

Migrating from IGES to STEP: one to one translation of IGES drawing to STEP drafting data

Mangesh P. Bhandarkar ^a, Blair Downie ^b, Martin Hardwick ^b, Rakesh Nagi ^{a,*}

^a *Department of Industrial Engineering, 342 Bell Hall, State University of New York, Buffalo, NY 14260, USA*

^b *Department of Computer Science and Center for Advanced Technology in Automation, Robotics and Manufacturing, Rensselaer Polytechnic Institute, Troy, NY 12180, USA*

Received 12 January 1998; accepted 21 June 1999

Abstract

This paper focuses on a procedure for converting product design data from an Initial Graphic Exchange Specification (IGES) format into Standard for the Exchange of Product Model Data (STEP) format. IGES is used for representing design and topology information in a platform independent format. The failure of the IGES standard to provide information other than drawing and solid modeling has led to the emergence of a new international standard for product model data exchange (ISO 10303) called STEP. STEP is designed to support a product through its entire life cycle. The STEP standard consists of a set of application protocols. An application protocol is specific to a particular engineering domain. In this case, STEP application protocol 202 is used because it defines the constructs necessary to represent the associative draughting information described by IGES. The translator parses the IGES file and converts it into the corresponding STEP (AP202) format. To test the conformance of the generated STEP file we have written code to enable visualization of an Application Protocol 202 (AP202) STEP file. The visualization uses the Open Inventor Toolkit available for the Silicon Graphics platform. The visualizer reads the STEP (AP202) file and converts the geometry and related annotations into the corresponding Inventor specific format. The file can then be viewed using the default Inventor viewer. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: IGES; STEP; Translation

1. Introduction

The past few years have seen a change in manufacturing strategies and technologies. The markets have become more and more fragmented, and competitiveness has increased. For a manufacturing enterprise to be competitive and thrive in such a dynamic environment it must be capable of adjusting

rapidly to the market. This has led to the emergence of a new manufacturing paradigm called Agile Manufacturing, where multiple companies cooperate under flexible virtual enterprise structures. The impetus is on producing products of better quality in the least amount of time and at the best possible price.

Agile manufacturing makes use of modern information technology to form a virtual enterprise. Partners must maintain a high amount of communication and data exchange between themselves for the success of the virtual enterprise. Data exchange does not

* Corresponding author. Tel.: +1-7166452357; fax: +1-7166453302; e-mail: nagi@eng.buffalo.edu

mean just verbal communication, but the companies must exchange complete information about their products, all the way from design, manufacturing to inspection and shipping. This information should be available to each relevant partner over the entire life cycle of the product.

Researchers have recognized the need for an international standard for product data exchange which would facilitate the seamless exchange of data. Companies in their own efforts have also recognized the same need. Attempts were made in the past to develop an international standard for data exchange. The earlier efforts led to the Initial Graphic Exchange Specification (IGES), which stores drawing information in an ASCII or binary neutral format which can then be exchanged between various users. Many Computer Aided Design (CAD) software vendors support IGES. Files in the IGES format can be loaded into these packages and in turn saved back into the IGES format. As effective as it might seem, IGES is currently being used in the industry but lacks considerably in many respects such as the lack of a formal information model, problems during file exchanges and manipulations, lack of life-cycle support and hard to understand file formats.

This led to the development of the Standard for the Exchange of Product Model Data (STEP). STEP became an ISO standard in 1993. STEP is published in a series of parts which include the following: description methods, integrated resources, application protocols, abstract test suites, implementation methods and conformance testing. The information in STEP is represented using an information modeling language called EXPRESS [13]. STEP uses application protocols (APs) to specify the representation of product information for one or more applications. For example the application protocol that defines the constructs for representing product shape and assembly information is called 'Configuration Controlled Design', AP203. The draughting (in ISO terminology drafting is spelled draughting, we will be using the ISO spelling throughout the paper) information can be represented in AP202, 'Associative Draughting' schema.

STEP has gained considerable importance due to active support from the automobile, aerospace and the defense sector. PDES Inc., an industry/government consortium, has been formed for accelerating

the development and implementation of STEP in the United States. Similar organizations exist in Germany, United Kingdom and Japan. Many companies are providing tools to access and manipulate STEP data. Also, the development in Internet related software has led to a rapid increase in STEP-based product data exchange over the world wide web. CAD software vendors and third parties for systems like AutoCAD, PRO/Engineer, CATIA, Unigraphics, Microstation, Trispectives and ACIS, have STEP interfaces in their new releases. As the acceptance and the use of STEP-based information exchange increases, a need for translation services to convert information represented in legacy file formats like IGES into STEP will arise. In addition to this is also the fact that the current version of IGES may be the last and a policy will be developed for companies to migrate to STEP. We have realized this requirement and the current paper focuses on our efforts in the development of a translator from the IGES drawing specification into the relevant STEP format.

The paper is organized as follows. We first present a brief overview of IGES and STEP followed by a literature review. We go on to describe the current problem on hand and present the work done in terms of the coding and testing of the translator. To validate the converted STEP file, work has also been done to develop a visualizer. Finally we conclude with a discussion and proposed further work.

2. Background

This section briefly covers the history of electronic data exchange for product design information. The IGES standard which can be regarded as the "first modern electronic data exchange tool" is described. We have also provided reasons as to why IGES did not gain much popularity, and also taken a look at the STEP standard and its benefits over IGES.

2.1. Product data exchange history

In the 1960s and most of the 1970s, each CAD system had its own file format for the representation of product design information. Information transfer was done by producing hard copy drawing from the first systems and then manually reentering the draw-

ing information into the second CAD system. This was a very tedious process, it was prone to errors and caused the repetition of data in multiple file formats.

As the number of CAD-based systems increased, the need for exchange of information between the various systems increased. At this early stage, electronic data transfer between the various systems was achieved by using custom translator software to convert data files from one system format to another. Although custom translators were fast, efficient and reasonably accurate if written properly, this approach suffered from the following drawbacks. The number of custom translators required to transfer files between all the CAD systems is equal to $N \times (N - 1)$, where N is the number of CAD and engineering systems. Having custom translators between every CAD system, even in a moderately sized organization, far outweighs the effectiveness of the translators. Also, writing, testing and maintaining custom translators is expensive and consumes considerable amount of resources.

With the increase in the number of CAD-based engineering applications like designing, finite element analysis and process planning, along with the advent of the computer integrated manufacturing philosophy, the need for a standard for representing CAD and other engineering information became increasingly apparent. This led to the development of the IGES standard, which specifies a neutral file format for exchanging data between CAD and/or engineering systems. CAD systems, using IGES to exchange data, only require a pre-processor to import an IGES file into the CAD system and a post-processor to convert the data back to the IGES format. This eliminates the need for custom translators between each of the individual CAD systems. Section 2.2 describes the IGES specification and some of the difficulties associated with IGES data exchange which has led to the development of the STEP standard.

