

# RECONHECIMENTO DE FEATURES EM PEÇAS ROTACIONAIS MODELADAS POR SÓLIDOS NUM SISTEMA CAD/CAM VOLTADO PARA A INTERNET

**João Carlos Espíndola Ferreira**

Universidade Federal de Santa Catarina, Departamento de Engenharia Mecânica  
GRIMA/GRUCON, Caixa Postal 476, 88040-900, Florianópolis, SC, Brasil. jcarlos@emc.ufsc.br.

**Darlan Vivian**

Universidade Federal de Santa Catarina, Departamento de Engenharia Mecânica  
GRIMA/GRUCON, Caixa Postal 476, 88040-900, Florianópolis, SC, Brasil. darlan@inf.ufsc.br.

**Resumo.** Nos últimos anos cada vez mais pessoas têm tido acesso à Internet, a qual tem reduzido ou até eliminado as distâncias entre pessoas e/ou grupos localizados fisicamente distantes entre si. A Internet tem também possibilitado o desenvolvimento de aplicações para o suporte às atividades de projeto e manufatura de peças industriais. Neste trabalho será apresentado um sistema para o reconhecimento de features em peças rotacionais criadas através do modelador de sólidos ACIS via Internet. Na etapa de projeto da peça o usuário introduz em seu browser os dados sobre os elementos que compõem a peça. Estes dados são analisados e caso não exista nenhuma inconsistência, são enviados via CGI (Common Gateway Interface) para o software que se encontra no servidor. Para visualizar a peça, o arquivo do sólido no formato SAT (Standard ACIS Text) é transformado para o formato VRML (Virtual Reality Markup Language), e em seguida enviado para o browser do cliente. A peça pode conter cilindros, cones e chanfros. O usuário pode seccionar a peça ao longo do eixo de rotação, e também pode obter o arquivo do sólido criado em formato SAT, caso deseje manusear a peça em seu próprio software de CAD. Terminado o projeto da peça, o software faz um estudo sobre a geometria da peça. Para isso, foi desenvolvido um algoritmo cujo objetivo é reconhecer automaticamente as features, o qual é realizado através da análise da metade do perfil da peça em 2D. Estas features poderão ser usadas no processo de geração do código G para a máquina CNC.

**Palavras-chave:** Projeto Assistido por Computador (CAD), Features, Internet, Modelagem de Sólidos, Peças Axissimétricas.

## 1. INTRODUÇÃO

Atualmente o acesso à tecnologia da Internet tornou-se um fato comum e necessário, o que tem facilitado muito a comunicação entre pessoas e/ou grupos localizados fisicamente distantes entre si. A Internet tem também possibilitado o desenvolvimento de diversas aplicações, e entre elas estão as atividades de projeto e manufatura de peças e produtos industriais. Com o crescimento da popularidade dos navegadores baseados na WWW percebe-se que o ambiente de projeto de peças orientado a rede pode se tornar um novo paradigma.

Dentre estas aplicações, tem-se o Cybercut (<http://cybercut.berkeley.edu>) desenvolvido por Smith e Wright<sup>(1)</sup>, que possibilita a concepção de uma peça prismática que será usinada utilizando-se um sistema CAD/CAM desenvolvido em Java usando *Applets*.

Muitas técnicas têm sido desenvolvidas para a criação de sistemas voltados para a modelagem de peças baseada em *features*. *Features* são definidas por Shah e Mäntylä<sup>(2)</sup> como sendo “formas

genéricas ou características de um produto com as quais um engenheiro pode associar certos atributos e conhecimentos que serão úteis no raciocínio sobre o produto”. Desta forma, as *features* representam um significativo encapsulamento das informações sobre as partes que compõem a geometria do produto, informações estas que serão úteis no processo de manufatura do produto.

Estas técnicas de modelagem baseadas em *features* podem ser divididas, segundo Shah e Mäntylä<sup>(2)</sup>, em duas categorias principais: Projeto por Features e Reconhecimento de Features. Esta segunda categoria pode ser subdividida em outras duas: Reconhecimento Interativo de Features e Reconhecimento Automático de Features. Nesta última categoria um modelo geométrico é criado e posteriormente um programa processa o modelo para encontrar automaticamente as *features*.

Segundo Álvares e Ferreira<sup>(3)</sup>, o reconhecimento automático de *features* tem a vantagem de que o projetista não é obrigado a ter um amplo conhecimento de manufatura. Assim, sobra-lhe mais tempo para estudar a forma da peça desejada e nos seus aspectos funcionais.

Neste contexto, o objetivo deste trabalho é a implementação de um sistema que permita o projeto e reconhecimento de *features* de peças axissimétricas via Internet, independente da plataforma utilizada pelo *browser* do usuário. As *features* da peça poderão ser usadas na determinação do código G apropriado para a máquina CNC que irá usinar a peça. Portanto, a peça poderá ser fabricada e enviada posteriormente para o projetista e desta forma, ele terá um protótipo funcional dentro de poucos dias por um custo menor que a fabricação própria exigiria.

