

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

UM SISTEMA PARA O PROJETO E FABRICAÇÃO REMOTA DE PEÇAS
PRISMÁTICAS VIA INTERNET

dissertação submetida à

UNIVERSIDADE FEDERAL DE SANTA CATARINA

para a obtenção do grau de

MESTRE EM ENGENHARIA MECÂNICA

JULIO CÉSAR TICONA BENAVENTE

Florianópolis, fevereiro de 2007

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

UM SISTEMA PARA O PROJETO E FABRICAÇÃO REMOTA DE PEÇAS
PRISMÁTICAS VIA INTERNET

JULIO CÉSAR TICONA BENAVENTE

Esta dissertação foi julgada adequada para a obtenção do título de

MESTRE EM ENGENHARIA

ESPECIALIDADE ENGENHARIA MECÂNICA
sendo aprovada em sua forma final.

João Carlos Espíndola Ferreira, Ph. D. - Orientador

Fernando Cabral, Ph. D. – Coordenador do curso

Banca Examinadora

Fernando A. Forcellini, Dr. Eng. - Presidente

Carlos Henrique Ahrens, Dr. Eng.

Marcelo Texeira dos Santos, Dr. Eng.

DEDICATÓRIA

*Esta dissertação de mestrado está dedicada
aos meus pais,*

Gumercinda e Augustín,

AGRADECIMENTOS

Ao meu criador e mantenedor, ao Deus do céu, a quem devo tudo quanto eu sou.

Ao departamento de Engenharia Mecânica pela oportunidade e por me dar as condições para a elaboração do presente trabalho de mestrado.

Aos meus pais Gumercinda e Agustín que me apoiaram em todo momento.

Aos meus irmãos César e Sonia que sempre me tiveram presente nos seus corações.

Aos meus colegas do GRIMA, especialmente à Izabel que me ajudou nas correções de vários capítulos da dissertação, além de ser essa pessoa maravilhosa que é, assim como ao Mikos pela sua importante ajuda, também ao Roman, pelo apoio no meu trabalho ao longo do mesmo, e pela sua fundamental ajuda no estágio final do mesmo.

Ao professor João Carlos, pela orientação deste trabalho de mestrado, e pela ajuda ao longo do desenvolvimento do mesmo.

SUMÁRIO

CAPÍTULO 1	1
INTRODUÇÃO.....	1
1.1 Objetivos e contribuições	2
1.1.1 Objetivo geral:.....	2
1.1.2 Objetivos específicos:.....	2
1.2 Limites e escopo do trabalho.....	3
1.3 Contribuições do Trabalho	3
1.4 Estrutura do documento.....	4
CAPÍTULO 2	5
CONCEITOS FUNDAMENTAIS	5
2.1 Manufatura Eletrônica (e-Mfg).....	5
2.2 Manufatura Remota	5
2.3 Teleoperação.....	6
2.3.1 Teleoperação e o CAM.....	9
2.4 Laboratórios virtuais e remotos	9
2.5 Modelo Cliente Servidor	10
2.6 World Wide Web - WWW	10
2.7 Conceitos relativos ao CAD	11
2.7.1 Projeto por <i>Features</i>	12
2.8 CAPP (Computer Aided Process Planning)	15
2.8.1 Abordagens de Sistemas CAPP.....	15
2.8.2 Abordagem variante	16
2.8.3 Abordagem generativa.....	17
2.8.4 Híbrido.....	18
2.8.5 Ferramentas Computacionais para CAPP	18

2.9	CAM.....	18
2.10	Prototipagem Rápida	19
2.11	Conclusões do capítulo	20
CAPÍTULO 3		21
REVISÃO BIBLIOGRÁFICA DOS SISTEMAS CORRELATOS		21
3.1	Sistema Webmachining	21
3.1.1	Interface Gráfica com o Usuário (GUI).....	23
3.1.2	Mapeamento de <i>Features</i>	23
3.1.3	Operações de usinagem	24
3.2	Sistema Cybercut.....	24
3.3	Sistema WebSPIFF	26
3.4	Outros Sistemas de Manufatura Remota	29
3.4.1	O “Toronto Virtual Enterprise” (TOVE).....	29
3.4.2	O projeto ESPIRIT STEP-NC	29
3.5	Conclusões do capítulo	33
CAPÍTULO 4		34
DESCRIÇÃO DO SISTEMA INTEGRADO CAD/CAPP/CAM		34
4.1	A Linguagem de programação	37
4.2	Módulo CAD	37
4.2.1	Decomposição por faces.....	39
4.2.2	Criação do modelo da peça.....	42
4.2.2.1	A biblioteca de <i>features</i>	44
4.2.3	Validação do modelo da peça.....	51
4.2.4	Modelador gráfico	59
4.3	Módulo CAPP	60
4.3.1	Mapeamento das <i>features</i> de projeto para <i>workingsteps</i>	63

4.3.2	Determinação das estratégias de usinagem para cada tipo de <i>feature</i>	68
4.3.2.1	Estratégia de usinagem para o furo.....	68
4.3.2.2	Estratégia de usinagem para o degrau e a ranhura.....	71
4.3.2.3	Estratégia de usinagem para a cavidade retangular.....	72
4.3.3	Formação de grupos de <i>workingsteps</i>	75
4.3.4	Determinação das seqüências de usinagem.....	77
4.4	Módulo CAM.....	78
4.4.1	Geração do código HPGL.....	80
4.4.2	Envio do código HPGL para a máquina por porta serial.....	83
4.5	Envio da ordem de fabricação.....	84
CAPÍTULO 5.....		86
RESULTADOS OBTIDOS, ESTUDOS DE CASOS.....		86
5.1	Definição das peças a serem estudadas.....	86
5.2	Peça 1.....	87
5.2.1	Modelagem da peça 1 no módulo CAD por <i>features</i>	87
5.2.2	Módulo CAPP.....	88
5.2.3	Módulo CAM.....	90
5.3	Peça 2.....	92
5.3.1	Modelagem da peça 2 no aplicativo CAD por <i>features</i>	92
5.3.2	Módulo CAPP.....	93
5.3.3	Módulo CAM.....	95
5.4	Peça 3.....	98
5.4.1	Modelagem da peça 3 no aplicativo CAD por <i>features</i>	98
5.4.2	Módulo CAPP.....	99
CAPÍTULO 6.....		102
6.1	Contribuições do presente trabalho.....	102

6.2	Disponibilização e implementação do sistema computacional na Internet..	103
6.2.1	CAD por <i>features</i>	103
6.2.2	Módulo CAPP	104
6.2.3	Módulo CAM	104
6.2.4	Banco de dados.....	105
6.3	Sugestões para futuros trabalhos	105
	REFERÊNCIAS BIBLIOGRÁFICAS	107

LISTA DE FIGURAS

2.1. Modelo de manufatura cliente-servidor (adaptado de ÁLVARES, 2005).	10
2.2. Exemplos de <i>features</i> .	12
2.3. Abordagem de criação de <i>features</i> utilizando projeto por <i>features</i> (SHAH e MANTYLA, 1995).	14.
2.4. Tipos de CAPP, (KERRY, 1997).	16
3.1. Diagrama das camadas de multiagentes do sistema <i>WebMachining</i> (ÁLVARES, 2005)	22
3.2. Interface gráfica do Módulo CAD do sistema <i>WebMachining</i> .	24
3.3 Interface Gráfica WebCAD3D (Cybercut).	25
3.4 Algumas regras de manufatura do sistema Cybercut.	26
3.5. Arquitetura do sistema WebSpiff (BIDARRA, 2001).	27
3.6 Interface Gráfica do sistema WebSpiff (VAN DER BERG et al. 2001).	28
3.7. Arquitetura de um sistema de manufatura integrado através da internet (ZHANG, et. al, 2000).	33
4.1 Diagrama IDEF0: Sistema integrado CAD/CAPP/CAM, nível A0.	35
4.2 Diagrama IDEF0 Sistema CAD/CAPP/CAM integrado, nível 1.	36
4.3 Diagrama IDEF0 da atividade CAD.	38
4.4 Decomposição por faces.	39
4.5 Bloco com degrau em perspectiva, e vistas das faces envolvidas	40
4.6 Diagramas UML das classes (a) Bloco, (b) Face, (c) <i>Feature</i> .	41
4.7 Instanciação de uma nova peça bruta.	42

4.8 Bloco fixado na máquina com a face XY como face de trabalho.	43
4.9 Bloco fixado com o vértice 1 ativado (girado 90 graus).	43
4.10 Modificação das referências de projeto.	44
4.11 <i>Features</i> consideradas na implementação da “biblioteca de <i>features</i> ”.	44
4.12 Diagrama UML da biblioteca de <i>features</i>	45
4.13 <i>Feature</i> Furo.	46
4.14 Interface gráfica para a instanciação de um novo furo.	47
4.15 Possibilidades de criação de um novo degrau.	48
4.16 Interface gráfica para a criação de um novo degrau.	49
4.17 Interface gráfica para a criação de uma nova ranhura.	50
4.18 <i>Feature</i> cavidade retangular.	50
4.19 Interface gráfica para criar uma nova cavidade retangular.	51
4.20 <i>Features</i> inválidas por estarem fora dos limites da face.	52
4.21. Mensagem de erro emitido pelo sistema CAD informando a inconsistência da cavidade que se pretende criar.	52
4.22 a) furo válido, b) e c) furos inválidos devido à posição Z.	53
4.23. Mensagem de erro devido à inconsistência da posição Z da <i>feature</i> .	53
4.24. Mensagem de erro informando a inconsistência da <i>feature</i> devido à posição Z da <i>feature</i> aninhada.	54
4.25. Furo aninhado dentro da cavidade.	55
4.26. Interseção de uma cavidade e uma ranhura.	55
4.27. Interseção de uma cavidade e uma ranhura em vista lateral.	56

4.28. Furo inválido devido à sua profundidade ser maior do que a profundidade máxima da peça bruta.	56
4.29. Mensagem de erro devido a inconsistência na profundidade da <i>feature</i> .	57
4.30. Inconsistência de uma <i>feature</i> ao ser comparada com a geometria da ferramenta.	57
4.31. mensagem de erro ao tentar fazer uma ranhura com largura menor do que o da ferramenta.	58
4.32. Inconsistência da <i>feature</i> por causa da máquina.	58
4.33. Mensagem de erro emitida quando tenta-se criar <i>features</i> em uma face que não pode ser fixada dentro da máquina.	59
4.34. Uma peça prismática modelada no aplicativo CAD.	60
4.35. Diagrama IDEF0 das atividades do módulo CAPP.	61
4.36 Máquina de prototipagem rápida por usinagem Roland MDX-20 utilizada neste trabalho	62
4.37. Diagrama UML da classe <i>Workingstep</i> .	64
4.38. Diagrama UML da classe Mapeadora.	65
4.39. Mapeamento de <i>features</i> para <i>workingsteps</i> .	65
4.40. Condições de usinagem $a_{p_máxima}$ e $a_{e_máximo}$.	67
4.41. Estratégia de movimentação do furo.	68
4.42. Determinação da profundidade de corte $a_{p_utilizada}$.	69
4.43. Determinação de $a_{e_utilizado}$ para o furo.	70
4.44. Estratégia de movimentação da ferramenta para a ranhura	71
4.45. Determinação do valor de $a_{e_utilizado}$ para uma ranhura.	72

4.46. Movimentação da ferramenta para a cavidade retangular.	73
4.47. Trajetórias na região de movimentação circular e linear.	75
4.48. Determinação da seqüência de usinagem para duas <i>features</i>	76
4.49 Peça exemplo com <i>features</i> em uma face.	76
4.50 Organização das <i>features</i> por camadas, em função da face onde elas se originam.	77
4.51 Possíveis pontos de início da usinagem do degrau.	77
4.52. Critério da mínima distância para o sequenciamento de <i>features</i> em uma camada.	78
4.53. Diagrama IDEF0 das atividades do módulo CAM.	79
4.54. Número de pontos utilizados na interpolação circular.	81
4.55. Fixação da peça na máquina.	82
4.56. Ilustração do funcionamento dos comandos HPGL utilizados pelo módulo CAM.	83
4.57. Interface gráfica do aplicativo que envia o código HPGL por porta serial.	84
4.58 Arquitetura do sistema rodando na internet.	85
5.1 Peça 1 visualizada no módulo CAD.	88
5.2. <i>Features</i> presentes na peça 1 e sua seqüência no vetor de <i>features</i> .	89
5.3. Código HPGL para usinar a cavidade 3.	91
5.4. Peça 1 usinada na máquina de prototipagem rápida Roland MDX-20.	92
5.5. Peça 2 modelada no aplicativo CAD por <i>features</i> .	93
5.6. <i>Features</i> presentes na peça 2 e sua seqüência no vetor de <i>features</i> .	94

5.7. Seqüência de usinagem para os furos da camada 2.	94
5.8. código HPGL gerado para a camada zero da face XY.	96
5.9. Execução do código HPGL para a peça 2.	97
5.10. Peça 2 fabricada.	98
5.11. Peça 3 modelada no aplicativo CAD.	98
5.12. <i>Features</i> presentes na face XY da peça 3	99
5.13. Peça 3 usinada.	101
6.1. Página <i>web</i> que disponibiliza o sistema computacional.	103
6.2 Transformação de <i>features</i> de projeto em <i>features</i> compostas de usinagem	106

LISTA DE TABELAS

4.1 Dados técnicos da Máquina Roland MDX-20	62
4.2 Potências unitárias para diferentes materiais para fresamento de topo (FERREIRA, 1996)	66
5.1. <i>Features</i> presentes nas peças estudadas.	87
5.2. <i>Features</i> presentes na peça 1.	89
5.3. Etapas executadas visando seqüenciar os Workingsteps para a peça 1	90
5.4. <i>Features</i> presentes na peça 2 (vetor de <i>features</i>).	93
5.5. Etapas executadas visando seqüenciar os Workingsteps para a peça 2.	95
5.6 <i>Features</i> presentes na peça 3 na seqüência em que foram inseridas em cada face.	99
5.7 Agrupador de <i>workingsteps</i> e as seqüências de usinagem para a peça 3.	100

LISTA DE ABREVIATURAS

CAD	Projeto Auxiliado por Computador
CAM	Manufatura Auxiliada por Computador
CAE	Engenharia Auxiliada por Computador
CAPP	Planejamento do Processo Auxiliado por Computador
CNC	Comando Numérico Computadorizado
CIMS	Sistemas de Manufatura Integrados por Computador
CNC	Comando Numérico Computadorizado
CORBA	Common Object Request Brokerage Architecture
FTP	File Transfer Protocol
HTTP	Hyper Text Transfer Protocol
HPGL	Hewlett Packard Graphics Language
IDEF	Integration Definition for Function Modeling
IDL	Linguagem de Definição de Interface
NC	Comando Numérico
NCFU	Unidades de Features de Controle Numérico
NCP	Programação do Comando Numérico
WWW	World Wide Web
STEP	Standard for Exchange Product Model Data
UML	Unified Modeling Language

RESUMO

A globalização tem causado um aumento significativo na competição entre as empresas, mas em contrapartida ela também permite a cooperação entre empresas, mesmo que estejam geograficamente distantes umas das outras. Uma área de pesquisa decorrente dessa aproximação virtual é a fabricação de peças a distância, que conta com a participação dos seguintes atores: (i) um cliente remoto que introduz as encomendas; (ii) uma empresa modeladora responsável pelo desenvolvimento e disponibilização de módulos computacionais para o projeto das peças e para o planejamento das operações de manufatura; e (iii) uma empresa onde a fabricação propriamente dita será efetuada. No presente trabalho de Mestrado descreve-se o desenvolvimento de um sistema computacional visando a integração CAD/CAPP/CAM para a fabricação a distância de peças prismáticas via Internet. O módulo CAD é baseado na abordagem de projeto por *features*, onde um cliente remoto pode modelar a peça para logo depois enviar a ordem de fabricação para o servidor. O módulo CAPP recebe as informações vindas do CAD e adiciona dados relativos à manufatura, gerando então o plano de processo. Finalmente, de posse das informações acima, o módulo CAM gera o programa para a usinagem da peça e o envia para a máquina. Os resultados deste sistema são testados em uma máquina de prototipagem rápida por usinagem. O sistema computacional foi baseado nas técnicas de programação orientada a objetos.

ABSTRACT

Globalization has been causing a significant increase in the competition among companies, but on the other hand it also allows the cooperation between companies, even if they are geographically distant among themselves. A research area due to that virtual approach is the remote manufacture of parts, which has the participation of the following actors: (i) a remote customer who inputs the orders; (ii) a modeling company, who is responsible for the development and provision of software modules for the design of the parts and for the manufacturing process planning; and (iii) a company where actual manufacture will take place. In the present Master's research the development of a software system is described aiming at the integration of CAD/CAPP/CAM for the remote manufacture of prismatic parts through the Internet. The CAD module is based on the features approach, where a remote customer models the part and then sends the production order to the server. The CAPP module receives the information from the CAD module, and adds manufacturing data to it, generating the process plan. Finally, after getting hold of the above information, the CAM module generates the program for the machining of the part and sends it to the machine. The results of this system are tested in a rapid prototyping machine (by machining). The software system was developed based on the object-oriented programming technique.

CAPÍTULO 1

INTRODUÇÃO

Há vários anos as empresas de manufatura em todo o mundo têm atuado em um ambiente competitivo, no qual produtos de boa qualidade com preços relativamente reduzidos têm sido lançados com uma frequência cada vez maior, visando conquistar o mercado consumidor. Ao mesmo tempo, a competição causada pela globalização tem levado as empresas a buscar cooperações entre si, buscando manter-se competitivas neste mercado.

Componentes e produtos variados e de boa qualidade, fabricados em diferentes países, têm sido disponibilizados às pessoas em qualquer parte do mundo, muitas vezes a preços iguais ou inferiores aos produtos locais. A Internet tem sido apontada como uma das grandes precursoras da globalização, pois ela permite uma comunicação rápida entre pessoas e também empresas. Uma prova disto é a quantidade de recursos que têm sido gastos no comércio eletrônico.

Além da venda de itens como bens de consumo através da Internet, esta tecnologia tem sido utilizada com sucesso em outras atividades importantes, como a manutenção remota de equipamentos (ROCKWELL, 2002) e a tele cirurgia efetuada por um robô comandado remotamente por um cirurgião localizado a centenas de quilômetros de distância (SCIENTIFIC AMERICAN, 2000).

Um fator extremamente importante, que contribui significativamente para o aumento da competitividade de produtos, diz respeito às decisões referentes aos estágios iniciais do processo de desenvolvimento de novos produtos. Nesses estágios normalmente são tomadas muitas decisões fundamentais que irão gerar economia na posterior etapa de manufatura.

Os gastos com o desenvolvimento do projeto do produto ainda são relativamente pequenos, e a relação custo/benefício nos estágios iniciais do desenvolvimento do produto é bem mais favorável do que nos estágios posteriores, onde a introdução de mudanças pode implicar em custos bastante elevados (KUSIAK, 2004).

A prototipagem rápida corresponde a um conjunto de tecnologias que surgiram nos últimos anos visando reduzir o tempo gasto durante a fase de desenvolvimento de um novo produto. Estas tecnologias permitem criar objetos sólidos diretamente a partir de dados gerados em sistemas CAD. Elas podem ser: aditivas, subtrativas ou conformativas. Dentre os processos aditivos tem-se a estereolitografia, a sinterização seletiva a laser (SLS – *Selective Laser Sintering*) e a modelagem por deposição por fundição (FDM – *Fused Deposition Modeling*). No caso da prototipagem rápida por subtração, o exemplo típico é a usinagem.

Uma das motivações deste trabalho está associada à importância da tecnologia da Internet nos últimos anos, em especial a sua utilização como meio que permite o projeto e fabricação remota de peças. Além disso, instituições de ensino que não possuem este equipamento poderão utilizar o sistema, disseminando de maneira prática aos seus alunos temas importantes como CAD, CAPP e CAM.

1.1 Objetivos e contribuições

1.1.1 Objetivo geral:

O presente trabalho de Mestrado tem como objetivo a implementação de um protótipo de um sistema computacional para a integração CAD/CAPP/CAM (Projeto Assistido por Computador / Planejamento do Processo Assistido por Computador / Manufatura Assistida por Computador) voltada para manufatura remota de peças prismáticas em uma máquina de prototipagem por usinagem de três eixos (Roland MDX – 20) utilizando a Internet, para disponibilizar on-line os módulos para a modelagem das peças prismáticas por meio de *features*, e a sua respectiva fabricação. O sistema permite que o usuário remoto, aqui chamado de cliente, possa fazer a modelagem da peça *on-line*, e posteriormente criar e enviar uma ordem via Internet para que a mesma venha a ser executada na máquina de prototipagem por usinagem localizada no laboratório GRIMA/GRUCON/EMC/UFSC.

1.1.2 Objetivos específicos:

- Desenvolver um módulo de CAD baseado em *features* para os usuários através da Internet.

- Desenvolver um módulo de CAPP, que será o responsável pela seleção das condições de usinagem, bem como a geração das trajetórias das ferramentas.
- Desenvolver um módulo de CAM, onde o programa para a usinagem da peça será gerado, carregado e testado na máquina de prototipagem rápida por usinagem.
- Desenvolver de um banco de dados para o cadastramento e armazenamento de dados relativos ao projeto e ao usuário remoto que o criou, além das ordens de fabricação das peças.

1.2 Limites e escopo do trabalho

Este trabalho limita sua abrangência ao desenvolvimento dos módulos de projeto, planejamento do processo e de manufatura de peças prismáticas. As *features* consideradas são: furo, degrau, ranhura e cavidade retangular. Não é objetivo deste trabalho a geração de planos de processo alternativos nem a otimização dos mesmos.

Para a geração do sistema desenvolvido neste trabalho, foi empregado o sistema operacional Windows, a linguagem de programação Java e o software MySQL (2003).

1.3 Contribuições do Trabalho

Este trabalho aplica métodos voltados para a Internet visando o desenvolvimento de um sistema integrado CAD/CAPP/CAM. Segundo ÁLVARES (2005), os ambientes CAD/CAPP/CAM disponibilizados para uso via Internet serão globais, centrados na rede e distribuídos, tendo como *front-end* navegadores *Web*.

Sistemas CAD/CAPP/CAM desenvolvidos a partir destas tecnologias poderão ser utilizados tanto na indústria quanto em instituições de ensino em disciplinas relativas a estes temas. Além disso, tal sistema permitirá que um cliente remoto, que não necessariamente conte com os recursos nem equipamentos para o projeto, planejamento do processo e manufatura para a fabricação de uma peça prismática, possa ter o contato direto com o fabricante das peças.

O sistema proposto apresenta contribuições para o desenvolvimento de sistemas de telemanufatura baseados em serviços *Web*, integrando as atividades de projeto,

planejamento de processos e fabricação, através da modelagem por *features*, utilizando o protocolo TCP/IP.

1.4 Estrutura do documento

O trabalho começa com uma breve apresentação do problema a ser resolvido e dos objetivos a serem atingidos. Os conceitos fundamentais relacionados ao presente trabalho, que visam facilitar a compreensão do mesmo são apresentados no capítulo dois. Já o capítulo três apresenta uma revisão bibliográfica relativa aos trabalhos desenvolvidos nesta área, ou seja de manufatura remota ou telemanufatura. O capítulo quatro apresenta a descrição do que é o objeto da presente dissertação de Mestrado, descrevendo detalhadamente os módulos que compõem o sistema desenvolvido, já citados anteriormente nos objetivos específicos. Os resultados obtidos, na forma de exemplos de utilização da interface CAD/CAPP/CAM para a geração do projeto, plano de processo e fabricação de peças prismáticas serão apresentados no capítulo cinco. Finalmente no capítulo seis são apresentadas as conclusões, limitações e as sugestões para futuros trabalhos.

CAPÍTULO 2

CONCEITOS FUNDAMENTAIS

Neste capítulo são apresentados os conceitos mais relevantes ao presente trabalho de Mestrado, e que serão úteis para a melhor compreensão do mesmo. São eles: manufatura eletrônica, manufatura remota, tecnologia *Web*, laboratórios remotos e prototipagem rápida.

2.1 Manufatura Eletrônica (e-Mfg)

A tecnologia da informação, em especial, a tecnologia de redes de comunicação e Internet, está abrindo um novo domínio para construção dos futuros ambientes de manufatura, denominados de ambientes de manufatura eletrônica (*e-Mfg*). Uma das aplicações desta tecnologia consiste nos trabalhos eletrônicos colaborativos (*e-Work*), especialmente no que se refere às atividades desenvolvidas durante o ciclo de desenvolvimento de produto em ambientes CAD/CAPP/CAM integrados e colaborativos (MALEK *et al.*, 1998; LEE, 2003 & NOF, 2004).

O *e-Work* é definido (NOF, 2004) como qualquer atividade produtiva que seja colaborativa, suportada por computador e apoiada por comunicação em organizações altamente distribuídas de robôs e/ou de pessoas ou sistemas autônomos. Em essência, *e-Work* é composto por atividades eletrônicas baseadas e executadas através do uso de tecnologia da informação.

Esta nova tendência de desenvolvimento de sistemas computacionais centrados em rede e espacialmente distribuídos, permite que os desenvolvedores de produtos e projetistas tenham maior facilidade de comunicação, possibilitando o compartilhamento e o projeto colaborativo durante o desenvolvimento do produto, bem como a teleoperação e monitoração dos dispositivos de manufatura (ÁLVARES, 2005).

2.2 Manufatura Remota

A manufatura remota ou telemanufatura, pode ser definida como uma atividade onde uma empresa (cliente) utiliza serviços oferecidos por centros especializados (servidores) disponibilizados via rede de comunicação através das supervias da informação (Internet) para executar, em tempo real, operações e processos necessários para o projeto e a

produção de bens. Desta forma as atividades relativas à manufatura remota estão presentes em todo o ciclo de desenvolvimento do produto, desde sua concepção até a sua fabricação e distribuição (MALEK et al, 1998).

Atualmente existem empresas com alto nível de especialização que oferecem serviços em um ambiente voltado para telemanufatura, por exemplo, o sistema Cybercut desenvolvido na Universidade de Berkeley (<http://cybercut.berkeley.edu>), onde é possível modelar e usinar peças prismáticas (BROWN e WRIGHT, 1998). ÁLVARES et al.(2002) desenvolveram um sistema que permite a teleoperação de uma máquina de oxi-corte CNC (<http://weboxicorte.graco.unb.br>). Estes centros especializados detêm o estado da arte na tecnologia de software (sistemas CAE/CAD/CAPP/CAM/ERP, entre outros), especialidades avançadas e acesso às informações atualizadas em seus campos, a fim de auxiliar seus clientes no desenvolvimento de novos produtos e processos (AHN *et al.*, 1999 e MALEK *et al.*, 1998).

2.3 Teleoperação

Na indústria da manufatura de nossos dias, grande parte das operações têm sido automatizadas, contudo muitos processos são difíceis de serem automatizados e por isso devem confiar na supervisão e tomada de decisão de um especialista humano; em áreas tais como a identificação de peças com defeito e variações do processo (PESHKIN et al., 2001), ou aplicações médicas (TAYLOR e STOIANOVICI, 2003). Quando tais atividades são requeridas, ainda assim os humanos têm que realizar tarefas complicadas, AARNO et al. (2005) dizem que os sistemas colaborativos homem-máquina (HMCS) poderiam ser usados para prevenir dificuldades ergonômicas do operador permitindo a cooperação entre um humano e um sistema robótico de uma maneira flexível.

O citado autor também afirma que devido às recentes mudanças na manufatura tais como a produção *just in time* e novos processos, é necessário ter a capacidade de alterar rapidamente o fluxo de trabalho, fazendo-se necessário o auxílio dos sistemas colaborativos homem-máquina.

Segundo NOF (1999), a teleoperação pode ser definida como o controle contínuo e direto de um teleoperador, que corresponde a um manipulador que requer comandos ou supervisão de um ser humano remoto. Telepresença se refere à intensa utilização de

realimentação sensorial para a teleoperação, fornecendo realismo para o operador em uma abordagem idealizada de “presença a distância”.

A telerobótica amplia o domínio do manipulador em um contexto de controle compartilhado pela máquina e pelo operador humano. Um sistema dito teleoperador/teleoperator (NOF, 1999) consiste de uma unidade remota (manipulador), uma unidade de comando para entrada dos comandos do operador (interface homem/máquina) e um canal de comunicação como elo de ligação da unidade de comando com a unidade remota.

SHERIDAN, 1984 categoriza os sistemas de teleoperação tomando como base o grau de automação do sistema. Em um espectro variando da mínima para a máxima autonomia, a teleoperação pode ser classificada:

1. Controle manual sem auxílio computacional;
2. Controle manual com significativo auxílio computacional;
3. Controle supervisão com predomínio do controle realizado pelo operador humano;
4. Controle supervisão com predomínio do controle realizado pelo computador; e
5. Controle completamente automático, onde os operadores humanos observam o processo sem intervenções.

Diversos modelos relacionam esta interface, sendo os seguintes os tipos principais

1. Modelo Mestre-Escravo;
2. Modelo de Telepresença;
3. Modelo Professor-Aluno; e
4. Modelo Supervisor-Companheiro.

A comunicação homem-máquina pode ser facilitada com o uso de gráficos interativos, controle com vários graus de liberdade e interfaces híbridas. Uma vez estabelecida e efetivada a comunicação entre homem e máquina deve-se observar a

relação entre máquina remota e seu ambiente. Dessa forma são propostos esquemas de classificação de modelos de ambientes:

1. Ambiente Remoto Totalmente Modelado;
2. Ambiente Remoto Parcialmente Modelado; e
3. Ambiente Remoto Desconhecido.

Há diversas situações em que a operação de um determinado dispositivo é viabilizada através de um operador remoto, o qual se encontra fisicamente distante do objeto a ser controlado. Estes dispositivos remotamente operados vêm sendo utilizados em diversas áreas tais como robótica, medicina, pesquisa submarina, etc. (ÁLVARES, 2005).

Geralmente uma tarefa de manipulação remota requer uma máquina que possa realizar uma variedade de manipulações físicas. Essas tarefas podem ser altamente variáveis. As tarefas remotas variam de acordo com o local onde a tarefa tem de ser realizada, e se essa tarefa é conhecida a priori. Um sistema ideal para realizar estas tarefas tem de ter a mesma capacidade de um operador humano para reconhecer qual tarefa tem de ser realizada e quais os passos que devem ser tomados para a sua realização, além de possuir flexibilidade para realizar as manipulações físicas, de maneira análoga à operação de um ser humano.

Aspectos importantes no controle efetivo de dispositivos remotos envolvem:

- o atraso entre uma ação de controle do operador e sua correspondente visualização, como informação de realimentação para a sua ação, mostrada no seu *display*;
- largura de banda do sistema de comunicação, que determinam uma parcela importante do atraso relacionado à ação de controle;
- experiência na tarefa de controle contínuo, manual, do dispositivo controlado, questões de segurança e erros de posicionamento e autonomia (SHERINDAM, 1992).

A questão de autonomia é particularmente importante, já que define o quanto o dispositivo teleoperado pode funcionar independentemente do operador humano. Isto

pode ser obtido através da implementação de parte da estratégia de controle em um computador localizado no *site* remoto. O operador pode estabelecer altos níveis de controle ao computador remoto, o qual pode executar os níveis mais baixos de controle (ÁLVARES, 2005).

2.3.1 Teleoperação e o CAM

O CAM (Manufatura Auxiliada por Computador) é o estágio final do auxílio do computador na fabricação de peças mecânicas. Dessa forma, em referências mais antigas (antes da década de 90), CAM é definido como o uso de sistemas de computação para planejar, gerenciar, e controlar as operações de uma planta, através de interface computadorizada direta ou indireta com a produção (GROOVER, 1987). Segundo esta definição, o CAM pode ser dividido em duas categorias:

1. Execução (monitoramento e controle por computador): estas são as aplicações diretas na qual o computador se conecta diretamente com o processo de manufatura com o propósito de monitorar ou controlar o processo. A teleoperação está intimamente ligada a esta etapa do CAM.
2. Planejamento (aplicações de suporte à manufatura): Estas são as aplicações indiretas na qual o computador é usado como suporte das operações de produção na planta, mas não há interface direta entre o computador e o processo de manufatura. O ambiente integrado CAD/CAPP/CAM se insere nesta categoria.

2.4 Laboratórios virtuais e remotos

ÁLVARES (2005) define os laboratórios virtuais como sendo simulações de dispositivos físicos por meio de software, podendo ser desde uma bancada de testes para um sistema visando elaborar métodos para o seu controle, até mesmo a simulação de um sistema de manufatura baseado em teoria de filas.

Os laboratórios acessíveis através da Internet estão se transformando em uma maneira popular para reduzir os custos de equipamento e disponibilizar conceitos através do laboratório em cursos de Educação a Distância (ALVES, 2001).

Por sua vez, os laboratórios remotos permitem que experimentos reais do laboratório sejam controlados remotamente através de uma conexão de Internet por exemplo. Este

tipo de laboratório adequa-se, por exemplo, a cursos de Educação a Distância, pois os estudantes não necessitam estar fisicamente presentes no laboratório. Os parâmetros de controle de um laboratório remoto podem ser ajustados em uma página Web e enviados para um servidor que controla o experimento. Os dados reais são gravados durante o experimento e retornados ao usuário através da conexão Web (ÁLVARES, 2005).

2.5 Modelo Cliente Servidor

Neste tipo de modelo os dados e processamentos são distribuídos entre um conjunto de processadores. Os principais componentes desse modelo segundo HALL (1999) e TOURINO (2000) são:

1. Os Servidores, os quais oferecem serviços para outros sub-sistemas;
2. Os Clientes, os quais requisitam serviços oferecidos pelos servidores; e
3. Rede de computadores, que permite que os clientes acessem a esses serviços.

A figura 2.1 mostra um modelo deste sistema.

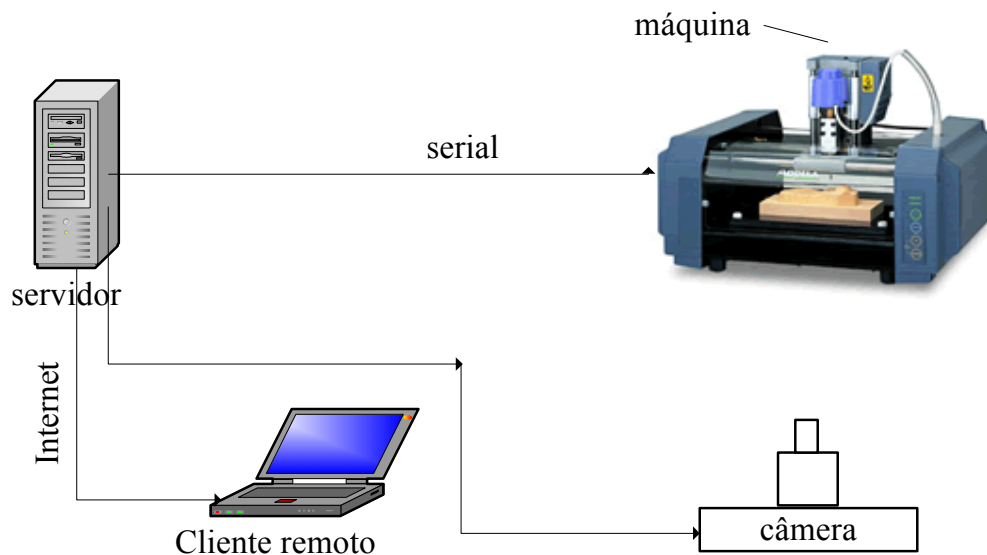


Figura 2.1. Modelo de manufatura cliente-servidor (adaptado de ÁLVARES, 2005).

2.6 World Wide Web - WWW

Até o final dos anos 80 as informações compartilhadas na Internet consistiam, primariamente, de trocas de mensagens de correio eletrônico e arquivos de dados de

computadores. Nessa época começaram a surgir os arquivos multimídia que, além de textos, continham também figuras, sons e ligações (*hyperlinks*) que permitiam ao usuário "saltar" dentro de arquivos, ou até mesmo saltar para outros arquivos contendo informações co-relacionadas.

Surge então a necessidade de criação de novos protocolos para atender a esses requerimentos. O padrão de arquivo HTML (*HyperText Markup Language*) e o padrão servidor HTTP (*HyperText Transfer Protocol*) resultaram de um projeto do CERN (*European Particle Physics Laboratory*), em 1989. Estes padrões forneceram as bases para o surgimento da WWW - o serviço de maior popularidade da Internet e grande responsável pelo seu atual crescimento exponencial (ZOTTO, 1996).

2.7 Conceitos relativos ao CAD

Segundo ROBERTSON e ALLEN (1992), os sistemas de CAD (*Computer-Aided Design*) podem ser definidos como as ferramentas que dão suporte ao projeto e ao processo da engenharia do projeto. Esta definição inclui normalmente as ferramentas computacionais classificadas como Projeto Assistido por Computador (CAD), *Computer-Aided Drafting* e a Engenharia Assistida por Computador (CAE).

Os sistemas de CAD podem potencialmente conduzir a importantes benefícios. Por exemplo, eles podem reduzir a duração do ciclo de desenvolvimento do produto e melhorar as relações com vendedores e clientes. Um menor ciclo de projeto permite às empresas responder mais rápido aos desafios de competitividade, incorporar tecnologias mais novas nos produtos etc.

Em anos recentes, a indústria tem aplicado consideráveis esforços à redução do tempo de desenvolvimento do produto. O uso apropriado dos sistemas de Projeto Assistido por Computador (CAD) poderia ajudar significativamente a atingir essa meta (ROBERTSON e ALLEN, 1992). Os sistemas CAD deveriam suportar qualquer atividade de projeto na sua criação, modificação, recuperação ou documentação.

Apesar da sigla "CAD" incluir o termo "*design*", observa-se que são poucos os casos em que o computador efetivamente projeta alguma coisa, servindo mais como uma ferramenta de auxílio à confecção de desenhos de engenharia. Sua maior contribuição ocorre na modelagem dos produtos e componentes, e no detalhamento de seus

desenhos. Em alguns sistemas CAD, o termo "*design*" foi trocado por "*drafting*", tal sua aplicação como elemento puramente voltado à documentação do projeto, o que em alguns casos pode levar à subutilização do sistema (KERRY, 1997).

2.7.1 Projeto por *Features*

Features correspondem a uma forma de representar o significado de engenharia numa peça, ou de uma montagem, ou de um processo de manufatura, através de entidades geométricas. Assim, as *features* descrevem um modelo de produto de maneira mais inteligente do que a geometria explícita (WARD, 1996). Tem-se como exemplos de *features* um furo, cavidade, chanfro e ranhura, como mostrado na figura 2.2.

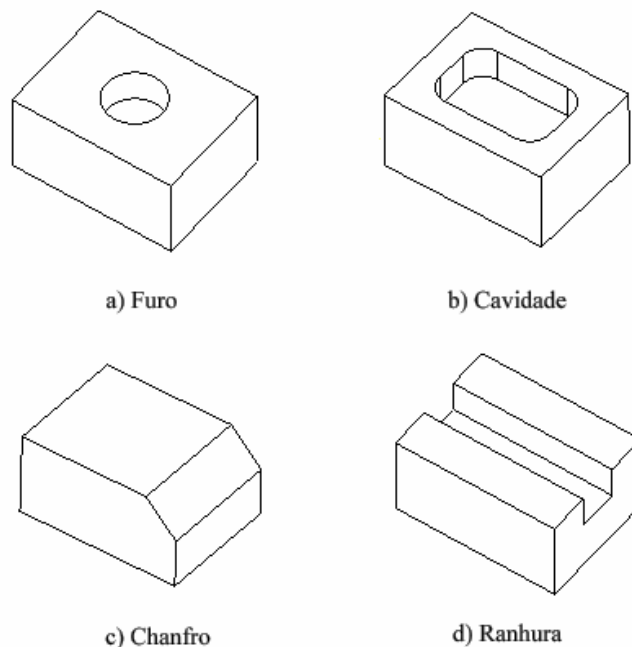


Figura 2.2 exemplos de *features*.

Através de *features*, pode-se compor uma estrutura de dados para representar uma peça ou uma montagem em termos dos seus constituintes, onde:

- cada *feature* tem uma identidade identificável;
- a forma pode ser expressa em termos de:
 - parâmetros dimensionais;
 - enumeração geométrica;

- relações e entidades topológicas;
- etapas de construção para produzir a geometria da *feature*.

As *features* têm significado de engenharia:

- há uma formalização da função que será efetuada pela *feature*;
- pode-se inferir como ela pode ser produzida;
- pode-se identificar as ações que precisam ser feitas para executar uma análise de engenharia;
- pode-se avaliar como a *feature* se comporta em diversas situações.

Segundo SHAH e MÄNTYLA (1995) duas metodologias de projeto por *features* são comumente utilizadas:

1. *Destruction by Machining Features*, também conhecida como *Destructive Solid Geometry* ou *Deforming Solid Geometry* (DSG); e
2. Síntese por Features de Projeto.

A abordagem *Destruction by Machining Features* se inicia com um modelo da matéria-prima (peça bruta) na qual a peça desejada será usinada. O modelo da peça é criado pela subtração na peça bruta de *features* que correspondem ao material removido por operações de usinagem. O sistema Cybercut (<http://cybercut.berkeley.edu>) utiliza esta abordagem, obrigando ao usuário a remover entidades geométricas de uma peça de formato regular através do fresamento e furação da peça bruta (p.ex. um paralelepípedo), de tal forma que o processo de manufatura da peça é incorporado de forma inerente ao projeto. Alguns sistemas comerciais também utilizam esta abordagem. Estes sistemas usam um conjunto de *features* pré-definidas que são subtraídas do sólido base (peça bruta).

