IMPLEMENTAÇÃO DAS NORMAS 10303 - STEP EM UM AMBIENTE DE DESENVOLVIMENTO INTEGRADO DO PRODUTO

Prof. Dr.-Ing. Klaus Schützer
Lab. de Sistemas Computacionais para Projeto e Manufatura - FEMP/UNIMEP
e-mail: schuetzer@unimep.br

Eng. Antonio Álvaro de Assis Moura
Lab. de Sistemas Computacionais para Projeto e Manufatura - FEMP/UNIMEP
e-mail: a3moura.scpm@unimep.br

The integration of the information during all the cycle of life of the product is one of the biggest difficulties in the management of the information; the solution for this is the objective of STEP standards.

This article presents the use of the software tools for implementation of STEP standards, through the programming language C++ and of the application protocol AP-224 ("Mechanical Product Definitions for Process Planning Using Machining Features").

The Application protocols are subsets of STEP standards that contain the information of the product for a specific application (aeronautical, mechanics, electric, etc.).

The implementation of an application protocol, to be possible to read and to write a physical archive with the information of the product, is made with several software tools and in several programming languages. One of the more important software tools for the implementation of STEP standards is the "STEP Class Library - SCL" created by NIST and used in the described implementation in this article.

Keywords: ISO 10303 - STEP, Integration CAD/CAM/CAPP, Manufacturing features;

1. Introdução

A busca pela integração total das informações durante o ciclo de vida de um produto, i.e. durante todas as etapas do seu desenvolvimento iniciando em sua concepção e indo até o seu descarte, sempre foi um propósito da engenharia, para o qual várias convenções foram criadas. A introdução dos meios eletrônicos digitais deu grande impulso ao desenvolvimento do produto ao mesmo tempo em que sistemas proprietários em cada fase do desenvolvimento criaram uma barreira a esta integração.

As ferramentas mais modernas, que tem como objetivo a integração e representação dos dados do produto, são as normas ISO 10303 - STEP. Neste artigo apresenta-se como foi feita a implementação das normas STEP em um ambiente integrado de desenvolvimento do produto dentro do projeto *FESTEVAL*.

O projeto INCO-DC# 96-2161 - *FESTEVAL* - "Feature based Support for the Development Process Chain 'design-planning-manufacturing'" - atingiu seu objetivo de desenvolver um protótipo de sistema CAD/CAPP/CAM integrado, sendo que a integração entre as diversas etapas do desenvolvimento do produto foi feita através de um arquivo físico elaborado de acordo com as normas ISO 10303 - STEP - conforme o protocolo de

aplicação AP 224 - “Mechanical Product Definition for Process Planning Using Machining Features”.

Os parceiros deste projeto são: Universidade Metodista de Piracicaba, Indústrias Romi S.A., CEGOS - Kade-Tech S.A., “Institute of Production Engineering and Machine Tools” e “Department for Computer Integrated Design of Darmstadt University of Technology”.

A seguir é feita uma breve explanação sobre as normas STEP e sua interface de acesso de dados e uma descrição da implementação efetuada.

2. STEP

A série de normas ISO 10303: Representação e Troca de Dados do Produto, também conhecida como STEP (“STandard Product data and Exchange”) foi criada com o objetivo de integrar todas as informações do produto durante todo o seu ciclo de vida [1].

Com as normas ISO 10303 - STEP é possível ultrapassar a barreira imposta pela limitação de uma comunicação livre entre os vários atores do desenvolvimento do produto, por ser, além de outras características, um sistema neutro. Os benefícios a serem obtidos com a utilização das normas STEP para a troca de informações do produto são [2]:

- Redução do tempo de lançamento do produto pela facilidade de comunicação que incrementa o desenvolvimento simultâneo;
- Aumento da capacidade de múltiplas empresas trabalharem em conjunto no desenvolvimento do produto, através do compartilhamento das informações;
- Redução do custo do desenvolvimento do produto pela redução de atividades que não agregam valor e pelo aumento da eficiência na elaboração de revisões;
- Aumento da capacidade de utilizar novos softwares.

