THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS
345 E. 47th St., New York, N. Y. 10017

93-WA/EEP-23

AUTOMATING ROUTINE ANALYSIS IN ELECTRONIC PACKAGING USING PRODUCT MODEL-BASED ANALYTICAL MODELS (PBAMS), PART I: PBAM OVERVIEW

Russell S. Peak and Robert E. Fulton
Advanced Electronic Packaging Laboratory
Manufacturing Research Center
Georgia Institute of Technology
Atlanta, Georgia USA

ABSTRACT

Previous work introduced a new representation of engineering analysis models, termed *product model-based analytical models* (PBAMs) [Peak and Fulton, 1992b]. Since analysis information is linked with detailed design information, PBAMs enable rapid, flexible analysis in support of product design. In this approach, a catalog of ready-to-use PBAMs can be created that automate established, "routine" analysis models.

This document, Part I of two companion papers, discusses the use of PBAMs in routine analysis and overviews the PBAM representation. Part II describes PBAMs of representative solder joint fatigue models that illustrate and evaluate the PBAM representation. Combined, these papers emphasize how PBAMs can automate the interaction of heterogeneous analysis models.

A new notation, *constraint schematics,* is included in this paper that graphically represents analysis models. Since PBAMs are founded upon constraint graph theory and object-oriented information modeling concepts, a high degree of flexibility and modularity is achieved.

1 INTRODUCTION

I have more than enough to do than to extract my own FEA data. I am talking to manufacturing, tooling, vendors, fatigue and fractures, the stress group - I simply don't have time.
Design Engineer, Airplane Structures
[Liker, et al., 1992]

Unfortunately, the above quote often typifies the priority given to analysis in the product design process today. In their survey of industrial CAD/CAE usage, Liker and associates identify five "unfulfilled promises of CAD," one of which is "there will be an iterative and seamless link between CAD and CAE analysis." They found that even the largest users of CAD/CAE often had inadequately staffed analysis groups compared to the analysis needs of the design groups.

An earlier paper [Peak and Fulton, 1992b] introduced a new representation of analysis models, termed *product model-based analytical models* (PBAMs), that automates some analysis tasks to support such needs during product design. That paper reviewed related work and described concepts for automatically creating analysis models directly from detailed design data. Results emphasized how engineering analysis is an information-driven process.

More recent work [Peak, 1993] defines the formal structure and operation of the PBAM representation. This paper (Part I) and its companion (Part II) are largely extracted from that work. The purpose of these two papers is to show how the PBAM representation enables automated interaction of diverse analysis models and the product model.

This Part discusses the concept of "routine analysis models" and how PBAMs can be used during the design of printed wiring assemblies (PWAs). It gives an introduction to the general PBAM representation itself including constraint schematic notation. Part II presents solder joint fatigue case studies which evaluate and further illustrate the PBAM representation.

2 USING PBAMS IN PWA DESIGN

This section gives a "black box" view of how one uses PBAMs without going into the internal details. The next section is an introduction to this latter point.

2.1 Terminology

A **PBAM** is a representation of engineering analysis models that includes linkages to product model design information. The term **representation** here means a computable approximation of "reality" for an intended purpose. The "reality" a PBAM approximates is an analysis model, in contrast to geometric representations that approximate physical objects. Thus, a PBAM is a *model of analysis models*.

The PBAM representation has a defined structure and defined operations analogous to mathematical entities such as matrices and graphs. *Creating* a PBAM to represent a specific analysis model involves filling in this general structure with analysis model-specific information. One *uses* the resulting PBAM by performing the operations defined for the general representation.

PBAMs are intended to represent **routine analysis models**, a term derived from the simplest of three classes of design:

Presented at the ASME Winter Annual Meeting
New Orleans, Louisiana - November 28 - December 3, 1993

routine, adaptive, and original design [Opitz et al., in Pahl and Beitz, 1988]. A routine analysis model is an established analysis model that has been developed for application to a specific type of product.

Routine analysis emphasizes repeatedly *using* proven analysis models on new product instances and on design iterations for the same instance. The term routine is not meant to imply that the analysis models involved are simplistic. After an analysis model has been developed, and its utilization in design is understood, even the most sophisticated model can be considered a routine model. Understanding model limitations and knowing how to apply the results to design are the primary skills required to use a routine analysis model.

Routine analysis can be contrasted with two other categories of analysis: **adaptive analysis** and **original analysis** [Peak, 1993]. In these cases the emphasis is on model *development* rather than model *usage*. These categories are not dogmatic and some overlap may exist.

2.2 Routine Analysis in PWA Design

Figure 2-1 gives specific examples of product requirements that could be checked using routine analysis models when a PWA is in the board layout design stage.

