# INTEGRATING TRADEOFF ANALYSIS AND PLAN-BASED EVALUATION OF DESIGNS FOR MICROWAVE MODULES

Vinai S. Trichur* Michael O. Ball† John S. Baras‡ Kiran Hebbar§
Ioannis Minis¶ Dana S. Nau|| Stephen J.J. Smith**

August 19, 1996

**Abstract**

Previously, we have described two systems, EDAPS and EXTRA, which support design and process planning for the manufacture of microwave modules, complex devices with both electrical and mechanical attributes. EDAPS integrates electrical design, mechanical design, and process planning for both mechanical and electrical domains. EXTRA accesses various component and process databases to help the user define design and process options. It then supports the user in choosing among these options with an optimization based tradeoff analysis module.

In this paper, we describe our current work towards the integration and enhancement of the capabilities of EDAPS and EXTRA. We integrate EXTRA's functionality with the initial design step of EDAPS. In the resultant system, the user, supported by an enhanced tradeoff analysis capability, can select and describe a promising preliminary design and process plan based on the analysis of a variety of alternatives from both an electrical and mechanical perspective. This preliminary design is then subjected to further analysis and refinement using existing EDAPS capabilities. In addition to the integration of these two systems, specific new functions have been developed, including tradeoff analysis over a much broader set of criteria, and the ability of the tradeoff module to query the process planner to determine costs of individual options.

## 1 Motivation

One of the primary motivations for Integrated Product and Process Design (IPPD) is the observation that 70% to 80% of manufacturing cost is determined while the product is being designed [42]. In

*Institute for Systems Research and College of Business and Management, University of Maryland, College Park, Maryland. Email: vtrichur@isr.umd.edu

†Institute for Systems Research and College of Business and Management, University of Maryland at College Park. Email: mball@bmgtmail.umd.edu

‡Institute for Systems Research and Department of Electrical Engineering, University of Maryland at College Park. Email: baras@isr.umd.edu

§Institute for Systems Research and Department of Mechanical Engineering, University of Maryland at College Park. Email: hebbar@isr.umd.edu

¶Institute for Systems Research and Department of Mechanical Engineering, University of Maryland at College Park. Email: minis@isr.umd.edu

||Institute for Systems Research and Department of Computer Science, University of Maryland at College Park. Email: nau@cs.umd.edu

**Institute for Systems Research and Department of Computer Science, University of Maryland at College Park. Email: sjsmith@cs.umd.edu

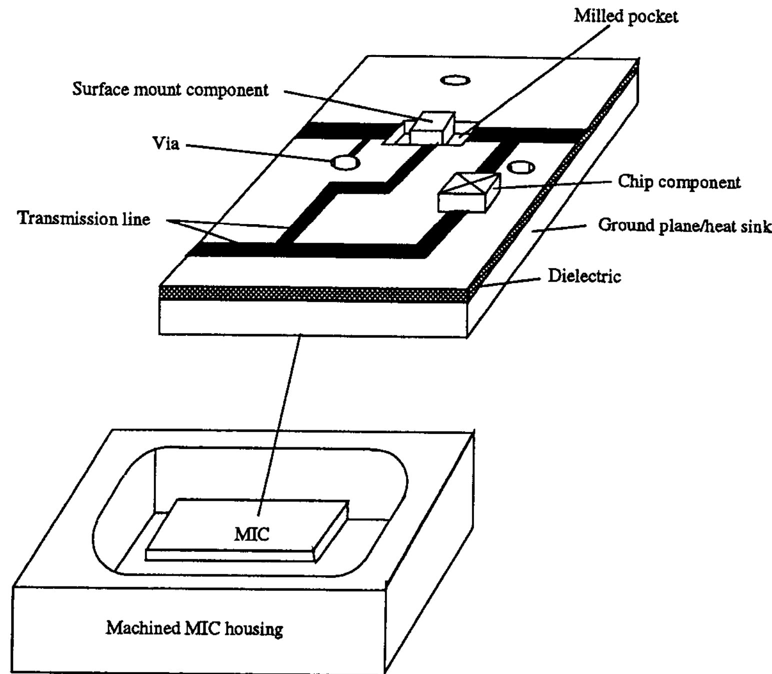


Figure 1: A typical microwave module, consisting of the MIC substrate, and its housing.

order to avoid expensive and time-consuming *design-manufacture-test-redesign* cycles, it is important to address design and manufacturing issues concurrently.

