

# Driving PWA Thermomechanical Analysis from STEP AP210 Product Models

Diego R. Tamburini<sup>1</sup>, Russell S. Peak<sup>2</sup>, Robert E. Fulton<sup>1</sup>  
Engineering Information Systems Laboratory (<http://eislabs.gatech.edu>)  
<sup>1</sup>School of Mechanical Engineering  
<sup>2</sup>CALS Technology Center  
Georgia Institute of Technology  
Atlanta, GA

## ABSTRACT

This paper describes experiences using the STEP AP210 Printed Wiring Assembly (PWA) product model to drive engineering analysis. It describes how this product design data is mapped into an intermediate analyzable representation (the Analyzable Product Model, or APM) which supports the information requirements of several thermomechanical analyses, including product idealizations. Examples from the DARPA-sponsored TIGER project are included, in which AP210 models were generated from the Mentor Graphics board layout tool. The paper describes other issues encountered such as how to integrate product data that spans more than one tool, how to add missing analysis data that is not generated during design, and how to use data stored in the APM from different programming environments.

Experiences show the value of semantically rich product models like STEP AP210 for analysis integration (vs. straight geometry-oriented models like AP203). However, the multi-fidelity idealization nature of analysis leads to an insatiable information appetite that no product model, no matter how rich, can continually satisfy. Thus, the APM technique is necessary as a general link to design tools in order to harmonize diverse data and add idealizations and missing data. Overall, experiences in TIGER confirmed the basic thrusts of the APM approach and its usage of STEP.

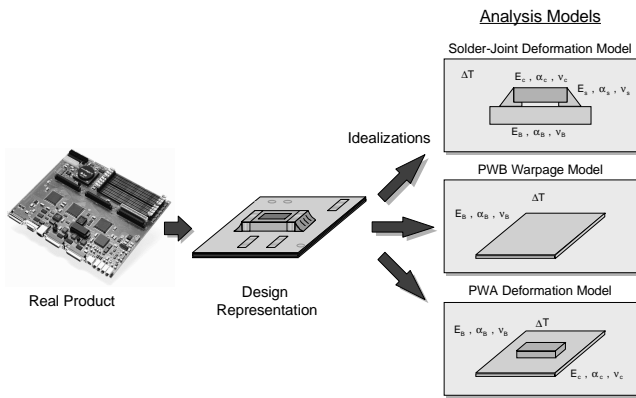
## NOMENCLATURE

AP	Application Protocol
APD	Analyzable Product Database
APM	Analyzable Product Model
DAI	Design-Analysis Integration
ECAD	Electrical Computer-Aided Design
ISO	International Standards Organization
MRA	Multi-Representation Architecture
PBAM	Product Model-Based Analysis Model
PCA	Printed Circuit Assembly

PCB	Printed Circuit Board
PIT	Product Idealization Transformation
PM	Product Model
PTH	Plated-Through Hole
PWA	Printed Wiring Assembly
PWB	Printed Wiring Board
STEP	Standard for the Exchange of Product Data
TIGER	Team Integrated Electronic Response
$\alpha_c$	Component's Coefficient of Thermal Expansion

## 1 DESIGN-ANALYSIS INTEGRATION USING PRODUCT DATA EXCHANGE STANDARDS

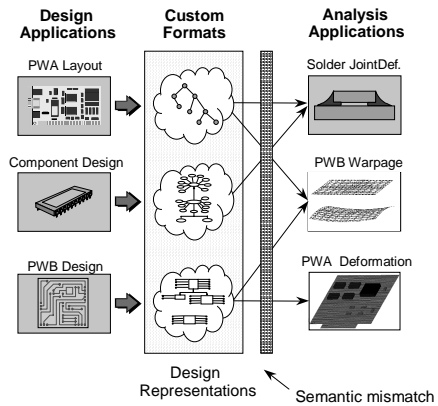
During the computer-aided development of a product, the primary task of design engineers is to create a detailed electronic description that contains enough information to support several aspects of its life cycle. At certain stages of the product development cycle, this design representation is used to drive a series of engineering analyses that validate the design against several criteria and help predict the behavior of the product under various circumstances. In order to perform these analyses, the design representation must be first idealized and transformed into some form that admits mathematical evaluation. We will refer to this last form as "analysis representations", "analysis models" or, more specifically, as "product-based analysis models" (to differentiate them from *generic* analysis models, which are not linked to any particular product). Next, analysis applications provide the necessary interfaces to enable user interaction with these analysis models. Figure 1 illustrates this design-idealize-analyze sequence with an example from thermomechanical analysis of printed wiring assemblies.



**Figure 1:** Design-Analysis Integration Example

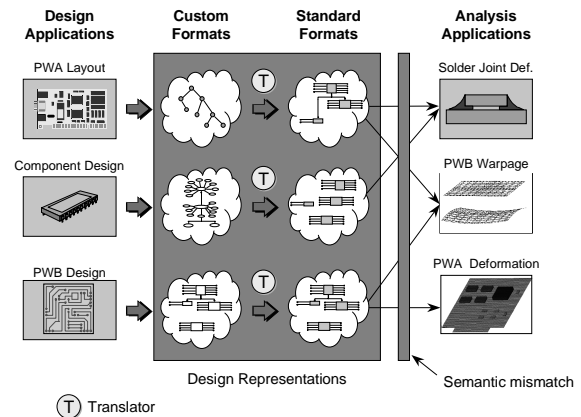
Although both the design and the analysis representations are views of the same product, they describe it at very different levels of semantic content, and obtaining the latter from the first is normally a difficult task. Hence, design-analysis integration often turns out to be a difficult proposition. The unstructured development of computer-aided design and analysis systems over the years has made it difficult to integrate both the systems themselves and the information they manipulate (Brooke et al. 1995). As a result, even though there is a large number of sophisticated computer aided engineering tools available, the current status is that typically design and analysis software tools are not compatible enough to exchange data directly - without cumbersome (manual or semi-automatic) transformation (Kemper and Morkotte 1994). Tools interface with their own proprietary data formats, which are not accessible by other applications. In many cases, data needed by the analysis models has to be manually retrieved and re-inputted in some other computer application for analysis. In addition, due to the large syntactic and semantic gap between design and analysis representations, some raw design information must undergo significant transformation, simplification and/or idealization before being fed into the analysis models on which the analysis applications are based (Armstrong 1994; Shephard et al. 1990). This is usually a tedious, slow, and error-prone process that illustrates the infamous “islands of automation”.

Added to these incompatibility problems is the fact that, in a real scenario, the development of a product requires participation of designers from several disciplines who use a wide variety of independent software systems. These multiple design applications generate a large and complex aggregation of diverse design information, scattered across several data sets with different, often proprietary, formats and data structures. As a result, there is rarely a single, integrated source of analysis information readily available. Moreover, this disjoint set of design information is often both redundant and incomplete for analysis purposes. Figure 2 illustrates this scenario of multiple design and analysis applications.