2.2. Initial Graphic Exchange Specification (IGES)

IGES provides a standard format by which the user can transfer product design data from one CAD system to another. IGES specification is concerned with the data required to describe and communicate

the essential engineering characteristics of physical objects. The application areas covered by IGES include electrical, plant design, as well as mechanical applications.

IGES has information structures or entities to be used for digital representation and communication of product definition data. In other words, the IGES format treats the product definition as a file of entities. Each entity, which is the fundamental unit of data in an IGES file, is represented in an application-independent format. The entity representation contains enough information to support the needs of most of the currently available CAD/CAM systems. The entities can be broadly classified into geometry entities and non-geometry entities which are elaborated in the following.

Geometry entities represent the definition of the physical shape and include points, curves, surfaces, solids and relations which are collections of similarly structured entities. Non-geometry entities provide a viewing perspective of which a planar drawing is composed, by providing annotation and dimensioning appropriate to the drawing. Non-geometry entities further serve to provide specific attributes or characteristics for individual entities. Non-geometry entities include view, drawing, dimensions, text, notation, witness lines and leaders. For details about the IGES specification, the descriptions and arrangement of the data in an IGES file the readers are directed to [9].

Since IGES specifies a neutral file format for the representation and exchange of product data, it has a clear advantage over custom translator development and maintenance. Although IGES seems to be quite useful and facilitates electronic data exchange, IGES has some problems. IGES does not have a formal data model which leads to ambiguities in certain cases. The IGES files because of their 80-column file format are very verbose. These files are also very difficult to understand because of the complicated way in which the data is arranged. If there is an error in the IGES file, it is very difficult to find the mistake and correct it. Errors may arise due to changes made in the file. Further, due to lack of any conformance requirements imposed by IGES, CAD vendors have developed their own flavors for the IGES files which are not often consistent with other vendors. This leads to incomplete translation and

loss of information, which is unacceptable for engineering applications.

IGES does not provide data that may be relevant to applications other than drawing or 3D modeling. For example the designer may be interested only in the drawing or the geometry of the part to be manufactured, but the process planner would be more interested in the manufacturing view of the part which may contain form features and the manufacturability information of the part, and the inspection department may be interested in the tolerance information for the part. Each of the departments would share the same IGES file and it is left to the interpretation of the individual to derive the relevant information. Thus IGES does not provide any life-cycle related information and is also not very extensible due to the lack of any integrated resources.

To summarize: (i) IGES does not have a formal information model, (ii) there is a problem of incomplete exchange due to various ‘flavors’ added by CAD vendors, (iii) IGES does not support life-cycle information which may be relevant for engineering applications other than design, and (iv) the IGES 80-column format is not easily human understandable and hence mistakes if any cannot be easily found and corrected. All these reasons led to the development of the STEP standard which aims at addressing all the above issues.

2.3. Standard for the Exchange of Product Model Data (STEP)

The shortcomings of the IGES standard led to the development of the ISO STEP standard. STEP provides representation of product information along with the necessary mechanisms and definitions to enable product data to be exchanged. The exchange is among different computer systems and environments associated with the complete product life cycle including design, manufacturing, utilization, maintenance and disposal. The information generated about a product during these processes is used for many purposes. Such use may involve many different computer systems within the same organization or between different organizations.

STEP is based on a three layered architecture: (i) the reference model which develops a number of topical models specific to individual applications, (ii)

the logical layer which specifies the format definition language called EXPRESS, and (iii) the physical layer which defines the communication file structure called STEP file.

STEP uses a formal information modeling language, EXPRESS [13], to specify the product information to be represented. The use of a formal language enables precision and consistency of representation and facilitates implementation. Application protocols (APs) are used to specify the representation of product information for one or more applications. Application protocols are based on four main ideas: (i) scope and context of application, (ii) an Application Reference Model (ARM) defining the requirements, (iii) an Application-Interpreted Model (AIM) which satisfies the requirements given in the ARM using STEP constructs, and (iv) conformance requirements and test procedures for compliance with the application protocol. Several APs have already been developed to support different kinds of engineering applications. A graphical view of the STEP architecture can be found at [17].

STEP enables all people contributing to the design, manufacturing, marketing and supply of a product and its components to contribute to, to access, and to share information. STEP aims at eliminating the concept of ‘islands of automation.’ STEP also attempts to unite manufacturing efforts among corporate partners, distant subsidiaries and suppliers across diverse computer environments. It is intended to fit in with the new emerging paradigm of virtual enterprises and agile manufacturing. STEP addresses the issues of diversified engineering applications and covers security aspects, which become relevant now that several companies would be sharing the same product information.

Due to its increasing awareness and the presence of compatible tools STEP has started to gain considerable industry acceptance, notably in the aerospace and defense industry, automotive industry and ship building industry. Companies have now started looking at STEP as a means of defining and storing product data information. Already product geometry and 3D rendering information are being stored as documents defining the product in enterprise product data management (PDM) systems. This allows viewing and modification of geometry using any CAD tool capable of interpreting STEP geometry and

breaks the dependency between CAD systems and product definition. However the problem still exists for drafting data. Storing drafting information in a legacy format, such as IGES, leads to redundant representation of product definition and geometry information in multiple formats. Such redundant storage, in an environment where product data is being continuously changed and updated, leads to incoherent product data and incompatibility between different forms of representation of the same information.

Two dimensional representation of geometry in the form of drawings can be perceived as one particular view of product information (as opposed to 3D view). In fact many CAD systems treat drawing information as 3D geometry projected onto a 2D plane. The representation of drawing data in a STEP format facilitates uniform representation of multiple views of product data thus enabling better control. All the above factors warrant a translation between IGES and STEP to achieve representation of product data in a single standard compliant and system independent format. Section 3 provides a literature review about existing work done in this area.

3. Literature review

The authors are not aware of any large scale efforts to write a translator from IGES to STEP, in particular for 2D draughting information of designs. However STEP Tools Inc., [18] have a translator which converts IGES geometry information into NURBS (Non-Uniform Rational B-Splines) for visualization purposes. This translator however translates only the geometry and does address any drafting type information. Another aspect of this translator is that it converts all the geometry into STEP NURBS and thus does not make use of the vast number of geometry resources offered by STEP. Other STEP software vendors, namely ProSTEP and EuroSTEP may also may have such commercially available translators however to the best of the authors knowledge none of them specifically translate more than geometry and product structure information.

There may however be certain small scale translators available which have been written for particular applications. Basu and Kumar [2] have written a

translator for finite element analysis. The translator extracts mesh entities (nodes and elements) from an IGES file and a simple Fortran 77 program converts these into relevant entities for Finite Element/Boundary element analysis.

Ssemakula and Satsangi [16] have done some work with IGES and PDES¹ for computer integrated process planning. However they have not written any direct translation between IGES and PDES. Liu et al. [14] have written a translator to convert IGES entities into an object-oriented format. The objects are then used for feature extraction. The translator does not deal with all the IGES entities and does not convert the entities into the STEP format.