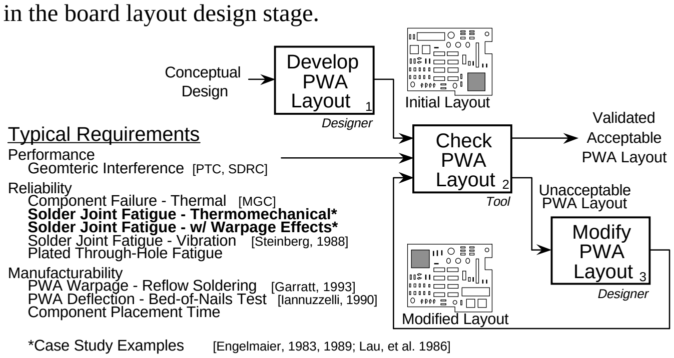


Figure 2-1 PWA Design Validation Process

Each product requirement may have its own analysis model or series of models to judge if the preliminary PWA design is acceptable. It is important to note that numerous analysis models exist that can be used to check such PWA requirements. For example, Mentor Graphic's Autotherm computes component temperatures to check component reliability. Lau [1991] contains numerous solder joint fatigue models (though not all are oriented towards design use). Steinberg [1988] devotes a whole book to electronic equipment vibration analysis. Garratt [1993] simulates thermomechanical deformations in a simplified bare printed wiring board (PWB) during the reflow soldering process. Furthermore, Iannuzzelli [1990] describes a suite of analysis models to simulate the behavior of a PWA during various manufacturing processes, including bed-of-nails testing.

The results obtained from such analysis models may indicate the need for a design change. For example, if the design is deemed unacceptable from a component reliability point of view, the component could be moved to a cooler area on the PWA, or the enclosure could be modified to provide more cooling to that area of the PWA. Resources permitting, this modified design should then be re-checked.

One can see how a variety of checks needs to be made during the design of a product. Typically for a given product type (e.g. PWAs) the same type of analyses need to be done several times for each new product instance at various stages during the design process. In such cases it would be helpful to have pre-defined catalogs of routine analysis models that can be used after being populated with the specific data of a new design. This section shows how PBAMs provide a way to create such catalogs.

With checking solder joint fatigue as an example context, Figure 2-2 illustrates how a user generally would utilize a PBAM selected from such a catalog of analysis models. Depending on the design stage and problem being addressed, models of varying complexity and computational cost could be needed and would be part of the catalog.

First, the user selects which analysis model (represented by a PBAM) to use. Second, the required product and analysis entities are connected to the selected PBAM at a high level. In Figure 2-2 the component of interest (R110), the load frequency, and the temperature extremes are such inputs. The PBAM automatically extracts detailed information from the connected entities to create the analysis model it represents. The creation, execution, and interaction of submodels within the analysis model (if applicable) are also handled automatically. Finally, the PBAM allows the user to obtain the result in a form that is meaningful to the problem at hand. A solder joint fatigue of 3536 cycles is the result shown in the figure.

Often design checks which involve analysis models are performed less frequently than would be desired due to the large amount of resources required to perform the analysis. The major consumer of human resources is often the creation of the model itself (even for routine analysis). The solution phase of analysis is typically well developed in terms of computer-aided automation. Other problem areas include the interaction of several possibly heterogeneous submodels within an analysis model, and the feed back of final results to drive design changes