**Figure 2:** Multiple Design and Analysis Applications Scenario

A step forward towards facilitating design-analysis integration is the utilization of neutral exchange formats such as STEP. In this approach, as shown in Figure 3, the data of each design application is translated to a neutral (standard) format. Analysis applications read the data from these standard formats without regard of the application that generated this data, eliminating the need for point-to-point translators and to update the analysis applications each time a new release of a design system is released.



**Figure 3:** Design-Analysis Integration using Neutral Data Exchange Standards

## 2 OVERVIEW OF STEP AND AP210

Perhaps the most important current effort to develop neutral representations of product data is the Standard for the Exchange of Product Model Data (STEP - ISO 10303) (Hardwick 1994; ISO 10303-1 1994; Laurance 1994; Owen 1993; Peak and Tamburini 1997). Since its Initial Release in March of 1994, new STEP parts and numerous application protocols<sup>1</sup> are being developed,

<sup>1</sup> Application Protocols (APs) are information models that specify the structure of the data for the exchange of information between applications of a specific domain. Examples of application protocols are AP210 (“Printed Circuit Assembly Product Design Data”) (ISO DIS 10303-210 1993), which describes the structure of the data needed to provide a manufacturable description of a PCA, and AP203

and different organizations and consortia are implementing prototype applications of the standard. STEP's ultimate intent is to become a single and better standard that supports the information needs of all aspects of product life cycle.

STEP and its textual conceptual schema language EXPRESS (ISO 10303-11 1994; Schenck and Wilson 1994) were selected for this research because they provide the neutral mechanisms for describing and exchanging product information and because there are several development tools commercially available (Spooner 1993; Spooner and Hardwick 1993; Wilson 1995) to aide in the development of STEP applications. Another reason for selecting STEP is its growing international acceptance by industry, government and academia.

STEP Application Protocol 210 ("Printed Circuit Assembly Product Design Data") (ISO DIS 10303-210 1993) defines the information shared between design and manufacturing process planning engineers for transforming a PCA detailed logical design into a manufacturable description. The detailed logical design identifies the PCA components and the connections between those components. The manufacturable description describes the physical realization of that logical design. This AP includes information about products that is sufficient to allow their use as components of a PCA, as well as information needed to support configuration management of the PCA product data. Figure 4 shows the main groups of information contained in AP210.

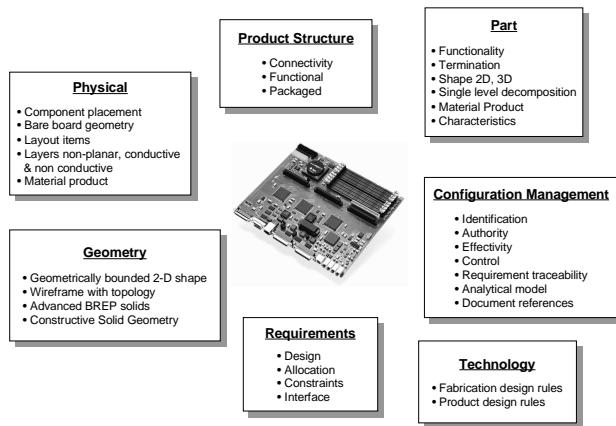


Figure 4: AP210 Information

### 3 TIGER PROJECT SCENARIO

TIGER was a DARPA-sponsored program, in which the authors participated, that began in October 1995 and was completed in February of 1997 (Peak et al. 1997; SCRA 1997). The goal of TIGER was to demonstrate a collaborative design and manufacturing scenario in which a small manufacturing enterprise (SME)

("Configuration Controlled Design") (ISO 10303-203 1994) which specifies the structures for the exchange between application systems of configuration controlled three-dimensional product definition data of mechanical parts and assemblies. It is expected that several hundred AP's will be developed to support the many industrial applications that STEP is expected to serve.

exchanged design information with the prime contractor early in the design process, thus reducing the iterations necessary to produce a successful design. For this purpose, a suite of design, manufacturing, and communications tools integrated across the Internet was made available to the product development team. The domain demonstrated was the design, fabrication, and assembly of printed wiring boards (PWBs) and printed wiring assemblies (PWAs).

In the TIGER scenario, a PWA designer generates PWA/B design information that is sent to a PWB manufacturer in STEP AP210 format. When the PWB manufacturer receives this file, he uploads it to an Internet-based engineering service bureau over the Internet (Scholand et al. 1997) that provides a variety of design and analysis services including design-for-manufacturability (DFM) and thermomechanical analysis. These services are integrated in an analysis environment developed for TIGER called *DaiTools-PWA/B* (Peak et al. 1997). Once the AP210 file is uploaded to the engineering service bureau, the PWB manufacturer invokes - from *DaiTools'* interface - a tool called PWB Layout Design Tool (shown in Figure 5). He uses this tool to specify the detailed layout of the PWB by selecting specific laminates, prepregs, and copper foils that physically realize the requirements specified by the PWA designer in the AP210 model. As these layout details affect PWB thermomechanical behavior, the PWB manufacturer runs some analyses to check the impact of his decisions (Figure 6). He invokes the Warpage Analysis Application to assess the warpage undergone by the board due to changes in temperature that occur during manufacturing. Two levels of warpage analysis detail can be requested: a quick formula-based warpage analysis and an FEA-based plane strain warpage analysis. The PWB manufacturer performs this design-analysis iteration until he is satisfied with the layout. Other analyses modules offered by *DaiTools* are a PWA deformation analysis (to assess the warpage of the board with the components on it), a solder-joint deformation and fatigue analysis (to assess joint deformation and fatigue life due to temperature changes on a component basis), and a plated-through hole deformation module (to assess deformation inside plated-through holes due to changes in temperature).

The Analyzable Product Model (APM, described later in Section 6) provides the integration of information needed to drive this design-analysis process. When the AP210 file is uploaded to the

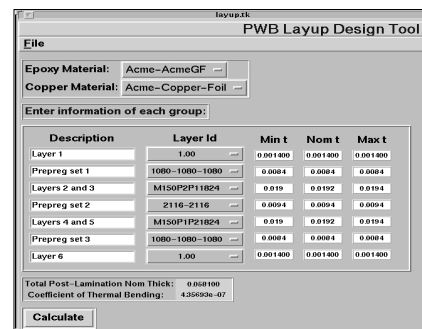


Figure 5: TIGER PWB Layout Design Tool

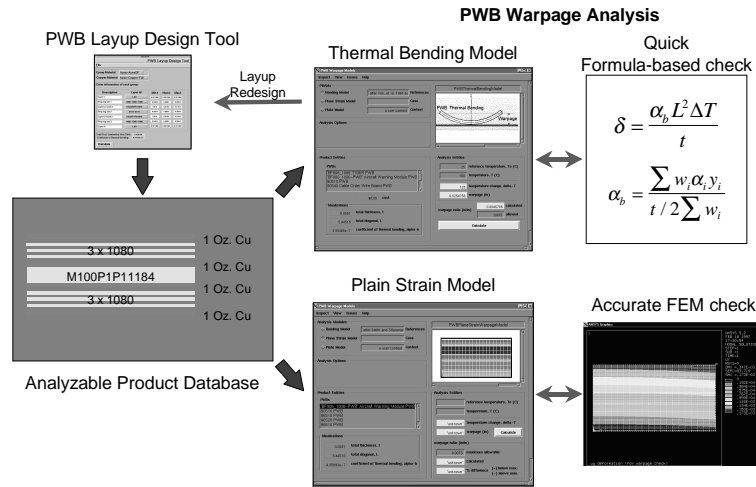


Figure 6: PWB Layup Design and Analysis Cycle

engineering service bureau, *DaiTools* reads it in and combines it with other information to form the Analyzable Product Database (APD, Figure 7). The APD becomes the only source of information required to support the analyses offered in *DaiTools*.