CAD software companies, like AutoCAD [1], PRO/Engineer [10] and CATIA [4], have developed their own in-house translators to convert from IGES into their software specific format. This is to facilitate import and export of IGES files from their CAD product to enhance acceptability or migration to their system. However these translators cannot be used independently for information exchange without their respective proprietary post-processor which write out files from their CAD format into IGES or STEP.

STEP Tools Inc., [18], has a visualizer that can be used to view AP203 STEP files. However we are not aware of any visualizer for AP202 [11,12] STEP files. There has been some work done to convert IGES entities into Silicon Graphics Inventor format. This effort has been done by a third party vendor for Silicon Graphics [15] but again nothing that handles geometry as well as drafting data.

4. Approach

STEP has gained considerable popularity in the recent past, mainly due to the active support for STEP from the automobile and the aircraft industries. The number of STEP-based applications has been on the rise for the past few years. CAD vendors are also developing STEP interfaces to their products. Companies that have been using IGES-based CAD systems have considerable data in this legacy

¹ PDES was the precursor to STEP in the US.

format. To avoid the regeneration of this data in the STEP format, our objective is to develop a translator from the legacy format to the new file format. For this current project we have concentrated on the geometry, topology and draughting information.

The STEP scheme which is most relevant for the task on hand is the AP202 [11,12], Associative Draughting Schema. AP202 has recently been released as an International Standard (IS). This section describes AP202, the associative draughting schema. To approach the translation effort in a formal manner we have developed an information model of the IGES specification using EXPRESS. Then, the mapping between IGES and AP202 has been described in terms of their respective EXPRESS schema entities. Along with the conceptual clarity, this approach has other advantages including database population of large files (see Section 5).

4.1. Application protocol 202 (associative draughting)

AP202 is the STEP schema intended for inter-organization exchange of computer-interpretable drawing information and associated product definition data. The EXPRESS definition of the AP202 schema can be obtained from Ref. [12]. The following fall under the scope of AP202:

- Structures for representation drawings for the purpose of exchange, suitable for mechanical and architectural engineering, and construction applications.
- Drawing management information.
- Information on product versions being documented by drawing.
- Computer-interpretable product shape model defined in two or three dimensional coordinate space.
- Transformations and appearance characteristics used for the generation of drawing views.
- Association between dimensions, draughting callouts, or fill areas and product geometry.
- Mechanisms for the grouping of the elements depicted on a drawing.

Drawings have been the traditional way of conveying product shape information, by means of standard orthographic projections. For many years drawings were generated manually and today CAD sys-

tems are used to generate the same drawings. The method used by CAD systems differs: some use the principles of 2D representation as used in manual drawings, while others provide the facility to create a 3D geometric representation or draughting shape model of the product shape. The 2D lines and curves are then generated from this draughting shape model using viewing information. Finally there are some systems, for integrated modeling and draughting, which provide a three-dimensional draughting shape model. A window into the draughting shape model is utilized so that the model may be seen or viewed as a part of the drawing. AP202 supports draughting of product in the following three ways: (i) 2D Annotation Geometry on the drawing sheet, (ii) 2D View Projections of the product shape, and (iii) 2D or 3D Shape with information to derive view projections.

Constructive Solid Geometry (CSG) solids cannot be converted into STEP (AP202 to be specific) directly, as AP202 uses Boundary Representation (B-Rep) to represent solid models. CSG solids can be converted by using a solid modeler like ACIS to interpret the CSG tree and convert it to the corresponding B-Rep format. For this reason we have eliminated CSG entities and concentrated our efforts on converting the rest of the IGES entities. Finite element modeling entities are not supported in AP202, hence these have not been considered in our current work.

4.2. IGES as EXPRESS schema

EXPRESS is an information model specification language which was developed to enable writing formal information models for describing mechanical products. As we noted earlier, STEP uses the EXPRESS modeling language to represent the product information model. The advantages of using a formal language have been emphasized in Section 2.3. To add to the same, EXPRESS is also “object-oriented like” allowing the use of object-oriented methodologies like encapsulation, abstraction and inheritance [13]. EXPRESS also has a formally defined graphical language called EXPRESS-G, which is used for representing subsets of the textural form. EXPRESS-G allows the easy understanding and manipulation of the information model.

Due to all these advantages of using EXPRESS, a schema for the IGES specification has been written. The complete listing of the EXPRESS model for IGES can be found at [8]. The top level schema is called *iges_types_schema*; Fig. 1 shows the *iges_entity_type_schema*. This schema contains an entity called *iges_file* which is the highest level entity. The *iges_file* entity contains a list of strings which define the start section of the IGES file, a pointer to an entity which defines the global section of the IGES file, and a list of *directory_entry*. The

directory_entry is an abstract supertype of *geometric_entity*, *constructive_solid_geometry_entity*, *boundary_representation_entity*, *annotation_entity*, or *structure_entity*. These five types of entities are referenced from five separate schemata:

1. The *iges_curve_and_surface_geometry_schema* covers all the geometry related entities.
2. The *iges_annotation_schema* contains the definitions of all the annotation entities.
3. The *iges_boundary_representation_schema* covers all the B-Rep topology entities.

```

SCHEMA iges_types_schema;
REFERENCE FROM
    iges_curve_and_surface_geometry_schema (geometric_entity);
REFERENCE FROM
    iges_constructive_solid_geometry_schema (constructive_solid_geometry_entity);
REFERENCE FROM
    iges_boundary_representation_schema (boundary_representation_entity);
REFERENCE FROM
    iges_annotation_schema (annotation_entity, general_note_entity);
REFERENCE FROM
    iges_structure_schema (structure_entity, definition_levels_entity,
        associativity_entity, color_definition_entity,
        line_font_definition_entity, view_entity,
        views_visible_color_entity, views_visible_entity,
        segmented_views_visible_entity, property_entity,
        text_display_template_entity, attribute_table_entity);
:
:
ENTITY iges_file;
    start_section : LIST OF STRING;
    global_section : global;
DERIVE
    directory_entries : LIST OF directory_entry :=
        QUERY ( de < * directory_entry | TRUE );
END_ENTITY;
ENTITY directory_entry
    ABSTRACT SUPERTYPE OF (ONEOF (geometric_entity,
        constructive_solid_geometry_entity,
        boundary_representation_entity,
        annotation_entity,
        structure_entity));
    label : STRING;
    subscript : INTEGER;
    associations : LIST OF associativities;
    properties : LIST OF properties;
END_ENTITY;
END_SCHEMA; -- iges_types_schema

```

Fig. 1. IGES types SCHEMA.

4. The *iges_constructive_solid_geometry_schema* covers all CSG elements.
5. The *iges_structure_schema* contains all the structure entities.

Each schema covers a particular class of entities which are written as EXPRESS classes and classified into one of the five schemata. The parameters of each entity, as defined by IGES, form the data

elements of each class. Fig. 2 shows the EXPRESS definition of a *geometric_entity* and *circular_arc_entity* which is a subtype of the former.