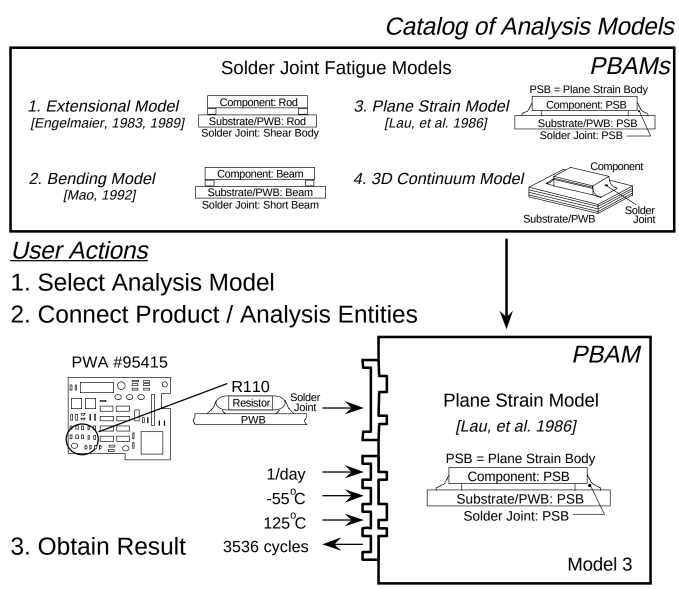


Figure 2-2 Routine Analysis Using PBAMs

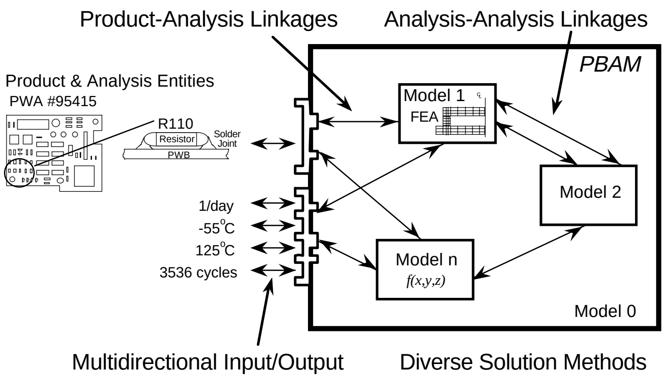


Figure 2-3 Challenges in Automated Analysis

(Figure 2-3). Furthermore, if analysis models are to be used both for design analysis and design synthesis, one must be able to run analysis models with different input/output directions as indicated by the two-way arrows (e.g., input the fatigue life to determine allowable temperature extremes). Therefore, the key areas of research that will most likely increase the automation of the analysis process as a whole are:

- Automated model creation
- Heterogeneous multi-model interaction
- Results feedback
- Multidirectional input/output

This research addresses these problems specifically for routine analysis problems.

3 INTRODUCTION TO THE PBAM REPRESENTATION

As noted above, the PBAM representation has both general structure and general operations which are formally defined in [Peak, 1993]. Six different views are also defined which help capture and communicate the structural and operational aspects of a PBAM. Some of these views are utilized later in this paper. This section introduces the internal workings of the PBAM representation, which is based on constraint graph theory and object-oriented concepts.

3.1 Constraint and Object Representations

The basic idea of **constraints** is that **variables**, a_i, and **relations**, r_j, among those variables can be declared explicitly by a user or application without explicitly specifying how any unknown variables are to be determined [Freeman-Benson, et al., 1990]. In a **constraint graph**, variables and relations are both vertices, and edges represent the participation of a variable in a relation. For example, consider the variables a_1, a_2, a_3, a_4 and the relations $a_1 + a_2 - a_3 = 0$ and $a_4 - (a_3 \cdot a_2) = 0$. The

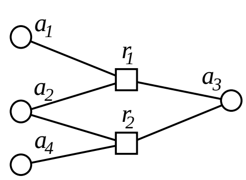


Figure 3-1 A Constraint Graph

constraint graph of this case is given in Figure 3-1 where the convention of designating relations by boxes and variables by open circles has been used .

A key advantage to viewing relations and variables as constraints is that constraints can be *multidirectional* (assuming such inversions are mathematically possible.) In the above example if a_1 and a_2 are given as inputs, a_3 will be determined, and then a_4. Likewise, a_2 and a_4 could be input to determine a_3 and then a_1.

A review of constraints in general and their applications to engineering problems in particular is given in [Peak, 1993]. None of the papers reviewed discuss the linkage between detailed product and analysis models in terms of constraints. Also, analysis relations represented as constraints are typically formula-based and are presented as "tangled knots" of constraints. No work was found that viewed the interactions between heterogeneous analysis models as constraints.

The advantages of object-oriented (OO) representations for engineering applications, including information hiding and encapsulation, are discussed in [Peak and Fulton, 1992b]. The remainder of this section shows how constraints and objects can be combined to represent engineering analysis models and address the above gaps.