As an example, consider the design and manufacture of complex devices such as *microwave modules*. These are electronic devices that operate in the 1-20 GHz frequency range and are used in many modern telecommunications systems. A microwave module consists of several *Microwave Integrated Circuits (MICs)*. The functional components of MICs, such as transistors, resistors, and capacitors can be classified as either "integrated" or "hybrid". *Integrated* components are fabricated as portions of the artwork. *Hybrid* components are assembled using techniques such as soldering, wire bonding, and ultrasonic bonding. An MIC can be classified as a layered electrical product (LEP) [9] because it is made up of several functional layers. Figure 1 illustrates a typical microwave module. The topmost layer is usually made of the *artwork*, which contains an etched pattern of transmission lines that realize the functionality of the circuit. The next layer is the *dielectric layer* that serves as a wave-conducting medium. The bottom layer is an aluminum layer that provides grounding for the device, and acts as the heat sink. Discrete components are mounted on the top layers. The aluminum ground plane has other features such as electrical and thermal vias and machined pockets to mount the components. Other than MICs, a microwave module consists of a machined housing, assembly components to mount the MICs to the housing, and heat removal mechanisms to conduct away heat from the module to heat sinks.

Figure 2 illustrates the design and manufacturing cycle for microwave modules, which is highly interdisciplinary in nature. Electronic designers develop the detailed circuitry; mechanical designers design the device to have desired dynamic, heat transfer, and strength characteristics; and manufacturing engineers select the tooling, processes, and process parameters to manufacture the module. The choice of manufacturing processes depends on several factors, such as the type of dielectric material, and the degree of integration of functional elements of the design. If all elements are assembled as hybrids, then lamination, photomask deposition, etching, plating, adhesive deposition,

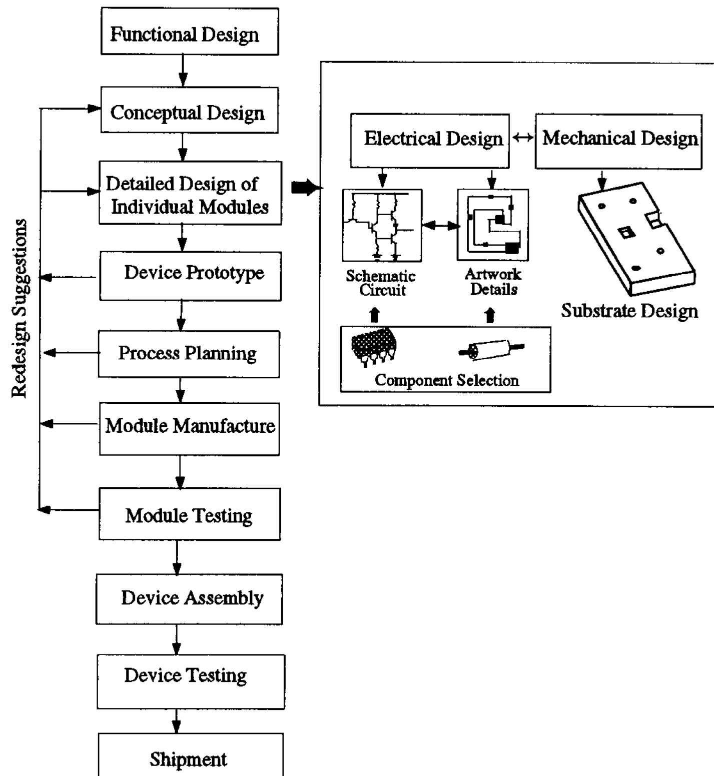


Figure 2: Design and manufacturing cycle for microwave modules.

application of flux, reflow soldering, trimming, cleaning, testing, tuning, drilling, milling, and casting form a superset of the operations used [12, 8]. If, however, some components are fabricated as integrated elements, thin film and thick film deposition techniques may be used [20].

At each of the above stages the designer is faced with the task of choosing among competing alternatives. Specifically, even for a given circuit schematic, a large number of choices are available for components and processes. For example, a resistor of given specifications could be available as both leaded and surface mount, and offered by a number of vendors with differing cost and quality attributes. These differences could, in turn, require different processes for assembly (placement on the board) and electrical connection (soldering). Additionally, there may be quantity discounts associated with purchasing– a fact that should be taken into account when selecting components. It is thus apparent that designers are faced with a large number of options and with the associated tradeoffs. Consequently, along with the manufacturability tools reported in the literature, there is a distinct need for models that efficiently explore the search space to identify "good" design options in terms of cost, quality and other metrics.

Furthermore, the tasks of an IPPD team for electro-mechanical products could be greatly aided by a carefully designed computer system that integrates both electronic and mechanical computer-aided design (CAD) tools, supports tradeoff analysis, and provides access to process planning and design evaluation capabilities.