Nesta abordagem, o projetista é forçado a definir a geometria da peça usando um conjunto de *features* associadas a um processo de manufatura específico. Exemplos de utilização desta abordagem são os sistemas *Quick Turnaround Cell – QTC* (CHANG et al., 1998), *First Cut* e *Cybercut* (SMITH e WRIGHT, 2000). Para usinagem, as

features disponíveis para o projetista são limitadas às *features* negativas, e todas são subtraídas da peça bruta. A vantagem deste método é que as *features* de usinagem estão diretamente disponíveis no modelo da peça, não sendo necessário o reconhecimento ou mapeamento de *features* de manufatura. O projetista está interessado inicialmente na forma da peça e nos aspectos funcionais.

A segunda abordagem, síntese por *features* de projeto, difere da abordagem anterior com relação ao modelo que pode ser construído tanto pela adição quanto pela subtração de *features*, não sendo necessário iniciar com um modelo de peça bruta.

Em todas as abordagens de projeto por *features*, as peças são criadas diretamente usando *features* e o modelo geométrico é gerado do modelo de *features*. Isto requer que o sistema de projeto (CAD) tenha definições genéricas de *features* disponibilizadas pela biblioteca de *features*, permitindo a instanciação das *features* pela especificação das dimensões, parâmetros de localização, a *feature*/face/aresta sobre a qual está localizada e vários outros atributos (propriedades), restrições e relacionamentos. A figura 2.3 apresenta um diagrama da técnica de criação de *features* utilizada na metodologia.

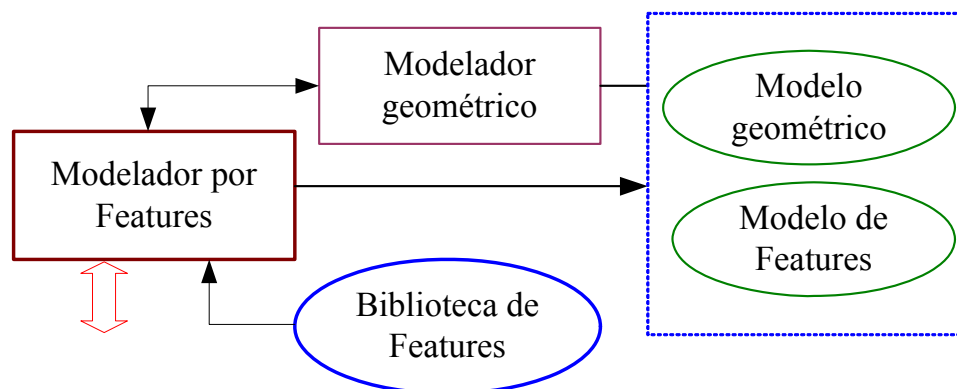


Figura 2.3. Abordagem de criação de *features* utilizando projeto por *features* (SHAH e MANTYLA, 1995).

O usuário interage com o modelador de *features* e constrói o modelo de *features* a partir da instanciação de *features* disponibilizadas na biblioteca de *features*. Este modelo de *features* é utilizado pelo modelador geométrico que cria então o chamado modelo geométrico da peça. O modelo de *features* define a forma, atributos dimensionais, posição das *features*, restrições geométricas e atributos não-geométricos (*features* geometrias de materiais, *features* de tolerância, entre outros). Já o modelo

geométrico é constituído por entidades topológicas, gráficos topológicos, entidades geométricas e ponteiros topológicos-geométricos. Modelos geométricos são representados normalmente por modelos gráficos 2D e 3D (arame), modelos de superfície e modelos sólidos.

2.8 CAPP (Computer Aided Process Planning)

Segundo GROOVER, (1987), o planeamento do processo pode ser definido como sendo “a atividade responsável pela determinação dos processos e sua seqüência para a transformação da matéria prima na peça acabada”. O planeamento do processo é considerado uma tarefa difícil e bastante especializada. Alguns dos fatores que causam esta dificuldade são:

- Existem diferentes maneiras de fabricar-se uma peça; e
- O chão de fábrica é um sistema dinâmico que normalmente apresenta restrições à execução do documento resultante do planeamento do processo, chamado de “plano (ou folha) de processos”.

Sistemas CAPP podem ser definidos como a aplicação de computadores para assistir o processista no planeamento do processo. O CAPP tem por objetivo reduzir o tempo e esforço necessários para preparar planos de processo, e obter-se planos mais consistentes. Um outro objetivo é fornecer a interface entre o CAD e o CAM, visando a integração da manufatura”. Segundo FERREIRA (1996), “a comunicação entre o CAD e o CAM é fundamental em um ambiente CIM (Manufatura integrada por Computador), e pode até certo ponto determinar o sucesso do mesmo.

2.8.1 Abordagens de Sistemas CAPP

Segundo GROOVER, 1987 e DETAND, 1993, os diversos métodos de planeamento do processo assistido por computador podem ser classificados em dois tipos básicos: variante e generativo. Contudo, alguns sistemas que foram construídos apresentam uma combinação das anteriores dando origem à abordagem semi-generativa, também chamada de híbrida. A figura 2.4 apresenta um esquema dos dois tipos principais, variante e generativo, de abordagens de CAPP.

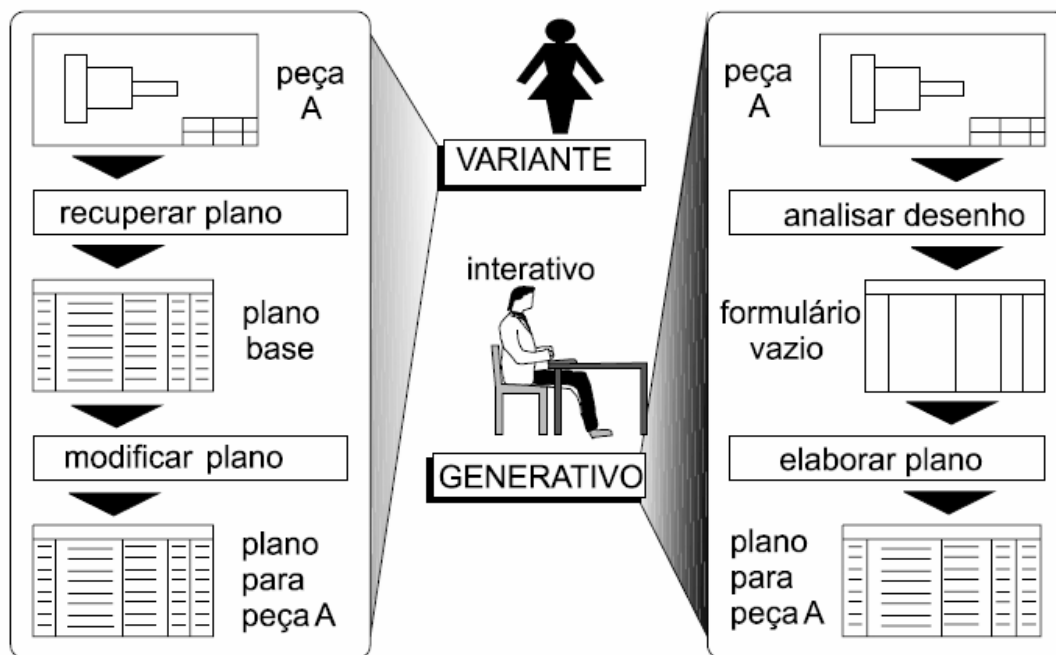


Figura 2.4. Tipos de CAPP, (KERRY, 1997).

A seguir é feita uma breve descrição destas abordagens, variante e generativo, de CAPP.

2.8.2 Abordagem variante

Esta abordagem foi a primeira técnica de planejamento do processo computadorizado, e é baseada no conceito de que peças similares terão planos de processos similares. O planejamento variante utiliza um plano de referência, também chamado de plano base, a partir do qual são feitas modificações para obter-se um novo plano de processo. Esta abordagem pressupõe que haja famílias de peças e existam planos padrão para cada uma destas famílias. Entretanto, há alguns casos em que não se utilizam famílias de peças, e sim apenas planos semelhantes, que não requerem a formação de famílias e que proporcionam uma sistematização de curto período e investimento. A grande desvantagem de se trabalhar somente com planos semelhantes, é que estes normalmente devem ser muito modificados para cada nova peça (WANG e LI, 1991).

Em comparação com o planejamento do processo manual, a abordagem variante é bastante vantajosa, pois a manipulação de informações se torna bem mais simples e os planos de processos podem ser padronizados. Contudo, neste tipo de abordagem, a qualidade final do plano do processo ainda depende da habilidade do processista que realiza as modificações necessárias às particularidades de cada peça. Sendo assim, nos

sistemas variantes, o computador é apenas uma ferramenta de auxílio às atividades de planejamento do processo. Em ÁLVARES et al. (2002) é apresentada a concepção de um sistema CAPP variante concebido para *Web*.

O CAPP variante está associado à codificação e classificação de peças baseadas na Tecnologia de Grupo¹, onde as peças são classificadas e codificadas baseadas em várias características ou atributos. Um sistema de codificação de Tecnologia de Grupo pode ser usado para a recuperação de planos de processos para peças semelhantes. Grande parte dos sistemas CAPP variantes relatados na literatura utilizam a Tecnologia de Grupo como ferramenta de implementação.

2.8.3 Abordagem generativa

Segundo FERREIRA (1996), o planejamento do processo generativo pode ser definido como um sistema que sintetiza a informação dos processos visando gerar automaticamente um plano de processo para uma nova peça. A lógica de decisão e fórmulas de otimização são codificadas no próprio sistema, reduzindo-se ao mínimo a intervenção humana. Um sistema generativo tem por objetivo produzir um plano de processo completo, iniciando-se com os dados de projeto advindos do desenho da peça, e continuando através do conhecimento completo dos processos de manufatura. Ele também requer técnicas matemáticas sofisticadas para os aspectos combinatórios de programação e descrição da peça.

A entrada de informações sobre a peça pode ser do tipo texto, onde o usuário responde a uma série de questões predefinidas; ou graficamente, onde as características da peça são definidas através de um módulo de CAD. Quando se pensa em um sistema CIM, a utilização de uma interface gráfica para a definição da peça é a maneira mais interessante, pois desta forma a comunicação entre os módulos CAD e CAPP fica prontamente estabelecida. A grande vantagem deste tipo de abordagem é que os planos de processos gerados são padronizados e completamente automatizados. Este tipo de abordagem torna-se bastante atraente para empresas que trabalham com uma grande variedade de produtos que são produzidos em pequenos lotes.

¹ A tecnologia de grupo (GT) pode ser definida como “o estudo de uma grande população de itens aparentemente diferentes e a sua divisão em grupos com características similares. A utilização típica da tecnologia de grupo aparece no conceito de famílias de peças, onde a codificação e a classificação são utilizadas” (REZENDE, 1996).

2.8.4 Híbrido

Enquanto o planejamento do processo completamente generativo é a meta do desenvolvimento de sistemas CAPP, os sistemas que combinam as abordagens variante e generativa também são úteis, os quais são chamados de sistemas híbridos. Um outro termo utilizado para referir-se a esta abordagem é o termo semi-generativos (ALTING e ZHANG, 1989).

Uma possível implementação de um sistema híbrido consiste na utilização parcial de um sistema variante, baseado em Tecnologia de Grupo, para recuperar um plano do processo existente, e posteriormente o mesmo sistema pode utilizar técnicas de um sistema generativo, como por exemplo, para modificar o plano recuperado anteriormente visando satisfazer os requerimentos de uma nova peça (SALEH, 2005).

Um outro tipo de implementação consiste na utilização inicial de técnicas do planejamento generativo, e posteriormente na possibilidade de interação do usuário com o sistema. O usuário atua de maneira a ligar as funções de planejamento ou torna-se parte do ciclo de realimentação em um ciclo avaliado e atualizado. Deste modo, o usuário, sempre tem o controle sobre o planejador e faz as decisões finais quando houver conflitos que não possam ser resolvidos automaticamente.

2.8.5 Ferramentas Computacionais para CAPP

Existem vários métodos para descrever a estrutura de decisão no planejamento do processo. Os métodos de representação do conhecimento se relacionam diretamente à lógica de decisão nestes sistemas. Segundo WANG e LI (1991), os seguintes métodos de lógica de decisão são utilizados no planejamento do processo: tabelas de decisão e técnicas baseadas em inteligência artificial, destacando-se os sistemas especialistas, lógica difusa, redes neurais, sistemas de multigentes e algoritmos genéticos.

Nas últimas décadas muitos sistemas de planejamento do processo generativos têm utilizado técnicas de inteligência artificial. O uso destas técnicas tem melhorado o desempenho dos sistemas de CAPP (CHU et al., 2000).

2.9 CAM

O CAM (*Computer-Aided Manufacturing*) refere-se a todo e qualquer processo de fabricação controlado por computador. Sua origem remonta-se ao desenvolvimento das

máquinas controladas numericamente (CN) no final dos anos 40 e início dos 50. Quando estas máquinas começaram a ser controladas por computador, no fim dos anos 50 e início dos 60, surgiu o termo CNC (*Computer Numeric Control*). Atualmente a sigla CNC engloba diversos processos automáticos de fabricação, tais como; fresamento, torneamento, oxicorte, corte a laser, entre outros. Assim sendo, o termo CAM pode ser empregado para todas estas disciplinas e para qualquer outra que possa surgir.

Apesar de toda esta abrangência, o termo CAM, às vezes, ainda é sinônimo da programação CN, conceito que ficou muito difundido com a sigla CAD/CAM, que representa módulos de programação CN em sistemas CAM.

Esse conjunto de *softwares* (CAD-CAM-CNC) é condição necessária, porém não suficiente, para que uma dada empresa possa ter um sistema integrado de manufatura. Não obstante, esse “pacote tecnológico” propicia um diferencial competitivo. Entretanto, o investimento em *softwares* CAM é muito elevado. Nesse investimento, destacam-se dois componentes: o preço de aquisição e o valor desembolsado em treinamento de pessoal (DA COSTA e PEREIRA, 2006).

2.10 Prototipagem Rápida

O termo prototipagem rápida designa um conjunto de tecnologias usadas para fabricar objetos físicos diretamente a partir de fontes de dados gerados por sistemas de projeto auxiliado por computador (CAD). O termo "rápido" associado a esses processos é relativo. A construção de alguns protótipos pode levar de 3 a 72 horas, dependendo do tamanho e complexidade do objeto. (GORNÍ, 2001).

O aumento no uso dos computadores tem contribuído para os avanços em muitas áreas relacionadas à computação, como CAD, CAM e CNC. No entanto, quando se examina os numerosos sistemas disponíveis hoje, pode-se concluir que junto com o CAD muitas outras tecnologias e avanços em áreas como sistemas de manufatura e materiais também são essenciais para o desenvolvimento das tecnologias de prototipagem rápida. No processo de desenvolvimento do produto a prototipagem rápida tem seu lugar, a qual ocorre antes da produção de um lote de peças, devido a que antes do produto ser lançado no mercado, deve ser testada a funcionalidade, fabricabilidade, montabilidade, e outras propriedades que não estão visíveis no modelo 3D (FERREIRA e LAFRATTA,

1993). Segundo JACOBS (1992), os processos de prototipagem rápida são classificados como:

- **Processo subtrativo:** O qual se inicia com um bloco de material sólido maior que o tamanho final desejado e o material é removido até que a forma desejada seja obtida.
- **Processo Aditivo:** É o inverso do processo subtrativo, onde o produto final possui muito mais material do que quando ele iniciou o processo. O material é manipulado de forma que sucessivas porções ou camadas deste material combinem-se para formar o objeto desejado.
- **Processo conformativo:** É aquele onde forças mecânicas são aplicadas sobre o material em sua geometria original para se obter a forma desejada.

Os processos de usinagem são os tradicionais métodos de prototipagem rápida, os quais são subtrativos por natureza. Porém, a maioria dos outros métodos são aditivos.

A máquina de prototipagem rápida presente no laboratório GRIMA usa o método de prototipagem rápida por subtração que é o método mais tradicional. Os detalhes técnicos e demais características desta máquina de prototipagem rápida por usinagem serão apresentados no capítulo 4.

2.11 Conclusões do capítulo

Neste capítulo foram apresentados conceitos que servem de fundamentação teórica para o presente trabalho de mestrado.

Alguns desses conceitos como, por exemplo, a abordagem de projeto por *features*, assim como a abordagem generativa de sistemas CAPP, o modelo cliente-servidor, entre outras, são adotados como paradigmas no desenvolvimento do sistema integrado que é motivo desta dissertação.

No próximo capítulo se verá alguns sistemas de fabricação remota desenvolvidos, e as abordagens que foram utilizadas no desenvolvimento dos mesmos.

CAPÍTULO 3

REVISÃO BIBLIOGRÁFICA DOS SISTEMAS CORRELATOS

Este capítulo apresenta trabalhos relevantes desenvolvidos na área do presente trabalho de Mestrado, e buscar-se-á descrever cada um desses sistemas.

3.1 Sistema Webmachining

Webmachining ÁLVARES (2005) é um sistema para o projeto e fabricação de peças rotacionais a distância via Internet, através do qual propõe-se uma metodologia que utiliza a modelagem baseada na síntese da peça por *features* de projeto (isto é, adição de *features* para operações de torneamento e subtração de *features* para operações de fresamento e furação), a fim de permitir as atividades de projeto (CAD), planejamento do processo (CAPP) e fabricação (CAM) utilizando como referência o modelo de *features* de manufatura.

O procedimento se inicia na modelagem de uma peça por *features* num contexto de manufatura remota utilizando a *Web* como meio de comunicação, num modelo computacional cliente/servidor. Os clientes se conectam ao modelador de *features* via *Web* e iniciam a instanciação de uma nova peça a partir de uma base de dados, usando a biblioteca de *features* padronizadas que o sistema disponibiliza. Após a conclusão e validação dos modelos, a peça criada é armazenada e disponibilizada para os demais módulos do sistema.

O sistema WebMachining foi concebido num ambiente distribuído de agentes de *softwares* interoperáveis denominado “Comunidade de Agentes de Manufatura”. Um agente pode ser definido como sendo "um software que pode efetuar tarefas específicas para um usuário, e possui um grau de inteligência que permite que ele efetue tarefas autonomamente, e interaja com o seu ambiente de uma maneira útil" (BRENNER et al. 1998).

A arquitetura do sistema WebMachining é estratificada em três camadas: Projeto, Planejamento do Processo e Fabricação (figura 3.1). Há um fluxo de informações entre as três camadas.

Na camada superior, reside um grupo de agentes de projeto, os quais atuam como uma ferramenta CAD colaborativa baseada em *features* e permite aos usuários conectar-se aos níveis inferiores. Esse módulo chama-se *WebCADbyFeatures*.

A camada intermediária consiste do agente de planejamento do processo que incorpora as características de um CAPP generativo utilizando métodos de representação do conhecimento baseados em inteligência artificial. Este agente interpreta as definições de projeto, modelagem por *features* e geometria utilizada pelo cliente remoto, e determina como fabricar a peça.

O agente de fabricação (*Webturning*), que reside na camada inferior, permite a teleoperação da máquina de controle numérico via Internet e a monitoração do processo de usinagem. As informações sobre recursos de fabricação disponíveis (máquinas, ferramentas, materiais, dispositivos de fixação, etc.) estão disponíveis através de um banco de dados que pode ser acessado pelos diversos agentes do sistema.

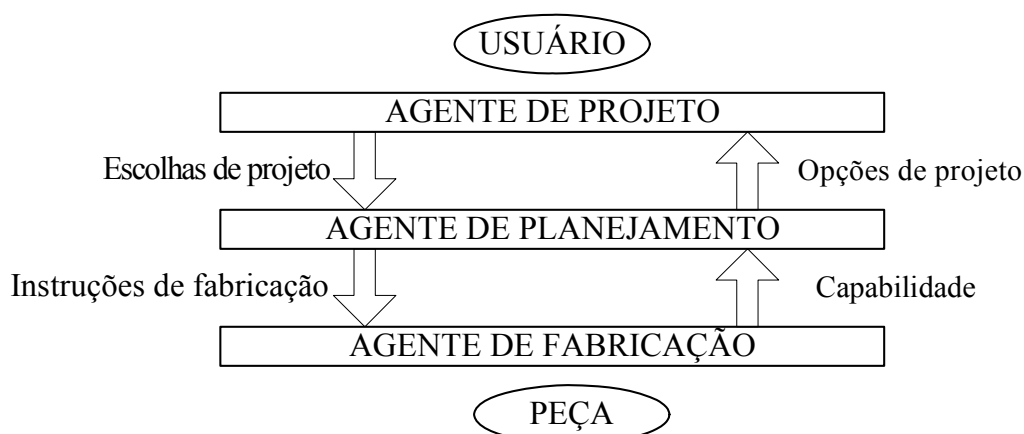


Figura 3.1. Diagrama das camadas de multiagentes do sistema *WebMachining* (ÁLVARES, 2005)

O sistema utiliza a plataforma JATLite (*Java Agent Template Lite*) (<http://java.stanford.edu/index.html>), que é um pacote de programas escritos em Java que permite aos usuários a criação de agentes de softwares que se comunicam de forma robusta na Internet. JATLite oferece uma infra-estrutura básica na qual agentes registrados com o *Agent Message Router* (AMR) usam um nome e uma senha, conectando-se e desconectando-se da Internet, mandando e recebendo mensagens, transferindo arquivos via FTP, e geralmente trocando informações com outros agentes que podem estar sendo executados em diferentes computadores (ÁLVARES e

FERREIRA, 2003) . Esta ferramenta também foi utilizada na implementação do sistema *CyberCut* (DORNFELD et. al, 2001).

A seguir serão apresentadas algumas técnicas utilizadas pelo sistema WebMachining nos seus diversos módulos, em especial no módulo associado ao CAPP.

3.1.1 Interface Gráfica com o Usuário (GUI)

A funcionalidade necessária para suportar operações baseadas em *features* em uma interface para sistema CAD está associada a menus, opções de visualização, *prompts* e mensagens de erro, interações com o usuário de forma gráfica e textual, seleção de *features* e de bibliotecas de *features*, instanciação de *features*.

Os principais componentes deste módulo GUI são: Interface gráfica 2D, componentes de visualização 2D e 3D processadas localmente na própria interface, componentes de visualização 2D (arquivos jpg e gif) e 3D (Kernel ACIS² e VRML) com processamento remoto, *browser*, *plug-ins* Java, *applets* Java e *sockets* TCP/IP.

3.1.2 Mapeamento de *Features*

O módulo de projeto WebMachining fornece uma entrada para o sistema de planejamento de processo. Para evitar dificuldades para o projetista em termos da entrada de dados a ser realizada a partir de um menu, se faz uso do conceito de *features* no nível do usuário e *features* no nível do modelo. Assim, o usuário projeta a peça com um simples conjunto de *features* no nível do usuário disponibilizadas pela applet GUI. Estas *features* são convertidas em *features* no nível de modelo com maior detalhamento. O ambiente de projeto se baseia em *features* de alto nível como chanfro, furo, rasgo de chapeta, etc. que serão mapeadas posteriormente em operações de usinagem. Operações de torneamento podem ser determinadas a partir de um desenho que representa o perfil 2D da peça modelada por *features* de forma, segundo o seu contorno. O volume a ser usinado é obtido em função da forma da peça bruta especificada. A interface gráfica deste sistema pode ser vista na figura 3.2, onde foi modelada uma peça rotacional em 2D.

² *Kernel ACIS* (www.spatial.com) é uma biblioteca que proporciona, a través de software, as rotinas matemáticas adequadas as quais permitem aplicações para construir modelos em 3D.

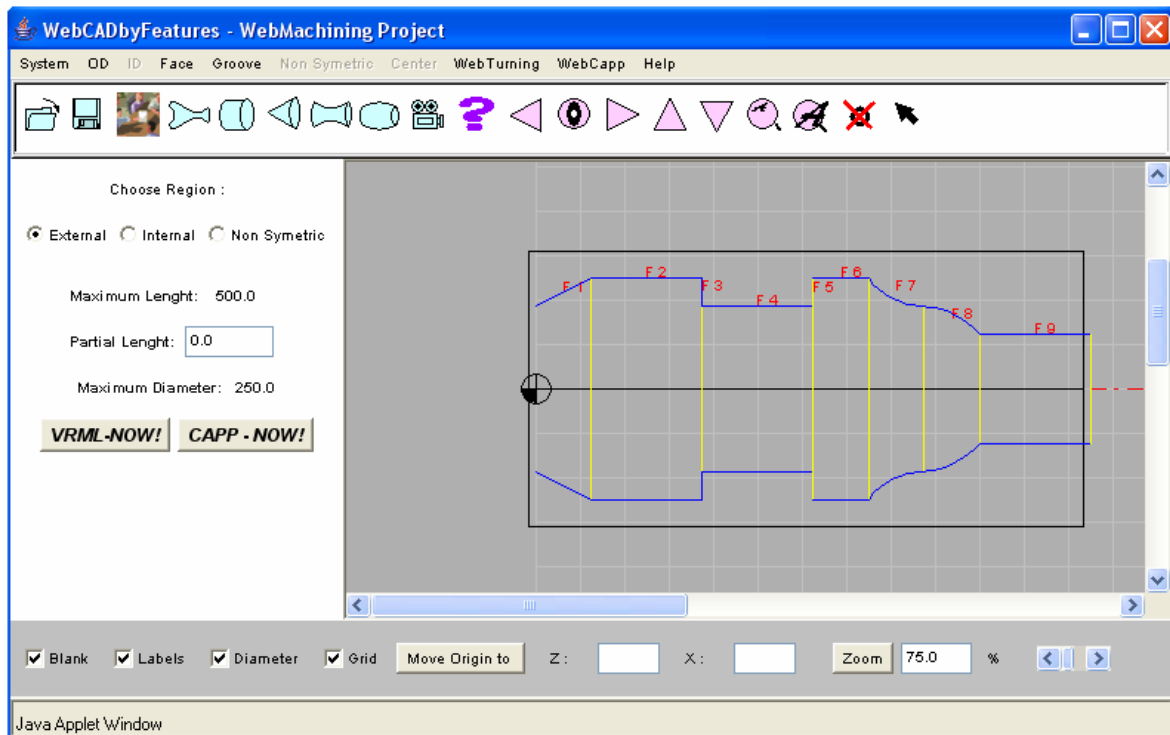


Figura 3.2. Interface gráfica do Módulo CAD do sistema *WebMachining*.

3.1.3 Operações de usinagem

O sistema *WebMachining* considera operações de torneamento cilíndrico externo e interno (mandrilamento), faceamento, furação, sangramento interno e externo e rosqueamento interno e externo em um centro de torneamento. E também são permitidas operações de fresamento e furação radiais e axiais através da ferramenta rotativa (eixo C) do centro de torneamento.

3.2 Sistema Cybercut

Cybercut é um sistema CAD/CAM baseado na Internet, e o mesmo foi especificamente projetado para ser um sistema automatizado (AHN et al., 2001, BROWN e WRIGHT, 1998). É constituído por três componentes:

1. Software de CAD escrito em linguagem de programação Java usando *applets* via páginas *Web*. O sistema CAD é baseado no conceito de *Destructive Solid Geometry* (DSG), o qual obriga o usuário a remover entidades geométricas (*features*) de uma peça de formato de paralelepípedo através do fresamento da peça bruta, de tal forma que o processo de manufatura da peça é incorporado, ao longo do processo de projeto.

2. Um sistema de Planejamento de Processo Auxiliado por Computador (CAPP) que acessa uma base de conhecimento contendo informações sobre ferramentas e dispositivos de fixação.
3. Uma fresadora CNC de arquitetura aberta (marca HAAS, modelo VF-1) que pode receber informações de planejamento e projeto em linguagem de alto nível e executar a usinagem .

Com acesso à interface de CAD, denominada WebCAD3D (WANG, 1998), a partir da Internet, qualquer engenheiro com um *browser Web* se torna um potencial usuário desta ferramenta de prototipagem rápida *on-line*. A GUI, ilustrada na figura 3.3, possibilita que um usuário remoto seja capaz de fazer o *upload* do arquivo CAD em um formato específico, para o servidor Cybercut, o qual irá executar o planejamento do processo e gerar o código G apropriado para a fresadora. A peça poderá ser usinada e enviada para o projetista. Assim, o engenheiro pode ter um protótipo dentro de poucos dias por uma fração do custo que a fabricação *in-house* exigiria. O Cybercut trabalha com processos de fresamento e furação .

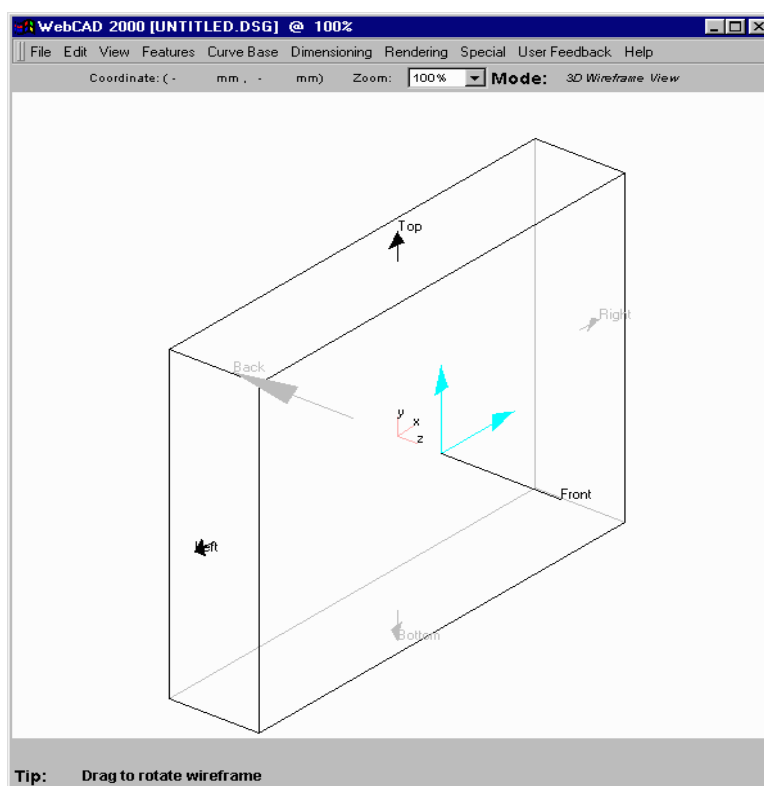


Figura 3.3 Interface Gráfica WebCAD3D (Cybercut).

Para a troca de dados entre os ambientes de projeto e manufatura, o formato dos dados pode incluir tolerâncias e informações específicas de usinagem, como raio de canto. O usuário faz uso de um conjunto de *features* de usinagem pré-definidas, armazenadas em uma biblioteca, que podem ser fabricadas na máquina disponível. Esta biblioteca de projeto está situada no servidor.

No sistema Cybercut a modelagem por *features* de usinagem permite uma fácil interoperabilidade entre projeto e fabricação, e é adequado para sistemas de projeto e fabricação de peças prismáticas. Durante a criação das *features*, um verificador de regras de manufatura garante a manufaturabilidade do projeto. Exemplos de regras de manufatura incluem: (a) a largura de uma ranhura deve ser pelo menos igual ao diâmetro da menor fresa presente no banco de dados (figura 3.4a); (b) a profundidade da ranhura deve ser no máximo igual ao comprimento da fresa a ser usada na usinagem (figura 3.4b).

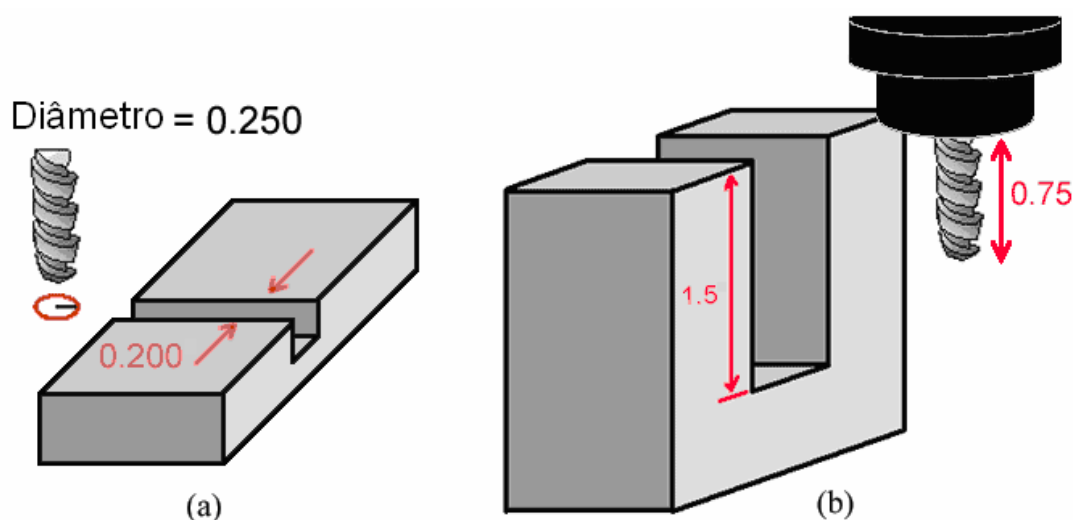


Figura 3.4 Algumas regras de manufatura do sistema Cybercut.

3.3 Sistema WebSPIFF

O sistema WebSpiff (BIDARRA et. al., 2001) é baseado em uma arquitetura cliente/servidor consistindo de muitos componentes (figura 3.5). No lado do servidor tem-se dois componentes principais:

1. Sistema de modelagem SPIFF: fornece toda a funcionalidade para modelagem baseada em *features*, utilizando o *kernel* de modelagem ACIS.

2. Gerador de Sessão: fornece a funcionalidade para iniciar, associar fechar uma sessão de modelagem, bem como gerencia todas as comunicações entre o sistema SPIFF e os clientes (usuários).

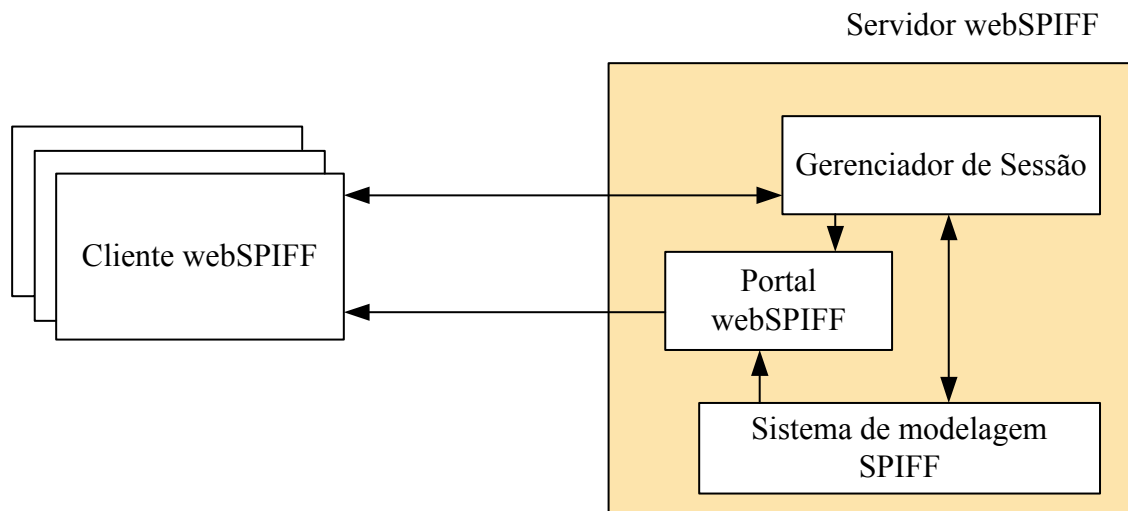


Figura 3.5. Arquitetura do sistema WebSpiff (BIDARRA, 2001).

Os componentes do portal WebSPIFF fornecem o acesso inicial a uma sessão para um novo cliente, que inclui um servidor Web onde os dados no modelo são disponibilizados para *download* pelos clientes. Os clientes executam operações localmente, que incluem visualização, interação com o modelo baseado em *features*, mensagens semânticas de alto nível, especificação de operações de modelagem, bem como atualização de dados do cliente enviados via rede.

O servidor coordena a sessão de colaboração mantendo um modelo de produto centralizado e provê todas as funcionalidades que não devem ser implementadas no lado do cliente. Uma importante vantagem desta arquitetura é que existe apenas um modelo de produto centralizado, evitando-se inconsistências entre múltiplas versões do mesmo modelo.

A atual versão de sistema WebSpiff (no lado do servidor) oferece múltiplas visões sobre o modelo do produto. Cada visão consiste de um modelo de *features* contendo *features* específicas para a aplicação correspondente à visão. Há duas visões disponíveis: uma de projeto e uma para o planejamento do processo e manufatura de peças. Na visão do projeto, o modelo de *features* consiste de *features* aditivas (protuberâncias) e *features* subtrativas (ranhuras e furos). Na visão de planejamento do processo e manufatura, o

modelo de *features* consiste apenas de *features* subtrativas. Também estão disponíveis funcionalidades associadas à validação e manutenção de *features*.

Os clientes do sistema WebSpiff utilizam o *Web browser* mais *plug-ins* para Java e Java 3D. O *applet* Java, que implementa a GUI, efetua a conexão com o Gerenciador de Sessão, disparando-o no lado do servidor. O usuário pode então agora conectar-se com uma sessão colaborativa em andamento ou iniciar uma nova sessão. Uma peça modelada nesse sistema é apresentada na figura 3.6.

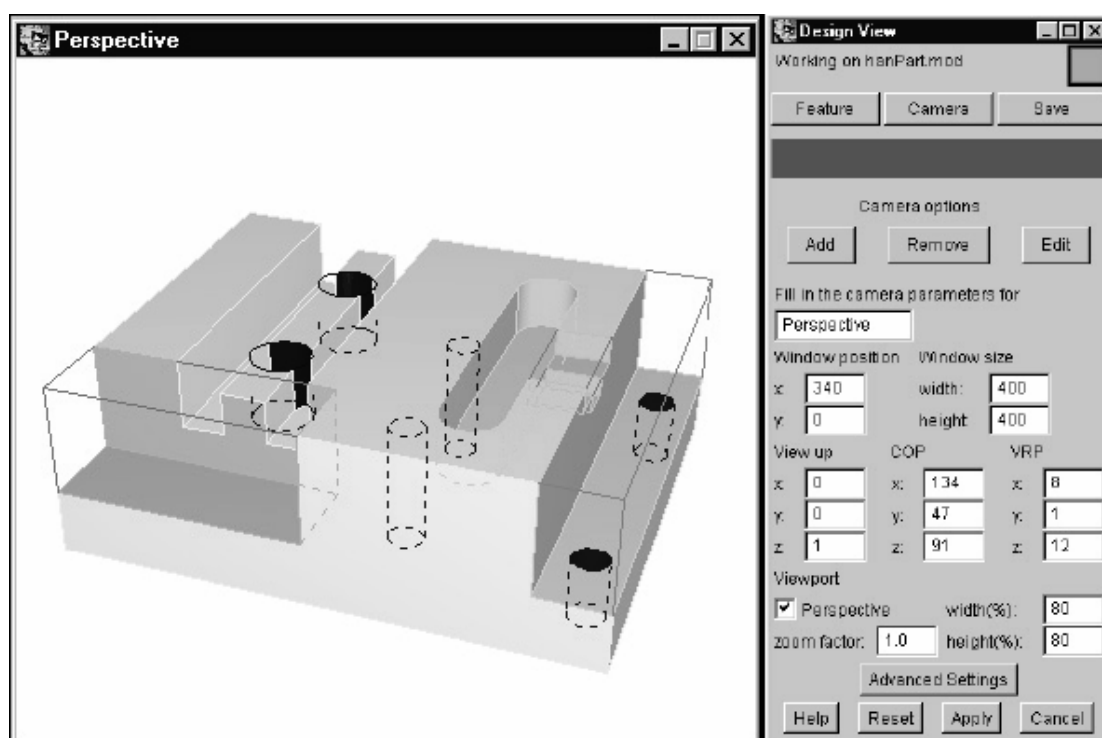


Figura 3.6 Interface Gráfica do sistema WebSpiff (VAN DER BERG et al. 2001).

Muitas das funcionalidades do sistema de modelagem SPIFF, baseado no *kernel* ACIS e na linguagem TCL/TK, são portados para o cliente. Dentre estas funcionalidade, se destacam câmeras *windows*, ou seja, janelas separadas nas quais uma representação gráfica do modelo do produto é apresentada. Facilidades de visualização incluem a apresentação da imagem do modelo em 3D utilizando a API Java3D, realizando a renderização no servidor. A apresentação das imagens renderizadas é feita no formato “jpeg” no lado do cliente. O servidor também gera o formato VRML, o qual é enviado para o cliente.

O envio de um arquivo VRML para o cliente demanda pequena largura de banda, algo em torno de 10 kbytes para um modelo de *feature* de média complexidade. Os arquivos no formato jpeg são da ordem de 100 kbytes.

3.4 Outros Sistemas de Manufatura Remota

3.4.1 O “Toronto Virtual Enterprise” (TOVE)

O laboratório de integração de empresas do departamento de engenharia mecânica e industrial da universidade de Toronto, no Canadá, tem esta linha de pesquisa. As pesquisas incluem a modelagem de uma empresa, engenharia concorrente e gerenciamento da cadeia de suprimentos integrada (FOX, 1996).

O resultado deste projeto é um ambiente baseado em métodos e técnicas da inteligência artificial distribuída (DAI). A DAI utiliza agentes inteligentes que são entidades computacionais básicas implementadas como sistemas de software e hardware distribuídos chamados de sistemas de multi-agentes.

A modelagem da empresa define as ações e as restrições para o planejamento e a produção que satisfaça as metas da empresa. Para a modelagem das empresas se utiliza a teoria das ações complexas (GRUNINGER e PINTO, 1995). O sistema TOVE foi utilizado para modelar duas empresas: uma empresa de manufatura de computadores e uma companhia de engenharia aeroespacial. As ferramentas de software utilizadas foram a linguagem de programação C++ e a ferramenta de representação do conhecimento RockTM.