## 2. MÉTODO PARA O RECONHECIMENTO DE FEATURES

O método para o reconhecimento de *features*<sup>1</sup> baseia-se em um algoritmo recursivo que pode ser aplicado em peças que possuam inúmeros diâmetros distintos. Ele pode ser aplicado também em peças escalonadas em ambos os sentidos (isto é, que podem necessitar de duas fixações). A matéria-prima para a criação da peça é considerada como sendo uma barra cilíndrica, com diâmetro igual ao maior diâmetro da peça. O sistema de coordenadas que foi adotado é o XZ (X para o diâmetro e Z para o comprimento), pois é o mesmo utilizado pela máquina CNC.

O método inicia fazendo uma análise da metade do perfil da peça em 2D o qual deverá conter apenas segmentos de retas. Outro requisito do método é que o perfil corresponda a uma *polyline*.

Para obter todos os vértices e arestas (figura 1) que compõem a metade do perfil 2D da peça projetada pelo usuário, primeiramente o sólido é seccionado duas vezes ao longo do seu centro de rotação, e a face que representa o perfil 2D da peça é identificada. Em seguida, os vértices e arestas são identificados e ordenados. Os pontos associados aos vértices são classificados em pontos no contorno (por ex., ponto 12 na figura 1) e em pontos internos (por ex., pontos 4 a 11 na figura 1).

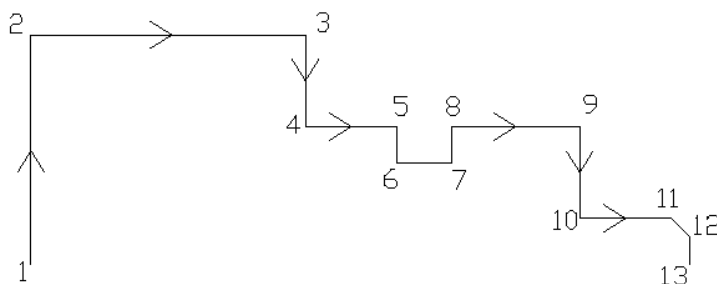


Figura 1. Exemplo de um perfil de uma peça, e os pontos ao longo do perfil

Depois, é realizada a identificação das áreas que serão usadas na operação de varredura (*sweeping*) para a criação dos sólidos no formato de anel, que são adicionados recursivamente à peça original até que ela alcance a forma de uma barra cilíndrica sólida. Estas áreas são representadas por faces, e tem-se como exemplo as faces  $F_1$ ,  $F_2$ ,  $F_3$  e  $F_4$  ilustradas na figura 2. A operação de varredura consiste na criação de um sólido a partir de uma face. Neste artigo, tal

<sup>1</sup> Neste artigo as *features* são representadas por sólidos na forma de anel.

operação é realizada varrendo-se a face 360° em torno do eixo Z da peça, dando origem a um sólido na forma de um anel.

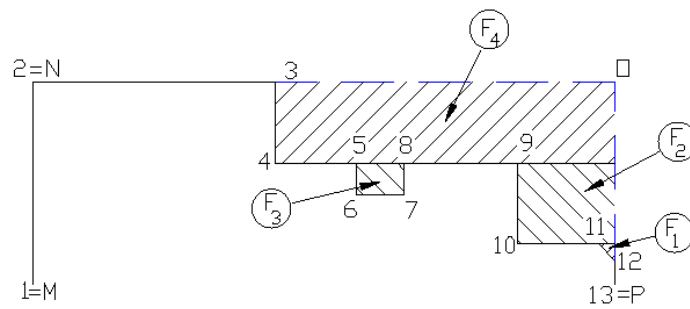


Figura 2. Faces reconhecidas, que sofrerão a operação de varredura (*sweeping*)

Na figura 3 é apresentado um diagrama para facilitar a compreensão de como funciona o algoritmo para o reconhecimento de *features*. O processo tem início quando o usuário finaliza a modelagem da peça (retângulo azul) e é concluído quando não existirem mais pontos internos (retângulo vermelho).