Figure 8 shows an instance of the FEA-based warpage analysis in which we can see how idealizations supported by the APM are used and displayed in the lower left region of the screen. The user enters the temperatures of interest (e.g., for the lamination temperatures as here) and clicks the <Calculate> button to see the deformations on a local PWB section. The program automatically extracts idealized attributes from the APM and creates a finite element job file and submits it to the FEA tool (Ansys in this case) for automated execution. The user receives results back in PWB-specific terms that are then checked against allowable values.

#### 4 TIGER ANALYSIS MODULES INFORMATION REQUIREMENTS

As mentioned in the previous section, four different analysis modules were supported in the TIGER scenario, namely, PWB Warpage, Solder Joint Deformation, PWA Warpage and PTH Deformation. The analysis modules themselves and the theory behind them will not be discussed here (for this, we refer the reader to (Peak et al. 1996)). For the purposes of this paper, the analysis modules may be viewed as “black boxes”, which consume information and employ some solution method to obtain analysis results. Table 1 shows the product information required by each of these analysis modules. As we can see in the PWB Warpage Module, this information depends on the solution method employed (e.g., formula-based vs. finite-element). Two important points about this table should be emphasized: first, some product information is used by more than one analysis module. For example, the detailed PWB layout is used by all analysis modules. Second, the information requirements of a given analysis module may span more than one repository. For example, the solder joint deformation analysis requires data about

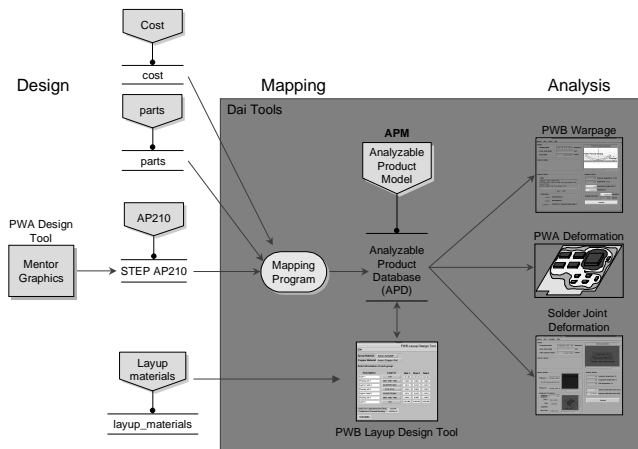


Figure 7: TIGER Data Flow

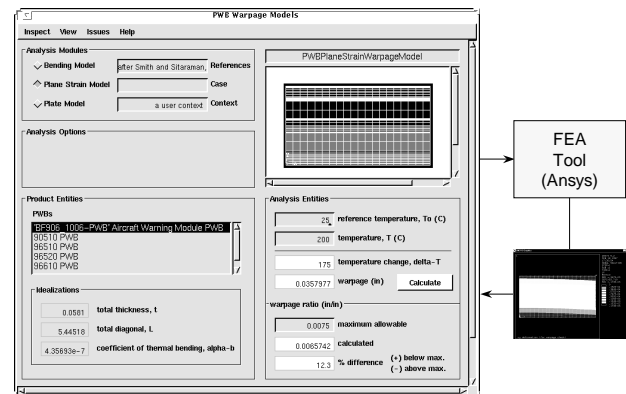


Figure 8: FEA-Based PWB Warpage Analysis Application

the detailed layout of the PWB (which is stored in one file) and the package geometry of the components (which is stored in a different file). The next sections will explain how these two issues are addressed by the technique presented in this paper.

## 5 TIGER DESIGN DATA SOURCES

In general, no single product model can satisfy the data requirements of all the analyses being supported. As a consequence, analysis applications normally feed from a variety of data sources which, in turn, conform to different data models. These data models may be standard data models (such as STEP Application Protocols) or arbitrary. This mix of standard and non-standard representations of data is due to the fact that there will be invariably some analysis that requires information not supported by any *existing* standard representation. Obviously, it would be impractical to develop a new standard representation for each new analysis situation that arises, as the number of these representations would grow exponentially. Therefore, non-standard repositories of data are frequently needed to complement the information that is missing.

As shown in Figure 7, the product data that drives the TIGER analyses is contained in four different STEP files (of which only one conforms to a standard data model, AP210). The first file is an AP210 file (“STEP AP210”, in the figure), obtained by translating data created with Mentor Graphics into STEP. This file is the main source of product information, containing geometric and assembly information about the board and the electrical compo-

nents on it, as well as the electrical interconnections among them. However, this file does not provide information as to the func-

tional information of the component (i.e., whether it is a resistor, a capacitor, etc.). This information is contained in a second file (called “parts”) which is a simple database of parts with their part numbers and their function. The third file (“cost”) contains cost information about different parts in the PWA. Finally, the fourth file (“layout\_materials”) contains information about commercial layout materials used to make up the board.

## 6 TIGER ANALYZABLE PRODUCT MODEL (APM)

### 6.1 APM Approach Overview

The Analyzable Product Model (APM) is a formal engineering representation – originally introduced in (Tamburini et al. 1996) - specifically tailored for analysis whose purpose is to facilitate design-analysis integration. This representation is an object-oriented data model for a single, integrated repository which sits between the design and analysis representations (as shown in Figure 9) providing a more analysis-oriented view of the product. From this APM, analysis applications can extract the information they need, including product idealizations. Besides providing a single source of analysis information, the APM bridges the semantic and syntactic gap between design and analysis and enables reusability by supporting data entities and idealizations that are shared among multiple analyses.

**Table 1:** Information Required by TIGER Analysis Modules

	PWB Warpage Module (Formula-based)	PWB Warpage Module (FEA-based)	Solder Joint Deformation Module (FEA-based)	PWA Warpage Module (FEA-based)	PTH Deformation Module (FEA-based) <sup>2</sup>
Detailed PWB layout (materials, nominal thicknesses)	✓	✓	✓	✓	✓
Layer functions (signal, ground, etc.)	✓				✓
PWB outline	✓	✓		✓	
Package geometry			✓	✓	
Package materials			✓	✓	
Lead/Termination shape			✓	✓	
Lead/Termination material			✓	✓	
Component location orientation, board side				✓	
Solder joint shape			✓		
Solder joint material			✓		
PTH geometry					✓
PTH plating material					✓

<sup>2</sup> The PBAM for this analysis module was fully designed but not implemented in TIGER.