3.4.2 O projeto ESPIRIT STEP-NC

O projeto europeu ESPIRIT STEP-NC começou em 1 de janeiro de 1999 e terminou em 31 de dezembro de 2001, no entanto grande parte do começo da pesquisa começou na metade da década dos noventa. Vinte parceiros acadêmicos e industriais com ampla experiência nos campos de CAD/CAM (Dassault, Open Mind), de controle (Siemens, OSAI), de máquinas ferramentas (CMS), e usuários (Daimler Chrysler, Volvo) na Europa juntaram este projeto. O protótipo para fresamento foi realizado pelo consorcio

STEP-NC³, e está baseado em sistemas comerciais da Siemens, da Open Mind e da Dassault System.

Alguns dos benefícios da utilização do STEP-NC são:

- Devido à facilidade com que as tarefas são definidas, há uma redução significativa no tempo de planejamento no CAM e a preparação dos dados pode-se conseguir.
- O tempo de usinagem para lotes de trabalho de tamanho meio a pequeno pode ser reduzido devido a que a otimização inteligente pode ser construída dentro do controlador CNC.
- pós-processadores serão eliminados devido a que interface não requiere informação para máquina específica.
- as máquinas ferramentas são seguras y melhor adaptáveis porque o STEP-NC é independente do vendedor da máquina ferramenta.
- o STEP-NC fornece um completo e estruturado modelo de dados, ligado com a informação geométrica e tecnológica, deste modo, a informação não é perdida entre os diferentes estágios do processo.
- Os arquivos XML podem ser utilizados para transferir informação, e em conseqüência permite a manufatura baseada na Web ou e-manufatura.

Os princípios fundamentais do modelo de dados do STEP-NC é a visão de programação orientada a objetos em termos de *features* de manufatura no lugar de direto código de seqüências de movimentação dos eixos e funções da ferramenta como é por definição o ISO 6983. Os objetos neste caso são *features* de manufatura e seus dados do processo associados.

O modelo de dados é uma camada que fornece uma interface standart entre o interpretador do controlador e os diferentes recursos de dados fornecidos. Os dados interpretados são fornecidos para o centro CNC, onde as movimentações nos eixos e a

³ STEP-NC foi feito para substituir o ISO 6983 (códigos standard G e M) por uma moderna linguagem associativa que conecte os dados utilizados no projeto de CAD para determinar os requerimentos de usinagem para uma operação com os dados do processo de CAM que resolva esses requerimentos.

funcionalidade da máquina são gerados. Este novo modelo pode aceitar dados de diferentes fontes. Os dados podem ser gerados por um sistema CAD/CAM, partes de bibliotecas ou grafos e/ou. Estes dados gerados são normalmente processados automaticamente com computadores, assim as definições, às descrições tecnológicas tem sido dadas mais alta prioridade que à compressão do código (XU e HE, 2004)

Os módulos de CAM da Open Mind (Hyper-Fact) e da Dassault Syatem (Catia V5) geraram os programas STEP-NC incluindo dados da geometria do CAD de alto nível, dados da operação e informação do seqüenciamento.

Daimler Chrysler e a Volvo, participando como usuários finais do projeto STEP-NC utilizaram o protótipo para ganhar experiência com a nova interface de programação NC. Desta maneira, não existe mais códigos G e M simplesmente passados para a máquina, logo os câmbios na tecnologia, seqüência do programa ou ainda a geometria, podem ser facilmente realizados e processados sem o necessário tempo de pós-processamento e iterações no departamento de planejamento. Especialmente durante o primeiro período de testes, foram feitos alguns dos últimos câmbios na seqüência do programa e na otimização da tecnologia (XU e HE, 2004).

Outros trabalhos de investigação relatam importantes esforços para a integração CAD/CAPP/CAM, baseados na tecnologia do STEP-NC como apresentado anteriormente, outro exemplo pode ser o de ZHANG et. al. (2000), que apresenta o conceito de Unidades de *Features* de Controle Numérico (NCFU), que desde o ponto de vista dos Sistemas de Manufatura Integrados por Computador (CIMS), podem ser definidas como sendo sub-*features* derivadas das *features* de projeto e de manufatura, e por tanto, as NCFU no somente contêm as estruturas e características gerais de uma *feature*, mas podem também herdar a informação de projeto e manufatura de alto nível de suas super-classes.

O autor apresenta os problemas dos tradicionais sistemas de CAM. Um deles é que a geração das trajetórias da ferramenta tem que ser feita em dois sistemas separados e independentes: o primeiro que é o sistema de programação do controle numérico (NCP) onde se gera a trajetória da ferramenta e o segundo que é o sistema de controle numérico computadorizado (CNC) onde é executado o programa em tempo real. Desse

modo não é possível o controle do processo de usinagem, somente dois parâmetros podem ser controlados: a taxa de remoção e a velocidade de rotação da ferramenta.

Devido ao incremento na velocidade e na capacidade de memória dos computadores, o modo tradicional de programa NC fixo tem sido desafiado pela tendência de desenvolver sistemas NC inteligentes e abertos ou dinâmicos. Uma combinação entre a geração das trajetórias e o controle das mesmas tem se tornado em um problema chave na construção da próxima geração de sistemas de usinagem NC.

As unidades de feature NC são utilizadas para gerar em tempo real as trajetórias e o controle efetivo de uma máquina NC.

O objetivo desse projeto de pesquisa era integrar um sistema de usinagem NC dentro de um sistema de manufatura integrado pela internet. Desenvolveram-se os sistemas de programação do controle numérico (NCP) e de controle do comando numérico (CNC) baseados em NCFU em locais diferentes. O sistema NCP gera um programa NC de nível intermediário na forma de um conjunto de unidades de features de controle numérico (NCFU), logo depois atravessando a internet, este programa NC de intermediário nível pode ser diretamente baixado para um sistema de CNC baseado em NCFU e então poder ser utilizado para controlar a centro NC. Considerando a NCFU como um objeto de operação, a inter-operação baseada no CORBA⁴ pode ser implementada em uma situação de computação distribuída a través da definição das interfaces IDL (Interface Definition Language) comuns.

A arquitetura do sistema de manufatura integrado através da internet está ilustrada na figura 3.7,

⁴ CORBA (Common Object Request Brokerage Architecture) é um padrão de indústria para o modelo de objetos distribuídos. Uma característica chave do CORBA é o IDL (Interface Definition Language). Cada linguagem que suporta CORBA tem seu próprio mapeador IDL.

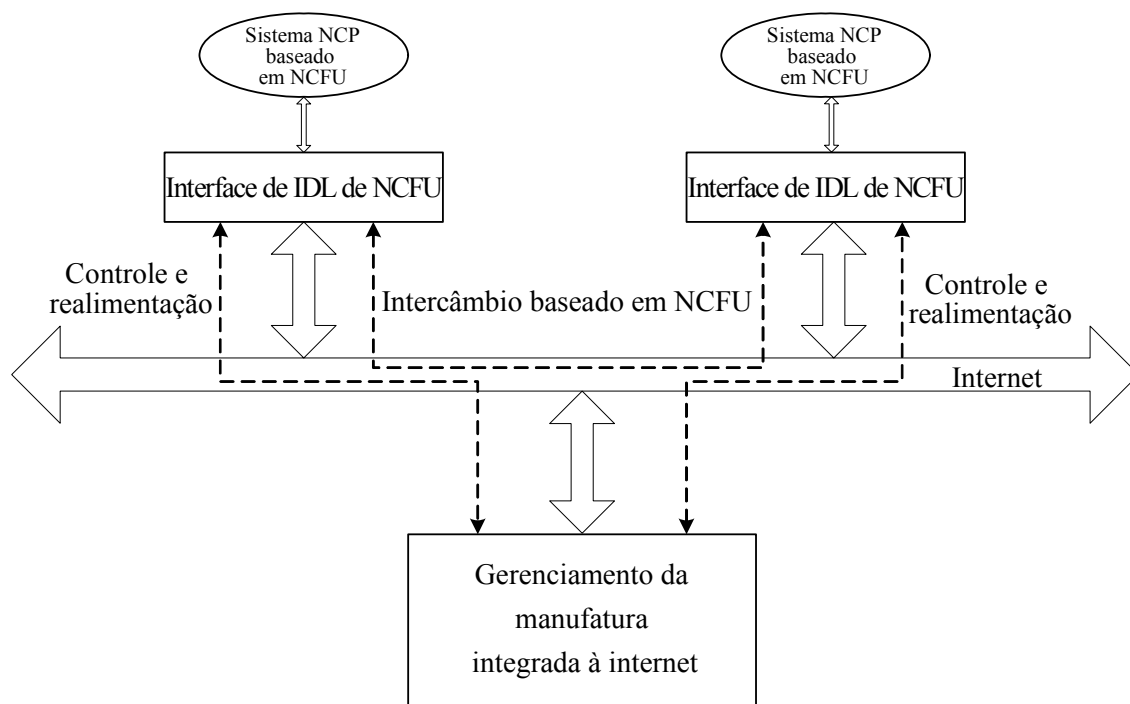


Figura 3.7. Arquitetura de um sistema de manufatura integrado através da internet
(ZHANG, et. al, 2000)

O desenvolvimento pode proceder ao longo de duas frentes. Uma é o desenvolvimento de um sistema de usinagem NC baseado na intranet no meio da rede local. A segunda é o desenvolvimento de um sistema de manufatura verdadeiramente integrado na internet em um meio de manufatura virtual.

3.5 Conclusões do capítulo

No presente capítulo foram relatados alguns sistemas de manufatura remota encontrados na literatura, e como foi visto, nestes sistemas, o projeto por *features* é um paradigma adotado, assim como a abordagem cliente-servidor. Com base nisto, o presente trabalho de mestrado também adota esses paradigmas.

No próximo capítulo ver-se-á a implementação destes conceitos de maneira prática no desenvolvimento do sistema integrado CAD/CAPP/CAM para peças prismáticas de que é motivo a presente dissertação.

CAPÍTULO 4

DESCRIÇÃO DO SISTEMA INTEGRADO CAD/CAPP/CAM

Este capítulo descreve a implementação do sistema integrado CAD/CAPP/CAM voltado para manufatura remota de peças prismáticas utilizando a Internet. Também apresenta as ferramentas utilizadas para o desenvolvimento do mesmo.

O sistema proposto é composto de: (i) módulo CAD, onde são criadas as *features*, (ii) módulo CAPP, onde é gerada a seqüência de usinagem seguindo algumas regras pré-estabelecidas e as trajetórias da ferramenta, e (iii) módulo CAM, onde é gerado o código HPGL (<http://en.wikipedia.org/wiki/HPGL>) e carregado na máquina de prototipagem rápida por usinagem Roland MDX-20.

A figura 4.1 mostra, através de um diagrama IDEF0⁵ (<http://www.idef.com/>), as variáveis de entrada necessárias para o desenvolvimento deste sistema, os mecanismos, os controles, assim como os resultados ou saídas que o sistema gera. Na figura 4.2 é apresentado o diagrama IDEF0 no nível um do sistema integrado, em que podem se ver os três módulos e o fluxo das informações ou dados necessários para que este sistema funcione de maneira integrada.

⁵ IDEF (*Integration DEFinition*) é baseado na Técnica de Análise e Projetos Estruturados (*Structured Analysis and Design Technique - SADT*), que é uma abordagem gráfica para a descrição de um sistema, introduzida por Douglas T. Ross na década de 70. Em 1981 a Força Aérea Americana padronizou e levou ao conhecimento do público um subconjunto do SADT, chamado de **IDEF0**. A necessidade desta padronização deveu-se ao fato de que a Força Aérea trabalhava com diversas indústrias espaciais e cada uma trabalhava de uma forma diferente. Isto dificultava o controle e a documentação de seus processos. Ela permite a representação de uma coleção de atividades e outras ações utilizando-se de ICOMs (*Input, Control, Output, Mechanism, ou* entrada, controle, saída e mecanismo), setas e caixas. Cada atividade ou função é conceitualmente representada por uma caixa retangular, sendo que esta atividade pode ser decomposta em vários níveis. Estes sub-níveis seguem as mesmas convenções. Portanto, um modelo completo de IDEF0 é uma representação hierárquica do processo, decomposta por atividades ou funções em quantos níveis forem necessários (MICHEL, 2002).

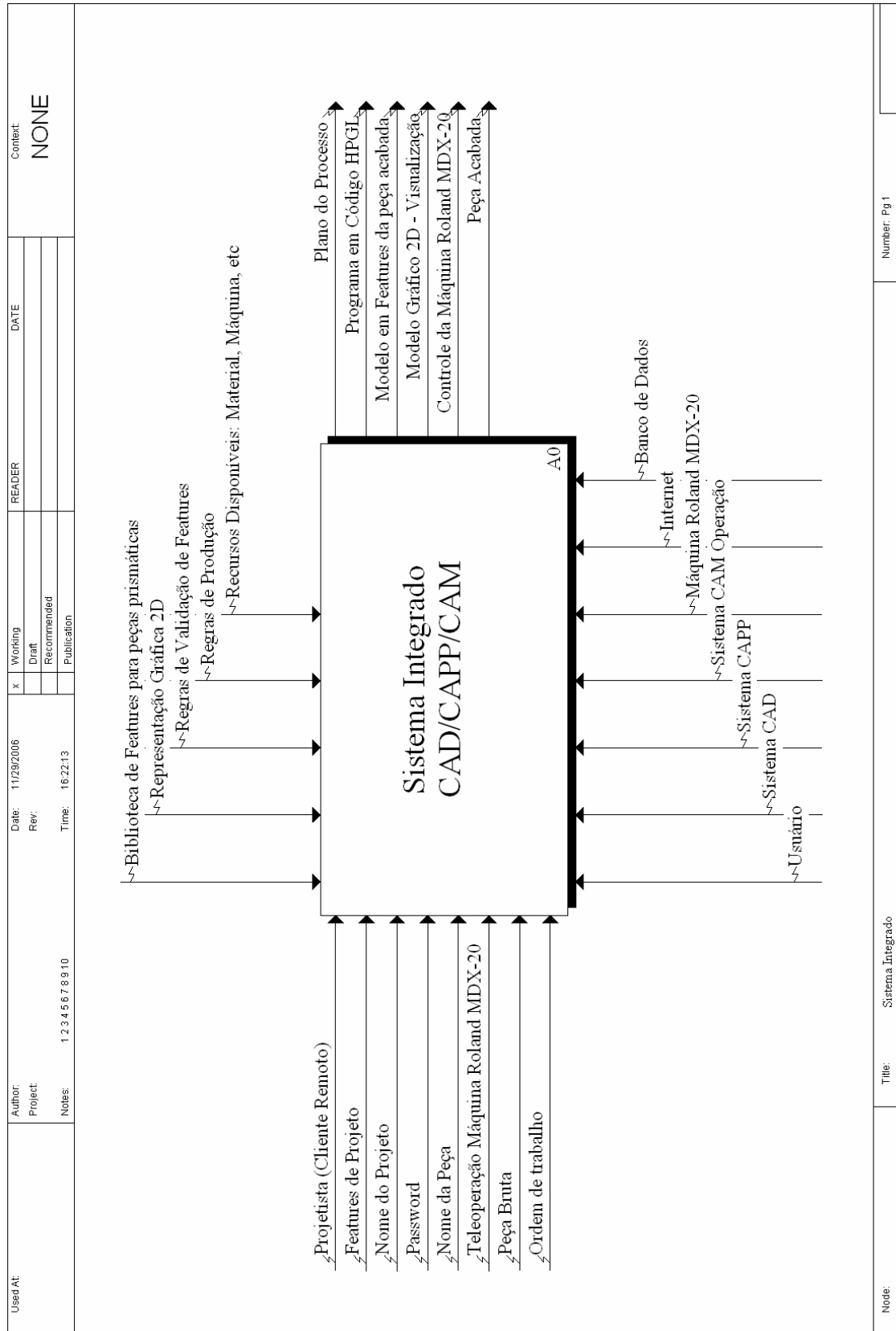


Figura 4.1 Diagrama IDEF0: Sistema integrado CAD/CAPP/CAM, nível A0.

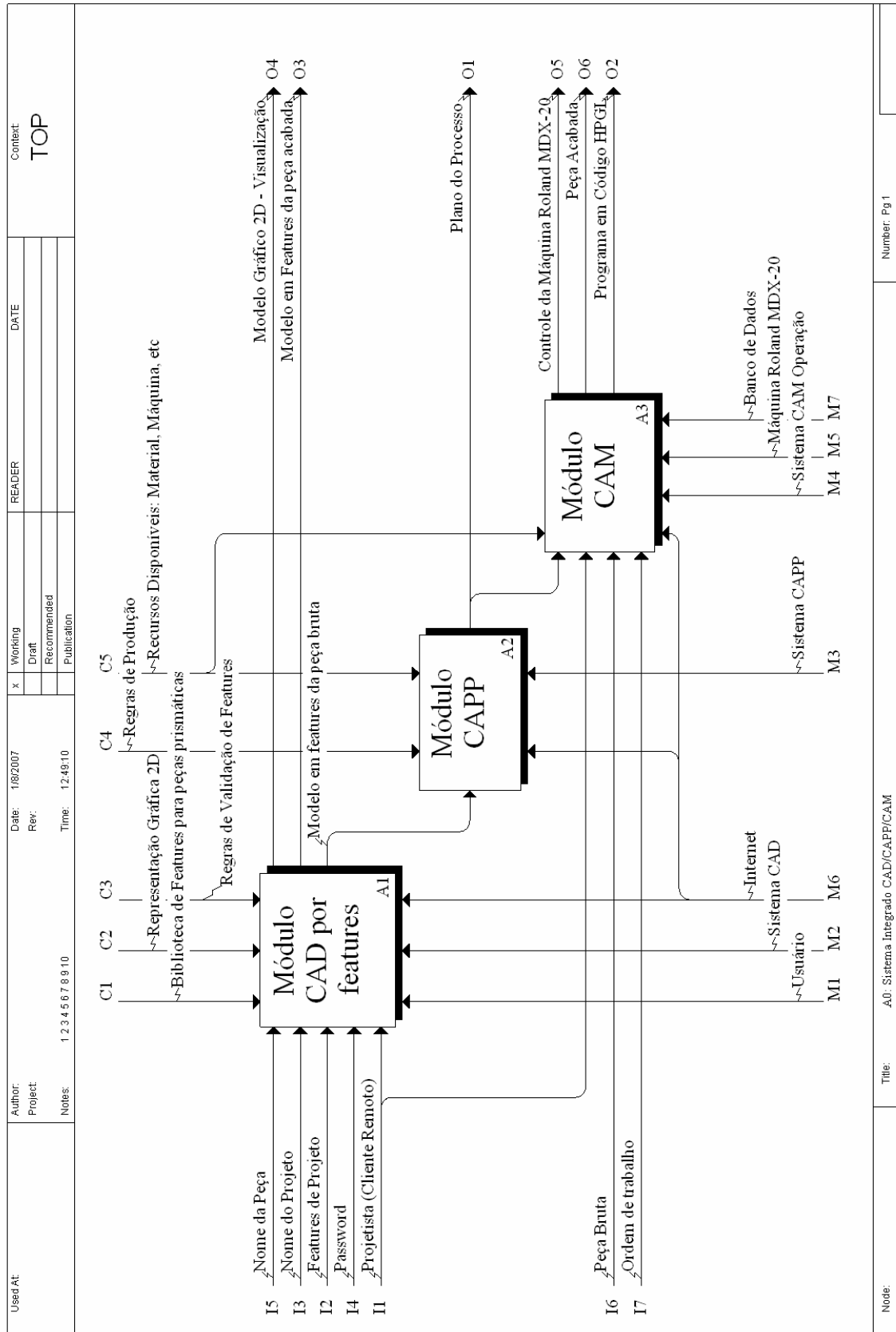


Figura 4.2 Diagrama IDEF0 Sistema CAD/CAPP/CAM integrado, nível 1.

4.1 A Linguagem de programação

Uma das grandes revoluções no mundo do software aconteceu quando a linguagem Java foi anunciada em 1995 pela empresa Sun Microsystems, a qual poderia ser utilizada em sites produzidos na *World Wide Web*. O que chamava a atenção era o fato de que ela podia ser portátil para outros sistemas operacionais. O tamanho também foi um fator importante, pois depois de instalado uma máquina virtual no computador, os aplicativos (*Applets*) eram pequenos, e como a velocidade de transmissão na época era reduzida, os aplicativos poderiam ser visualizados com uma certa rapidez. Com o tempo a linguagem Java evoluiu e se transformou em uma linguagem robusta e complexa (GONÇALVES, 2006).

Nos dias de hoje a linguagem Java é utilizada por grandes bancos, pois fornece extrema segurança, além de grandes empresas que trafegam uma grande quantidade de dados entre outras empresas, uma vez que Java proporciona estabilidade e portabilidade.

Segundo HOSTMANN e CORNELL (2001), algumas aplicações do Java podem ser encontradas em: Servidores Web; Bancos de dados; Computadores de grande porte; Telefones móveis; Sistemas de cartão de crédito.

O Java pode ser executado nas seguintes plataformas: Microsoft Windows. Linux. Sun Solaris. MacOS.

4.2 Módulo CAD

Este módulo, que utiliza a abordagem *Destruction by Machining Features*, citada no capítulo dois, pode ser dividido em três atividades principais:

1. Criação do modelo da peça bruta e acabada utilizando classes de *features* disponíveis na biblioteca de *features*.
2. Validação do modelo da peça criada.
3. Modelador gráfico no lado do cliente remoto.

A figura 4.3 mostra o diagrama IDEF0 da atividade CAD por *features* com as três atividades mencionadas. Como para a criação do modelo da peça é fundamental saber quais são as *features*, esse dado está na entrada da primeira atividade, e o usuário pode

criar as *features* utilizando a biblioteca de *features* Como resultado desta atividade, tem-se o modelo em *features* da peça que depois terá de ser validado, para finalmente ser apresentada na tela do computador do cliente remoto. Os outros dados como nome do projeto, nome da peça, etc, servem para salvar as peças modeladas e as ordens de fabricação de maneira organizada no banco de dados do servidor.

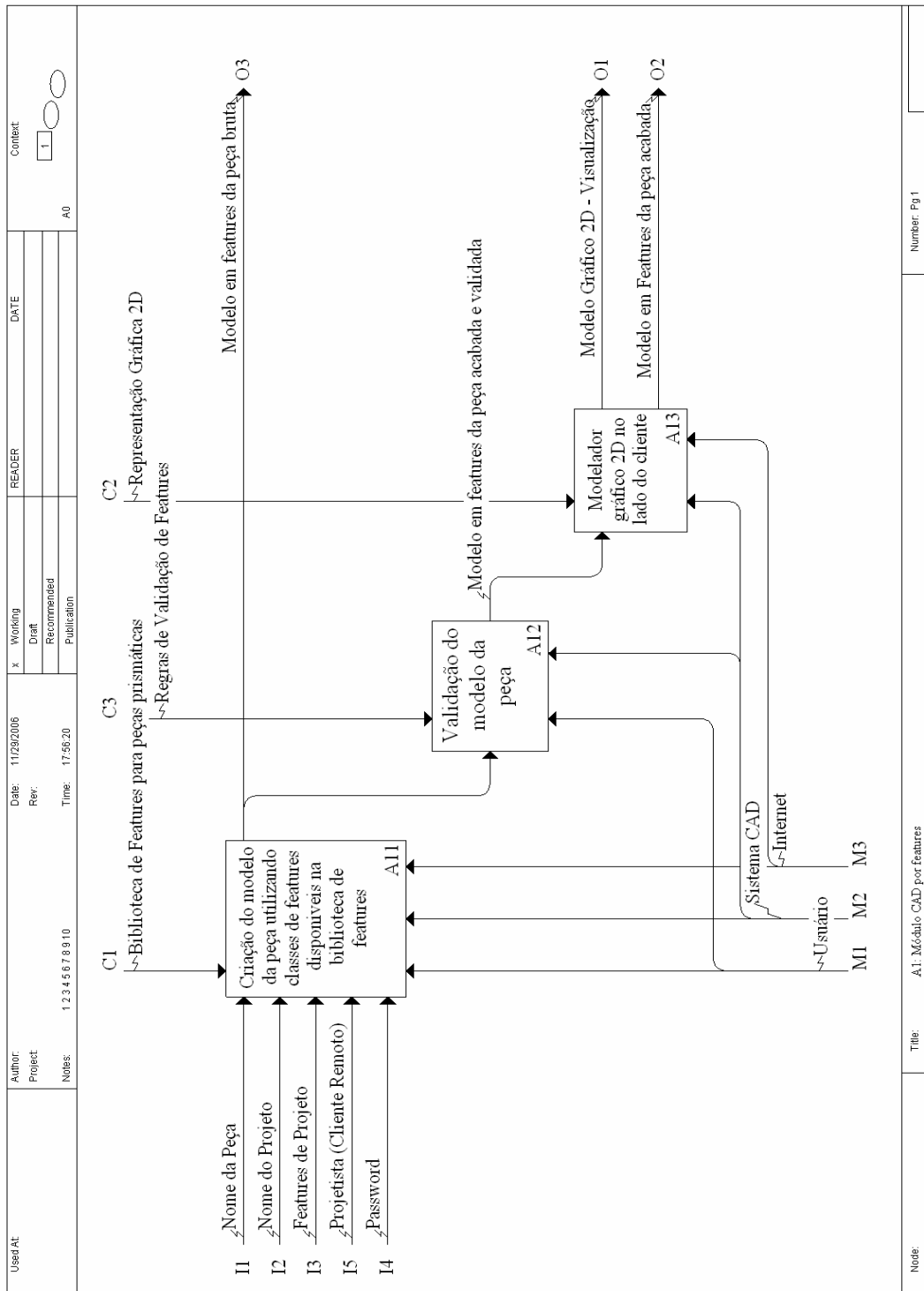


Figura 4.3 Diagrama IDEF0 da atividade CAD

A interface gráfica do usuário (GUI) disponibiliza as funcionalidades necessárias para suportar a modelagem da peça baseada em *features*. Isto é, possui menus, botões para selecionar a face a ser visualizada, bem como para mudanças na origem de referência, visando auxiliar o cliente remoto na criação das *features*. A GUI também contém a apresentação das *features* criadas em uma estrutura de árvore. Cada uma destas atividades serão detalhadas a seguir.

4.2.1 Decomposição por faces

Para descrever de uma maneira organizada o desenvolvimento deste sistema CAD, se utiliza a abordagem de “decomposição por faces”. Isto quer dizer que para criar uma *feature*, primeiro deve-se especificar a face na qual será feita a mesma, e logo depois especificar os parâmetros da *feature*.

Para ilustrar esta abordagem, na figura 4.4 é apresentada uma peça prismática com cinco *features* feitas em diferentes faces do bloco (furo1, furo2, cavidade1, ranhura1 e degrau1)

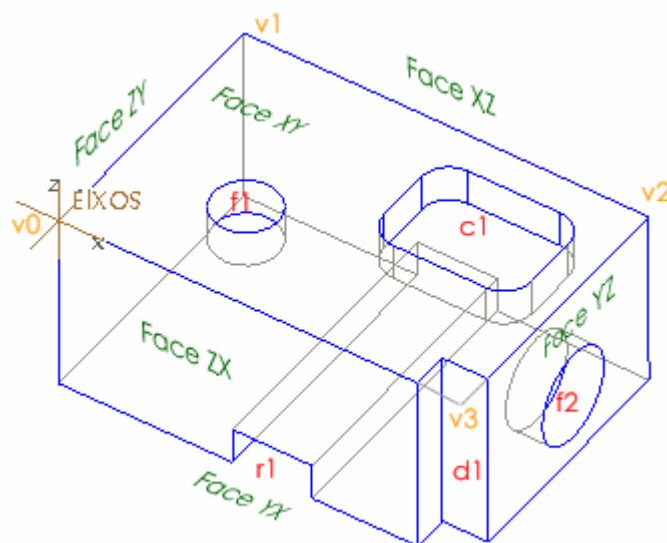


Figura 4.4 Decomposição por faces.

Aplicando-se a decomposição por faces, ter-se-á o seguinte:

Face XY → furo1, cavidade1.

Face YZ → furo2.

Face XZ \rightarrow Não há *features* (nulo).

Face YX \rightarrow ranhura1.

Face ZY \rightarrow Não há *features* (nulo).

Face ZX \rightarrow degrau1.

Deve-se notar que o degrau é uma *feature* que altera a geometria de duas faces (ver figura 4.5), porém, o sistema associa esta *feature* a apenas uma face (a face na que o projetista instanciou o degrau).

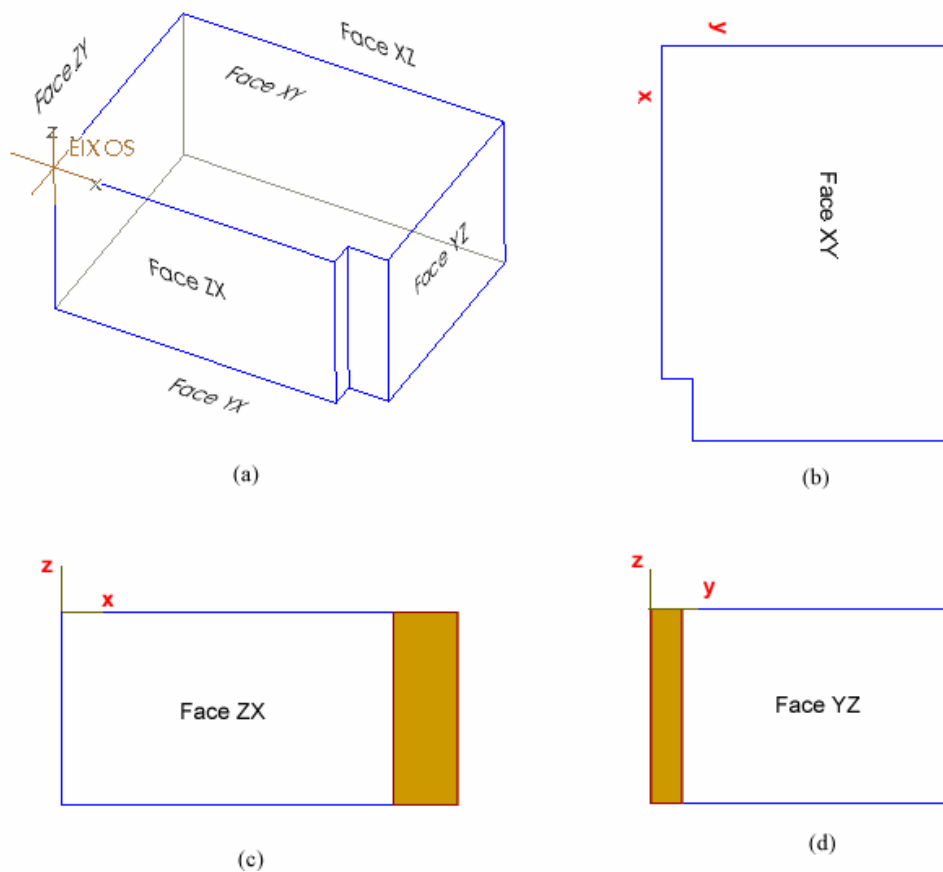


Figura 4.5 (a) Bloco com degrau em perspectiva, (b) Vista da face XY, (c) Vista da face ZX, (d) Vista da face YZ.

Os principais atributos da classe Face são apresentados em um diagrama UML⁶ (figura 4.6b). A cada face é atribuído um valor inteiro, assim, a face XY = 0, face YZ = 1, face XZ = 2, face YX = 3, face ZY = 4 e a face ZX = 5.

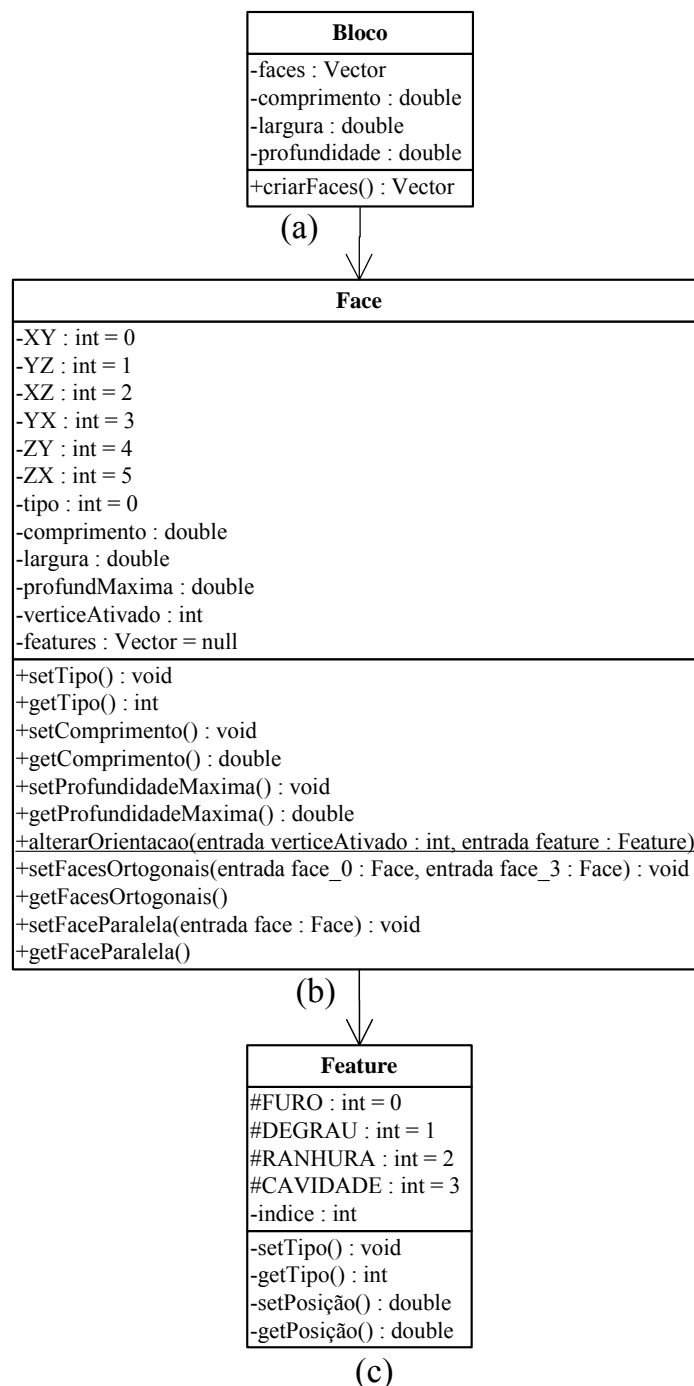


Figura 4.6 Diagramas UML das classes (a) Bloco, (b) Face, (c) *Feature*.

⁶ UML (*Unified Modeling Language*, <http://www.uml.org>) é uma ferramenta que ajuda a visualizar, especificar e documentar modelos de sistemas de software, incluindo suas estruturas e projetos, de uma forma que se possam conhecer todos os requerimentos do mesmo. UML também pode ser utilizado para modelar outros sistemas que não sejam software.

4.2.2 Criação do modelo da peça

A criação de uma peça bruta requer a instanciação das faces que comporão a mesma. Isto significa que haverá um bloco (instanciado da classe Bloco) que tem um vetor que contém as seis faces (vetor faces da figura 4.6 (a)). A interface gráfica é apresentada na figura 4.7.

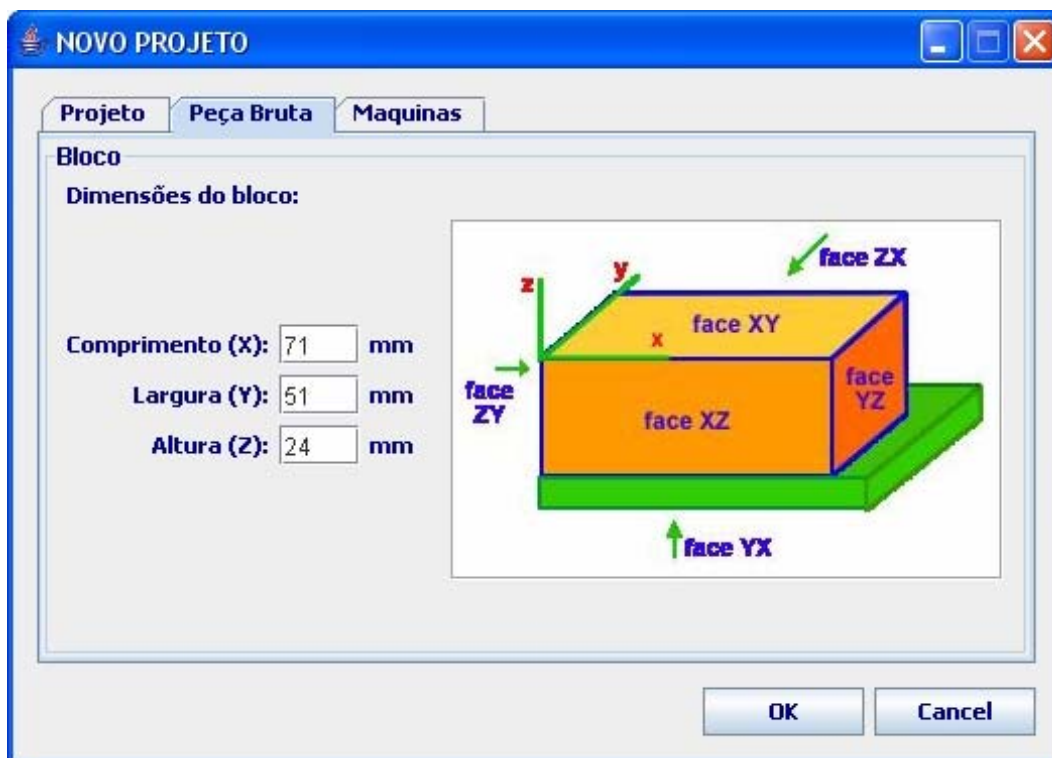


Figura 4.7 Instanciação de uma nova peça bruta.

No momento da criação de um bloco novo, são atribuídas a cada face suas dimensões (comprimento, largura e profundidade máxima). As faces têm um atributo chamado “vértice ativado” (um número inteiro), que informa ao programa em qual orientação está a face, quais são as quatro faces ortogonais a ela, e qual é a face paralela a ela e suas orientações respectivas relativas ao zero peça. Como exemplo, na figura 4.8 pode-se ver um bloco o qual está na mesa da máquina, a sua face de trabalho é a face XY. Como o vértice zero (v_0) da face XY é um ponto que também pertence aos pontos do eixo Z da máquina, logo se diz que a face tem como vértice ativado o vértice zero.

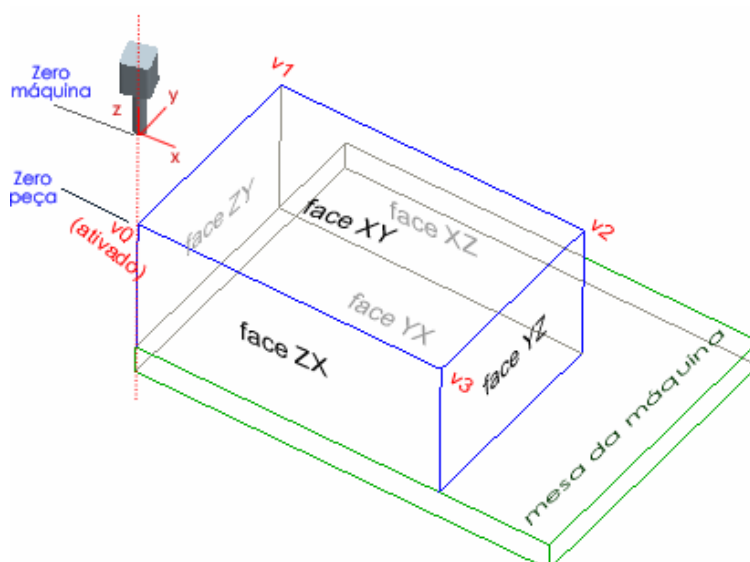


Figura 4.8 Bloco fixado na máquina com a face XY como face de trabalho.

Para este caso particular (face principal = XY e vértice ativado = 0), a sua face paralela será a face YX, e suas ortogonais serão: face ZY, face XZ, face YZ e a face ZX. No entanto, quando se faz um giro da peça de 90 graus no sentido anti-horário (em torno do eixo Z) e logo uma translação na direção do eixo +X, de modo que o vértice 1 (v1) seja um dos pontos do eixo Z da máquina, então se diz que o vértice 1 está ativado como ilustrado na figura 4.9.

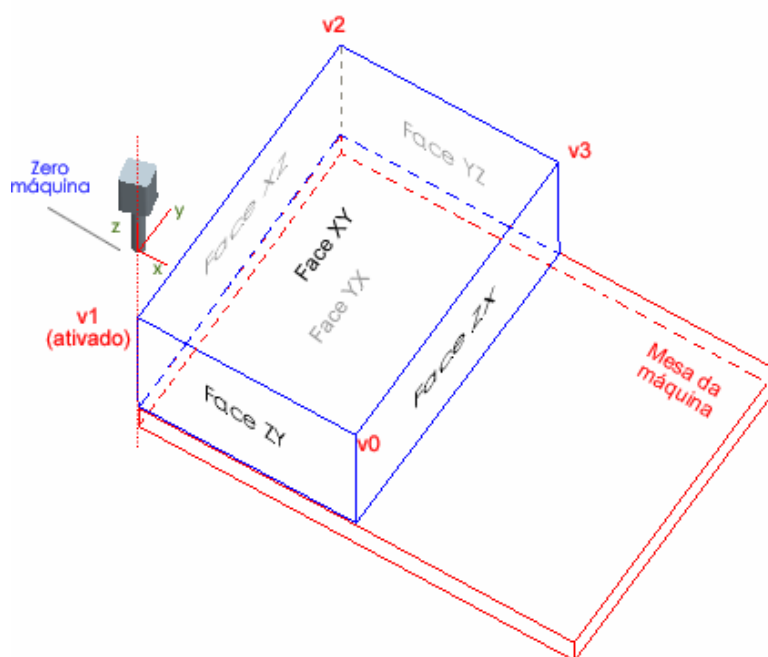


Figura 4.9 Bloco fixado com o vértice 1 ativado (girado 90 graus).

