# *On Extending the Standard for the Exchange of Product Data to Represent Two-Dimensional Apparel Pattern Pieces*

by
Y. Tina Lee
National Institute of Standards and Technology
June, 1990

## ABSTRACT

An Apparel Pattern Information Model (APIM) is introduced to demonstrate the feasibility of extending the emerging international Standard for the Exchange of Product Data (STEP) to include the exchange of apparel pattern data. This paper focuses on a representation of two-dimensional (flat) patterns. We show how this representation is capable of capturing the same information that can be expressed in one widely-used, but proprietary, format.

## 1 INTRODUCTION

The apparel industry is increasingly reliant on computer technologies to support activities throughout the product life cycle. As computers become more widely used in design, pattern making, production, and even equipment manufacture, the industry is becoming more aware of the potential benefits of exchanging information electronically. There are already national and international efforts to develop standards for electronic ordering, scheduling, billing, and other life-cycle activities.

In the U.S., efforts have recently focused on developing standards to support the technological areas of apparel manufacturing. In September 1988, the American Apparel Manufacturers Association (AAMA) authorized an effort "to facilitate implementation of Computer Integrated Manufacturing (CIM) in the apparel industry through the establishment of voluntary standards for communication among merchandising manufacturing-related functions" with the formation of a Computer Integrated Manufacturing Committee (CIM/COM). At about the same time, the Department of Defense (DoD) began a program within the Defense Logistics Agency (DLA) aimed at improving the technology used in procuring uniforms, combat gear, and other apparel products used by the military. As part of this effort, the DLA asked the National Institute of Standards and Technology (NIST) to help develop product data exchange standards for apparel manufacturing. This report documents one aspect of the NIST work: a prototype specification for representing flat pattern data in a neutral, STEP format.

The STEP is an emerging standard, the result of an international effort to develop a mechanism for the exchange of product data among all stages of the life cycle of a product. In the U.S., an effort called Product Data Exchange using STEP (PDES), involving hundreds of individuals and companies, is aimed at developing the technology needed to implement the STEP objectives. The applications addressed by PDES to date include mechanical piece parts, mechanical assemblies, architecture engineering construction models, finite element models, manufacturing applications, drafting applications, and electrical printed wiring board products, including both schematic and physical designs [SMIT89]. PDES is a key standard in the DoD Computer-aided Acquisition and Logistic Support (CALS) program, which seeks to establish an integrated set of standards and specifications for the creation, management, and exchange of development and logistic data by computer. NIST has, under sponsorship of the Office of the Secretary Defense (OSD), recently established the National PDES Testbed to support industry and government projects in developing and testing PDES [FURL89].

## 2 APPAREL PATTERN INFORMATION MODEL

PDES is founded on the use of information models to define the data types that can be used to define products. These information models are written in a formal language called *Express* [SCHE89]. In this section, we introduce the *Express* model of apparel product data. The *Express* information modeling language was developed by the IGES/PDES Organization as a way to precisely and completely describe all the data elements for defining objects. The exchange medium for actual PDES product models is the STEP physical file. A STEP physical file contains instances of the various entities defined by the *Express* information model. The STEP exchange format, and the mapping from *Express* to the STEP physical file, are described in "The STEP File Structure" [NIST88] and "Mapping from *Express* to Physical File Structure" [NIST88]. These methodologies form the basis of Apparel PDES (APDES).

The Apparel Pattern Information Model (APIM) is an information model for APDES. This model describes the pattern pieces of an apparel design. The APIM model presented in this paper is not intended as a complete apparel pattern data exchange standard; rather, our purpose is to demonstrate the feasibility of neutral data exchange of two-dimensional design pattern data using the PDES methodology. A plan for a full implementation of APDES is currently under development.

An *Express* schema is composed of declarations of types, entities, constraints, and their relationships. The APIM schema is explained in detail in Section 3. The following indented list provides the classification structure of the entities defined in this model.

TWO D PATTERN

PATTERN PIECE

GRADE RULE

GRADE DATA

TWO D DELTA COORDINATE

BASE PATTERN

GRADE POINTS

TWO D COORDINATE

PATTERN GEOMETRY ENTITY

MARK FEATURE

COMPOSITE CURVE FEATURE

ORIENTATION CONSTRAINT

ANCILLARY FEATURE

LINE

PATTERN SIZE

MENS PATTERN SIZE

WOMENS PATTERN SIZE

As can be seen, the entities include a broad range of data types, from simple two-dimensional coordinates to complex entities such as patterns. The way these entity classes are related is specified by the APIM model schema, which is described next.