4.3. IGES to AP202 mapping

The development of the translator involves determining the mapping between the IGES entities to the

```

SCHEMA iges_curve_and_surface_geometry_schema;
  ENTITY geometric_entity
    ABSTRACT SUPERTYPE OF (ONEOF (circular_arc_entity,
                                   composite_curve_entity,
                                   conic_arc_entity,
                                   coordinate_pairs_entity,
                                   coordinate_triples_entity,
                                   coordinate_sextuples_entity,
                                   linear_path_entity,
                                   :
                                   )
    SUBTYPE OF (directory_entry);

    level : entity_level;
    view : entity_view;
    transformation : transformation_matrix_entity;
    label_display : label_display_entity;
    blank : blank_status;
    color : entity_color;
  END_ENTITY;

  ENTITY circular_arc_entity
    SUBTYPE OF (geometric_entity);

    line_font : lineFont;
    subordinate : subordinate_switch;
    used : use_flag;
    line_weight : INTEGER;
    zt : REAL;
    x1 : REAL;
    y1 : REAL;
    x2 : REAL;
    y2 : REAL;
    x3 : REAL;
    y3 : REAL;
  DERIVE
    entity_type : INTEGER := 100;
    form : INTEGER := 0;
  END_ENTITY;
  :
END_SCHEMA; -- iges_curve_and_surface_geometry_schema;

```

Fig. 2. EXPRESS definition for *geometric_entity* and *circular_arc_entity*.

corresponding STEP (AP202) entities. The mapping is based on the perception and understanding of the IGES and AP202 definitions of entities. The complete mapping can be found in Table 1. Further explanation and rationale for specific mapping can be found at Ref. [3]. Some of the entity which warrant special mention are listed below.

(1) IGES *rational b_splines* curves and surfaces are mapped to a composite entity formed by “AND’ing” the *b_spline* (curve or surface) with knots and a *rational b_spline* (curve or surface). This is necessary because IGES specifies knots and weights for *rational b_splines* whereas STEP does not put any such restriction. Hence to achieve complete mapping a composite entity must be used.

(2) IGES *witness line entity* used for dimension projection lines on the drawing sheet is mapped into AP202 *projection curve*. The witness line entity always has an odd number of points, where the lengths of the visible and invisible segments are specified and the first segment being invisible. The visibility and invisibility is dealt with using the AP202 *curve style font pattern*. The visible–invisible portions are important for proper display of dimension lines on the drawing sheet.

(3) IGES dimension entities are composite entities which are composed of a general note entity, one or more witness line entities, and one or more leader arrow entities. All these individual components are mapped into a *set of draughting callout elements* and then tied back into the respective AP202 dimension entity. The *angular dimension entity* is a special case in which the points of the leader arrows are joined using circular arcs rather than lines; this has been treated separately by passing an arc between the points of the *leader arrow entity*. For complete definition of the 2D geometry the dimension is tied-back to the corresponding geometry element which is represented by the dimension.

(4) IGES *manifold solid brep entity* is a solid model representation entity in the B-Rep format which is mapped into an AP202 *brep with voids or manifold solid brep* depending on the presence or absence of void shells. A more detailed explanation on the mapping of IGES manifolds can be found in Ref. [3].

(5) IGES *view entity* is mapped into an AP202 *view volume*. The view volume has the required

information to project the geometry from drawing space coordinates to view space coordinates. This is used for orienting and placing the geometry in the right perspective on the drawing sheet. There is also information required to clip the geometry for partial views of the parts.

(6) IGES *drawing size entity* is converted into AP202 *planar box*. This planar box is used to calculate the extent of the drawing sheet.

Having described the mapping between IGES and AP202, we now go on to describe the development of the translator and visualizer.

5. Implementation

This section describes the implementation of the IGES to AP202 translator. A discussion on the working of the visualizer is provided at the end of this section along with examples to validate the translation from IGES to AP202. The implementation is done in C++ using object oriented programming techniques. An object oriented database called Rensselaer Object Store for Engineering (ROSE) [6] developed at Rensselaer Polytechnic Institute is used for storing and retrieving the objects. The ROSE database works using the compiled EXPRESS definition of the schema. ROSE contains capabilities of storing aggregate data types like lists, sets, bags and arrays supported by EXPRESS, and also has functionalities to manage the objects. A complete description of ROSE can be found in [6]. The EXPRESS schemata for IGES and AP202 are compiled into C++ classes using the *express2c++* utility provided in ST-Developer by STEP Tools Inc. Class extension functions are then written to read-in the IGES file and to convert from IGES to AP202. Due to the complex nature of the IGES file, the IGES input/output mechanism is presented first and then the translator implementation is explained.

5.1. IGES input/output

The IGES input/output (I/O) is used to read in the data from the IGES file and write output to a file in the IGES format. We are currently using the IGES I/O only for reading in the IGES file and hence every reference to IGES I/O will be made as IGES

Table 1
IGES to AP202 mapping

No.	IGES entity	AP202 entity
1	Angular Dimension Entity	Draughting Annotation Occurrence and Angular Dimension
2	Basic Dimension Entity	Annotation Curve Occurrence
3	Boundary Entity	Boundary Curve
4	Bounded Surface Entity	Curve Bounded Surface
5	Centerline Entity	Annotation Curve Occurrence and Draughting Annotation Occurrence
6	Circular Arc Entity	Trimmed Curve or Circle
7	Color Definition Entity	Colour RGB
8	Composite Curve Entity	Composite Curve
9	Conic Arc Entity	Trimmed Curve with <i>basis_curve</i> as Parabola, hyperbola, ellipse
10	Coordinate Pair Entity	Polyline
11	Coordinate Sextuples Entity	Composite Curve
12	Coordinate Triples Entity	Polyline
13	Curve Dimension Entity	Curve Dimension and Draughting Annotation Occurrence
14	Curve on a Parametric Surface Entity	Boundary Curve
15	Diameter Dimension Entity	Diameter Dimension Entity and Draughting Annotation Occurrence
16	Edge Entity	Oriented Edge
17	Edge List Entity	Edge Curve
18	External Reference Entity	Externally defined item
19	Face Entity	Advanced Face
20	Flash Entity	Circle or Planar Box or Composite Curve
21	General Label Entity	Leader Directed Callout and Draughting Annotation Occurrence
22	General Note Entity	Annotation Text Occurrence and Draughting Annotation Occurrence
23	General Symbol Entity	Leader Directed Callout and Draughting Annotation Occurrence
24	Leader Arrow Entity	Dimension Curve Terminator and Draughting Annotation Occurrence and Leader Terminator
25	Line Entity	Trimmed Curve
26	Linear Dimension Entity	Linear Dimension and Draughting Annotation Occurrence
27	Loop Entity	Face Outer Bound or Face Bound
28	Manifold Solid B-REP Object Entity	Manifold Solid B-Rep or B-Rep with voids
29	New General Note Entity	Annotation Text Occurrence and Draughting Annotation Occurrence
30	Offset Curve Entity	Offset Curve 3D
31	Offset Surface Entity	Offset Surface Entity
32	Ordinate Dimension Entity	Ordinate Dimension Entity and Draughting Annotation Occurrence
33	Parametric Spline Curve Entity	B-Spline Curve with Knots
34	Parametric Spline Surface Entity	B-Spline Surface with Knots
35	Piecewise Curve Entity	Polyline
36	Planar Piecewise Curve Entity	Polyline
37	Plane Surface Entity	Plane
38	Point Entity	Annotation Subfigure Occurrence and Draughting Annotation Occurrence
39	Radius Dimension Entity	Radius Dimension and Draughting Annotation Occurrence
40	Rational B-Spline Curve Entity	Rational B-Spline Curve and B-Spline Curve with Knots
41	Rational B-Spline Surface Entity	Rational B-Spline Surface and B-Spline Surface with Knots
42	Simple Closed Planar Curve Entity	Polyline
43	Section Entity	Annotation Curve Occurrence and Draughting Annotation Occurrence
44	Sectioned Area Entity	Annotation Fill Area Occurrence and Draughting Annotation Occurrence
45	Spherical Surface Entity	Spherical Surface
46	Text Font Definition Entity	Draughting Pre-defined Text Font
47	Toroidal Surface Entity	Toroidal Surface
48	Trimmed Parametric Surface Entity	Curve Bounded Surface
49	Units Data Entity	Global Units Assigned Context
50	Vertex List Entity	Vertex Point
51	View Entity	Presentation View
52	Witness Line Entity	Annotation Curve Occurrence and Draughting Annotation Occurrence

