.8 - Ângulo de saída [MED, 1988].

Um exemplo do conteúdo colocado no guia pode ser visto na figura 3.9, que mostra furos obtidos na geometria da peça que podem ser feitos no processo de fundição através da colocação de machos nos moldes ou perfurados depois da peça pronta. Recomenda-se furar a peça depois de pronta quando o diâmetro do furo for inferior a 19mm. Estas orientações apresentadas na figura 3.9 servem de orientação para as dimensões mínimas de furos, para o processo de fundição em areia.

Em resumo, os furos são obtidos pela colocação de machos nos moldes e às vezes é preferível perfurar depois da peça pronta, se sua localização for muito precisa em relação a outras superfícies das peças.

A seguir serão mencionadas algumas informações básicas de projeto válidas para os três principais processos de fundição (fundição em: areia, molde permanente e sob pressão). Informações gerais:

- as tolerâncias devem ser a mais liberais possíveis. Indique tolerâncias fechadas somente onde necessárias.

Considerações sobre formas e proporções se referem a: espessura na seção; raio de borda; raios de extremidade; seções adjacentes; proporções das nervuras; junções

afiladas; saliência com bossas ou ressaltos; inserções; ressaltos para assentamento de cabeça de prendedor; acréscimo de material para reforço.

Para qualquer material usado neste processo. As dimensões mínimas dos furos são:	4,76 até 6,35 mm
Diâmetro Mínimo do Furo pode ser obtido também através da formula ao lado.	$D = 1/2t$ onde: D= diâmetro do macho > .35 T= espessura das secção [mm]
<p>RECOMENDAÇÕES</p> <p>Furos menores que 19 mm de diâmetro: são mais baratos furar a peça depois pronta.</p>	<p>ILUSTRAÇÃO</p> <p>É melhor fazer uma pequena marcação quando o furo for menor que 19 mm.</p>

Figura 3.9 – Orientações para furo [Bralla, 1986].

Estas informações sobre formas e proporções já foram mencionadas anteriormente e servem para os três principais processos de fundição. É através destas informações básicas que o projetista deverá projetar uma peça fundida, de modo que esta não apresente problemas depois de pronta.

Alguns exemplos deste guia são apresentados nas figuras que se seguem. A figura 3.10 mostra as diferentes formas de concordância para uma peça fundida no processo de fundição em areia. O modo errado apresenta um exemplo de variação brusca da seção (evitar), o modo correto apresenta um exemplo de seção regularmente crescente (optar).

Outro fator a ser considerado em peças de concordância em prolongamento seria o resfriamento direcional que é obtido por diferentes modos (Sobreesspessuras de alimentação e massalotes). Sobreesspessuras de alimentação é conseguida através da uniformidade na espessura de parede (da maior seção até a menor seção). Já os massalotes devem ter ponto de alimentação certo, ou seja alimentar a peça através da maior seção. Outros métodos usuais para a obtenção de solidificação direcional incluem: o uso de canais e alimentadores capazes de estabelecerem gradientes de

<p>RECOMENDAÇÃO</p> <p>Deve-se evitar a seguintes formas:</p> <p><i>Modo errado</i> do traçado</p> <p>1- mau</p> <p>2- regular</p>	<p>ILUSTRAÇÃO</p> <p>Deve-se optar pelas seguintes formas:</p> <p><i>Modo correto</i> do traçado</p> <p>3- bom</p> <p>4- 5- muito bom</p>
<p>Maneira correta de traçar uma concordância progressiva:</p> <p>- o sobremetal deve ser previsto em função da distância de alimentação que deve ser de 4,5 vezes a espessura da parede.</p>	<p><i>Modo correto</i> do traçado de uma concordância progressiva.</p>

Figura 3.10 - Concordância em Prolongamento [SENAI,1987].

temperaturas favoráveis; o uso de vazões de vazamento e temperaturas favoráveis; o uso de resfriadores; e o uso de materiais no molde com características térmicas diferentes em algumas partes críticas do molde.

A espessura da peça deve ser tão regular quando possível e obedecer valores mínimos e máximos. A espessura mínima que a parede da peça deve ter para que não haja falta de metal no interior da peça a ser fundida no processo de fundição em areia é ilustrado na figura 3.11.

O exemplo apresentado na figura 3.11 ilustra a espessura máxima e a mínima para diferentes tipos de materiais que poderão ser fundido no processo de fundição em areia.

RECOMENDAÇÃO

As espessuras mínimas compatíveis com a obtenção de peças sem falta de metal são determinadas em função da maior dimensão da peça, da complexidade de suas formas e do processo de moldagem utilizado para obtê-la.

Nota: é necessário incluir as sobreespessuras (sobremetal) de usinagem nos valores do diagrama ao lado.

ILUSTRAÇÃO

Diagrama empregado para peças de traçado simples, nem muito largas nem muito estreitas.

Valores recomendados para a espessura máxima e mínima de modo a evitar a falta de metal na peça

Material	Espessura Mínima	Espessura máxima
Alumínio e sua ligas	5 mm	Não há limite
Cobre e suas ligas	2,5 mm	Não há limite
Ferro fundido cinzento	3,0 mm	Não há limite
Ferro maleável	4,0 mm	Não há limite
Magnésio e suas ligas	3,97 mm	Não há limite
Latão	3,5 mm	Não há limite

Figura 3.11 - Espessura Mínima/ Máxima [Bralla, 1986] e [SENAI, 1987].

A regularidade da parede da peça é a forma de se evitar mudanças bruscas entre as seções que é principal causador de ponto de concentração de tensão e formação de vazios devido à ponto quente (acúmulo de material).

3.3 GUIA DE AUXÍLIO AO PROJETISTA NO PROJETO DE PEÇAS FUNDIDAS EM AREIA

Moldagem em Areia

A moldagem em areia é o mais antigo, mais usado e econômico para produção de peças fundidas. É o processo em que o molde não pode ser reutilizado, pois é destruído na desmoldagem. A areia de fundição usada na confecção dos moldes é um material constituído de:

- uma substância granular refratária;
- elementos aglomerantes minerais e ou orgânicos;
- outros elementos (serragem, carvão etc.); nem sempre são empregados [Pereira, 1991].

Uma consideração importante é saber qual o tipo de moldagem em areia a ser utilizado. Os processos de moldagem em areia (Natural e Sintética) compreendem:

- a. Moldagem em areia - verde;
- b. Moldagem em areia cimento;
- c. Moldagem em areia de macho;
- d. Moldagem no processo CO₂.

A seguir serão descritos os tipos de processos de moldagem em areia:

O maior constituinte dos moldes de areia modernos é a sílica (SiO₂). Ela é obtida em depósitos naturais de areia, possuindo normalmente forma arredondada e tamanho relativamente uniforme. A quantidade de sílica é usualmente maior que 90%, com o restante constituído de ligantes de argila, bentonita e carvão.

A moldagem em areia cimento é o processo em que se mistura cimento (Portland) em aproximadamente 10% com areia, com um conteúdo de mistura elevado (até 5%). Esta mistura é usada para conformar as superfícies em contato com o modelo; a região restante é preenchida com areia recuperada.

A moldagem em areia de macho são basicamente as mesmas que as do molde, exceto que a elas são ainda adicionados óleo e palha de arroz para a obtenção de uma ligação mais forte que aumenta a consistência do molde. Uma areia típica de macho contém 1% de óleo de linhaça e 2% de amido.

A moldagem no processo CO₂, envolve uma mistura de areias silicosas sem argila, com até 5% de silicato de sódio, que é utilizada para conformar moldes e machos, que posteriormente são endurecidos pela ação do gás carbônico (CO₂) [Campos Filho, 1978].

Guia de Auxílio ao Projetista

O guia de auxílio ao projetista foi desenvolvido em um trabalho anterior do autor Silva, (1998).

Para formar o guia, a seguinte metodologia foi aplicada:

- reunir o conhecimento sobre o processo de fundição em areia.
- analisar e interpretar o conhecimento reunido de modo a isolar o conhecimento essencial para auxiliar o projetista no desenvolvimento do projeto do produto.
- formar um guia de orientações de projeto de modo a auxiliar o projetista de peças fundidas.

Como as orientações (figuras 3.2 a 3.11) se encontravam espalhadas, um guia de auxílio ao projetista de peças fundidas foi desenvolvido de modo a organizar estes tipos de informações, tornando-as mais fáceis de serem acessadas. Este guia fornece recomendações de projeto para a geometria da peça, para o fundido e para o processo de fundição em areia.

Maiores detalhes sobre o guia podem ser encontrados nas figuras 3.12 e 3.13 e no trabalho anterior do autor [Silva, 1998].

A figura 3.12 mostra a organização dos diversos tipos de informações, visando um fácil acesso com relação às orientações sobre a geometria da peça.

Figura 3.12 – Mostra o acesso às orientações sobre a geometria da peça.

A figura 3.13 mostra a organização dos diversos tipos de informações, visando um fácil acesso com relação às orientações sobre o fundido.

Figura 3.13 – Mostra o acesso às orientações sobre o fundido

Nas figuras 3.12 e 3.13 pode-se obter as informações sobre um determinado assunto, selecionando com o *mouse* no ponto desejado e outra tela irá aparecer trazendo mais informações.

Essas informações foram dispostas em uma estrutura de árvore com os diversos ramos acumulando informações importantes para o projeto da peça. Exemplos destes tipos de informações podem ser vistos nas figuras 3.2, 3.8, 3.10 e 3.11. No caso da figura 3.12, ao selecionar “concordância em prolongamento” o projetista obtém uma recomendação como mostrada na figura 3.10.

CONCLUSÃO

Neste capítulo foram apresentadas informações técnicas sobre as etapas de fabricação de peças fundidas, bem como recomendações de projeto para os processos de fundição em areia. O conhecimento apresentado neste capítulo tem por objetivo mostrar a importância de relacionar os fatores de fabricação que influenciam o processo de projeto. As informações coletadas passaram a ser chamadas de regras de projeto para a geometria da peça e para manufatura do fundido.

Na pesquisa bibliográfica realizada neste trabalho de pesquisa encontrou-se um volume grande de informações. Essas informações, muitas vezes conflitantes, se encontram espalhadas por dezenas de livros e outras publicações, o que dificulta sua utilização pelo projetista do produto.

Para auxiliar o projetista na obtenção destas informações de forma rápida e confiável este trabalho utilizará o guia de auxílio ao projetista para a formação de uma base de conhecimento sobre o processo de fundição em areia, ou seja, o conhecimento necessário para se ter um bom projeto de peças fundidas em areia. Com esta base de conhecimento, será possível criar um sistema computacional que possa integrar este conhecimento com um sistema CAD (PRO-E), como forma de auxílio ao projetista nas dificuldades encontradas no processo de projeto de fundidos.

CAPÍTULO 4 - SISTEMA DE AUXÍLIO AO PROJETISTA NO PROJETO DE PEÇAS FUNDIDAS (SEMF)

4.1 VISÃO GERAL DO SISTEMA

Por meio da informatização, pode-se facilitar o trabalho do projetista diminuindo o tempo de projeto e aumentando a qualidade final do projeto. Uma das formas de fazer isto é através de programas CAD, como por exemplo o PRO-E. O problema é que, só com o uso de programas CAD, não se consegue o resultado esperado, ou seja, reduzir o tempo de desenvolvimento do produto, porque o projetista não detém o conhecimento nas diversas áreas que constituem o projeto. Uma das formas de auxiliar o projetista na aquisição do conhecimento é a implementação de sistemas de auxílio ao projetista no projeto da peça, como proposto neste trabalho.

Com o objetivo de implementar um sistema de auxílio ao projetista no projeto de peças fundidas em areia, foi necessário conhecer os dados de saída do programa CAD (PRO-E) e criar um procedimento que reconhecesse nos dados de saída; as diversas *features* (características) da peça. Através da implementação de uma base de conhecimento, que contém regras para o projeto de uma peça fundida em areia; essas *features* puderam ser verificadas quantas as possíveis violações resultando em um diagnóstico sobre a peça sendo projetada. O sistema aqui proposto, visa servir de ligação entre o conhecimento do processo de fundição em areia e de ferramentas computacionais (PRO-E), para o desenvolvimento do projeto de uma peça fundida.

De acordo com a figura 4.1, o sistema proposto parte do projeto de uma peça a ser fundida em areia utilizando o programa (PRO-E), o qual gera um arquivo de saída, que é o arquivo *trail*. Este arquivo *trail* servirá para criar os procedimentos que vão checar possíveis violações de regras (processo de fundição em areia) no projeto da peça feita no PRO-E. Este sistema é formado por sete procedimentos. O primeiro procedimento determina o *trail* mais atual para dar sequência aos outros procedimentos. O segundo procedimento faz a leitura deste *trail* mais atual e o armazena na memória RAM do micro,

de uma maneira mais adequada para o reconhecimento das *features* (características). Após feita a leitura do *trail* mais atual, entra em operação o terceiro procedimento que faz o reconhecimento das *features* (*Solid*, *Cut*, *Hole*, etc.) desenhadas no PRO-E. Reconhecidas as *features*, entra em operação o quarto procedimento que arquiva estas *features* (reconhecidas) na memória RAM do micro, formando assim o banco de dados texto (APD/LR). Uma vez arquivada as *features* (banco de dados texto) entra em operação o quinto procedimento que é verificar a *feature*. A *feature* é verificada através do quinto procedimento (verificar *feature*) em conjunto com o sexto procedimento que é a base de conhecimento que contém regras sobre o processo de fundição em areia (como por exemplo, espessura mínima de parede). O sétimo e último procedimento (diagnóstico) pega os resultado da verificação e diz se aquela *feature* analisada sofre ou não violação quanto a regra verificada.

4.2 DETALHAMENTO DA CONCEPÇÃO DO SISTEMA

O sistema de auxílio ao projeto voltado para a manufatura de peças fundidas (SEMF), foi desenvolvido utilizando-se o programa *Borland Delphi 4*, que usa linguagem de programação em Pascal.

Entre as linguagens visuais que surgiram, nenhuma veio tão completa e acabada quanto o *Delphi*. O *Delphi* vem com um compilador capaz de gerar código diretamente executável pelo Windows, proporcionando uma velocidade de execução de 5 a 20 vezes maior que as linguagens interpretadas. Além disso, vem também com um gerenciador de banco de dados. O tempo de desenvolvimento de qualquer sistema é reduzido a uma fração de tempo que seria necessário usando outras linguagens e o resultado é superior.

Sob esta ferramenta de implementação, observa-se as seguintes vantagens:

- o *Delphi* tem ferramenta de desenvolvimento bem definida, como “*experts*” de aplicação e formulários;
- tempo de implementação reduzido e
- possibilidade de manipular informações qualitativas (textos, símbolos, declarações, etc.)

O sistema foi implementado segundo o fluxograma da figura 4.1. A seguir serão detalhados os procedimentos desenvolvidos para a implementação das atividades mostradas na figura 4.1.

Figura 4.1 – Fluxograma do SEMF

4.2.1 – DESENHAR PEÇA

O desenho da peça é feito pelo projetista no CAD (PRO-E) em 3D, seguindo alguns procedimentos que garantem que o SEMF irá analisar a peça de forma correta. Esses passos são importantes porque garantem que as informações apareçam no “*trail*” de modo compatível com a programação do SEMF (para um possível reconhecimento da *feature*). No PRO-E, deve-se seguir a seguinte ordem de desenho:

1. Iniciar um novo desenho no modo *parts*.
2. Criar 3 planos ortogonais.
3. Iniciar a construção de um “*Protrusion*”.
4. Escolher respectivamente os planos horizontal e vertical como referências.
5. Rascunhar um retângulo.
6. Modificar as cotas de referência.
7. Modificar as cotas que mostram as dimensões das linhas do rascunho.
8. Regenerar.
9. Clicar em “*Done*” para finalizar o rascunho.
10. Digitar o valor da cota de extrusão.
11. Clicar em “*Done*” para terminar o sólido.

Estes passos são melhores detalhados no capítulo 5, onde um exemplo foi montado com o objetivo de mostrar uma aplicação do sistema.

4.2.2 – DETERMINAÇÃO DO *TRAIL* ATUAL

O SEMF automaticamente trabalha sempre com o *trail* atual, porém há opção de entrar com o nome do arquivo (*trail* anterior). Para escolher o *trail* anterior é necessário um sistema de codificação para identificar qual *trail* está associado com o nome da peça. Um exemplo de associação seria nome da peça carcaça 50, isto significaria que o nome carcaça 50 está salvo no diretório de trabalho do PRO-E como *trail*. txt.50. Outra forma de escolher o *trail* anterior seria criar uma pasta que está vinculada com diretório de trabalho do PRO-E, ou seja, para cada peça (carcaça) que está sendo desenhada existe um pasta de trabalho que está associada ao *trail* desta peça. Este algoritmo só é aplicado para achar o último *trail* do desenho que está sendo realizado pelo projetista, ou seja, assume-se que a peça atual é a última peça do conjunto de peças já desenhadas no PRO-E.

A implementação desta atividade foi feita através da procura no diretório de trabalho pelo arquivo *trail.txt.1*, se for encontrado, o algoritmo procura pelo *trail.txt.2*, se não encontrar, o *trail.txt.1* é o último, caso contrário, ele procura pelo *trail.txt.3* e assim por diante. A figura 4.2 mostra o algoritmo utilizado para determinar o último *trail*.

Figura 4.2 – Algoritmo para localizar último *trail*.

Maiores detalhes sobre a implementação do *trail* pode ser visto no APÊNDICE 1.

4.2.3 – LEITURA DO TRAIL E RECONHECIMENTO DE FEATURES

Para manipular de maneira eficaz as informações do *trail*, dois tipos de bancos de dados foram implementados. Um armazena apenas caracteres alfanuméricos em forma de cadeias e o outro é um banco de dados típico de um sistema CAD. O primeiro banco de dados é um banco de dados texto, e vai armazenar, na memória RAM, o texto contido no arquivo *trail* do PRO-ENGINEER, de modo a tornar mais fácil e rápido o acesso às linhas de comandos. Um conjunto de procedimentos foi desenvolvido para comparar os textos, contidos no banco de dados (texto), com palavras que identificam as *features* da peça. Esses procedimentos implementam a Árvore Principal de Dados (APD), de acordo com as informações encontradas no banco de dados texto. As informações são colocadas na APD obedecendo a topologia da peça desenhada. Existem três procedimentos especiais que fazem esse tipo de tradução de informação. O primeiro procedimento procura pela palavra “*PROTRUSION*”. Essa palavra marca o início da construção de um sólido. O segundo procedimento procura pela palavra “*CUT*”, que marca o início da construção de um corte.

O terceiro procedimento procura, inicialmente, pela palavra “*HOLE*”. Essa palavra mostra o início da construção de um furo de seção circular.

A figura 4.3 mostra uma peça contendo as 3 *features*: sólido, furo e chanfro. O segundo banco de dados, que armazena apenas caracteres alfanuméricos, funciona como o banco de dados de editores de texto ASCII. Ele armazena caracteres em forma de cadeias.

A implementação desse banco de dados é indispensável, porque no reconhecimento de *features* é preciso procurar várias palavras, escritas no *trail*, várias vezes e o acesso direto ao arquivo, em disco, seria muito lento. Esse banco de dados possibilita acessar o arquivo, em disco, apenas uma vez em cada conferência.

O procedimento que reconhece um sólido inicia procurando pela cadeia “*#PROTRUSION*”. Ao encontra-lá o procedimento verifica se o sólido foi terminado procurando pela cadeia “*#MODIFY*”, que confirma o término do *PROTRUSION*. A cadeia “*#MODIFY*”, representa o comando usado para iniciar a mudança das cotas. A seguir, o programa procura pela cadeia “*!%CP ENTER A NEW VALUE*”. Na frente desta cadeia aparece, entre colchetes, o valor antigo da cota que será modificada. A cadeia “*!%CP ENTER A NEW VALUE [389.841084161]*” é um exemplo. O valor da nova cota aparece logo após a cadeia “*!TR*”. Esse valor será armazenado para futuro uso pelo procedimento de verificação da *feature*. Esse procedimento assume que o usuário desenhou o seu sólido seguindo os passos definidos no item 4.2.1 (Desenhar Peça).

Figura 4.3 – Exemplo de informação: sólido com furo, chanfro e arredondamento.

A tabela 4.1, detalha o algoritmo utilizado para o reconhecimento das *features*.

Tabela 4.1 – Procedimento de reconhecimento das *features*.

Passo	Descrição
1	ler o arquivo <i>trail</i> e armazenar as informações;
2	entrar em operação o primeiro procedimento, que procura pela cadeia “# <i>PROTRUSION</i> ”; (construção do sólido);
3	verificar se o sólido foi terminado procurando pela cadeia “ <i>MODIFY</i> ”, que confirma o término do “ <i>PROTRUSION</i> ”;
4	procurar pela cadeia “!%CP ENTER A NEW VALUE” [389.841] e armazenar o valor da cota;
5	entrar em operação o segundo procedimento que procura pela cadeia “ <i>CUT</i> ” (construção de um corte);
6	verificar se o corte foi terminado procurando pela cadeia <i>MODIFY</i> ” que confirma o término do “ <i>CUT</i> ”;
7	idem do passo 4
8	entrar em operação o terceiro procedimento que procura pela cadeia “ <i>HOLE</i> ” (construção de furo);
9	Verificar se o furo foi terminado procurando pela cadeia “ <i>MODIFY</i> ” que confirma o término do “ <i>HOLE</i> ”;
10	Idem do passo 4
O ciclo termina depois que todas as <i>features</i> forem reconhecidas e armazenadas no banco de dados.	

Uma descrição mais detalhada da implementação do procedimento de reconhecimento de *features* é dado a partir do passo 26 do APÊNDICE 1.

4.2.4 – ÁRVORE PRINCIPAL DE DADOS (APD) E LISTA DE REFERÊNCIA (LR)

A árvore principal de dados e a lista de referência são criadas, na memória RAM, durante a fase de reconhecimento das *features*, formando um banco de dados tipo texto contendo as informações sobre as *features* da peça como mostrado na figura 4.4.

Figura 4.4- Exemplo de como as informações de um projeto estão dispostas na APD/LR, (típico *model tree* semelhante de um sistema CAD).

Dois conceitos foram usados para o desenvolvimento do banco de dados principal: UNIDADE DE INFORMAÇÃO E TOPOLOGIA.