# 3 APPAREL PATTERN SCHEMA

This section describes the detailed information for the APIM schema. Types and entities are defined formally here in *Express*. The concept of a type in *Express* is the same as that of a data type in a standard programming language. It defines the kind of values that an object may assume. Entities are the focal point of an *Express* information model. An entity declaration describes the information content of an object, as well as some of the constraints on the object. The schema is presented here in a "bottom-up" order: primitive type definitions are presented first, followed by entity definitions. The more specific entity models are described before they are used in the definition of more complex entities.

## 3.1 Type Definitions

This section contains the type definitions which may be common to more than one of the schemas.

### 3.1.1 Men's Size Group

Men's wear sizes are divided into six groups based on length: Extra Short, Short, Regular, Long, Extra Long, Big. *Men's size group* is an enumeration of these six groups. It is used as

```
TYPE mens_size_group = ENUMERATION OF
        ( extra_short,
```

```
                    short,
                    regular,
                    long,
                    extra_long,
                    big );
 END_TYPE;
```

### 3.1.2 Women's Size Group

Women's wear sizes are divided into seven groups based on proportion: Junior Petite, Junior, Miss Petite, Miss, Miss Tall, Women, Women Petite. *Women's size group* is an enumeration of these seven groups. It is used as the type of an attribute defined in the *women's pattern size* entity.

```
TYPE womens_size_group = ENUMERATION OF
                   ( junior_petite,
                    junior,
                    miss_petite,
                    miss,
                    miss_tall,
                    women,
                    women_petite );
END_TYPE;
```

### 3.1.3 Mark Feature Type

A mark or a set of marks may be placed on the pattern to indicate the positions of darts, gathers, and other design features. *Mark feature type* is an enumeration of Drill Hole and Notch. It is used as the type of an attribute defined in the *mark feature* entity.

```
TYPE mark_feature_type = ENUMERATION OF
                   ( drill_hole,
                    notch );
END_TYPE;
```

### 3.1.4 Composite Curve Feature Type

*Composite curve feature type* provides a means of expressing the purpose of a curve on a pattern. This type is an enumeration of Boundary Cut, Internal Cutout, and Sew Line. It is used as the type of an attribute defined in the *composite curve feature* entity.

```
TYPE composite_curve_feature_type = ENUMERATION OF
        ( boundary_cut,
          internal_cut,
          sew_line );
END_TYPE;
```

### 3.1.5 Orientation Constraint Type

*Orientation constraint type* provides a means of specifying the orientation of the pattern piece on the fabric. The pattern piece may be aligned with the fabric's grain, or with some feature of the fabric's decorative design. *Orientation constraint type* is an enumeration of Grain Line, Stripe, and Plaid. It is used as the type of an attribute defined in the *orientation constraint* entity.

```
TYPE orientation_constraint_type = ENUMERATION OF
        ( grain,
          stripe,
          plaid );
END_TYPE;
```

### 3.1.6 Ancillary Feature Type

*Ancillary feature type* provides a means of expressing the annotation on the drawing. It is an enumeration of Annotation, Style Line, and Grade Reference. *Ancillary feature type* is used as the type of an attribute defined in the *ancillary feature* entity.

```
TYPE ancillary_feature_type = ENUMERATION OF
        ( annotation,
          style,
          grade_reference );
END_TYPE;
```

### 3.1.7 Grade Line Option Type

Grade line options include turning automatic smoothing on at the grade point and turning automatic smoothing off at the grade point. *Grade line option type* is an enumeration of Smooth and No Smooth. It is used as the type of an attribute defined in the *grade data* entity.

```
TYPE grade_line_option = ENUMERATION OF
        ( smooth,
          no_smooth );
END_TYPE;
```

### 3.1.8 Measurement Unit Type

The units of length include inch and centimeter. *Unit* is an enumeration of Inch and Cm.

```
TYPE unit = ENUMERATION OF
        ( inch,
          cm );
END_TYPE;
```

### 3.1.9 Pattern Type

*Pattern type* is an enumeration of Basic, Horizontal Mirror, and Vertical Mirror. A mirror pattern is a mirror-image of the basic pattern in the horizontal or vertical direction. *Pattern type* is used as the type of an attribute defined in the *pattern piece* entity.

```
TYPE pattern_type = ENUMERATION OF
        ( basic,
          horizontal_mirror,
          vertical_mirror );
END_TYPE;
```

## 3.2 Entity Definitions

This section defines the entities for the apparel pattern schema. All pattern geometry is defined in a Cartesian coordinate system.