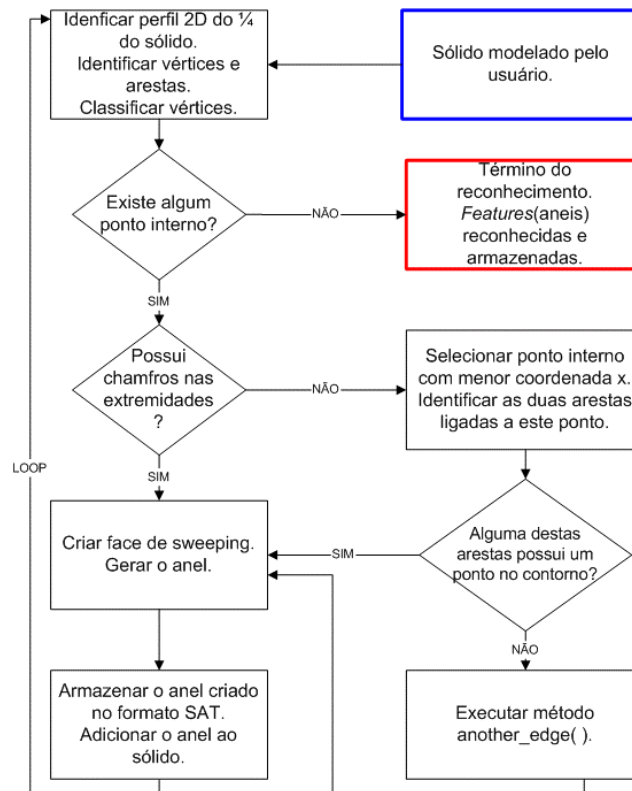


Figura 3. Diagrama do funcionamento do reconhecedor de *features*

### 3. TECNOLOGIAS UTILIZADAS

Para a implementação do sistema para o projeto e reconhecimento de *features* em peças axissimétricas, foi utilizado o *kernel* do modelador de sólidos ACIS. Foram utilizadas também outras tecnologias tais como: JSP (*Java Server Pages*), Java, Java Beans, C++, CGI (*Common Gateway Interface*), VRML (*Virtual Reality Markup Language*) e SSI (*Server Side Includes*).

O modelador de sólidos ACIS ([www.spatial.com](http://www.spatial.com)) detalhado por Corney e Lim<sup>(4)</sup> foi utilizado na construção de classes escritas na linguagem C++ exemplificada por Deitel e Deitel<sup>(5)</sup>, que compõem o programa CGI presente no servidor, e são responsáveis pela geração do sólido e pelo reconhecimento de *features*. O modelador de sólidos ACIS também é escrito em C++ e consiste

num *framework* composto por várias classes e métodos que podem ser usados para criar uma aplicação 3D para o usuário final. Para armazenar um sólido o ACIS utiliza o formato SAT (*Standard ACIS Text*) o qual contém todas as informações necessárias para uma visualização do sólido.

As páginas com os formulários nos quais o usuário define as características do sólido foram desenvolvidas usando JSP<sup>(6)</sup>, tecnologia usada para servir conteúdo dinâmico para o usuário, usando lógica e dados no lado do servidor. A verificação da consistência dos dados que foram passados pelo usuário foi desenvolvida utilizando o conceito de *Java Beans*, que são classes escritas na linguagem Java<sup>(7)</sup> e exemplificada por Deitel e Deitel<sup>(8)</sup>, que seguem um conjunto de convenções simples de projeto e nomenclatura.

A tecnologia CGI foi utilizada no envio dos dados dos formulários para o servidor responsável pela criação do sólido. Um CGI permite interatividade entre um cliente e um servidor através do protocolo HTTP (*Hyper Text Transfer Protocol*). Um programa CGI é executado em tempo real, logo pode apresentar como resultado uma informação dinâmica, como por exemplo o sólido gerado.

A linguagem VRML ([www.vrml.org](http://www.vrml.org)) foi necessária para permitir a visualização 3D do sólido em qualquer *browser*<sup>2</sup>. VRML é uma linguagem independente de plataforma que permite a criação de ambientes virtuais que possibilitam visualizar objetos por ângulos diferentes e interagir com eles.

Finalmente, a ferramenta SSI<sup>(9)</sup> possibilitou a inclusão do sólido em VRML na página de resposta, no formato SHTML, enviada pelo servidor do sistema para o *browser* do usuário. Comandos SSI são ferramentas bastante úteis na criação dinâmica de páginas Web, e são normalmente utilizados como interface para programas CGI.

#### 4. ESTRUTURA DO SISTEMA

A estrutura deste sistema<sup>(10)</sup> é representado na figura 4, a qual é composta por vários arquivos, cada qual responsável por uma etapa da criação do sólido. Além de peças axissimétricas, o usuário pode modelar sólidos simples como paralelepípedo, cilindro, cone, prisma e cunha.

Através da página inicial (*Select\_solido.jsp*), o usuário seleciona o tipo de sólido que deseja modelar. Após a escolha, a opção contendo o tipo de sólido é enviada para a página *Principal.jsp* que, de acordo com a informação recebida, decide qual o formulário que deverá ser mostrado para que o usuário entre com as especificações do sólido.

Então, após a escolha das medidas dos parâmetros do sólido, estas informações são enviadas para a página de verificação dos dados (por ex., arquivo *vCone.jsp* na figura 4). Caso o sólido escolhido seja uma peça axissimétrica, estes dados são enviados para outra página (*ParametrosEixo.jsp*) para obtenção de mais especificações (por ex., presença de chanfros) e depois todos estes dados são enviados para as suas respectivas páginas de verificação.

Quando algum dado estiver incorreto o fluxo do sistema retorna para as páginas de verificação, sendo necessário então que o usuário faça a devida correção. Para facilitar a identificação do erro, a própria página de verificação acusa através de uma mensagem a localização do dado incorreto.

Após verificar-se que os dados estão corretos, os dados são enviados, de acordo com o tipo do sólido, para o programa *Modeling.cgi* presente no servidor o qual é responsável pela criação da peça axissimétrica e reconhecimento das *features*, assim como a criação dos demais sólidos simples.

Terminado o reconhecimento de *features*, a peça axissimétrica modelada que se encontra no formato SAT é convertido para VRML da mesma forma que um outro sólido simples. Feito isto, o servidor envia uma página SHTML para o *browser* do usuário, a qual contém o arquivo no formato VRML para visualização 3D do sólido que foi gerado.