A UNIDADE DE INFORMAÇÃO –

A base de dados do SEMF foi projetada de modo a satisfazer as necessidades de armazenar informações geométricas, topológicas e informações auxiliares sobre os objetos desenhados no CAD. Escolheram-se 3 tipos básicos de informações para representar a geometria do objeto e suas características adicionais: número inteiro, número real e ponto tridimensional. Assim, a geometria do objeto é representada como um conjunto de números inteiros, números reais e pontos tridimensionais. Essas informações ficam dispostas em forma de listas. Cada elemento do banco de dados possui uma lista de números inteiros, números reais e pontos tridimensionais. O SEMF foi criado para tratar exclusivamente com projetos de fundição. O elemento do banco de dados possui dois subgrupos de informações. Um deles se chama fundido, porque armazena as informações do projeto referentes especificamente ao processo de fundição. O outro subgrupo se chama geometria, porque armazena informações referentes à geometria do objeto.

A figura 4.5, mostra o esquema da unidade de informação, que é usada na APD e como os campos se relacionam. Dentro do campo “Dado” pode-se encontrar os campos: Identificação, Tipo e Dados. Dentro do campo “Dados” pode-se encontrar Geometria e Fundido. Dentro dos campos Geometria e Fundido encontram-se os 3 campos: pontos, reais e inteiros. O registro ponto é formado por coordenadas (x, y, z).

Figura 4.5 – Unidade de informação (UI) da APD na forma de diagrama de blocos

Na figura 4.5, o campo Identificação é um registro composto de um campo cadeia chamado de ID e outro inteiro chamado de Pos. O ID é a identificação da *feature*, não existem duas com o mesmo ID. O Pos indica a ordem em que a *feature* foi cadastrada no banco de dados. O campo Tipo indica as *features* furo, chanfro, etc.

A seguir serão mostrados as definições de termos usados na figura 4.5. Estas definições são:

- **Pontos** – este campo é uma fila de pontos. Um ponto é um conjunto de três variáveis reais (x, y, z) formando as coordenadas espaciais.
- **Reais** – este campo é uma fila de variáveis reais.
- **Inteiros** – este campo é uma fila de variáveis inteiras.
- **Geometria** – geometria é um registro formado por uma lista de variáveis reais, outra de variáveis inteiras e outra de pontos.
- **Fundido** – é um registro idêntico à geometria.
- **Dados** – é um registro composto pelos campos **Geometria** e **Fundido**.
- **Tipo** – é uma variável inteira.
- **Identificação** – é um registro formado por uma variável do tipo cadeia chamada **ID** e outra variável inteira chamada de **Pos**.
- **Dado** – é um registro composto pelos campos **Identificação**, **Tipos** e **Dados**.
- **Abaixo e Afrente** – são ponteiros que apontam para outras UI's.

Além dos conjuntos de informações (geometria e fundido) a unidade de informação (UI), possui também dois campos ponteiro. Um desses campos vai apontar para a unidade

de informação de mesmo nível hierárquico. O outro vai apontar para a unidade de nível hierárquico imediatamente inferior. Isso organiza a topologia do banco de dados.

Cada uma dessas unidades possui também um campo de cadeia chamado de ID. Esse campo funciona como uma identificação para a unidade de informação e comporta até 200 caracteres.

TOPOLOGIA –

O ponteiro que aponta para a unidade de mesmo nível hierárquico recebeu o nome de ponteiro **Afrente**. O que aponta para a unidade de nível inferior chama-se **Abaixo**. Esses dois ponteiros garantem a organização do banco de dados quanto à topologia. Existem dois procedimentos para incrementar o banco de dados. Um deles se chama **inicia** e tem a seguinte sintaxe: **Inicia (UI)**, onde UI é uma unidade do banco de dados. Esse procedimento determina a UI como sendo a primeira unidade de informação colocada no banco de dados. O outro procedimento relacionado com o incremento do banco de dados é o procedimento **Iabaixo** que insere uma nova unidade de informação no banco de dados fazendo com que o ponteiro **Abaixo** de uma outra unidade aponte para essa nova unidade. Sua sintaxe é: **Iabaixo (id, unidade)** onde “id” é uma cadeia que vai informar qual unidade vai ter seu ponteiro abaixo apontado para esta nova unidade, sendo esta a nova unidade a ser colocada no banco de dados. Na construção do banco de dados o procedimento **inicia** é usado apenas uma vez, para armazenar a primeira unidade. Depois de armazenada a primeira unidade apenas o procedimento **Iabaixo** é utilizado. Assim, tem-se uma unidade, no “topo” do banco de dados, acompanhada das outras ligadas a ela, através de seu ponteiro **Abaixo**. A estrutura formada pela possível disposição das unidades de informação tem o formato de uma árvore binária, por isso, o banco de dados principal recebeu o nome de “Árvore Principal de Dados”, que será tratada daqui para frente como APD.

A figura 4.6 mostra o esquema do nó da APD. “Dado” é o conjunto de informações úteis para o SEMF. **Afrente** é o ponteiro que aponta para a informação de mesma topologia. **Abaixo** é o ponteiro que aponta para a informação de topologia imediatamente inferior.

A figura 4.6 mostra também os 3 campos principais do nó da APD. Os campos **Dados**, **Abaixo** e **Afrente**. O campo “Dados” contém informações sobre a *feature* que o nó

representa. Os campos **Afrente** e **Abaixo** são ponteiros que informam a localização de outro nó.

Figura 4.6 – Esquema do nó na APD (*model tree* semelhante de um sistema CAD).

No exemplo mostrado na figura 4.4 pode-se notar, acima do esquema da árvore, que existe uma fila. As informações dessa fila são ponteiros que apontam para os nós da APD. Essa fila é necessária para facilitar a localização do nó certo na APD, já que existem várias “ramificações” de informação. Os nós dessa fila funcionam como referências para os nós da APD, por isso ela recebeu o nome de *Lista de Referência* (LR). A lista de referência completa o banco de dados e será referido como APD/LR.

Na figura 4.4 representou-se um desenho formado por um sólido com um furo e um chanfro (disposição das informações do desenho na APD). O furo tem um arredondamento que por sua vez tem uma rugosidade. A figura 4.3 mostra o desenho com as características da peça, ou seja, no PRO-E.

As informações chamadas sólido, furo e chanfro são informações sobre a geometria da peça. Já a rugosidade não é uma informação visível em um CAD, e sim uma informação adicional. Essa informação será indispensável na manufatura da peça.

A implementação adotada utiliza a APD da seguinte forma: a primeira *feature* tem suas informações colocadas na primeira unidade de informação (UI) da APD através do procedimento **inicia**. Todas as outras unidades são colocadas abaixo dessa primeira através do procedimento **iabaixo**. Entretanto, o banco de dados do SEMF foi construído de forma a tornar flexível o modo de armazenamento dessas informações. Por isso a configuração atual pode ser modificada conforme a necessidade.

Maiores detalhes sobre a implementação da APD/LR são dados no APÊNDICE 1, do passo 1 ao passo 18.

4.2.5 – BASE DE CONHECIMENTO DO SEMF (REGRAS)

Com o objetivo de se obter peças fundidas de qualidade garantida com o mínimo de custos na sua confecção é necessário dar à peça uma forma apropriada, ou seja, prever uma manufatura facilitada. Isto é possível através da técnica de DFM com suas regras específicas (por exemplo: ângulo de desmolde, raio de concordância, espessura mínima de parede, diâmetro mínimo do furo, etc.) baseadas nos conhecimentos e experiências a respeito do processo de fundição. Alguns exemplos implementados são:

1. Concordância da forma com as condições de serviço:

Por fundição podem-se obter peças de praticamente qualquer forma, devendo oferecer a máxima resistência às solicitações a que estejam sujeitas, observando-se:

- Escolha da seção transversal mais apropriada, de acordo com o tipo de solicitação a qual a peça estará sujeita e o material selecionado;
- Evitar entalhes, cantos vivos, mudanças bruscas de seção, furo e outras discontinuidades, nos quais aparecem pontos de tensão capazes de provocar a ruptura prematura da peça.

2. Adaptação da forma levando em conta as peculiaridades da fundição como:

- Paredes de espessura uniforme e mínima de 3.0mm;
- Paredes internas de 0.7 a 0.9 da espessura externa;
- Evitar ângulos vivos usando raios de concordância;
- Evitar variações bruscas de espessuras;
- Evitar superfícies horizontais muito extensas;
- Evitar reentrâncias e encaixes;
- Adotar inclinação nas paredes para facilitar o desmoldagem

Para implementar a base de conhecimento foi necessário a criação de classes (com procedimentos) que são capazes de percorrer a APD e achar a *feature* cadastrada e aplicar as regras de fundição.

Através de um conjunto de procedimentos que usam as informações contidas na APD como entradas para uma tabela de decisão (tabela 4.2), pode-se criar a base de conhecimento. As informações contidas na APD são do tipo: sólido, furo e corte sendo estes representados por UI's. Através destas UI's é possível fazer a checagem e a aplicação das regras, utilizando um procedimento que busca na APD uma das três *features* (UI). Uma vez escolhida a *feature*, entra em operação um procedimento que faz, por exemplo, a comparação entre a espessura da parede atual (desenhada) com a recomendada. Se for constatado que existe alguma dimensão menor que a mínima recomendada, o SEMF exibe mensagens de advertência.

O SEMF possui dois procedimentos: um chamado de “checatudo”, ou seja, executa todos os procedimentos que aplicam as regras em todas as *features*, e o outro procedimento que verifica uma *feature* por vez, ou seja, a escolhida na tela principal do SEMF.

Alguns procedimentos de regra precisam de um número que indica se a peça é de alumínio ou ferro fundido. Um procedimento é responsável por aplicar uma regra em função de uma dada ação selecionada. Ou seja, para cada tipo de *feature*, reconhecível pelo SEMF, existe um número de descrições de condição e ação. A tabela 4.2 mostra um exemplo de uma tabela de decisão onde ação (Geometria ou Fundido) seria sólido, furo e corte e condição (regras) seria diâmetro mínimo do furo, espessura mínima de parede, e diâmetro menor que 19mm.

Tabela 4.2 - Tabela de decisão

CONDIÇÃO	AÇÃO		
	SÓLIDO	FURO	CUT
Espessura de Parede	X	X	X
Diâmetro Mínimo Furo	NA	X	X
Dimensão/Diâmetro < 19mm	NA	X	X

Nota: - NA = não se aplica esta condição;

- X = mostra a ação que será executada naquela coluna.

No caso do sólido, a tabela 4.2 mostra que, para a ação sólido existe uma condição que se aplica, que é a espessura de parede. As demais condições não se aplicam.

Atualmente, a base de conhecimento do SEMF é formada por regras para:

1. Paralelepípedo Sólido: possui apenas uma regra, que é espessura mínima de parede.
2. Furo Cilíndrico - possui três regras existentes: diâmetro mínimo do furo, espessura mínima de parede e diâmetro menor que 19mm.
3. Corte (CUT) - possui duas regras existentes: diâmetro mínimo do furo, espessura mínima de parede e dimensão menor 19mm.

EXEMPLOS DE REGRAS

A montagem de regras numa base de conhecimento é referente ao domínio específico representado neste programa. A seguir são apresentados exemplos de tópicos de conhecimento transformados em regras:

REGRA 1. Para material ferro fundido, espessura mínima é 3.0mm.

SE material é ferro fundido

E espessura da parede $\leq$ espessura mínima

ENTÃO exibir mensagem de violação de regra

REGRA 2. Furar a peça depois de pronta, furo menores que 19mm,

SE existe furo

E diâmetro do furo $\leq$ 19mm

ENTÃO exibir mensagem sugerindo fazer furo depois da fundição.

REGRA 3. Não é aconselhável construir no processo de fundição em areia, diâmetro $<$ que 6.35mm.

SE existe furo

E diâmetro furo $<$ diâmetro mínimo

ENTÃO exibir mensagem de violação de regra.

Outras regras (quantitativas) que podem ser representadas desta forma:

- Raio de concordância ($r = E_p$ ou $r/E_p = 1$);

- Distância de alimentação ($4.5 \times E_p$);
- Acúmulo de material ($d = 1.25 \times E_p$)
- Concordância em prolongamento (concordância progressiva- $R=1.5 \times E_p$, declividade de 15% 8.32°) pode ser visto em detalhes no cap.2 orientações e
- Rugosidade /Tolerância ($100 - 1000 \text{ RMS}^*$) / (ferro fundido = 3mm/m, alumínio = 2mm/m, etc.) * RMS = unidade de rugosidade expressa em milésimos de polegada (micro – inches).

REGRAS DISPONÍVEIS NO SEMF

1) REGRA PARA SÓLIDO –

Para encontrar a espessura de parede de um paralelepípedo é necessário conhecer as suas três dimensões. Essa informação existe na APD. Se alguma dessas dimensões for inferior ao valor mínimo recomendado para o material usado uma mensagem de alerta será exibida. Caso contrário a mensagem de não violações será exibida.

Para verificar as regras de parede mínima para o prisma sólido e cilindro sólido, o procedimento “CHECKPAREDEMIN” recebe um argumento, que representa o material usado (1 – Alumínio, 2 – Ferro fundido). Esse procedimento percorre os nós da LR até encontrar um que aponte para a *feature* de tipo 1 (prisma sólido) e ID igual ao nome selecionado pelo usuário. Encontrada a *feature*, esse procedimento verifica as suas dimensões (campo/reais/inteiro/pontos) e, se existir um valor menor que o mínimo recomendado, é exibido o texto de recomendação. Se não, o procedimento exibe um texto dizendo que não há violações. O procedimento “CHECKPAREDEMIN” verifica erros de espessura de parede em um sólido determinado. O procedimento “TCHECKPAREDEMIN” verifica erros de espessura de parede em todos os sólidos da APD. O procedimento “CHECHPAREDEMINE” verifica erros de espessura de parede em um sólido cilíndrico determinado. O procedimento “TCHECHPAREDEMINE” verifica erros de espessura de parede em todos os sólidos cilíndricos da APD.

As regras para este caso são detalhadas a seguir:

A) Sólido – Material do tipo alumínio e suas ligas

A regra diz : o valor recomendado para espessura mínima (alumínio e suas ligas) de modo a evitar a falta de metal na peça, é de 5mm.

B) Sólido – Material do tipo ferro fundido cinzento

A regra diz: o valor recomendado para espessura mínima (ferro fundido cinzento) de modo a evitar a falta de metal na peça, é de 3mm.

2) REGRA PARA FURO –

É semelhante ao sólido. Lembrando que para o furo são duas (regras): diâmetro mínimo do furo e espessura mínima de parede. E uma recomendação: diâmetro menor que 19mm.

A) Furo – diâmetro mínimo

A regra diz: o diâmetro mínimo de furos que podem ser construídos no processo de fundição em areia, para qualquer material, devem ser maiores que 6.35mm.

B) Furo – diâmetro menor que 19mm

A regra diz: para furos menores que 19 mm de diâmetro é mais barato furar a peça depois de pronta.

3) REGRA PARA CORTE (CUT) É SEMELHANTE AO SÓLIDO E AO FURO –

Esse procedimento vai enviar uma mensagem para o usuário, fornecendo o diagnóstico de corte feito no paralelepípedo (sólido). Lembrando que são duas: dimensão mínima do corte e a espessura mínima de parede.

Corte – dimensão mínima menor que 19mm

A regra diz: se a dimensão do corte (largura) for menor que 19mm é mais barato fazer este corte usando outros processos de fresamento/furação.

EDITORES DE PREFERÊNCIAS E REGRAS –

O editor de preferência e de regras servem para personalização dos textos enviados ao usuário através mensagens. Essas mensagens exibidas pelo SEMF podem ser editadas usando-se um editor de texto existente no programa. Para encontrá-lo basta clicar em arquivo—textos—regras (ou preferências).

4.2.6 - VERIFICAR *FEATURE*

A atividade “verificar *feature*” tem a função de buscar na APD/LR as *features* reconhecíveis e aplicar o conhecimento disponível (base de conhecimento - regras). Na tabela 4.3 estão detalhadas as várias opções implementadas.

Tabela 4.3 - Opções de verificação de *features* no SEMF:

OPÇÕES	VERIFICAR
1	Verificar se o diâmetro do furo é menor que o recomendado (6.35mm)
2	Verificar se o diâmetro do furo é menor que 19 mm.
3	Verificar se a espessura de parede deixada, no sólido, pelo furo é menor que a recomendada (alumínio - 5mm e ferro fundido – 3mm).
4	Existem procedimentos que verificam todas as <i>features</i> de uma vez, por exemplo (“TCHECKPAREDEMIN”). Os procedimentos que formam a base de conhecimento começado pela letra T (“TCHECKPAREDEMIN”) aplicam as regras em todas as <i>features</i> de uma vez.
5	Através da verificação da <i>feature</i> escolhida, o SEMF exibe as regras disponíveis para a <i>feature</i> .

A figura 4.7 mostra o desenho de uma peça feita no PRO-E, para uma possível verificação no SEMF (espessura mínima de parede). Esta peça é composta de 3 *features*, são elas: prisma sólido, furo cilíndrico e furo retangular.

As figuras 4.8 e 4.9 possuem o mesmo significado e demonstram como encontrar a espessura de parede, de acordo com as cotas horizontais, deixada por um furo de base retangular no sólido.

Figura 4.7 - Desenho feito no PRO-ENGINEER para um exemplo de checagem no SEMF.

Figura 4.8 - Como encontra as espessuras de parede através da cotas e *datum plane*.

Figura 4.9 – Determinação da espessura de parede.

A espessura do lado do *datum plane* (entre o plano e o furo) pode ser facilmente encontrada por $AX - BX$. A espessura do lado oposto ao *datum plane* (do outro lado do furo) é dada por $COTAHB + BX - (COTAHA + AX)$. Esses valores ficam disponíveis na APD. AX e $COTAHA$ são armazenadas na UI correspondente ao furo. BX e $COTAHB$ são armazenadas na UI correspondente ao sólido.

Para se encontrar as espessuras de parede, na direção vertical, o procedimento é análogo. A figura 4.10 mostra como encontrar a espessura de parede, para o furo retangular, usando-se as cotas verticais.

Figura 4.10 – Esquema mostrando as dimensões para cálculo da regra espessura mínima de parede.

A espessura do lado do *datum plane* (entre o plano e o furo) pode ser facilmente encontrada pela relação $AY - BY$. A espessura do lado oposto ao *datum plane* (do outro lado do furo) é dada por $COTAVB + BY - (COTAVA + AY)$. Esses valores ficam disponíveis na APD. AY e $COTAVA$ são armazenadas na UI correspondente ao furo. BY e $COTAVB$ são armazenadas na UI correspondente ao sólido.

A verificação da espessura de parede para um furo cilíndrico é análoga à de um furo retangular.

Maiores detalhes sobre a implementação da base de conhecimento/regras podem ser vistos no APÊNDICE 1, nos passos 35 a 44 .

CONCLUSÃO

Neste capítulo foram mostrados dois tópicos: concepção do sistema e sua implementação. Na concepção foram mostrados o fluxo de atividades, que começa com: desenhar peça, determinar o *trail* atual, reconhecimento de *features*, APD/LR, base de conhecimento e verificar *features*, e termina com o diagnóstico quanto a possíveis violações de projeto de uma peça que será manufaturada no processo de fundição em areia. Em seguida é mostrado a implementação do sistema, ou seja, o detalhamento de cada uma das atividades que constituem o SEMF.

CAPÍTULO 5 – APRESENTAÇÃO DO SISTEMA E EXEMPLO DE APLICAÇÃO

5.1 SISTEMA DE AUXÍLIO AO PROJETISTA DE PEÇAS FUNDIDAS - SEMF

O SEMF é dotado de alguns formulários gráficos que são usados para enviar informações ao usuário e também “pedir” informações a ele. Esses formulários formam a parte do SEMF que consiste na interface com usuário.

A figura 5.1 mostra a janela principal do SEMF. Com as principais informações e opções disponíveis ao usuário. Maiores detalhes dos itens de 1 a 11 serão mostrados a seguir.

Figura 5.1 – Área de trabalho do SEMF.

1 - Botão carregar –

Ao clicar nesse botão o usuário dá um comando ao SEMF para carregar o arquivo *trail* para o banco de dados de texto, reconhecer as *features* desenhadas pelo usuário, aplicar regras e deixar os resultados disponíveis para o usuário. O usuário tem duas opções diferentes para carregar o arquivo. A primeira opção se chama “último *trail*” a segunda se chama “escolher”. A figura 5.2 mostra o botão “Carregar” da área de trabalho do SEMF. Se a primeira opção estiver selecionada o SEMF vai carregar o *trail* criado pelo último desenho feito pelo usuário. Se a segunda opção estiver selecionada o SEMF vai abrir uma outra janela onde o usuário poderá escolher o arquivo que será carregado.

Figura 5.2 – Botão carregar da área de trabalho do SEMF

Figura 5.3 – Botão carregar com a opção “Escolher” selecionada.

A figura 5.4 mostra a área caixa de diálogo usada para escolher um determinado arquivo *trail*. Essa caixa aparece quando se clica em “carregar” com a opção escolher selecionada, como mostra a figura 5.3. A caixa mostrada na figura 5.4 apresenta os vários *trails*, a partir do *trail.txt.1* que o PRO-E vai armazenando no diretório de trabalho ao desenhar as peças. O problema é como escolher o *trail* específico da peça que foi desenhada anteriormente, neste caso o usuário precisa de um sistema de codificação ou criar uma pasta de trabalho como mostrado no capítulo 4, item 4.2.2.

Figura 5.4 - Caixa de diálogo onde o usuário procura um *trail* específico.

2) Guia – Regras Qualitativas

O sistema fornece diversas regras qualitativas e quantitativas sobre a geometria e o processo de fundição que o usuário pode acessar, obtendo diversas informações adicionais.

Ao clicar em um dos textos (Fundido ou Geometria), ou na área dentro do retângulo o SEMF abre uma janela com o navegador de internet mostrado na figura 5.5.

Figura 5.5 – Janela do guia.

Essa janela permite acessar o guia do SEMF. Bastando selecionar uma das duas opções, para acessar as informações existentes na base de conhecimento. O sistema permite, nesta opção, importar textos e figuras de “sites”, relacionados com a área de fundição, objetivando aumentar a base de conhecimento.