To explore these issues, we created two systems - *Electro-mechanical Design And Planning System* (EDAPS) [21], a toolkit for microwave module design that integrates electronic and mechanical computer-aided design, electronic and mechanical process planning, and plan-based design evaluation, and EXTRA [3, 4], an optimization tool that chooses from a set of alternate parts and

processes, and performs tradeoff analysis with respect to a set of metrics (cost, yield, lead time and so forth). In this paper we describe our enhancement of EXTRA's capabilities, and its integration with EDAPS into a single environment for microwave module design.

# 2 Related Research

There have been a number of efforts in the direction of achieving IPPD in industry. Initial efforts were focused on establishing guidelines to inform designers of manufacturing and assembly concerns to be addressed at the design stage [7, 6, 2].

Using design for assembly guidelines, Jakiela and others [23] built a rule-based Design-For-Assembly system that gives feedback about assemblability when the designer adds new features to the design. Gupta *et al.* [17] have developed IMACS, which generates the best operation plans for machined components and gives feedback about manufacturing infeasibilities in the design. However, none of these tools are applicable to the electronic domain.

In the electronic domain, Harhalakis *et al.* [18] have developed a rule-based system for critiquing the manufacturability of microwave modules. This system is not directly linked to an electronic or mechanical CAD system. Feldmann and others [16] describe a system that integrates electronic and mechanical CAD tools for three-dimensional molded printed circuit boards, where circuits are no longer in planar configurations. These tools do not evaluate the designs with respect to cost and lead times.

Schemes for costing a given microwave module design can be found in a number of sources (see, for example, [32, 19, 5]). Reviewing this body of literature reveals the following:

- Attempts to determine optimal designs (rather than assessing a given design) with respect to cost have been rather limited. Oh and Park [32] use a dynamic programming approach to optimize the assembly processes; however, their procedure does not appear to be very practical for situations having a large search space of design alternatives. The only other optimization application we have come across is Russell [35] (cf., [34]).
- The quality of a given design is factored into the analysis usually via a rework cost. On the other hand, we see cost and quality as competing (and often conflicting) evaluation metrics for which a *multi-criteria* approach seems to be more appropriate.
- 'Business metrics', such as supplier lead times and quantity discounts have not been taken into account when evaluating a design.

Commercially, several CAD tools are available for electronic circuitry design (such as Mentor Graphics, OrCAD, EEsof, and MAGIC). These electronic CAD packages automatically check design rules, and some even perform manufacturing yield analysis of the design [15]. However, since these packages use a two-dimensional representation of the design, they neither represent three-dimensional mechanical features nor perform any sort of mechanical feasibility and manufacturability analysis. Such tasks require a three-dimensional solid-model representation of the design.

*Process planning* can be defined as the act of preparing detailed operating instructions that transform an engineering design to a final part [10]. *Computer-Aided Process Planning (CAPP)* systems have been traditionally classified as *variant* or *generative*. In the variant approach, new designs are matched with old designs the process plans of which are stored in a database. Process plans for the matching old designs are retrieved and manually modified to suit the manufacturing of the new design. Thus, a degree of human involvement is necessary in generating the plans. In the generative approach, decisions needed to convert the stock to final design specifications are automatically taken by the computer by means of process knowledge, logic that performs geometric

reasoning on the part, and other decision logic that is built into the system. A comprehensive review of numerous CAPP systems in the mechanical domain can be found in [40].

Some efforts have focused on CAPP for electronic applications (for a review, see [28]). The PWA-Planner [11] is a rule-based system that performs planning for assembly of components on placement machines. Sanii [39] and others have used Artificial Intelligence (AI) approaches to develop plans for assembling PCBs. Young *et al.* [27] have developed a process planning and concurrent engineering system for PCBs that represents process knowledge as constraints and provides manufacturability feedback on the design. It is pointed out, however, that PCB manufacturing usually does not involve traditional mechanical processes, such as milling, that are necessary in the manufacture of a microwave module.

For classifying electro–mechanical designs, Candadai and others [9] have taken a Group-Technology-based approach. Based on this approach, Lam [26] generates high level process plans for the manufacture of these designs in a multi-enterprise setting. Though Lam's system considers the manufacturing processes in both domains, it does not work concurrently with an electronic CAD tool.

Tradeoff analysis is an important part of design evaluation whenever designs can be evaluated according to more than one criterion, such as in the electro-mechanical domain. The MSDA advisor is a software tool that evaluates system level tradeoffs between physical size, weight, thermal characteristics, reliability, cost, performance, and so forth in the selection of packaging technologies for components used in PCBs and ceramic substrates. This tool has been used to perform tradeoffs [38].