O programa CGI que se encontra no servidor foi criado através da implementação de várias classes na linguagem C++. No diagrama de classes da figura 5, apenas os principais atributos e métodos de cada classe são apresentados. Todas as classes apresentam em sua implementação vários tipos de dados e métodos da API (*Application Program Interface*) da biblioteca ACIS.

---

<sup>2</sup> É necessário que o *browser* possua um visualizador VRML previamente instalado.

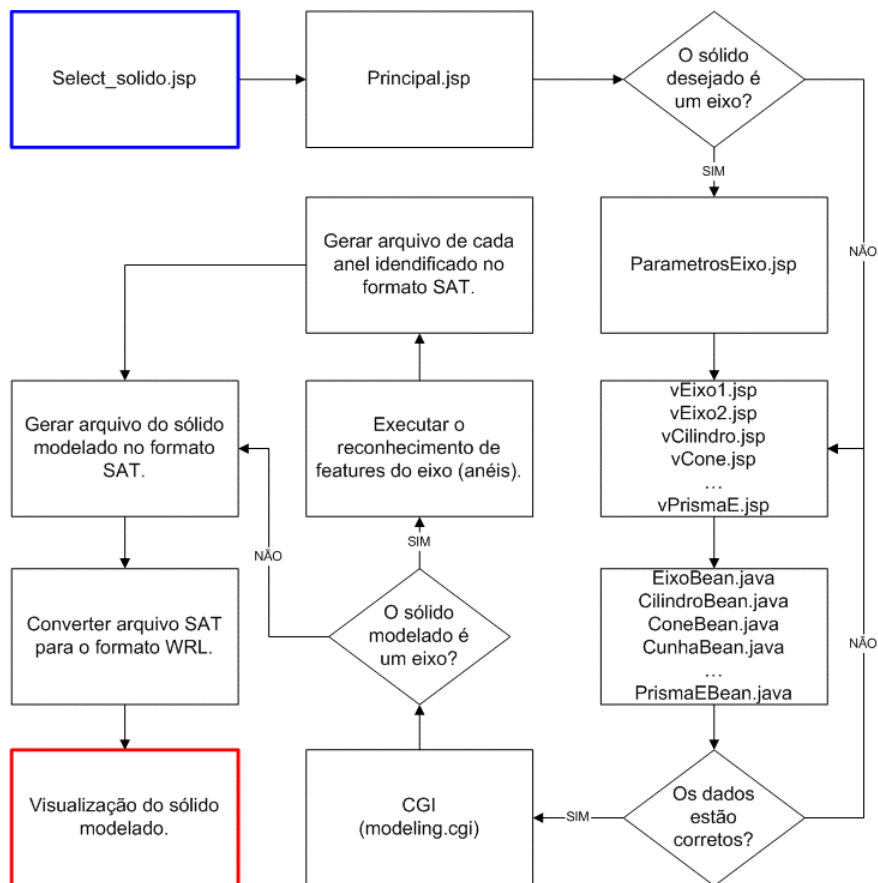


Figura 4. Diagrama da estrutura do software

A classe *Modeling* é a classe principal, a qual verifica que sólido será criado e repassa os dados para a instância da classe adequada. Para que o programa CGI receba e identifique corretamente os dados da peça provenientes da página JSP, foi utilizada na classe *Modeling* a biblioteca *cgiparser* implementada por RudeServer<sup>(11)</sup>, a qual fornece os métodos necessários para tal operação.

A classe *SimpleSolid* é a responsável pela criação de todos os sólidos simples como paralelepípedo e cunha. Ela possui o atributo *solid\_body* que representará o sólido que será gerado através do método *create\_simple\_body()*.

A classe *Axis* é a responsável pela criação da peça axissimétrica. Ela apresenta dois atributos principais, ambos do tipo *BODY*: *axis\_body*, que representa o eixo criado através do método *create\_axis()* e *axis\_body\_sweep*, que representa a peça durante e após a execução do método *execute\_afrc()*. O método *apply\_blends()* é chamado caso o usuário tenha aplicado algum chanfro no eixo e portanto deve ser executado antes do reconhecimento das *features*.

A classe *Save* fornece: (i) o método *save\_ent()* que gera o arquivo no formato SAT (“*Standard ACIS Text*”), e (ii) o método *save\_wrl()* que utiliza as funcionalidades da classe *VrmlMM* implementada por Lapadat<sup>(12)</sup> para converter o arquivo no formato SAT para o formato VRML.

Para seccionar a peça axissimétrica foi criada a classe *Section* a qual possui o método *sectioning\_body()* que permite que o sólido seja seccionado uma ou duas vezes pelo usuário.

Para realizar a operação de varredura, foi criada a classe *Sweep*. A geração das faces para o *sweeping* é realizada pelos métodos *sweep\_three\_ang\_face()* e *sweep\_four\_ang\_face()* os quais criam, respectivamente, uma face a partir de 3 ou 4 vértices. Após a criação da face, o método *apply\_sweep()* é executado para gerar os sólidos na forma de anéis e adicioná-los ao eixo representado pelo atributo *axis\_body\_sweep* da classe *Axis*.