Para adicionar as regras (conhecimento), o usuário deve seguir três passos:

1º Passo: digitar o texto (regra) no word, ou seja criar o arquivo;

2º Passo: salvar o texto em HTML, no diretório do SEMF (Guia);

3º Passo: criar o Hiperlink, ou seja vincular o arquivo com geometria ou fundido.

3) Pop

Quando a opção “Sim” está selecionada aparece, na janela do PRO-ENGINEER, um menu *pop up*. Isto permite o trabalho simultâneo do PRO-E com o SEMF.

4) Arquivo

Ao clicar no botão “Arquivo” aparece um menu *pop up*, com mostrado na figura 5.6. Na opção “Material” é possível selecionar ou modificar o tipo de material utilizado. Na opção texto como mostrado na figura 5.7 é possível modificar os textos que aparecem na janela do SEMF e os textos das regras contidos nos arquivos.

Figura 5.6 – Parte da área de trabalho do SEMF que contém os menus principais. O usuário clicou em “ARQUIVO”.

O editor de preferência e de regras servem para personalização dos textos enviados ao usuário através mensagens. Essas mensagens exibidas pelo SEMF podem ser editadas usando-se um editor de texto existente no programa. Para encontrá-lo basta clicar em arquivo—textos—regras (ou preferências)

Figura 5.7 - Área de trabalho do SEMF, quando se clica no botão “Arquivo” e seleciona a opção textos/regras.

Quando se clica na opção regras, como mostra a figura 5.7 aparece a janela como mostra a figura 5.8. Esta janela é um editor onde o usuário vai modificar os textos exibidos pelo SEMF como resultado de checagem.

Figura 5.8 – Editor de texto para as regras.

A aparência do editor das preferências é mostrado na figura 5.9. Uma vez que o texto é mudado ele é salvo ao fechar a tela.

Figura 5.9 - Editor de preferência do SEMF.

5) O botão “Ajuda”

Esse botão abrir uma janela de ajuda padrão do tipo Windows, porém não foi ainda implementado integralmente.

6) O botão CAD/CAE

Abre um menu *pop up* com o qual se pode carregar o PRO-ENGINEER ou o Ansys. A identificação do diretório deve ser colocado nos arquivos de preferências do SEMF, como mostrado na figura 5.10.

Figura 5.10 – Mostra o menu “pull down” que aparece quando se clica em “CAD/CAE”

7) Opções de checagem

Quando o usuário escolhe o nome de uma *feature*, que aparece na área 11, surge na área 7 as regras que podem ser aplicadas na *feature* em questão. Quando se dá um duplo clique em uma regra na área 7, aparece na área 8 uma mensagem de ocorrência ou não da violação da regra escolhida.

A figura 5.11 ilustra um exemplo, onde não houve violação da regra.

Figura 5.11 – Área de trabalho do SEMF com um texto de não violações nos resultados de checagem.

8) Editor de texto usado para exibir os resultados da checagem.

9) Cancelar e Todos

O botão “todos” manda que o SEMF verifique todas as *features* do desenho automaticamente e exiba todos os resultados de violações. O botão “cancela” limpa todos os bancos de dados do SEMF, deixando-o pronto para abrir um novo desenho.

Figura 5.12 - Botão “cancela” e botão “todos” da área de trabalho do SEMF.

10) Checar

A figura 5.13 mostra, com maiores detalhes o local onde o usuário escolhe entre ver violações ou recomendações sobre sua peça. Na área 11, aparecem as *features* pertencentes à peça desenhada. Nesta hora o usuário pode checar uma *feature* ou acessar uma recomendação relacionada à *feature*.

Figura 5.13 – Local onde o usuário escolhe entre ver violações ou recomendações.

5.2 EXEMPLO DE APLICAÇÃO

A seguir serão descritos os passos (figuras 5.15 a 5.73) na construção do exemplo da figura 5.14, utilizando-se os programas PRO-E e SEMF. Todas as mensagens do SEMF são para o material alumínio.

Figura 5.14 – Peça a ser construída e checada

1 - Iniciar o PRO-ENGINEER.

Ao executar o PRO-ENGINEER o usuário obtém a janela mostrada na figura 5.15.

Figura 5.15 – Área de trabalho do PRO-ENGINEER

- 2 - Iniciar um novo desenho através do menu “Arquivo” da área de trabalho do PRO-ENGINEER como mostrado na figura 5.16

Figura 5.16 – Parte da área de trabalho do PRO-ENGINEER e seleção da opção *new*.

Ao clicar na opção *new* o usuário abre a janela como mostrada na figura 5.17. Através dessa janela ele pode escolher o tipo e o subtipo de *feature* que vai desenhar. As opções que devem ser escolhidas para este exemplo são, opção tipo: *PART* e opção subtipo: *SOLID*.

Figura 5.17 – Janela onde o usuário pode escolher o tipo e o subtipo de desenho.

Ao solicitar um novo desenho o usuário obtém a janela como mostrada na figura 5.18.

Figura 5.20 – Parte do menu *manager* mostrando a opção “*Create*”.

3.3 - Clicar em “*Datum*”. Ao Clicar em *Create* o usuário abre a opção “*Datum*”, entre outras. Mostrado na figura 5.21.

Figura 5.21 – Parte do menu *manager* mostrando a opção “*Datum*” selecionada.

Ao clicar em *Datum* o usuário abre um novo menu mostrado na figura 5.22.

Figura 5.22 – Parte do menu *manager* mostrando as opções do menu “Datum”

3.4 - Clicar em *Plane*. Quando o usuário clica na opção “*Plane*” abre um menu com duas opções, como mostrado na figura 5.23. O usuário deve clicar na opção “*Default*”.

Figura 5.23 – Menu mostrando as opções de planos.

3.5 - Clicar em *Default*. Quando o usuário clicar na opção “*Default*”, mostrado na figura 5.23, o PRO-ENGINEER gera o 3 planos ortogonais mostrados na figura 5.24.

Parte do *trail* correspondente a criação dos três planos (figura 5.24) é mostrado abaixo.

```
#FEATURE
#CREATE
#DATUM
#PLANE
#DEFAULT
!%CIDATUM PLANE HAS BEEN CREATED SUCCESSFULLY.
#CREATE
#PROTRUSION
#DONE
#DONE
```

Figura 5.24 – Os 3 planos ortogonais chamados de “Datum Planes”.

4 - Criar o sólido de base. Primeiro o usuário deve desenhar o sólido maior, para depois desenhar as demais *features* neste sólido.

4.1 - Clicar em *Feature>Create* no menu *manager*, como mostra a figura 5.25.

Figura 5.25 – Menu *manager* mostrando a opção “Feat_Create”.

4.2 - Clicar em *Solid*. Ao clicar em *Create* o usuário abre o menu mostrado na figura 5.26. Nesse menu o usuário precisa clicar na opção “*Solid*”

Figura 5.26 – Menu *manager* mostrando a opção “*Solid*”.

4.3 - Clicar em *Protrusion*. Ao clicar em *solid*, como mostrado na figura 5.26 o usuário obtém o menu mostrado na figura 5.27, onde ele deverá clicar na opção “*Protrusion*”.

Figura 5.27 – Menu *manager* mostrando a opção “*Protrusion*”.

4.4 - Selecionar *Extrude* e *Solid* e clicar em *Done*. Ao Clicar em “*Protrusion*” o usuário abre o menu mostrada na figura 5.28. Nesse *menu* o usuário precisa selecionar as opções “*Solid*” e “*Extrude*” e em seguida clicar em *Done*.

Figura 5.28 – Menu *manager* mostrando as opções “Extrude, Solid e Done”.

4.5 - Escolher “One Side” e clicar em “Done”.

Figura 5.29 – Menu *manager* mostrando opções de desenho sólido

Ao clicar em done, no menu mostrado na figura 5.28, o usuário inicia o rascunho do sólido. Aparece então o menu mostrado na figura 5.29. Neste menu o usuário deve selecionar a opção “One Side” e Clicar em “Done”.

4.6 - Escolher o Plano de desenho. Ao clicar em “Done” do menu mostrado na figura 5.29 o PRO-ENGINEER pede ao usuário que escolha um dos 3 planos ortogonais, para desenhar o sólido, através do menu mostrado na figura 5.30.

Figura 5.30 – Menu onde o PRO-ENGINEER pede que o usuário escolha o plano de desenho.

O usuário deverá clicar em um dos 3 planos ortogonais para selecioná-lo, como por exemplo no DTM3.

4.7- Escolha do sentido do desenho. Após selecionar o plano de desenho, o usuário precisa escolher o sentido do desenho. Na área de trabalho do PRO-ENGINEER vai aparecer uma seta vermelha no plano selecionado como mostra a figura 5.31. Essa seta indicará o sentido positivo de elevação do desenho. Esse sentido poderá ser modificado clicando-se em “*Flip*” no menu *manager*, como mostra a figura 5.32.

Figura 5.31 – A seta no *datum plane* indica o sentido positivo de elevação do desenho

Figura 5.32 – Janela do menu *manager* mostrando a opção “*Flip*” que modifica o sentido do desenho

4.8 - Selecionar a vista do rascunho. Quando o usuário clica na opção *Okay*, do menu mostrado na figura 5.32, o PRO-ENGINEER pede ao usuário que selecione um plano para servir de referência para o desenho. Antes disso o PRO-ENGINEER pede que o usuário escolha a orientação desse plano de referência. Este comando é feito através do menu mostrado na figura 5.33.

Figura 5.33 – Menu *manager* mostrando as opções de referência para o desenho.

O usuário vai desenhar no primeiro plano selecionado por ele, mas o PRO-E precisa saber qual plano vai ficar na horizontal ou na vertical. Por isso o usuário precisa escolher *TOP*, para colocar o plano selecionado em cima do plano de desenho, *BOTTOM* para colocar o plano selecionado abaixo do plano de desenho, *RIGHT* para colocar o plano

selecionado à direita do plano de desenho ou *LEFT* para colocar o plano selecionado a esquerda do plano de desenho.

4.9 - Selecionar os planos como referência para a cota. Após o usuário selecionar o plano que determina a vista do desenho o PRO-ENGINEER modifica a posição dos planos e coloca o plano de desenho paralelo a tela do micro e coloca os outros dois planos perpendiculares a vista do usuário, como mostra a figura 5.34. As linhas vermelhas são os planos perpendiculares ao plano de desenho.

Figura 5.34 – Parte da área de trabalho do PRO-ENGINEER mostrando os 3 planos ortogonais.

Um deles, o dtm3, está paralelo à folha. Os outros dois, um está na horizontal (dtm2), e outro está na vertical (dtm1). Os dtm1 e dtm2 estão destacados na vertical.

4.10 - Selecionar os planos que serão referência para cotas. O PRO-ENGINEER pede ao usuário para indicar dois planos, perpendiculares entre si, para serem referência para as cotas. Os únicos planos do desenho são o DTM1 e o DTM2. O usuário precisa selecioná-los clicando neles. O SEMF espera que o plano horizontal seja o primeiro a ser selecionado. A figura 5.35 mostra a resposta do PRO-ENGINEER quando o usuário clica no plano horizontal.

Figura 5.35 – Mostra a linha pontilhada criada depois que o usuário selecionou o plano horizontal.

4.11 - Clicar em *Done Sel*. Após escolher os planos como referências para cota o usuário precisa clicar em *Done Sel*, como mostra a figura 5.36.

Figura 5.36 – Mostra a opção *Done Sel* selecionada no menu *manager*.

4.12 - Rascunhar um retângulo. O usuário deve clicar na palavra *rectangle*, que aparece no menu mostrado na figura 5.37, quando se clica em *Done Sel*. Com a opção *rectangle* selecionada o usuário deve dar dois cliques no primeiro quadrante formado por DTM1 E DTM2, (dois *datum plane*). Esses dois clique serão os dois vértices opostos do retângulo rascunhado.

Figura 5.37 – Mostra uma janela do menu *manager* onde o usuário pode escolher o tipo de rascunho que vai fazer. Para o exemplo em particular o usuário deve clicar em *rectangle*.

Quando o usuário der dois cliques na área de desenho com a opção *rectangle* selecionada o PRO-ENGINEER desenha um retângulo semelhante ao da figura 5.38.

Figura 5.38 – Retângulo rascunhado pelo usuário.

4.13 - Usar o comando *modify*. Após rascunhar o retângulo o usuário precisa usar o comando *modify*, mostrado na figura 5.39, para modificar as cotas do desenho.

Figura 5.39 – Parte do menu *manager* mostrando a opção *modify*.

Para modificar a cota o usuário precisa clicar na cota, que deseja mudar, e digitar o novo valor na caixa de texto que aparece. Essa caixa é mostrada na figura 5.40.

Figura 5.40 – Área de trabalho do PRO-ENGINEER. A seta mostra a caixa de texto que o usuário deve utilizar para digitar o novo valor da cota.

O SEMF espera que o usuário modifique as cotas na seguinte ordem:

1º - Cota de referência horizontal. É a cota que fica entre a linha vertical do retângulo e a linha vertical de referência. Esta ação está destacada na seta da figura 5.41.

Figura 5.41 – Rascunho do retângulo. A cota horizontal de referência está em destaque.

2º - Cota de referência vertical. É a cota que fica entre a linha horizontal do retângulo e a linha horizontal de referência. Está destacada pela seta na figura 5.42.

Figura 5.42 – Rascunho do retângulo. A cota vertical de referência está em destaque.

3° - Cota de horizontal. É a cota da linha horizontal do retângulo. Está destacada pela seta na figura 5.43.

Figura 5.43 – Rascunho do retângulo. A cota horizontal está em destaque.

4° - Cota de vertical. É a cota da linha vertical do retângulo. Está destacada pela seta na figura 5.44.

Depois de modificar as cotas o usuário precisa clicar em “Regenerate” para que o desenho aceite as novas cotas. Em seguida clicar em *done*, para finalizar o rascunho.

Figura 5.44 – Rascunho do retângulo. A cota vertical está em destaque.

4.14 - Selecionar a opção “*Blind*” e clicar em *done*. A opção *blind* permite que o usuário determine a cota da altura do sólido. A figura 5.45 mostra a parte do menu *manager* onde o usuário pode encontrar o comando *blind*.

Figura 5.45 – Parte do menu *manager* mostrando o comando “*Blind*”.

4.15 - Digitar o valor da altura. Quando o usuário clica em “Done”, no menu mostrado na figura 5.45, o PRO-ENGINEER pede que ele digite a cota da altura, como mostra a figura 5.46.

Figura 5.46 – A área de trabalho do PRO-ENGINEER. A seta mostra o campo onde o usuário deve digitar a cota da altura.

Após definir a cota da altura o usuário clica em “*Ok*”, no menu mostrado na figura 5.47, para finalizar a construção do sólido. O sólido aparece construído na área de desenho do PRO-ENGINEER. A figura 5.48 mostra o sólido pronto.

Figura 5.47 – Mostra a janela onde o usuário precisa clicar em OK para finalizar a construção do sólido.

Figura 5.48 – Área de trabalho do PRO-ENGINEER com o sólido pronto

5 - Utilizar o SEMF para procurar defeitos no sólido. Para localizar erros de fundição o usuário precisa iniciar o SEMF e clicar no botão “carregar” da área de trabalho do SEMF. A área de trabalho do SEMF é mostrada na figura 5.49.

Figura 5.49 – Área de trabalho do SEMF.

Em resposta ao clique no botão “carregar” o SEMF lê o *trail* do desenho e cria a APD. A figura 5.50 mostra a janela do SEMF exibindo as informações sobre o sólido do desenho.

Figura 5.50 – SEMF exibido a descrição do prisma sólido do desenho.

O usuário precisa dar um duplo clique no texto “Espessura mínima” para o SEMF exibir o resultado. Se nenhuma dimensão for menor que a mínima exigida para o processo de fundição a mensagem exibida será: não foram encontradas violações. Se não, a mensagem será: o valor recomendado para espessura mínima (alumínio e suas ligas) de modo a evitar a falta de metal na peça é de 5mm. A figura 5.51 mostra a exibição do resultado sem violação de regra.

Figura 5.51 – Resultado da checagem do prisma sólido sem violação.

6 - Desenhar um furo. Para desenhar um furo o usuário precisa acionar, no menu *manager*, a opção *Feature* e, em seguida, o clicar em *Create* e *Solid*. Isso irá exibir um menu com a opção *Hole*. Esse menu pode ser visto na figura 5.52.

Figura 5.52 – Comando *hole* selecionado no menu *manager*.

Após clicar no comando *Hole* o usuário precisa selecionar o comando *Straight* e clicar em *Done* para iniciar a construção do furo. A figura 5.53 mostra o menu que aparece quando se inicia a construção do furo.

Figura 5.53 – Menu *manager* com as opções de furo.

Na tela, mostrada na figura 5.53, o usuário precisa deixar a opção *Linear* selecionada e clicar em *Done*. Então o PRO-ENGINEER solicita ao usuário que escolha a face onde será feito o furo. Em seguida o PRO-ENGINEER pede que o usuário escolha um dos planos para servir como referência para cota. O usuário deverá escolher duas faces do sólido. Após a escolha do usuário o PRO-ENGINEER solicita a distância entre a face e o centro do furo, como mostra a figura 5.54.

Figura 5.54 – Área de trabalho do PRO-ENGINEER durante a criação de um furo.

O campo destacado pela seta (distancia = 12) é a caixa de texto onde o usuário digita a cota entre o centro do furo e a face do retângulo. A seta no topo destaca a

superfície onde o furo será desenhado. A seta (lateral do sólido) estaca a superfície escolhida pelo usuário para servir como referência para cota.

Após dimensionar a posição do furo o usuário precisa clicar em *One Side* e, em seguida, clicar em *Done* em um menu semelhante a figura 5.29. Depois escolher a opção *Blind* e clicar em *Done*, em um menu semelhante a figura 5.45. Então o PRO-ENGINEER pede que o usuário digite o valor da altura do furo cilíndrico em uma caixa de texto semelhante a mostrada na figura 5.46. Em seguida deve digitar o valor do diâmetro do furo na caixa de texto mostrada pela figura 5.55.

Figura 5.55 – Caixa de texto onde o usuário deve digitar o valor do diâmetro.

Após definir o valor do diâmetro o usuário clica em OK na janela mostrada pela figura 5.56.

Figura 5.56 – Janela mostrando botão OK que conclui a construção do furo.

Após o clique no botão OK o usuário obtém o furo desejado. A figura 5.57 mostra esse furo.

Figura 5.57 - Sólido + furo

Essas linhas informam as cotas de referência, a altura e o diâmetro do furo.

7 - Como usar o SEMF para analisar o furo desenhado (figura 5.57). Clicando no botão “carregar” o usuário faz com que o SEMF “releia” o arquivo *trail* e reconheça a nova *feature* criada, que é o furo. A figura 5.58 mostra a descrição do furo na área de trabalho do SEMF.

Figura 5.58 – Parte da área de trabalho do SEMF mostrando o reconhecimento do furo cilíndrico.

8 - Desenhar os três furos restantes, e o furo maior do corte, de maneira análoga ao primeiro furo. A figura 5.59 mostra o aspecto da peça após a criação de 5 furos.

Figura 5.59 – Sólido com 5 furos

9 - Utilizar o SEMF para analisar todos os cinco furos clicando no botão “Carregar”. Quando o usuário clicar no botão “Carregar” o SEMF mostra os nomes de todas as *features* desenhadas até o momento na área “Escolher” a *feature* a ser checada como pode ser vista na figura 5.60.

Figura 5.60 – Janela do SEMF mostrando descrição das *features*.

Para verificar se houve alguma violação o usuário precisa apenas clicar no nome da *feature* e dar um duplo clique na regra desejada (Opções de Checagem).

10 - Desenhar o corte. Para desenhar um corte em um sólido o usuário precisa clicar no comando *create*, do menu *manager* (figura 5.61).

Figura 5.61 – Parte do menu *manager* com o comando *create* destacado.

O comando *create* vai abrir o menu mostrado na figura 5.62. O usuário deverá clicar no comando *CUT*.

Figura 5.62 – Parte do menu *manager* com as opções disponíveis para o sólido.

Após clicar no comando *cut* o usuário precisa fazer os mesmos passos que faria, se estivesse desenhando um paralelepípedo sólido. O comando *cut* vai abrir uma tela igual a da figura 5.28.

Figura 5.63 – Rascunho do *cut*

O rascunho do retângulo deve se feito conforme mostrado na figura 5.63: a linha horizontal superior do retângulo deve passar no centro do círculo. O PRO-ENGINEER vai perguntar ao usuário se ele deseja alinhar o *cut* com o furo. O usuário deve escolher a opção “não alinhar”. A figura 5.64 mostra a janela usada pelo PRO-ENGINEER para questionar o usuário.

Figura 5.64 - Janela que pergunta se o usuário deseja alinhar determinada *feature* com outra.

Feito o *cut* a peça toma o aspecto mostrado na figura 5.65.

Figura 5.65 – Peça com o corte (cut)

11 - Clicar no botão “carregar” do SEMF. A saída para essa peça é mostrada na figura 5.66.

Figura 5.66 – Saída do SEMF para a peça desenhada na figura 5.65

12 - Desenhar o cilindro. Para criar um cilindro o usuário deve clicar nos comandos: *CREAT*→*PROTRUSION*→*DONE*→*ONE SIDE*→*DONE*.

Em seguida o usuário deve selecionar uma das faces do sólido principal, como mostrado na figura 5.67.

Figura 5.67 – A seta na lateral do sólido, mostra onde o usuário deve clicar

Em seguida o usuário escolhe a direção de desenho (como mostrado anteriormente na construção do paralelepípedo) e escolhe duas referências para cota. Finalmente o usuário deve rascunhar um círculo e modificar suas cotas na seguinte ordem: primeiro a cota horizontal de referência, segundo a cota vertical de referência e terceiro o diâmetro, como mostra a figura 5.68.

Figura 5.68 – Cotas do cilindro

Feito o rascunho, o usuário precisa clicar em *DONE* e escolher a opção *Blind* no menu que aparece. Esse menu é semelhante ao que aparece na figura 5.45. Então o usuário escolhe um valor para a altura do cilindro da mesma forma que escolheu o valor da altura do prisma sólido e termina a construção do cilindro. A figura 5.69 mostra esse cilindro pronto.

Figura 5.69 – Peça de demonstração com o cilindro terminado

13 - Clicar no botão “carregar”, do SEMF, para que ele reconheça o cilindro. A figura 5.70 mostra o resultado do reconhecimento.