Estes recursos do módulo de CAD são ferramentas para auxiliar o projetista a alterar os pontos de referência das *features* a serem criadas, pois assim o projetista poderá escolher o vértice que melhor se acomode à sua necessidade. Por exemplo, na figura 4.10 o projetista colocou as referências do furo a partir do canto inferior direito (figura 4.10a), e para criar esse furo ter-se-ia que calcular as cotas a e b (figura 4.10b). Mas se a peça for girada 90 graus sentido anti-horário, então $a_1 = b_0$ e $b_1 = a_0$, e deste modo as referências originais do projeto são utilizadas na criação do furo.

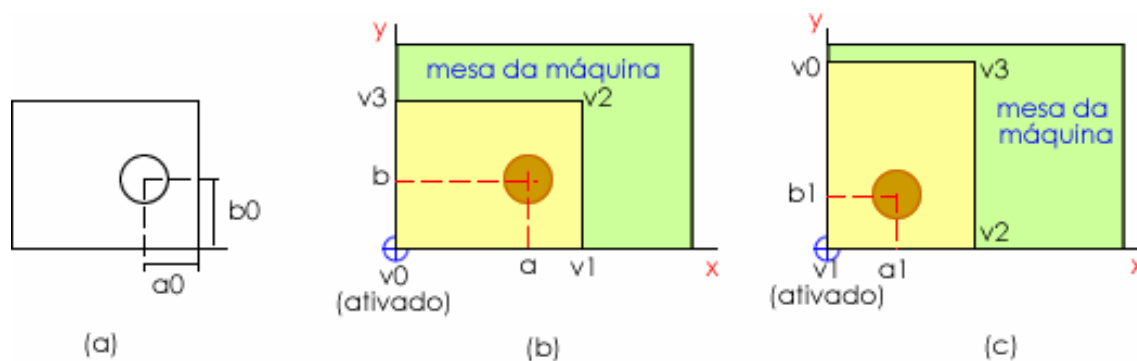


Figura 4.10 Modificação das referências de projeto.

4.2.2.1 A biblioteca de *features*

Na implementação do módulo CAD, as *features* consideradas na biblioteca são: furo, degrau, ranhura e cavidade retangular (com raios de concordância iguais) (ver figura 4.11).

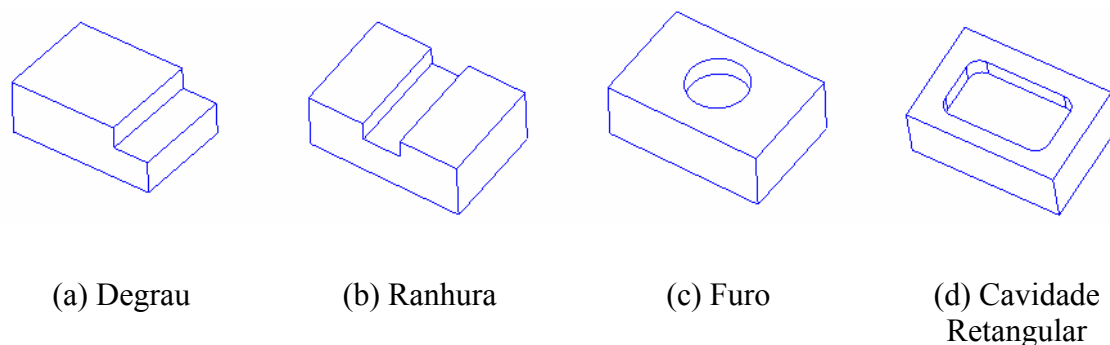


Figura 4.11 *Features* consideradas na implementação da “biblioteca de *features*”.

A abordagem para a implementação da biblioteca de *features* foi feita considerando-se que todas as *features* possuem atributos em comum, isto é, todas as *features* possuem um ponto representativo no espaço (x, y, z) , a partir do qual se pode criar ou referenciar

uma nova *feature*. Por exemplo, para um furo, o ponto representativo ou de referência será o seu centro geométrico. A classe “mãe” da biblioteca é a classe *Feature*. Assim, a classe *Feature* tem variáveis inteiras estáticas para poder definir o tipo de *feature* instanciada, por exemplo, um Furo é uma *Feature*, mas não é garantido que o contrário seja válido, isto é, uma ranhura, uma cavidade retangular ou um degrau também são *features*. Para isso utiliza-se o método `getTipo()`, como pode-se ver no diagrama UML da biblioteca de *features* das classes (figura 4.12). As classes Furo, Degrau, Ranhura e Cavidade derivam da classe *Feature*.

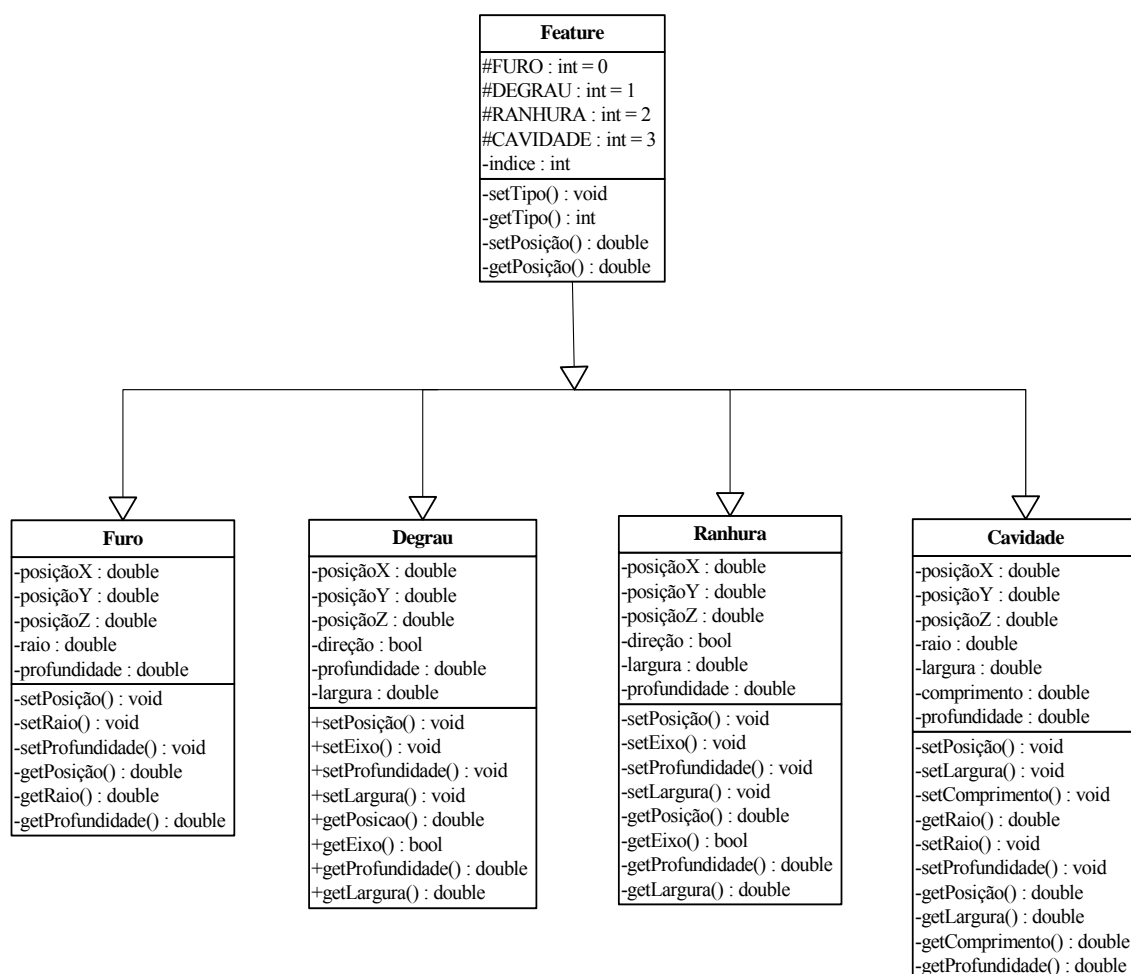


Figura 4.12 Diagrama UML da biblioteca de *features*

- **Feature Furo**: Os principais atributos desta classe são mostrados no diagrama UML da figura 4.12. Um novo furo deve ser definido em termos de sua posição no espaço (posição do centro geométrico do furo), o seu raio e a sua profundidade. Isto quer dizer

que em uma determinada face do bloco, existirá um e somente um furo com essas características, como ilustrado na figura 4.13, onde o ponto de referência da *feature* (chamado de ponto “p”) está localizado a X_0 , Y_0 , Z_0 unidades da origem.

Como mencionado anteriormente, para a instanciação de um novo furo, deve-se inicialmente especificar qual é a face na qual será inserido tal furo, e a orientação relativa à origem (isto é, vértice da face que está momentaneamente na origem).

A equação 4.1 apresenta os parâmetros para a criação de um novo furo.

$$\begin{aligned} \text{Furo furo} = & \text{ new Furo(double posX0, double posY0,} \\ & \text{double posZ0, double profundidade, double raio)} \end{aligned} \quad (4.1)$$

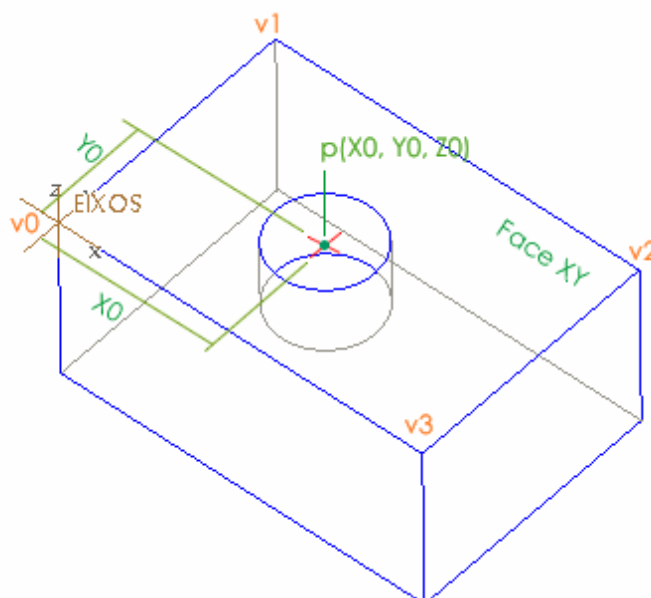


Figura 4.13 *Feature* Furo.

Os métodos que esta classe possui são utilizados para armazenar dados e devolvê-los quando for necessário. A figura 4.14 mostra a interface gráfica para a criação de um novo furo.

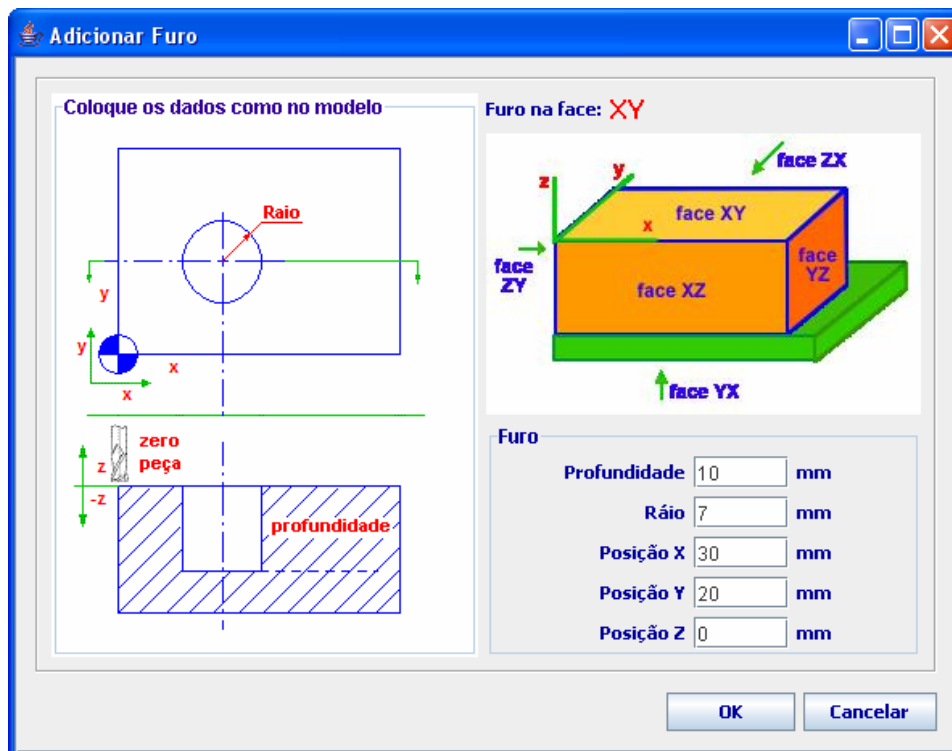


Figura 4.14 Interface gráfica para a instanciação de um novo furo.

- **Feature Degrau:** esta classe possui basicamente os mesmos atributos que a sua classe mãe (Classe *Feature*), com a diferença que esta classe tem uma variável que é útil para saber a orientação do degrau (vertical ou horizontal), a qual é a “direção”. Em uma face, existem somente quatro possibilidades de localização de um degrau, as quais são apresentadas na figura 4.15 onde se tem apresentam as quatro possíveis formas de criar um degrau em uma face.

A determinação das coordenadas do ponto “p” para cada caso é apresentada a seguir:

Caso a. Quando o degrau está à direita (figura 4.15a), as coordenadas do ponto “p” serão:

$$x = comprimento_BLOCO - largura \quad (4.2a)$$

$$y = 0 \quad (4.2b)$$

$$z = 0 \quad (4.2c)$$

A representação gráfica em 2D apresentada para o cliente remoto é um retângulo de área igual a: $largura * largura_BLOCO$.

Caso b. Quando o degrau está à esquerda (figura 4.15 b), as coordenadas do ponto “p” serão:

$$x = 0 \quad (4.3a)$$

$$y = 0 \quad (4.3b)$$

$$z = 0 \quad (4.3c)$$

Caso c. Quando o degrau está na parte inferior (figura 4.15 c), as coordenadas do ponto “p” serão as mesmas que no caso anterior, somente que a representação visual em 2D será um retângulo de dimensão $x = \text{comprimento_BLOCO}$ e $y = \text{largura}$.

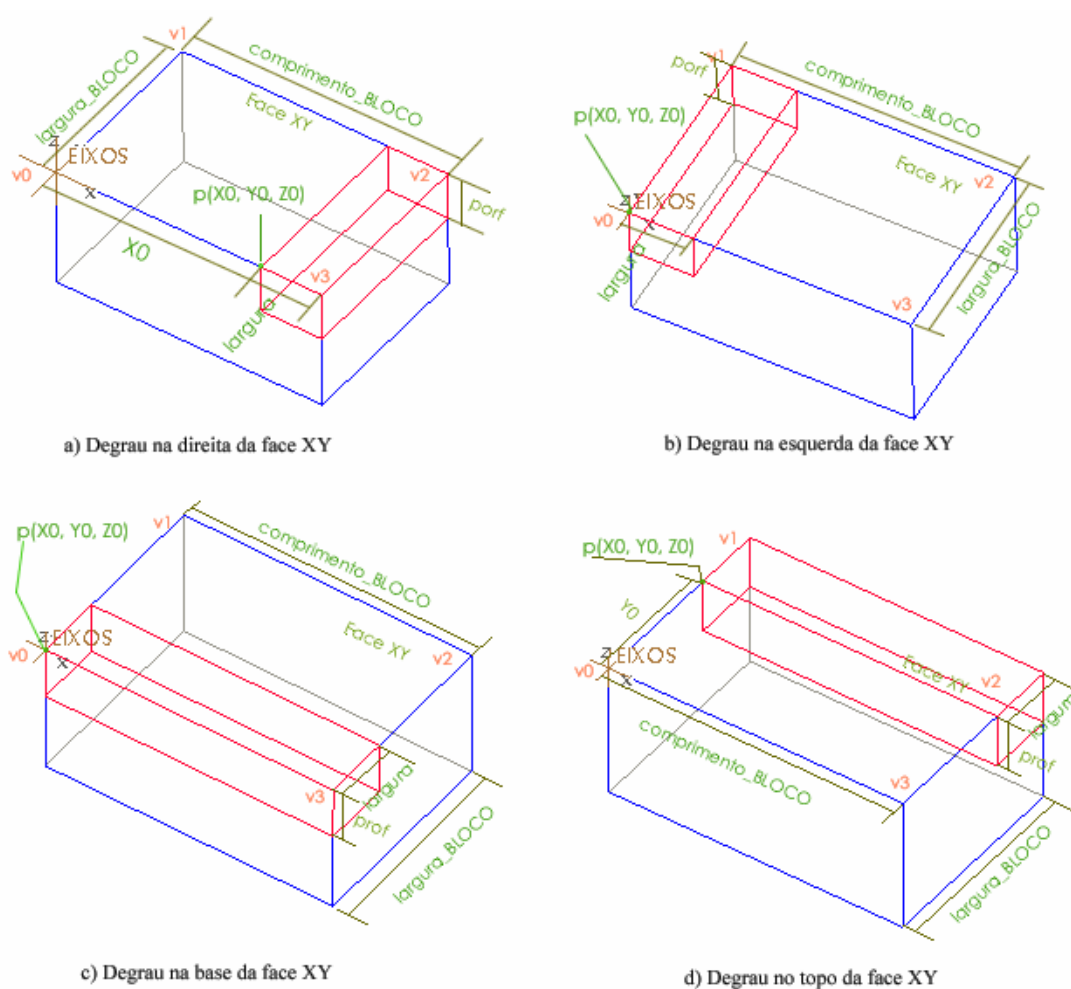


Figura 4.15 Possibilidades de criação de um novo degrau.

Caso d. Quando o degrau está na parte superior da face (figura 4.15 d), o ponto “p” terá as seguintes coordenadas:

$$x = 0 \quad (4.4a)$$

$$y = \text{largura_BLOCO} - \text{largura} \quad (4.4b)$$

$$z = 0 \quad (4.4c)$$

A representação visual será um retângulo de dimensões $x = \text{comprimento_BLOCO}$ e $y = \text{largura}$.

A figura 4.16 apresenta a interface gráfica para a instanciação de um novo degrau. Esta interface gráfica não solicita as coordenadas espaciais explicitamente como no caso do furo, e sim a posição Z, pois neste caso as outras coordenadas foram deduzidas das informações de orientação como mostrado anteriormente.

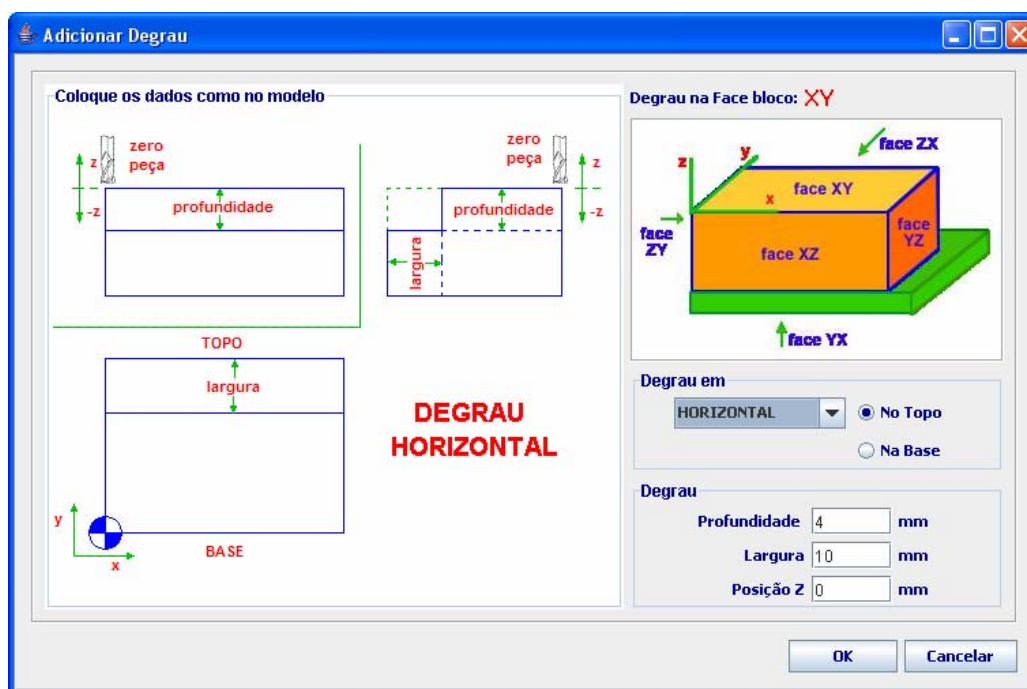


Figura 4.16 Interface gráfica para a criação de um novo degrau.

- **Feature Ranhura:** Esta classe é muito parecida com a anterior, pois um degrau poderia ser considerado como sendo uma ranhura posicionada em um canto da peça. No caso da *feature* Ranhura, ao comparar-se com a *feature* Degrau, deve-se introduzir os

seguintes dados: o deslocamento e a especificação se é vertical ou horizontal. A figura 4.17 apresenta a interface gráfica para a criação de uma nova ranhura.

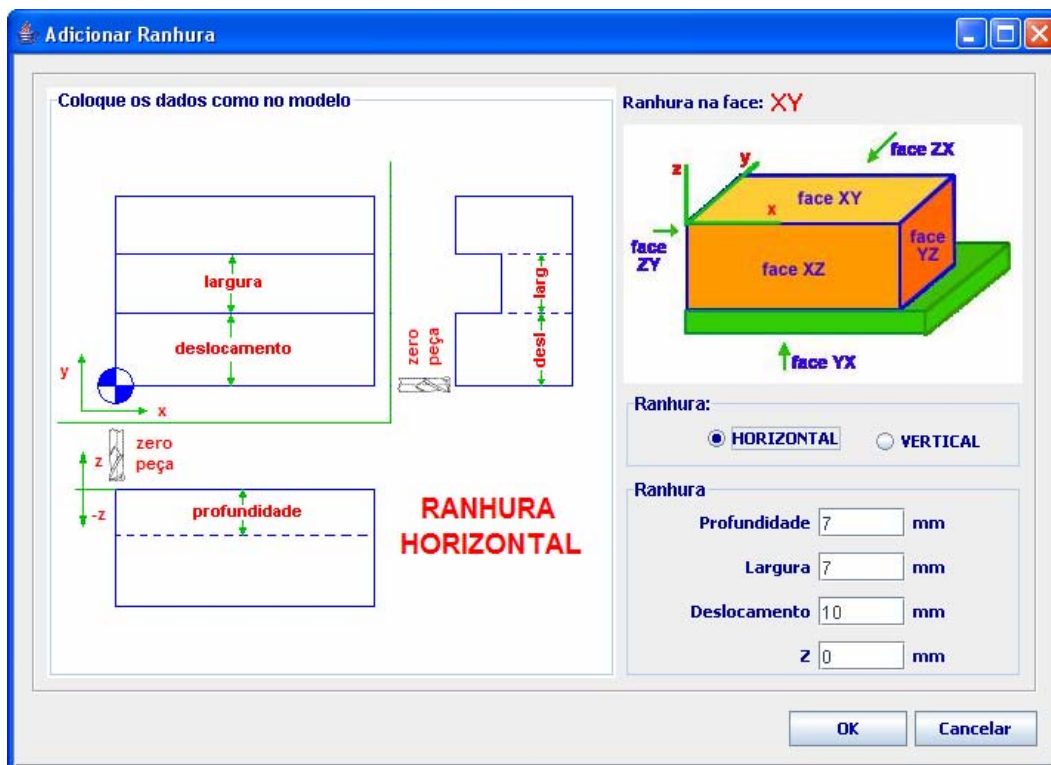


Figura 4.17 Interface gráfica para a criação de uma nova ranhura.

- **Feature Cavidade Retangular:** o ponto “p” é obtido diretamente da interface gráfica, e a partir deste ponto se desenhará a cavidade retangular na tela do computador.

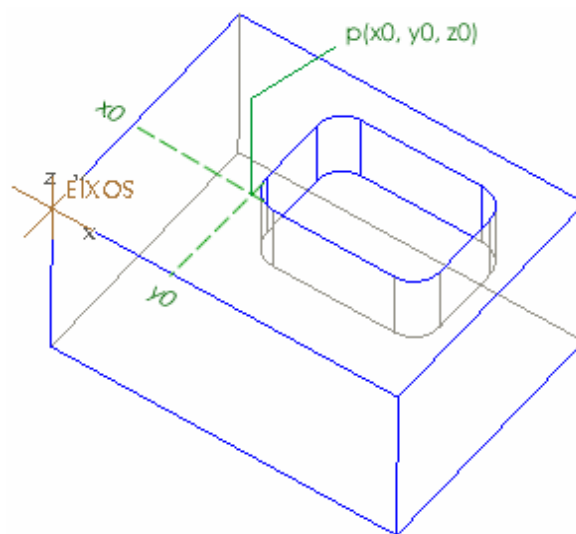


Figura 4.18 *Feature* cavidade retangular.

A interface gráfica é apresentada na figura 4.19, e neste caso todos os dados necessários para a criação da cavidade retangular são solicitadas explicitamente ao cliente remoto.

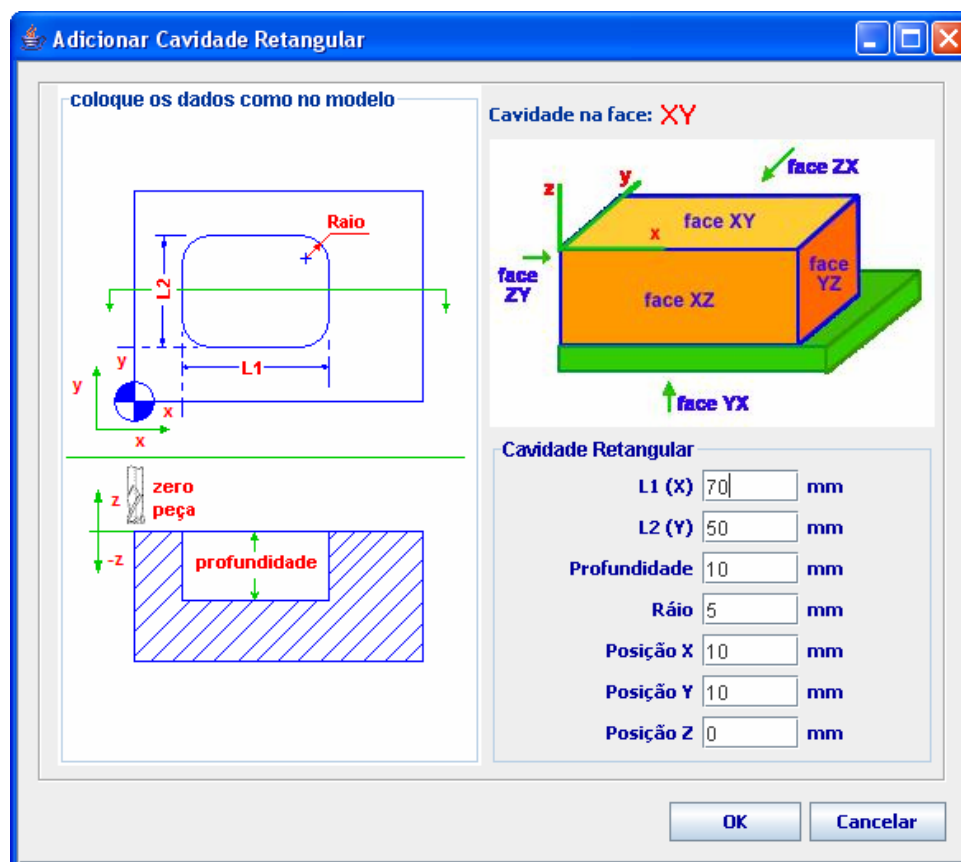


Figura 4.19 Interface gráfica para criar uma nova cavidade retangular.

4.2.3 Validação do modelo da peça

É possível que uma *feature* introduzida pelo usuário cause alguma inconsistência na peça criada, e por isso é necessária a validação da *feature*. Para isso, o módulo de CAD contém um conjunto de regras que auxiliam o projetista na tarefa da criação de uma peça consistente. Algumas destas regras são:

- uma *feature* é válida se ela está contida dentro dos limites da face de trabalho. Por exemplo, a cavidade retangular e o furo mostrados na figura 4.20 não são *features* válidas neste sistema, e nesses casos o sistema emitirá uma mensagem de erro, não permitindo a criação destas *features*.

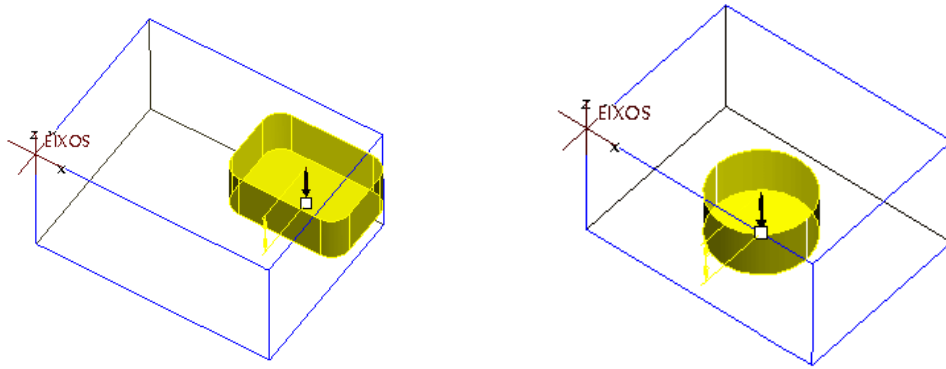


Figura 4.20 *Features* inválidas por estarem fora dos limites da face.

A figura 4.21 apresenta um caso em que o uma cavidade de dimensões $L1(X) = 60$ mm. e $L2(Y) = 50$ mm e com posicionamento $X = 10$ mm, $Y = 70$ mm e $Z = 0$ mm pretende-se inserir na face XY (de dimensões $X_{face} = 140$, $Y_{face} = 90$ e profundidade máxima = 60 mm). O programa detectou que há uma inconsistência nas dimensões relativas ao eixo “y” já que $Y + L2(Y) = 70 + 50 = 120$ mm, que comparando com a dimensão Y_{face} que é 90 mm indica que essa cavidade está fora dos limites da face, por tanto não pode ser criada, a mensagem emitida informa este erro ao projetista.

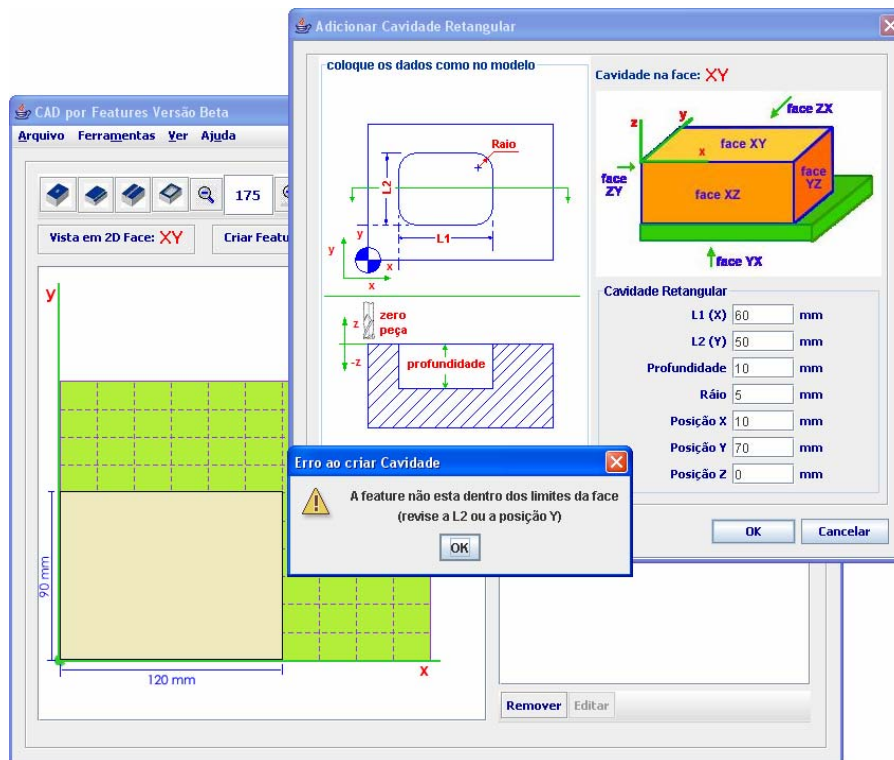


Figura 4.21. Mensagem de erro emitido pelo sistema CAD informando a inconsistência da cavidade que se pretende criar.

- para que uma *feature* seja considerada válida, a posição Z deve ser consistente, isto é, uma *feature* não pode ser criada se sua posição Z não coincidir com uma superfície da face da peça bruta (figura 4.22).

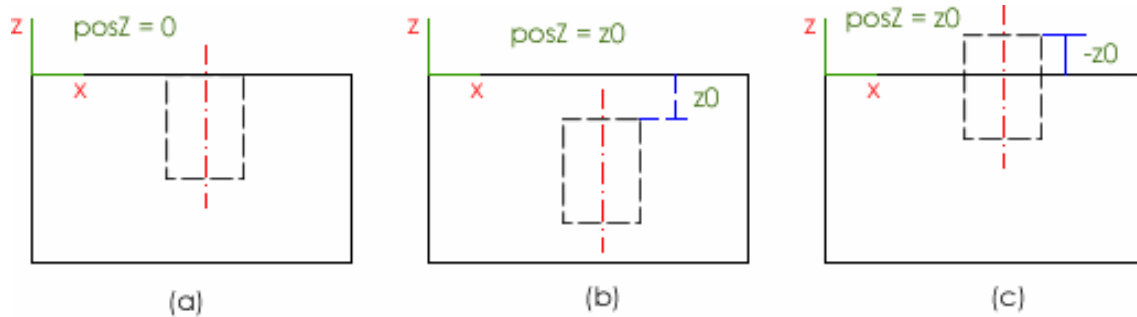


Figura 4.22 a) furo válido, b) e c) furos inválidos devido à posição Z.

Na figura 4.23, o projetista tenta fazer um furo com a posição Z de 5 mm dentro da superfície da face, o sistema reconhece esse erro e o informa indicando que a posição do furo deve coincidir com a superfície da face.

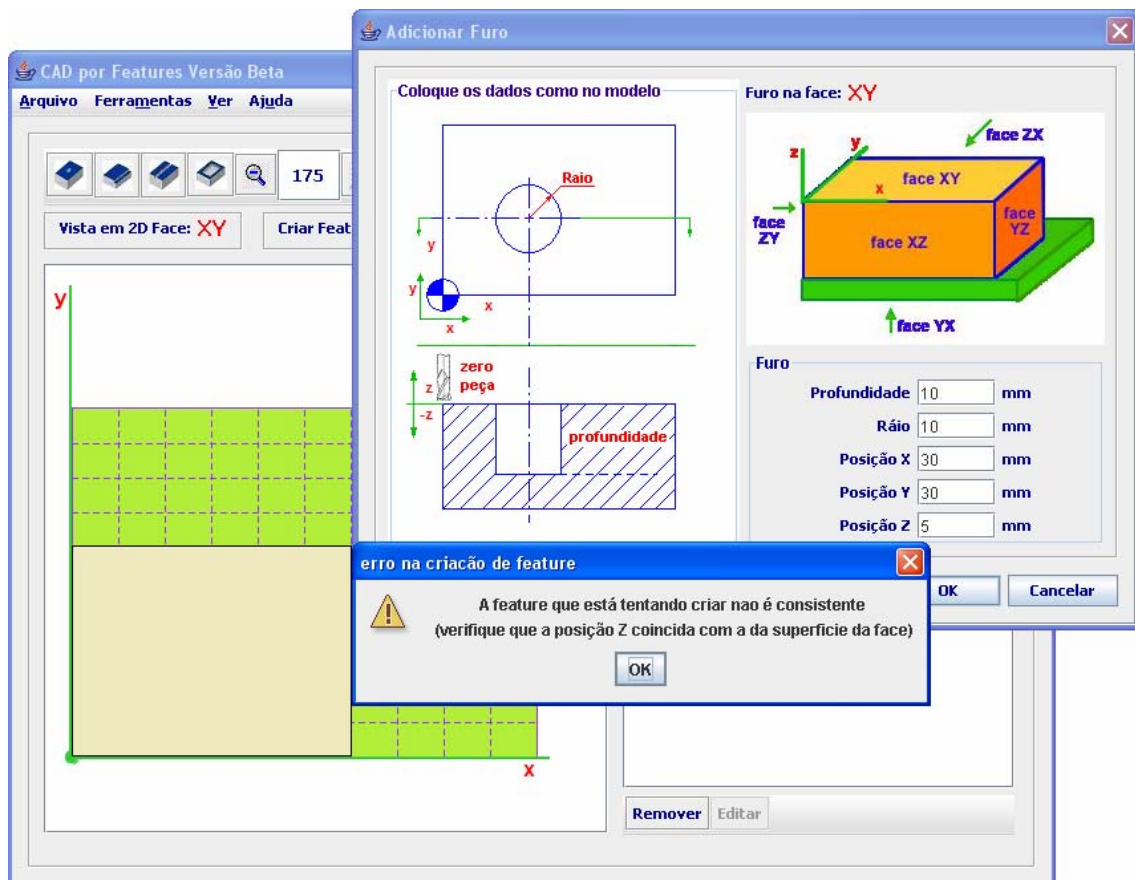


Figura 4.23. Mensagem de erro devido à inconsistência da posição Z da *feature*.

É permitida a criação de *features* aninhadas⁷ desde que a posição Z da *feature* aninhada coincida com a base da *feature* feita anteriormente. Na figura 4.24 se apresenta a mensagem de erro informando ao projetista que o furo que está tentando criar está aninhado dentro da cavidade, mas a posição Z ($z = 0$ mm) do furo não coincide com a profundidade da cavidade (20 mm).

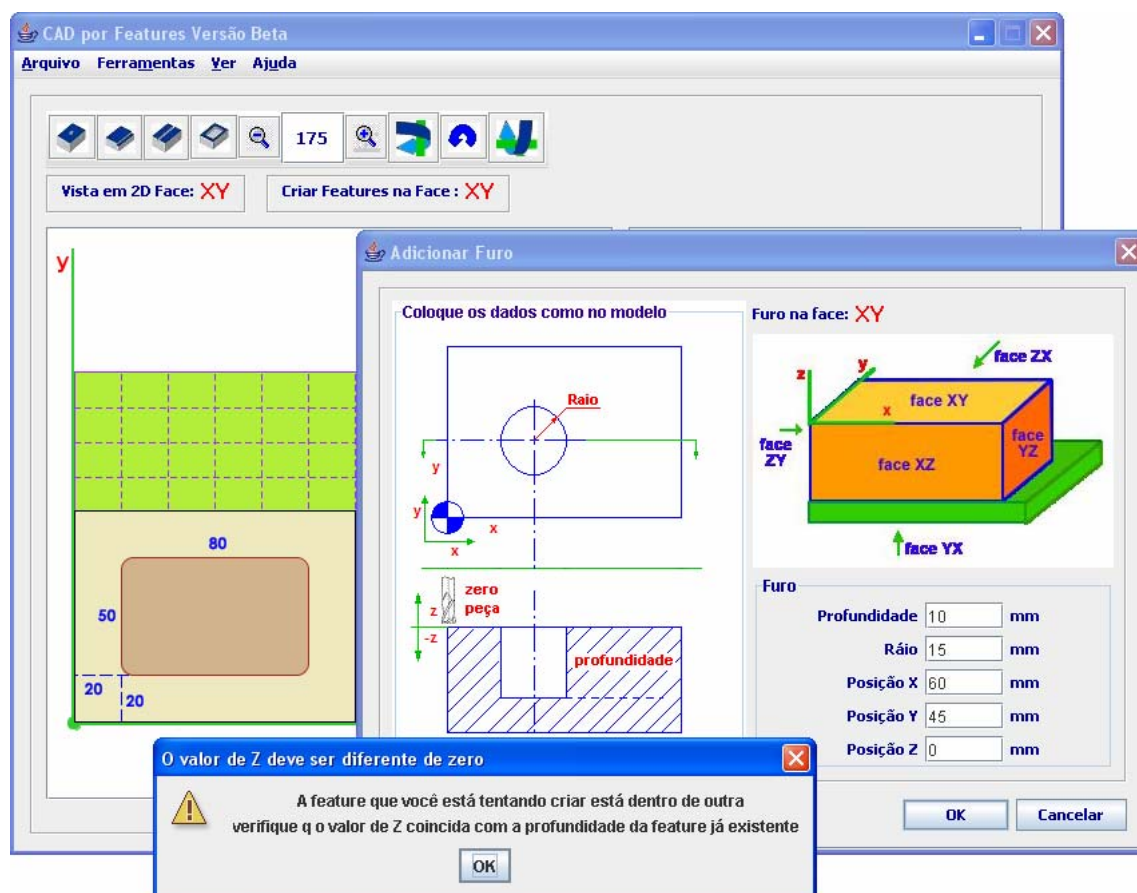


Figura 4.24. Mensagem de erro informando a inconsistência da *feature* devido à posição Z da *feature* aninhada.

Na figura 4.25 se apresenta o furo aninhado dentro da cavidade em vista lateral, note-se que a posição Z do furo agora coincide com a profundidade da cavidade antes inserida.

No caso de *features* que se intersectam, o sistema de CAD permite inserir uma *feature* intersectando outra desde que a posição Z da nova *feature* seja consistente. Por exemplo, na figura 4.26, a ranhura2 foi inserida depois da cavidade, o programa exige

⁷ *Features* Aninhadas, do inglês “nested features”, o qual aqui significa a criação de uma *feature* dentro de uma outra já existente.

que esta ranhura tenha a posição Z igual a zero, na figura 4.27 se apresenta a mesma peça em vista lateral.