### 3.2.1 Pattern Size

A *pattern size* entity is an abstraction of *men's pattern size* and *women's pattern size*. Both sub-types inherit the integer *size number* attribute. The XOR keyword indicates that a pattern size must be either a men's size or a women's size, but not both.

```
ENTITY pattern_size
  SUPERTYPE OF
    ( mens_pattern_size     XOR
      womens_pattern_size );
   size_number: INTEGER;
END_ENTITY;
```

### 3.2.2 Men's Pattern Size

A *men's pattern size* entity defines the size number ( e.g., 34, 42) and the men's wear size group.

```
ENTITY mens_pattern_size
  SUBTYPE OF ( pattern_size );
   size_group: mens_size_group;
END_ENTITY;
```

### 3.2.3 Women's Pattern Size

A *women's pattern size* entity defines the size number and the women's wear size group.

```
ENTITY womens_pattern_size
  SUBTYPE OF ( pattern_size );
   size_group: womens_size_group;
END_ENTITY;
```

### 3.2.4 Two-Dimensional Coordinate

A *two-d coordinate* entity specifies a point or location on a drawing. It consists of an *X* value and a *Y* value. Coordinates are defined from an unspecified origin, determined by an application.

```
ENTITY two_d_coordinate;
    x: REAL;
    y: REAL;
END_ENTITY;
```

### 3.2.5 Grading Points

Points on a pattern may or may not be subject to a grading rule. A *grade points* entity defines an ordered collection of points which are subject to a grading rule.

```
ENTITY grade_points;
    locations: LIST [0:#] OF UNIQUE two_d_coordinate;
END_ENTITY;
```

### 3.2.6 Line

A *line* entity consists of two points.

```
ENTITY line;
    location1, location2 : two_d_coordinate;
END_ENTITY;
```

### 3.2.7 Pattern Geometry Entity

A *pattern geometry entity* is an abstraction of *mark feature* entity, *composite curve feature* entity, *orientation constraint* entity, and *ancillary feature* entity.

```
ENTITY pattern_geometry_entity
    SUPERTYPE OF ( mark_feature XOR
                   composite_curve_feature XOR
                   orientation_constraint XOR
                   ancillary_feature );
END_ENTITY;
```

### 3.2.8 Mark Feature

A *mark feature* entity is a drill hole or a notch on the pattern piece. It is a point together with a mark feature type.

```
ENTITY mark_feature
    SUBTYPE OF ( pattern_geometry_entity );
    type: mark_feature_type;
    location:  two_d_coordinate;
END_ENTITY;
```

### 3.2.9 Composite Curve Feature

A *composite curve feature* entity is a geometry entity that defines a curve in the drawing. It is a composite curve (an ordered sequence of curves sharing common end points along the sequence), together with a composite curve feature type. A composite curve is represented by a start grading point, zero or more non-grading points, and an end grading point.

```
ENTITY composite_curve_feature
    SUBTYPE OF ( pattern_geometry_entity );
    type: composite_curve_feature_type;
    composite_curve: LIST [2:#] of UNIQUE  two_d_coordinate;
END_ENTITY;
```

### 3.2.10 Orientation Constraint

A*n orientation constraint* entity is a direction specification on the pattern piece. It is characterized by a *line* that defines the location of the orientation on the drawing as well as the orientation of the constraint. It is characterized by an orientation constraint type.

```
ENTITY orientation_constraint
    SUBTYPE OF ( pattern_geometry_entity );
    type: orientation_constraint_type;
    location :  line;
END_ENTITY;
```

### 3.2.11 Ancillary Feature

An *ancillary feature* entity is a graphical feature on the pattern piece presented for informational purposes; it does not represent a feature of the cut piece. It consists of a *line* for locating and orienting text, and is characterized by an ancillary feature type.

```
ENTITY ancillary_feature

    SUBTYPE OF ( pattern_geometry_entity );

     type: ancillary_feature_type;

     text: STRING;

     location: line;

 END_ENTITY;
```