A classe *AFR* é a responsável por grande parte da implementação do reconhecedor de *features*, cujo funcionamento pode ser visualizado através do diagrama da figura 3. Ela possui 3 atributos principais: *face\_afrc* que representa a face atual de *sweeping*, *vertexes\_list* e *edges\_list* que representam, respectivamente, a lista de vértices e a lista de arestas que compõem a *face\_afrc*.

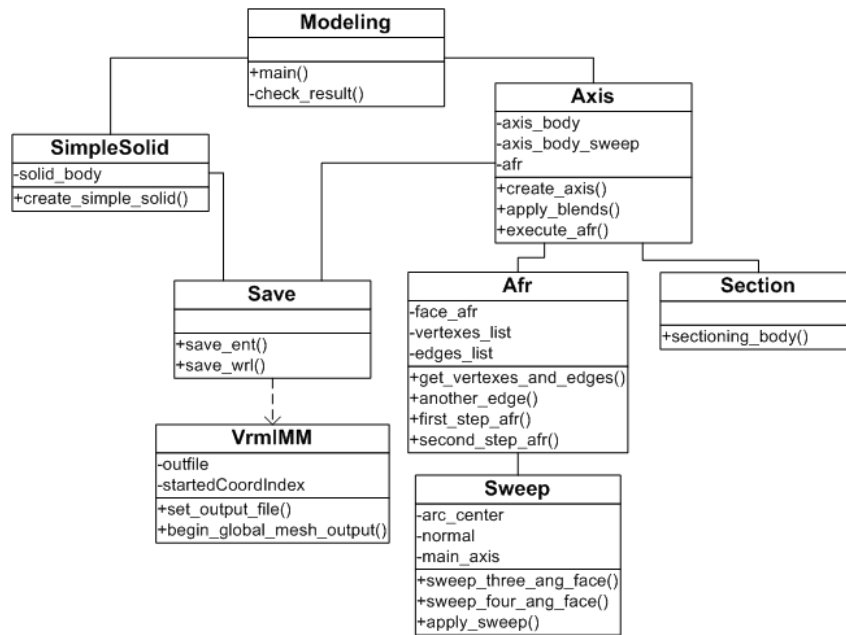


Figura 5. Diagrama de classes do programa executado via CGI

O método *get\_vertexes\_and\_edges()* implementado na classe AFR identifica todos os vértices e arestas que formam a metade do perfil 2D da peça e é executado sempre que um novo anel é adicionado a peça axissimétrica. O método *another\_edge()* é utilizado na implementação do método *second\_step\_afr()* para auxiliar no processo de identificação de *features* que apresentam como vértices apenas pontos internos, isto é, nenhum ponto no contorno.

Como pode ser observado na figura 6, o método *another\_edge()* consiste em traçar duas linhas reta (linhas azuis) ligando o ponto interno (ponto verde) à extremidade das outras duas arestas que são adjacentes as duas arestas (linhas vermelhas) que contêm este ponto interno. Ela retorna 0 (zero) caso a aresta da esquerda não tenha intersecção com a metade do perfil 2D da peça, 1 (um) caso a aresta da direita não tenha intersecção e -1 (um negativo) caso as duas possuam intersecção.

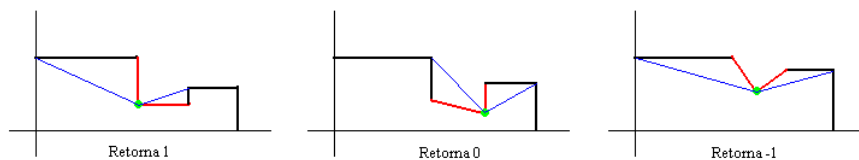


Figura 6. Funcionamento do método *another\_edge()*

O método *first\_step\_afr()* realiza a primeira etapa do reconhecedor de *features* verificando a existência de chanfros nas extremidades do eixo. Em caso positivo, ele define quais serão os três pontos da face e aciona o método *sweep\_three\_ang\_face()* da classe *Sweep*. Já o método *second\_step\_afr()* identifica todas as *features* restantes, definindo se a face de *sweeping* terá três ou quatro pontos e chama o método adequado da classe *Sweep*. O funcionamento deste método pode ser mais bem compreendido observando-se o *loop* no diagrama da figura 3.

## 5. EXECUÇÃO DO SISTEMA PARA O PROJETO DE UMA PEÇA AXISSIMÉTRICA

As funcionalidades do sistema serão apresentadas através de um exemplo no qual será gerada uma peça axissimétrica composta por oito cilindros de diversos diâmetros e alguns chanfros. Primeiramente tem-se a página inicial (lado esquerdo da figura 7) do sistema, na qual o usuário deverá selecionar qual o tipo de sólido que deseja modelar.

Feita a escolha de qual o sólido se deseja criar, a próxima etapa consiste em definir quais serão as características da peça. No caso da peça axissimétrica, o usuário deverá indicar a quantidade de cilindros/cones que irão compor a peça (lado direito da figura 7).

Figura 7. Definindo uma peça axissimétrica

Quando o usuário selecionar o botão “Continuar”, os dados são enviados para a página verificadora (*vEixo1.jsp*) e caso não houver nenhuma inconsistência, uma nova página será apresentada para a continuação da modelagem da peça (Figura 8).