reader hereafter. We would also like to point to the reader that, similar to the reading functions, there exists writing functions for the IGES I/O mechanism. The input phase creates objects for each of the entities in the directory entry section of the file, and populates the objects with appropriate attribute data. The IGES I/O is written as a C++ class called *IOIGES*. The class has private members like a pointer to the IGES file, a file position pointer, the sequence number and the directory entry index of the current record being read, and a character array to store the current line being read. Also functions to read a single record, functions to read the common data types like integer, string, double and float, form the private member functions. Other functions that are used to open, close and parse each of the sections of the IGES file are written as public members of the *IOIGES* class.

When an IGES file is read, an instance of the *IOIGES* class is created. The reader first opens the file for input and function calls are made to read each of the sections. Each record in the file is read into a character string. The length of the record is checked to ascertain that it contains exactly 80 characters. The IGES reader first parses through the start section of the IGES file. Next the global section of the file is read, an object of the type *global*, which defines all the elements in this section, is created and populated using the data being parsed. The global section defines the parameter delimiters and the record delimiters which are used throughout the file. The other data contained in the global section, as described in Section 2.2, is subsequently read.

After the global section is populated, the IGES reader parses through the directory entry section of the file. The directory entry section is parsed in two passes. In the first pass the record is read and the entity type number is read from that record. The entity number is used to create an instance of an object of that type. A Dictionary of created IGES objects is maintained. The Directory Entry number in the IGES file is converted into a string and this is used as the key to identify the objects. In the second pass the elements of the directory entry are read in. The directory entry number of the particular entry is used to index into the Dictionary, retrieve the object, and to populate its other data elements. Next, the parameter data section is parsed. The sequence num-

ber of the record in the parameter data section is used to retrieve the instance of that particular object. This is the same as the second pass of the directory entry section wherein a unique key is used to retrieve the instance, and the data in the record is used to populate the object.

Finally, the terminate section of the IGES file is read and the file closed. The IGES data is now available in the form of objects which are to be converted into AP202 objects.

5.2. IGES to AP202

Fig. 3 contains a flowchart describing the working of the translator, and elaborated in the following steps below.

(1) First the IGES file is read in using the IGES reader. Each IGES entity is now available as a persistent object in the ROSE database. Persistent objects do not cease to exist after the execution of the program is completed and hence can be written out to a physical file and transferred without loss of information.

(2) After the IGES file has been read into the ROSE library, the translator goes through each of the entities in the directory entry section of the IGES file and maps it into the corresponding AP202 entity. The IGES Schema has been written in such a way that there is a common abstract class to each of the IGES entities. This allows the use of polymorphism (as virtual functions in C++) to be used during the conversion. One important aspect worth mentioning here is that the IGES geometry entities have an interpretation flag which specifies use of the particular entity either as geometry or as annotation. If the flag is set to annotation, the mapped entity is converted into an *annotation_curve_occurrence*.

(3) At the end of stage two, all the IGES entities have been converted into AP202. The mapped AP202 shape elements are then parsed and grouped into a *geometrically_bounded_surface_shape_representation* or an *advanced_brep_shape_representation*, depending on the existence or absence of any topology information in the file. The *geometrically_bounded_surface_shape_representation* or the *advanced_brep_shape_representation* are finally connected back to a *draughting_model*. The *draughting_model* is the main collection point for the geometry in the AP202 STEP file and can be conceptu-

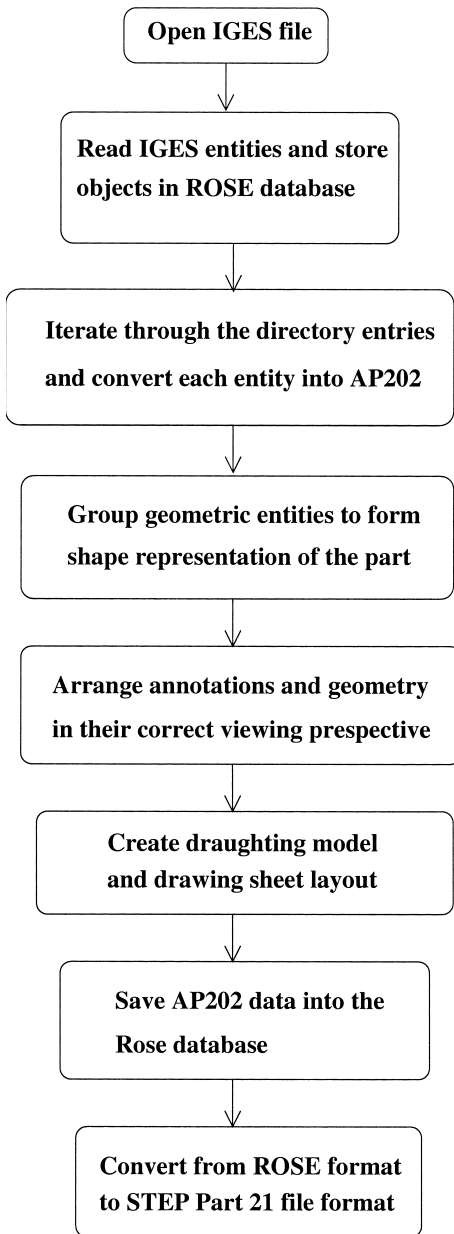


Fig. 3. Flow chart of IGES to AP202 translation.

ally thought-of as the abstract representation of the drawing sheet.