O produto final e mais visível das normas STEP é o arquivo físico seqüencial que contém toda a informação do produto em uma forma definida que pode ser lido e interpretado corretamente sem perda de informação por qualquer software que utilize as normas STEP.

Em sua seção 11 as normas STEP definem a linguagem EXPRESS como uma linguagem orientada a objetos criada especificamente para a descrição do modelo de referência da aplicação de uma maneira inequívoca.

3. SDAI - “Standard Data Access Interface”

A interface de acesso a dados padronizada define uma API - Interface de Aplicação de Programação (“Application Programmers Interface”) - para os dados definidos em EXPRESS. As normas STEP definem as operações com a interface SDAI e as informações do modelo em sua seção 22. Estas operações são implementadas para uma linguagem de programação específica como C, C++, Java e CORBA/IDL (“Common Object Request Broker Architecture/Interface Definition Language”). Neste artigo será apresentada a implementação com C++ definida na seção 23 das normas STEP.

As interfaces SDAI podem ser geradas de duas maneiras: básica ou avançada. Na implementação básica as funções de acesso da API são definidas já no modelo de informação em EXPRESS. As interfaces criadas no modo básico são implementadas por um compilador de EXPRESS que lê o modelo e gera o código que implementa funções para acesso aos dados. Na criação da interface de modo avançado a API fica sempre independente dos modelos de informação usados para representação dos dados.[3,4]

O código gerado deve permitir a criação de instâncias das classes, inserir dados e registrá-los em um arquivo físico STEP, permitindo ainda a leitura de um arquivo físico e apresentando o conteúdo deste como instâncias das classes [4].

4. SCL - “STEP Class Library”

A SCL (STEP Class Library) é um conjunto de bibliotecas de classes em C++ , criadas pelo “NIST - National Institute of Standards and Technology” que permite a representação da informação de acordo com a especificação de dados do EXPRESS (ISO 10303-11). As bibliotecas podem ser utilizadas para a construção de um programa executável em C++ nas aplicações que usam informações contidas em um arquivo EXPRESS. As bibliotecas contêm as entidades como um dicionário do esquema de informações do EXPRESS e funções para a representação e manipulações de instâncias dos objetos descritos no arquivo EXPRESS. Aplicações simples como a gravação e recuperação de dados do EXPRESS em arquivos na forma descrita nas normas STEP - Parte 21, podem ser facilmente elaboradas com a SCL [5,6].

O desenvolvimento da SCL teve como objetivo atingir uma série de propostas, a mais importante foi ser útil para a execução final dos conceitos das normas STEP, como a implementação dos métodos e a criação de software. A criação da SCL teve interações com as seguintes atividades: desenvolvimento das normas STEP, desenvolvimento da SCL baseada nas normas STEP e verificação das aplicações através de implementações concretas [7].

No projeto *FESTEVAL* a utilização das ferramentas do NIST foi feita através do software WinSTEP, desenvolvido pelo “Laboratory for graphical data processing of University of the Federal Armed Forces - Munich”. Este software é composto por um conjunto de ferramentas para utilização da linguagem EXPRESS, que auxilia em: edição de arquivos EXPRESS; compilação de arquivos EXPRESS; transformação de arquivos EXPRESS em código C++ e transformação de arquivos EXPRESS no formato HTML.

O WinSTEP fez com que as ferramentas fornecidas pelo NIST fossem integradas em um ambiente gráfico baseado nas classes fundamentais da Microsoft (MFC) de maneira que as ferramentas possam se comunicar em janelas comuns, facilitando o seu uso [8].