3.2 Constraint Schematics of Analysis Models

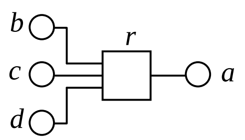
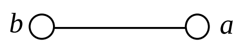
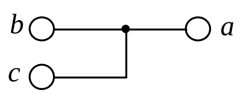
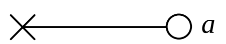
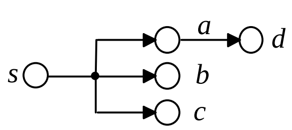
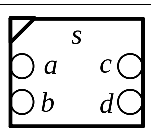
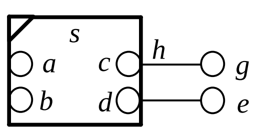
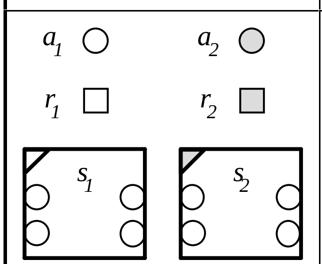
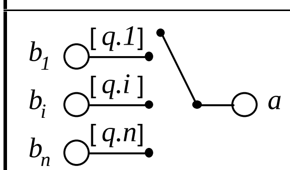
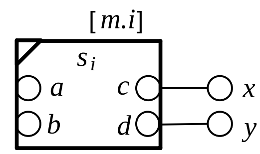
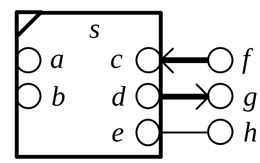
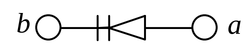
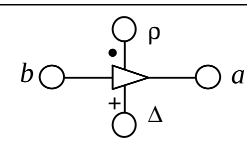
This section shows how constraints and objects can be used to represent analysis models. With this intent in mind, a new notation, termed **constraint schematics**, has been developed in this research to graphically illustrate analysis model constraint graphs. The basics of this notation are given in Table 3-1. Since the underlying structure from a constraint viewpoint is a graph much like that encountered in electrical schematics, analogous terminology and features have been used. For example, the subsystem symbol is based on an integrated circuit pin-out diagram.

Figure 3-2a is a constraint schematic of the constraint graph given in Figure 3-1. Note the inclusion of the part-of relationships (as indicated by the small filled triangles) in order to organize variables into natural hierarchies. The variable *s* is an object that encapsulates all the variables and relations shown. Hence, a constraint schematic capitalizes on the strengths of both objects and constraints.

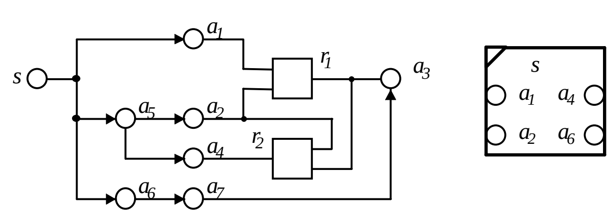
Figure 3-2b is one possible **subsystem** view of this object (an object can have many different such views). In a subsystem view, only variables of interest to a particular use of the object need be shown. Note that a_3 can still be accessed through a_6 via the part-of relationship. The full constraint schematic is present and active in a subsystem since this view is simply an abstraction that hides unnecessary detail.

A subsystem view is somewhat analogous to an integrated circuit pin-out diagram. Just as an integrated circuit can be a component in a larger electrical circuit, an object can be a subsystem in the constraint schematic of another object. Since constraint schematics can contain arbitrarily deep subsystem nestings, they provide a way to organize the previously mentioned "tangled knots" of constraints into meaningful bundles.

Table 3-1 Constraint Schematic Notation

Symbol	Meaning
	*a* is a **variable**.
	Variables *a*, *b*, *c*, and *d* are related by **relation** *r*. This relation can be written as *r(a,b,c,d)*.
	Variables *a* and *b* are equal, i.e. an **equality relation** exists between them.
	Variables *a*, *b*, and *c* are equal.
	Variable *a* is not valid in the current context.
	Variable *s* has **attributes** *a*, *b*, and *c* which are variables (i.e., they are *part-of s*). Variable *d* is an attribute of *a* and a **subattribute** or **subvariable** of *s*. I.e., variables *s*, *s.a*, *s.b*, *s.c*, and *s.a.d* are shown.
	Variable *s* is a **subsystem** which has attributes/sub-attributes *a* through *d*.
	Variable *s.c* is known as *h* in the **scope** outside of *s* (i.e., *h*=*s.c* and *g*=*h*). Variables *a*, *b*, and *d* have the same names in both scopes (i.e., *a*=*s.a*, *b*=*s.b*, *d*=*s.d*, and *d*=*e*)
	Unshaded variable a_1, relation r_1 and subsystem s_1 are inherited from the super class. Shaded variable a_2, relation r_2, and subsystem s_2 and related connections are new to this class.
	An **n-position pole** equates *a* and b_i when **position** *i* of **switch** *q* is selected for *i* = 1...*n* (in other words, when **option** *i* of **option category** *q* is chosen).
	**Subsystem Substitution** Switch *m* contains an *n*-position pole between each connected variable pair (e.g., $s_i.c$ = *x* when subsystem *i* is selected for *i* = 1...*n*).
	*a* and *b* are connected by a **jumper** (i.e., the *user* has made them equal).
	**Instance View** An *instance* of subsystem *s* has variable *f* input into variable *s.c*. *g* is read as an output from *s.d*. In contrast, variables *h* and *s.e* are always equal.
	An **absolute value relation** exists between *a* and *b*: $a = \| b \|, \quad b = \pm a$
	A **scale & offset relation** exists between *a* and *b*: $a = \rho \cdot b + \Delta,$ $b = (a - \Delta) / \rho$