(4) AP202 uses the concept of Model-View-Draw coordinates for the representation. The geometry is represented in the model space coordinates where

one point (generally one end) of the solid is used as the origin for all the geometry. The annotations are represented in their own drawing space called the view space coordinates. And the drawing sheet level elements (such as the border of the drawing sheet, the identification box etc.) are represented in the drawing space coordinates. To complete the translation and create a valid AP202 STEP file, the annotations are arranged into their respective views and the geometry is clipped onto the particular views, converting it from Model space to View space using a *camera_image_3D_with_scale*. The drawing sheet layout is created in the drawing space coordinate system and all the views are transformed into this coordinate system. This final step is very important because the proper clipping of the geometry into the drawing space coordinates and associating the annotations with the specific view is essential for complete definition and eventual correct viewing of the drawing sheet.

(5) The STEP data is then, saved in the ROSE database format and can be converted into a STEP Part 21 file format using a format utility provided with the ST-Developer software by STEP Tools Inc. [18].

The translator has been tested and checked using sample IGES files. Files tested include those obtained from National Institute of Science and Technology (NIST) [7], WIZ WORKX [20] and Parametric Technology Corp. [10].

5.3. Visualizer

After the IGES to AP202 translator had been completed it was necessary to write our own visualizer due to the absence of a commercially available AP202 viewer. The visualizer serves the dual purpose of: (i) validating the generated STEP file, and (ii) displaying the drawing. The discussion on the Visualizer is included for the sake of completeness.

We decided to use Open Inventor 3D Graphic language [19] toolkit on an SGI platform to write our visualization software. In Inventor, scene graphs used to render images on the screen are arranged in the form of a tree. Each branch of the tree contains information to describe a particular entity on the

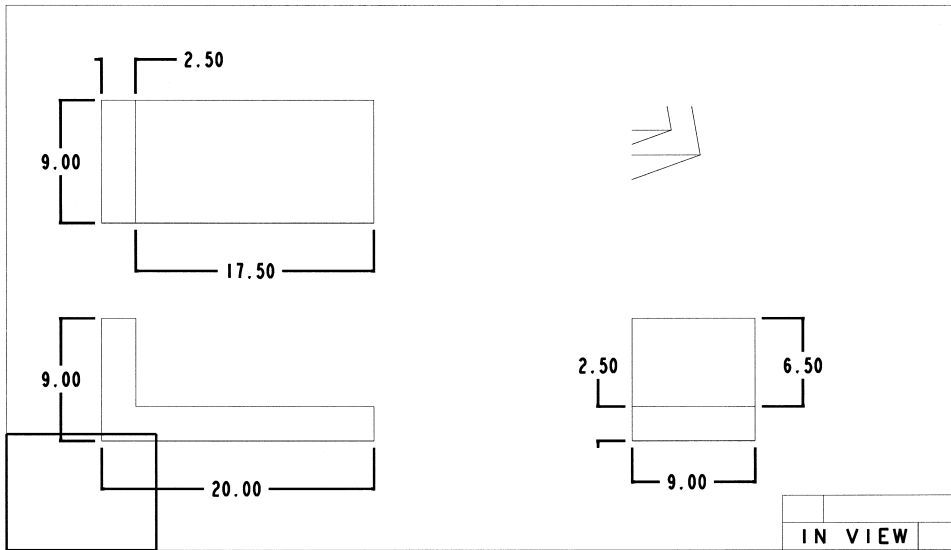


Fig. 4. View of IGES file from the IGES manual.

scene graph. All these branches are attached to a parent node which is the starting point for the traversal

of the graph for rendering. Inventor can be used for 3-dimensional visualization of curves, surfaces,

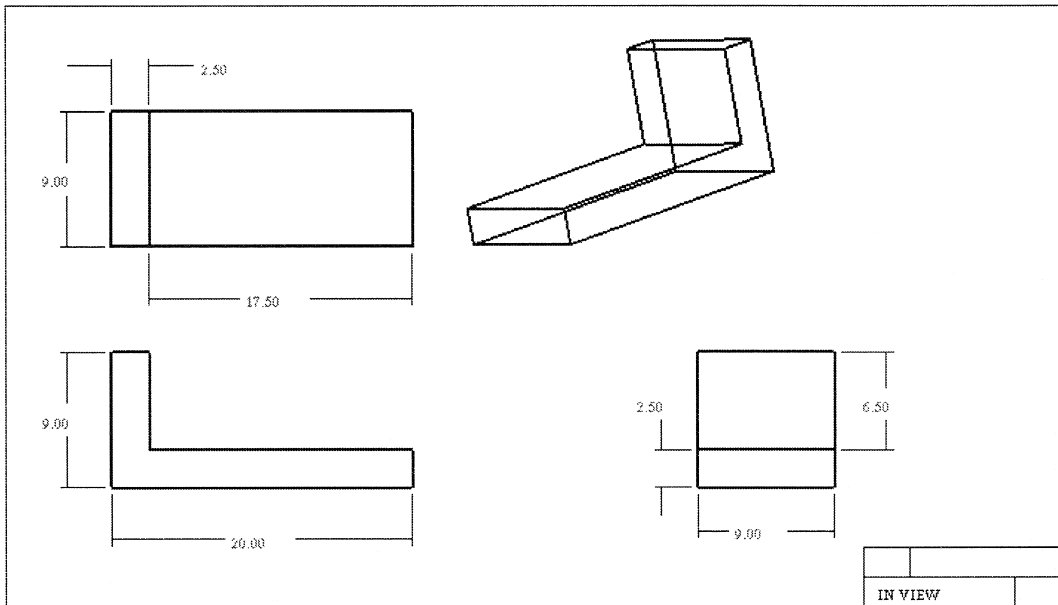


Fig. 5. View of converted AP202 file.

textures, text and camera views, thus it suits our requirements quite well. The way the visualizer works is listed below.

(1) The STEP file is read into the ROSE library so that it can be easily manipulated.

(2) After the file is read into memory, all the geometry and the topology information is converted into the Inventor format. All the geometry is converted into Non-Uniform Rational B-Splines (NURBS) curves and surfaces. This is because Inventor uses NURBS to render curves and surfaces. We have used the DT-NURBS Fortran library developed by Boeing Information and Support Services for the US Navy [5]. This library is used to convert lines, arc, circles, conics and surfaces into NURBS. The geometry is then converted into an Inventor node.

(3) At the next step we convert all the views and their view dependent annotations into corresponding Inventor nodes. The geometry node is then transformed and added below the annotation nodes. This essentially performs the task of converting the geometry from drawing space coordinates into view space coordinates.

(4) The Inventor file is written out as a “.iv” file which can then be visualized using a the *ivview* or *sceneviewer* programs provided with the Inventor toolkit.