Pode-se então introduzir o diâmetro esquerdo, o diâmetro direito e o comprimento de cada cilindro/cone que irá compor a peça assim como a presença ou não de chanfros. Existe ainda a possibilidade de seccionar uma ou duas vezes a peça ao longo do seu centro de rotação para uma visualização alternativa do sólido.

Figura 8. Continuação da definição da peça axissimétrica

Quando o usuário selecionar no botão “Criar eixo”, todos os dados que foram digitados nesta página serão enviados para outra página de verificação (*vEixo2.jsp*). Neste processo é verificado, por exemplo, se o comprimento de todos os cilindros/cones é um número positivo e maior que zero, e se é possível criar todos os chanfros nas medidas que foram definidas. Se o processo de verificação detectar algum problema, o usuário será informado do seu erro através da visualização de uma página idêntica à anterior (Figura 9) na qual poderá corrigir o erro.

Então quando o usuário selecionar o botão “Criar eixo” e todos os dados estiverem corretos, eles serão enviados para o programa CGI que está no servidor, o qual irá gerar a peça e fazer o reconhecimento das *features*. A figura 10 ilustra a interface para a visualização do sólido modelado. O arquivo no formato SAT da peça gerada pode ser obtido através do *link* presente no lado direito do visualizador VRML, para que o usuário possa manusear a peça em seu próprio software de CAD.

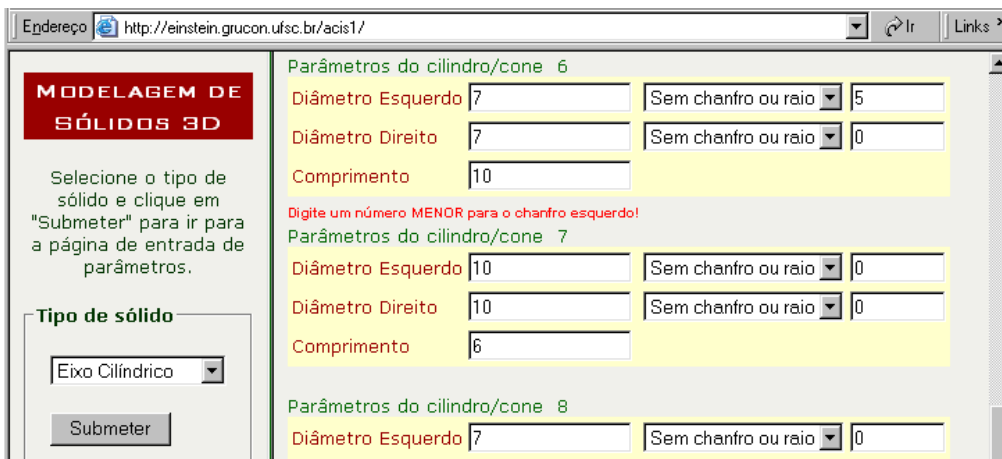


Figura 9. Exemplo de um erro na definição da peça

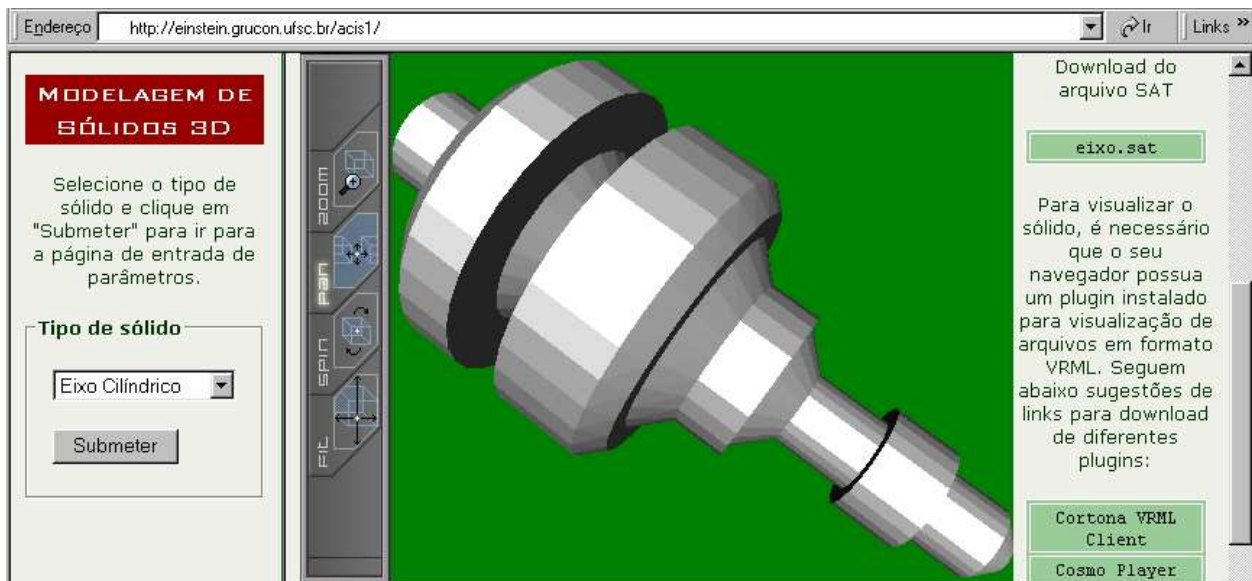


Figura 10. Visualização da peça na página de modelagem

Como resultado da execução do reconhecimento de *features*, foram criados 12 sólidos na forma de anel que foram salvos separadamente em arquivos no formato SAT, cada qual com uma cor distinta. Na figura 11 são apresentados alguns anéis que foram criados, sendo que a direção de cada um foi modificada para permitir uma melhor visualização<sup>3</sup>.