Analytical primitives and systems like those described in [Peak and Fulton, 1992b] can be partially represented by constraint schematics, and analytical systems can be viewed as one type of subsystem. Analytical variables and analytical relations have a somewhat natural correspondence with different analytical primitives as discussed in that paper.



a. Constraint Schematic b. Subsystem

Figure 3-2 Views of an Object

Note that when an analytical system is used for *design verification (design analysis)* purposes versus *design synthesis* purposes, the roles played by at least one input and one output variable reverse. Therefore, the ability to go in multiple directions is critical if an analysis model representation is to be used for different design purposes.

Example

Figure 3-3 is an example analytical system which is used in the PBAM representing Engelmaier's extensional model (Part II). Hence, it contains the following relations that determine the thermal expansion mismatch and calculate the shear strain in body 3 (corresponding to Eqns. 2-4 and 2-5 in Part II).

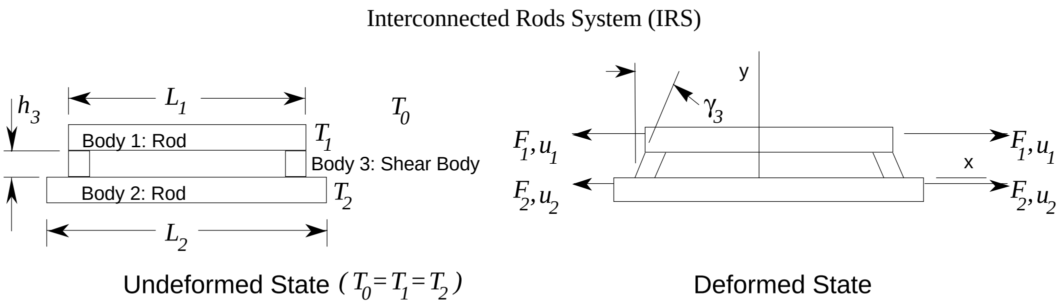
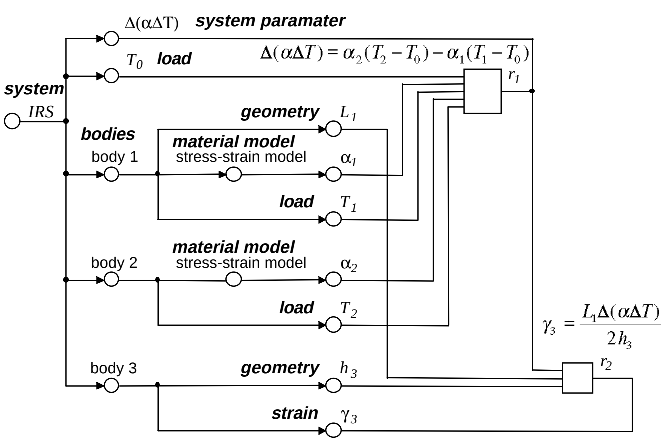


Figure 3-3 An Exemplar Analytical System

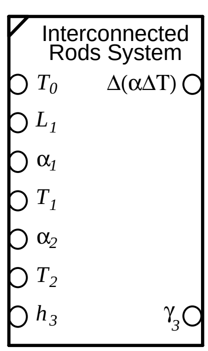
$$\gamma_3 = \frac{L_1 \Delta(\alpha \Delta T)}{2 h_3} \tag{3-1}$$

$$\Delta(\alpha \Delta T) = \alpha_2 (T_2 - T_0) - \alpha_1 (T_1 - T_0) \tag{3-2}$$

Note that here the subscripts refer to generic body numbers and not to the physical entities being modeled in a specific product. A partial constraint schematic of this system and one subsystem view are given in Figure 3-4 (for the case where $F_1 = F_2 = 0$). The specific variable names have been augmented with the category of analytical variable they belong to in ***bold***.



a. Constraint Schematic



b. Subsystem View

Figure 3-4 Views of an Analytical System

3.3 Simple PBAMs

Thus far only analytical variables and relations have been considered. It will now be shown how PBAMs are representations that relate analytical variables to *product variables* in a specific application via transformations (Figure 3-5). Remember that analytical primitives, systems, etc. are generic in the sense that they potentially can be used to model many different products. The basic idea of a PBAM is to utilize these generic resources but link them with product-specific data.

A PBAM has **product variables** (p_1 and p_2 in PBAM$_0$ in the figure) which define what product model data must be "plugged in." Internally the PBAM can decompose these variables into their subattributes (p_3, p_4, p_5 in PBAM$_0$) per the part-of relationship. Analytical variables (a_1 through a_8 in PBAM$_0$) may be decomposed into subattributes in the same manner as product variables.