# 3 System Architecture

This study provides the designer with CAD tools for electronic and mechanical design, an integrated process planning and evaluation mechanism for manufacturing processes both in the mechanical and in the electronic domain, and with an integrated tradeoff analysis tool that can select among alternate parts and processes and handle a variety of metrics. Parts of the system are currently being developed. When completed, as illustrated in Figure 3, the resulting system will consist of four modules that can be invoked from a common user interface:

- In the *circuit schematic and circuit layout module*, the designer generates electronic circuitry. An integrated set of packages supplied by EEsof's Series IV [15] software forms the core of this module. To complete this module, we have developed routines to extract manufacturing related information from electrical designs. Section 4.1 addresses the circuit layout module in more detail.

- In the *tradeoff analysis module*, alternate realizations of the circuit schematic are generated; all these realizations are 'efficient' with respect to the metrics being considered. Each realization is obtained by making specific choices among the available alternatives for parts and processes. This module consists of an optimization 'engine' written in C, and calls CPLEX, a state of the art linear and integer programming package. This module is currently under construction and is described in more detail in Section 5.

- In the *substrate design module*, the designer develops mechanical features of the substrate containing the artwork. Bentley Systems' Microstation CAD software application [29] supplies the set of tools required to achieve this functionality. We have developed routines in the Microstation Development Language to integrate Microstation with the rest of the system and to extract and supply relevant manufacturing information to individual modules. We address the substrate design module in more detail in Section 4.2.

- In the *process planning and plan evaluation module*, our AI-based process planner creates a process plan for the detailed design, and reports to the designer the cost, manufacturing lead

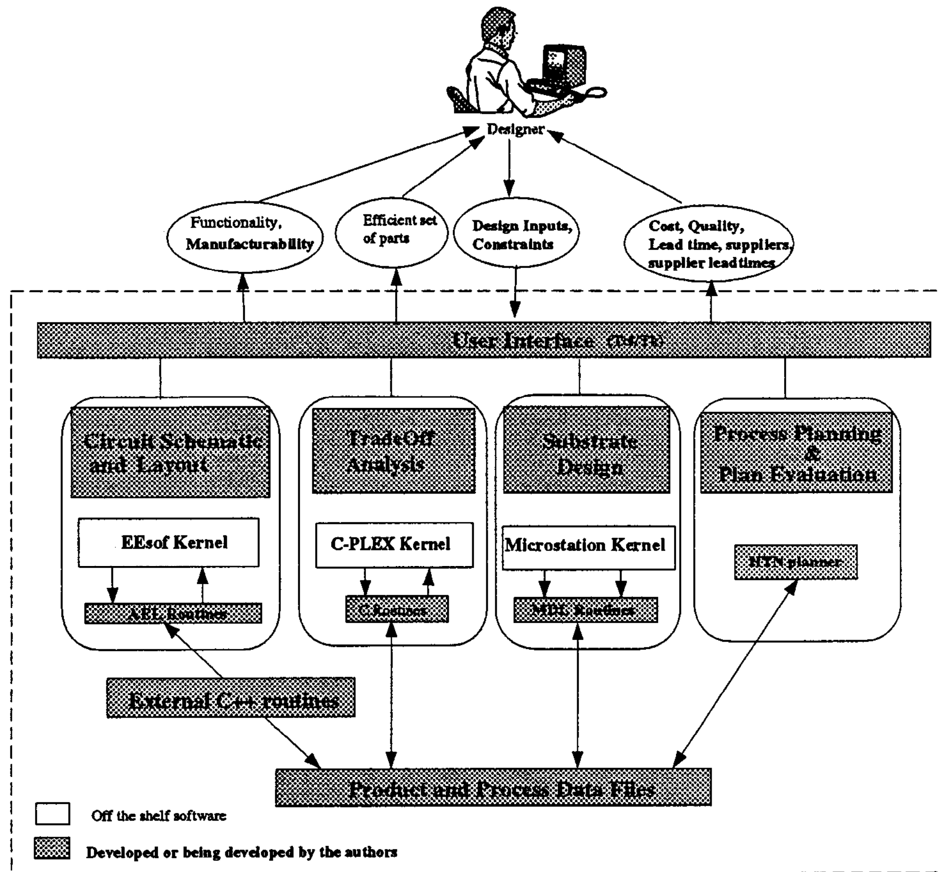


Figure 3: System architecture.

time, and quality for the design. We are currently extending the process planner to deal with schematics; given a schematic and any alternative combination of components, the enhanced planner will be able to provide a rough (perhaps incomplete) plan and rough estimates of cost, manufacturing and purchasing lead times, quality, and so on. We address the process planning and plan evaluation module in more detail in Section 4.3.