Figura 5.70 – Parte da tela do SEMF mostrando o reconhecimento do cilindro sólido.

13 - Desenhar um *cut* e um segundo paralelepípedo, onde este sofre um furo. Este último furo vai violar a regra de espessura de parede. A figura 5.71 mostra a peça pronta.

Figura 5.71 – Janela do PRO-ENGINEER mostrando a peça pronta.

14 - Clicar no botão “carregar”, do SEMF, para que o programa reconheça as novas *features*.

Após o reconhecimento o SEMF exibe uma lista com os nomes das *features* como é mostrado na figura 5.72.

Figura 5.72 – Parte da janela do SEMF mostrando todas as *features* da peça desenhada na figura 5.71.

O furo cilíndrico tem diâmetro igual a 9.8 mm e está colocado muito próximo da lateral do paralelepípedo, o que vai provocar uma violação da regra relativa à espessura de parede. A figura 5.73 mostra a saída do SEMF, quando o usuário aciona a regra espessura mínima de parede para esse furo.

Figura 5.73 - Saída do SEMF, quando o usuário aciona a regra (espessura mínima de parede).

Na verdade, cada uma das *features* mostradas na figura 5.72 pode ser verificada individualmente, bastando selecioná-las uma por vez. Também, como um simples clique no botão “Todos” o programa verifica todas as *features* em sucessão, dando o diagnóstico de cada uma delas na área “Resultado da Checagem quanto a Violações Encontradas” (área 8 da figura 5.1).

CONCLUSÃO

Neste capítulo foram detalhados as áreas que constituem o SEMF, ou seja, as suas funções numa possível análise de uma peça desenhada no CAD (PRO-E). E também um exemplo de aplicação que mostra passo a passo a construção de uma peça desenhada no PRO-E e a interação do SEMF com o Usuário no processo de checagem quanto às possíveis violações de projeto. Esta interação foi realizada através de formulários gráficos que envia informações ao usuário e também “solicita” informações dele.

A implementação foi feita de maneira a tornar o uso do sistema o mais fácil possível.

CAPÍTULO 6 - CONCLUSÕES E PROPOSTAS PARA TRABALHOS FUTUROS

6.1 CONCLUSÕES

Como foi mencionado anteriormente existe um grande volume de informações de ferramentas de auxílio ao projetista, porém observou-se que estas ferramentas trabalham isoladamente.

No caso da Engenharia Simultânea, notou-se que existe uma dificuldade na obtenção das orientações devido à falta de disponibilidade do especialista.

As técnicas de DFM, DFA e Projeto voltado para a fundição, são de muita valia para o projetista desde que sejam sempre de fácil acesso

Ferramentas Computacionais (CAD, CAE, Software de Solidificação), são muito utilizadas, porém falta a ligação com o conhecimento (regras).

Através das *Features* (características) é possível representar as peça (acabada) e analisar separadamente cada uma dessas *features* quanto as possíveis violações de projeto.

A Base de Conhecimento é de grande importância porque através dela pode-se armazenar o conhecimento específico desejado.

Como foi mencionado acima existem várias maneiras de auxílio ao projetista, porém pode-se constatar que existem limitações e falta de integração entre elas. Para sanar esses problemas um sistema de auxílio ao projetista (SEMF) foi desenvolvido. O sistema utiliza um programa CAD, que trabalha com o conceito de *features* para o reconhecimento de cada característica da peça. Ajuda a minimizar a necessidade da presença física do especialista através da base de conhecimento, permitindo assim um melhor uso do conceito de ES. A base de conhecimento pode ser alimentada com experiências do dia a dia. As soluções de problemas são implementadas de forma instantânea.

Com o emprego do programa SEMF que integra a técnica de DFM e a ferramenta computacional CAD é possível chegar às seguintes conclusões:

- A vida do projetista é facilitada;
- O tempo de desenvolvimento do projeto de peças fundidas é menor (*time-to-market*);
- Probabilidade de erro de projeto é minimizada (regras implementadas);
- Possibilidade de implementação de novas regras, proporcionando um aumento no volume de soluções de problema (base de conhecimento);

Outras características do sistema desenvolvido podem ser resumidos em:

- Programa aberto;
- Programa interativo (interage com o usuário);
- Portabilidade;
- Necessidade de mercado (não há softwares ou sistemas que faça as mesmas funções);
- Trabalha em conjunto com programas CAD (usa o conceito de *feature*).

O objetivo proposto era auxiliar o projetista no projeto de uma peça fundida realizado no PRO-E. Pelo exemplo desenvolvido no capítulo 5, observa-se que esse objetivo foi atendido. O PRO-E fornece dados de saída, relativos à peça que podem ser reconhecidos tornando possível fazer a verificação de diversas *features* baseado em regras. O exemplo de aplicação também comprova a facilidade de uso do sistema proposto.

6.2 TRABALHOS FUTUROS

Uma primeira proposta para trabalhos futuros seria a implementação de outras regras tais as apresentadas na figuras 3.12 e 3.13, visando complementar o sistema e possibilitando uma verificação mais completa de uma peça.

A segunda proposta para trabalhos futuros seria a integração PRO-E/SEMF e ANSYS. A vantagem desta interação é que se poderia fazer um modelo no PRO-E, ou seja, gerar uma malha que pudesse ser analisada no ANSYS. Esta análise poderia ser por

exemplo o processo de solidificação de uma peça, auxiliando assim de forma mais completa o projetista.

Um terceiro trabalho seria um melhor conhecimento da visão 3D do PRO-E, porque algumas das *features* desenhadas pelo PRO-ENGINEER não puderam ser reconhecidas porque o SEMF não tem acesso à sua localização e orientação no espaço 3D. Neste caso seria importante “saber” como funciona o mecanismo de visão tridimensional do PRO-ENGINEER (com o fabricante).

Um quarto trabalho seria criar no programa SEMF um sistema de checagem de massalotes (através de espessura de parede) de uma peça criada no PRO-E.

Um quinto trabalho seria usar esta metodologia num CATIA.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALBER, T. (2000). *Creativity-tool for DFA*. International Forum on DFMA. Neuport, RI USA. June 8 & 9.
- ALMEIDA, F.J. (1997). *Sistematização e automação de projeto: o enfoque da sintaxe*. Tese (Doutorado). Escola De Engenharia de São Carlos, Universidade de São Paulo. São Carlos. Julho.
- AMERICAN FOUNDRYMEN'S SOCIETY (1957). *Cast Metals Handbook*.
- ANSYS User's Manual (1994). *Expanded Workbook Volume I*, SAS IP.
- ASSOCIAÇÃO BRASILEIRA DE METAIS (1978). *Fundição Sob Pressão*. Coordenação Ettore. B. F.
- ASSOCIAÇÃO BRASILEIRA DE METAIS (1980). *Curso de Fundição*. Coordenação Bradaschia F. E.
- ASSOCIAÇÃO BRASILEIRA DE METAIS (1980). *Curso de Fundição*. Coordenação Siegel, M.
- BACK, N. (1983). *Metodologia de projeto de produtos industriais*. ed. Guanabara Dois.
- BEELEY, P.R (1972). *Foundry Technology*. Department Metallurgy, University of Leeds.
- BOOTHROYD, G. (1994). *Product Design for Manufacturing and Assembly*. Marcel Dekker, inc. New York, Basel-Hong Kong.
- BOOTHROYD, G., DEWHURST. P. (1983) *Design for Assembly*, University of Massachsetts, Amherst, USA.
- BRALLA, J. G.(1986). *Handbook of Product Design for Manufacturing - A Practical Guide for Low-Cost Production*. New York: McGrw-Hill,.
- CAMPOS FILHO, M.P (1978). *Solidificação e fundição de Metais e suas Ligas*. Rio de Janeiro. Livros Técnicos e Científicos.

- CHEN, K.H., CHEN, S.J. LIN, L., CHANGCHIEN, S.W. (1998). *An integrated graphical user interface (GUI) for concurrent engineering design of mechanical parts*. Computer Integrated Manufacturing Systems, v.11, nº1-2, p.91-112.
- CHIUSOLI, R. (1996). *Engenharia Simultânea: estudo de casos na indústria brasileira de autopeças*. Dissertação (Mestrado). 147p. São Carlos, S.P. Universidade Federal de São Carlos.
- COMMERCIAL SYSTEMS (1998). <http://mtiac.hq.iitri.com/mtiac/pubs/rp/rp31.htm>, 01/outubro.
- COSTA, H.C.B.de (1996). *E Banco de Dados*. Universidade Federal de São Carlos. São Carlos.
- CHIAVERINI, V (1977). *Tecnologia Mecânica*. v.1: São Paulo, McGraw-Hill do Brasil.
- DON, C.W. (1998). Key features of an integrated product development system. *International Forum on DFMA*. Neuport, RI USA. June 8 & 9.
- DIETER, G.E (1983). *Engineering design*. New York: McGraw-Hill.
- DOYLE, L. E., C. A. Keyser, J. L. Leach, G.F. Schrader,; M. S. Singer (1985). *Manufacturing Processes and Materials for Engineers*, 3d ed. *Englewood Cliffs, N. J.: Prentice-Hall*.
- ESTRIN, L. (1995). *Uma análise dos softwares de solidificação de fundidos*. *Fundição e Serviços*, p.16-27, dezembro 1995, janeiro 1996.
- FERREIRA, J.C.V & NAVEIRO, R.M (2001). *O Uso de Projetos Orientados no Desenvolvimento de Produtos*. 1º Congresso Brasileiro de Engenharia de Fabricação (COBEF) – Curitiba (PR).
- FUNDIÇÃO & MATÉRIAS-PRIMAS [1998]. Informativo Oficial da Associação Brasileira de Fundição –ABIFA.
- GANTER, M. A., SKOGLUND, P. A. (1993). *Feature extraction for casting core development*. Transactions of the ASME, v.115, p. 744-751, december.
- GASPERVICH, J.1991. *How to mold better small parts*. Machine Design, v.8, p. 40-43, august.

- GAYRETLI, A., ABDALLA, H.S. (2000). *A feature-based prototype system for the evaluation and optimisation of manufacturing processes*. Computers & Industrial Engineering, v.37, p.481-484.
- GEDDAM, A.; KALDOR, S (1993). *Process-driven engineering - a key element in integration design and manufacture*. Annals of the CIRP, v.42, p. 155-158, january.
- GONÇALVES FILHO, E. V (1994). *Gerenciamento e Integração de Atividades na Engenharia Simultânea/ Integrada. Modulo I Apostila do curso de Sistema de Fabricação e Tecnologia de Grupo*, junho.
- GONÇALVES FILHO, E. V (2001). *Objetivos de uma Empresa. Notas de aula do curso SEM 5806 - Sistema de Fabricação e Tecnologia de Grupo*.
- GORNI, A. A. (2001). Prototipagem rápida: o que é, quem faz, e por que utiliza-lá. Máquinas e Metais, abril.
- HEINE, R. W.; LOPER, C. R.; ROSENTHAL, P. C. (1967). *Principles of Metal Casting*. New York. McGraw-Hill Book Company.
- KALPAKJIAN, S.(1989). *Manufacturing Engineering and Technology*. Reading, Mass.: Addison-Wesley.
- KONDIC, V. (1973). *Princípios metalúrgicos de fundição*. Traduzido por Cláudio Luiz Mariatto e outros. São Paulo, Polígono, ed. da Universidade de São Paulo.
- KUSIAK, A.; LEE, G. H. (1997). *Design of Part and Manufacturing Systems for Reliability and Maintainability*. The International Journal of Advanced Manufacturing Technology. Springer Veriag London Limited, v. 13 p.67-76.
- KUSIAK, A.; WANG J.(1993). *Decomposition of the design process*. Journal of Mechanical Design, v.115, p. 687-695, december.
- LAU, P., JI, K.H. (2000). *Design for manufacturing: a dimensioning aspect*. Journal of Materials Processing Technology, v.91, p.121-127.
- LUTTERS, D., BRINKE, A.H., STREPPPEL, H.J.J.K. (2000). *Computer aided process planning for sheet metal based on information management*. Journal of Materials Processing Technology, v.103, p.120-127.
- MAREK, C. T. (1952). *Fundamentals in the Production and Design of Castings*: John Wiley & Sons, second printing, june.

- MED, Embraer [1998]. *Seção de Engenharia de Materiais*.
- MEERKAMM, H.; WARTZACK, S. (2001). *Potencial dos projetos orientados à manufatura*. Máquina e Metais. Aranda Editora – p.136-147, Abril.
- METALS HANDBOOK - ASM. *Casting design Handbook*.
- NAVEIRO; (1995). *Projeto de peça fundidas*. Gestão & Produção, v.3, n.5, p.21-32, novembro.
- NAVEIRO, R. M.; GRADY, P. O.; (1997). *Discast: um sistema inteligente para auxílio ao projeto de peça fundidas*. Gestão & Produção, v.4, n.3, p.251-263, dezembro.
- NNAJI, V.; ANAND, S. (1990). *Machine Understanding of Manufacturing Features*. *International Journal of Production Research*, v.34, n. 7, p. 1791-1819.
- PARAMETRIC TECHNOLOGY CORPORATION (1998). *Introduction to Pro/Engineer. Release 20.0*. Parametric Technology Corporation . USA.
- PEREIRA, R. L.(1991). *Curso de Fundição*: Publicações São Carlos.
- PROVENZA, F. (1989). *Projetista de Máquinas*. Ed. F. Provenza.
- REDFORD, A. H. (1994). *Design for Assembly: principles and practice*. New York, Mc Graw-Hill, 202p.
- RESENDE, M.O (1992). *Princípios de Processo de Produção*. Publicação 013/95 Vol.1.
- RUIZ, C.; KOENIGSBERGER, F. (1970). *Design for Strength and Production*. Macmillan and Coltd.
- SHAH, B., BETTIG, J. (1998). *An object-oriented program shell for integrating CAD software tools*. *Advances in Engineering Software*, v.30, p.529-541.
- SCHÜTZER, K. (1988). *Desenvolvimento de um sistema de projeto de engrenagens auxiliado por computador*. São Carlos, 236p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.
- SENAI - (1987). *Departamento Regional de Minas Gerais*. Publicação técnica, v.1-23.
- SILVA, A.S.C. (1998). *Projeto voltado para a manufatura de peças fundidas: Guia para auxiliar o projetista*. Dissertação de mestrado, v.1 e v.2, São Carlos.
- SMALLEY, O. (1975), *Mechanical details for product design*. McGraw-Hill Book Company, p.68-72, New York - Toronto - London.

- SOUZA SANTOS, A.B. & Castelo Branco C. H. *Metalurgia dos Ferros Fundidos Cinzentos e Nodulares*. São Paulo, IPT, 1977. P.241 (Publicação IPT N° 11000).
- STEMMER, C. E. (1979). *Projeto e Construção de Máquinas: regras gerais de projeto, elementos de máquinas*. Porto Alegre, Globo.
- WHITNEY D. (1994). *Integrated Design and Manufacturing in Japan*; Manufacturing Review, vol. 6, n° 4, A.S.M.E, U.S.A.
- YOGUI, R. (1998). *Engenharia com tecnologia aplicada*. CADware Technology. V.2, n.8, p.30.
- YOUNG, C. F.; YOU, S. S. (2000). *Automatic Feature Recognition from engineering Drawings*. The International Journal of Advanced Manufacturing Technology. Springer Veriag London Limited, v. 14 p.495-507.
- ZHA, X. F.; LIM, S. Y. E.; FOK, S. C. (1998). *Integrated Knowledge Assembly Sequence Planning*. The International Journal of Advanced Manufacturing Technology. Springer Veriag London Limited, v. 14 p.50-64.

BIBLIOGRAFIA

- ABM - Metalurgia (1990). Tradução e adaptação do artigo "*Selecting Materials for Strength*", divulgado na revista "Advanced Materials & Processes" v. 37, n° 6, Junho por Vicente Chiaverini.
- AGOSTINHO, O. L.; RODRIGUES, C. S; LIRANI, J; (1981). *Princípios de Engenharia de fabricação Mecânica.- Processos de Fabricação*, v. 2
- ASME Handbook, (1958). *Metal Engineering Process*, McGraw-Hill book Company.
- ASME HANDBOOK (1965). *Metal Engineering Desing*. McGraw-Hill.
- ASSOCIAÇÃO BRASILEIRA DE METAIS (1973). *Custos em Fundição*, São Paulo.
- ASSOCIAÇÃO BRASILEIRA DE METAIS (1978). *Curso de Fundição Sob Pressão*. Coordenação Bresciani F. E.
- ASHLEY, S (1995). *Cutting costs and time with DFMA*. Mechanical Engineering, p. 74-77, march..

- BARCLAY, I.; POOLTON, J. (1994). *Concurrent engineering: concept and practice*. Int. J. of Vehicle Design, v.15, p.529-544, n. 3/4/5.
- BRAZIER, D. (1990). *Concurrent Engineering: Participating in Better Designs*, Mechanical Engineering, january.
- BREMER, O C. F., ROZENFELD, H. H. (1992). *Flexible reference model as basic for simultaneous engineering in an integrated environment*.
- BROOKS, H. M. (1987). *Expert systmes and intelligent information, retrieval*. Information –Processing Management, v. 23, p. 367-382.
- CEGLAREK, D.; SHI, J.; WU, S. M (1994). *A Knowledge - based diagnostic approach for the launch of the auto-body assembly process*. Journal of Engineering for Industry, v. 116, p. 491-499, november.
- CHEN, Y.M., WEI, C.L. (1997). *Computer-aided feature-based design for net shape manufacturing*. Computer Integrated Manufacturing Systems, v.10, n°2, p.147-164.
- CHOW, W (1978). *Cost reduction in product design*: Van Nostrand Reinhold Company.
- CHU, X.; HOLM, H (1993). *Produt manufacturability control for concurrent engineering*, *Computers in Industry*, v.24 p. 29-38.
- CLARK, K. B., FUJIMITO T. (1991). *Product Development Performance*. Harward Business Shool Press, Boston, Mass.
- COSTA, C.P. (1994). *Desenvolvendo produtos com engenharia simultânea e workgroup computing*. 6° Congresso Nacional da Automação Industrial (CONAI). São Paulo.
- CORBETT, J (1986). *Design for economic manufacture*. Annals of the CIRP, v.35, p.93-97, january.
- DALLAS, D. B. (3th ed.), (1976). *Tool and Manufacturing Engineers Handbook*, Dearborn, Mich.: Society of Manufacturing Engineers.
- DEWHURST, P (1988). *Cutting assembly costs with molded parts*. Machine Design, v.21, p. 68-72, july.
- DIXON, J. R (1995). *Knowledge-based systems for design*. Specal 50th Anniversary Design Issue, v.117, p.11-16, june.
- DOEHLER, H. H. (1951). *Die Casting*, McGraw-Hill Book Compnay, Inc. 1951.

- DOWLASTSHAHI, S. (1997). *A modeling approach to logistics in concurrent engineering*. European Journal of Operational Research, v.115, p.59-76.
- EVANS, S.; LETTICE, F.; SMART, P. (1995). *A Faster, Cheaper and Safer Route to CE*. World Class Design to Manufacture, v.2, n. 2, p.10-16.
- FAZIO, T.L. De; OUTROS (1993). *A prototype of feature-Based design por assembly*. Journal of mechanical Design, v. 115, p. 723-733.
- FENG, C.X.; KUSIAK, A. (1995). *Costraint-based design of parts*. Computr-Aided Design, v.27 n. 5 , p. 343-352, may.
- GADH, r. (1995). *Automatic Determination of feature interactions in Design-for-Manufacturing Analysis*. Transactions of the ASME, v.117, march.
- GAYRETLI, A., ABDALLA, H.S. (1999). *An object-oriented constraints-based system for concurrent product development*. Robotics and Computer-Integrated Manufacturing, v.15, p.133-144.
- GUPTA, S. K.; NAU, D. S. (1995). *Systematic approach to analysing the manufacturability of machined parts*. Computer-Aided Design, v. 27 n° 5, may.
- FREIRE, J. M. (1978). *Tecnologia Mecânica por J.M.Freire*. Rio de Janeiro, livros técnicos e científicos.
- FS. Fundação e Serviço (2002). Aranda Editora – Ano 12.
- JAWAHAR, N.; ARAVINDAN, P.; PONNAMBALAM, S. G.; RAGHAVENDRA, L.N. (1998). *Knowledge-Based Workcell Attribute Oriented Dynamic Schedulers for Flexible Manufacturing Systems*. The International Journal of Advanced Manufacturing Technology. Springer Veriag London Limited, v. 14 p.514-538.
- JOINES, S. and AYOUB, M. (1995). *A Design for assembly: an ergonomic approach*. Industrial Engineering, p. 42-46, january.
- KAGAN, P., FISCHER, A. (2000). *Integrated mechanically based CAE system using B-spline finite elements*. Computer-Aided Design, v.32, p.539-552.
- KRAUSE, F.-L. (1991). *Configuração de produtos com sistemas especialistas*. Máquinas e Metais, agosto.
- KRAUSE, F. L., ULBRICH, A.; WOLL R. (1993). *Methods for quality-driven product development*. Annals of the CIRP, v.42, p. 151-154, january.

- KUSIAK, A.; WANG J.; HE, D. W.; FENG, C. X. (1995). *A Structured Approach Analysis of desing Process*. Ieee Transactions on componentes, Packaging, and Manufacturing Technology-Part A, v. 18, n. 3, september.
- KWEI LI, R.; HUANG, C. L. (1992). *Assembly code generation from a feature-based geometric model*. *Int. J Prod. Res*, v. 30, nº 3, p 627-646.
- LEBRETON, H. (1975). *Defectos de las peizas de fundicion*. Urmo, S. A. de ediciones.
- LOUVO, A.; (1997). *Casting Simulation as a Tool in Concurrent Engineering*. International ADI and Simulation Conference, may.
- LIN, J. C.; TAI, C. C. (1998). *The Runner Optimisation Design of a Die-Casting Die and the Part Produced*. The International Journal of Advanced Manufacturing Technology. Springer Veriag London Limited, v. 14 p.133-145.
- LINDERG, L. (1993). *Note on concurrent engineering*. *Annals of the CIRP*, v.42, p. 159-162, january.
- LIONEL,J.D. AD SULLY, E., *Die Casting*. Metal-Handbook, Manuais, v.15.
- PARK, H.G., BAIK, J.M. (1999). *Enhancing manufacturing product development through learning agent system over internet*. *Computers & Industrial Engineering*, v.37, p.117-120.
- PARK, H.G., KIM, Y.N. (1998). *A simultaneous design modeling support system*. *Computers & Industrial Engineering*, v.35, nº 1-2, p.9-12.
- PEREIRA, A. A. T. (1961). *Alimentadores para peças fundidas em areia*. Contribuição Técnica ao XVI Congresso da Associação Brasileira de Metais. Volta Redonda -RJ, junho.
- SCHMITZ, J. M.; DESA, S. (1994). *Development and implementation of a design for producibility method for precision planar stamped products*. *Journal of Mechanical Design*, v.116, p. 349-357, june.
- POLI, C., DASTIDAR, P., MAHAJAN, P.; GRAVES, R. J. (1993). *Design for stamping-analysis of part attributes that impact die construction costs for metal stampings*. *Journal of Mechanical Design*, v.115, p. 735-743, december.