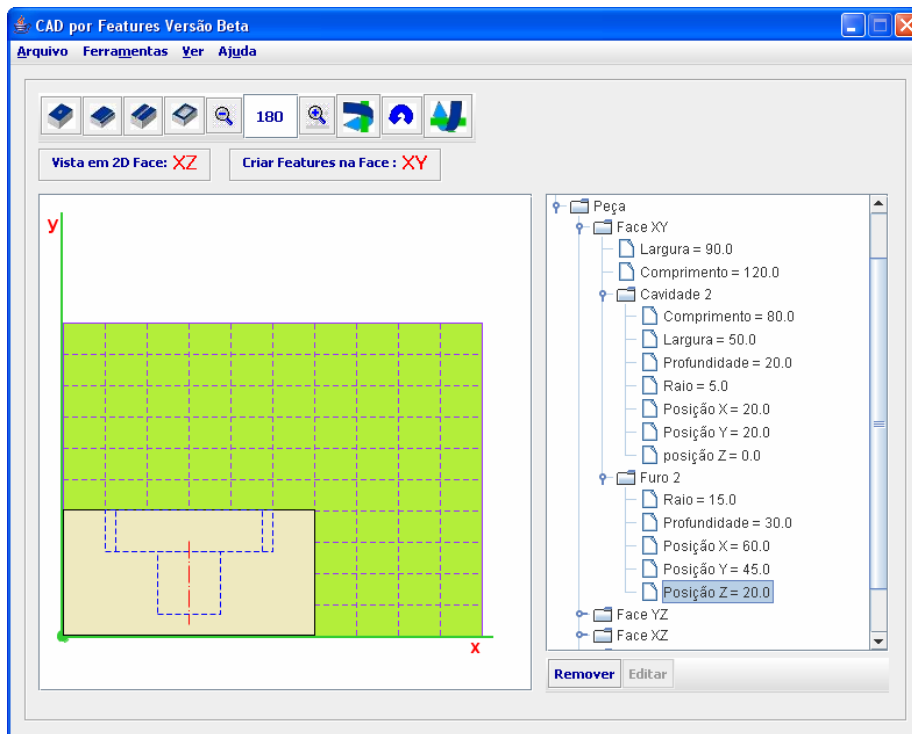


Figura 4.25. Furo aninhado dentro da cavidade.

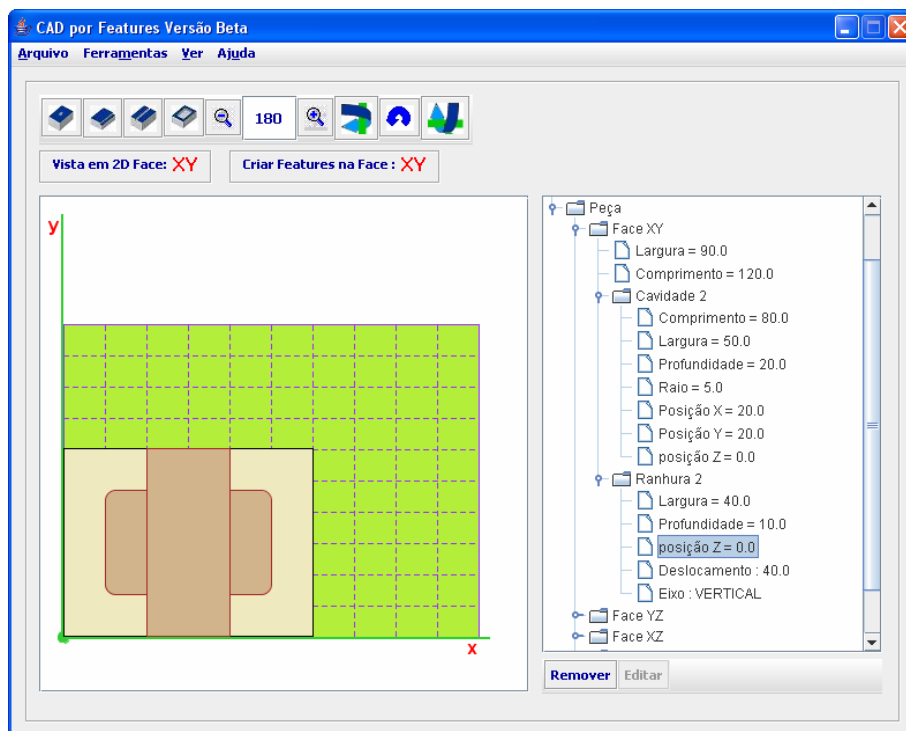


Figura 4.26. Interseção de uma cavidade e uma ranhura.

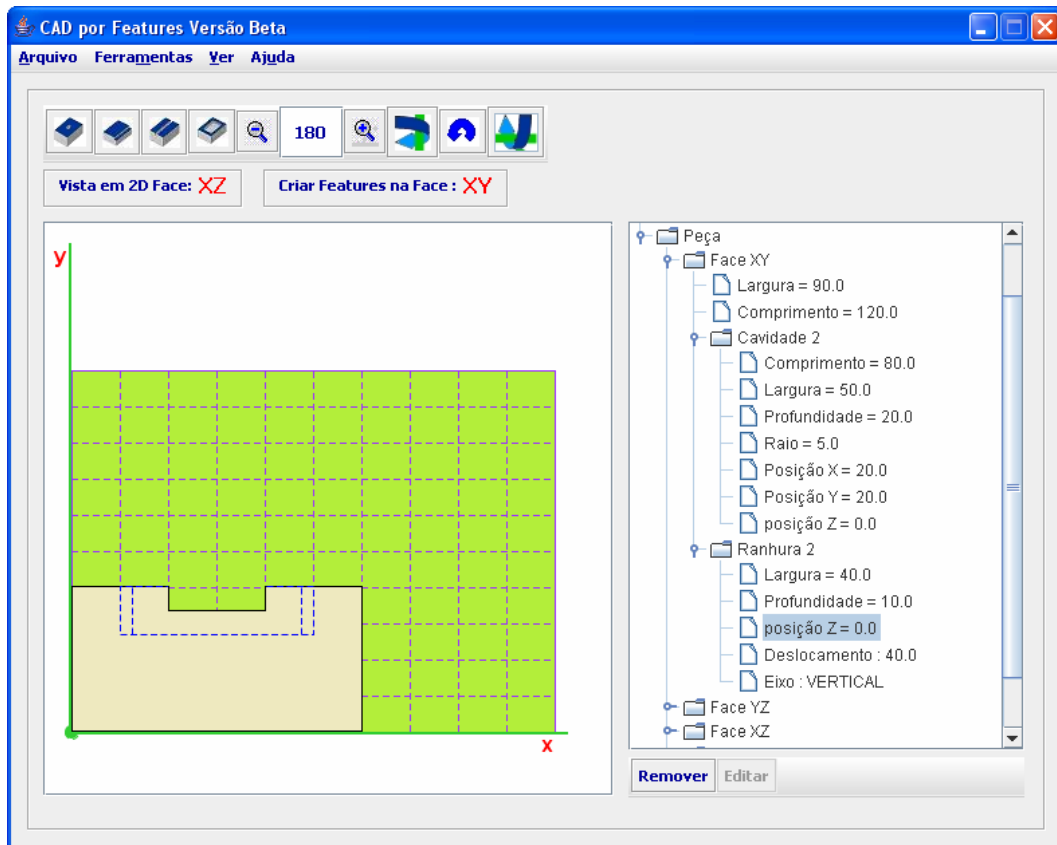


Figura 4.27. Interseção de uma cavidade e uma ranhura em vista lateral.

- uma outra regra de validação é que uma *feature* não pode ultrapassar a profundidade máxima da peça bruta, como mostrado na figura 4.28.

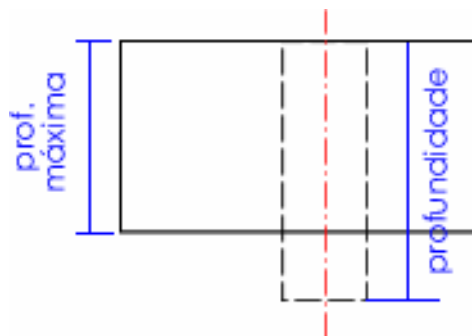


Figura 4.28. Furo inválido devido à sua profundidade ser maior do que a profundidade máxima da peça bruta.

A figura 4.29 apresenta um caso em que o projetista tenta fazer um furo com profundidade de 70 mm na face XY que tem uma profundidade máxima de 60 mm.

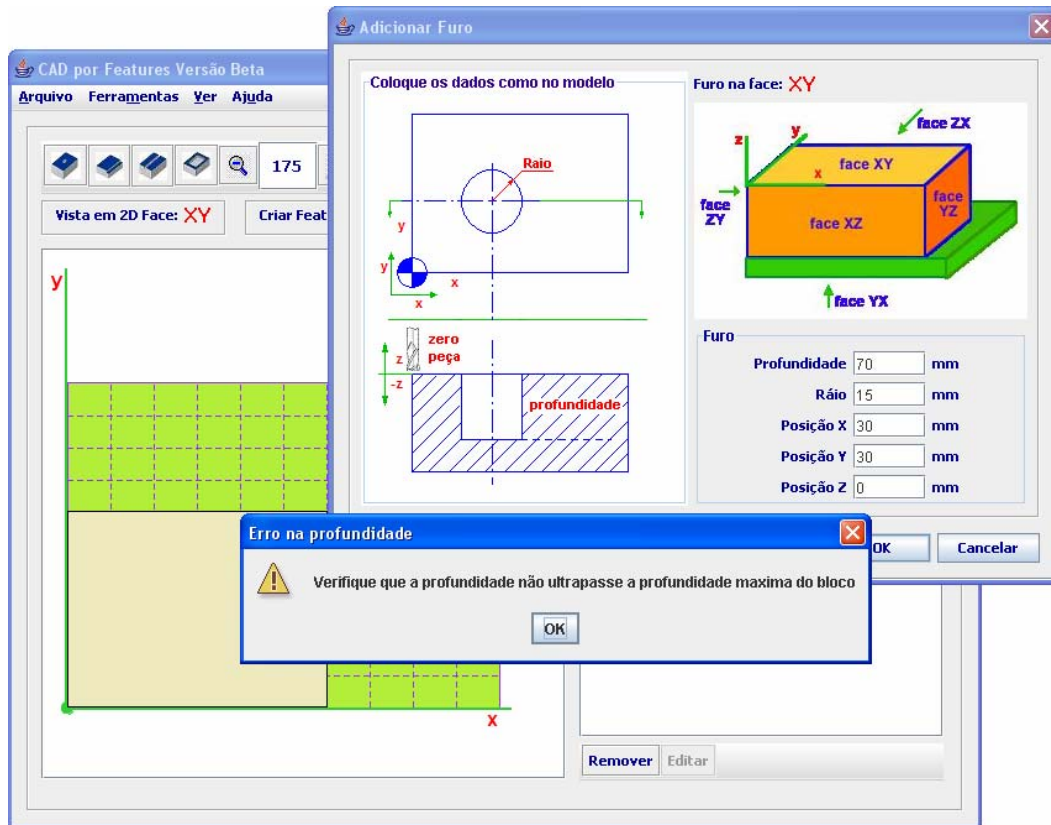


Figura 4.29. Mensagem de erro devido a inconsistência na profundidade da *feature*.

- não pode ser criada uma *feature* que não possa ser usinada com a ferramenta disponível. Por exemplo, uma ranhura não pode ter largura menor do que o diâmetro da ferramenta, e não pode ter profundidade maior do que a ferramenta possa penetrar (figura 4.30).

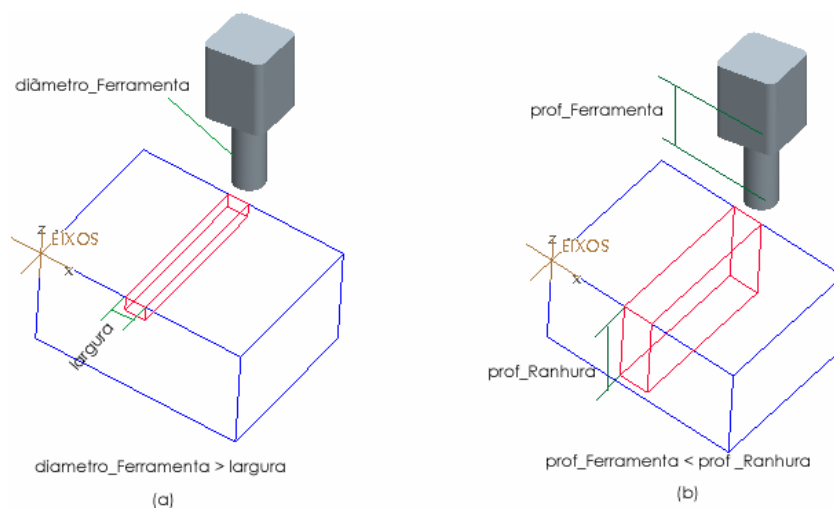


Figura 4.30. Inconsistência de uma *feature* ao ser comparada com a geometria da ferramenta.

A figura 4.31 apresenta a mensagem de erro emitida pelo programa ao tentar fazer uma ranhura com largura (2 mm) menor do que o da ferramenta (3 mm).

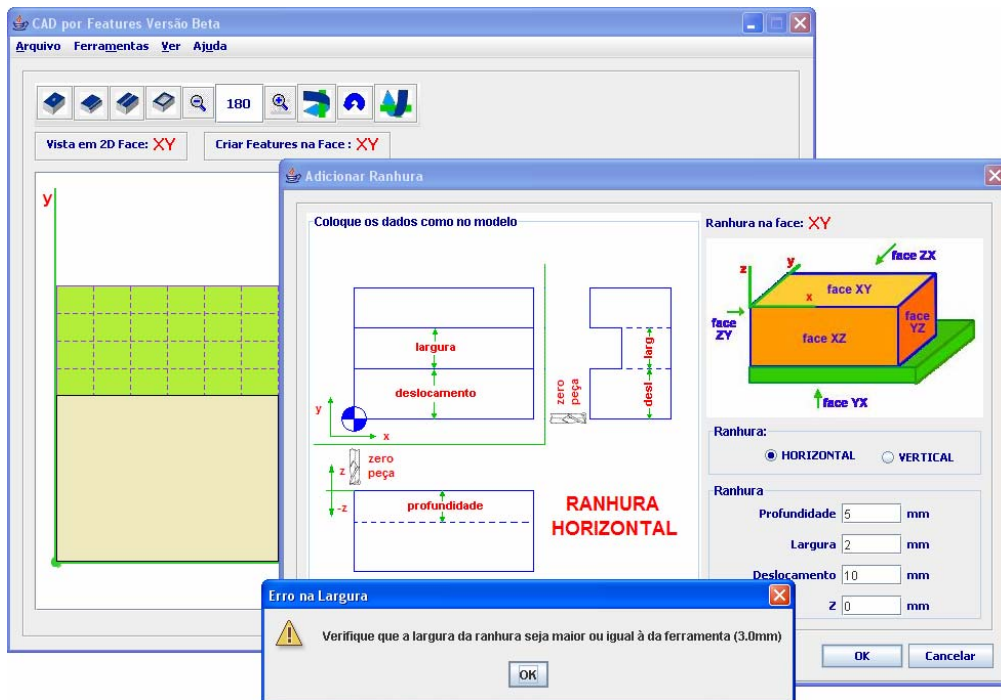


Figura 4.31. Mensagem de erro ao tentar fazer uma ranhura com largura menor do que o da ferramenta.

- não é possível criar uma *feature* em uma face que não possa ser fixada dentro da máquina de prototipagem rápida. Por exemplo, na figura 4.32 pode-se ver que é possível criar *features* na face XY, mas não na face YZ.

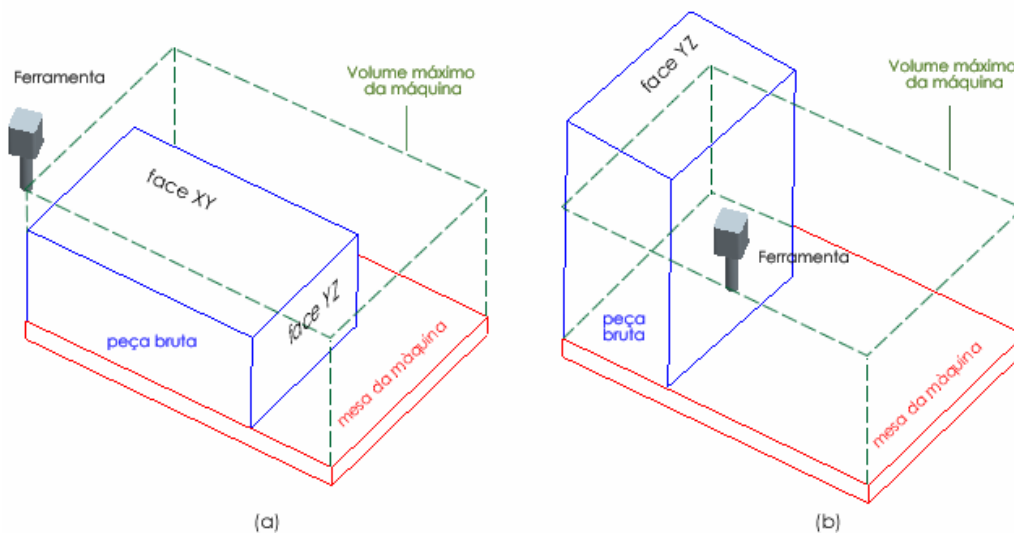


Figura 4.32. Inconsistência da *feature* por causa da máquina.

A figura 4.33 apresenta a mensagem emitida pelo programa quando o projetista tenta criar *features* nessas faces.

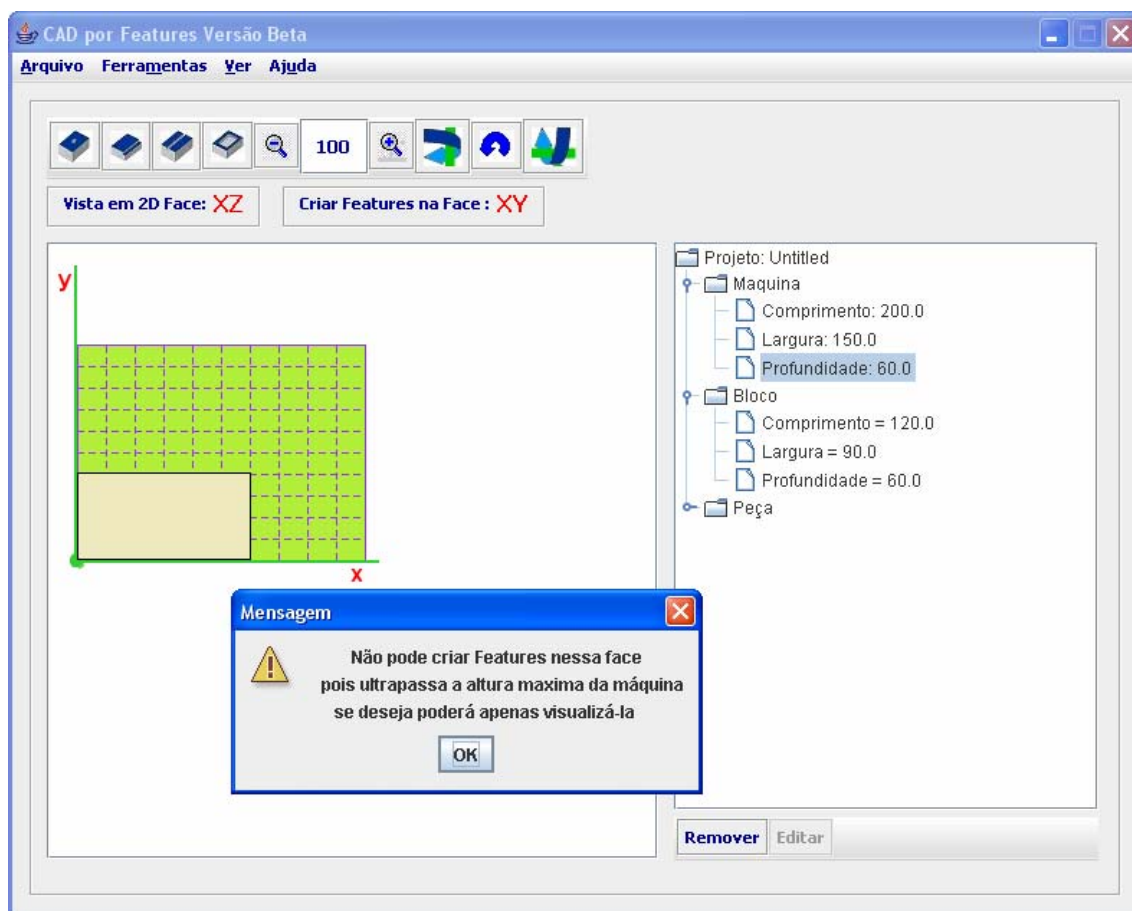


Figura 4.33. Mensagem de erro emitida quando tenta-se criar *features* em uma face que não pode ser fixada dentro da máquina.

4.2.4 Modelador gráfico

No lado do cliente remoto, uma “applet” na linguagem Java é executada, na qual inicialmente o sistema percorre os vetores de faces e os vetores de *features* para ver se em algum deles contém *features*. Então, a representação gráfica em 2D da peça é desenhada por faces na tela do computador do cliente remoto, podendo-se escolher a face que se deseja visualizar. Uma peça modelada neste aplicativo de CAD é apresentada na figura 4.34.

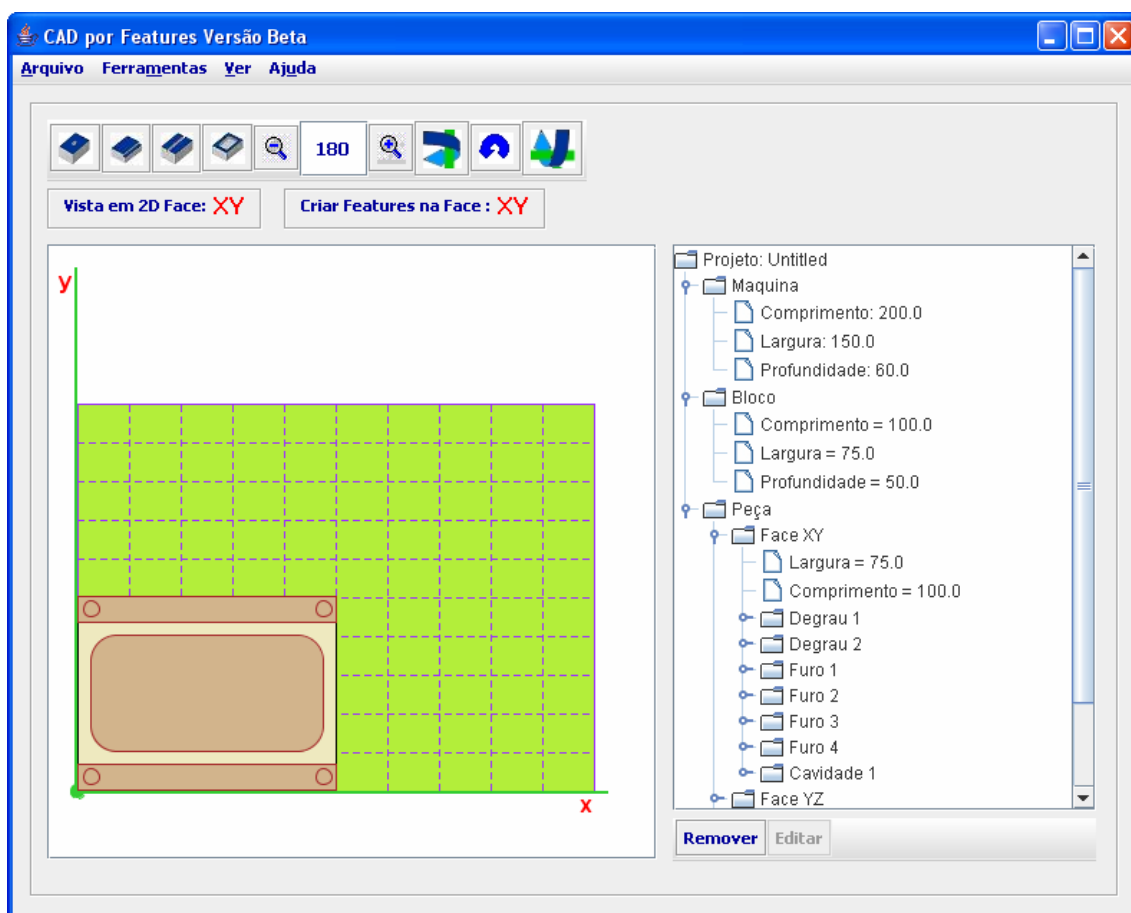


Figura 4.34. Uma peça prismática modelada no aplicativo CAD.

No lado esquerdo do aplicativo, é apresentada a peça modelada, e no lado direito aparecem dados da máquina e da peça bruta, assim como as *features* presentes em cada face (se existirem).

4.3 Módulo CAPP

O módulo CAPP deste sistema integrado executa as seguintes atividades, as quais serão detalhadas posteriormente:

1. Mapeamento de *features* de projeto para *workingsteps*⁸;
2. Determinação das estratégias de usinagem para cada tipo de *feature*;
3. Formação dos grupos de *workingsteps*;

⁸ Segundo STEP-Tools (www.steptools.com) *machining workingstep* descreve as seqüências de remoção de material, localização, e parâmetros associados.

4. Determinação das seqüências de usinagem.

Estas atividades são apresentadas no diagrama IDEF0 mostrado na figura 4.35.

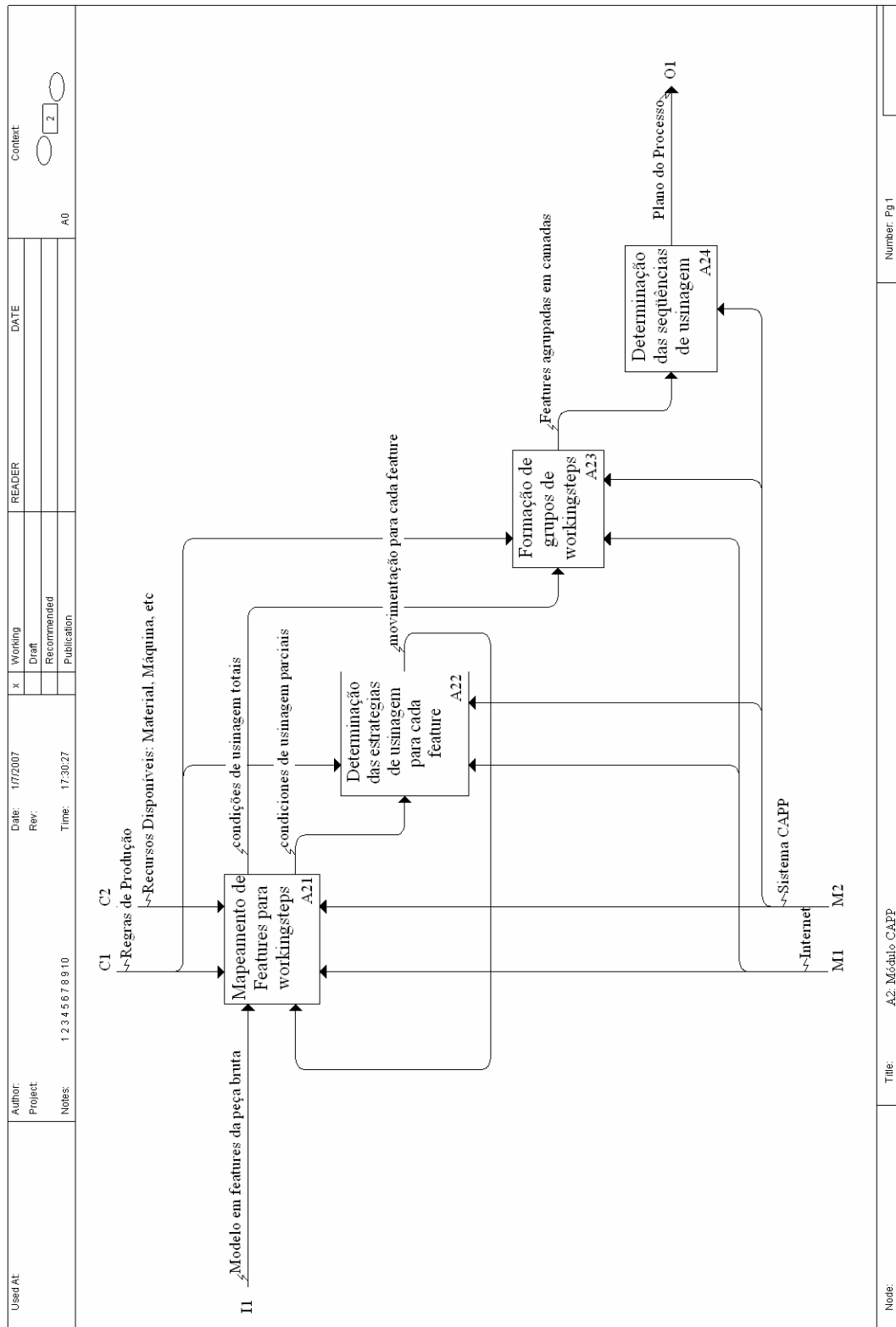


Figura 4.35. Diagrama IDEF0 das atividades do módulo CAPP.

Um dado de entrada importante para a geração do plano do processo da peça modelada é conhecer as informações técnicas da máquina disponível para a fabricação. A máquina é mostrada na figura 4.36, e as suas características técnicas encontram-se na tabela 4.1.



Figura 4.36 Máquina de prototipagem rápida por usinagem Roland MDX-20 utilizada neste trabalho

Tabela 4.1 Dados técnicos da Máquina Roland MDX-20

Tamanho da mesa	220 (X) x 160 (Y) mm
Área de trabalho	203 (X) x 152 (Y) x 60 (Z) mm
Carga máxima	1 kg
Interface	Serial (RS-232C)
Alimentação	Adaptador AC/DC (DC+19V 2,1 A).
Peso	13,7 kg
Motor de Usinagem	10W (motor DC)
Velocidade do fuso	6.500 rpm
Diâmetro da ferramenta	3 mm
Precisão mecânica	0,01 mm/passo
Velocidade de movimentação	0,1 a 15 mm/s
Materiais indicados	Madeira, Resinas, Plásticos e Metais Leves como o Zinco, Alumínio e Latão.

Esta máquina tem o seu próprio software para o projeto chamado Modela Player, o qual é de domínio privado.

Para o desenvolvimento deste sistema de CAPP, foram consideradas as seguintes restrições:

- Utilizou-se uma fresa de topo de diâmetro de 3 mm;
- Apenas uma fixação por face (a fixação da peça na mesa da ferramenta é feita com fita adesiva);
- A velocidade de avanço em cada um dos três eixos foi constante (igual a aproximadamente 10 mm/s),
- A velocidade de rotação da ferramenta foi constante (igual a 6500 rpm).

Assim, o módulo CAPP em sua atual versão busca gerar uma seqüência de usinagem das *features* que seja consistente, isto é, que resulte exatamente nas superfícies das *features* introduzidas pelo usuário remoto.

4.3.1 Mapeamento das *features* de projeto para *workingsteps*

A primeira atividade do módulo CAPP é o mapeamento das *features* para *workingsteps*. Na figura 4.37 se apresenta um diagrama em UML da classe *Workingstep*. Em essência, um *workingstep* contém além dos dados relativos à geometria da *features*, dados úteis em termos de manufatura como as condições de usinagem, o tipo de movimentação (radial, ou zig zag), vetor com os pontos de movimentação da ferramenta entre outros dados. Todos estes dados devem ser determinados pelo sistema. Mais adiante se detalha o cálculo dos mesmos.

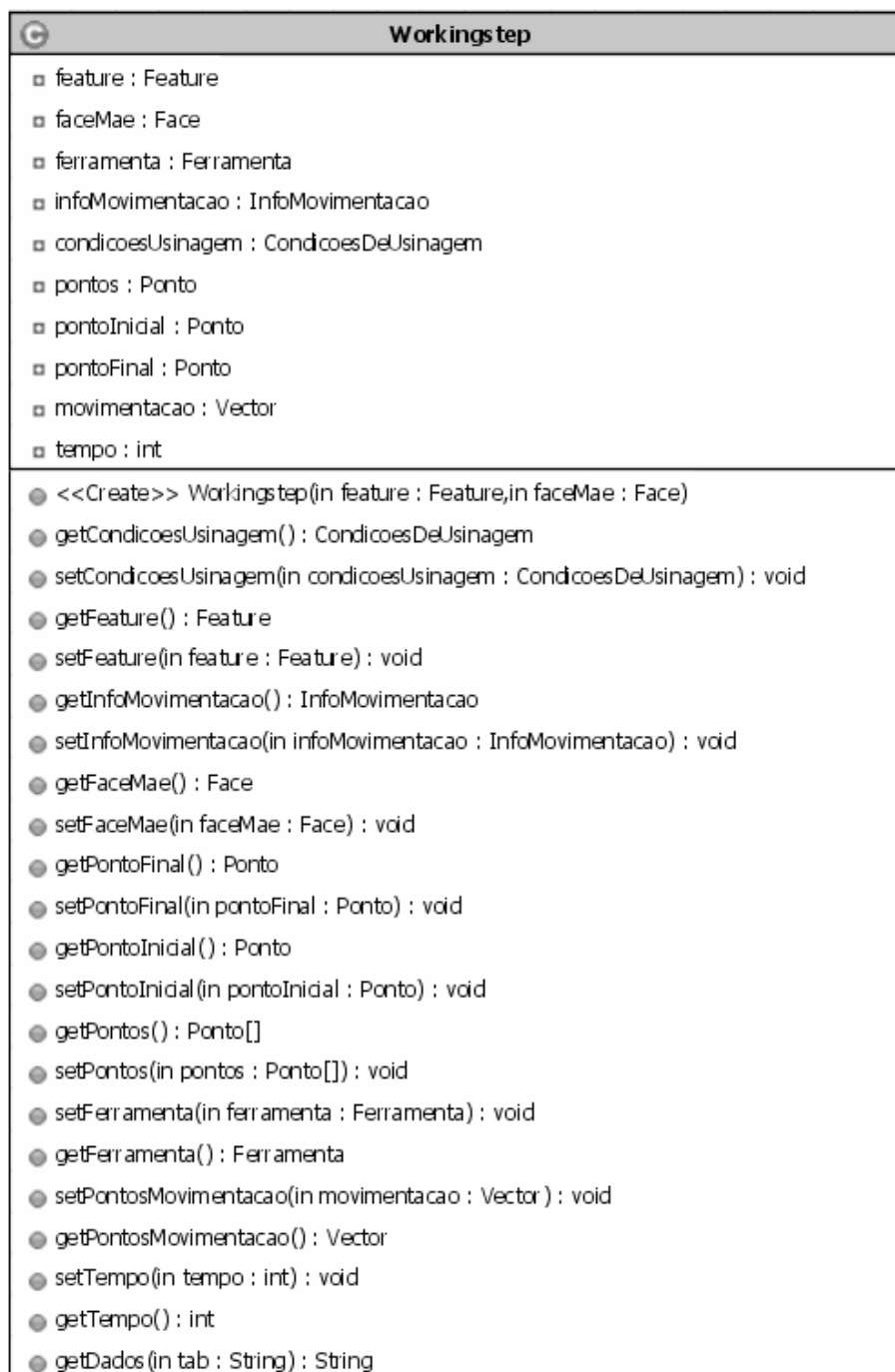


Figura 4.37. Diagrama UML da classe *Workingstep*.

A classe Mapeadora é encarregada de “transformar” as *features* em *workingsteps*, o diagrama em UML desta classe se apresenta na figura 4.38.

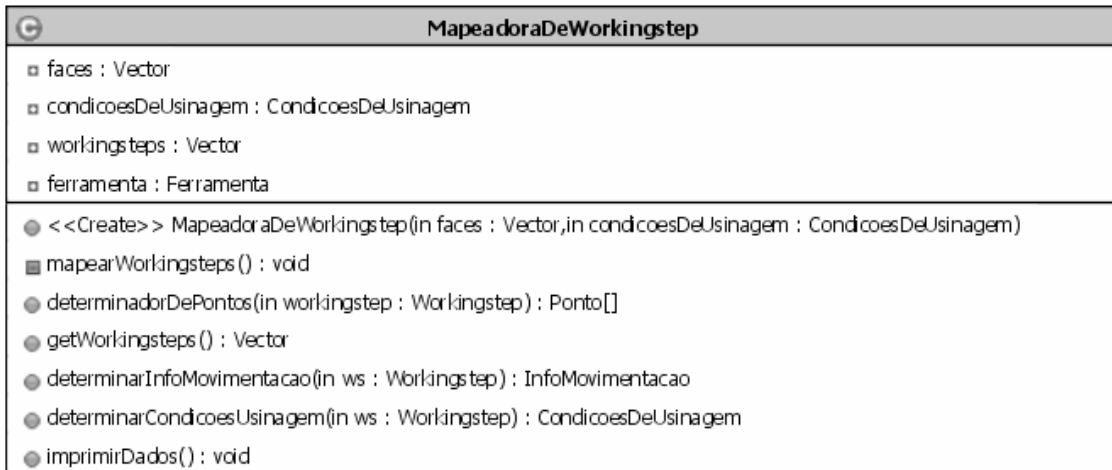


Figura 4.38. Diagrama UML da classe Mapeadora.

Esta classe recebe o vetor de faces do bloco, que até este ponto contem apenas *features*, e como resultado devolve um outro vetor contendo *workingsteps* do bloco inteiro como ilustrado na figura 4.39.

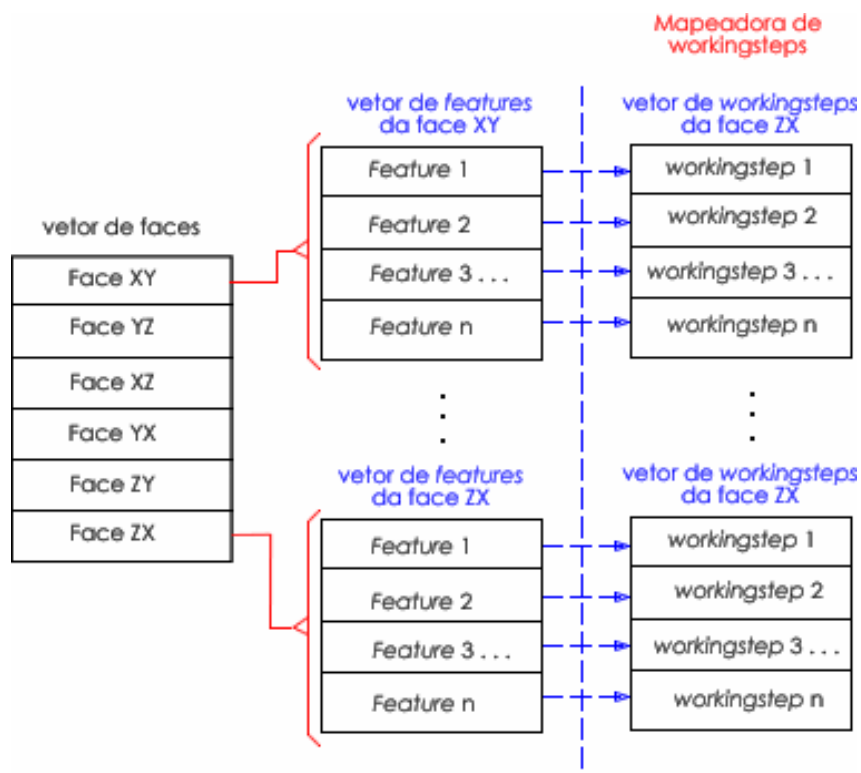


Figura 4.39. Mapeamento de *features* para *workingsteps*.

Segundo FERREIRA (1996), pode-se calcular as condições de usinagem para fresamento de topo utilizando-se a equação (4.5).

$$Potência da máquina [W] = Potência unitária^* \times Taxa de remoção [mm^3 / s] \quad (4.5)$$

Onde a potência unitária para diferentes materiais é mostrada na tabela 4.2.

Tabela 4.2 Potências unitárias para diferentes materiais para fresamento de topo (adaptado de FERREIRA, 1996).

<i>Material</i>	<i>Dureza (HB)</i>	<i>Potência unitária* baseada em ferramentas de aço rápido e metal duro</i>
Alumínio	30 a 50	0,82191
Bronze	50 a 145	1,64382
	145 a 240	2,7397
Ferro fundido	110 a 190	1,64382
	190 a 320	3,01367
Aço suave	85 a 200	3,01367
	330 a 370	4,10955
	485 a 560	5,75337
Aço inoxidável	135 a 257	3,83558
	275 a 430	4,10955
Plásticos	N/A	0,136985

Então, considerando-se que a máquina dispõe de uma potência útil no fuso igual a 10 W (obtido da tabela 4.1), e com auxílio da tabela 4.2, pode-se determinar a taxa de remoção máxima para materiais plásticos:

$$10 W = 0,136985 \times Taxa de remoção \left(\frac{mm^3}{s} \right) \quad (4.6)$$

$$Taxa de remoção \approx 73 \left(\frac{mm^3}{s} \right) \quad (4.7)$$

Considerando a profundidade radial de corte da ferramenta a cada passada como sendo igual a 75 % do diâmetro da ferramenta, logo $a_{e_máximo}$ seria 2,25 mm, mas existem situações nas que a ferramenta tem de entrar com toda a profundidade radial ($a_{e_máximo} = 3,00mm.$) como no caso da ranhura como pode ser observado na figura 4.40. A equação 4.8 apresenta a relação existente entre a velocidade da ferramenta e a taxa de remoção.

$$velocidade_Ferramenta \times \acute{a}rea = taxa\ de\ remo\c{c}\tilde{a}o \quad (4.8)$$

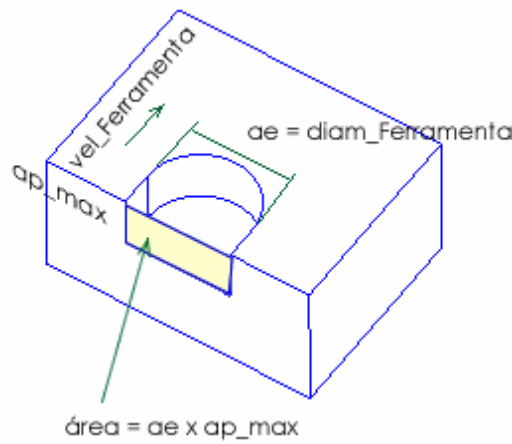


Figura 4.40. Condições de usinagem $a_{p_máxima}$ e $a_{e_máximo}$.