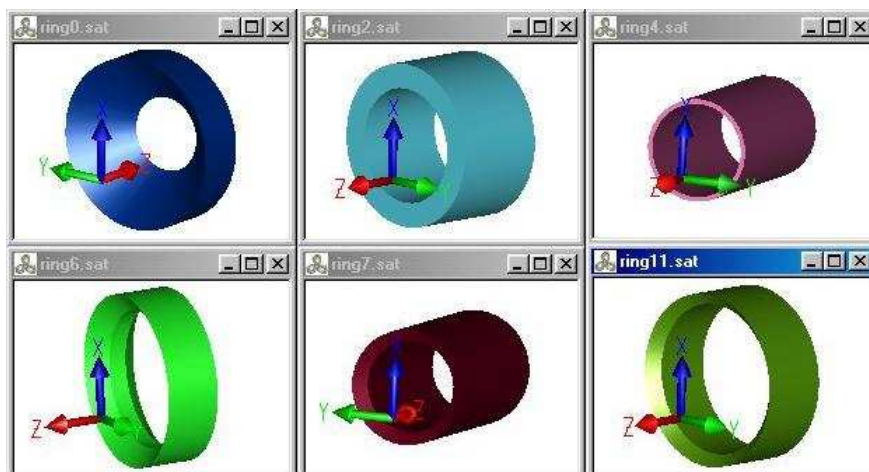


Figura 11. Visualização de alguns dos anéis gerados

<sup>3</sup> A visualização dos sólidos das figuras 11 e 12 foi feita através do software *HOOPS 3D Part Viewer for ACIS*.



A união de todos os anéis que foram gerados pode ser observada na figura 12. São apresentados através de sombreamento e também por arestas. Estes anéis representam todo o material que deverá ser removido pela máquina CNC para dar origem, a partir de uma barra cilíndrica, à peça axissimétrica projetada pelo usuário. Estes anéis representam um significativo encapsulamento das informações importantes para a geração do código G apropriado para a máquina CNC.

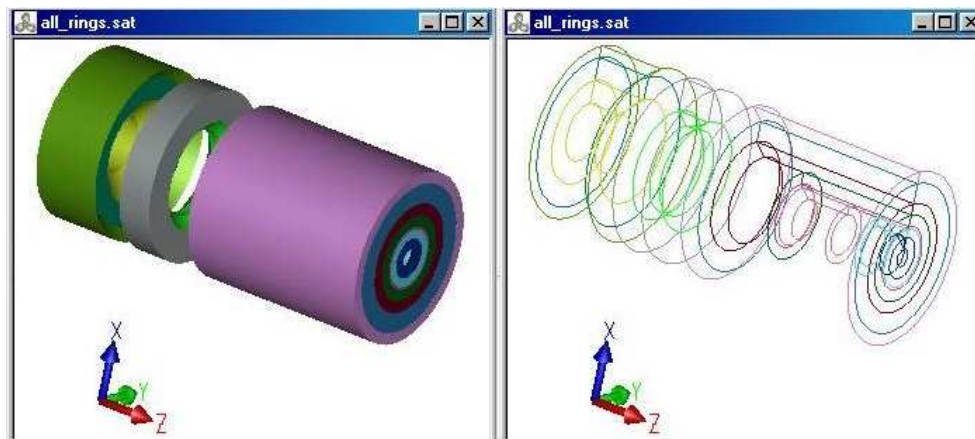


Figura 12. Visualização da união de todos os anéis criados

## 6. CONCLUSÕES

Neste artigo foi descrito um sistema para o projeto e manufatura de peças axissimétricas via Internet usando o modelador de sólidos ACIS, sendo que as peças criadas baseiam-se em primitivas sólidas. Foram criadas páginas em que o usuário escolhe o tipo de peça que deseja criar, e então são introduzidos os parâmetros do sólido. O sólido pode ser visualizado pelo usuário remoto através de qualquer visualizador VRML. Este sistema pode ser executado no link <http://einstein.grucon.ufsc.br/acis1> e apresenta as seguintes características:

- Realiza o reconhecimento de *features* em peças axissimétricas modeladas pelo usuário. Estas *features* poderão ser utilizadas na construção de um grafo de precedência, o qual possibilitará que seja determinado o código G mais apropriado para a máquina CNC que irá usinar a peça, diminuindo o custo e o tempo de fabricação da peça.
- O usuário não necessita possuir o kernel ACIS. O sólido é criado no computador servidor, o que desonera o computador cliente, ao mesmo tempo em que o usuário tem acesso a um software comercial (o qual neste caso é disponibilizado para uma aplicação específica).
- Além de este sistema possibilitar a visualização da peça em 3D, as informações topológicas e geométricas da peça estão disponíveis. Tais informações são importantes para o planejamento do processo (CAPP) e para a manufatura (CAM).