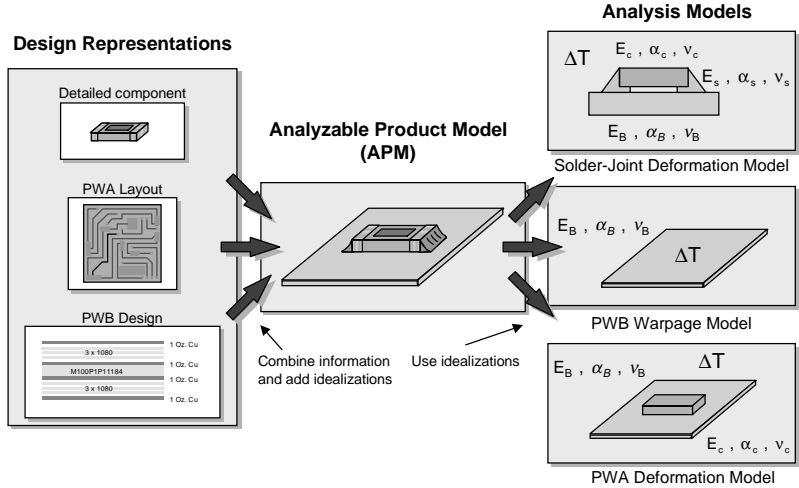


Figure 9: The Analyzable Product Model

Figure 10 is a diagram of the APM population technique reproduced from (Tamburini et al. 1996). The purpose of this technique is to populate the APM with design data spanning several design sources, in order to support the information requirements of the analysis modules downstream. As represented in Figure 10, the technique consists of four steps:

- 1. Design:** design applications ( $D_1$  to  $D_N$ ) generate data about various aspects of the product and store it in files with custom data structures and formats (shown in the figure as Design Files  $D_1$  to  $D_N$ , conforming to Custom Schemas  $D_1$  to  $D_N$ ).
- 2. Translation:** each custom design file is translated into a neutral format (STEP, shown as STEP Files  $D_1$  to  $D_N$ ).

- 3. Mapping:** data spanning different STEP files is integrated into a single repository (the Analyzable Product Database, or APD, which conforms to the APM).
- 4. Analysis:** finally, analysis applications ( $A_1$  to  $A_M$ ) retrieve the data they need from the APD.

The APM approach is integrated with the MRA (Multi-Representation Architecture) developed at Georgia Tech by (Peak et al. 1995). The APM complements the MRA by providing the product information required by PBAMs (Product Model-Based Analysis Models). Together, the APM and the MRA provide a highly modular and flexible design-analysis architecture.

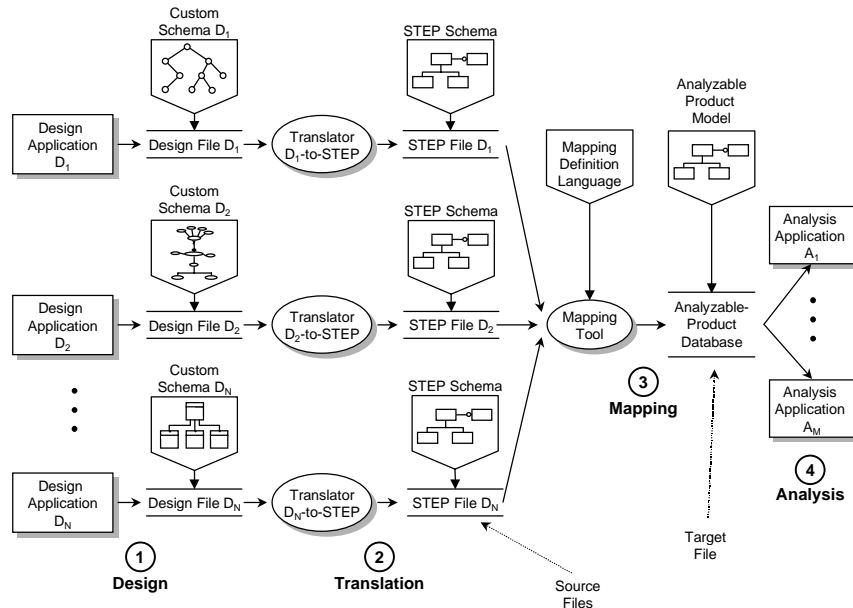


Figure 10: APM Population Technique (Tamburini et al. 1996)

## 6.2 TIGER APM Overview

As shown in Figure 7, the APM population technique described above was applied to the TIGER project. For this, a new APM - the “TIGER APM” - was developed for the project. The development of the TIGER APM was driven by the information requirements of the analysis modules described in Section 4. Figure 11 shows a high-level EXPRESS schema of the TIGER APM, which provides a single source for the following information about the product:

- **PWB information:** the outline of the PWB, its thickness and detailed layup (including thicknesses and materials of each layer that makes up the board).
- **Electrical components:** their location on the board, packaging (geometry and material), leads and solder joints (geometry and material).
- **Plated-through holes and vias:** geometry, location, and plating materials.
- **PWB layup materials:** nominal thicknesses and material properties (such as coefficients of thermal expansion, Young’s Modulus, etc.) for commercial copper foils, copper-cladded laminates and prepreg sheets that make up the PWB.

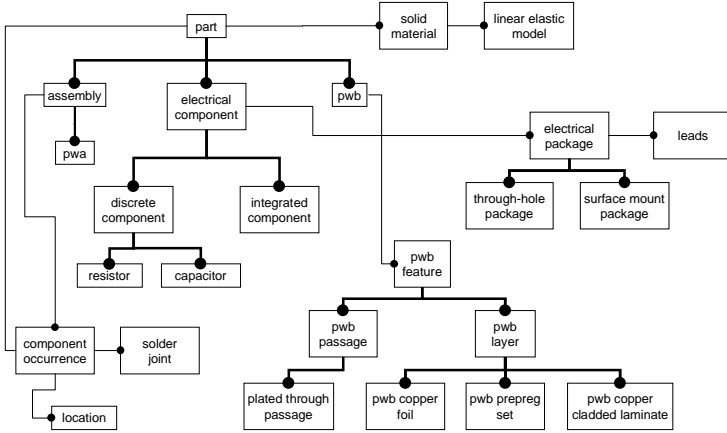


Figure 11: High-Level EXPRESS Schema of TIGER APM

## 6.3 AP210 - TIGER APM Mappings

Once the information requirements of the analysis modules are reflected in the APM, the next step is to design and implement the mappings that populate this APM with data coming from the four source files listed in Section 5. To illustrate this mapping process we will describe how the outline of the board is mapped from the source representations to the APM (for this particular example, only one of the four source representations – AP210 – is needed). The outline information is then used to calculate the total diagonal of the board; a product idealization required by the PWB Warpage Analysis Module.