### 3.2.12 Base Pattern

A *base pattern* entity defines the base shape (the shape of one selected size) of a particular pattern piece for a particular garment. It has identification information, the *pattern size* it defines, and a set of *pattern geometry entities*. The identification information includes a piece name and a description. The piece name is a unique pattern piece name in the system. The description is a pattern identification for the operator to use. The base pattern may define the shape of several pattern pieces in the garment (e.g., mirrored pieces).

```
ENTITY base_pattern;

    piece_name: STRING;         (* Piece name (pn) is a unique pattern
                                 piece name in the system. *)

    description: STRING;        (* The description is a pattern identification
                                 for operator to use. *)

    base_size: pattern_size;

    grade_point_set: grade_points;

    pattern_piece_lines: LIST [2:#] OF pattern_geometry_entity;

   UNIQUE   pn;

END_ENTITY;
```

### 3.2.13 Two-Dimensional Delta Coordinate

A *two-d delta coordinate* entity specifies a displacement vector on a drawing. It consists of a *delta-X* value and a *delta-Y* value. The *delta-X* and *delta-Y* values are the amounts of growth in the *X* and *Y* directions at the grading point for two grading sizes.

```
ENTITY two_d_delta_coordinate;
    delta_x: REAL;
    delta_y: REAL;
END_ENTITY;
```

### 3.2.14 Grading Data

A *grade data* entity is an ordered collection of the displacements of the specified grading point for a set of predefined grading sizes. It is defined by a grading point, a grade line option (which turns automatic smoothing on or off at the indicated grading point), and a list of *two-d delta coordinates.*

```
ENTITY grade_data;
    grade_point: two_d_coordinate;
    option: grade_line_option;
    grade_factors: LIST [1:#] OF two_d_delta_coordinate;
END_ENTITY;
```

### 3.2.15 Grading Rule

A *grade rule* entity is a rule library created for storing all grading sizes and *grade data* for each grading point of a pattern.

```
ENTITY grade_rule;
    break_sizes: LIST [0:#] OF UNIQUE pattern_size;
                          (* The break sizes define all grade
                           breaking sizes that include the base size. *)
    grade_data_collection: LIST [0:#] OF grade_data;
END_ENTITY;
```

### 3.2.16 Pattern Piece

A *pattern piece* entity is a representation of the shape of one piece of a *two-d pattern* for all sizes of the garment. A *pattern piece* is represented in terms of a *base pattern,* a pattern type, and, optionally, a *grade rule*. The *base pattern* and *grade rule* are defined with a common unit of measurement.

```
ENTITY pattern_piece;

    unit_type: unit;

    piece_type: pattern_type;

    base_size_pattern: base_pattern;

    grade_rule_library:  OPTIONAL grade_rule;

END_ENTITY;
```

### 3.2.17 Two-Dimensional Pattern

A *two-d pattern* entity belongs to a garment and defines the garment shape for all sizes. It is defined by a garment name, a garment description for the operator to use, and a group of *pattern piece*s.

```
ENTITY two_d_pattern;

    garment_name: STRING;

    description: STRING;      (* The description is a garment identification
                                 for operator to use. *)

    pattern_pieces: LIST [1:#] OF pattern_piece;

END_ENTITY;
```

# 4 APDES EXCHANGE FILE

This section identifies the physical file structure for an APDES exchange file. A sample APDES exchange file is presented in Section 4.2.

## 4.1 APDES File Structure

As mentioned previously, all APDES exchange files will be in the STEP file format. STEP is used to represent product-definition data. It has been designed to meet both present and future requirements for exchanging such data between dissimilar CAD/CAM systems. The STEP physical file is a sequential file and is organized in a modular manner. The file consists of two sections: the HEADER section and the DATA section. These sections may contain one or many entities. An entity consists of attributes; each attribute is a fact (datum) about an entity.