Da figura 4.40 se observa que a área pode ser substituída por $(a_{p_máxima}) \times (a_{e_máximo})$

Substituindo a área da equação 4.8, por $a_{p_máxima}$ e $a_{e_máxima}$, pode-se deduzir a profundidade de corte $a_{p_máxima}$:

$$a_{p_máxima} = \left(\frac{Taxa\ de\ remo\c{c}\tilde{a}o}{velocidade\ de\ movimenta\c{c}\tilde{a}o * a_e} \right) [mm] \quad (4.9)$$

$$a_{p_máxima} = \left(\frac{73 \left(\frac{mm^3}{s} \right)}{10 \left(\frac{mm}{s} \right) * 3,00(mm)} \right) [mm] \quad (4.10)$$

$$a_{p_máxima} = 2,43\ mm \approx 2,00.$$

4.3.2 Determinação das estratégias de usinagem para cada tipo de *feature*

Para cada tipo de *feature* é associada uma estratégia de usinagem, e a seguir se descreverão essas estratégias:

4.3.2.1 Estratégia de usinagem para o furo

Na estratégia de usinagem associada a um furo, a ferramenta se movimenta até o centro geométrico do furo e com três milímetros acima da superfície da face, como ilustrado na figura 4.41a. Logo depois a ferramenta desce e penetra até a seguinte profundidade axial de corte: $a_{p_utilizada} = a_{p_máxima}$. Ao atingir esta profundidade, a movimentação é dividida em dois tipos: primeiramente radial e depois circular, como ilustrado na figura 4.41b.

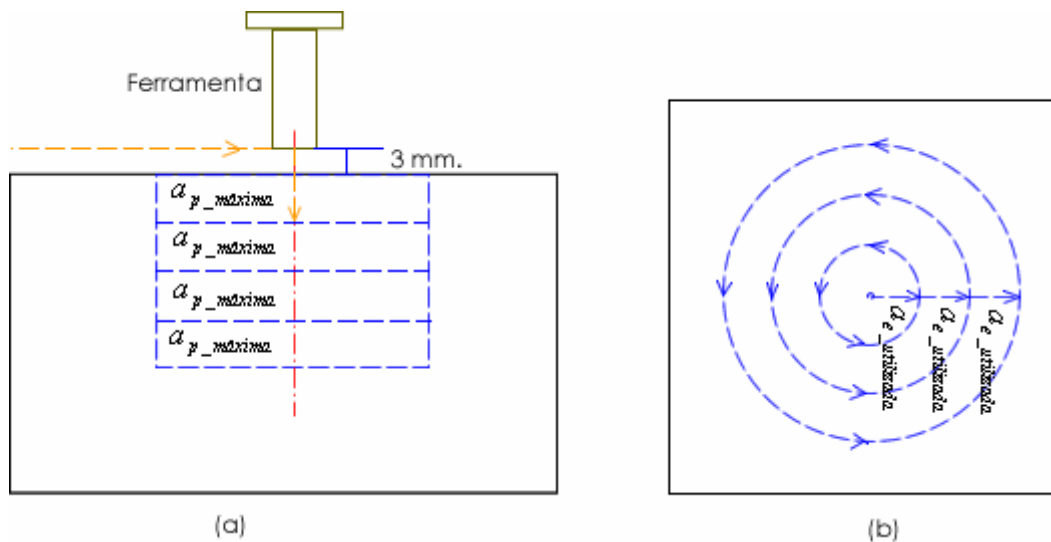


Figura 4.41. Estratégia de movimentação do furo. (a) movimentação axial (b) movimentação radial e circular.

Para usinar a profundidade total do furo, a ferramenta descera uma quantidade $a_{p_utilizada}$ (movimentação no eixo z) que é calculada da seguinte maneira:

(a) caso a diferença ($prof_FURO - z_{atual}$) (figura 4.42a) for maior ou igual do que a profundidade de corte máxima ($a_{p_máxima}$), tem-se:

$$a_{p_utilizada} = a_{p_máxima} \quad (4.11)$$

(b) caso a diferença ($prof_FURO - z_{atual}$) for menor do que a profundidade de corte máxima (figura 4.42b), tem-se:

$$a_{p_utilizada} = \text{profundidade Total} - z_{atual} \quad (4.12)$$

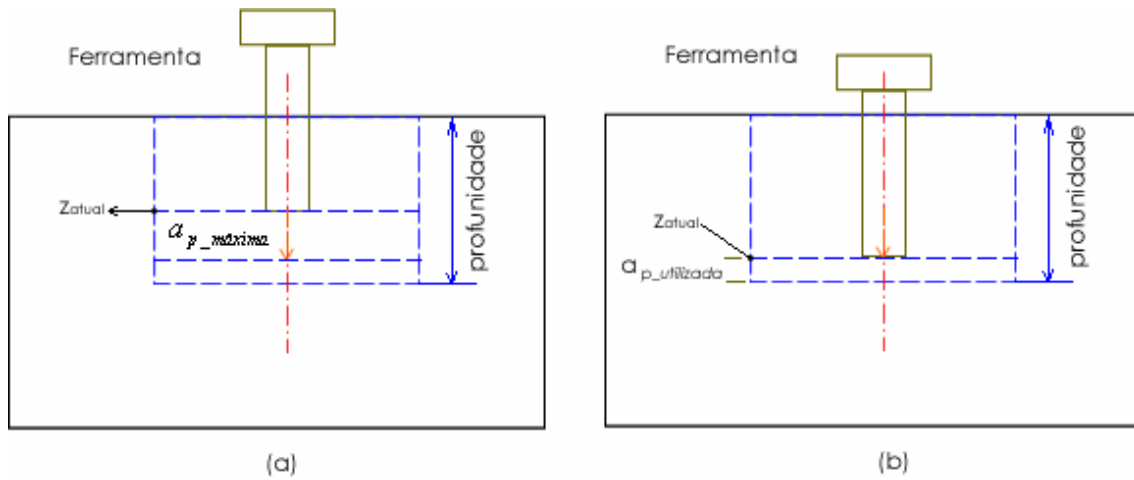


Figura 4.42. Determinação da profundidade de corte $a_{p_utilizada}$.

Uma vez atingida a profundidade total da *feature*, a ferramenta volta para a posição inicial, isto é, ao centro geométrico do furo e com três milímetros acima da peça.

No sentido radial, deve-se calcular o valor de $a_{e_utilizado}$, e isto é feito da seguinte maneira (Figura 4.43).

- se $raio_Atual + a_{e_máximo} + raio_Ferramenta \leq raio_FURO$:

$$a_{e_utilizada} = a_{e_máximo} \quad (4.13)$$

- se $raio_Atual + a_{e_máximo} + raio_Ferramenta > raio_FURO$:

$$a_{e_utilizada} = raio_FURO - (raio_Atual + raio_Ferramenta) \quad (4.14)$$

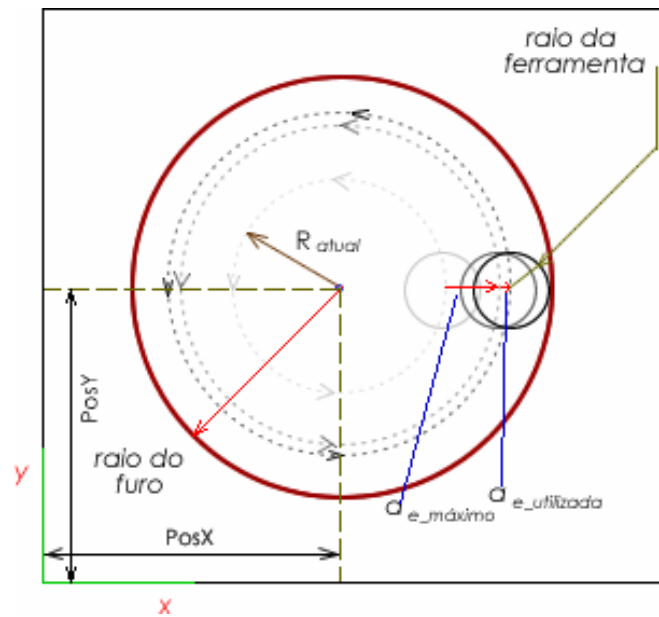


Figura 4.43. Determinação de $a_{e_utilizado}$ para o furo.

Uma vez feita a movimentação radial, a ferramenta começará a fazer a movimentação circular através da interpolação circular, e finalmente a ferramenta volta para o centro. Os pontos para a interpolação circular são determinados da seguinte maneira:

$$x_i = posX + R_{atual} * \cos(\alpha_{inicial} + \alpha_i) \quad (4.15)$$

$$y_i = posY + R_{atual} * \sin(\alpha_{inicial} + \alpha_i) \quad (4.16)$$

Onde:

$$\alpha_i = \frac{2 * \pi}{n} \quad (4.17)$$

n é o número de pontos a interpolar na circunferência,

$$\alpha_{inicial} = 0^0$$

$posX$ e $posY$ são as coordenadas do centro da circunferência.

4.3.2.2 Estratégia de usinagem para o degrau e a ranhura

A estratégia associada a estas *features* consiste em começar a usinagem por qualquer das quatro extremidades do degrau ou da ranhura, como mostrado na figura 4.44(a). Para estas *features*, há três tipos de movimentações: no eixo z , no eixo x e no eixo y .

Uma vez que a ferramenta se posiciona sobre uma extremidade acima da peça (figura 4.44b), a ferramenta descerá até atingir $a_{p_utilizada}$ (movimentação no eixo z), como ilustrado na figura 4.44(b). Se a ranhura tiver o seu comprimento na direção x , a ferramenta fará inicialmente a movimentação nesta direção até chegar ao final da face, e depois faz a movimentação na direção y , como se vê na figura 4.44(a).

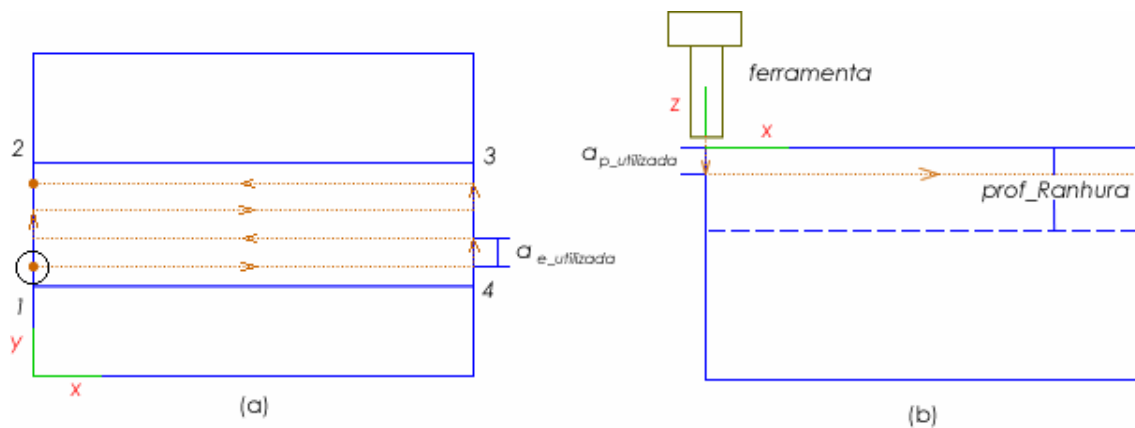


Figura 4.44. Estratégia de movimentação da ferramenta para a usinagem de uma ranhura

Enquanto o cálculo de $a_{p_utilizada}$ é similar ao caso do furo, o cálculo de $a_{e_utilizada}$ para uma ranhura ou degrau é feito da seguinte maneira(ver figura 4.45):

- se $y_{atual} + \text{ráio_Ferramenta} \leq \text{largura} + \text{posY}$

$$a_{e_utilizada} = a_{e_máximo} \quad (4.18)$$

- se $y_{atual} + \text{ráio_Ferramenta} > \text{largura} + \text{posY}$

$$a_{e_utilizada} = \text{largura} + \text{posY} - (y_{atual} + \text{ráio_Ferramenta}) \quad (4.19)$$

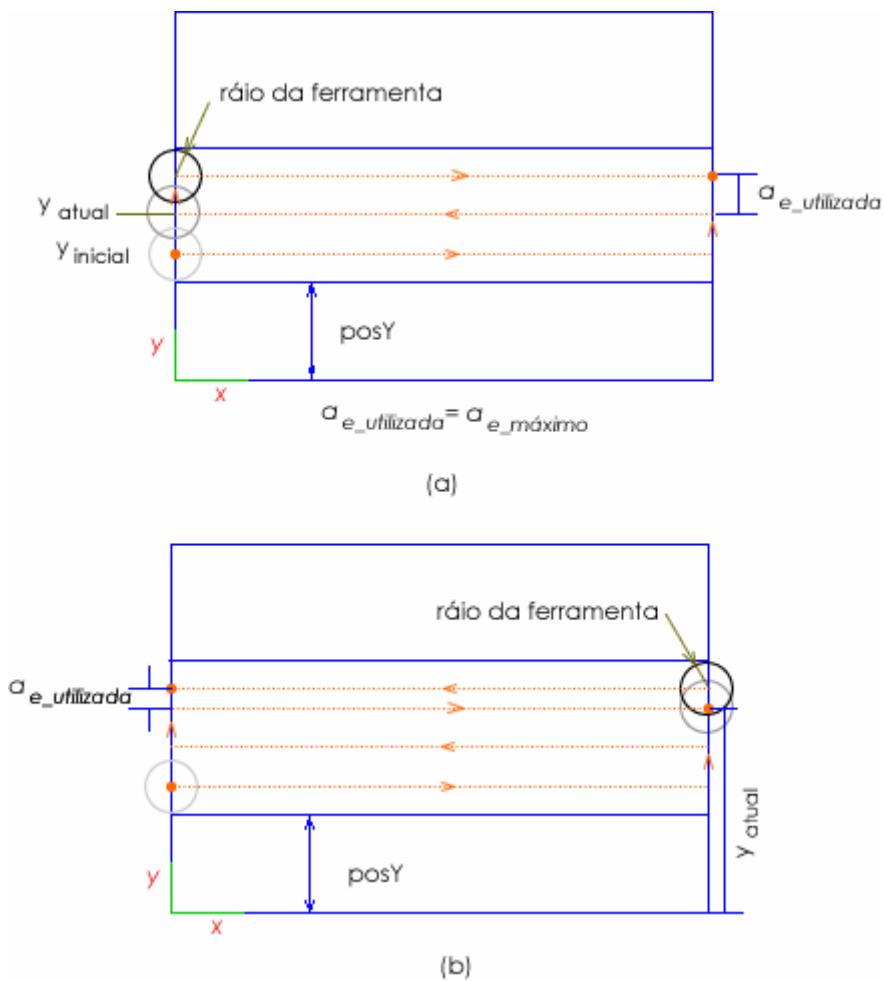
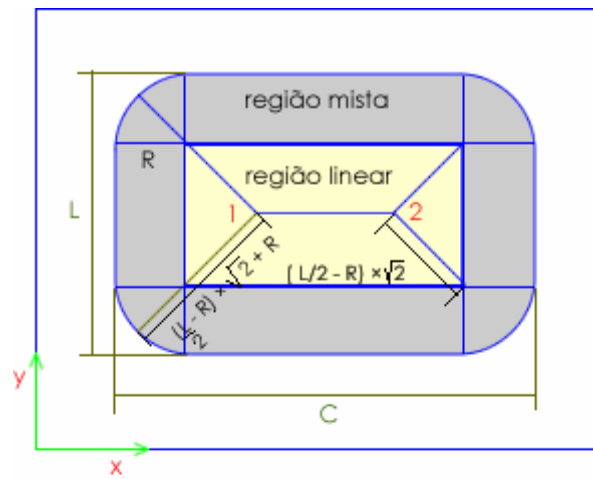


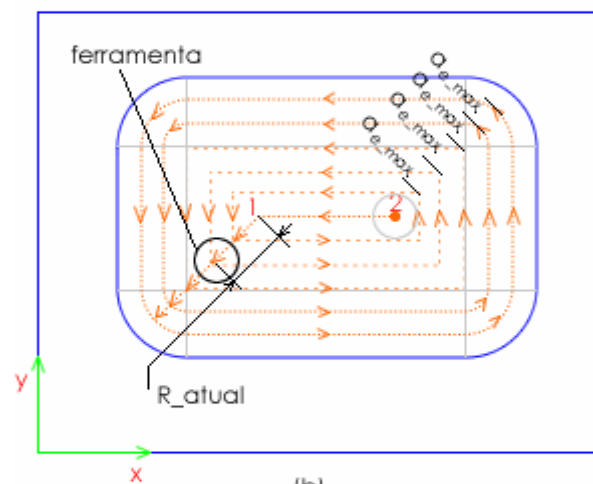
Figura 4.45. Determinação do valor de $a_{e_utilizado}$ para uma ranhura.

4.3.2.3 Estratégia de usinagem para a cavidade retangular

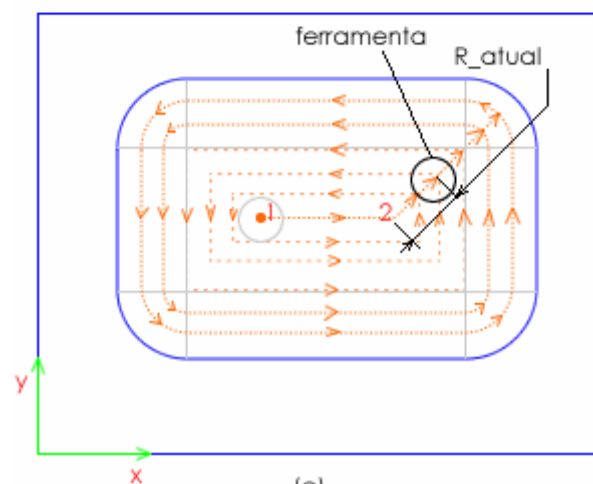
No caso de uma cavidade retangular, a movimentação divide-se em movimentos lineares e movimentos circulares como ilustrado na figura 4.46a. Para usinar esta *feature*, a ferramenta pode posicionar-se em qualquer um dos dois pontos iniciais mostrados nas figuras 4.46b e 4.46c.



(a)



(b)



(c)

Figura 4.46. Movimentação da ferramenta para a usinagem de uma cavidade retangular.

O cálculo do valor de $a_{p_utilizada}$ também é similar aos casos anteriores, enquanto o cálculo de $a_{e_utilizado}$ para uma cavidade retangular cujo comprimento (dimensão x) é maior do que a sua largura (dimensão y), se apresenta a seguir:

Para a região onde a movimentação é somente linear (isto é,

$$R_{atual} + raio_Ferramenta \leq \left(\frac{L}{2} - R\right) \times \sqrt{2}:$$

$$\text{- se } R_{atual} + a_{e_m\acute{a}xima} + raio_Ferramenta \leq \left(\frac{L}{2} - R\right) \times \sqrt{2}:$$

$$a_{e_utilizado} = a_{e_m\acute{a}ximo} \quad (4.20)$$

$$\text{- se } R_{atual} + a_{e_m\acute{a}xima} + raio_Ferramenta > \left(\frac{L}{2} - R\right) \times \sqrt{2}$$

$$a_{e_utilizado} = \left(\frac{L}{2} - R\right) \times \sqrt{2} - (R_{atual} + raio_Ferramenta) \quad (4.21)$$

Para a região onde a movimentação é circular e linear (isto é,

$$\left(\frac{L}{2} - R\right) \times \sqrt{2} > R_{atual} \geq \left(\frac{L}{2} - R\right) \times \sqrt{2} + R):$$

$$\text{- se } R_{atual} + a_{e_m\acute{a}xima} + raio_Ferramenta \leq \left(\frac{L}{2} - R\right) \times \sqrt{2} + R:$$

$$a_{e_utilizado} = a_{e_m\acute{a}ximo} \quad (4.22)$$

$$\text{- se } R_{atual} + a_{e_m\acute{a}xima} + raio_Ferramenta > \left(\frac{L}{2} - R\right) \times \sqrt{2} + R:$$

$$a_{e_utilizado} = \left(\frac{L}{2} - R\right) \times \sqrt{2} + R - (R_{atual} + raio_Ferramenta) \quad (4.23)$$

Na parte circular desta região, são interpolados os pontos de um arco de circunferência de 45° , logo a ferramenta deverá seguir trajetória linear até chegar ao outro arco de circunferência e fará a interpolação circular de 90° , e assim até chegar à posição na qual

começou (figura 4.47). A maneira de calcular os pontos da interpolação circular é similar ao caso do furo (equações 4.15 e 4.16), mas o ângulo α_i é calculado de modo diferente:

$$\alpha_i = \frac{(\alpha_{inicial} - \alpha_{final})}{n} \quad (4.24)$$

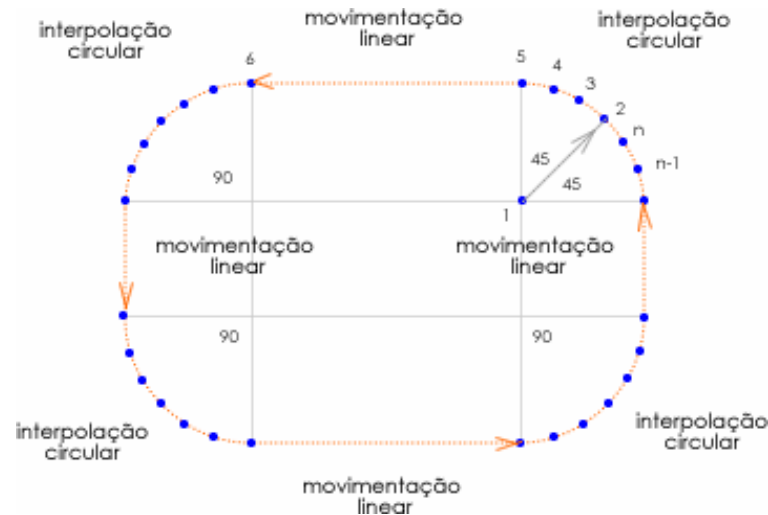


Figura 4.47. Trajetórias na região de movimentação circular e linear.

Com essas condições de usinagem ($a_{p_utilizado}$ e $a_{e_utilizado}$), se calculam os pontos de movimentação da ferramenta, e tais pontos são armazenados em um vetor chamado “movimentação”, que será adjunto ou acrescentado a cada *workingstep*, completando assim os dados necessários para gerar o código HPGL.

4.3.3 Formação de grupos de *workingsteps*

Existem algumas regras de produção que se devem levar em conta antes da geração da seqüência em que serão usinadas as diversas *features* de uma peça. Como exemplo de regra, tem-se que não se pode começar a usinagem por uma *feature* cuja posição z seja maior do que a posição z de uma outra *feature*. Por exemplo, na figura 4.48 se apresenta um degrau e uma cavidade retangular a serem usinados, e nesse exemplo dever-se-á usinar o degrau em primeiro lugar, e depois a cavidade retangular.

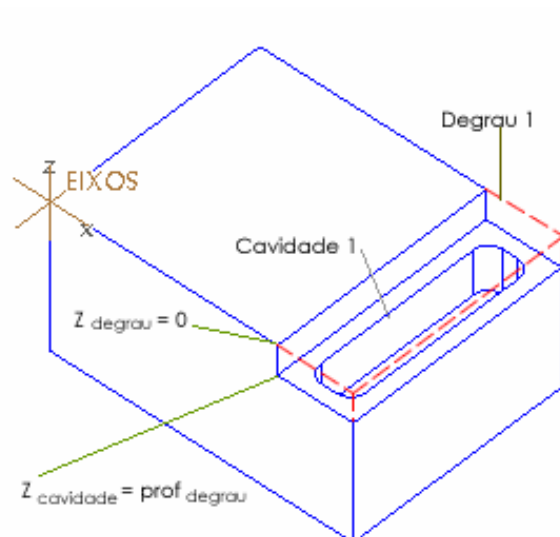


Figura 4.48. Determinação da seqüência de usinagem para duas *features*

O critério utilizado neste sistema CAPP é que se deve garantir que as *features* que estão mais próximas da superfície da face serão as primeiras a serem usinadas. O grupo de *workingsteps* de uma face é então reagrupado em grupos chamados aqui de “camadas”, e um exemplo deste agrupamento é mostrado na figura 4.49, onde é apresentada uma peça que tem na sua face XY quatro *features* (um degrau e três furos). Como todas estas *features* se originam na face XY, mas elas têm diferentes coordenadas *z*, faz-se um agrupamento dos *workingsteps* em função da face que origina a *feature*. Na figura 4.50 se ilustra a estruturação dos *workingsteps* antes e depois de serem agrupados. Dentro de cada camada criada, existirá al menos um *workingstep* a ser usinado.

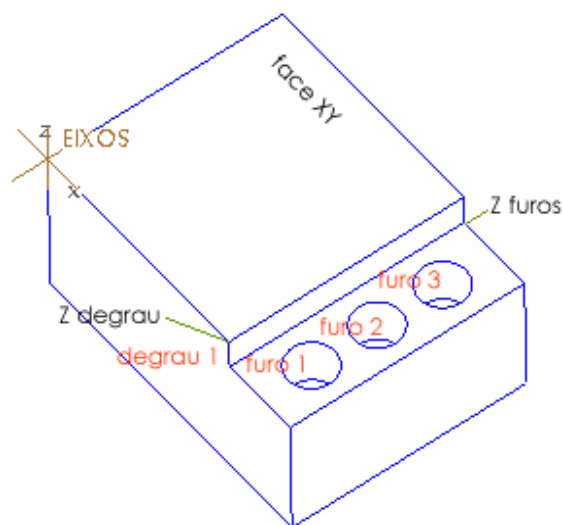


Figura 4.49 Peça exemplo com *features* em uma face.

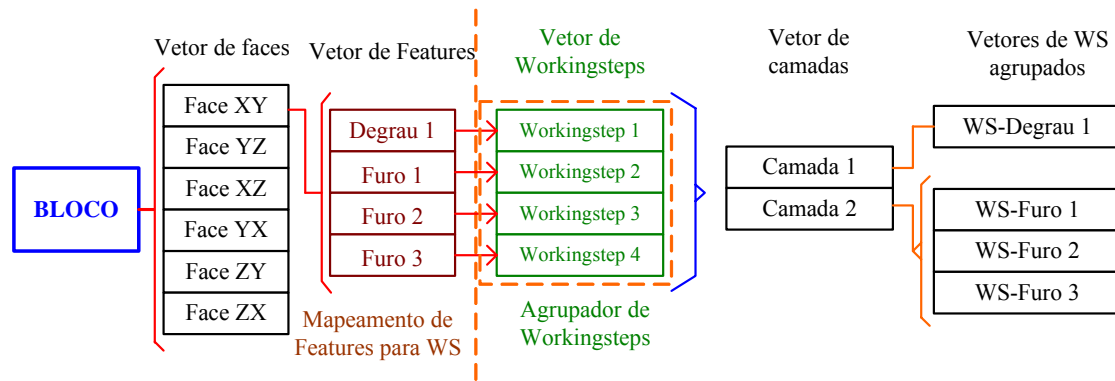


Figura 4.50 Organização das *features* por camadas, em função da face onde elas se originam.

4.3.4 Determinação das seqüências de usinagem

Até este ponto, um *workingstep* contém todos os dados relacionados à sua manufatura, uma vez que já foram calculadas as condições de usinagem, e tem-se à disposição o vetor de pontos referentes à movimentação que a ferramenta deve fazer para usinar cada *workingstep* individual. Além disso, foram agrupadas as *workingsteps* por camadas. Agora deve-se determinar a seqüência em que os *workingsteps* serão usinados, e isto é feito através de algumas regras que calculam a mínima distância que a ferramenta deverá percorrer entre os *workingsteps* da mesma camada. Se a camada somente tiver somente um *workingstep*, então a seqüência consistirá da usinagem desse único *workingstep*. Por exemplo, para a peça apresentada na figura 4.51, que foi descomposta em duas camadas, a primeira camada contém apenas um degrau, e assim esse degrau deverá ser usinado, mas o problema para este caso consiste em escolher dentre os quatro pontos potenciais de início da usinagem. O sistema implementa algoritmos que calculam essas distâncias e seleciona o ponto mais próximo do ponto atual (ver figura 4.51).

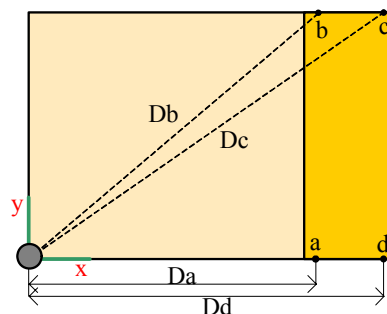


Figura 4.51 Possíveis pontos de início da usinagem do degrau.

No caso da existência de várias *features* em uma mesma camada, como no caso da figura 4.52a, onde naquela camada há três furos, a *feature* a ser usinada primeiro é selecionada considerando-se o ponto atual da ferramenta (supondo que a usinagem do degrau anterior tenha deixado a ferramenta nesse ponto). O algoritmo implementado calcula todas as distâncias deste ponto até cada um dos furos (ou seja, distância 1, distância 2 e distância 3), e a *feature* a ser usinada primeiro será a que tem menor valor dentre estas distâncias. No caso da figura 4.52b, após o primeiro furo ter sido usinado (F3), a ferramenta está posicionada no centro do mesmo (ponto atual). A partir desse ponto, se calcula novamente as distâncias das *features* restantes, e assim prossegue-se até determinar-se a seqüência em que serão usinadas as *features* de toda a camada.

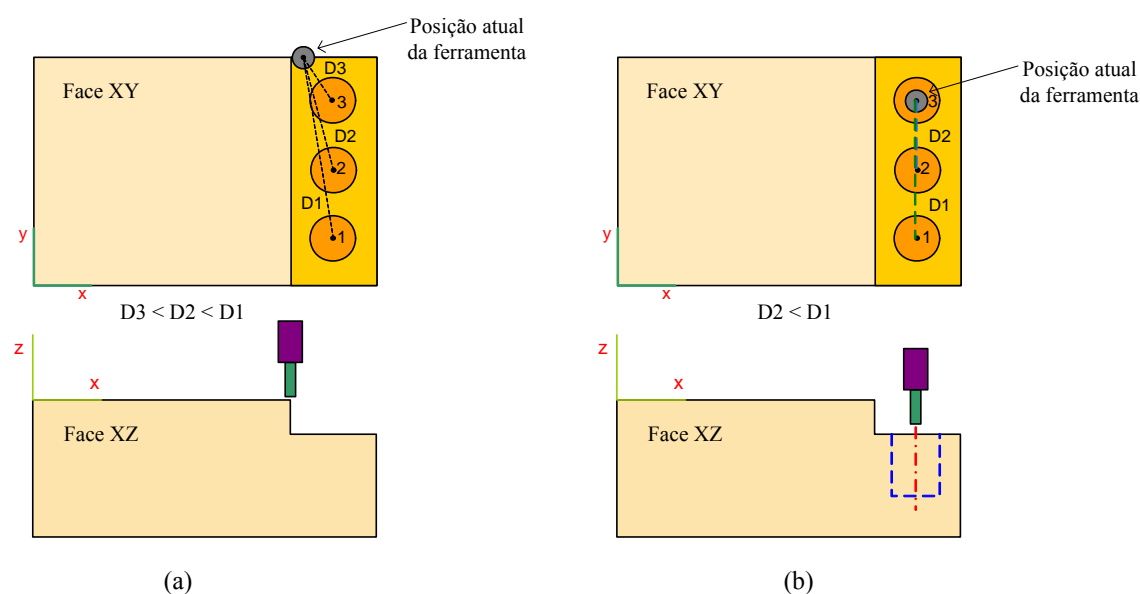
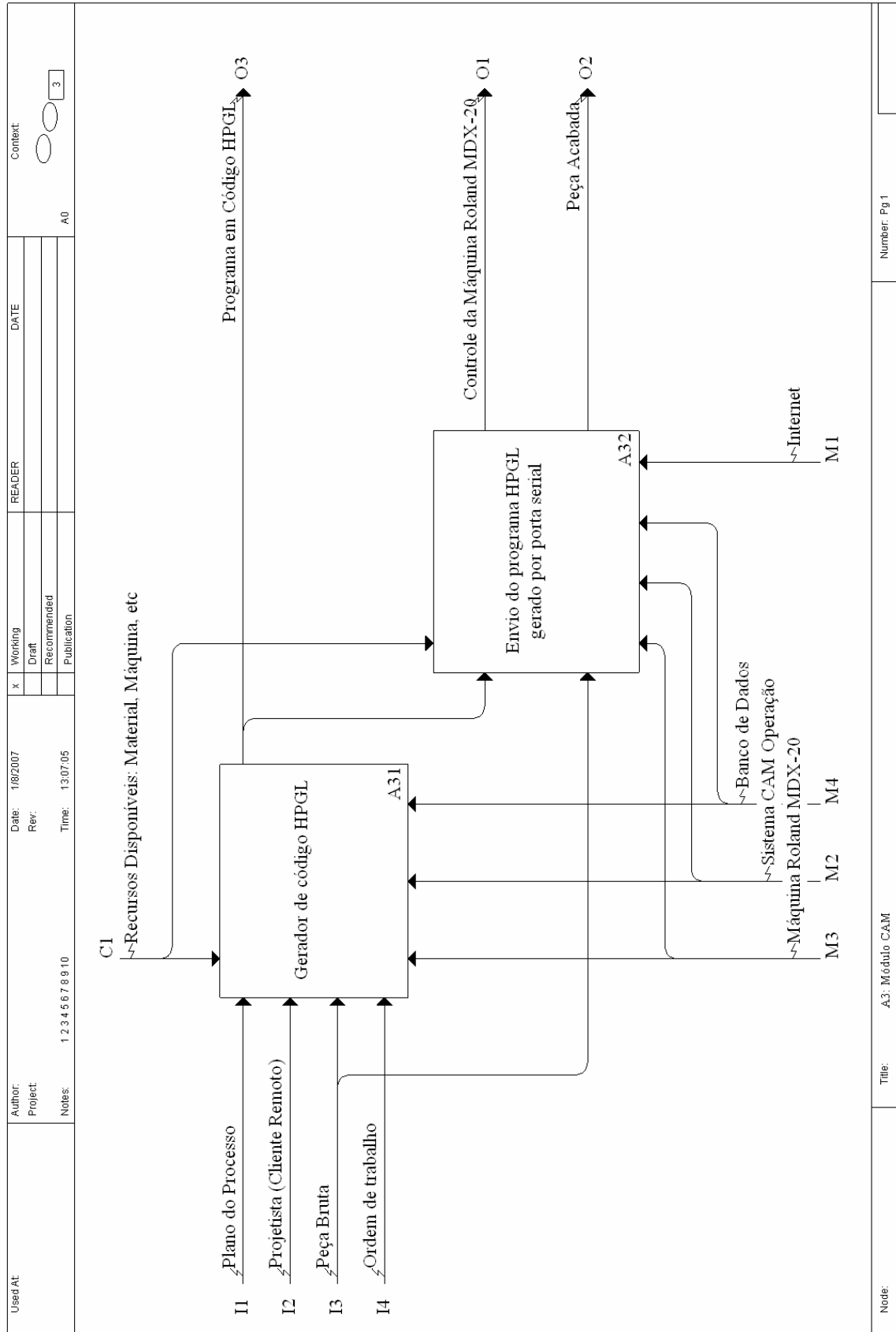


Figura 4.52. Critério da mínima distância para o sequenciamento de *features* em uma camada.

O resultado disto é um vetor contendo os *workingsteps* ordenados, e cada um desses *workingsteps* conterà a especificação dos pontos de controle a partir dos quais a usinagem iniciará.

4.4 Módulo CAM

O módulo CAM está encarregado da geração do código HPGL e de carregá-lo na máquina de prototipagem rápida por usinagem Roland MDX-20 pela porta serial. As atividades deste módulo estão apresentadas na figura 4.53 em um diagrama IDEF0.



Title: A3: Módulo CAM

Number: Pg 1

Figura 4.53. Diagrama IDEF0 das atividades do módulo CAM.

4.4.1 Geração do código HPGL

Cada *workingstep* contém toda a informação necessária para a geração do código HPGL. Este código é composto por uma série de comandos para controlar em sua maior parte plotters, e impressoras, e dependendo do modelo, do fabricante, etc, implementam comandos que nem sempre têm todas as máquinas.

Alguns comandos HPGL que esta máquina de prototipagem rápida implementa são os seguintes:

- Comando D (*Draw – Line-cutting*)

Seqüência de segmentos de reta entre as coordenadas especificadas. Os valores das coordenadas são absolutos, e devem ser do tipo *float*. Exemplo:

D x1, y1, x2, y2,xn, yn;

- Comando I (*Relative draw*)

Realiza a usinagem desde a posição atual da ferramenta com uma altura *z* até as coordenadas relativas *dx* e *dy*. Exemplo:

I dx1, dy1, dx2, dy2dxn, dyn;

- Comando H (*Home – Origin-movement*)

Move a ferramenta até a posição zero da máquina, e quando esta posição é atingida o motor da ferramenta para de girar. Exemplo:

H;

- Comando Z (*XYZ move – three-axis simultaneous feed*)

Move a peça em relação à ferramenta simultaneamente ao longo dos três eixos (x, y, z) das coordenadas atuais até as coordenadas especificadas (coordenadas absolutas). Exemplo:

Z x1, y1, z1;

- Comando IN (*Initialize-setting*)

Este comando realiza as operações de fixar as coordenadas absolutas, fixar as velocidades de movimentação nos três eixos e inicializar a rotação do eixo da ferramenta.

Existem ainda outros comandos implementados por outros tipos de máquinas ou plotadores, que implementam algumas formas geométricas como por exemplo movimentação circular. Mas no caso da máquina presente no laboratório, o comando para a movimentação circular não está presente, e devido a isso foi necessária a implementação da interpolação circular já mencionada no item 4.3.2.1 onde se fez o cálculo dos pontos para uma circunferência (equações 4.14 e 4.15). O número de pontos para a interpolação é variável, pois quando o furo tem diâmetro maior que duas vezes o diâmetro da ferramenta, então se utiliza separação entre um ponto e outro de aproximadamente o diâmetro da ferramenta dividido por quatro. E quando o furo tem diâmetro menor que duas vezes o diâmetro da ferramenta, a separação entre pontos é igual ao diâmetro da ferramenta dividido por oito, aumentando deste modo a precisão para estes furos (figura 4.54).

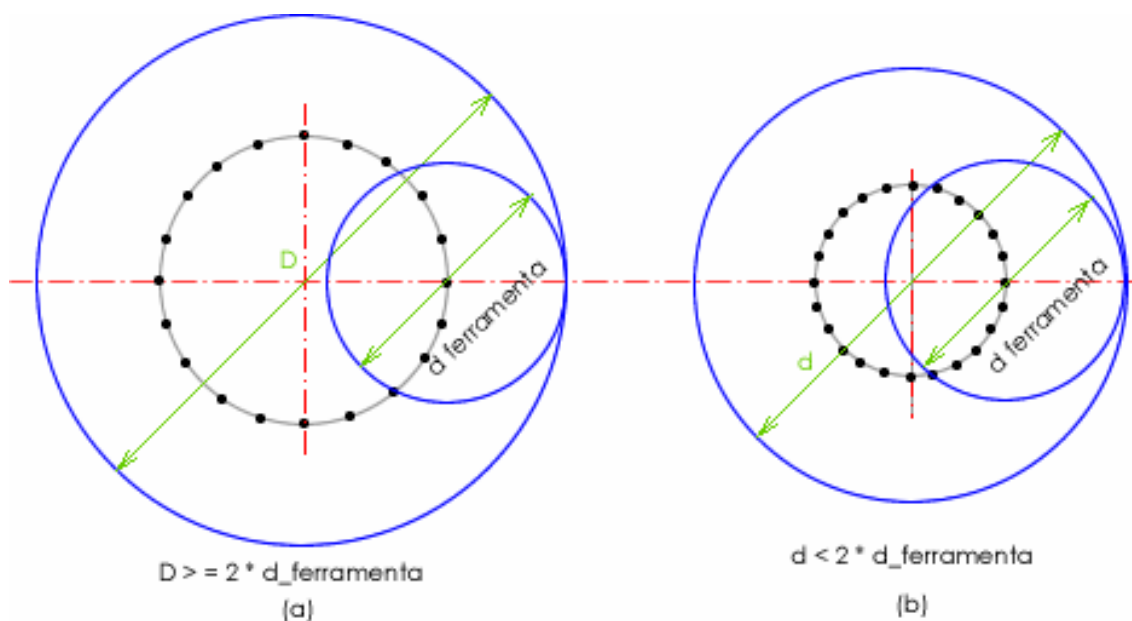


Figura 4.54. Número de pontos utilizados na interpolação circular.

Agora o problema ser resolvido consiste em transformar as coordenadas dos pontos que estão armazenados no vetor de movimentação em coordenadas absolutas da máquina. Por exemplo, na figura 4.55 pode-se ver que o zero máquina não coincide com o zero peça, e o programa deverá determinar este ponto.