Most of the files which have been converted into AP202 have been converted into the Inventor format and visualized. Fig. 4 shows a view of a sample IGES file which can be found at the end of the IGES manual [7], and the view of the corresponding STEP file is as shown in Fig. 5.

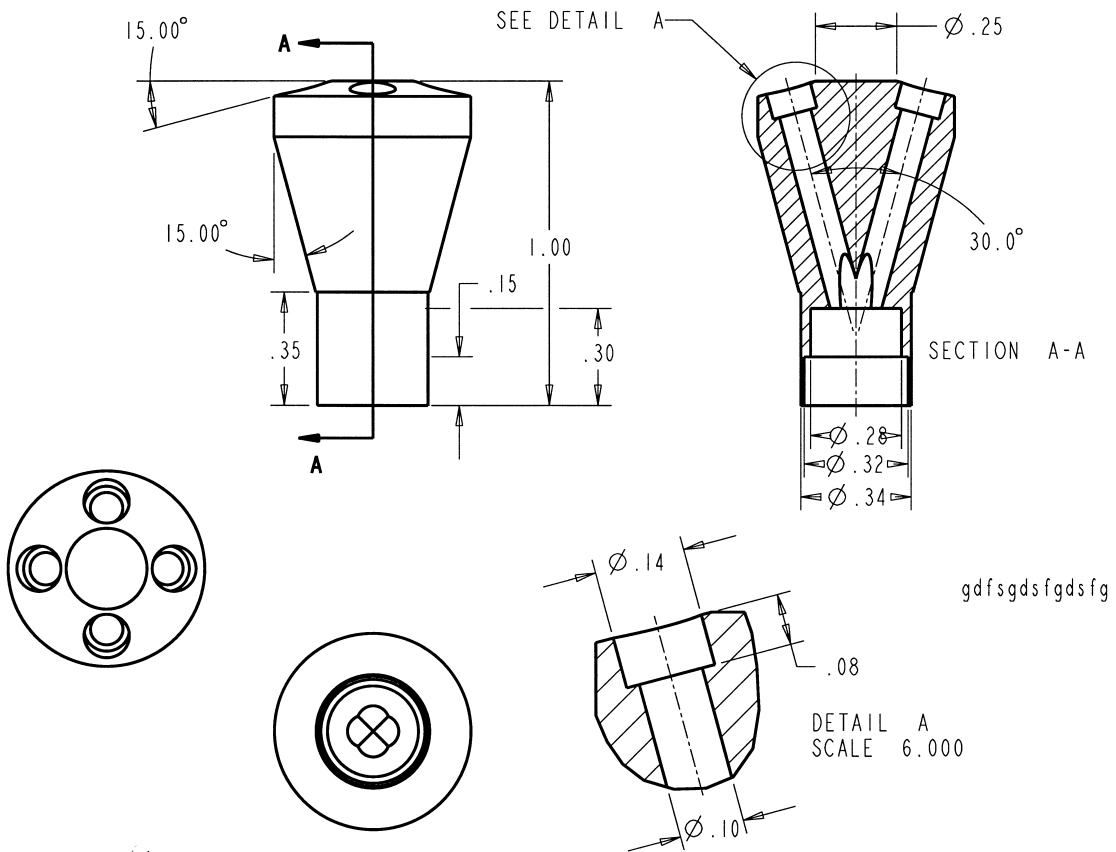


Fig. 6. View of a nozzle in the IGES format.

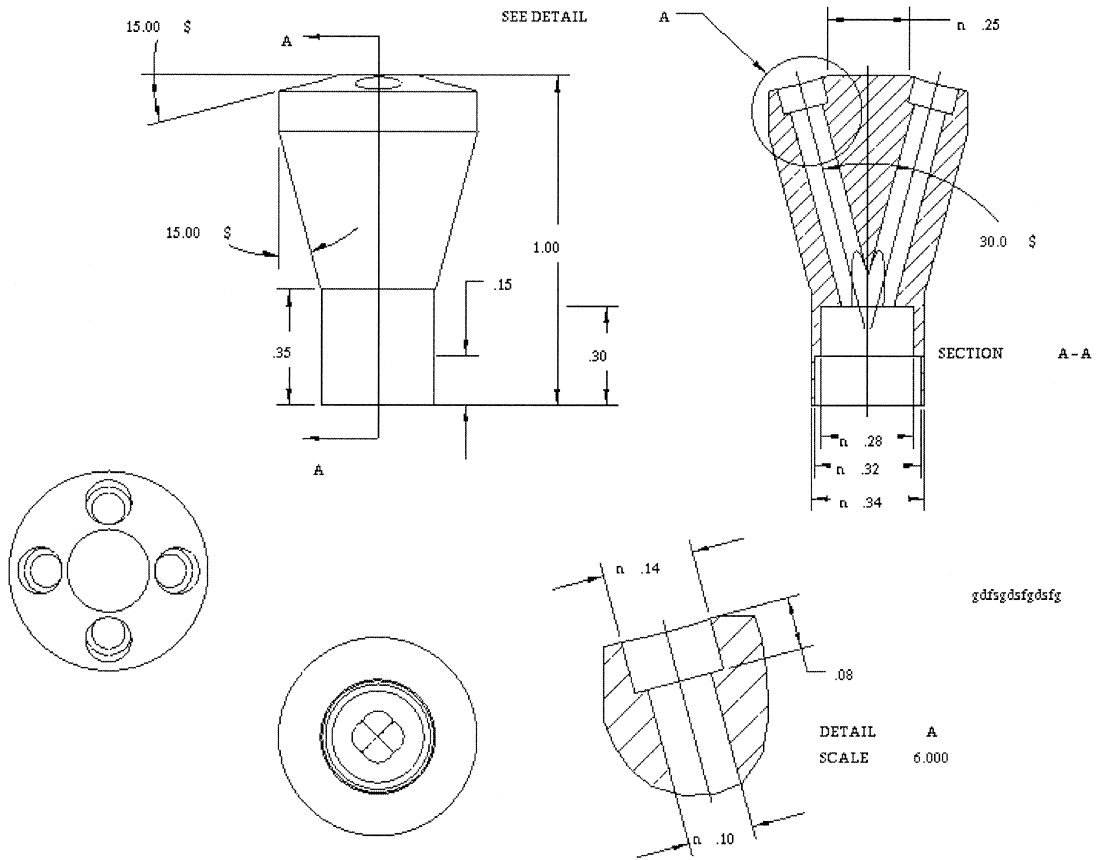


Fig. 7. Converted nozzle file in the AP202 format.