Figure 12 is a partial EXPRESS schema showing how the outline information is represented in AP210. As the reader may realize (even with this relatively simple example), locating the required

information is not a trivial task and requires a considerable amount of knowledge about the source schemas.

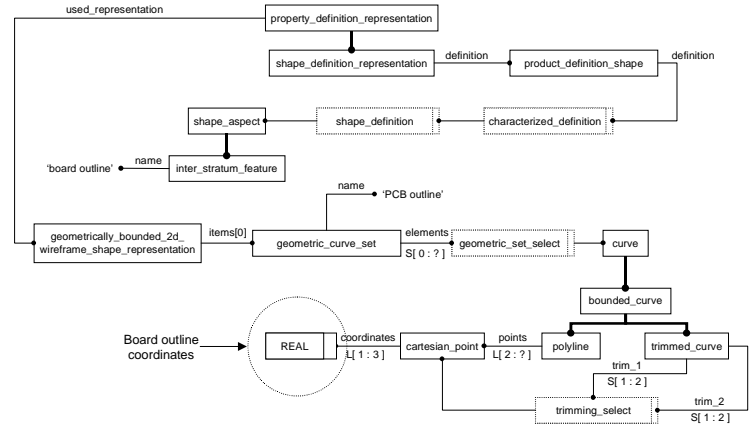


Figure 12: Board Outline Information in AP210

After identifying where the information is in the source schema (in this example, AP210), the next step is to define the mappings that must take place between this schema and the target schema (the APM, see Figure 13). Basically, defining the mappings means describing how data that conforms to one or more source schemas is mapped into a target (or view) schema. In general, the definition of the mappings requires the following information (Spooner et al. 1995; Spooner et al. 1996):

- **The base schema(s):** for the original product model(s) from which the view will be derived. For the board outline example, AP210 is the only source schema. In general, there could be more than one base schema.
- **The target schema:** which defines the product model for the materialized view - i.e., the entity types that will be in the view and the attributes for each of these entity types. In our example, the APM is the target schema.
- **Source entities:** the entities in the source schema from which the entities in the target schema will be created. In our example, the source entities are shape\_definition\_representations.
- **Target entities:** the entities in the target schema that will be created from the source entities. The target entities are xy\_coordinates, each representing one point of the outline of the board. These xy\_coordinates will be grouped in a list called listOfXYCoordinates.
- **Mapping conditions:** the conditions that must be true for a new view instance to be created. In the example, there is only one mapping condition, which states that only shape\_definition\_representations whose “definition” attribute is a product\_definition\_shape, whose “definition” attribute is an inter\_stratum\_feature, whose “name” attribute is a

string equal to 'board outline', will be taken into account for the mapping<sup>3</sup>.

- **Mapping assignments:** how the values of each attribute of a new view instance are computed. In our example, the mapping assignments specify how instances of `xy_coordinate` are created from the end points of each line and trimmed curve that define the outline of the board in the AP210 file. The pseudocode of these mapping assignments is shown in Figure 14.

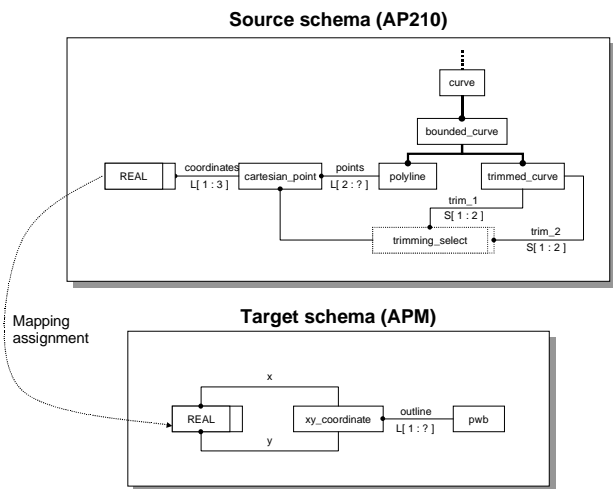


Figure 13: AP210 – APM Mappings

```

Create empty list of xy_coordinates (aListOfXYCoordinates) in the target database.
For each shape_definition_representation (aShapeDefinitionRepresentation) in the
AP210 file:
Do {
  If ( aShapeDefinitionRepresentation->definition->definition->name == 'board outline' )
  {
    aGeometricallyBounded2DWireframeRepresentation = aShapeDefinitionRepresentation->used_representation
    aGeometricCurveSet = aGeometricallyBounded2DWireframeRepresentation->item( 0 )
    aListOfCurves = aGeometricCurveSet->elements
    For each curve (aCurve) in aListOfCurves
    Do {
      If ( aCurve is of type 'polylines' )
      {
        aListOfCartesianPoints= aCurve->points
        For each cartesian_point (aCartesianPoint) in aListOfCartesianPoints
        Do {
          aListOfReals = aCartesianPoint->coordinates
          Create a new xy_coordinate (anXYCoordinate)
          anXYCoordinate->x = aListOfReals( 0 )
          anXYCoordinate->y = aListOfReals( 1 )
          Add anXYCoordinate to aListOfXYCoordinates
        }
      }
      If ( aCurve is of type 'trimmed_curve' )
      {
        aCartesianPoint = aCurve->trim_1
        aListOfReals = aCartesianPoint->coordinates
        Create a new xy_coordinate (anXYCoordinate)
        anXYCoordinate->x = aListOfReals( 0 )
        anXYCoordinate->y = aListOfReals( 1 )
        add anXYCoordinate to aListOfXYCoordinates
        aCartesianPoint = aCurve->trim_2
        aListOfReals = aCartesianPoint->coordinates
        Create a new xy_coordinate (anXYCoordinate)
        anXYCoordinate->x = aListOfReals( 0 )
      }
    }
  }
}

```

<sup>3</sup> Actually, there is only one of these `shape_definition_representation`s in the entire AP210 file, as there is only one board outline.

```

anXYCoordinate->y = aListOfReals( 1 )
add anXYCoordinate to aListOfXYCoordinates
}
}
}
}
}
Create a new pwb entity (thePwb)
Assign thePwb->outline = aListOfXYCoordinates
End of outline mapping

```

Figure 14: Pseudocode of Board Outline Mapping Assignments

The pseudocodes of the mapping assignments were implemented in ROSE C++ (Spooner 1993; Spooner and Hardwick 1993). An alternative way to describe these mappings is by utilizing a STEP mapping language such as EXPRESS-V (Spooner et al. 1995) or EXPRESS-X (Spooner et al. 1996)<sup>4</sup>. STEP mapping languages allow the definition of views of a product model defined in EXPRESS. They are usually extensions of EXPRESS that provide the capability of describing how data that conforms to one or more source schemas is mapped into a target schema. There are commercial mapping toolkits available that include a compiler to validate the syntax of the definition of the views and a run-time system for materializing them (create the instances in the target schema). No code needs to be written to implement the mappings when these mapping languages are used.