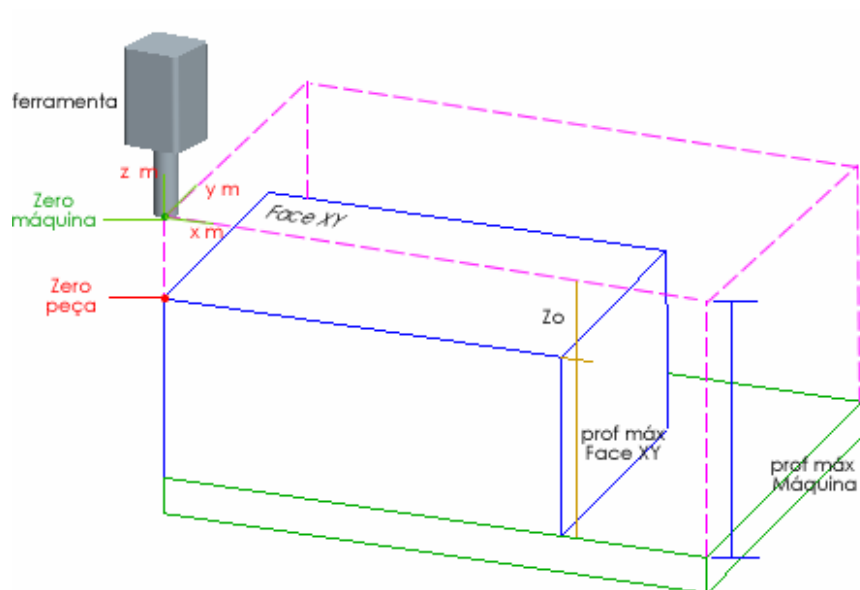


Figura 4.55. Fixação da peça na máquina.

Então a ferramenta deverá se deslocar no eixo z uma quantidade z_0 :

$$z_0 = \text{profundidade_Máquina} - \text{profundidade_máxima_Face} \quad (4.24)$$

Todos os pontos do vetor de movimentação são calculados tendo em conta que o zero peça é onde começa a superfície da face, e assim todos esses pontos devem ser deslocados no eixo z a quantidade z_0 e ser multiplicados por -1, pois a máquina considera $(+z)$ como movimentação no sentido ascendente, e $(-z)$ no sentido descendente.

Os únicos comandos utilizados pelo módulo CAM deste sistema computacional são: o comando IN, para iniciar a movimentação do eixo da ferramenta; o comando Z, para enviar a ferramenta às coordenadas absolutas de cada ponto do vetor de movimentação; e o comando H, para uma vez acabada a usinagem de uma face mandar a ferramenta para a origem absoluta. No exemplo apresentado na figura 4.56, a ferramenta desce do ponto $z = (0, 0, 0)$ para $z = (0, 0, -Z_0)$, e depois faz a movimentação ao longo da aresta da face até o ponto $z = (c, 0, -Z_0)$, e finalmente volta para a posição de origem $z = (0, 0, 0)$.

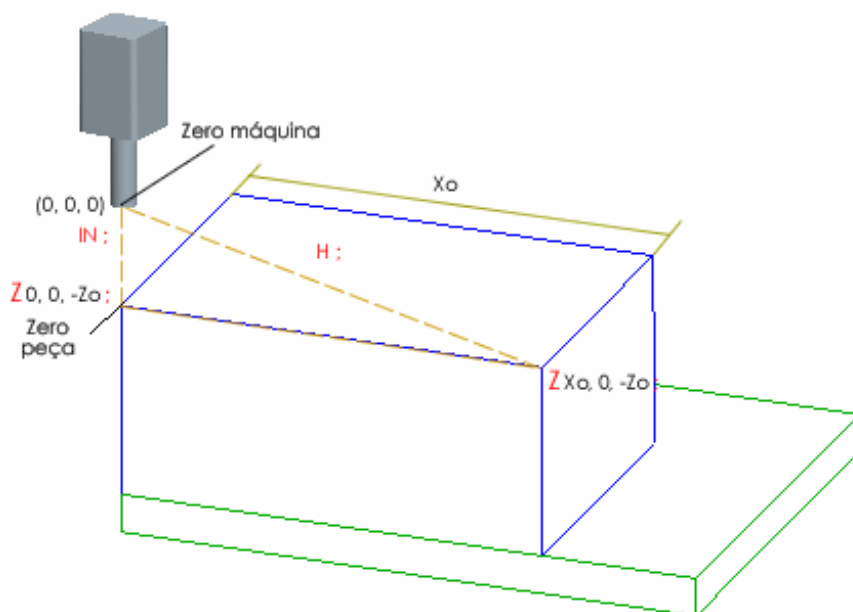


Figura 4.56. Ilustração do funcionamento dos comandos HPGL utilizados pelo módulo CAM

O código para esta movimentação seria:

IN ;

Z 0 , 0 , -Z0 ;

Z X0, 0, -Z0 ;

H ;

4.4.2 Envio do código HPGL para a máquina por porta serial

Uma vez gerado o código, e salvo no servidor em um arquivo HCF (“HPGL Code File”), ele é enviado à máquina através da porta serial, e isto pode ser feito utilizando-se o seguinte comando em DOS :

```
D:\> type endereço_do_arquivo \ Nome_do_arquivo.HCF > com1
```

Este arquivo também pode ser carregado na máquina através da utilização de um aplicativo especializado para o envio de arquivos por porta serial. A interface gráfica deste aplicativo se apresenta na figura 4.57.

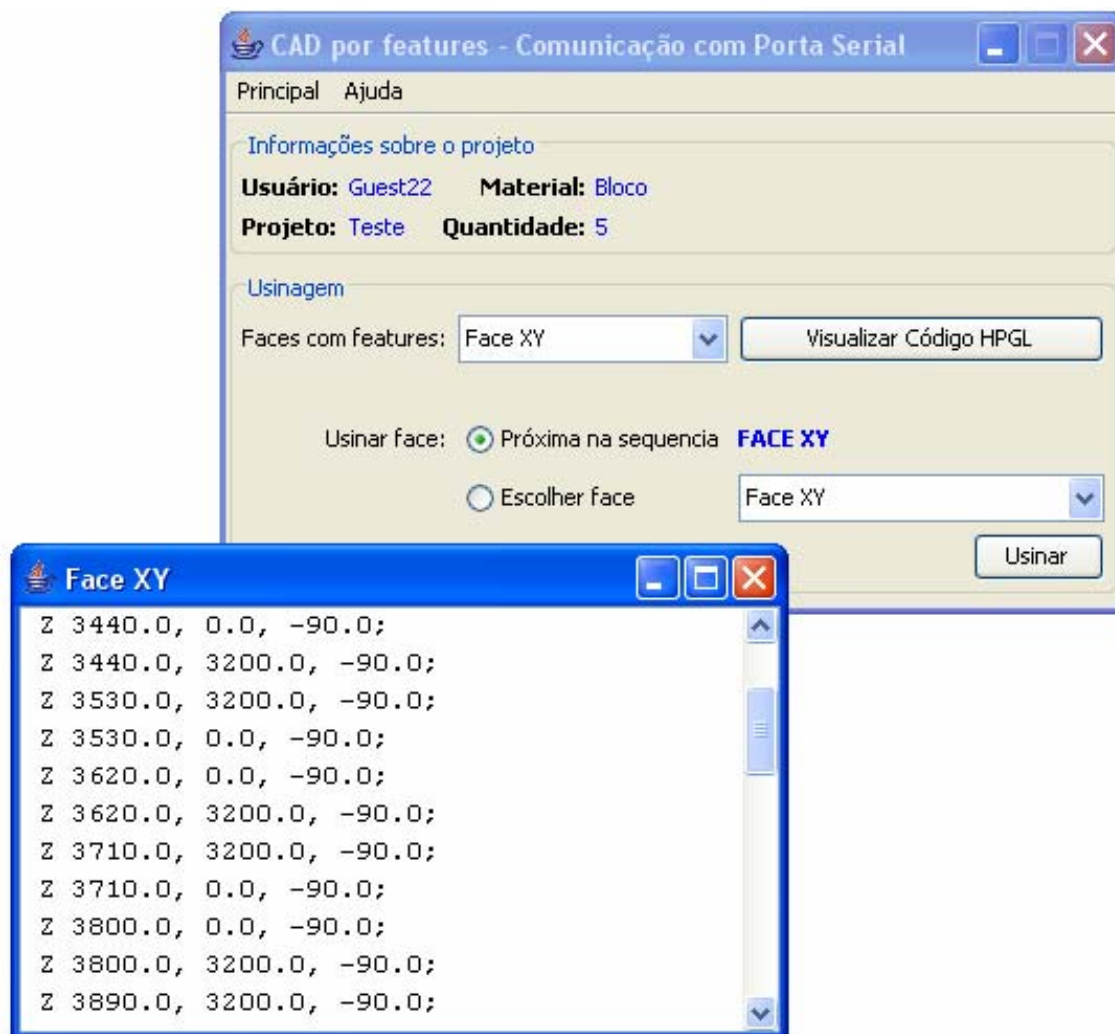


Figura 4.57. Interface gráfica do aplicativo que envia o código HPGL por porta serial.

A peça bruta deve ser fixada na mesa da máquina levando-se em consideração a posição onde o novo bloco foi instanciado, isto é, com o vértice zero da face a ser usinada coincidindo com a origem de coordenadas da máquina.

4.5 Envio da ordem de fabricação

Como mencionado anteriormente, este sistema computacional em primeiro lugar é carregado no lado do cliente remoto na forma de um *applet*, através do qual efetua-se a modelagem de peças prismáticas. Uma vez concluída a modelagem, o cliente remoto pode enviar a ordem de fabricação ou simplesmente pode salvar o projeto no servidor e continuar modelando a peça em outra ocasião. Porém no momento do envio da ordem de fabricação, o pedido é cadastrado no banco de dados, e o mesmo é colocado na lista de ordens de fabricação, onde se começará a fabricar as peças segundo a ordem da lista,

onde o primeiro pedido que chegar é o primeiro em ser fabricado. A figura 4.58 apresenta a estrutura do sistema computacional.

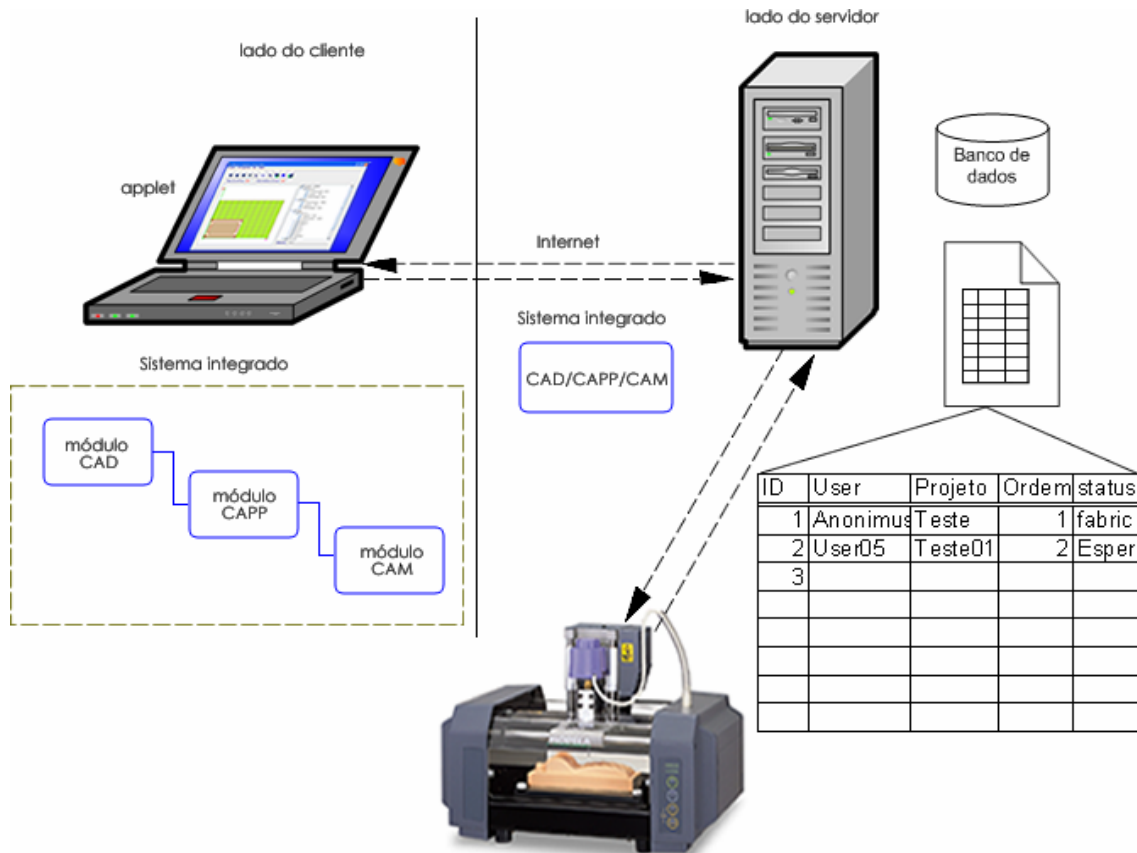


Figura 4.58 Arquitetura do sistema sendo executado na internet.

Toda vez que uma peça está sendo fabricada, o banco de dados é atualizado e o cliente remoto poderá saber o status do seu pedido, que pode ser “fabricando”, “na fila” ou “pronto”.

CAPÍTULO 5

RESULTADOS OBTIDOS, ESTUDOS DE CASOS

Neste capítulo são apresentados três estudos de casos, acompanhando o produto desde a modelagem até chegar ao resultado final que é a fabricação da peça prismática, com o objetivo de testar o funcionamento do sistema desenvolvido.

5.1 Definição das peças a serem estudadas

Três peças foram modeladas, e na tabela 5.1 são apresentadas as *features* que as constituem.

1. Uma peça prismática gerada a partir da usinagem de um bloco de 75 mm de comprimento por 50 mm de largura, e 30 mm de profundidade. O material utilizado é uma resina de boa usinabilidade (NECURON – 301⁹) em termos de resistência, mas o cavaco gerado tem forma de pó.
2. Uma peça prismática gerada a partir da usinagem de um bloco de 66 mm de comprimento por 42 mm de largura e 21 mm de profundidade. O material é o cibatool, que é um dos materiais mais utilizados para prototipagem rápida, tem boa usinabilidade, tanto do ponto de vista do cavaco quanto do acabamento da superfície.
3. Uma peça prismática gerada a partir de uma peça bruta de 73 mm de comprimento por 50 mm de largura e 28 mm de profundidade. O material é similar ao da primeira peça.

⁹ NECURON – 301, polímero, utilizado em modelagens de estilo, de arquitetura, modelos de grande volume, etc. Informações acerca das suas características técnicas e de materiais poliméricos podem ser achadas em <http://www.necumer.de/com>.

Tabela 5.1. *Features* presentes nas peças estudadas.

	Face	Furo	Degrau	Ranhura	Cavidade	Total <i>features</i>
Peça 1	XY	16	1	1	6	24
	YZ	-	-	-	-	-
	XZ	-	-	-	-	-
	YX	-	-	-	-	-
	ZY	-	-	-	-	-
	ZX	-	-	-	-	-
Peça 2	XY	12	2	1	2	17
	YZ	-	-	-	-	-
	XZ	-	-	-	-	-
	YX	-	-	-	-	-
	ZY	-	-	-	-	-
	ZX	-	-	-	-	-
Peça 3	XY	6	2	0	2	10
	YZ	-	-	-	-	-
	XZ	0	0	0	4	4
	YX	3	0	1	0	4
	ZY	-	-	-	-	-
	ZX	0	0	0	4	4

5.2 Peça 1

5.2.1 Modelagem da peça 1 no módulo CAD por *features*.

Na figura 5.1 é apresentada a peça 1 modelada a partir do bloco especificado (vista superior).

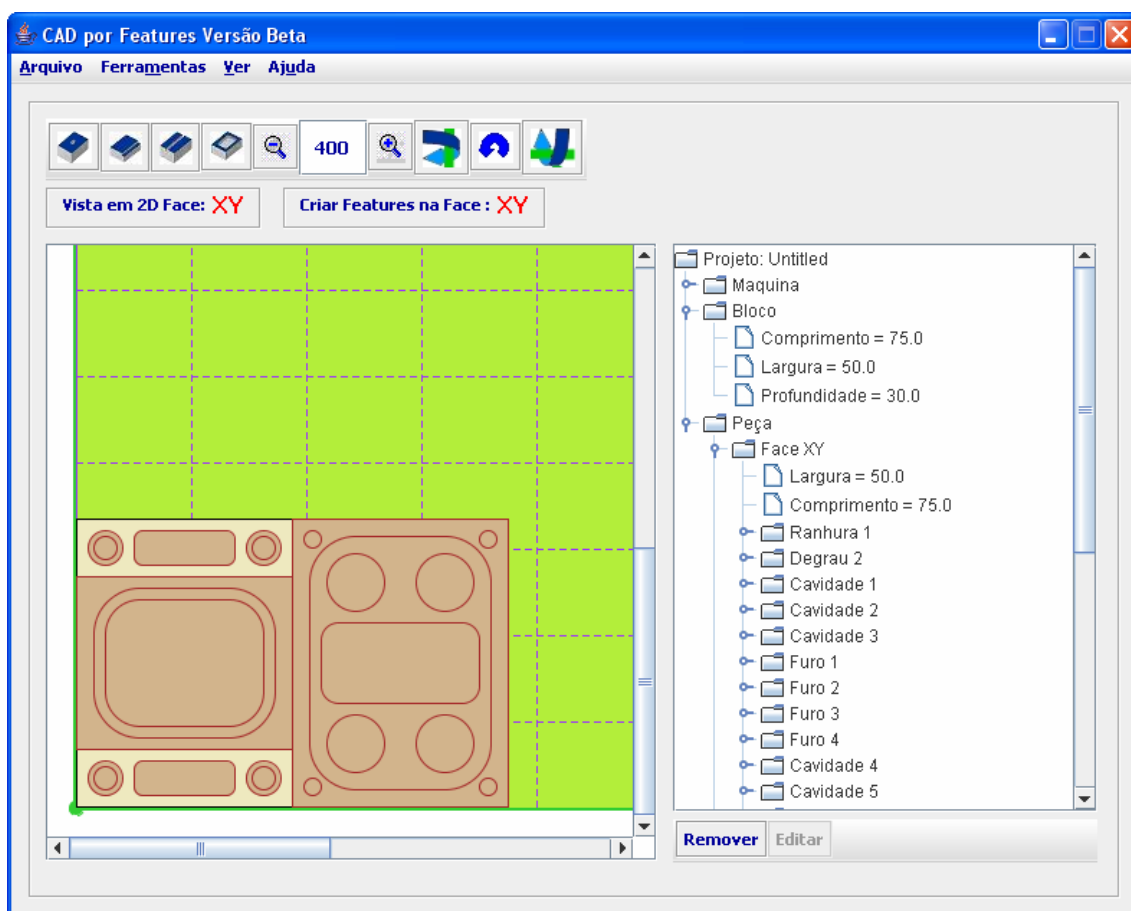


Figura 5.1 Peça 1 visualizada no módulo CAD.

Na parte direita do aplicativo CAD pode-se observar dados importantes referentes à peça, e também à máquina, como por exemplo: as dimensões da máquina selecionada, as dimensões do bloco instanciado, as dimensões da face atual ou “de trabalho” e as *features* presentes na peça.

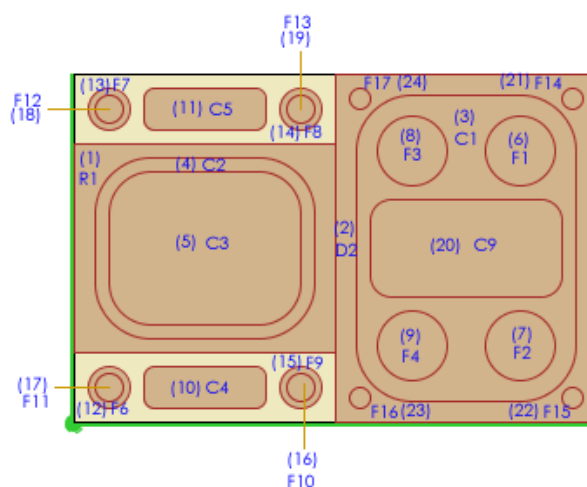
5.2.2 Módulo CAPP

O módulo CAPP recebe as *features* na seqüência em que elas aparecem na estrutura da árvore da interface gráfica do aplicativo CAD. Esta seqüência foi a ordem de introdução das *features* pelo usuário. Na tabela 5.2 são apresentadas as *features* presentes na peça 1.

Tabela 5.2. *Features* presentes na peça 1.

N° de ordem	1	2	3	4	5	6
<i>feature</i>	ranhura 1	degrau 2	cavidade 1	cavidade 2	cavidade 3	furo 1
N° de ordem	7	8	9	10	11	12
<i>feature</i>	furo 2	furo 3	furo 4	cavidade 4	cavidade 5	furo 6
N° de ordem	13	14	15	16	17	18
<i>feature</i>	furo 7	furo 8	furo 9	furo 10	furo 11	furo 12
N° de ordem	19	20	21	22	23	24
<i>feature</i>	furo 13	cavidade 9	furo 14	furo 15	furo 16	furo 17

Na figura 5.2 são apresentadas as *features* que compõem a peça 1. Os números que aparecem entre parêntesis representam a seqüência na qual cada *feature* foi inserida pelo cliente remoto. As letras significam o tipo de *feature* (F → furo, D → degrau, R → ranhura e C → cavidade). Por exemplo, “(5) C3” significa que existe uma cavidade chamada “C3” e que foi inserida em quinto lugar, e portanto ela ocupa o quinto lugar no vetor de *features* da face.

Figura 5.2. *Features* presentes na peça 1 e sua seqüência no vetor de *features*.

O módulo CAPP recebe as informações provenientes do CAD e faz um mapeamento das *features* para adicionar-lhes informações relativas à manufatura, que incluem por exemplo as condições de usinagem e as movimentações da ferramenta. Então, são gerados os grupos de camadas, e finalmente são geradas as seqüências de usinagem para cada camada, começando com a mais próxima à superfície da face até a que está mais longe. Os resultados destas são mostrados na tabela 5.3.

Tabela 5.3. Etapas executadas visando seqüenciar os *Workingsteps* para a peça 1.

AGRUPADOR DE WORKINGSTEPS				ORDENADOR	
Faces	Camadas	Workingsteps	Features	Seqüência	
Face XY (possui 6 camadas)	camada 0 (possui 8 workingsteps)	workingstep 0	Ranhura (1)	Furo (6)	
		workingstep 1	Degrau (2)	Cavidade (4)	
		workingstep 2	Cavidade (4)	Ranhura (1)	
		workingstep 3	Cavidade (5)	Degrau (2)	
		workingstep 4	Furo (6)	Furo (8)	
		workingstep 5	Furo (7)	Cavidade (5)	
		workingstep 6	Furo (8)	Furo (7)	
		workingstep 7	Furo (9)	Furo (9)	
	camada 1(possui 1 workingsteps)	workingstep 0	Cavidade (2)	Cavidade (2)	
	camada 2 (possui 4 workingsteps)	workingstep 0	Furo (10)	Furo (10)	
		workingstep 1	Furo (11)	Furo (11)	
		workingstep 2	Furo (12)	Furo (12)	
		workingstep 3	Furo (13)	Furo (13)	
	camada 3 (possui 5 workingsteps)	workingstep 0	Cavidade (1)	Furo (17)	
		workingstep 1	Furo (14)	Cavidade (1)	
		workingstep 2	Furo (15)	Furo (14)	
		workingstep 3	Furo (16)	Furo (15)	
		workingstep 4	Furo (17)	Furo (16)	
	camada 4(possui 1 workingsteps)	workingstep 0	Cavidade (3)	Cavidade (3)	
	camada 5 (possui 5 workingsteps)	workingstep 0	Furo (1)	Cavidade (9)	
		workingstep 1	Furo (2)	Furo (3)	
		workingstep 2	Furo (3)	Furo (1)	
		workingstep 3	Furo (4)	Furo (2)	
		workingstep 4	Cavidade (9)	Furo (4)	
	Face YZ	-	-	-	-
	Face XZ	-	-	-	-
	Face YX	-	-	-	-
	Face ZY	-	-	-	-
Face ZX	-	-	-	-	

5.2.3 Módulo CAM

O módulo CAM é responsável pela geração do código HPGL, e pelo carregamento do programa na máquina de prototipagem rápida por usinagem Roland MDX-20. Por motivos de espaço, não se colocou o código HPGL inteiro neste trabalho, uma vez que o

código gerado possui mais de duas mil linhas. Assim, somente é apresentado o código utilizado para a usinagem da cavidade 3 (figura 5.3). Na figura 5.4 se apresenta a peça 1 já usinada como resultado deste sistema integrado.

```

IN: -----> iniciar
...
Começa a usinagem da cavidade (3)
Z 860.0, 1000.0, -1090.0;
Z 860.0, 1000.0, -1370.0;
Z 860.0, 1000.0, -1490.0;
Z 640.0, 1000.0, -1490.0;
Z 576.3604, 936.3604, -1490.0;
Z 923.6396, 936.3604, -1490.0;
Z 923.6396, 1063.6396, -1490.0;
Z 576.3604, 1063.6396, -1490.0;
Z 576.3604, 936.3604, -1490.0;
Z 512.72076, 872.72076, -1490.0;
Z 987.27924, 872.72076, -1490.0;
Z 987.27924, 1127.2792, -1490.0;
Z 512.72076, 1127.2792, -1490.0;
Z 512.72076, 872.72076, -1490.0;
Z 449.08118, 809.0812, -1490.0;
Z 1050.9188, 809.0812, -1490.0;
Z 1050.9188, 1190.9188, -1490.0;
Z 449.08118, 1190.9188, -1490.0;
Z 449.08118, 809.0812, -1490.0;
Z 440.0, 800.0, -1490.0;
Z 1060.0, 800.0, -1490.0;
Z 1060.0, 1200.0, -1490.0;
Z 440.0, 1200.0, -1490.0;
Z 440.0, 800.0, -1490.0;
Z 376.36038, 736.3604, -1490.0;
Z 405.5585, 716.8508, -1490.0;
Z 440.0, 710.0, -1490.0;
Z 1060.0, 710.0, -1490.0;
Z 1087.8115, 714.4049, -1490.0;
Z 1112.9006, 727.1885, -1490.0;
Z 1132.8115, 747.0993, -1490.0;
Z 1145.5951, 772.1885, -1490.0;
Z 1150.0, 800.0, -1490.0;
Z 1150.0, 1200.0, -1490.0;
Z 1145.5951, 1227.8115, -1490.0;
Z 1132.8115, 1252.9006, -1490.0;
Z 1112.9006, 1272.8115, -1490.0;
Z 1087.8115, 1285.5951, -1490.0;
Z 1060.0, 1290.0, -1490.0;
Z 440.0, 1290.0, -1490.0;
Z 412.18848, 1285.5951, -1490.0;
Z 387.09933, 1272.8115, -1490.0;
Z 367.18848, 1252.9006, -1490.0;
Z 354.4049, 1227.8115, -1490.0;
Z 350.0, 1200.0, -1490.0;
Z 350.0, 800.0, -1490.0;
Z 356.85083, 765.5585, -1490.0;
Z 376.36038, 736.3604, -1490.0;
Z 312.72076, 672.72076, -1490.0;
Z 334.19867, 654.37695, -1490.0;
Z 358.2817, 639.61884, -1490.0;
Z 384.37695, 628.8098, -1490.0;
Z 411.8418, 622.2161, -1490.0;
Z 440.0, 620.0, -1490.0;
Z 1060.0, 620.0, -1490.0;
Z 1091.2567, 622.7346, -1490.0;
Z 1121.5636, 630.85535, -1490.0;
Z 1150.0, 644.1154, -1490.0;
Z 1175.7018, 662.112, -1490.0;
Z 1197.888, 684.2982, -1490.0;
Z 1215.8845, 710.0, -1530.0;
Z 1229.1447, 738.4364, -1490.0;
Z 1237.2654, 768.74335, -1490.0;
Z 1240.0, 800.0, -1490.0;
Z 1240.0, 1200.0, -1490.0;
Z 1237.2654, 1231.2567, -1490.0;
Z 1229.1447, 1261.5636, -1490.0;
Z 1215.8845, 1290.0, -1490.0;
Z 1197.888, 1315.7018, -1490.0;
Z 1175.7018, 1337.888, -1490.0;
Z 1150.0, 1355.8845, -1490.0;
Z 1121.5636, 1369.1447, -1490.0;
Z 1091.2567, 1377.2654, -1490.0;
Z 1060.0, 1380.0, -1490.0;
Z 440.0, 1380.0, -1490.0;
Z 408.74332, 1377.2654, -1490.0;
Z 378.43637, 1369.1447, -1490.0;
Z 350.0, 1355.8845, -1490.0;
Z 324.29822, 1337.888, -1490.0;
Z 302.112, 1315.7018, -1490.0;
Z 284.11542, 1290.0, -1490.0;
Z 270.85532, 1261.5636, -1490.0;
Z 262.73462, 1231.2567, -1490.0;
Z 260.0, 1200.0, -1490.0;
Z 260.0, 800.0, -1490.0;
Z 262.2161, 771.8418, -1490.0;
Z 268.8098, 744.37695, -1490.0;
Z 279.61884, 718.28174, -1490.0;
Z 294.37695, 694.19867, -1490.0;
Z 312.72076, 672.72076, -1490.0;
Z 312.72076, 672.72076, -1450.0;
Z 860.0, 1000.0, -1450.0;
Z 860.0, 1000.0, -1530.0;
Z 640.0, 1000.0, -1530.0;
Z 576.3604, 936.3604, -1530.0;
Z 923.6396, 936.3604, -1530.0;
Z 923.6396, 1063.6396, -1530.0;
Z 576.3604, 1063.6396, -1530.0;
Z 576.3604, 936.3604, -1530.0;
Z 512.72076, 872.72076, -1530.0;
Z 987.27924, 872.72076, -1530.0;
Z 987.27924, 1127.2792, -1530.0;
Z 512.72076, 1127.2792, -1530.0;
Z 512.72076, 872.72076, -1530.0;
Z 449.08118, 809.0812, -1530.0;
Z 1050.9188, 809.0812, -1530.0;
Z 1050.9188, 1190.9188, -1530.0;
Z 449.08118, 1190.9188, -1530.0;
Z 449.08118, 809.0812, -1530.0;
Z 440.0, 800.0, -1530.0;
Z 1060.0, 800.0, -1530.0;
Z 1060.0, 1200.0, -1530.0;
Z 440.0, 1200.0, -1530.0;
Z 440.0, 800.0, -1530.0;
Z 376.36038, 736.3604, -1530.0;
Z 405.5585, 716.8508, -1530.0;
Z 440.0, 710.0, -1530.0;
Z 1060.0, 710.0, -1530.0;
Z 1087.8115, 714.4049, -1530.0;
Z 1112.9006, 727.1885, -1530.0;
Z 1132.8115, 747.0993, -1530.0;
Z 1145.5951, 772.1885, -1530.0;
Z 1150.0, 800.0, -1530.0;
Z 1150.0, 1200.0, -1530.0;
Z 1145.5951, 1227.8115, -1530.0;
Z 1132.8115, 1252.9006, -1530.0;
Z 1112.9006, 1272.8115, -1530.0;
Z 1087.8115, 1285.5951, -1530.0;
Z 1060.0, 1290.0, -1530.0;
Z 440.0, 1290.0, -1530.0;
Z 412.18848, 1285.5951, -1530.0;
Z 387.09933, 1272.8115, -1530.0;
Z 367.18848, 1252.9006, -1530.0;
Z 354.4049, 1227.8115, -1530.0;
Z 350.0, 1200.0, -1530.0;
Z 350.0, 800.0, -1530.0;
Z 356.85083, 765.5585, -1530.0;
Z 376.36038, 736.3604, -1530.0;
Z 312.72076, 672.72076, -1530.0;
Z 334.19867, 654.37695, -1530.0;
Z 358.2817, 639.61884, -1530.0;
Z 384.37695, 628.8098, -1530.0;
Z 411.8418, 622.2161, -1530.0;
Z 440.0, 620.0, -1530.0;
Z 1060.0, 620.0, -1530.0;
Z 1091.2567, 622.7346, -1530.0;
Z 1121.5636, 630.85535, -1530.0;
Z 1150.0, 644.1154, -1530.0;
Z 1175.7018, 662.112, -1530.0;
Z 1197.888, 684.2982, -1530.0;
Z 1215.8845, 710.0, -1530.0;
Z 1229.1447, 738.4364, -1530.0;
Z 1237.2654, 768.74335, -1530.0;
Z 1240.0, 800.0, -1530.0;
Z 1240.0, 1200.0, -1530.0;
Z 1237.2654, 1231.2567, -1530.0;
Z 1229.1447, 1261.5636, -1530.0;
Z 1215.8845, 1290.0, -1530.0;
Z 1197.888, 1315.7018, -1530.0;
Z 1175.7018, 1337.888, -1530.0;
Z 1150.0, 1355.8845, -1530.0;
Z 1121.5636, 1369.1447, -1530.0;
Z 1091.2567, 1377.2654, -1530.0;
Z 1060.0, 1380.0, -1530.0;
Z 440.0, 1380.0, -1530.0;
Z 408.74332, 1377.2654, -1530.0;
Z 378.43637, 1369.1447, -1530.0;
Z 350.0, 1355.8845, -1530.0;
Z 324.29822, 1337.888, -1530.0;
Z 302.112, 1315.7018, -1530.0;
Z 284.11542, 1290.0, -1530.0;
Z 270.85532, 1261.5636, -1530.0;
Z 262.73462, 1231.2567, -1530.0;
Z 260.0, 1200.0, -1530.0;
Z 260.0, 800.0, -1530.0;
Z 262.2161, 771.8418, -1530.0;
Z 268.8098, 744.37695, -1530.0;
Z 279.61884, 718.28174, -1530.0;
Z 294.37695, 694.19867, -1530.0;
Z 312.72076, 672.72076, -1530.0;
Z 312.72076, 672.72076, -1490.0;
Z 860.0, 1000.0, -1490.0;
Z 860.0, 1000.0, -1090.0;
...
H: -> ir para a posição inicial

```

Figura 5.3. Código HPGL para usinar a cavidade 3.

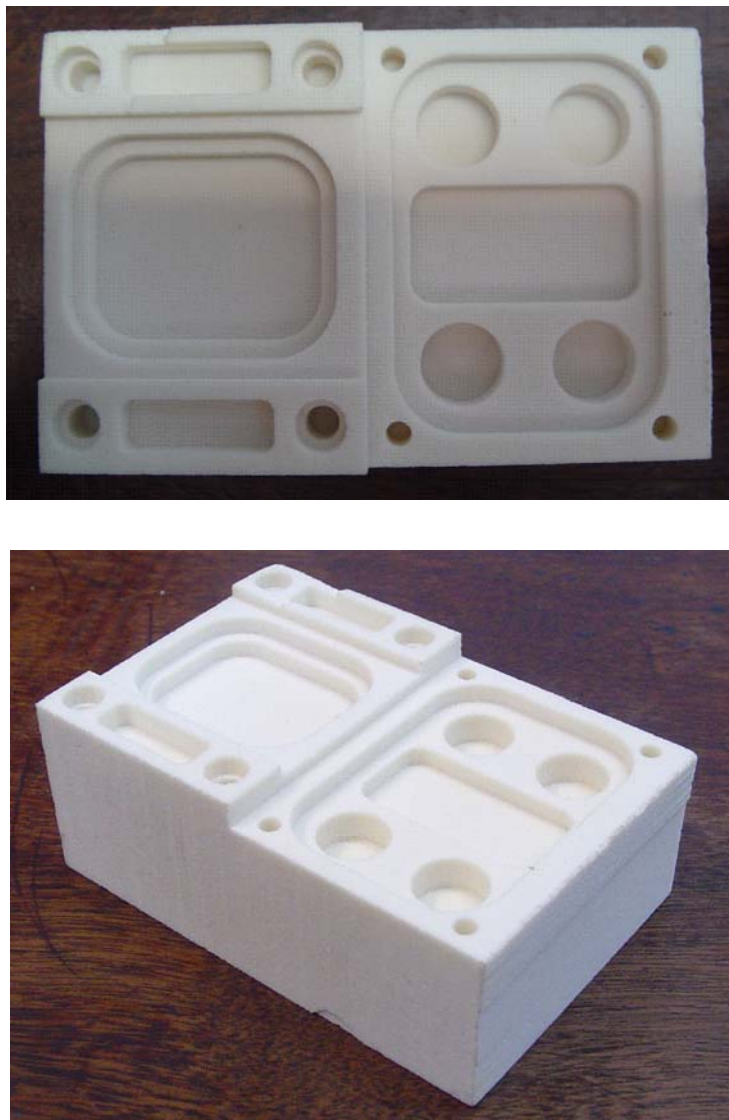


Figura 5.4. Peça 1 usinada na máquina de prototipagem rápida Roland MDX-20.

O tempo necessário para a usinagem desta peça foi de nove minutos e quarenta e sete segundos. Este tempo é somente da usinagem, sem levar em consideração o tempo de preparação (tempo de setup).

5.3 Peça 2

5.3.1 Modelagem da peça 2 no aplicativo CAD por *features*

A modelagem feita no aplicativo de CAD é apresentado na figura 5.5.

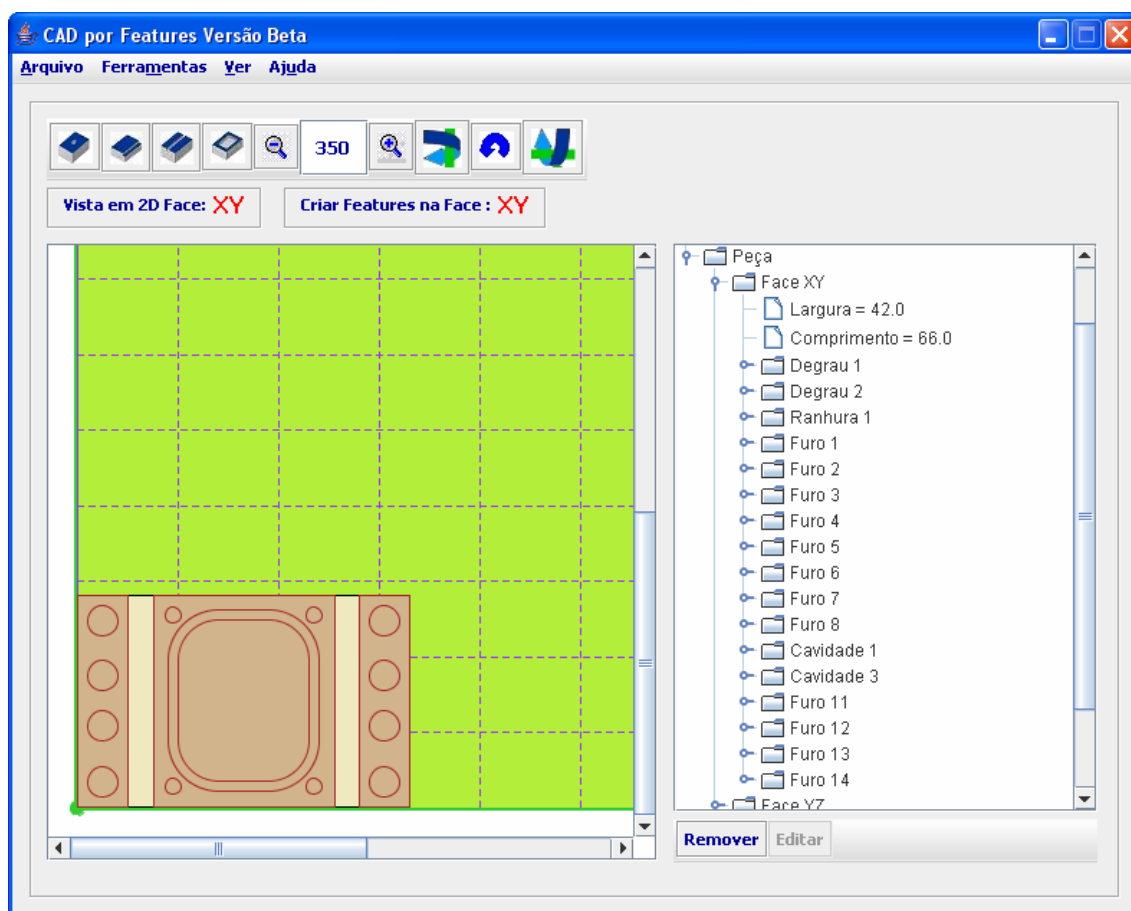


Figura 5.5. Peça 2 modelada no aplicativo CAD por *features*.

Esta peça contém 17 *features* introduzidas somente numa face (XY). Foram colocados furos um ao lado do outro na base de um degrau para testar o algoritmo da mínima distância.

5.3.2 Módulo CAPP

A seqüência na qual as *features* foram inseridas aparece na parte direita do módulo CAD (figura 5.5), e na tabela 5.4 aparecem as *features* que compõem a peça 2.

Tabela 5.4. *Features* presentes na peça 2 (vetor de *features*).

N de ordem	1	2	3	4	5	6
<i>feature</i>	degrau 1	degrau 2	ranhura 1	furo 1	furo 2	furo 3
N de ordem	7	8	9	10	11	12
<i>feature</i>	furo 4	furo 5	furo 6	furo 7	furo 8	cavidade 1
N de ordem	13	14	15	16	17	18
<i>feature</i>	cavidade 3	furo 11	furo 12	furo 13	furo 14	-

Na figura 5.6 pode-se ver a peça 2 com a especificação de quais são as *features* a serem fabricadas, e entre parêntesis aparece a seqüência na qual as *features* foram inseridas pelo cliente.