## 4.2 Sample APDES Exchange File

A sample pattern is provided for illustrating an example of a STEP physical file. Figure 1 presents the sketch of this sample pattern (pattern piece name = '1004'). This pattern was modeled on a Gerber Garment Technology's AM-5 system. The pattern data was extracted into a *piece number* (PN) file in a format specified by Gerber [GGT88]. Appendix A contains the listing of the Gerber PN file for the sample pattern. The PN file contains both the pattern data for the base pattern as well as grading information for that pattern. The same garment was represented in a STEP physical file based on the APDES schema presented above. Appendix B contains the listing of the STEP physical file. The STEP physical file contains all the information contained in the PN file, as well as more detail about the sizing characteristics of the pattern.

Each STEP physical file may comprise one or more pattern pieces. The STEP file format may be considered a continuous stream; however, spaces and lines have been inserted into the sample file to aid readability. Each APDES/STEP entity is identified by a unique entity name, which is an unsigned integer of 1 to 9 digits. An APDES/STEP entity identifier consists of an "at" sign (@) followed by an entity name, and an APDES/STEP entity reference consists of a "number" sign (#) followed by an entity name.

# 5 DISCUSSION

Gerber systems are widely used for the design, grading, and marking of patterns. It was felt that it was important to demonstrate the capabilities of APDES as compared to the Gerber PN file format. The Gerber PN file format has several attractive features: it has a simple file structure and compact file size, it is easy to parse, it can be edited with a standard text editor, and it has a grading rule feature. However, this file format specification is not in the public domain. More importantly, only a limited number of graphic entities can be represented. The measurement precision is also limited by its fixed format to represent numerical values. The PN file format provides only the representation of two-dimensional patterns. The APDES, when completely designed, will provide product representations of both patterns and complete garments to support computer integration of the apparel life cycle.

One may notice from Appendix A and Appendix B that the file sizes have been changed tremendously as a result of conversion from the sample Gerber PN file to the STEP physical file. The sample PN file is 1420 bytes long; however, the sample STEP physical file is about 12500 bytes long. Also, the file grew from one page of data to five pages. The large STEP physical file size is due to several factors. As mentioned previously, the STEP physical file should be a continuous stream of characters, but spaces have been inserted into the sample file to aid readability. The main reason, however, is that a penalty is being paid for generality. Some entities, which are not supported by the PN file, have been added to the APIM model to enhance its flexibility and completeness. For example, the pattern size in Gerber PN file is limited to digits only, whereas the pattern size defined in APIM model includes both men's and women's size numbers and size groups. Other size types, such as infants', babies', toddlers', children's, boys', girls' and young men's, are not included here; they will be included in the future version of the APIM model. The major shortcoming of APDES is the large file size. It is expected that this can be overcome by using a standard called Abstract Syntax Notation (ASN.1) to convert the file from text to binary format.

## 6 SUMMARY

This report describes the approach being taken by NIST in developing a neutral format for exchanging apparel product data. We have developed a prototype specification of a format based on the emerging Standard for the Exchange of Product Data (STEP) and have demonstrated that this approach has at least the power to represent the same data as one popular, although proprietary, file format.

The APDES specification will continue to evolve based on experience and feedback from others involved in this effort. Current plans are to demonstrate the use of APDES first for archiving data from a single system, then for exchanging data between two similar systems, and finally, for exchanging two-dimensional pattern data between dissimilar systems. A final version of Apparel Pattern Information Model (APIM) for two-dimensional patterns will be developed by the fall of this year. Work will then focus on extending APDES to include three-dimensional garment models, linkages to textile, anthropometric, and other data, and additional information needed to support the apparel manufacturing life cycle.

**Figure 1. Sketch of "Pattern Piece 1004 "**

# 7 REFERENCES

[FURL89] Furlani, C. M., "The National PDES Testbed - An Overview," UPCAEDM ’89, NIST, July, 1989

[GGT88] Gerber Garment Technology, Inc., "AM-5/Gerber 400 Training Work- book," Document No. 052298001, Issue No.2, GGT, January 1988

[NIST88] National Institute of Standards and Technology, "Product Data Exchange Specification - First Working Draft," NISTIR 88-4004, NIST, December, 1988

[SCHE89] Schenck, D., ed., "Information Modeling Language Express: Language Reference Manual," ISO TC184/SC4/WG1 Document N362, May 1989

[SMIT89] Smith, B ., "Product Data Exchange: The PDES Project - Status and Objectives," NISTIR 89-4165, NIST, September, 1989