A base do desenvolvimento do software com uma aplicação STEP é a transformação do protocolo de aplicação das normas STEP em uma biblioteca de classes em linguagem de programação. Possibilitando assim a criação de instâncias das entidades definidas no protocolo de aplicação com os seus atributos, bem como a transposição destas instâncias em um arquivo físico STEP [4]. Neste trabalho será discutida a forma de atingir o objetivo descrito acima utilizando as ferramentas criadas pelo NIST, bem como a aplicação prática executada no projeto *FESTEVAL*.

5. Implementação

A implementação do software é composta de quatro etapas:

- Transformação do modelo de referência da aplicação em classes definidas em linguagem de programação adequada (C++);
- Leitura e transferência dos parâmetros do modelo geométrico;
- Criação de instância com os parâmetros do modelo geométrico;
- Criação do arquivo físico STEP.

ARM Modelo de Referência da Aplicação

Cada protocolo de aplicação das normas STEP é um documento formal que descreve: uma parte do ciclo de vida do produto, chamado modelo de atividade da aplicação - AAM (“Application Activity Modell”); as informações necessárias a cada uma destas atividades - ARM (“Application Reference Modell”); e um modelo com a

informação descrita no ARM utilizando uma biblioteca já definida - AIM ("Application Interpreted Modell") [8].

A *Figura 1* apresenta a descrição de um furo de acordo com o protocolo de aplicação AP 224 em linguagem EXPRESS. Conforme podemos ver um furo cilíndrico é um sub-tipo de furo e dele herda o atributo "edge_radius" que armazena um parâmetro numérico, além disto tem os seguintes atributos específicos: diâmetro; profundidade; mudança no diâmetro (conicidade), condição de fundo (passante, cego com fundo plano ou cego com fundo cônico). Também pode ser observado que a entidade "hole" é um supertipo para as entidades "countersunk" e "counterhole", e é um subtipo da entidade "machining feature".

No extrato apresentado na Figura 1 é possível notar duas importantes características de uma linguagem orientada a objetos: a de encapsulamento e a de heranças. Da característica de encapsulamento nota-se que todas as informações necessárias à descrição de um furo estão contidas na entidade "hole". Já a característica de heranças, pode ser notada por ser a entidade "round_hole" derivada da entidade "hole" e dela herdar seus atributos como também ser aquela uma superclasse para os demais tipos de furos transmitindo assim seus atributos.

```
     ENTITY hole
       ABSTRACT SUPERTYPE OF (
ONEOF(countersunk_hole,round_hole,counterbore_hole) )
       SUBTYPE OF (machining_feature);
     (*
     UOF:T1
     *)
       edge_radius : numeric_parameter;
      END_ENTITY;
     ENTITY round_hole
       SUBTYPE OF (hole);
     (*
     UOF:T1*)
       diameter : circular_closed_profil;
       hole_depth : linear_path;
       change_in_diameter : OPTIONAL taper_select;
       bottom_condition : hole_bottom_condition_select;
      END_ENTITY;
```

Figura 1: *Descrição de um furo em linguagem EXPRESS conforme o AP 224.*

No projeto *FESTEVAL* foram feitas modificações no AP-224 e na para sua implementação foi utilizado não um AIM e sim uma modificação do ARM [8].

Também existe a forma gráfica para a representação do protocolo de aplicação, onde pode-se ver com mais clareza as entidades e seus atributos. Na *Figura 2* tem-se o mesmo furo, agora representado de forma gráfica através do EXPRESS-G.