- SEHDEV, K.; FAN, I. S.; COOPER, S.; WILLIAMS, G. (1995). *Design for Manufacture in the Aerospace Extended Enterprise*. World Class Design to Manufacture, v.2, n. 2, p.28-33.
- SHIGLEY, J. E. (1984) *Elementos de máquinas/ Joseph . Edward. Shiegly* - Tradução de Edival Ponciano de Carvalho - Rio de Janeiro: LTC. Livros técnicos e Científicos Editora S.A.
- SINGH, K. J. (1992). *Concurrent engineering pilot project at GE aircraft engineers*. Concurrent Engineering Research Review, autumn, v.4, Special Issue, p. 14-23.
- SPOTTS, M. F. *Design Engineering Projects*. Prentice-Hall, Inc., Englewood Cliffs, N.j.
- STRASSER, F. (1975). *Designing parts that are easy to machine*. Machine Design, v.7, p. 65-69, august.
- UPTON, B. (1982) *Pressure diecasting Pergamon Press*.
- VAN VLACK, L. H. (1970). *Princípios de Ciências dos materiais*, traduzido pelo Eng Luiz Paulo Camargo ferrão, São Paulo. Edgard Blúcher.

APÊNDICE 1- IMPLEMENTAÇÃO DO PROGRAMA SEMF

A implementação do SEMF nasceu da necessidade de se aplicar regras de fundição a um desenho feito em um programa CAD e informar, ao usuário, se o desenho viola alguma (as) dessas regras. Para tal, o SEMF precisa de um banco de dados que armazene, de forma conveniente, as informações do desenho. Por isso, criou-se a APD (Árvore Principal de Dados) e a LR (Lista de Referência), que juntos formam um banco de dados que atende às necessidades do SEMF. A segunda etapa de criação do SEMF veio da necessidade de se extrair as informações do desenho feito no programa CAD. O CAD utilizado nesse trabalho foi o PRO-ENGINEER. Esse programa armazena, em um arquivo texto, todas os comandos dados pelo seu usuário, no momento da criação do desenho. O SEMF então, precisa de um grupo de procedimentos que extraíam as informações desse arquivo. Um grupo de procedimentos foi criado para reconhecer, nessas informações (comandos dados pelo usuário), as *features* do desenho e alimenta a APD com essas *features* para que possam ser submetidas as regras de fundição. A criação desse procedimento foi a terceira etapa na construção do SEMF. Um outro grupo de procedimentos, que formam a base de conhecimentos do SEMF, verificam, nas informações contidas na APD, se há algum erro de projeto, ou seja, aplicam as regras de fundição nas informações contidas na APD. Esse procedimentos também são responsáveis por informam o resultado da verificação ao usuário. Esses procedimentos foram organizados em classes, o que torna o arquivo fonte do SEMF um código orientado a objeto.

APD + LR

Procedimento usado para implementação da APD e da LR:

Procedimento usado para implementação da APD e da LR:
Passo1 - Criar o registro identificação com os campos ID e pos. O nome do registro é Ridentificação.
Ridentificação – é o registro responsável por armazenar as informações de identificação da UI. Ele tem dois campos: um inteiro, chamado pos, que indica a posição da UI na APD e

um outro chamado ID, que é do tipo cadeia e funciona como um índice da APD. Não pode haver duas Uis com mesmo ID. O campo ID aceita no máximo de 50 caracteres. O campo identificação mostrado na figura 4.5 é uma variável desse tipo.

Em pascal :

```
TYPE
  RIDENTIFICACAO = RECORD
    POS : WORD;
    ID : STRING[50];
  END;
```

Passo 2 – implementar um registro com 3 variáveis reais para representar as 3 coordenadas de um ponto espacial. O nome do registro é Rponto.

Rponto – é um registro formado por três campos do tipo real. O texto abaixo mostra a sua construção em linguagem pascal.

```
TYPE
  (** REGISTRO PONTO **)
  RPONTO = RECORD
    X, Y, Z : EXTENDED;
  END;
  (***** X, Y E Z REPRESENTAM A 3 COORDENADAS ESPACIAIS *****)
```

Passo 3 – implementar o nó de uma fila de variáveis do tipo Rponto. Esse nó é um registro chamado Rpponto. Pponto é um ponteiro que aponta para uma variável do tipo rpponto, ou seja, para um nó da fila de pontos.

Rpponto – é o nó de uma fila cujos dados são os registros Rponto. Em linguagem pascal:

```
TYPE
  (** NÓ DA FILA DE PONTOS **)
  PPONTO = ^RPPONTO;
  RPPONTO = RECORD
    DADO : RPONTO;
    PRÓXIMO : PPONTO;
  END;
```

Passo 4 – implementar a classe que vai gerenciar a entrada e a saída de nós na fila de pontos. O nome dessa classe é Tponto.

Tponto – é o nome da classe que “administra” a fila de pontos. Ela possui procedimentos. Um dos procedimentos se chama inclui e recebe como argumento um registro rponto. A cada vez que esse procedimento é executado é inserido um novo registro na fila de pontos (rponto). O campo pontos da UI, mostrada na figura 4.5, é uma variável desse tipo. Parte da implementação, em linguagem pascal, é mostrada a seguir:

```
TYPE
  (** CLASSE TPONTO **)
  TPONTO = CLASS
```

```

PRIVATE
PROTECTED
    PRIMEIRO,
    ATUAL : PPONTO;
    Q : WORD;
PUBLIC
    CONSTRUCTOR CREATE;
    PROCEDURE INICIA;
    PROCEDURE INCLUI(X : RPONTO);
    PROCEDURE DESTROI;
    FUNCTION FPRIM : PPONTO;
END;

```

Passo 5 – Implementar o tipo com um campo inteiro. O nome do registro é Rint. A implementação em Pascal é mostrada abaixo:

```

TYPE
    RINT = INTEGER;

```

Passo 6 – implementar o registro que será o nó da fila de inteiros. Esse registro tem um campo chamado dado (do tipo inteiro) e outro chamado próximo do tipo ponteiro para um outro nó. A implementação em Pascal é mostrada no texto abaixo.

```

TYPE
    PINT = ^RPINT;
    RPINT = RECORD
        DADO : RINT;
        PROXIMO : PINT;
    END;

```

Passo 7 – implementar a classe responsável pelo gerenciamento da fila de inteiros. O nome da classe é Tint.

Tint – é o nome da classe que “administra” uma fila de variáveis do tipo inteiro.

Tem também o procedimento “inclui” análogo ao da classe tponto.

Parte da implementação em pascal é:

```

TYPE
(*** CLASSE TINT ***)
    TINT = CLASS
    PRIVATE
    PROTECTED
        PRIMEIRO,
        ATUAL : PINT;
        Q : WORD;
    PUBLIC
        CONSTRUCTOR CREATE;
        PROCEDURE INICIA;
        PROCEDURE INCLUI(X : RINT);
        PROCEDURE DESTROI;
        FUNCTION FPRIM : PINT;
    END;

```


Passo 8 – implementar o tipo real. O nome do tipo é Rreal. A implementação é mostrada abaixo:

```
INTERFACE
TYPE
  RREAL = EXTENDED;
```

Passo 9 – implementar o nó da fila de real. Esse nó é um registro que tem o nome de Nreal. A implementação em Pascal é mostrada abaixo:

```
TYPE
(** NO DA FILA DE REAIS **)
  PREAL = ^RPREAL;
  RPREAL = RECORD
    DADO : RREAL;
    PRÓXIMO : PREAL;
  END;
```

Passo 10 – implementar a classe que vai gerenciar a fila de nós com dados reais. O nome da classe é Treal.

Treal – é o nome da classe que “administra” uma fila de números reais. Tem também o procedimento “inclui” análogo ao da classe tponto.

Parte da implementação em Pascal é mostrada no texto abaixo:

```
TYPE
(** CLASSE TREAL **)
  TREAL = CLASS
  PRIVATE
  PROTECTED
    PRIMEIRO,
    ATUAL : PREAL;
    Q : WORD;
  PUBLIC
    CONSTRUCTOR CREATE;
    PROCEDURE INICIA;
    PROCEDURE INCLUI(X : RREAL);
    PROCEDURE DESTROI;
    FUNCTION FPRIM : PREAL;
  END;
```

Passo 11 – implementar uma classe com 3 campos: um do tipo tint, outro do tipo tponto e outro do tipo treal. Criando assim uma estrutura com as 3 filas de informações (pontos inteiros e reais). O nome da classe é Tdados.

Tdados – é uma classe que possui um campo chamado pontos (do tipo Tponto), outro chamado reais (do tipo Treal) e outro (do tipo Tint) chamado inteiro. Assim, essa classe possui as três filas de informações primárias usadas pelo SEMF. Essa classe possui, também, três subrotinas chamadas de inicioponto, incluiint, e incluiereal. Esses procedimentos executam os procedimentos de inclusão de dados dos 3 campos da classe.

Parte da implementação em pascal é mostrada no texto abaixo.

Type

Tdados = class

Private

Protected

Pontos : tponto;

Reais : treal;

Inteiros : tint;

Public

Constructor create;

Procedure inicia;

Procedure destroi;

(* Pontos *)

procedure inclui ponto(x : rponto);

function ponto acesso : pponto;

function nponto(x : byte) : rponto;

(* Reais *)

procedure inclui real(x : rreal);

function real acesso : preal;

(* Inteiros *)

procedure inclui int(x : rint);

function int acesso : pint;

end;

Passo 12 – implementar uma classe com dois campos do tipo Tdados. O nome desses campos serão GEOMETRIA e FUNDIDO. O campo Geometria irá armazenar informações sobre a geometria da peça. O campo fundido irá armazenar informações sobre o processo de fundição da peça. O nome dado a essa classe foi: Tgeofund.

Tgeofund – essa classe divide a UI em dois conjuntos principais de informação: as informações de geometria e as informações inerentes ao processo de fundição. Essa classe possui um campo chamado geometria e outro chamado de fundido ambos do tipo Tdados. Ou seja, essa classe possui as 3 filas de informações primárias, para armazenar informações sobre o processo de fundição, e 3 filas de informação primária para armazenar informações sobre a geometria da peça.

O texto abaixo mostra parte da implementação em pascal:

TGEOFUND = CLASS

PRIVATE

PROCEDURE INICIA;

PROTECTED

PUBLIC

GEOMETRIA,

FUNDIDO : TDADOS;

CONSTRUCTOR CREATE;

END;

Passo 13 – implementar uma classe que une um campo do tipo Tgeofund com campos do tipo Ridentificação e Rtipo. O tipo Rtipo é um número inteiro que, no SEMF, vai servir

para indicar o tipo de *feature* que uma UI armazena. O nome dessa classe é Rfeature.

Rfeature – é uma das principais classes do banco de dados do SEMF. Ela é a própria definição do campo dados da UI da APD. Ela possui um campo chamado identificação, outro chamado tipo e outro chamado dados. O campo identificação é do tipo ridentificação, explicado acima. O campo tipo é uma variável inteira cujo valor representa o tipo de informação que a UI armazena. Por exemplo, para o tipo 1 a UI armazena as informações de um sólido prismático de base retangular. O campo chamado dados é do tipo Tgeofund. Essa classe possui também um procedimento chamado de regras. Esse procedimento atualmente está inativo, mas pode ser usado para se colocar regras diretamente nas Uis do banco de dados. Assim, cada UI do banco de dados do SEMF, além de armazenar informações, pode também manipular suas próprias informações. O procedimento regras pode conter sub-rotinas especialistas em determinada área de conhecimento de engenharia e fazer com que cada unidade informação encontre possíveis erros em suas informações.

Parte da implementação, em pascal, é mostrada no texto abaixo.

```
RFEATURE = CLASS
PRIVATE
PROTECTED
PUBLIC
    IDENTIFICACAO : RIDENTIFICACAO;
    TIPO : RTIPO;
    DADOS : TGEOFUND;
    CONSTRUCTOR CREATE;
    PROCEDURE REGRAS;
    PROCEDURE INICIA;
    PROCEDURE DESTROI;
END;
```

Passo 14 – implementar a classe que será o nó da APD, unindo um campo do tipo Rfeature com dois campos ponteiros. O nome dessa classe é Rpfeature.

Rpfeature – é a classe que forma o nó da APD, mostrado na figura 4.5. Ela é composta por quatro campos: dado, anterior, afrente e abaixo. O campo dado é do tipo rfeature. Os outros campos são ponteiros para uma variável do tipo rpfeature. Isso forma o nó da APD que corresponde ao seguinte esquema:

```
UNIT UNFEAT;
INTERFACE
USES URFEAT;
TYPE
    PFEATURE = ^RPFEATURE;
    RPFEATURE = CLASS
    PUBLIC
        DADO : RFEATURE;
```



```

    ANTERIOR,
    AFRENTE,
    ABAIXO : PFEATURE;
    CONSTRUCTOR CREATE;
    PROCEDURE INICIA;

```

```
END;
```

Passo 15 – implementar a classe que vai ser, literalmente, a APD. O nome dessa classe é Tfeature.

Tfeature – é uma classe que tem dois campos chamados de primeiro e atual. O campo primeiro é um ponteiro que vai apontar para a primeira UI da APD. O campo atual é outro ponteiro, do mesmo tipo que o primeiro, que vai apontar para a última UI da APD. Essa classe tem, também, um procedimento chamado inicia. Esse procedimento é responsável por alocar a primeira UI na APD. Essa é parte vital da primeira classe da hierarquia das principais classes que constituem o SEMF. O texto abaixo mostra parte da implementação, em pascal, dessa classe.

```

    TFEATURE = CLASS
    PRIVATE
        Q : WORD;
    PROTECTED
    PUBLIC
        PRIMEIRO, ATUAL : PFEATURE;
        PROCEDURE INICIA(X : RFEATURE);
        PROCEDURE INCREMQ;
    END;

```

Passo 16 – implementar o nó da LR. Esse nó é um registro chamado Rlr.

Rlr – é um registro que tem três campos. Um campo se chama dado, outro se chama próximo e outro se chama anterior. O rlr é o nó da LR. O campo dado é um ponteiro que vai apontar para um determinado nó da APD. Os campos próximo e anterior são ponteiros que vão apontar para outros registros rlr na LR.

```

TYPE
    PLR = ^RLR;
    RLR = RECORD
        DADO : PFEATURE;
        PRÓXIMO,
        ANTERIOR : PLR;
    END;

```

Passo 17 – implementar a classe que vai gerenciar a LR. Essa classe também será encarregada de incluir *features* na APD de forma hierárquica. O nome dessa classe é Tlr.

Tlr – é uma das classes mais complexas do SEMF. É a primeira classe da hierarquia principal do SEMF. Essa classe é responsável por incluir Uis de forma organizada na APD, retirar Uis da APD, acionar o procedimento de regra embutida em cada UI da APD.

O texto abaixo mostra parte da implementação em Pascal dessa classe.

```
TLR = CLASS
PRIVATE
    ARVORE : TFEATURE;
PROTECTED
    PRIMEIRO, ATUAL : PLR;
    Q : WORD;
PUBLIC
    PROCEDURE INICIA(X : RFEATURE);
        {Esse procedimento executa o procedimento inicia da APD}
    PROCEDURE IABAIXO(ID : STRING; X : RFEATURE);
        {Esse procedimento inclui uma UI no ponteiro Abaixo da UI indicada por ID}
    PROCEDURE RETIRA( X : STRING);
        {Esse procedimento retira do banco de dados a UI cujo ID seja igual a X do argumento}
    PROCEDURE MODIFY(X : STRING; Y : RFEATURE);
        {Esse procedimento substitui as informações de uma UI pelo registro Y ou seja, ID seja igual ao X do
argumento}
    FUNCTION RETORNA(X : STRING) : PLR;
    FUNCTION PRIM : PLR;
    PROCEDURE REGRAS;
END;
```

O procedimento iabaixo inclui uma nova *feature* na hierarquia abaixo de outra *feature*. Pode-se notar que não existe um procedimento que coloca uma UI ligada ao ponteiro afrente de outra. Isso ocorre para garantir que uma nova UI seja colocada no banco de dados relacionando-se com uma UI de topologia superior a sua. O código do procedimento iabaixo é mostrado no texto abaixo:

```
PROCEDURE TLR.IABAIXO(ID : STRING; X : RFEATURE);
VAR
    LRPERCORRE{,
    LRANT} : PLR;
    APDPERCORRE{,
    APDANT} : PFEATURE;
BEGIN
    (** PROCURA POR FEATURE COM ID IGUAL A VARIÁVEL ID DO ESCOPO **)
    LRPERCORRE:=PRIMEIRO;
    // LRANT := PRIMEIRO;
    WHILE (LRPERCORRE <> NIL) AND (LRPERCORRE^.DADO.DADO.IDENTIFICACAO.ID <> ID) DO
        BEGIN
            // LRANT := LRPERCORRE;
            LRPERCORRE := LRPERCORRE^.PRÓXIMO;
        END;
    (** SE A FEATURE NÃO FOR ENCONTRADA **)
    IF LRPERCORRE = NIL THEN
        BEGIN
            SHOWMESSAGE('ERRONN');
            SHOWMESSAGE(ID);
            AVERRO.MEMO1.LINES.ADD('OCORREU O SEGUINTE ERRO:');
            AVERRO.MEMO1.LINES.ADD('NA PROCEDURE IABAIXO DE UM OBJETO DA CLASSE');
            AVERRO.MEMO1.LINES.ADD('TLR A VARIÁVEL PERCORRE TERMINOU COM VALOR');
            AVERRO.MEMO1.LINES.ADD('NIL APOS PERCORRER A FILA LR A PROCURA DA ');
            AVERRO.MEMO1.LINES.ADD('FEATURE COM A ID '+ ID);
```

```

    AVERRO.SHOWMODAL;
END
(** SE A LR ESTIVER VAZIA **)
ELSE
IF LRPERCORRE = NIL THEN
BEGIN
    AVERRO.MEMO1.LINES.ADD('OCORREU O SEGUINTE ERRO:');
    AVERRO.MEMO1.LINES.ADD('NA PROCEDURE IABAIXO DE UM OBJETO DA CLASSE');
    AVERRO.MEMO1.LINES.ADD('TLR A VARIÁVEL LRANT TERMINOU COM VALOR IGUAL');
    AVERRO.MEMO1.LINES.ADD('AO DA VARIÁVEL LRPERCORRE, OU SEJA, A FEATURE');
    AVERRO.MEMO1.LINES.ADD('PRIMARIA NÃO FOI ENFILEIRADA');
    AVERRO.MEMO1.LINES.ADD('FEATURE COM A ID '+ ID);
    AVERRO.SHOWMODAL;
END
(** SE A FEATURE FOR ENCONTRADA **)
ELSE
BEGIN
    (** PROCURAR FIM DA FILA ABAIXO DA FEATURE ESPECIFICADA EM ID **)
    //SE A FILA ABAIXO DA FEATURE ESPECIFICADA POR ID ESTIVER VAZIA...
    IF LRPERCORRE^.DADO^.ABAIXO = NIL THEN
    BEGIN
        NEW(LRPERCORRE^.DADO^.ABAIXO); //ALOCA
        ARVORE.ATUAL := LRPERCORRE^.DADO^.ABAIXO;
        ARVORE.ATUAL^. := RPFEATURE.CREATE;
        ARVORE.ATUAL^.DADO := X; //COLOCA A FEATURE
        ARVORE.ATUAL^.ANTERIOR := LRPERCORRE^.DADO; //O ANTERIOR É A FEATURE ACIMA
        ARVORE.ATUAL^.AFRENTE := NIL; //NADA AFRENTE
        ARVORE.ATUAL^.ABAIXO := NIL; //NADA ABAIXO
        (*COLOCAR REFERÊNCIA PARA A ULTIMA FEATURE NA LR *)
        NEW(ATUAL^.PRÓXIMO);
        ATUAL^.PROXIMO^.ANTERIOR := ATUAL;
        ATUAL := ATUAL^.PROXIMO;
        ATUAL^.DADO := ARVORE.ATUAL;
        ATUAL^.PROXIMO := NIL;
        ARVORE.INCREMQ;
        INC(Q);
    END
    //SE NÃO...
    ELSE
    BEGIN
        (*PROCURAR O ULTIMO*)
        APDPERCORRE := LRPERCORRE^.DADO^.ABAIXO;
        WHILE APDPERCORRE^.AFRENTE <> NIL DO APDPERCORRE:=APDPERCORRE^.AFRENTE;
        (*ALOCAR MEMÓRIA*)
        NEW(APDPERCORRE^.AFRENTE);
        APDPERCORRE^.AFRENTE^. := RPFEATURE.CREATE;
        (*COLOCAR A FEATURE*)
        ARVORE.ATUAL := APDPERCORRE^.AFRENTE;
        ARVORE.ATUAL^.DADO := X;
        ARVORE.ATUAL^.ANTERIOR := APDPERCORRE;
        ARVORE.ATUAL^.AFRENTE := NIL; //NADA AFRENTE
        ARVORE.ATUAL^.ABAIXO := NIL; //NADA ABAIXO
        (*COLOCAR REFERÊNCIA PARA A ULTIMA FEATURE NA LR *)
        NEW(ATUAL^.PROXIMO);
        ATUAL^.PROXIMO^.ANTERIOR := ATUAL;
        ATUAL := ATUAL^.PROXIMO;
    END

```


<pre> ATUAL^.DADO := ARVORE.ATUAL; ATUAL^.PROXIMO := NIL; ARVORE.INCREMQ; INC(Q); END; END; END;</pre>
<p>Passo 18 – implementar uma classe que herda todas as características de TLR. Esta classe sofrerá mudanças futuras deixando a classe TLR protegida. Tlr1 – é a classe derivada de TLR e tem as mesmas características desta.</p>

Carregar arquivo *trail*

O arquivo *trail*, criado pelo PRO-ENGINEER, fica armazenado no disco rígido do micro. O acesso as informações no disco rígido é muito lento, se comparado ao acesso a informações na memória RAM do micro. Por isso, o SEMF copia os caracteres do arquivo *trail* na memória RAM do micro para aumentar a velocidade no reconhecimento de *features*. Para armazenar tais caracteres o SEMF precisa criar uma fila onde os nós são formados por textos. A fila formada pelas filas de caracteres forma uma estrutura semelhante ao texto do *trail*.