It is important to point out that, in this example, only one source schema (AP210) is participating in the mapping. However, as mentioned before, there may be more than one source schemas. When this is the case, the definition of the mappings requires additional conditions to specify when instances of one source schema are to be joined with instances of another source schema. A way to do this is by specifying key attributes, shared among source entities, that can be used to define the additional mapping conditions needed.

Other mappings were defined for TIGER to populate the APM with information about:

- The electrical components on the board, including their packaging, location and functional description (resistor, capacitor, etc.). These mappings used both the AP210 file and the part library file as source schemas. The part numbers of the components were used as the key attributes to correlate the data from both files.
- Preliminary information about the layout of the board such as total thickness, number of layers, function of each layer (signal, power, ground, etc.), and material specifications. AP210 was the source schema for this mapping.

<sup>4</sup> STEP mapping languages resulted from the need to be able to easily create views of STEP product models tailored to individual application systems. These views omit unnecessary details and are conceptually easier to understand than the STEP representations from which they are derived. The main STEP mapping languages available to date are EXPRESS-V (Spooner et al. 1995), EXPRESS-M (ISO TC184/SC4/WG5 N243 1995), EXPRESS-X (Spooner et al. 1996) (a combination of EXPRESS-V and EXPRESS-M currently under development, and that is likely to become the standard STEP mapping language in the near future) and BRITY (Sauter and Käfer 1996). At the time of writing, the authors were currently porting the mappings defined for TIGER to EXPRESS-X.



## 6.4 Adding Missing Data

As mentioned in the previous subsection, only *preliminary* information about the layout of the board was mapped from AP210 to the APM. This information is not sufficient to perform the analyses, as they require more detailed information about the layout of the board, such as the sequence of layers and their material properties and thicknesses. This information is not provided by the electrical engineer who designed the PWA, because he or she is not concerned with that level of detail. Instead, this information is supplied later in the design process by the PWB fabricator, who chooses the specific combination of layers used to build the PWB. For this purpose, the PWB fabricator uses the PWB Layout Design Tool (Figure 15). As shown in the figure, once the PWB fabricator specifies the detailed layout in the PWB Layout Design Tool, the APM is augmented with the more detailed layout information.

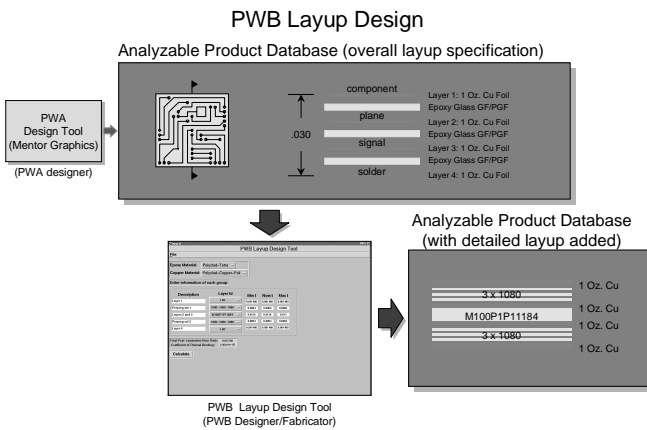


Figure 15: PWB Layout Design

## 6.5 Idealizations Supported

One of the key features of the APM is that it is not just a static repository of product data. The APM also contains the operations required to transform the product information (populated with the mappings) into idealized product attributes that can be shared among multiple analyses. These operations are also referred to as *product idealization transformations* (Peak et al. 1995; Tamburini et al. 1996). The values of these idealized attributes are calculated on demand as they are queried by the analysis modules. This strategy of storing the *operations* instead of the *values* of idealized attributes reduces the complexity of the mappings and the size of the APM significantly. In addition, potentially complex calculations are performed only when needed, allowing a more efficient utilization of computer resources.

The three main idealizations implemented for TIGER were the total diagonal of the board, total thickness of the board and coefficient of thermal bending ( $\alpha_B$ ) of the board. The first idealization (total diagonal of the board) is computed considering an imaginary bounding rectangle surrounding the outline of the board, and assuming the length of the diagonal of this rectangle as the

total diagonal of the board. The second idealization (total thickness of the board) requires the detailed layout of the board to calculate the post-lamination thickness, which takes into account the flow of epoxy material between the traces of the conductive layers when the board is heated and subjected to pressure during lamination. Finally, the coefficient of thermal bending is a lumped material property of the total layout and is calculated as a weighted sum of individual stratum properties.

To illustrate how idealizations were defined and implemented, let us return to the board outline example introduced in Section 6.3. Once the board outline is mapped into the APM, analysis applications could use it directly to determine the total width and length of the board, and with it calculate the total diagonal. However, since the total diagonal is an idealization of the product that is likely to be used by more than one analysis, it is better to have it readily available in the APM for the sharing. But instead of storing the *value* of the total diagonal in the APM (which would make the mappings more complex), we define it as an idealized attribute of the board and store the *operation* (or product idealization transformation) required to calculate its value from the outline.

In order to do this, we use EXPRESS to define the pwb entity and its regular and idealized attributes (as shown in Figure 16). Then we define WHERE rules for each product idealization transformation (PITs) that exists between these attributes. For example, WHERE rule wr1 defines a relationship between the width and the outline of the board. This relationship is further defined by FUNCTION pwb-pit-1 (Figure 17). Likewise, wr3 establishes a relationship between width, length and total diagonal, which is defined as pwb-pit-3 also in Figure 17. Notice that there is nothing implied in the WHERE rules as to which variables are inputs and which are outputs.

```

ENTITY pwb
SUBTYPE OF( multimaterial_part );

( * Product Attributes * )
outline : LIST (1:?) OF xy_coordinates;
layout : LIST (1:?) OF pwb_layer;

( * Idealized Attributes * )
width : positive_length_measure;
length : positive_length_measure;
total_diagonal : positive_length_measure;
total_thickness : positive_length_measure;
coefficient_of_thermal_bending : REAL;

WHERE
( * Product Idealization Transformations * )
wr1 : pwb-pit-1( width , outline );
wr2 : pwb-pit-2( length , outline );
wr3 : pwb-pit-3( width , length , total_diagonal );
wr4 : pwb-pit-4( coefficient_of_thermal_bending , layout );
wr5 : pwb-pit-5( total_thickness , layout );

END_ENTITY;

```

Figure 16: PWB EXPRESS definition in TIGER APM (partial)

```

FUNCTION pwb-pit-1( aWidth : positive_length_measure , anOutline : LIST (1:?) OF
xy_coordinates) : boolean;
(
    EXPRESS code that finds the maximum and minimum X and Y coordinates
(Xmax, Xmin, Ymax, Ymin, respectively) in anOutline and returns TRUE if
width = min( (Xmax-Xmin) , (Ymax - Ymin) ).
)
END_FUNCTION;

FUNCTION pwb-pit-3( W, L, D : positive_length_measure ) : boolean;
RETURN( D = SQRT(W*W+ L*L) );
END_FUNCTION;

(
Definitions for the other functions (pwb-pit-2, pwb-pit-4, and pwb-pit-5)
)

```

**Figure 17:** EXPRESS Definition of Product Idealization Transformations (PITs)

These EXPRESS definitions are then compiled to generate C++ code, which can be used by analysis applications to access the values of the attributes (regular and idealized) of the entities defined in the APM. As shown in Figure 18, the pwb entity defined in Figure 16 using EXPRESS becomes the C++ class pwb, and the attributes of this entity become member variables of the class. The EXPRESS to C++ compilation automatically generates member functions to access and update the values of the attributes.