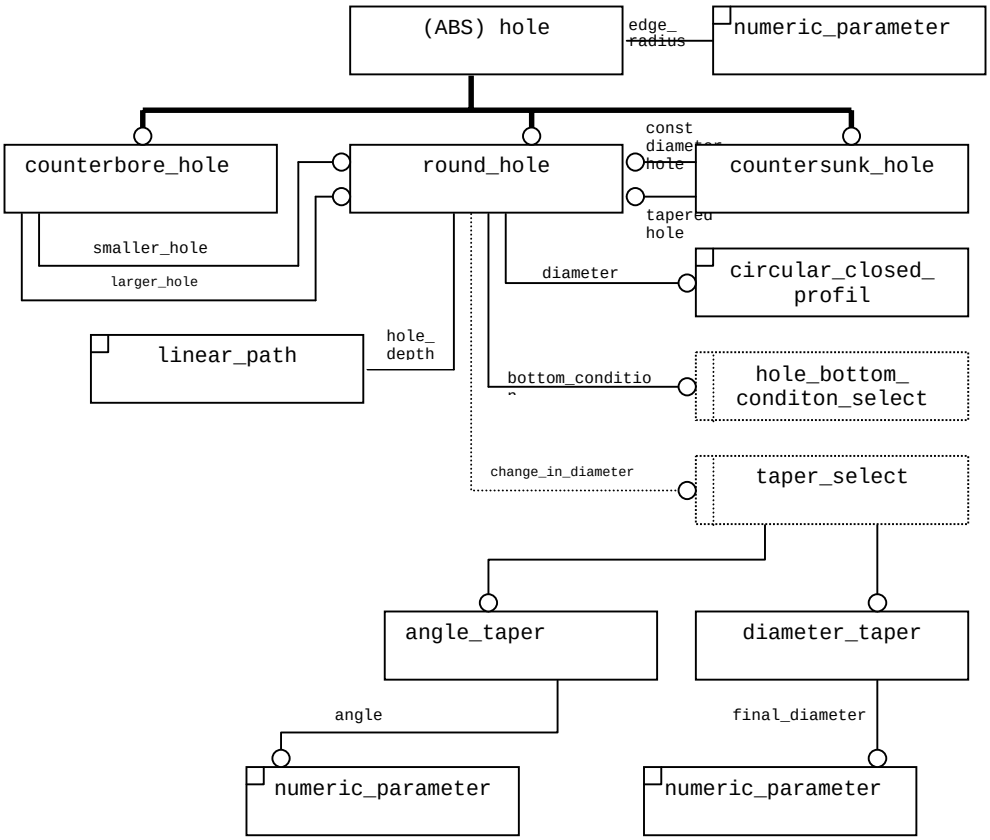

Figura 2: *Descrição dos furos em EXPRESS-G conforme AP 224*

Classe em C++

Para a criação das classes em C++ foi feita a conversão do arquivo em linguagem EXPRESS através do gerador de código “fedex_plus” do kit de ferramentas criado pelo NIST. Cada entidade existente no protocolo de aplicação gera uma classe em C++. Na Figura 3 pode ser visto uma parte do código em C++ que apresenta, na classe “hole”, os métodos “edge_radius” que dão acesso a um parâmetro numérico. Nota-se que o método é sobrecarregado pelo seu argumento, podendo ser utilizado para escrever um valor no atributo (argumento “Numeric_Parameter”) ou para ler o valor do atributo (sem argumento). Da mesma forma na Figura 4, podem ser vistos os métodos para ler e escrever os valores dos atributos: profundidade, conicidade e condição de fundo [8].

```
class SdaiHole : public SdaiMachining_feature{
...
SdaiHole ();
SdaiHole (SCLP23(Application_instance) *se, int *addAttrs=0);
int opcode() {return 205;}
const SdaiNumeric_parameter_ptr edge_radius_() const;
void edge_radius_ (const SdaiNumeric_parameter_ptr x);
...
```

Figura 3: *Código em C++ com a classe Hole e método edge_radius_*

```
class SdaiRound_hole : public SdaiHole {
...
SdaiRound_hole ( );
SdaiRound_hole (SCLP23(Application_instance) *se, int *addAttrs =0);
SdaiRound_hole (SdaiRound_hole& e);
~SdaiRound_hole();
...
public:

int opcode() {return 343};
const SdaiCircular_closed_profil_ptr diameter_() const;
void diameter_ (const SdaiCircular_closed_profil_ptr x);

const SdaiLinear_path_ptr hole_depth_() const;
void hole_depth_ (const SdaiLinear_path_ptr x);

const SdaiTaper_select_ptr change_in_diameter_() const;
void change_in_diameter_ (const SdaiTaper_select_ptr x);
...
```