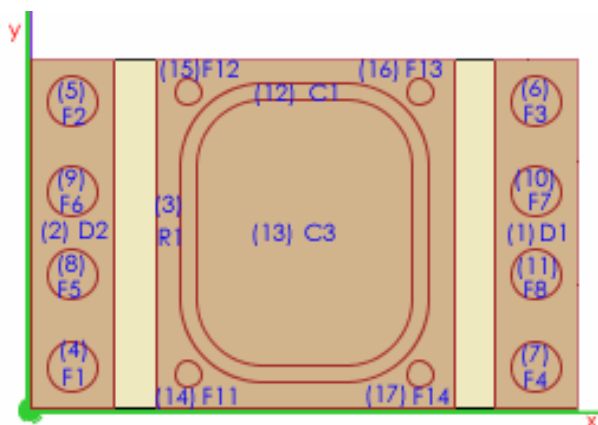


Figura 5.6. *Features* presentes na peça 2 e sua seqüência no vetor de *features*.

Como pode ver-se na tabela 5.5, para o conjunto de furos F1, F2, F3, F4, F5, F6, F7 e F8, contidas dentro da camada 2, que foram inseridos aleatoriamente (a ordem aparece entre parêntesis com letras azuis nas figura 5.6 e 5.7), o ordenador gerou a seqüência apresentada tanto na figura 5.7 (a seqüência de usinagem aparece em números vermelhos) quanto na tabela 5.5 na coluna “ORDENADOR” (para a camada 2). Esta seqüência é coerente com o critério da mínima distancia.

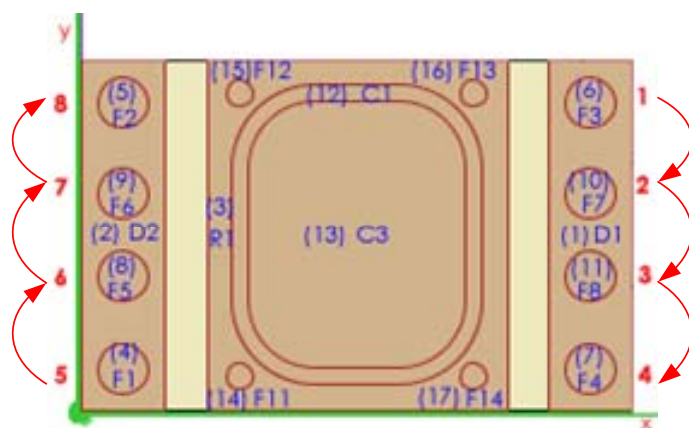


Figura 5.7. Seqüência de usinagem para os furos da camada 2.

As tarefas do módulo CAPP, como apresentadas no capítulo 4 são em primeiro lugar o mapeamento das *features* e a sua transformação em *workingsteps*. A tabela 5.5

apresenta os resultados das etapas de geração dos *Workingsteps*, bem como o seu seqüenciamento.

Tabela 5.5. Etapas executadas visando seqüenciar os *Workingsteps* para a peça 2.

AGRUPADOR DE WORKINGSTEPS				ORDENADOR
Faces	Camadas	Workingsteps	Features	Seqüência
Face XY (possui 4 camadas)	camada 0 (possui 3 workingsteps)	workingstep 0	Degrau (1)	Degrau (2)
		workingstep 1	Degrau (2)	Ranhura (1)
		workingstep 2	Ranhura (1)	Degrau (1)
	camada 1(possui 5 workingsteps)	workingstep 0	Cavidade (1)	Furo (14)
		workingstep 1	Furo (11)	Cavidade (1)
		workingstep 2	Furo (12)	Furo (11)
		workingstep 3	Furo (13)	Furo (12)
	camada 2 (possui 8 workingsteps)	workingstep 4	Furo (14)	Furo (13)
		workingstep 0	Furo (1)	Furo (3)
		workingstep 1	Furo (2)	Furo (7)
		workingstep 2	Furo (3)	Furo (8)
		workingstep 3	Furo (4)	Furo (4)
		workingstep 4	Furo (5)	Furo (1)
	camada 3 (possui 1 workingsteps)	workingstep 5	Furo (6)	Furo (5)
		workingstep 6	Furo (7)	Furo (6)
			workingstep 7	Furo (8)
		workingstep 0	Cavidade (3)	Cavidade (3)
Face YZ	-	-	-	-
Face XZ	-	-	-	-
Face YX	-	-	-	-
Face ZY	-	-	-	-
Face ZX	-	-	-	-

5.3.3 Módulo CAM

A figura 5.8 apresenta um trecho do código HPGL gerado pelo módulo CAM para a peça 2.


```

IN; --→ iniciar o programa
      (camada (0))
Z 0.0, 0.0, -1450.0;
Z 0.0, 0.0, -1570.0;
Z 0.0, 0.0, -1650.0;
Z 0.0, 1680.0, -1650.0;
Z 90.0, 1680.0, -1650.0;
Z 90.0, 0.0, -1650.0;
Z 180.0, 0.0, -1650.0;
Z 180.0, 1680.0, -1650.0;
Z 270.0, 1680.0, -1650.0;
Z 270.0, 0.0, -1650.0;
Z 340.0, 0.0, -1650.0;
Z 340.0, 1680.0, -1650.0;
Z 340.0, 1680.0, -1690.0;
Z 340.0, 0.0, -1690.0;
Z 250.0, 0.0, -1690.0;
Z 250.0, 1680.0, -1690.0;
Z 160.0, 1680.0, -1690.0;
Z 160.0, 0.0, -1690.0;
Z 70.0, 0.0, -1690.0;
Z 70.0, 1680.0, -1690.0;
Z 0.0, 1680.0, -1690.0;
Z 0.0, 0.0, -1690.0;
Z 0.0, 0.0, -1450.0;
Z 660.0, 0.0, -1450.0;
Z 660.0, 0.0, -1570.0;
Z 660.0, 0.0, -1650.0;
Z 660.0, 1680.0, -1650.0;
Z 750.0, 1680.0, -1650.0;
Z 750.0, 0.0, -1650.0;
Z 840.0, 0.0, -1650.0;
Z 840.0, 1680.0, -1650.0;
Z 930.0, 1680.0, -1650.0;
Z 930.0, 0.0, -1650.0;
Z 1020.0, 0.0, -1650.0;
Z 1020.0, 1680.0, -1650.0;
Z 1110.0, 1680.0, -1650.0;
Z 1110.0, 0.0, -1650.0;
Z 1200.0, 0.0, -1650.0;
Z 1200.0, 1680.0, -1650.0;
Z 1290.0, 1680.0, -1650.0;
Z 1290.0, 0.0, -1650.0;
      Z 1380.0, 0.0, -1650.0;
      Z 1380.0, 1680.0, -1650.0;
      Z 1470.0, 1680.0, -1650.0;
      Z 1470.0, 0.0, -1650.0;
      Z 1560.0, 0.0, -1650.0;
      Z 1560.0, 1680.0, -1650.0;
      Z 1650.0, 1680.0, -1650.0;
      Z 1650.0, 0.0, -1650.0;
      Z 1740.0, 0.0, -1650.0;
      Z 1740.0, 1680.0, -1650.0;
      Z 1830.0, 1680.0, -1650.0;
      Z 1830.0, 0.0, -1650.0;
      Z 1920.0, 0.0, -1650.0;
      Z 1920.0, 1680.0, -1650.0;
      Z 1980.0, 1680.0, -1650.0;
      Z 1980.0, 0.0, -1650.0;
      Z 1980.0, 0.0, -1450.0;
      Z 2300.0, 0.0, -1450.0;
      Z 2300.0, 0.0, -1570.0;
      Z 2300.0, 0.0, -1650.0;
      Z 2300.0, 1680.0, -1650.0;
      Z 2390.0, 1680.0, -1650.0;
      Z 2390.0, 0.0, -1650.0;
      Z 2480.0, 0.0, -1650.0;
      Z 2480.0, 1680.0, -1650.0;
      Z 2570.0, 1680.0, -1650.0;
      Z 2570.0, 0.0, -1650.0;
      Z 2640.0, 0.0, -1650.0;
      Z 2640.0, 1680.0, -1650.0;
      Z 2640.0, 1680.0, -1690.0;
      Z 2640.0, 0.0, -1690.0;
      Z 2550.0, 0.0, -1690.0;
      Z 2550.0, 1680.0, -1690.0;
      Z 2460.0, 1680.0, -1690.0;
      Z 2460.0, 0.0, -1690.0;
      Z 2370.0, 0.0, -1690.0;
      Z 2370.0, 1680.0, -1690.0;
      Z 2300.0, 1680.0, -1690.0;
      Z 2300.0, 0.0, -1690.0;
      Z 2300.0, 0.0, -1450.0;
      ....
H;----→ fim do programa

```

Figura 5.8. código HPGL gerado para a camada zero da face XY

Observando tanto o código da figura 5.8 quanto o da figura 5.3, pode-se ver que essas coordenadas espaciais absolutas são valores grandes, e isto se deve à aplicação de um fator de conversão para a máquina. Todas as coordenadas em milímetros devem ser multiplicadas por 40, o valor de 40 foi tirado do manual de programação da máquina. Assim o valor da primeira coordenada, por exemplo, é (0, 0, -1450.0), que na realidade seria o ponto (0, 0, -36.25). Na figura 5.9 são apresentados os primeiros pontos do código HPGL transformados em milímetros.

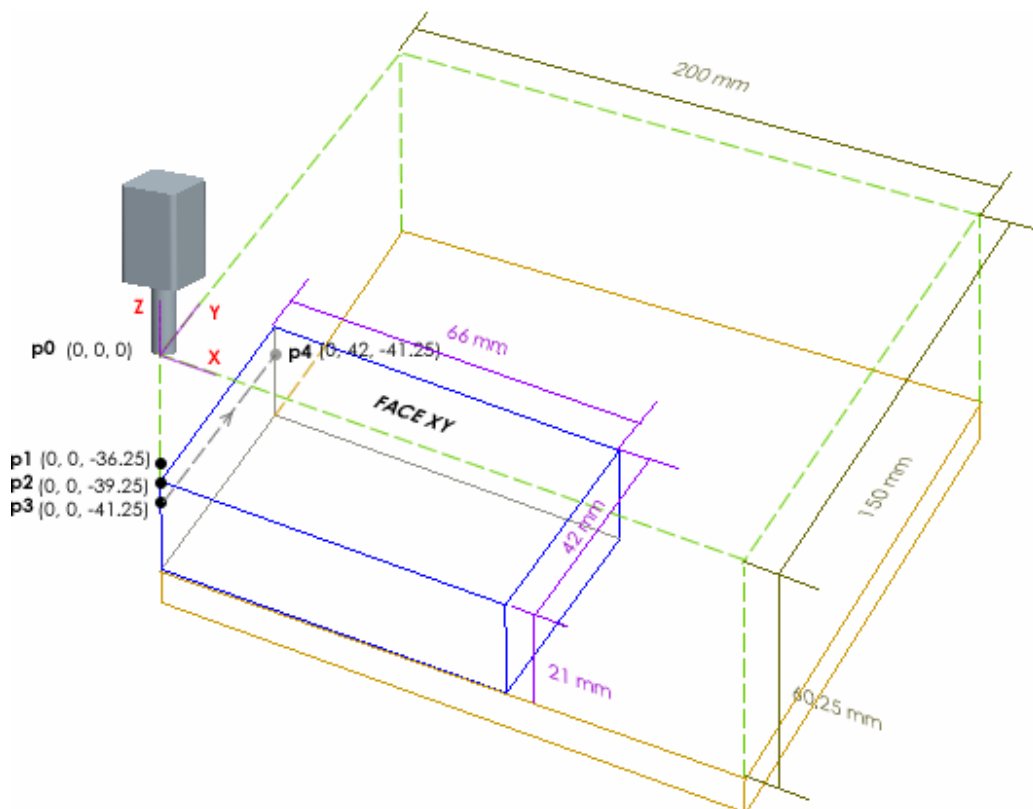


Figura 5.9. Execução do código HPGL para a peça 2.

O primeiro ponto no código HPGL ilustrado na figura 5.8, corresponde a uma posição três milímetros acima da superfície da face XY da peça 2. A ferramenta é movimentada até este ponto, e esta é a condição para a movimentação entre uma *feature* e outra, pois assim se garante que a ferramenta não colide com alguma parte da peça. Ao se movimentar a ferramenta para o segundo ponto, ela fica sobre a superfície da face, coincidindo com a superfície da face. O terceiro ponto já está dentro do bloco (2,00 mm dentro), e a usinagem começa. O quarto ponto corresponde a uma posição 42 mm no eixo “y”, e assim por diante.

O tempo de usinagem para esta peça foi de quatro minutos e trinta e um segundos.

Na figura 5.10 é apresentada a peça 2 depois da usinagem.

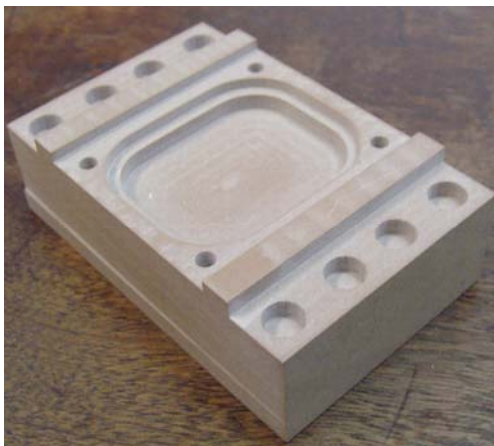


Figura 5.10. Peça 2 fabricada.

5.4 Peça 3

5.4.1 Modelagem da peça 3 no aplicativo CAD por features

Diferentemente das peças anteriores, nesta peça mostra-se o funcionamento do sistema integrado quando se tem features feitas em diferentes faces do bloco. A figura 5.11 apresenta a peça 3 modelada no sistema CAD.

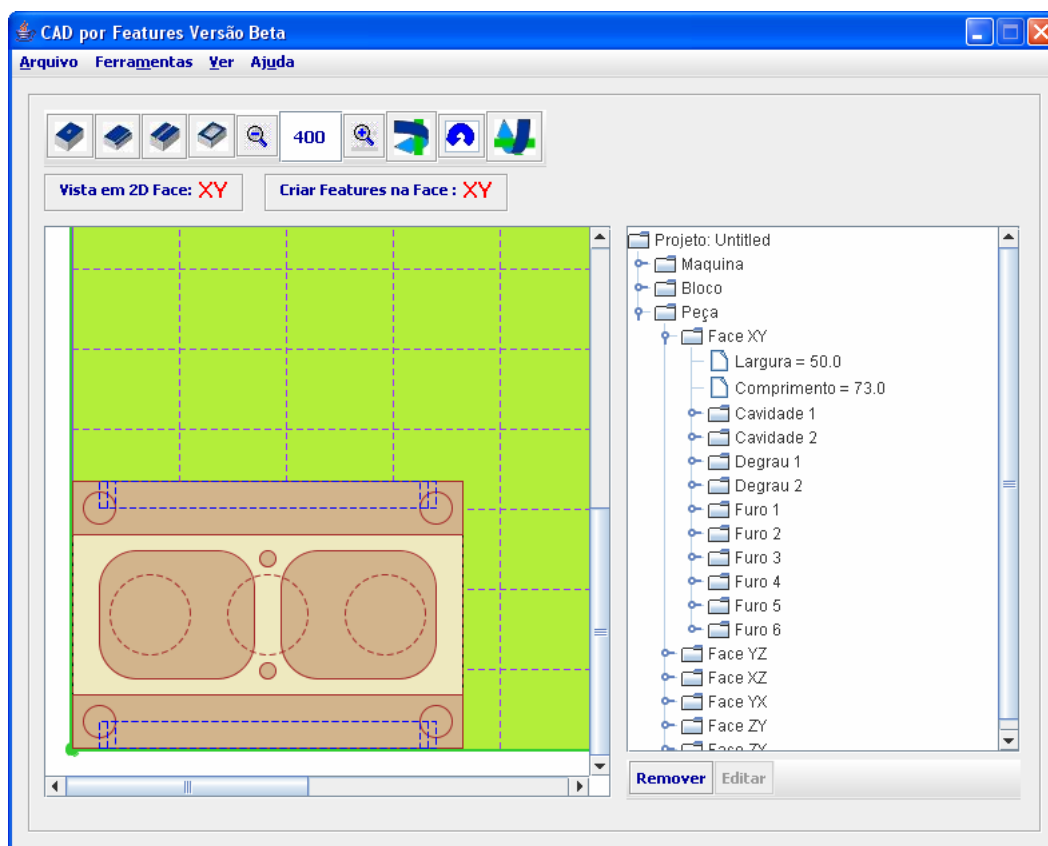


Figura 5.11. Peça 3 modelada no aplicativo CAD.

As linhas azuis pontilhadas representam as *features* feitas nas faces ortogonais, e as linhas pontilhadas em cor marrom representam as *features* feitas na face paralela à face visualizada.

5.4.2 Módulo CAPP

A seqüência original das *features* introduzidas pelo usuário é mostrada na tabela 5.6

Tabela 5.6 *Features* presentes na peça 3 na seqüência em que foram inseridas em cada face.

FACE XY						
N de ordem	1	2	3	4	5	6
<i>feature</i>	Cavidade 1	Cavidade 2	Degrau 1	Degrau 2	Furo 1	Furo 2
N de ordem	7	8	9	10	11	12
<i>feature</i>	Furo 3	Furo 4	Furo 5	Furo 6	-	-
FACE XZ						
N de ordem	1	2	3	4	5	6
<i>feature</i>	Cavidade 1	Cavidade 2	Cavidade 3	Cavidade 4	-	-
FACE YX						
N de ordem	1	2	3	4	5	6
<i>feature</i>	Ranhura 1	Furo 1	Furo 2	Furo 3	-	-
FACE ZX						
N de ordem	1	2	3	4	5	6
<i>feature</i>	Cavidade 1	Cavidade 2	Cavidade 3	Cavidade 4	-	-

Na figura 5.11 são apresentadas as *features* a serem usinadas na face XY para obter a peça 3, e entre parêntesis aparece a seqüência na qual as *features* foram inseridas pelo usuário. Para as outras faces o processo é similar e por tanto não se detalha aqui.

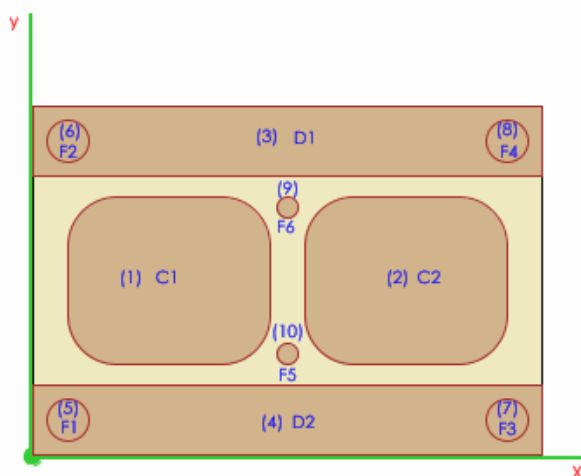


Figura 5.12. *Features* presentes na face XY da peça 3.

O procedimento é similar ao das peças anteriores. Os resultados disto podem ver-se na tabela 5.7.

Tabela 5.7 Agrupador de *workingsteps* e as seqüências de usinagem para a peça 3.

AGRUPADOR DE WORKINGSTEPS				ORDENADOR
Faces	Camadas	Workingsteps	Features	Seqüência
Face XY (possui 2 camadas)	camada 0 (possui 6 workingsteps)	workingstep 0	Cavidade (1)	Degrau (2)
		workingstep 1	Cavidade (2)	Cavidade (2)
		workingstep 2	Degrau (1)	Furo (5)
		workingstep 3	Degrau (2)	Cavidade (1)
		workingstep 4	Furo (5)	Furo (6)
		workingstep 5	Furo (6)	Degrau (1)
	camada 1(possui 4 workingsteps)	workingstep 0	Furo (1)	Furo (2)
		workingstep 1	Furo (2)	Furo (1)
		workingstep 2	Furo (3)	Furo (3)
		workingstep 3	Furo (4)	Furo (4)
Face YZ (possui 0 camadas)	-	-	-	-
Face XZ (possui 1 camadas)	camada 0 (possui 4 workingsteps)	workingstep 0	Cavidade (1)	Cavidade (1)
		workingstep 1	Cavidade (2)	Cavidade (2)
		workingstep 2	Cavidade (3)	Cavidade (3)
		workingstep 3	Cavidade (4)	Cavidade (4)
Face YX (possui 2 camadas)	camada 0 (possui 1 workingsteps)	workingstep 0	Ranhura (1)	Ranhura (1)
	camada 1(possui 3 workingsteps)	workingstep 0	Furo (2)	Furo (3)
		workingstep 1	Furo (3)	Furo (2)
		workingstep 2	Furo (4)	Furo (4)
Face ZY (possui 0 camadas)	-	-	-	-
Face ZX (possui 2 camadas)	camada 0 (possui 6 workingsteps)	workingstep 0	Cavidade (1)	Cavidade (1)
		workingstep 1	Cavidade (2)	Cavidade (2)
		workingstep 2	Cavidade (3)	Cavidade (3)
		workingstep 3	Cavidade (4)	Cavidade (4)

As condições de usinagem para este caso foram modificadas, isto por causa do material, que é apresenta menor resistência mecânica, e para que o tempo da usinagem seja menor. Utilizou-se uma profundidade de corte de 3 milímetros para a peça 3, enquanto para as peças 1 e 2 a profundidade de corte foi de 2 milímetros.

Com base na seqüência de *Workingsteps* apresentada na tabela 5.7, o código HPGL foi gerado e enviado para a máquina de prototipagem rápida. Os resultados da usinagem para a peça 3 podem ser vistos na figura 5.13.

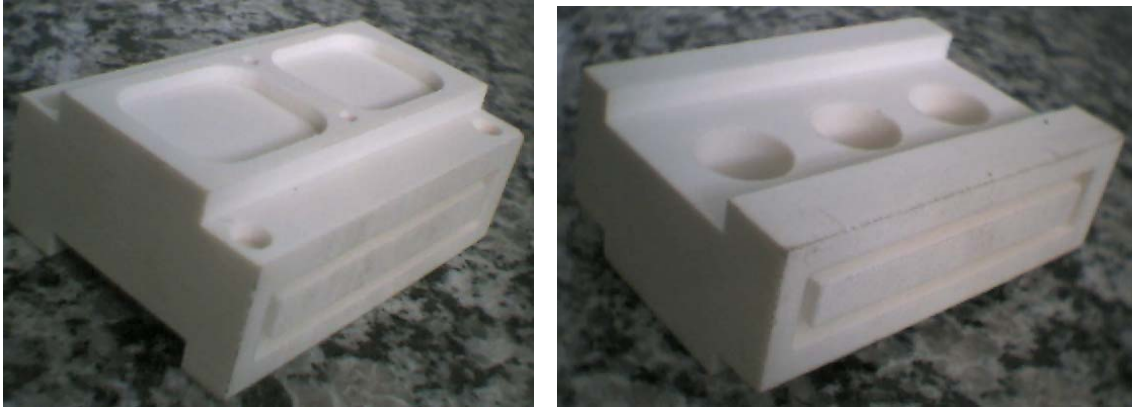


Figura 5.13. Peça 3 usinada.

O tempo necessário para a usinagem desta peça é:

Para a face XY → três minutos e vinte e seis segundos.

Para a face XZ → trinta e cinco segundos.

Para a face YX → três minutos e cinqüenta segundos.

Para a face ZX → trinta e cinco segundos.

CAPÍTULO 6

CONCLUSÕES

Este capítulo apresenta as conclusões do presente trabalho de Mestrado, e os resultados obtidos na utilização do sistema serão contrastados com os objetivos propostos. Também serão apresentadas algumas sugestões para o desenvolvimento de futuros trabalhos relativos ao tema e que venham a complementar o presente trabalho de Mestrado.

6.1 Contribuições do presente trabalho

Os módulos desenvolvidos poderão ser utilizados tanto na indústria, quanto para fins de ensino para disseminar de maneira prática os conceitos de CAD, CAPP, CAM e Internet, sem que obrigatoriamente as instituições de ensino possuam os equipamentos necessários como a máquina de prototipagem rápida.

O sistema integrado CAD/CAPP/CAM apresentado neste trabalho utiliza a Internet como um meio de comunicação, oferecendo desta maneira novas ferramentas de desenvolvimento de futuros sistemas. Além disso, a linguagem de programação utilizada (isto é, Java) possibilita a independência de plataforma computacional do cliente remoto.

À diferença do software da máquina (Modela Player), o sistema computacional desenvolvido pode ser executado via internet, é aberto e é baseado em *features*, pode ser expandido para outras máquinas para fabricação de peças prismáticas, enquanto o Modela Player é uma “caixa preta” não permitindo a utilização a través da internet, nem sua expansão.

A abordagem por *features* é fundamental, pois desta maneira foi possível a integração dos módulos CAD, CAPP e CAM, já que as informações relativas às geometrias a serem removidas ficam disponíveis para todos os módulos.

Este sistema é capaz de trabalhar com máquinas de prototipagem rápida por usinagem de três eixos de diferentes dimensões, e também com máquinas que tenham como linguagem de programação o código HPGL. Comparando-se com outros sistemas

semelhantes encontrados na literatura, o sistema desenvolvido neste trabalho é o único que efetua a geração do código HPGL.

6.2 Disponibilização e implementação do sistema computacional na Internet

O sistema computacional está disponível para os usuários remotos através da Internet acessando a URL <http://grima.ufsc.br/~jticona/CAD>. A figura 6.1 mostra a página inicial do sistema disponibilizado na Internet.



Figura 6.1. Página *web* que disponibiliza o sistema computacional.

A seguir são apresentadas as conclusões em relação a cada módulo desenvolvido.

6.2.1 CAD por *features*

A *applet* disponibilizada na Internet é composta pelos três módulos do sistema integrado, isto é, módulos CAD, CAPP e CAM. A implementação do módulo CAD foi

feita utilizando a abordagem de *features* de projeto para peças prismáticas, e dentre as características e funcionalidades oferecidas incluem-se as seguintes:

- Ele permite o projeto via Internet de peças prismáticas, possibilitando que qualquer pessoa que tenha acesso à Internet se torne um potencial usuário do sistema;
- Não é necessária a instalação de nenhum software complementar (além do Java) para a modelagem da peça, pois ele foi desenvolvido integralmente na linguagem Java.
- O uso de *features* permite a execução do projeto da peça em um nível alto, facilitando a identificação das operações a serem executadas sobre a peça.
- O módulo CAD é integrado internamente com os módulos CAPP e CAM.

6.2.2 Módulo CAPP

A atividade do planejamento do processo assistido por computador foi implementada com base do mapeamento das *features* do projeto e a sua transformação em *workingsteps*. Este sistema CAPP tem as seguintes características:

- O módulo CAPP gera automaticamente o plano do processo, contendo todas as informações necessárias para a usinagem da peça. Assim, o sistema CAPP desenvolvido tem abordagem generativa.
- Em sua versão atual, este módulo está restringido à utilização de somente um tipo de ferramenta. No caso de utilização de uma máquina que permita a troca automática de ferramenta, o módulo CAPP poderá ser alterado no futuro para esta nova condição
- Para gerar as seqüências de usinagem, este módulo agrupa as *features* de uma face em camadas, e este procedimento confere uma significativa organização das *features*, e da seqüência de usinagem para a sua fabricação.

6.2.3 Módulo CAM

Diferentemente de outros sistemas de manufatura remota que em geral sintetizam o programa NC (código “G”) para a usinagem da peça, o módulo CAM desenvolvido

neste trabalho gera o código em linguagem HPGL para uma máquina de prototipagem rápida por usinagem.

6.2.4 Banco de dados

- O banco de dados que foi desenvolvido, suporta os requerimentos de ordens de fabricação, assim como permite ao usuário salvar projetos de peças dentro do mesmo, para poder enviar a ordem de fabricação quando ele o desejar.
- As ordens de fabricação são colocadas em uma fila, e as peças são usinadas de acordo com a ordem de chegada das ordens de fabricação.

6.3 Sugestões para futuros trabalhos

Com relação ao módulo CAD, este aplicativo poderia ser estendido para a criação de outras *features* como cavidades abertas, cavidades com raios de concordância diferentes, *features* com profundidade variável, ranhuras e degraus com larguras variáveis, etc.

As ferramentas gráficas para o projetista podem ser melhoradas. Sugere-se criar uma representação visual da peça em 3D utilizando Java 3D ou VRML, ou outra linguagem voltada para a Internet.

Com relação ao módulo CAPP, futuros desenvolvimentos poderiam incluir a transformação das *features* de projeto em *features* compostas de usinagem. Por exemplo, na figura 6.2 se apresenta três furos concêntricos aninhados um dentro de outro (figura 6.2a), e estas três *features* poderiam ser transformadas em somente uma, reduzindo o número de *features* do vetor de *features* da face em questão (figura 6.2b).

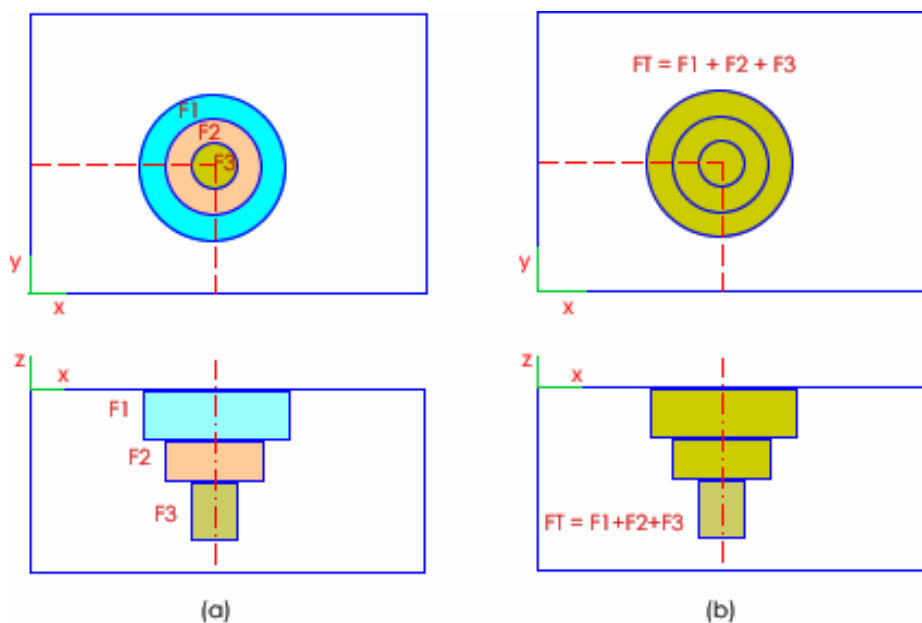


Figura 6.2 Transformação de *features* de projeto em *features* compostas de usinagem

O sistema atualmente considera apenas operações de desbaste. Sugere-se o desenvolvimento de operações de acabamento (e as suas respectivas ferramentas).

Em relação ao módulo CAM, sugere-se o desenvolvimento de um aplicativo para o envio das imagens da usinagem da peça em tempo real para o *browser* do usuário remoto via Internet. Isto tornaria mais interessante o ensino de temas relacionados à manufatura remota, CAD, CAPP, CAM, e prototipagem rápida.

Sugere-se estender este trabalho para gerar códigos para máquinas de comando numérico de diferentes fabricantes, cuja linguagem em geral segue a ISO 6983.

Finalmente, pode-se dizer que o objetivo proposto para este trabalho, que consistiu no desenvolvimento de um sistema integrado para a manufatura remota de peças prismáticas, considera-se que o mesmo foi atingido, pois este trabalho abrange fases importantes ao longo do ciclo de desenvolvimento de um produto (neste caso peças prismáticas), e o mesmo está disponível para ser utilizado por qualquer pessoa (que tenha acesso à internet) localizada em qualquer lugar do mundo.

REFERÊNCIAS BIBLIOGRÁFICAS

AARNO, D. EKVALL, S. KRAGIC, D. Adaptive Virtual Fixtures for Machine-Assisted Teleoperation Tasks, Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference, Stockholm, Sweden, p 1139-1144, 18-22, April 2005.

AHN, S., SEQUIN C. e WRIGHT. P., Internet-Based Design an Manufacturing, Final Report 1988-99 for micro Project, 98-136, TRW, 1999.

AHN S. H. et al. CyberCut: An Internet-based CAD/CAM System, Journal of Computing and Information Science in Engeneering, California, v1, p. 52 – 59, marzo de 2001.

ALTING, L; ZHANG, H. C. Computer Aided Process Planning: the State-of-the-Art Survey. International Journal of Production Research, v.27, n.4, p.553-585, 1989.

ÁLVARES, A. J., Uma metodologia para integração CAD/CAPP/CAM voltada para manufatura remota de peças rotacionais baseada na internet, 2005, 249f, Tese (Tese em Engenharia Mecânica) – Faculdade de Engenharia Mecânica, Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina.

ÁLVARES, A. J, FERREIRA, J. C. E., SHIMABUKURO, L. e FERNANDES, C. A.,Um sistema de telemanufatura baseado na web orientado ao processo de oxicorte, XXII ENEGEP, Curitiba, 2002.

ÁLVARES A. J, FERREIRA J. C. E. Uma metodologia para integração CAD/CAPP/CAM voltada para manufatura remota de peças rotacionais via Web, 2 Congresso de Engenharia de Fabricação, Uberlândia, Minas Gerais, Maio de 2003.

ALVES M. M., Ferramentas Virtuais na Educação Tecnológica a Distância : o caso dos Laboratórios virtuais e *softwares* de simulação, Dissertação de mestrado em Engenharia da Produção, UFSC, Florianópolis, 2001.

BIDARRA R., VAN DEN BERG E. and BRONSVOORT W. F. Interactive Facilities for Collaborative Feature Modeling on the Web, Proc. of the Tenth Portuguese Conference on Computer Graphics, Lisbon, Portugal, pp. 43 - 52 2001.

BRENNER, W., ZARNEKOW, R. e WITTIG, H., *Intelligent Software Agents*, Springer, 1998.

BROWN, S. M. e WRIGHT, P. K., 1998, A Progress Report on the Manufacturing Analysis Service, an Internet-Based Reference Tool, *Journal of Manufacturing Systems*, Vol. 17, No. 5, pp.389-401.

CHANG, T.C., Wysk R.A. e Wang, H.P., 1998, *Computer Aided Manufacturing*, Prentice Hall International Series in Industrial and Systems Engineering, W.J. Fabrycky e J.H. Mize (eds.), 2ns Edition, 1998.

CHU X. N., TSO S. K. e TU Y. L. A Novel Methodology for Computer-Aided Process Planning, *The international Journal of Advanced Manufacturing Technology*, Springer-Verlag London, vol 16, p. 714–719, 2000.

DA COSTA D. D, PEREIRA A. G., Desenvolvimento e avaliação de uma tecnologia de baixo custo para programação CNC em pequenas empresas, **Produção**, Paraná, Brasil, v. 16, n, 1, p 048-063, jan/abr 2006.

DETAND J., A Computer Aided Process Planning System Generating non-linear Process Plans, Catholic University of Leuven, Belgium, PhD thesis, 1993.

DORNFELD D, WRIGHT P. K, ROUNDY S, RANGARAJAN A, and AHN S. Agent Interaction in CAD/CAM, *Transactions of NAMRI/SME*, Volume XXIX, p 569 – 575, 2001.

FERREIRA, J.C.E., *Planejamento do Processo Assistido por Computador*, Apostila, Pós-Graduação em Engenharia Mecânica, Universidade Federal de Santa Catarina, 1996.

FERREIRA A. C. e LAFRATTA, F. H. “Conheça Alguns Meios para a Obtenção de Protótipos de Peças Injetadas”, *Plástico industrial*, Aranda Editora Técnica Ltda São Paulo, Brasil, setembro,1998.

FOX, S.M. Research in Enterprise Integration, *Proceedings Artificial Intelligence and Manufacturing Research Planning Workshop*, pp. 40-52,ISBN1-57735-003-0, USA, June 1996.

GONÇALVES E., Dominando o Eclipse, Ciência Moderna, Rio de Janeiro, 2006.

GORNI A. A, Introdução à prototipagem rápida e seus processos, Plástico Industrial, março 2001, págs 230 – 239.

GROOVER, M. Automation, Production System, and Computer Integrated Manufacturing, Prentice Hall, 1987.

GRUNINGER, M.; PINTO, J. A theory of Complex Actions for Enterprise Modelling. *Working Notes AAAI Spring Symposium Series 1995*, Extending Theory of Action: Formal Theory and Practical Applications, Stanford, <http://www.ie.utoronto.ca/EIL/public/complex.ps>, 1995.

HALL, B. B. Beej's guide to network programming using internet sockets. Disponível na Internet via WWW. URL: <http://www.eest.esuchico.edu/~beej/guide/net>, 1999.

HORSTMANN C. e CORNELL G., Core Java – fundamentos, Markon books, São Paulo, 2001.

JACOBS, P.F., “Rapid Prototyping & Manufacturing, Fundamentals of Stereolithography”, Society of Manufacturing Engineers, USA, 1992.

KERRY, H. T, Planejamento de processo automático para peças paramétricas. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo. (Disponível na biblioteca da EESC - USP), 1997.

KUSIAK A., Concurrent Engineering: Automation, Tools and Techniques, John Wiley & Sons, 2004.

LEE, J., E-manufacturing, fundamental, tools, and transformation, Robotics and Computer-Integrated Manufacturing, Volume 19, Issue 6, Pages 501-507, December 2003.

MALEK, L. A., WOLF, C. e GUYOT, P. D., Telemanufacturing: A Flexible Manufacturing Solution, Int. J. Production Economics, Vol. 56-57, pp. 1-12, 1998.

MICHEL B. A, Método de representação de processos em forma de fluxo – IDEF0, Apostila de Modelagem de Informações e Fluxo de Processos, Universidade de Caxias do Sul, 2002.

MySQL, "MySQL", <http://www.mysql.com>, último acesso em 1/5/2006.

NOF, S. Y, Handbook of Industrial Robotics, John Wiley Sons, 2 ed, Canada, 1999.

NOF, S. Y., Collaborative E-Work And E-Mfg.: The State Of The Art And Challenges For Production And Logistics Managers, International Federation of Automatic Control 11th IFAC Symposium on Information Control Problems in Manufacturing (INCOM), Salvador, Brazil, April 5-7th, 2004.

PESHKIN M. A, COLGATE J. E, WANNASUPHOPRASIT W, MOORE C, GILLESPIE R. B, and P. AKELLA, "Cobot architecture," IEEE Transactions on Robotics and Automation, vol. 17, no. 4, pp. 377–390, 2001.

REZENDE D. F, Planejamento de processos de fabricação assistido por computador através de um sistema especialista baseado na tecnologia de features: Um modelo de desenvolvimento voltado para a realidade industrial, Dissertação (dissertação de mestrado em Engenharia Mecânica) – Faculdade de Engenharia Mecânica, Universidade Federal de Santa Catarina, 189 f. Florianópolis, Santa Catarina, 1996.

ROBERTSON D. C, and ALLEN T. J, Managing CAD Systems in Mechanical Design Engineering, IEEE transactions on engineering management, vol. 39, no. 1, p. 22 - 31, february 1992.

ROCKWELL Automation, Making Sense of e-Manufacturing: A Roadmap for Manufacturers , Industry White Paper, 2002.

SALEH M. A, Development of a step feature-based intelligent process planning system for prismatic parts, Thesis, Mechanical Engineering Department - Middle East Technical University, Ankara, Turkey, 2005.

Scientific American Frontiers, 21st Century Medicine: Cyber surgery, http://www.pbs.org/safarchive/4_class/45_pguides/pguide_605/4565_cyber.html, 2000.

SHAH, J. J., MÄNTYLÄ, M., Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications, John Wiley Sons Inc, New York, 1995.

SHERIDAN, T. "Supervisory control of remote manipulators, vehicles and dynamic processes: experiment in command and display aiding", in Advances in Man Machine Systems Research, Vol 1, 49-137, 1984.

SHERIDAN, T. B. Telerobotics, Automation and Human Supervisory Control. MIT Press, 1992.

SMITH, C. S., WRIGHT, P. K., Cybercut: A World Wide Web Based Design to Fabrication Tool, 2000, <http://cybercut.berkeley.edu>.

TAYLOR R. H, and STOIANOVICI D, "Medical robotics in computer integrated surgery," in IEEE Transactions on Robotics and Automation, vol. 19, pp. 765 – 781, 2003.

TOURINO, S. G, Guiagem de Robô Móvel XR4000 para a Inspeção de Tubulações Industriais Soldadas via Internet, Projeto de conclusão de curso, GRACO, UnB, Brasília, 2000.

VAN DEN BERG E, BIDARRA R, BRONSVOORT W. F. Web-based interaction on feature models, From Geometric Modeling to Shape Modeling, Cugini U and Wozny M (eds). Kluwer Academic Publishers, Dordrecht, pp. 99-112, 2001.

WANG H. P. e LI, J. K. Computer-Aided Process Planing, Advances in Industrial Engineering, Vol. 13, Elseiver, 1991.

WANG, F. C. e WRIGHT, P. K., Web-Based CAD Tools for a Networked Manufacturing Service. Proceedings of DET'98 ASME Design Engineering Technical Conference, Atlanta, Georgia, CIE-5517, 1998.

WARD MICAHEL K. The Microstation 95 Workbook for Windows, Stipes Pub Llc, june, 1996.

XU X. W, and HE Q, Striving for a total integration of CAD, CAPP, CAM and CNC, Robotics and Computer-Integrated Manufacturing, Volume 20, Issue 2, Pages 101-109, April 2004.

ZHANG L, DENG J, and CHAN S. C. F, A Next Generation NC Machining System Based on NC Feature Unit and Real-Time Tool-Path Generation, The international Journal of Advanced Manufacturing Technology, Springer-Verlag London, v. 16, n 12, p. 889-901, 2000.

ZOTTO, OZIR F. A, Protocolo HTTP (HyperText Transfer Protocol), Bate Byte, Curitiba. PR, v. 57, p. 13, setembro de 1996.