It can be seen that the clipped view of the geometry appears complete in the STEP view. This is because clipping is done in Inventor using a camera, however only a single camera can be applied to an Inventor window. This would not allow us to see the rest of the sheet. We are currently trying to work our way around this problem to allow viewing of clipped views in our visualizer. Fig. 6 shows the view of a nozzle from an IGES file obtained from Parametric Technology Corp. (PTC) [10] and Fig. 7 shows the corresponding STEP file. The AP202 file has been sent to PTC for testing of AP202 support for the latest release of PRO/Engineer. It can be seen that the symbol “ ϕ ”, which is used as a diameter symbol in draughting terminologies and the “ \circ ” symbol appearing after the angular dimensions are not visible in Fig. 7. This is because the symbols are special characters supported by IGES (not supported in STEP

directly) and which are to be interpreted by the processor during display of the drawing. This can be done with the help of STEP *pre_defined_symbol* entity and by converting the IGES symbols into one of these types. These STEP symbols have to be then interpreted by the visualizer at run time and the appropriate symbol should be displayed. The symbols would depend on the type of fonts supported by the users machines. However, since the display of symbols is system dependent it has not been incorporated into the visualizer. It must be noted that these limitations are only for the viewer, while the STEP translation is complete.

6. Conclusion and further work

In the agile and virtual manufacturing environment, the STEP international standard for product

data exchange is becoming increasingly popular over the legacy format such as IGES. IGES has been found to be restrictive because it does not capture data about the product through its life-cycle. It is expected that in the long run STEP would completely replace IGES. The issue of migration from IGES to STEP is being addressed through a migration strategy being developed by IGES/PDES Organization. We hope that the current work contributes considerably to these efforts. It is expected that this work will gain more importance once CAD vendors develop AP202 support to their software products which is imminent now that AP202 has become an international standard.

In this effort we have focussed on the translation of IGES drawing information into the STEP AP202 drafting standard. The approach consists of: (i) object-oriented modeling of IGES specification using the EXPRESS language, and (ii) one-to-one entity mapping from IGES entities to the corresponding STEP entities. The implementation is done using C++, object-oriented techniques and use of an object-oriented database called ROSE for the storing and retrieval of objects. The implementation also involves development of a visualizer to check and validate the generated STEP drafting information. The visualizer is implemented on a Silicon Graphic platform using the Open Inventor 3D graphic language.

In addition to the geometry, STEP also includes solid and feature information. We are currently working on a STEP-based feature extraction system. The idea is to analyze the STEP B-Rep data and extract out manufacturing features from the same. The form feature would be stored in the STEP format using AP224 “Mechanical product definition for process planning using form features”. The feature information can be used to develop process plans which can be directly interfaced with numerically controlled machines.

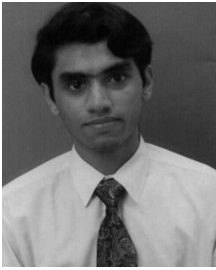
Acknowledgements

Work on the project was done partially at STEP Tools, Rensselaer Technology Park, Troy, NY and partially at the State University of New York at Buffalo. We would like to thank all personnel at

STEP Tools for all their help to complete this project. Rakesh Nagi acknowledges the support of the National Science Foundation career grant DMI-9624309.

References

- [1] Autodesk - AutoCAD, IGES Translator Release 13, Uniform resource locator (URL), <http://www.autodesk.com/products/mech/igestrans/igestrans.htm>.
- [2] D. Basu, S. Kumar, Importing MESH Entities through IGES/PDES, *Advances in Engineering Software* 23 (3) (1995) 151–161.
- [3] M. Bhandarkar, Satisfying Data Needs In Agile Manufacturing Through Translation And Feature Extraction Into STEP Product Data Models, Department of Industrial Engineering, State University of New York at Buffalo, 1997.
- [4] CATIA/CADAM Solutions, URL, <http://www.catia.ibm.com/html/catmain.html>.
- [5] DT_NURBS Generalized Spline Geometry Library, URL, <http://dtnet33-199.dt.navy.mil/>.
- [6] M. Hardwick, RPI DICE team, ROSE 3.0 User Manual, Rensselaer Design Research Center, Rensselaer Polytechnic Institute, Troy, NY, 1991.
- [7] IGES 5.2 Examples and figures, URL, <http://megavolt.eeel.nist.gov/iges/specfigures/index.html>.
- [8] IGES EXPRESS model, URL, <http://prema.eng.buffalo.edu/~mpb3/IGES.exp>.
- [9] Initial Graphics Exchange Specification (IGES) Version 5.3, US Product Data Association, 1994.
- [10] Parametric Technology Corp., URL, <http://www.ptc.com> and <http://www.ptc.com/products/exchange/index.htm>.
- [11] Part 202, Associative Draughting, US Product Data Association, 1996.
- [12] EXPRESS model for AP202, URL, <http://www.nist.gov/sc4/step/parts/part202/is/>.
- [13] D.A. Schenck, P. Wilson, *Information Modeling the EXPRESS Way*, Oxford University Press, USA, 1994.
- [14] S.C. Liu, M. Gonzales, J.G. Chen, Development of automatic part feature extraction and classification system taking CAD data as input, *Computers in Industry* 29 (1996) 137–150.
- [15] Silicon Graphics, Application Directory: IGES to Inventor, URL, http://www.sgi.com/Products/appsdirectory.dir/Applications/Data_Analysis/ApplicationNumber5794.html.
- [16] M.E. Ssemakula, A. Satsangi, Application of PDES to CAD/CAPP integration, *Computer in Industrial Engineering* 18 (4) (1990) 435–444.
- [17] STEP on a page, URL, <http://www.nist.gov/step/soap/soapgraf.htm>.
- [18] STEP Tools Inc., URL, <http://www.steptools.com>.
- [19] J. Wernecke, *The Inventor Mentor: programming Object-oriented 3D graphics with Open Inventor*, release 2, USA, 1994.
- [20] The WiZ WORX homepage, URL, <http://www.wiz-worx.com/wizworx.htm>.



Mangesh P. Bhandarkar is a software engineer with Inso, PDM Division. He represents Inso at the Object Management Group and is a co-author of the PDM-Enabler specification. He is currently a member of the finalization task force for the specification. He has an MS in Industrial Engineering from SUNY Buffalo (1997) and BS in Production Engineering from University of Bombay, India (1995).



Rakesh Nagi is an Associate Professor of Industrial Engineering at the State University of New York, Buffalo. His recent research interests are in Agile Enterprises, Information-Based Manufacturing, and Logistics and Supply Chain Management. He was awarded the National Science Foundation's faculty early career award in 1996, SME's Milton C. Shaw Outstanding Young Manufacturing Engineer Award of 1999 and IIE's Outstanding Young Industrial Engineer in Education Award of 1999.



Blair R. Downie is a research scientist in the Center for Advanced Technology in Automation, Robotics, and Manufacturing located on the campus of Rensselaer Polytechnic Institute (RPI) in Troy, New York. He also serves as a consultant for STEP Tools, an industry leader in the development of software tools for the emerging ISO STEP standard. He received his master's degree in Computer Science from RPI in 1987.



Martin Hardwick is a professor of computer science at RPI and the director of laboratory for Industrial Information Infrastructure. He is also the president of STEP Tools. Due to his intimate understanding of the ISO STEP standard he is a member of the board of PDES and the US Product Data Association. He has a PhD in mathematics and computer science from Bristol University, UK.