Figura 4: *Código em C++ com a classe Round_hole e alguns métodos*

Leitura e transferência dos parâmetros de uma entidade

A etapa de leitura e transferência dos parâmetros de uma entidade foi feita dentro de um dos módulos do projeto *FESTEVAL*. Cada objeto derivado da classe Feature tem um método que faz a leitura dos parâmetros e coloca os valores em um objeto da classe CtransferObject, este objeto pode ser entendido como um contenedor onde qualquer tipo de informação pode ser inserida e posteriormente recuperada [9].

O objeto CTransferObject foi criado dentro da biblioteca de classes AP224IM, desenvolvido pelo "DiK - Department for Computer Integrated Design" e pelo "PTW - Institut for Production Engineering and Machine Tools", na "Darmstadt University of Technology" dentro do projeto *FESTEVAL*.

```
     CtransferObject* F_Hole_Simple::get_step_param()
     {
      CTransferObject *param;
      param = F_Feature::get_step_param();
     ...
7     retval  = UF_MODL_ask_simple_hole_parms (this->feature_id,1,
                                               &diameter,
                                               &depth,
                                               &tip_angle,
                                               &thru_flag);
12    param->DoubleParam(diameter, "Diameter");
13    param->DoubleParam(depth, "Depth");
14    param->IntParam(thru_flag, "ThroughBottom");
     ...
     return param;}
```

Figura 5: *Código em C++ com a leitura e transferência dos parâmetros de um furo.*

Na Figura 5 vemos o método "get_step_param" da classe "F_Hole_Simple" em que é feita a leitura dos parâmetros - linha 7 - e a transferência para o objeto da classe "CTransferObject" - linhas 12, 13 e 14.

Criação de instância da entidade com seus parâmetros

A criação da instância de um objeto é feita em um banco de dados preparado para a geração do arquivo físico STEP e cada instância será representada por uma linha no arquivo físico. O método utilizado para esta criação é o “Create_Entity”, da biblioteca de classes AP224IM. Na Figura 6 pode-se ver na primeira linha a criação da instância da classe “SdaiRound_hole”, em cuja linha no arquivo físico será iniciada pelo texto: “ROUND_HOLE”. Na segunda linha é feita a criação da instância da classe “SdaiCircular_closed_profil” como descrito acima. Na terceira linha é feita a leitura do diâmetro através da recuperação da informação inserida no objeto da classe “CTransferObject”, e nas próximas duas linhas é feita a inserção do atributo diâmetro na instância de “SdaiCircular_closed_profil” e desta na instância de SdaiRoundHole [10].

```
STEPentity* Create_RoundHole (CInfoModel* IM, CTransferObject* param)
{
1 SdaiRound_hole* hole= (SdaiRound_hole*)IM->CreateEntity(“Round_Hole”);
2 SdaiCircular_closed_profil* profil = (SdaiCircular_closed_profil* IM-
      >CreateEntity(“Circular_Closed_Profil”);
3 StepParam =NumericParameter(IM, param->NumericParameter(“Diameter”,0);
4 profile ->diameter_(StepParam);
5 hole ->diameter_(profile);
```

Figura 6: *Código em C++ para criação de uma instância Round_hole e parâmetros.*

Arquivo físico

Com todas as informações inseridas no banco de dados como atributos das instâncias das classes definidas no protocolo de aplicação é feita a transferência destas informações para um arquivo texto, estruturado de acordo com a descrição das normas STEP, conforme pode ser visto na Figura 7. A linha 127 tem o texto “ROUND_HOLE” e indica em seu sexto atributo a linha 128 que tem o texto “CIRCULAR_CLOSE_PROFIL” e indica a linha 129, que, por sua vez, tem o texto “NUMERIC_PARAMETER” e indica um diâmetro de 10 mm.