Product-analysis transformations are linkages that relate product variables to analytical variables. An **idealization** is a process that simplifies the physical (product) situation into analysis attributes [Shephard, et al. 1992]. The term product-analysis transformation is used here to describe the *bi-directional* relation between product and analytical variables; an idealization is one direction of this relation.

Also, there may be **analysis-analysis transformations** within the PBAM scope (e.g. r_1 in PBAM$_0$). Both product variables and analytical variables can eventually be related to the analytical variables of the subsystem, s_1.

The **analysis context** shown hooked to PBAM$_0$ in Figure 3-5 is the user of the PBAM. This user could by either a person or a computer tool (e.g. an expert system checking the product design) or even another PBAM. This figure is a PBAM **detailed instance view** that shows how a PBAM is used by an analysis context. Therefore, bold lines are used to indicate connections between the two. Arrows on the bold lines indicate inputs and outputs for this particular usage of the PBAM (other notation is explained in Table 3-1).

To utilize a PBAM once it has been implemented, the **analysis context** initializes the PBAM, connects the necessary product and analysis entities, and specifies which variable is to be the output of the PBAM. If the analysis context is a designer, highly interactive use of the PBAM may be possible. For example, the designer can change which variables are inputs and which are outputs to answer "what if" type questions.

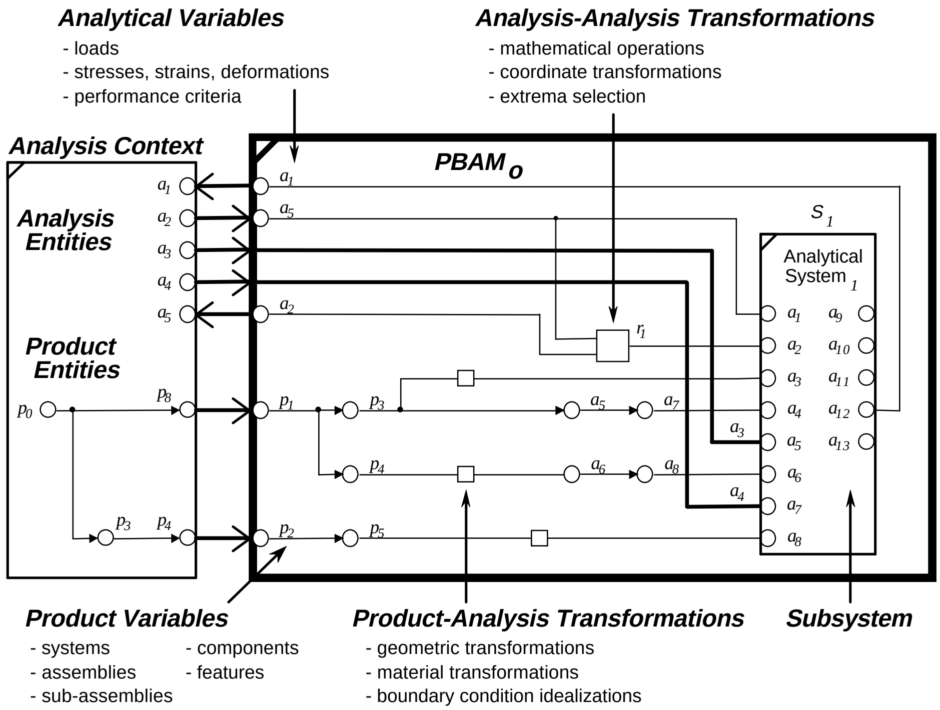


Figure 3-5 Structure of a Simple PBAM

Additionally, he or she can inject new values into variables and see the result by probing other variables.

Table 3-2 Characteristics of Simple and Complex PBAMs

	No. of Subsys.	Subsystem Types	Emphasis
Simple	1	Analytical system Matter model, etc.	Product-Analysis Transformations
Complex	1 or more	Same, plus other PBAMs	Subsystem Interactions

3.4 Complex PBAMs

Just as analytical systems can be used as subsystems in other analytical systems, PBAMs can be used as subsystems by other PBAMs. This nesting of PBAMs can be arbitrarily deep. Thus, it is helpful to distinguish between **simple PBAMs** and **complex PBAMs** by the characteristics given in Table 3-2.