Metodologia usada para a leitura do arquivo <i>trail</i> .
<p>Passo 19 – implementar um registro com um campo do tipo string. O nome do registro é rcarac é mostrado abaixo sua implementação em Pascal.</p> <pre> TYPE RCARAC = RECORD DADO : STRING; END;</pre>
<p>Passo 20 – implementar o nó da fila de caracteres. Esse nó é um registro que possui dois campos. Um do tipo rcarac e outro do tipo pcarac, que é um ponteiro para o próprio nó. O nome do nó é nocarac. Sua implementação em Pascal é mostrada abaixo.</p> <pre> PCARAC = ^NOCARAC; NOCARAC = RECORD DADO : RCARAC; PROXIMO : PCARAC; END;</pre>
<p>Passo 21 – implementar a classe que vai gerenciar a fila de caracteres. O nome da classe é Tcarac. Essa classe herda todas as características de Tlr1 e é responsável pelo armazenamento de cadeias em uma fila. Essa classe possui um procedimento chamado “Inclui” que inclui uma nova cadeia na fila. Possui também uma função chamada linha.</p>

Essa função precisa de um argumento do tipo inteiro e retorna a enésima cadeia armazenada na fila de cadeias. O índice da cadeia é dado pelo argumento inteiro. Possui também a função max que retorna o número de cadeias na fila.

O texto abaixo mostra sua implementação em pascal.

```
TCARAC = CLASS(TLR1)
PRIVATE
    Q : WORD;
PROTECTED
    PRIMEIRO, ATUAL : PCARAC;
PUBLIC
    CONSTRUCTOR CREATE;
    PROCEDURE LINICIA;
    PROCEDURE INCLUI(X : STRING);
    FUNCTION LINHA(X : WORD) : STRING;
    FUNCTION MAX : WORD;
END;
IMPLEMENTATION
CONSTRUCTOR TCARAC.CREATE;
BEGIN
    INHERITED;
    LINICIA;
END;
PROCEDURE TCARAC.LINICIA;
BEGIN
    NEW(PRIMEIRO);
    ATUAL := PRIMEIRO;
    ATUAL^.DADO.DADO := 'GORN';
    ATUAL^.PROXIMO := NIL;
    Q := 0;
END;
PROCEDURE TCARAC.INCLUI(X : STRING);
BEGIN
    NEW(ATUAL^.PROXIMO);
    ATUAL := ATUAL^.PROXIMO;
    ATUAL^.DADO.DADO := X;
    ATUAL^.PROXIMO := NIL;
    INC(Q);
END;
FUNCTION TCARAC.LINHA(X : WORD) : STRING;
LABEL
    FIM;
VAR
    PERCORRE : PCARAC;
    I : WORD;
BEGIN
    IF (X>Q) THEN
    BEGIN
        RESULT := 'ACIMA DO LIMITE';
        GOTO FIM;
    END;
    PERCORRE := PRIMEIRO;
    FOR I:= 1 TO X DO
        PERCORRE := PERCORRE^.PROXIMO;
```

```

    RESULT := PERCORRE^.DADO.DADO;
    FIM;
END;
FUNCTION TCARAC.MAX : WORD;
BEGIN
    RESULT := Q;
END;
END.

```

Passo 22 – implementar uma classe que possui procedimentos que permitam a busca de palavras. Tcarac1 é o nome da classe que herda todas as características de tcarac e possui quatro procedimentos de procura, que vai facilitar na hora da leitura do *trail*.

O texto abaixo mostra a implementação dessa classe.

```

INTERFACE
USES UHCARAC;
TYPE
TCARAC1 = CLASS(TCARAC)
PUBLIC
    FUNCTION PROCURA(X : WORD; Y : STRING) : WORD;
    FUNCTION PROCURAP(X : WORD; Y : STRING) : WORD;
    FUNCTION APROCURA(X : WORD; Y : STRING) : WORD;
    FUNCTION APROCURAP(X : WORD; Y : STRING) : WORD;
END;
IMPLEMENTATION
USES DIALOGS;
FUNCTION TCARAC1.PROCURA(X : WORD; Y : STRING) : WORD;
VAR
    I : WORD;
BEGIN
    I := X;
    WHILE (LINHA(I) <> Y) AND (I <= MAX) DO
        BEGIN
            //      SHOWMESSAGE(LINHA(I));
            INC(I);
        END;
    IF I <= MAX THEN
        RESULT := I
    ELSE
        RESULT := 0;
    END;
FUNCTION TCARAC1.PROCURAP(X : WORD; Y : STRING) : WORD;
VAR
    I : WORD;
    S : STRING;
BEGIN
    I := X;
    S := COPY(LINHA(I), 1, LENGTH(Y));
    WHILE (S <> Y) AND (I <= MAX) DO
        BEGIN
            INC(I);
            S := COPY(LINHA(I), 1, LENGTH(Y));
        END;
    IF I <= MAX THEN
        RESULT := I
    ELSE
        RESULT := 0;
    END;

```



```

    ELSE
    RESULT := 0;
END;
FUNCTION TCaracI.APROCURA(X : WORD; Y : STRING) : WORD;
VAR
    I : WORD;
BEGIN
    I := X;
    WHILE (I>0) AND (LINHA(I)<>Y) DO
    BEGIN
        DEC(I);
    END;
    RESULT := I;
END;
FUNCTION TCaracI.APROCURAP(X : WORD; Y : STRING) : WORD;
VAR
    I : WORD;
    S : STRING;
BEGIN
    I := X;
    S := "";
    WHILE (I>0) AND (S<>Y) DO
    BEGIN
        DEC(I);
        S := COPY(LINHA(I), 1, LENGTH(Y));
    END;
    RESULT := I;
END;
END.

```

Passo 23 – implementar a classe capaz de alimentar a fila de strings com os caracteres do *trail* no disco. TleText é a classe herdeira de TCaracI que possui procedimentos para carregar as linhas do arquivo *trail* na fila de cadeias e pode salvar essas linhas em um determinado arquivo.

O texto abaixo demonstra a implementação dessa classe em Pascal.

```

TLETEXT = CLASS (TCARACI)
PRIVATE
PROTECTED
PUBLIC
    PROCEDURE TEXTPVIRTUAL(X : STRING);
    PROCEDURE SALVA(X, Y : STRING);
    {SALVA AS LINHA NO ARQUIVO X ATÉ ENCONTRAR A LINHA Y}
    PROCEDURE SALVAS(X : STRING; Y, Z : WORD);
    {SALVA AS LINHAS DE NÚMERO Y ATÉ A DE NÚMERO Z NO ARQUIVO X}
END;

```

O procedimento textpvirtual recebe um argumento do tipo cadeia que vai informar o nome e o caminho para o arquivo *trail* a ser lido. Esse procedimento possui um algoritmo que carrega os caracteres do arquivo em velocidade otimizada tornando quase imperceptível, para o usuário, no momento em que o arquivo é carregado. Essa metodologia foi utilizada

devido às inúmeras leituras que o programa precisa fazer no arquivo. Existem ainda dois procedimentos usados para salvar as linhas do *trail* virtual (informações do arquivo na memória). O procedimento salva possui dois argumentos cadeia. Um dos argumentos informa o nome do arquivo. O outro argumento informa que o programa deverá parar de salvar quando encontrar uma linha igual a esse argumento. O procedimento salvass possui um argumento cadeia e dois argumentos inteiros. O argumento cadeia informa o nome do arquivo a ser criado. Os argumentos inteiros informam a primeira e a última linha a ser salva, respectivamente.

Passo 24 – implementar a classe que vai localizar o ultimo *trail* criado pelo PRO-ENGINEER.

Tletext1 – é a classe herdeira de Tletext. Essa classe possui dois procedimentos que vão procurar qual foi o último *trail* criado pelo PRO-E. O primeiro procedimento, chamado peclast, procura no diretório de trabalho pelo arquivo *trail.txt.1*, se encontrar procura pelo *trail.txt.2*, se não encontrar o *trail.txt.1* é o ultimo. A figura 1 mostra o algoritmo

Figura 1 – algoritmo para localizar último *trail*.

O segundo procedimento, chamado peclast2, para procura do último *trail* carrega em uma fila, de variáveis cadeia, os nomes de todos os arquivos do diretório de trabalho. Depois coloca em outra fila de cadeias, apenas os caracteres que estão depois do segundo ponto, para todos o nomes de arquivo *trail* contidos na fila anterior. Depois ele usa um

algoritmo de ordenação para descobrir qual cadeia representa o maior valor numérico. O arquivo correspondente a essa cadeia é o último *trail*. O diagrama da figura 2 mostra o algoritmo de ordenação.

Figura 2 – algoritmo de ordenação.

O texto abaixo mostra parte da implementação, em pascal, dessa classe.

```

TYPE
TLETEXT1 = CLASS(TLETEXT)
PRIVATE
PROTECTED
PUBLIC
    FUNCTION PECLAST: STRING;
    FUNCTION PECLAST2(X: STRING): STRING;
END;
IMPLEMENTATION
USES SYSUTILS;
FUNCTION TLETEXT1.PECLAST: STRING;
VAR

```



```

        CONTINUAR := TRUE;
    WHILE CONTINUAR DO
    BEGIN
        ASSIGNFILE(F,NTRAI+INTTOSTR(EXT));
        {$I-} F : FILE OF BYTE;
    EXT : WORD;
    CONTINUAR : BOOLEAN;
BEGIN
    EXT := 1;
    RESET(F);
    {$I+}
    IF IORESULT <> 0 THEN
    BEGIN
        CONTINUAR := FALSE;
    END
    ELSE CLOSEFILE(F);
        INC(EXT);
    END;
    RESULT := (NTRAI+INTTOSTR(EXT-1));
//    SHOWMESSAGE(RESULT);
END;

FUNCTION TLETEXT1.PECLAST2(X : STRING) : STRING;
VAR
    NOMES, NÚMEROS : TLETEXT1;
    DIRINFO : TSEARCHREC;
    I : WORD;
    J : BYTE;
    K : INTEGER;
    S, M : STRING;
BEGIN
    NOMES := TLETEXT1.CREATE;
    FINDFIRST(X, $01, DIRINFO);
    K := 0;
    WHILE K = 0 DO
    BEGIN
        NOMES.INCLUI(DIRINFO.NAME);
        K := FINDNEXT(DIRINFO);
    END;
    NUMEROS := TLETEXT1.CREATE;
    FOR I := 1 TO NOMES.MAX DO
    BEGIN
        S := "";
        FOR J := 11 TO LENGTH(NOMES.LINHA(I)) DO
            S := S + NOMES.LINHA(I)[J];
        TRY
//            N1 := STRTOINT(S);
            NUMEROS.INCLUI(S);
        EXCEPT
        END;
    END;
    M := NUMEROS.LINHA(1);
    FOR I := 2 TO NUMEROS.MAX DO
    BEGIN
        IF STRTOINT(NÚMEROS.LINHA(I)) > STRTOINT(M) THEN
            M := NÚMEROS.LINHA(I);
    
```

```

END;
RESULT := NTRAI+M;
//  SHOWMESSAGE(RESULT);
END;

```

Passo 25 – implementar a classe chamada *tlrm*. Essa classe herda todas as características de *tletext1*.

O código abaixo mostra a implementação dessa classe em pascal.

```

TYPE
TLRM = CLASS(TLETTEXT1)
END;

```

Reconhecimento.

Implementou-se algumas classes que possuem procedimentos capazes de percorrer as linhas do arquivo *trail*, colocadas na fila de cadeias (ver implementação de “carregar arquivo *trail*”) e reconhecer as *features* do desenho.

Metodologia para implementação do reconhecimento

Passo 26 – implementar uma classe que verifica se o usuário criou o 3 planos ortogonais (*Datum plane*). *Tlrm1* é a classe herdeira de *tlrm* que possui um procedimento para verificar se o usuário desenhou um datum plane no seu projeto. Sem o datum plane o número de *features* manipuladas pelo SEMF fica mais limitado. O nome do procedimento que verifica a existência da dtum é o *procdatum*. Esse procedimento usa os procedimentos de procura (implementados no passo 22) para verificar a existência das cadeias: #PLANE, #DATUM, #PLANE, #DEFAULT, !%CIDATUM PLANE HAS BEEN CREATED SUCCESSFULLY. Colocadas nessa ordem, no arquivo *trail*, elas comprovam que o usuário desenhou os 3 planos ortogonais. Possui também uma função que informa o módulo de um vetor de até três dimensões.

O código abaixo mostra a implementação, em linguagem pascal, dessa classe.

```

TLRM1 = CLASS(TLRM)
PRIVATE
PROTECTED
    CDATUM : BOOLEAN;
    K : WORD;
PUBLIC
    PROCEDURE PROCDATUM;
    FUNCTION MODULO1V(X, Y, Z : EXTENDED) : EXTENDED;
    FUNCTION MODULO3V(X, Y, Z : RPONTO) : EXTENDED;
    FUNCTION MODULO3VV(X : ARRAY OF EXTENDED) : EXTENDED;
END;
VAR
    LRM1 : TLRM1;

```

```

IMPLEMENTATION
USES Dialogs;
PROCEDURE TLRM1.PROCDATUM;
BEGIN
  IF PROCURA(1, '#PLANE') <> 0 THEN
  BEGIN
    K := PROCURA(1, '#DATUM');
    IF LINHA(K+1) = '#PLANE' THEN
    BEGIN
      IF LINHA(K+2) = '#DEFAULT' THEN
      BEGIN
        IF LINHA(K+3) = '!%CIDATUM PLANE HAS BEEN CREATED SUCCESSFULLY.' THEN
        BEGIN
          CDATUM := TRUE;
        END;
      END;
    END;
  END;
  //      SHOWMESSAGE('EEEEEEEE');
  END;
  END;
  END
  ELSE
  //      SHOWMESSAGE('ERRO');
  END;
FUNCTION TLRM1.MODULO1V(X, Y, Z : EXTENDED) : EXTENDED;
BEGIN
  RESULT := SQRT(SQR(X)+SQR(Y)+SQR(Z));
END;
FUNCTION TLRM1.MODULO3V(X, Y, Z : RPONTO) : EXTENDED;
BEGIN
  RESULT := EXP((1/3)*LN(MODULO1V(X.X, X.Y, X.Z)*MODULO1V(Y.X, Y.Y, Y.Z)*MODULO1V(Z.X, Z.Y, Z.Z)));
END;
FUNCTION TLRM1.MODULO3VV(X : ARRAY OF EXTENDED) : EXTENDED;
BEGIN
  RESULT := SQRT(SQR(MODULO1V(X[1], X[2], X[3]))+SQR(MODULO1V(X[4], X[5], X[6]))+SQR(MODULO1V(X[7], X[8], X[9])));
END;

```

Passo 27 – implementar a Tlrm2 que é uma classe que herda as características de tlrml e possui procedimentos e funções auxiliares. A função ppv, por exemplo, pega uma cadeia com ponto decimal e troca o ponto por uma vírgula. A função vcolch retorna um valor que está escrito entre colchetes, em uma cadeia. Por exemplo, vcolch ('valor encontrado [22.36]') vai funcionar como o valor 22.36.

```

TYPE
TLRM2 = CLASS(TLRM1)
PRIVATE
PROTECTED
  VISAO : ARRAY[1..3, 1..3] OF EXTENDED;
  { PEGA O PONTO DE VISTA EM UMA CERTA LINHA XX DO TRAIL }
  PROCEDURE CONVVISAO1(XX : WORD; VAR A1, A2, A3, A4, A5, A6, A7, A8, A9 : EXTENDED);
  { PEGA O PONTO DE VISTA EM UMA CERTA LINHA XX DO TRAIL }
  PROCEDURE CONVVISAO2(XX : WORD; VAR P1, P2, P3 : SHORTINT);
  { TROCA OS PONTOS DO TEXTO POR VIRGULAS }
  FUNCTION PPV(X : STRING) : STRING;

```


<pre> { RETORNA O VALOR REAL ENTRE COLCHTES DO TEXTO X } FUNCTION VCOLCH(X : STRING) : EXTENDED; { RETORNA O PONTO CORRESPONDENTE A STRING DADA } FUNCTION STRTOPONTO(X : STRING) : RPONTO; { RETORNA O PONTO CORRESPONDENTE A STRING COM * DADA } FUNCTION STRTOPONTOA(X : STRING) : RPONTO; PUBLIC END;</pre>
<p>Passo 28 – implementar Tlrm21. Esta classe é uma classe herdeira de tlr2 e possui duas funções. A primeira função recebe, como argumento, os dois pontos, os vértices contrários de um retângulo, e a escala do desenho. Essa função dá como respostas todas as outras informações do retângulo, como coordenadas dos quatro vértices e valor da dimensão dos quatro lados do retângulo. A outra função recebe, como argumento, dois pontos tridimensionais e retorna a distância entre eles em formato de número real.</p> <p>Abaixo está parte do código dessa classe.</p> <pre> TREC = ARRAY[1..4] OF RPONTO; TREC1 = RECORD V, B : TREC; D12, D23, EXT : EXTENDED; END; TLRM21 = CLASS(TLRM2) PROTECTED FUNCTION RECT(P1, P2 : RPONTO; ESCALA : EXTENDED) : TREC1; FUNCTION DISTPP(P1, P2 : RPONTO) : EXTENDED; END;</pre>
<p>Passo 29 – implementar Tlrm22, é uma classe herdeira de tlr21 que possui uma função que converte milímetros para polegadas. Essa função é necessário porque o PRO-ENGINEER trabalha com polegadas como unidade padrão.</p> <pre> TYPE TLRM22 = CLASS(TLRM21) PUBLIC { CONVERTE MILÍMETRO EM POLEGADA } FUNCTION MMTOINC(X : EXTENDED) : EXTENDED; END; IMPLEMENTATION FUNCTION TLRM22.MMTOINC(X : EXTENDED) : EXTENDED; BEGIN RESULT := X/25.4; END;</pre>
<p>Passo 30 – implementar uma classe herdeira de Tlrm22 e possui um procedimento chama de Prot1 que recebe, como argumento, um valor inteiro, e retorna as informações sobre um <i>protrusion</i> desenhado a partir do rascunho de um retângulo. O argumento inteiro vai indicar a linha do <i>trail</i> onde está a cadeia “#PROTRUSION”. Essa é a função mais importante do</p>

processo de reconhecimento do sólido. O nome dessa classe é *tlrn3*. Parte de seu código e o código do registro *infprismrec*, que é o tipo do valor de retorno da função, são mostrados no código a seguir.

```

INFPRISMREC = RECORD
  OK : BOOLEAN;
  COTHR, //COTA HORIZONTAL DA REFERÊNCIA
  COTVR, //COTA VERTICAL DA REFERÊNCIA
  COTHP, //COTA HORIZONTAL DO RETÂNGULO
  COTVP, //COTA VERTICAL DO RETÂNGULO
  COTEX //COTA DE EXTRUSÃO
  : REAL;
  PV1, //PLANO X
  PV2, //PLANO Y
  PV3 //PLANO Z
  : SHORTINT;
  CV, //TIPO DA COTA DE REFERÊNCIA VERTICAL
  CH //TIPO DA COTA DE REFERÊNCIA HORIZONTAL
  : BYTE;
END;
TLRM3 = CLASS(TLRM22)
PUBLIC
  FUNCTION PROT1(X : WORD{; VAR RRESULT : INFPRISMREC}) : INFPRISMREC;
END;

```

Passo 31 – implementar uma classe, herdeira de *tlrn3*, que possui uma função para o reconhecimento do corte. *Tlrm31* é a classe herdeira de *tlrn3* e possui uma função análoga a *prot1* de *Tlrm3*. Esse procedimento recebeu o nome de *cut1*. Recebe, como argumento, um valor inteiro e retorna as informações sobre um cut desenhado a partir do rascunho de um retângulo. O argumento inteiro vai indicar a linha do *trail* onde está a cadeia “#CUT”.

Parte do código dessa classe é mostrado no texto abaixo.

```

INFCUT = INFPRISMREC;
TLRM31 = CLASS(TLRM3)
PUBLIC
  FUNCTION CUT1(X: WORD): INFCUT;
END;

```

O valor que essa função retorna é igual ao valor que *prot1* retorna, por que o corte e o sólido tem formato semelhante.

Passo 32 – implementar uma classe com um procedimento para reconhecer um chanfro. *Tlrm33* é o nome da classe herdeira de *tlrn32* que possui um procedimento chamado *chanf45xd1*. Recebe, como argumento, um valor inteiro e retorna as informações sobre um chanfro de 45 graus. O argumento inteiro vai indicar a linha do *trail* onde está a cadeia “#CHAMFER” que indica que foi iniciada a construção da *feature*.