Portanto, este trabalho contribui para o desenvolvimento de um sistema mais complexo no qual a peça é projetada pela Internet, fabricada e enviada posteriormente para o projetista. Assim, ele terá um protótipo funcional dentro de poucos dias por um custo menor que a fabricação própria exigiria.

## 7. AGRADECIMENTOS

Ao CNPq, pelo suporte financeiro a este projeto.

## 8. REFERÊNCIAS

1. SMITH, C. S., WRIGHT, P. K., 2001, Cybercut: An Internet-based CAD/CAM System, ASME Journal of Computing and Information Science in Engineering, Vol. 1, No. 1, pp. 1-33.
2. SHAH, J. J., MÄNTYLÄ, M. **Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications**, John Wiley & Sons, New York, 1995.

3. ÁLVARES, A. J., FERREIRA, J. C. E. Uma Metodologia para Integração CAD/CAPP/CAM voltada para a Manufatura Remota de Peças Rotacionais via WEB. In: CONGRESSO BRASILEIRO DE ENGENHARIA DE FABRICAÇÃO (COBEF), 2., 2003, Uberlândia-MG.
4. CORNEY, J., LIM, T. **3D Modeling with ACIS**, Saxe-Coburg Publications, 2001.
5. DEITEL, H. M. e DEITEL J. **C++, como programar**, 3ª edição, Bookman, Porto Alegre, 2001.
6. SUN MICROSYSTEMS, Java Server Pages™ Technology. Disponível em: <<http://java.sun.com/products/jsp>>. Acesso em: 10 agosto 2003.
7. SUN MICROSYSTEMS, The Source for Java™ Technology. Disponível em: <<http://java.sun.com>>. Acesso em: 10 agosto 2003.
8. DEITEL, H. M. e DEITEL J. **Java, como programar**, 3ª edição, Bookman, Porto Alegre, 2001.
9. APACHE HTTP SERVER PROJECT, Introduction to Server Side Includes. Disponível em: <<http://www.apache.org/docs/howto/ssi.html>>. Acesso em: 08 março 2003.
10. FERREIRA, J. C. E., VIVIAN, D., NAKAHARA, K.C. e ÁLVARES, A.J. Implementação de um Software para o Projeto de Peças Axissimétricas Via Internet Usando Modelagem de Sólidos. In: CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA (CONEM), 3., 2004, Belém-PA.
11. RUDESERVER, Professional C++ CGI Development Libraries, C++ CGI Parser Library version 2.1. Disponível em: <<http://www.rudesserver.com/cgiparser/index.html>>. Acesso em: 01 dezembro 2003.
12. LAPADAT, D. Program to convert SAT files to VRML 2.0 file. Disponível em: <<http://edge.cs.drexel.edu/GICL/sat2vrml>>. Acesso em: 07 março 2003.

## **FEATURE RECOGNITION IN ROTATIONAL PARTS MODELED BY SOLIDS IN CAD/CAM SYSTEM FOR THE INTERNET**

**João Carlos Espíndola Ferreira**

Federal University of Santa Catarina, Department of Mechanical Engineering – GRUCON - GRIMA, Caixa Postal 476, CEP 88040-900, Florianópolis, SC, Brasil. [jcarlos@emc.ufsc.br](mailto:jcarlos@emc.ufsc.br).

**Darlan Vivian**

Federal University of Santa Catarina, Department of Mechanical Engineering – GRUCON - GRIMA, Caixa Postal 476, CEP 88040-900, Florianópolis, SC, Brasil. [darlan@inf.ufsc.br](mailto:darlan@inf.ufsc.br).

**Abstract:** *In the last years more and more people have been having access to the Internet, which has reduced or even eliminated the distances among people and/or groups located physically distant amongst themselves. The Internet has also enabled the development of applications for the support to design and manufacturing of industrial parts. In this work it is presented a system for automatic feature recognition in rotational parts created through the ACIS solid modeler via the Internet. In the part design stage the user, through his/her browser, inputs the data about the elements that compose the part. These data are analyzed and in case of no inconsistency, they are sent via CGI (Common Gateway Interface) to the software that resides in the server. To visualize the part, the file containing the data about the solid, which is in the SAT (Standard ACIS Text) format, is translated into the VRML (Virtual Reality Markup Language) format, and soon afterwards it is sent to the user's browser. The part may contain cylinders, cones and chamfers. The user can slice the part along its rotation center, and he/she can also obtain the solid file in the SAT format, in case he/she wants to model the part in his/her own CAD software. After the design is completed, the software investigates the geometry of the part. In order to do that, an algorithm was developed whose aim is to recognize the features, which is accomplished through the analysis of half of the 2D part profile. These features can be used in the G code generation for CNC machining.*

**Keywords:** *Computer-aided Design (CAD), Features, Internet, Solid Modeling, Axisymmetric Parts.*