In a complex PBAM (Figure 3-6) analytical systems may be

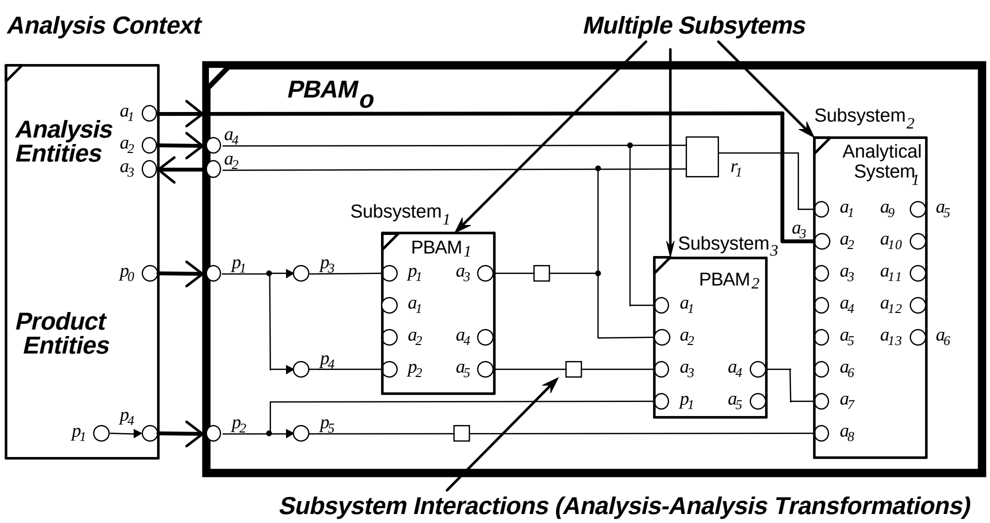


Figure 3-6 Structure of a Complex PBAM

used as subsystems and require analytical variable connections just as when used in a simple PBAM; in contrast, PBAMs used as subsystems also require connections to their *product variables*. Also note that subsystems can be connected to each other in a typically "my output = your input" fashion (where input/output is relative to which variable is viewed as the output of the outer PBAM.)

3.5 Summary

This section introduced the new constraint schematic notation which merges constraint and object concepts. The use of this notation in representing analytical primitives, analytical systems, and both simple and complex PBAMs was illustrated, along with the basic structure of a PBAM.

The following five PBAM views capture the *structure* of a given PBAM and are listed here for reference.

Table 3-3 PBAM Structural Views

1. Master View	4. Subsystems
2. Extended Constraint Graphs	5. I/O Tables
3. Constraint Schematic	

The master view fully defines a PBAM representation of an analysis model. This textual view is complete but can be difficult to comprehend as a whole. The other views aide comprehension but are incomplete. Therefore, all views together play complementary roles in the development, implementation, and use of a PBAM.

All views can be combined together to provide a human comprehensible (but redundant) documentation of a PBAM definition. This conglomeration is analogous to the "data sheet" description of electronic components in vendor data books. For increased modularity and decreased redundancy, these PBAM views can reference the following kinds of resources:

1. Parent PBAM "Data Sheet"
2. Analysis Building Block "Data Sheets"
3. Product Model

The basic operations of the PBAM representation include connecting the analysis context, assigning an I/O combination, and changing variable values. An instance view (e.g., Figure 3-6), is an *operational* view that shows how a PBAM is used to obtain a specific result (i.e., it shows the inputs, the outputs, and the options used).

Developing PBAMs to represent a given analysis model involves populating the structural views given above and is largely implementation independent. Once the views are completed, the PBAMs can be *implemented* in a computer system and subsequently *used*. Guidelines for both developing and implementing PBAMs are given in [Peak, 1993].

4 DISCUSSION

Analysis model descriptions may be difficult to piece together and typically will not contain all the information required to develop PBAMs. When it does exist, analysis model documentation typically suffers from a lack of product-analysis transformation intent. These situations make it difficult to reproduce an analysis model exactly, but an experienced analyst can probably fill in the gaps to a sufficient degree.

"Routine analysis models" may not be easy to identify for newer disciplines such as PWA thermomechanical analysis. In fact many analysis models that can be found in the literature may not have been originally intended to support product design directly.

Simply defining the term "routine analysis model" spurs the notion that such models *should* exist (and they often do in mature product domains). Furthermore, the need for routine analysis models can direct further research and development to fill in gaps that may exist. Once a routine analysis model has been identified, PBAMs can help maximize the use of that model during product design.

Other observations are given in Part II with specific reference to the solder joint fatigue case studies. Results show that PBAMs enable automated rapid analysis for mixed formula-based and finite element-based models.