O código abaixo mostra parte da implementação em linguagem pascal.

```

TYPE
INFCHANF45XD1 = RECORD
  DIM //DIMENSÃO DO CHANFRO
  : REAL;
  OK : BOOLEAN;
END;
TLRM33 = CLASS(TLRM32)
PRIVATE
PROTECTED
PUBLIC
  FUNCTION CHANF45XD1(X : WORD) : INFCHANF45XD1;
END;

```

Passo 33 – implementar uma classe com um procedimento para reconhecer um arredondamento. Tlrm34 é o nome da classe herdeira de tlr33 que possui um procedimento chamado round1. Recebe, como argumento, um valor inteiro e retorna as informações sobre um arredondamento. O argumento inteiro vai indicar a linha do *trail* onde está a cadeia “#ROUND” que indica que foi iniciada a construção da *feature*. O código abaixo mostra parte da implementação em linguagem pascal.

```

TYPE
INFROUND1 = RECORD
  DIM //DIMENSÃO DO CHANFRO
  : REAL;
  OK : BOOLEAN;
END;
TLRM34 = CLASS(TLRM33)
PRIVATE
PROTECTED
PUBLIC
  FUNCTION ROUND1(X : WORD) : INFROUND1;
END;

```

Passo 34 – implementar uma classe com um procedimento para reconhecer um furo. Tlrm4 é o nome da classe herdeira de tlr34 que possui um procedimento chamado hole1. Recebe, como argumento, um valor inteiro e retorna as informações sobre um furo. O argumento inteiro vai indicar a linha do *trail* onde está a cadeia “#HOLE” que indica que foi iniciada a construção da *feature*. O código abaixo mostra parte da implementação em linguagem pascal.

```

TYPE
INFHOLE = RECORD
  OK : BOOLEAN;
  PD1, //PAREDE1
  PD2, //PAREDE2
  DIA, //DIAMETRO
  ALTURA
  : REAL;
END;
TLRM4 = CLASS(TLRM34)

```



```

PRIVATE
PROTECTED
PUBLIC
    FUNCTION HOLE1(X : WORD) : INFHOLE;
END;

```

Regras.

A próxima etapa na construção do SEMF é a criação de classes capazes de percorrer a APD e aplicar as regras de fundição nas *features* cadastradas. Alguns desses procedimentos informam ao usuário se houve ou não violação de regras e exibe uma recomendação. Os textos exibidos por esses procedimentos ficam escritos em arquivos ASCII e podem ser personalizados. Todo texto de saída de regra, ou preferência, é escrito da seguinte forma: prefixo|texto principal. Ou seja, o prefixo é separado do texto principal pelo caracter “|”. A função, ou procedimento, de regra utiliza esse prefixo para encontrar o texto desejado. Assim, o usuário precisa digitar, nesse arquivo, todos os prefixos seguidos do caracter ”|” e do texto principal. Os prefixos são:

EspMinParedeFe – para espessura mínima de parede para ferro fundido.

EspMinParedeAl – para espessura mínima de parede para Alumínio.

DiaMinFuro – dimensão mínimo para furo prismático.

DimMaxChanfro – dimensão máxima do chanfro.

SecMinFuro – seção mínima do furo.

DiaMen19 – diâmetro menor que 19 mm.

NoViolac – para *feature* sem violação.

Metodologia para implementação das regras

Passo 35 – implementar a classe `tlrm5`. Essa classe é herdeira de `tlrm4` e possui uma função, chamada `pref`, que retorna o texto de uma regra ou preferência. As regras e preferências ficam escritas em editores de texto do SEMF. Essa função recebe um argumento texto (string), que será o prefixo texto, e um argumento do tipo `tedit` (uma classe do Delphi 4 para editor de texto) para indicar o editor do qual sairá o texto. O resultado da função é o texto principal relativo ao prefixo, ou, o texto: 'NÃO FOI ENCONTRADO O TEXTO DE PREFIXO'. Caso o prefixo procurado não exista.

O código abaixo mostra a implementação dessa classe.

```

TYPE
TLRM5 = CLASS(TLRM4)
PUBLIC
    FTUDO: BOOLEAN;
    PROCEDURE SETTUDO(CONST VALUE: BOOLEAN);
PUBLISHED
PROPERTY TUDO : BOOLEAN READ FTUDO WRITE SETTUDO;
PROTECTED
PUBLIC
    FUNCTION PREF( X : STRING; Y : TMEMO) : STRING;
    FUNCTION NOVIOLAC : STRING;
//  RETORNA O TEXTO DE UMA REGRA OU PREFERÊNCIA EDITADA
PUBLIC
END;
IMPLEMENTATION
USES DIALOGS, SYSUTILS, EDR1, UNIT1;
FUNCTION TLRM5.NOVIOLAC: STRING;
BEGIN
    RESULT := PREF('NoViolac', FORM2.MEMO1);
{ IF NOT FTUDO THEN
    RESULT := RESULT +
    ' PARA ' +
    FORM1.L2.ITEMS.STRING[FORM1.L2.ITEMINDEX];}
END;
FUNCTION TLRM5.PREF(X: STRING; Y : TMEMO): STRING;
VAR
    S : STRING;
    I,
    J
    : WORD;
BEGIN
    I := 1;//PERCORRE AS LINHAS DO MEMORANDO
    J := 1;
    S := "";
    WHILE (I < Y.LINES.COUNT)AND(S <> X)AND(X<>"") DO
        BEGIN
            J := 1;//PERCORRE OS CARACTERES DA LINHA[I] DO MEMORANDO
            S := "";
            WHILE ( J < LENGTH(Y.LINES.STRING[I]) )AND(Y.LINES.STRING[I][J]<>"") DO
                BEGIN
                    S := S + Y.LINES.STRING[I][J];
                    INC(J);
                END;
                INC(I);
            END;
//  SHOWMESSAGE(S+' '+Y.LINES.STRING[I-1]+INTTOSTR(I));
            RESULT := S;
            IF (X <> "")AND(S = X) THEN
                FOR J := (J+1) TO LENGTH(Y.LINES.STRING[I-1]) DO
                    BEGIN
                        RESULT := RESULT + Y.LINES.STRING[I-1][J];
                    END
                ELSE
                    RESULT := 'NÃO FOI ENCONTRADO O TEXTO DE PREFIXO '+X;
            END;
        END;
    PROCEDURE TLRM5.SETTUDO(CONST VALUE: BOOLEAN);

```

```

BEGIN
  FTUDO := VALUE;
END;LRM5 = CLASS(TLRM4)
PUBLIC
  FTUDO: BOOLEAN;
  PROCEDURE SETTUDO(CONST VALUE: BOOLEAN);
PUBLISHED
PROPERTY TUDO : BOOLEAN READ FTUDO WRITE SETTUDO;
PROTECTED
PUBLIC
  FUNCTION PREF( X : STRING; Y : TMEMO) : STRING;
  FUNCTION NOVIOLAC : STRING;
//  RETORNA O TEXTO DE UMA REGRA OU PREFERÊNCIA EDITADA
PUBLIC
END;

```

Passo 36 – implementar a classe para checar as regras de parede mínima para o prisma sólido e cilindro sólido. O procedimento `checkparedemin` recebe um argumento inteiro, que representa o material usado (1 – Alumínio, 2 – Ferro fundido). Esse procedimento percorre os nós da LR até encontrar um que aponte para a *feature* de tipo 1 (prisma sólido) e ID igual ao nome selecionado pelo usuário. Encontrada a *feature* esse procedimento verifica na fila de reais da ID da *feature* se existe, entre os nós da fila de real correspondentes as dimensões da *feature*, um valor menor que o mínimo recomendado. Se existir o procedimento exibe o texto de recomendação. Se não, o procedimento exibe um texto dizendo que não há violações. O procedimento `CHECKPAREDEMIN` verifica erros de espessura de parede em um sólido determinado. O procedimento `TCHECKPAREDEMIN` verifica erros de espessura de parede em todos os sólidos da APD. O procedimento `CHECHPAREDEMINE` verifica erros de espessura de parede em um sólido cilíndrico determinado. O procedimento `TCHECHPAREDEMINE` verifica erros de espessura de parede em todos os sólidos cilíndricos da APD.

```

TYPE
TLRM1_If = CLASS(TLRM5)
PUBLIC
PROTECTED
  PROCEDURE CHECKPAREDEMIN(X : BYTE);
  PROCEDURE TCHECKPAREDEMIN(X : BYTE);
  PROCEDURE CHECHPAREDEMINE(X : BYTE);
  PROCEDURE TCHECHPAREDEMINE(X : BYTE);
END;

```

Passo 37 – implementar uma classe com procedimentos que encontrem violações em um corte prismático. O nome da classe é `tlrm1_2f` e possui procedimentos para verificar se o corte possui dimensões menores que a mínima recomendada. Verifica também se o corte

deixa uma espessura de parede menor que a mínima recomendada.

Parte do código dessa classe é mostrado abaixo.

```
TYPE
TLRM1_2F = CLASS(TLRM1_1F)
  PUBLIC
  PROTECTED
    PROCEDURE CHECKCUTCOTA;
    PROCEDURE TCHECKCUTCOTA;
    PROCEDURE CHECKCUTPAREDE(X : WORD);
    PROCEDURE TCHECKCUTPAREDE(X : WORD);
  END;
```

Passo 38 – implementar a classe `tlrm1_3f`, herdeira de `tlrm1_2f`, com procedimentos capazes de encontrar violações nas *features* do tipo chanfro. Parte da classe é mostrada a seguir.

```
TYPE
TLRM1_3F = CLASS(TLRM1_2F)
  PRIVATE
  PROTECTED
    PROCEDURE CHECKCHANF1;
    PROCEDURE TCHECKCHANF1;
  PUBLIC
  END;
```

Passo 39 – implementar a classe chamada `tlrm1_4f` apenas para ser herdeira de `tlrm1_3f`.

Como mostra seu código.

```
TYPE
TLRM1_4F = CLASS(TLRM1_3F)
  END;
```

Passo 40 – implementar a classe `tlrm1f`, herdeira de `tlrm1_4f`, dotada de procedimentos com regras para furo. Seu código é mostrado a seguir.

```
TLRM1F = CLASS(TLRM1_4F)
  PROTECTED
    PROCEDURE CHECKHOLEDIA;
    PROCEDURE CHECKHOLEDIA2;
    PROCEDURE CHECKHOLEPAREDE(X : BYTE);
    PROCEDURE TCHECKHOLEDIA;
    PROCEDURE TCHECKHOLEDIA2;
    PROCEDURE TCHECKHOLEPAREDE(X : BYTE);
  PUBLIC
  END;
```

PROCEDURE CHECKHOLEDIA – Verifica se o diâmetro do furo é menor que o recomendado.

PROCEDURE CHECKHOLEDIA2 – Verifica se o diâmetro do furo é menor que 19 mm.

PROCEDURE CHECKHOLEPAREDE(X : BYTE) – Verifica se a espessura de parede deixada, no sólido, pelo furo é menor que a recomendada. Esse procedimento recebe um argumento inteiro (ver fundamentos de programação) que indica o material da *feature*.

Os procedimentos com nome começado por T verificam todas as *features* da APD. Já os 3 procedimentos acima verificam apenas uma *feature* indicada pelo usuário.

Passo 41 – implementar a classe que é herdeira de `tlrm1f` e possui um procedimento que vai percorrer a APD e mostrar ao usuário os campos ID das UI correspondentes às *features* existentes na APD. Essa classe também possui um procedimento que, em resposta a um clique de mouse no nome da *feature*, exhibe as regras disponíveis para a *feature* selecionada.

Os nomes desses procedimentos são `totalcheck` e `totalmostra`, respectivamente. O código, em pascal, de parte dessa classe é mostrado a seguir.

```
TYPE  
TLRM2F = CLASS(TLRM1F)  
PUBLIC  
  PROCEDURE TOTALCHECK(X : WORD);  
  PROCEDURE TOTALMOSTRA(X : STRING);  
  FUNCTION PEGANOME(X : STRING) : STRING;  
END;
```

`totalcheck` exhibe os campos ID na janela do SEMF mostrada em vermelho na figura 3.

Figura 3 – Área de trabalho do SEMF. Os procedimentos de regra escrevem o diagnóstico no campo destacado pela seta azul.)

A seta verde mostra o local onde o procedimento TOTALMOSTRA escreve as regras para a *feature* selecionada.

Passo 42 – implementar `tlrm3f`, uma classe herdeira de `TLRM2F` e possui um procedimento chamado `escolheregra`. Esse procedimento usa informações dadas pelo usuário para acionar a regra correta. O código abaixo mostra a implementação dessa classe em pascal.

```
TYPE
TLRM3F = CLASS(TLRM2F)
PUBLIC
  PROCEDURE ESCOLHEREGRA(X, Y: STRING);
END;
IMPLEMENTATION
USES UNIT1, EDR2, DIALOGS;
PROCEDURE TLRM3F.ESCOLHEREGRA(X, Y: STRING);
BEGIN
  FORM1.MEMO1.CLEAR;
  { ESPESSURA MÍNIMA PERMITIDA PARA AS COTAS DO SÓLIDO }
  IF ( X = PREF('EspMinPAREDE', FORM3.MEMO1) )
    AND
    (Y = 'PRISMA') THEN
  BEGIN
    IF FORM1.ALUMINIO1.CHECKED THEN
      CHECKPAREDEMIN(1) ELSE
    IF FORM1.FERROFUNDIDO1.CHECKED THEN
      CHECKPAREDEMIN(2) ELSE
    END ELSE
  { ESPESSURA MÍNIMA PERMITIDA PARA AS COTAS DO SÓLIDO CILINDRICO }
  IF (X = PREF('EspMinPAREDE', FORM3.MEMO1) )//FORM3.MEMO1.LINES.STRING[10])
    AND
    (Y = 'CILIND') THEN
  BEGIN
    IF FORM1.ALUMINIO1.CHECKED THEN
      CHECPAREDEMINC(1) ELSE
    IF FORM1.FERROFUNDIDO1.CHECKED THEN
      CHECPAREDEMINC(2) ELSE
    END ELSE
  { DIÂMETRO MÍNIMO PERMITIDO PARA UM FURO }
  IF (X = PREF('SecMinFURO', FORM3.MEMO1) )//FORM3.MEMO1.LINES.STRING[5])
    AND
    (Y = 'FURO C') THEN
  BEGIN
    CHECKHOLEDIA;
  END ELSE
  { DIÂMETRO MÍNIMO 19 PERMITIDO PARA UM FURO }
  IF (X = PREF('DimMen19', FORM3.MEMO1) )//FORM3.MEMO1.LINES.STRING[9])
    AND
    (Y = 'FURO C') THEN
  BEGIN
    CHECKHOLEDIA2;
  END ELSE
  { COTA ENTRE A PAREDE DO SÓLIDO E O RAO DE UM FURO CILINDRICO }
  IF (X = PREF('EspMinPAREDE', FORM3.MEMO1) )//FORM3.MEMO1.LINES.STRING[6])
    AND
```

<pre> (Y = 'FURO C') THEN BEGIN IF FORM1.ALUMINIO1.CHECKED THEN CHECKHOLEPAREDE(1) ELSE IF FORM1.FERROFUNDIDO1.CHECKED THEN CHECKHOLEPAREDE(2) ELSE END ELSE { COTA MÍNIMA ADMISSÍVEL PARA UM FURO PRISMÁTICO RETO DE BASE RETANGULAR } IF (X = PREF('SecMINFURO', FORM3.MEMO1))//FORM3.MEMO1.LINES.STRING[7]) AND (Y = 'FURO R') THEN BEGIN CHECKCUTCOTA; END ELSE { PAREDE CUT } IF (X = PREF('EspMINPAREDE', FORM3.MEMO1))//FORM3.MEMO1.LINES.STRING[8]) AND (Y = 'FURO R') THEN BEGIN IF FORM1.ALUMINIO1.CHECKED THEN CHECKCUTPAREDE(1) ELSE IF FORM1.FERROFUNDIDO1.CHECKED THEN CHECKCUTPAREDE(2) ELSE END ELSE { DIMENSÃO DO CHANFRO } IF (X = PREF('DimMAXCHANFRO', FORM3.MEMO1))//FORM3.MEMO1.LINES.STRING[8]) AND (Y = 'CHANFR') THEN BEGIN CHECKCHANF1; END ELSE END; </pre>	<p>O procedimento <i>escolheregra</i> aceita dois parâmetros (argumentos) do tipo <i>string</i> (texto), o argumento x e o argumento y. X será a violação que o usuário deseja verificar e y será as seis primeiras letras do nome da <i>feature</i>. Ele recebe as seis primeiras letras porque até a sexta letra o ID de toda <i>feature</i>, do mesmo tipo, é igual. Daí, “sabendo” o tipo de <i>feature</i> e a regra desejada, o procedimento executa o procedimento específico para aquele caso.</p>
<p>Exemplo:</p> <pre> {SE A REGRA ESCOLHIDA FOR A ESPESSURA DE PAREDE MÍNIMA E AS 6 PRIMEIRAS LETRAS FOREM P R I S M A E O MATERIAL UTILIZADO FOR ALUMINIO ENTÃO O PROCEDIMENTO <u>CHECKPAREDEMIN (PASSO 36)</u> DEVE SER EXECUTADO COM PARÂMETRO 1} IF (X = PREF('EspMINPAREDE', FORM3.MEMO1)) AND (Y = 'PRISMA') THEN BEGIN IF FORM1.ALUMINIO1.CHECKED THEN CHECKPAREDEMIN(1) ELSE {SENÃO SE A REGRA ESCOLHIDA FOR A ESPESSURA DE PAREDE MÍNIMA E AS 6 PRIMEIRAS LETRAS FOREM P R I S M A E O MATERIAL UTILIZADO FOR FERRO FUNDIDO ENTÃO O PROCEDIMENTO <u>CHECKPAREDEMIN</u> </pre>	

DEVE SER EXECUTADO COM PARÂMETRO 2}

IF FORM1.FERROFUNDIDO1.CHECKED THEN

CHECKPAREDEMIN(2) ELSE

END ELSE

{SENÃO SE A REGRA ESCOLHIDA FOR A ESPESSURA DE PAREDE MÍNIMA E AS 6 PRIMEIRAS LETRAS FOREM C I L I N D E O MATERIAL UTILIZADO FOR ALUMINIO ENTÃO O PROCEDIMENTO CHECKPAREDEMINC DEVE SER EXECUTADO COM PARÂMETRO 1}

{ ESPESSURA MÍNIMA PERMITIDA PARA AS COTAS DO SÓLIDO CILINDRICO }

IF (X = PREF('EspMinPAREDE', FORM3.MEMO1))//FORM3.MEMO1.LINES.STRING[10])

AND

(Y = 'CILIND') THEN

BEGIN

IF FORM1.ALUMINIO1.CHECKED THEN

CHECHPAREDEMINC(1) ELSE

{SENÃO SE A REGRA ESCOLHIDA FOR A ESPESSURA DE PAREDE MÍNIMA E AS 6 PRIMEIRAS LETRAS FOREM C I L I N D E O MATERIAL UTILIZADO FOR FERRO FUNDIDO ENTÃO O PROCEDIMENTO CHECKPAREDEMINC DEVE SER EXECUTADO COM PARÂMETRO 2}

IF FORM1.FERROFUNDIDO1.CHECKED THEN

CHECHPAREDEMINC(2) ELSE

END ELSE

Passo 43 – A classe implementada nesse passo se chama tlrnf. Essa classe é herdeira de tlrnf3f. Ela possui 3 procedimentos: percorre, checar, checatudo conforme o código abaixo.

TLRMF = CLASS(TLRNF3F)

PRIVATE

W//NÚMERO DE FEATURES

: WORD;

PUBLIC

FUNCTION PERCORRE : WORD;

PROCEDURE CHECAR;

PROCEDURE CHECATUDO;

END;

O procedimento *percorre*. O procedimento *percorre* corresponde ao reconhecimento de *features* e tem as seguintes funções:

1. Encontrar a parte do *trail* correspondente ao início do último desenho. O usuário pode Ter iniciado um desenho. O SEMF vai atuar no atual. O código abaixo mostra a implementação do algoritmo.

(*** PROCURANDO O PEDIDO DE NOVO DESENHO ***)

L := 0;

K := 1;

WHILE K <> 0 DO

BEGIN

L := K;

L := PROCURA(L, '~ ACTIVATE 'MAIN_DLG_CUR' 'ProCMDModelNew.FILE');

K := PROCURA(L+1, '~ ACTIVATE 'MAIN_DLG_CUR' 'ProCMDModelNew.FILE');

END;

(*** FIM DE PROCURANDO O PEDIDO DE NOVO DESENHO ***)

Esse código utiliza o procedimento *procura*, definido no passo 22, para encontrar a última linha igual a: ~ ACTIVATE 'MAIN_DLG_CUR' 'ProCMDModelNew.FILE'. A partir dessa

linha o *trail* descreve os comandos de um novo desenho.

A função procura retorna zero se a linha procurada não for encontrada.

- 1) Enquanto k é diferente de zero
- 2) L recebe o valor de K.
- 3) A variável L armazena o número da linha com o texto procurado.
- 4) A variável K armazena o número da próxima linha com o mesmo texto.
- 5) fim enquanto.

L terá o valor da ultima linha com o texto procurado.

Outra parte desse procedimento procura pela cadeia #PROTRUSION, que marca o início da construção de um sólido no *trail*. Também verifica se o protrusion foi concluído. O código abaixo mostra como é feito:

1 – para k recebendo de L até a ultimalinha do *trail* faça. Comentário: L tem o valor do número da linha onde começa o novo desenho.

2 – se a k-esima linha tem o valor #PROTRUSION e o sólido foi concluído então:

3 – crie uma UI.

4 – coloque o valor do campo POS dessa UI.

5 – coloque o valor do campo ID dessa UI.

6 – coloque o valor tipo igual a 1. Comentário: 1 corresponde a prisma sólido.

7 – coloque na fila de reais da geometria o valor da cota horizontal de referência.

8 – coloque na fila de reais da geometria o valor da cota vertical de referência.

9 – coloque na fila de reais da geometria o valor da cota horizontal.

10 – coloque na fila de reais da geometria o valor da cota vertical.

11 – coloque na fila de reais da geometria o valor da cota altura.

12 – se a *feature* for de alumínio então.

13 – coloque o valor 1 na fila de inteiros da parte fundido.

14 – se não.

15 – coloque o valor 2 na fila de inteiros da parte fundido.

16 – se não existirem *features* na APD então.

17 – inicie a APD com essa *feature*.

18 – se não.