In addition to the access and update member functions, each WHERE rule defines one access function for each possible combination of inputs and outputs. For example, from wr1 in Figure 16 we could generate a member function that outputs the width given the outline of the board. Likewise, from the same WHERE rule we could generate another member function to perform the opposite operation; calculate the outline of the board given the width (although, in this case, the problem must be somehow constrained in order to be able to reach a unique solution). However, the EXPRESS to C++ compiler does not automatically generate the appropriate access functions from the WHERE rules, and therefore they were implemented manually for TIGER<sup>5</sup>. Figure 19 shows two possible member functions created from wr1 and wr3, respectively. The first member function can be used to obtain the value of the width from the outline, and the second to calculate the total diagonal from the width and the length.

```

Class pwb {
Private:
    // Attributes
    ListOfxy_coordinates *outline;
    ListOfpwb_layer *layup;
    double width , length , total_diagonal ,
        total_thickness , coefficient_of_thermal_bending;
Public:
    // Attribute access functions
    ListOfxy_coordinates *outline( void )
    { return(outline) };
    ListOfpwb_layer *layup( void )
    { return(layup) };
    double width( void ) { return(width) }
    double length( void ) { return(length) }
    double total_diagonal( void )
    { return(total_diagonal) }
    double coefficient_of_thermal_bending ( void )
    { return(coefficient_of_thermal_bending) }
    double total_thickness( void )
    { return(total_thickness) }

    // Attribute update functions
    void outline(ListOfxy_coordinates *anOutline )
    { outline = anOutline };
    void layup( listOfpwb_layer *aListOfpwb_layer )
    { layup = aListOfpwb_layer };
    void width( double aWidth )
    { width = aWidth };
    void length( double aLength )
    { length = aLength};
    void total_diagonal( double aLength )
    { total_diagonal = aLength};
    void coefficient_of_thermal_bending( double aReal )
    { coefficient_of_thermal_bending = aReal };
    void total_thickness( double aThickness )
    { total_thickness = aThickness };
}; // End of definition of class pwb

```

**Figure 18:** Example APM Entity Implementation in C++

```

double pwb :: width( ListOfxy_coordinates *anOutline )
{
    double x , y , xMax , xMin , yMax , yMin;
    double deltaX , deltaY;
    xMax = anOutline->get(0)->x();
    xMin = xMax;
    yMax = anOutline->get(0)->y();
    yMin = yMax;
    for( int i = 0 ; i < anOutline->size() ; i++ )
    {
        x = anOutline->get(i)->x();
        if( x < xMin ) xMin = x;
        if( x > xMax ) xMax = x;

        y = anOutline->get(i)->y();
        if( y < yMin ) yMin = y;
        if( y > yMax ) yMax = y;
    }
    deltaX = xMax - xMin;
    deltaY = yMax - yMin;
    if( deltaX <= deltaY ) return( deltaX );
    else return( deltaY );
}

double pwb::total_diagonal( double width , double length
)
{
    return( sqrt( width*width + length*length ) );
}

```

**Figure 19:** Example APM PIT Implementation in C++

<sup>5</sup> The authors are currently investigating the use of constraint solvers to do this automatically.

## 6.6 APM Usage by Analysis Modules

Once the EXPRESS definitions of the TIGER APM entities and their attributes are translated into a target programming language such as C++, they can be used inside the codes of the analysis applications. To illustrate how this is done, let us refer to the code excerpt of Figure 20. This piece of code uses three of the idealizations defined in the APM (length, coefficient of thermal expansion and total thickness of the board) and Equation 1 below to calculate the deflection of the board due to a change of temperature.

$$\delta = \frac{\alpha_B L^2 \Delta T}{t} \quad (\text{Equation 1})$$

Where:

$\delta$  = maximum board deflection

$\alpha_B$  = coefficient of thermal bending of the board

$L$  = undeformed length of the board

$\Delta T$  = temperature change

$t$  = undeformed thickness of the board

```
/* Request temperature change from user */
cout << "Enter temperature change-> " ;
cin >> deltaT;

/* Open APD and make it active */
apd = ROSE.useDesign( apdFileName );
ROSE.useDesign( aopd );

/* Find PWBs in APD */
ROSE.findObjects( &listOfPwbs );

/* There should be only one PWB, therefore the first
element should be the pwb we are looking for */
pwb *thePwb = listOfPwbs( 0 );

/* Get idealized values, query pwb object directly */
totalLength = thePwb->length();
specificCoefficientOfThermalBending = thePwb-
>coefficient_of_thermal_bending();
totalThickness = thePwb->total_thickness();

/* Calculate deflection */
deflection =
specificCoefficientOfThermalBend-
ing*(totalLength*totalLength)*deltaT/totalThickness
```

Figure 20: PWB Warpage Analysis Code in C++ (excerpt)

This example shows how the APM greatly simplifies analysis code development. The developer of this code did not have to write additional code to calculate the total thickness, length and coefficient of thermal expansion of the board. Instead, he just queries the pwb object directly (for example, in `thePwb->length()`), and the values are calculated on demand. These values can be potentially queried by other analysis modules as well, thus reducing code replication and facilitating code maintenance.

## 7 DISCUSSION

This section discusses some Design-Analysis Integration (DAI) lessons learned during the TIGER project. First, we will list some

thoughts regarding the use of AP210 to drive thermomechanical analysis of PWAs. Next we will discuss the use of STEP in general as a neutral mechanism to enable Design-Analysis Integration.

### 7.1 TIGER Experiences Using AP210 to Drive Analysis

- The scope of AP210 is to provide a manufacturable description of a PWA, and is therefore geared toward *manufacturing*. However, it turns out to be a good source for thermomechanical *analysis* as well.
- Some information required for analysis is not available in the AP210 file. There are two reasons for this: First, the information may be supported by AP210 but not by the design tool. For example, it is possible to describe detailed PTH geometry, plating thickness and lead knee radius in AP210, but Mentor Graphics provides no means to populate this data. The second reason is that the information is not supported by AP210 at all as, for example, detailed parametric component packages.
- Some information (such as the detailed layout of the board) is not created by the PWA designer, because he is not concerned with that level of detail. Later in the design process, board fabricators specify the detailed layout based on their manufacturing and materials expertise.
- Even using a standard representation such as AP210 to exchange data between design and analysis, there is a potential for incompatibility due to certain tool idiosyncrasies. For example, Mentor Graphics Boardstation does not distinguish between the part number of the PWA and its PWB, even though the analysis tools using this data expect them to be different. This causes a conflict that had to be solved during mapping.
- When more than one source of design information is used to drive analysis, there must be some way to coordinate the integration of data spanning multiple repositories. For TIGER, the information in AP210 had to be integrated with information in external libraries about components, packages, layer materials (prepregs, etc.), and general materials. Unique identifiers (such as part numbers) were used to achieve this correlation. This requires some a priori standardization between the creators of these individual databases, since there must be some way to correlate this information.