6. Conclusões

A utilização das normas STEP para integração de dados do produto tem se mostrada muito vantajosa e novas aplicações estão sendo continuamente implementadas dentro dos vários protocolos de aplicação, nas diversas áreas do desenvolvimento do produto. As ferramentas disponíveis para a implementação das normas STEP são úteis e fáceis de usar, atingindo o objetivo de incentivar o uso das normas STEP. No projeto *FESTEVAL* foram utilizadas com sucesso as ferramentas criadas pelo NIST no processo de implementação das normas STEP, onde foi conseguido o objetivo da integração da cadeia CAD/CAM/CAPP através de um arquivo físico no formato determinado pelas normas STEP com o protocolo de aplicação AP 224.

```
    ISO-10303-21;
    HEADER;
    FILE_DESCRIPTION((‘Step Physical File’),’Level 1.0’);
    FILE_NAME(C:\UGS160\UGII\furo’,’2001-02-22T11:14:59’,(‘Current
User’);
    FILE_SCHEMA((‘HSCWF’));
    ENDSEC;
    DATA;
    #0=PART($,$,#2,$,$,$,$,$,$,#1,$);
    #1=FEATURE_LIST((#127));
    #2=SHAPE($,$,#3);
    ...
    #127=ROUND_HOLE((#42,#101),$,#132,#148,#147,#128,#130,$,$);
    #128=CIRCULAR_CLOSED_PROFIL($,129);
    #129=NUMERIC_PARAMETER(‘Diameter’,‘mm’,10.);
    #130=LINEAR_PATH(#132,#131);
    ...
    ENDSEC;
    END-ISO-10303-21;
```

Figura 7: *Arquivo físico STEP com informações do furo.*

8. Bibliografia

/1/ HARDWICK, M.: STEP Dat Exchange Standards moves into implementation phase - STEP Tools, Inc. Rensselaer Technology Park, Troy, New York 12180 -1997.

/2/ MITCHELL, M. J.: Capabilities for Product Data Exchange - National Institute of Standards and Technology. Gaitherburg, MD - 1997.

/3/ FRISCH, H. P.: A Product data life cycle information management system. Infrastructure with CAD/CAE/CAM, Task automation, an intelligent support capabilities - NASA/Goddard Space Flight Center, Greenbelt, MD 20771, USA - 1998

/4/ SAUDER, D. A.; Morris, K.C.: Design of a C++ Softwares Library for implementing EXPRESS. The NIST STEP Class Library - Express User Group - 1995

/5/ SDAI C Reference - STEP Tools, Inc. Rensselaer Technology Park, Troy, New York 12180 - 1997.

/6/ SCRA: STEP application handbook - SCRA - International Boulevard North Charleston - SC - 2000

/7/ LOFFREDO, D: Fundamentals of STEP Implementation - STEP Tools, Inc. Rensselaer Technology Park, Troy, New York 12180 - 1997.

/8/ LEIBRECHT, S: Development of a design environment with STEP processor for a feature base 3D-CAD System. Universidade Metodista de Piracicaba. Diploma Thesis - 2000.

/9/ SCHÜTZER, K.; GARDINI, N.; FOLCO, J.C.: Modelador Baseado em Manufacturing Features Integrando Projeto, Processo e Fabricação. In: Anais do XIX Encontro Nacional de Engenharia de Produção, Universidade Federal do Rio de Janeiro, Rio de Janeiro - 1999. CD-ROM.

/10/ SCHÜTZER, K, MOURA, A.A.A., GARDINI, N, LEIBRECHT, S., Improved Software Tools of the Feature Modeller: Software Tool for Recognition of most Geometrical and Technological Interdependencies, Deliverable: 14-A/C, Project Number: INCO-DC #962161. Acronym: *FESTEVAL*. Project Title: Feature based support for the development process chain ‘design - planning- manufacturing’ - 2000.