19 – inclua essa *feature*, na APD, abaixo da *feature* principal.


```

1)  FOR K := L TO MAX DO
    BEGIN
2)    IF (LINHA(K)='#PROTRUSION') AND PROT1(K).OK THEN
        BEGIN
3)          X := RFEATURE.CREATE;
4)          X.IDENTIFICACAO.POS := W;
5)          X.IDENTIFICACAO.ID := NF+INTTOSTR(W);
6)          X.TIPO := 1;
7)          X.DADOS.GEOMETRIA.INCLUIREAL(PROT1(K).COTHR);
8)          X.DADOS.GEOMETRIA.INCLUIREAL(PROT1(K).COTVR);
9)          X.DADOS.GEOMETRIA.INCLUIREAL(PROT1(K).COTHP);
10)         X.DADOS.GEOMETRIA.INCLUIREAL(PROT1(K).COTVP);
11)         X.DADOS.GEOMETRIA.INCLUIREAL(PROT1(K).COTEX);
12)         IF FORM1.ALUMINIO1.CHECKED THEN
13)           X.DADOS.FUNDIDO.INCLUINT(1)
14)         ELSE
15)           X.DADOS.FUNDIDO.INCLUINT(2)
16)         IF W = 1 THEN
            BEGIN
17)               INICIA(X);
            END
18)         ELSE
19)           IABAIXO(NF+INTTOSTR(W-1),X);
        END ELSE

```

Esse procedimento é responsável por incluir todas as *features* na APD. E faz isso de forma semelhante ao prisma sólido. Nesse procedimento X é uma UI, ou seja, é uma variável da classe *rfeature* implementada no passo 13. O procedimento *prot1* está definido no passo 30. Inclui real e incluiint são definidos no passo 11. Inicia e Iabaixo são implementados no passo 17.

O procedimento *checar* executa o procedimento *totalcheck*, definido no passo 41. O código abaixo mostra o procedimento.

```

PROCEDURE TLRMF.CHECAR;
BEGIN
    TOTALCHECK(W);
    SETTUDO(FALSE);
END;

```

O procedimento *checatudo* executa todos os procedimentos que aplicam as regras em todas as *features*. Alguns procedimento de regra precisam de um número que indica se a peça é de alumínio ou de ferro fundido. Esse procedimento é responsável por executar os procedimentos de regra com o argumento correto. O código abaixo mostra seu funcionamento.

```

PROCEDURE TLRMF.CHECATUDO;
BEGIN
    SETTUDO(TRUE);
    FORM1.MEMO1.CLEAR;
    (*PRISMA SÓLIDO*)

```

```

IF FORM1.ALUMINIO1.CHECKED THEN
TCHECKPAREDEMIN(1) ELSE//PAREDE MÍNIMA ALUMÍNIO
IF FORM1.FERROFUNDIDO1.CHECKED THEN
TCHECKPAREDEMIN(2);//PAREDE MÍNIMA FERRO FUNDIDO
(*CILINDRO SÓLIDO*)
IF FORM1.ALUMINIO1.CHECKED THEN
TCHECHPAREDEMINE(1) ELSE//PAREDE MÍNIMA ALUMÍNIO
IF FORM1.FERROFUNDIDO1.CHECKED THEN
TCHECHPAREDEMINE(2);//PAREDE MÍNIMA FERRO FUNDIDO
(*FURO CILINDRICO*)
IF FORM1.ALUMINIO1.CHECKED THEN
TCHECKHOLEPAREDE(1) ELSE//PAREDE MÍNIMA ALUMÍNIO
IF FORM1.FERROFUNDIDO1.CHECKED THEN
TCHECKHOLEPAREDE(2);//PAREDE MÍNIMA FERRO FUNDIDO
TCHECKHOLEDIA;//DIÂMETRO MÍNIMO
TCHECKHOLEDIA2;//DIÂMETRO MENOR QUE 19MM
(*CORTE*)
TCHECKCUTCOTA;//DIMENSÃO MÍNIMA
IF FORM1.ALUMINIO1.CHECKED THEN
TCHECKCUTPAREDE(1) ELSE//PAREDE MÍNIMA ALUMÍNIO
IF FORM1.FERROFUNDIDO1.CHECKED THEN
TCHECKCUTPAREDE(2);//PAREDE MÍNIMA FERRO FUNDIDO
SETTUDO(FALSE);

```

END;

Passo 44 – implementar a classe `tlrmf1` herdeira de `tlrmf`. Fazer uma variável chamada `LR` do tipo `tlrmf`. `LR` será um objeto da classe `tlrmf1`.

```

UNIT ULRMF1;
INTERFACE
USES ULRMF;
TYPE
TLRMF1 = CLASS(TLRMF)
PUBLIC
END;
VAR
    LR : TLRMF1;
IMPLEMENTATION
{ TLRMF1 }
END.

```

Interface com usuário.

O usuário vai interagir com o objeto “`lr`” através da interface do SEMF, mostrado na figura 4. Os elementos, utilizados pelo usuário, para acionar comandos do objeto “`lr`” estão destacados na figura 4.

Figura 4 – área de trabalho do SEMF

Procedimento usado para implementação da interface com o usuário (objeto lr).

Passo 45 – um clique no botão “carregar”, circulado de vermelho na figura 4, do SEMF executa os seguintes comandos:

- 1 - SE A OPÇÃO ULTIMO TRAIL ESTIVER SELECIONADA
- 2 – DESTRUA O OBJETO LR
- 3 – CRIE O OBJETO LT
- 4 – CARREGUE O TRAIL(ULTIMO TRAIL)
- 5 – RECONHEÇA AS FEATURES
- 6 – EXIBA AS FEATURES
- 7 - FIM SE SENÃO
- 8 - SE A OPÇÃO ESCOLHER ESTIVER SELECIONADA
- 9 – DESTRUA O OBJETO LR
- 10 – CRIE O OBJETO LR
- 11 – LEIA O TRAIL ESCOLHIDO
- 12 – RECONHEÇA AS FEATURES
- 13 – EXIBA AS FEATURES

No código real fica:

```

1) IF U.CHECKED THEN
  BEGIN
    2) LR.FREE;
    3) LR := TLRMFI.CREATE;
    4) LR.TEXPVIRTUAL('C:\PROE\' + LR.PECLAST2('PATH', FORM3.MEMO1) + '\TRAIL.TXT.*');
    5) LR.PERCORRE;
    6) LR.CHECAR;
  7) END ELSE
  
```



```

BEGIN
  9)IF OD.EXECUTE THEN
    BEGIN
      10) LR.FREE;
      11)LR := TLRMF1.CREATE;
      12)LR.TEXPVIRTUAL(OD.FILENAME);
      13)LR.PERCORRE;
      12)LR.CHECAR;
    END;
  END;
END;

```

Passo 46 – um clique em um dos textos do campo destacado pela seta vermelha, na figura 7, aciona os seguintes comandos:

```
LR.TOTALMOSTRA(L1.ITEMS.STRINGS[L1.ITEMINDEX]);
```

Esse comando está definido no passo 41 e mostra as regras disponíveis para a *feature* selecionada.

Passo 47 – um duplo clique no texto do campo, destacado com a seta azul da figura 4, executa o comando LR.ESCOLHEREGRA(L2.ITEMS.STRINGS[L2.ITEMINDEX], S); esse comando é definido no passo 42 e executa a regra escolhida pelo usuário para a *feature* também escolhida pelo usuário.

Passo 48 – um clique no botão “cancela”, circulado de verde na figura 4, aciona os seguintes comandos:

- 1 – destrua o objeto lr
- 2 - crie o objeto lr
- 3 – exiba as *features* da APD
- 4 – limpe as opções de checagem
- 5 – limpe o editor de saída de regras em pascal:

- 1) LR.FREE;
- 2) LR := TLRMF1.CREATE;
- 3) LR.CHECAR;
- 4) L2.ITEMS.CLEAR;
- 5) MEMO1.LINES.CLEAR;

Passo 49 – um clique no botão “todos”, circulado de azul na figura 4, executa o procedimento LR.CHECATUDO. Esse procedimento é definido no passo 43 e faz com que o SEMF cheque todas as *features* da APD.

APÊNDICE 2 – TABELA COM EXEMPLO DE UM *TRAIL*

Na tabela abaixo mostra um *trail* completo ao desenhar a figura 1.

```
trail file version No. 1052
!Pro/ENGINEER TM 2000i (c) 1988-99 by Parametric Technology Corporation All Rights Reserved.
~ Activate 'main_dlg_cur' 'ProCmdModelNew.file'
~ Activate 'new' 'OK'
@ ok
~ Move 'newtree' 'newtree' \
2 140.233723 140.233723
~ Minimize 'newtree' 'newtree'
~ FocusOut 'newtree' 'AssyTree'
#FEATURE
#CREATE
#DATUM
#PLANE
#DEFAULT
!%CIDATUM PLANE has been created successfully.
#CREATE
#PROTRUSION
#DONE
#DONE
!%CPSelect or create a SKETCHING PLANE.
** 0.207589 0.690476 0 1 1
0.207589 0.582589 L 0 0.448000000
@ sel view 0
0.869977 -0.288123 -0.228641 0.000000 0.587134 -0.739882
0.367820 0.681476 0.540786 500.000000 421.875000 -1664.322880 1.000000 0.000000
** 0.232143 0.685185 0 1 1
0.232143 0.578125 L 0 0.448000000
@ sel view 0
0.869977 -0.288123 -0.228641 0.000000 0.587134 -0.739882
0.367820 0.681476 0.540786 500.000000 421.875000 -1664.322880 1.000000 0.000000
!%CPArrow shows direction of feature creation. Pick FLIP or OKAY.
#OKAY
!%CPSelect or create a horizontal or vertical reference for sketching.
#TOP
** 0.785714 0.730159 0 1 1
0.785714 0.616071 L 0 0.448000000
@ sel view 0
0.869977 -0.288123 -0.228641 0.000000 0.587134 -0.739882
0.367820 0.681476 0.540786 500.000000 421.875000 -1664.322880 1.000000 0.000000
!%CPSelect a perpendicular surface, an edge, or vertex relative to which the section will be dimensioned and
constrained.
** 0.591518 0.500000 0 1 1
0.591518 0.421875 L 0 0.448000000
@ sel view 0
1.327015 0.000000 0.000000 0.000000 1.327015 0.000000
0.000000 0.000000 1.327015 500.000000 421.875000 -2338.268590 1.000000 0.000000
!%CPSelect a perpendicular surface, an edge, or vertex relative to which the section will be dimensioned and
constrained.
** 0.497768 0.629630 0 1 1
0.497768 0.531250 L 0 0.448000000
@ sel view 0
```

```

1.327015 0.000000 0.000000 0.000000 1.327015 0.000000
0.000000 0.000000 1.327015 500.000000 421.875000 -2338.268590 1.000000 0.000000
!%CPSelect a perpendicular surface, an edge, or vertex relative to which the section will be dimensioned and
constrained.
#DONE SEL
!%CPSelect start point.
!%CIRight click disables constraint, <SHIFT>-Right locks constraint, <TAB> changes which constraint is
active.
#RECTANGLE
!%CPPick two points to indicate diagonal of box. Use middle button to abort.
!%CIRight click disables constraint, <SHIFT>-Right locks constraint, <TAB> changes which constraint is
active.
1.149554 0.627232 N 0 0.448000000
** 0.225446 0.828042 0 1 1
0.225446 0.698661 L 0 0.448000000
@ sel view 0
1.327015 0.000000 0.000000 0.000000 1.327015 0.000000
0.000000 0.000000 1.327015 500.000000 421.875000 -2338.268590 1.000000 0.000000
!%CIRight click disables constraint, <SHIFT>-Right locks constraint, <TAB> changes which constraint is
active.
0.225446 0.698661 N 0 0.448000000
** 0.339286 0.738095 0 1 1
0.339286 0.622768 L 0 0.448000000
@ sel view 0
1.327015 0.000000 0.000000 0.000000 1.327015 0.000000
0.000000 0.000000 1.327015 500.000000 421.875000 -2338.268590 1.000000 0.000000
!%CPPick two points to indicate diagonal of box. Use middle button to abort.
!%CIRight click disables constraint, <SHIFT>-Right locks constraint, <TAB> changes which constraint is
active.
0.339286 0.622768 N 0 0.448000000
#MODIFY
** 0.410714 0.804233 0 1 1
0.410714 0.678571 L 0 0.448000000
@ sel view 0
1.327015 0.000000 0.000000 0.000000 1.327015 0.000000
0.000000 0.000000 1.327015 500.000000 421.875000 -2338.268590 1.000000 0.000000
@ sel2d 0 SKETCH_DIM
2 4 4 2 2 1 0 5 1
!%CPEnter a new value [121.109584295]
!%TR
12
** 0.350446 0.619048 0 1 1
0.350446 0.522321 L 0 0.448000000
@ sel view 0
1.327015 0.000000 0.000000 0.000000 1.327015 0.000000
0.000000 0.000000 1.327015 500.000000 421.875000 -2338.268590 1.000000 0.000000
@ sel2d 0 SKETCH_DIM
3 4 5 2 2 0 0 6 1
!%CPEnter a new value [151.386980368]
!%TR
15
!%CPSelect an entity (dimension, spline, text) to modify.
** 0.270089 0.849206 0 1 1
0.270089 0.716518 L 0 0.448000000
@ sel view 0
1.327015 0.000000 0.000000 0.000000 1.327015 0.000000

```



```

0.000000 0.000000 1.327015 500.000000 421.875000 -2338.268590 1.000000 0.000000
** 0.252232 0.846561 0 1 1
0.252232 0.714286 L 0 0.448000000
@ sel view 0
1.327015 0.000000 0.000000 0.000000 1.327015 0.000000
0.000000 0.000000 1.327015 500.000000 421.875000 -2338.268590 1.000000 0.000000
** 0.258929 0.838624 0 1 1
0.258929 0.707589 L 0 0.448000000
@ sel view 0
1.327015 0.000000 0.000000 0.000000 1.327015 0.000000
0.000000 0.000000 1.327015 500.000000 421.875000 -2338.268590 1.000000 0.000000
@ sel2d 0 SKETCH_DIM
1 4 3 1 1 4 4
!%CPEnter a new value [85.7859555421]
!%TR80
!%CPSelect an entity (dimension, spline, text) to modify.
** 0.328125 0.796296 0 1 1
0.328125 0.671875 L 0 0.448000000
@ sel view 0
1.327015 0.000000 0.000000 0.000000 1.327015 0.000000
0.000000 0.000000 1.327015 500.000000 421.875000 -2338.268590 1.000000 0.000000
@ sel2d 0 SKETCH_DIM
0 4 2 1 1 5 4
!%CPEnter a new value [57.1906370281]
!%TR
50
!%CPSelect an entity (dimension, spline, text) to modify.
#REGENERATE
!%CIShowing section modification driven by new dimension values.
!%CIRefitting the section.
!%CISection regenerated successfully.
#DONE
!%CPSpecify feature depth in direction indicated by the arrow.
#DONE
!%CPEnter depth [32.52]
!%TR
30
!%CPAll elements have been defined. Select element(s) or action(s) from dialog box.
~ Activate 'Odui_Dlg_01' `okay'
!%CIProcessing model PRT0001 ...
!%CI100% has been completed.
!%CPAll elements have been defined. Select element(s) or action(s) from dialog box.
!%CIProcessing model PRT0001 ...
!%CI100% has been completed.
!%CIPROTRUSION has been created successfully.
#DONE
~ Activate 'main_dlg_cur' `ProCmdEnvDtmDisp.ddisp' \0
!%CIDatum planes will not be displayed.
#FEATURE
#CREATE
#HOLE
#DONE
#DONE
!%CPSelect the placement plane.
** 0.470982 0.701058 0 1 1
0.470982 0.591518 L 0 0.448000000

```

```

@ sel view 0
7.908588 0.460056 -3.471287 1.142740 7.765630 3.632682
3.309929 -3.780289 7.039960 902.755960 165.928718 -1861.200664 1.000000 0.000000
!%CPSelect TWO edges, axes, planar surfaces or datums for DIMENSIONING.
** 0.533482 0.309524 0 1 1
0.533482 0.261161 L 0 0.448000000
@ sel view 0
7.908588 0.460056 -3.471287 1.142740 7.765630 3.632682
3.309929 -3.780289 7.039960 902.755960 165.928718 -1861.200664 1.000000 0.000000
!%CIDistance from the hilited reference: [32.27]
!%TR
30
!%CPSelect the second reference.
** 0.774554 0.478836 0 1 1
0.774554 0.404018 L 0 0.448000000
@ sel view 0
7.908588 0.460056 -3.471287 1.142740 7.765630 3.632682
3.309929 -3.780289 7.039960 902.755960 165.928718 -1861.200664 1.000000 0.000000
!%CIDistance from the hilited reference: [32.18]
!%TR
30
#DONE
!%CPSpecify feature depth in direction indicated by the arrow.
#DONE
!%CPEnter depth [24.75]
!%TR
15
!%CPEnter DIAMETER [0.9500]
!%TR
40
!%CPAll elements have been defined. Select element(s) or action(s) from dialog box.
~ Activate `Odui_Dlg_01` `okay`
!%CPAll elements have been defined. Select element(s) or action(s) from dialog box.
!%CIProcessing model PRT0001 ...
!%CI100% has been completed.
!%CIHOLE has been created successfully.
#DONE
#FEATURE
#CREATE
#HOLE
#DONE
#DONE
!%CPSelect the placement plane.
** 0.631696 0.457672 0 1 1
0.631696 0.386161 L 0 0.448000000
@ sel view 0
-5.359385 6.579726 -1.670739 1.516985 3.235609 7.876342
6.616866 4.587506 -3.158961 61.667286 267.896075 -1829.543238 1.000000 0.000000
!%CIDistance from the hilited reference: [10.15]
!%TR
30
!%CPSelect the second reference.
** 0.350446 0.510582 0 1 1
0.350446 0.430804 L 0 0.448000000
@ sel view 0
-5.359385 6.579726 -1.670739 1.516985 3.235609 7.876342

```

```

6.616866 4.587506 -3.158961 61.667286 267.896075 -1829.543238 1.000000 0.000000
!%CIDistance from the hilited reference: [27.02]
!%TR
30
#DONE
!%CPSpecify feature depth in direction indicated by the arrow.
#DONE
!%CPEnter depth [24.75]
!%TR
15
!%CPEnter DIAMETER [0.9500]
!%TR
20
!%CPAll elements have been defined. Select element(s) or action(s) from dialog box.
~ Activate `Odui_Dlg_01` `okay`
!%CPAll elements have been defined. Select element(s) or action(s) from dialog box.
!%CIProcessing model PRT0001 ...
!%CI5% has been completed.
!%CI100% has been completed.
!%CIHOLE has been created successfully.
#DONE
~ Select `main_dlg_cur` `MenuBar1` \
1 `View`
~ Close `main_dlg_cur` `MenuBar1`
~ Select `main_dlg_cur` `View.cb_view_mdlsetup`
~ Close `main_dlg_cur` `View.cb_view_mdlsetup`
~ Activate `main_dlg_cur` `psh_view_appearances`
!%CIAdd to, modify, or save the palette or set the model's appearance.
~ Activate `main_dlg_cur` `ProCmdEnvAxisDisp.ddisp` \
0
!%CIAxes will not be displayed.
~ Activate `main_dlg_cur` `ProCmdEnvPntsDisp.ddisp` \
0
!%CIDatum points will not be displayed.
~ Activate `main_dlg_cur` `ProCmdEnvCsysDisp.ddisp` \
0
!%CICoordinate Systems will not be displayed.
~ Select `main_dlg_cur` `MenuBar1` \
1 `View`
~ Close `main_dlg_cur` `MenuBar1`
~ Select `main_dlg_cur` `View.cb_view_mdlsetup`
~ Close `main_dlg_cur` `View.cb_view_mdlsetup`
~ Activate `main_dlg_cur` `psh_view_appearances`
!%CIAdd to, modify, or save the palette or set the model's appearance.
~ Activate `appearance_mgr` `color2.1` \
1
~ Activate `appearance_mgr` `BSet`
~ Activate `MatPlace` `StdOk`
~ Activate `appearance_mgr` `color3.1` \
1
~ Open `appearance_mgr` `OSelectType`
~ Close `appearance_mgr` `OSelectType`
~ Select `appearance_mgr` `OSelectType` \
1 `Surfaces`
!%CISelect surfaces to alter colors.
** 0.627232 0.574074 0 1 1

```



```

0.627232 0.484375 L 0 0.448000000
@ sel view 0
-6.233122 5.507953 2.370194 -3.836747 -6.290827 4.529022
4.608116 2.212494 6.976913 -4.246225 382.571601 -1996.681516 1.000000 0.000000
** 0.700893 0.481481 0 1 1
0.700893 0.406250 L 0 0.448000000
@ sel view 0
-6.233122 5.507953 2.370194 -3.836747 -6.290827 4.529022
4.608116 2.212494 6.976913 -4.246225 382.571601 -1996.681516 1.000000 0.000000
#DONE SEL
~ Activate `appearance_mgr` `BSet`
~ Activate `MatPlace` `StdOk`
~ Activate `appearance_mgr` `BSelect`
!%CISelect surfaces to alter colors.
** 0.649554 0.452381 0 1 1
0.649554 0.381696 L 0 0.448000000
@ sel view 0
-6.233122 5.507953 2.370194 -3.836747 -6.290827 4.529022
4.608116 2.212494 6.976913 -4.246225 382.571601 -1996.681516 1.000000 0.000000
#DONE SEL
~ Activate `appearance_mgr` `BSet`
~ Activate `MatPlace` `StdOk`
~ Activate `appearance_mgr` `color1.1` \
1
~ Activate `appearance_mgr` `BSelect`
!%CISelect surfaces to alter colors.
** 0.464286 0.521164 0 1 1
0.464286 0.439732 L 0 0.448000000
@ sel view 0
-6.233122 5.507953 2.370194 -3.836747 -6.290827 4.529022
4.608116 2.212494 6.976913 -4.246225 382.571601 -1996.681516 1.000000 0.000000
#DONE SEL
~ Activate `appearance_mgr` `BSet`
~ Activate `main_dlg_cur` `ProCmdViewRepaint.view`
~ Activate `appearance_mgr` `MClose`
*SETUP_SPIN 1.000000 0 20000
~ Activate `main_dlg_cur` `ProCmdViewZoomIn.view`
!%CPIndicate two locations to define a box for the zoom area.
** 0.171875 0.949735 0 1 1
0.066964 0.772321 L 0 0.448000000
0.066964 0.772321 N 0 0.448000000
** 1.120536 0.052910 0 1 1
1.015625 0.015625 L 0 0.448000000
!%CPIndicate two locations to define a box for the zoom area.
~ Activate `main_dlg_cur` `ProCmdEnvAxisDisp.ddisp` \
1
!%CIAxes will be displayed.
~ Move `main_dlg_cur` `main_dlg_cur` \
2 140.233723 140.233723
~ Minimize `main_dlg_cur` `main_dlg_cur`

```

Esse arquivo foi criado ao se desenhar um paralelepípedo sólido com dois furos cilíndricos como mostra figura 1 abaixo.

Figura 1 - Paralelepípedo sólido com dois furos cilíndricos.