### 7.2 Design-Analysis Integration Using STEP: Lessons Learned

- Domain-specific product models such as AP210 and AP207 are semantically richer than generic geometry-focused ones such as AP203, and therefore easier to use for design-analysis integration. For example, AP210 supports the concept of a lead, instead of just considering it as a surface. Therefore it is easier to design idealizations that deal with leads. However, due to the potentially infinite depth of idealization possibilities, no product model can ever be rich enough to support all possible idealizations.

- With the previous point in mind, we believe that the APM technique is needed be able to add missing and new information and to merge diverse data sources.
- Companies using the APM technique would have their own mix of “company PMs” and “company APMs”, in conjunction with standard product models, to support their specific design-analysis requirements. In the TIGER project, for example, PMs for electric parts and layer materials, the TIGER APM, and AP210 were used to support thermomechanical analysis of PWAs.
- The TIGER project provided a good testbed for the MRA concepts developed by (Peak et al. 1995). The MRA methodology enabled product data-driven analysis, analysis module plug-and-play, and extendibility.
- Design-analysis integration projects such as TIGER provide a good opportunity to debug and test existing APs because, in these projects, AP data is being used to drive engineering analysis as opposed to just being exchanged between similar applications. This new viewpoint provides additional insight about the data model and brings up issues that otherwise would have been missed.
- There will invariably some analysis that requires information not supported by any *existing* standard representation. Therefore, the overall design representation that drives analysis must be considered, in general, as an aggregation of standard and non-standard representations.

Based on TIGER experiences, the APM technique is presented as a general methodology that supports the modular extension of analysis capabilities while taking advantage of standards like STEP.

## 8 ACKNOWLEDGMENTS

TIGER was a DARPA-sponsored project under BAA 95-23. This work received helpful cooperation from the TIGER team members as acknowledged in (Peak et al. 1997). The authors would like to thank Lynwood Hines, of SCRA, for helping us with his invaluable expertise with AP210.

## 9 REFERENCES

- Armstrong, C. G. (1994). “Modelling Requirements for Finite-Element Analysis.” *Computer-Aided Design*, 26(7), 573-578.
- Brooke, D. M., Pennington, A., and Bloor, S. M. (1995). “An Ontology for Engineering Analyses.” *Engineering with Computers*(11), 36-45.
- Hardwick, M. (1994). “Towards Integrated Product Databases Using Views”, Report No. 94003, Design and Manufacturing Institute, Rensselaer Polytechnic Institute (Report No. 94003), Troy, NY.
- ISO 10303-1. (1994). “Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 1: Overview and Fundamental Principles”, International Organization for Standardization, Geneva, Switzerland.
- ISO 10303-11. (1994). “Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 11: Description Methods: The EXPRESS Language Reference Manual”, International Organization for Standardization.
- ISO DIS 10303-210. (1993). “Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 210, Application Protocol: Printed Circuit Assembly Product Design Data”, International Organization for Standardization, Geneva, Switzerland.
- Kemper, A. H., and Moerkotte, G. (1994). *Object-oriented Database Management : Applications in Engineering and Computer Science*, Prentice Hall, Englewood Cliffs, NJ.
- Laurance, N. (1994). “A High-level View of STEP.” *Manufacturing Review*, 7(1), 39-46.
- Owen, J. (1993). *STEP, An Introduction*, Information Geometers Ltd, Winchester, UK.
- Peak, R., and Tamburini, D. (1997). "A STEP Sampler", Internet WWW page, at URL: <http://eislab.gatech.edu/pubs/web/step-sampler/> (version current at August 12, 1997).
- Peak, R. S., Fulton, R. E., Nishigaki, I., and Okamoto, N. (1995). “Integrating Engineering Design and Analysis Using a Multi-Representation Approach.” *Engineering with Computers*, to appear.
- Peak, R. S., Fulton, R. E., and Sitaraman, S. K. (1997). “Thermomechanical CAD/CAE Integration in the TIGER PWA Toolset.” EEP-Vol. 19-1, E. Suhir, *InterPACK '97*, Kohala Coast, Hawaii, 957-962.
- Peak, R. S., Scholand, A., and Fulton, R. E. (1996). “On the Routinization of Analysis for Physical Design.” Application of CAE/CAD to Electronic Systems, EEP- Vol. 18, D. Ago-nafer, R. E. Fulton, G. J. Kowalski, and P. Ciekurs, *1996 ASME International Mechanical Engineering Congress and Exposition*, Atlanta, 33-46.
- Schenck, D., and Wilson, P. (1994). *Information Modeling the EXPRESS Way*, Oxford University Press, New York, NY.
- Shephard, M. S., Korngold, E. V., and Wentorf, R. (1990). “Design Systems Supporting Engineering Idealizations.” *Geometric Modeling for Product Engineering*, M. J. Wozny, J. U. Turner , and K. Preiss, eds., Elsevier Science Publishers, North-Holland, 279-299.
- Spooner, D. (1993). “Creating a Product Database with ROSE”, Report No. 93054, Design and Manufacturing Institute, Rensselaer Polytechnic Institute (Report No. 93054), Troy, NY.
- Spooner, D., and Hardwick, M. (1993). “Using Persistent Object Technology to Support Concurrent Engineering Systems.” *Concurrent Engineering: Methodology and Applications*, P. Gu and A. Kusiak, eds., Elsevier Science, New York, 205-234.
- Spooner, D., Hardwick, M., and Wen, J. (1995). "The EXPRESS-V Language Manual", Internet WWW page, at URL: <http://www.rdrp.rpi.edu/express-v/homepage.html> (version current at June 29, 1997).

Spooner, D., Hardwick, M., and Wen, J. (1996). "The EXPRESS-X Language Manual", Internet WWW page, at URL: <http://www.rdrp.rpi.edu/express-x/homepage.html> (version current at June 29, 1997).

Tamburini, D. R., Peak, R. S., and Fulton, R. E. (1996). "Populating Product Data for Engineering Analysis With Applications to Printed Wiring Assemblies." Application of CAE/CAD to Electronic Systems, EEP-Vol.18, D. Agonafer, R. E. Fulton, G. J. Kowalski, and P. Ciekurs, *1996 ASME International Mechanical Engineering Congress and Exposition*, Atlanta, 33-46.

Wilson, P. (1995). "EXPRESS Tools and Services", Catholic University of America and NIST, Gaithersburg, MD (Available at: <http://www.nist.gov/sc4/tools/express/>).