5 SUMMARY

A new analysis model representation, termed *product model-based analytical models* (PBAMs) has been developed that enables rapid, flexible routine analysis concurrent with product design. This structured representation fully automates the creation, execution, interaction, and (to some degree) the results feedback of a variety of *routine analysis models*. PBAMs are a blending of objects and constraints, resulting in the following characteristics:

- Linkage between product model and analysis model
- Support for multiple models of varying complexity
- Options that allow seamless variations of analysis model characteristics
- Modularity and flexibility
- Uniform treatment and interaction of analysis models requiring different solution methods
- Multiple input/output alternatives

The latter two points have been particularly emphasized in this paper and its companion.

ACKNOWLEDGMENTS

This work (both Part I and II) was funded by the Georgia Tech Manufacturing Research Center and its industrial sponsors: DEC, Ford, IBM, Motorola, and the U. S. Army Missile Command.

REFERENCES

Engelmaier, W., 1989, "Thermal-Mechanical Effects," in *Electronics Material Handbook. Vol 1 -Packaging,* Minges, M. L., ed., ASM Intl., Materials Park, OH, pp 740-753.

Engelmaier, W., 1983, "Fatigue Life of Leadless Chip Carrier Solder Joints During Power Cycling," *IEEE Trans. on Components, Hybrids, and Manufacturing Technology*, Vol CHMT-6, No. 3, Sept. 1983, pp 232-237.

Freeman-Benson, B. N., Maloney, J., Borning, A., 1990 "An Incremental Constraint Solver," *Communications of the ACM,* Vol 3, No 1, pp 54-63.

Fulton, R. E., Ume, C., Gabertan, M. Y., Fu, C. Y., Mao, J., Martin, T. L., Peak, R. S., Tsang, F., Yeh, C. P., Oct. 1992, *An Integrated Approach to Printed Wiring Board Design: Thermal Mechanical Behavior and Engineering Information Integration, Final Report: June 1991 - September 1992*, Manufacturing Research Center, Georgia Tech, Atlanta GA.

Garratt, J. D., Sept. 1993, *Prediction of Thermally Induced Printed Wiring Board Warpage* Masters Thesis, Georgia Institute of Technology, Atlanta GA.

Iannuzzelli, R., 1990, "Validation of Module Assembly Physical Models," *Proc. IEEE 40th CHMT Conf., Vol 2*, Las Vegas NV, pp 1071-1076.

Lau, J. H., Rice, D. W., Avery, P. A., 1986, "Nonlinear Analysis of Surface Mount Solder Joint Fatigue," *Proc. IEEE CHMT Symposium*, pp 173-184.

Lau, J. H., ed., 1991, *Solder Joint Reliability Theory and Applications*, Van Nostrand Reinhold, New York.

Liker, J., Fleischer, M., Arnsdorf, D., 1992, "Fulfilling the Promises of CAD," *Sloan Management Review*, Spring 1992, pp 74-86.

Pahl, G., Beitz, W., 1988, *Engineering Design - A Systematic Approach*, The Design Council, London.

Mao, J. and Fulton, R. E., 1992, "Thermal Fatigue Reliability of the Solder Joints of Leadless Chip Components," in Fulton, Ume, et al., 1992.

MGC, 1991, *Autotherm Reference Manual*, Mentor Graphics Corporation, Wilsonville OR.

Peak, R. S., Fulton, R. E., 1992a, "Selection of Printed Wiring Assembly Components Using a Multidisciplinary Integrated Information Framework," *Advances in Electronic Packaging 1992*, Proc. 1992 Joint ASME/JSME Conf. on Electronic Packaging, San Jose CA, pp 57-65.

Peak, R. S. and Fulton, R. E., 1992b, "Integrating Analysis and Design Information in Electronic Packaging Using Product-Based Analytical Models," *Computer Aided Design in Electronic Packaging,* ASME WAM, EEP-Vol 3, Anaheim CA, pp 41-56.

Peak, R. S. (expected 1993) *Product Model-Based Analytical Models (PBAMs): A New Representation of Engineering Analysis Models*, Doctoral Thesis, Georgia Institute of Technology, Atlanta GA.

PTC, 1992, *Pro/ENGINEER Interface Guide*, Parametric Technology Corp., Waltham MA.

SDRC, 1991, *I-DEAS Bi-Directional EDA Interface User's Guide*, Structural Dynamics Research Corp., Milford, OH.

Shephard, M. S., Baehmann, P. L., LeCoz, Y. L., Sham, T. L., 1992 "Methodology for the Integration of Global/Local Thermal and Thermo-Mechanical Analyses of Multichip Modules," *Computer Aided Design in Electronic Packaging,* ASME WAM, Anaheim CA, EEP-Vol 3, pp 65-72.

Steinberg, D. S., 1988, *Vibration Analysis for Electronic Equipment*, John Wiley & Sons, New York.