The substrate design module and the process planning and plan evaluation module are invoked for each realization of the circuit schematic that is generated by the tradeoff analysis module. Thus, the tradeoff analysis module essentially 'narrows' the search space upon which the other modules perform a more detailed analysis.

The coordination of these modules and the exchange of data among them takes place through a user interface written in the *Tcl/Tk* language [33]. This user interface allows the designer to smoothly interact with the heterogeneous modules that constitute the system.

# 4 Overview of EDAPS Modules

The modules of *EDAPS* are described in [21]. A brief overview of the components of EDAPS that are used in the proposed system is given in this section.

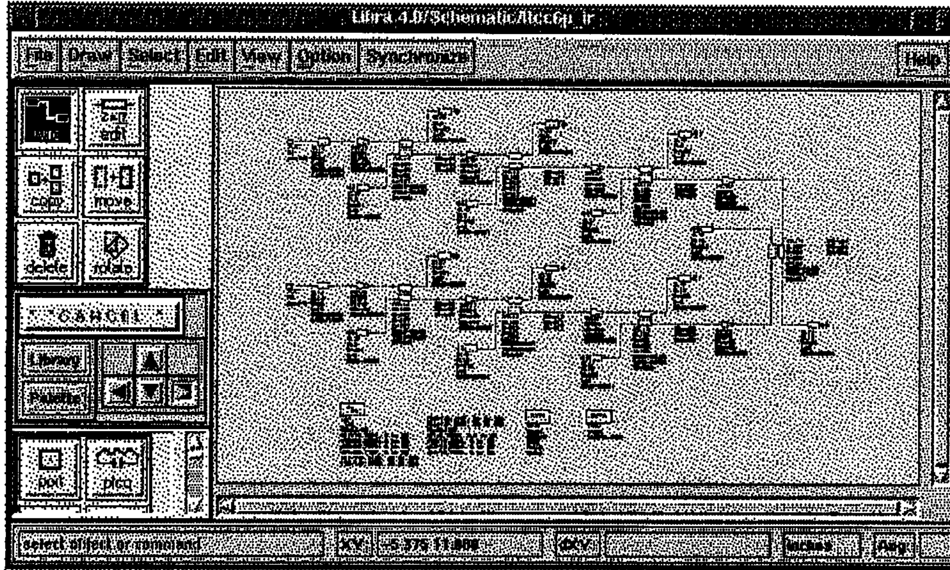


Figure 4: Schematic circuit design using Libra.

## 4.1 Circuit Schematic and Layout

For microwave circuit design and layout, EDAPS uses EEsof's Libra tool for linear and nonlinear schematic circuit design, and EEsof's ACADEMY for layout generation.

Using Libra, the designer designs the "schematic circuit", choosing components from pre-defined and user-defined device libraries. In *schematic circuits*, the components and transmission lines are represented as symbols. The designer subjects this circuit to time/frequency domain response analyses to test its functionality. The designer may perform several design iterations, until a functionally satisfactory circuit is obtained. Figure 4 shows the schematic circuit of an oscillator designed with Libra.

Libra incorporates some design-for-manufacturing principles. Based on the required circuit functionality, the limiting tolerances on each component's electrical parameters can be calculated and thus manufacturing yield can be predicted. The yield metric is the maximum expected yield and is useful in performing sensitivity analysis of the design. However, manufacturing yields are not only a function of electrical parameter tolerances; other manufacturing factors can affect yield too.

Once the schematic circuit is complete, the artwork shapes necessary to realize circuit interconnections and other metalizations on the substrate are automatically generated by ACADEMY. The layout can also be interactively developed to fit the artwork within specified size constraints, and to incorporate those artwork layer elements that do not have electronic significance. Examples of such elements are product identification numbers, design version numbers, fiducial marks, and the global origin for the microwave module. Figure 5 illustrates the layout for the schematic circuit for the oscillator of Figure 4.

We have built routines in EESof's *Application Extension Language (AEL)* supplied by EEsof to extract the list of components that form the design. We have also built routines in C++ to query the design database and obtain component parameters, and artwork parameters that influence manufacture, such as trace thickness. We have also developed routines in C++ that translates the layout data into the IGES format for export to the mechanical CAD system, described in Section 4.2.

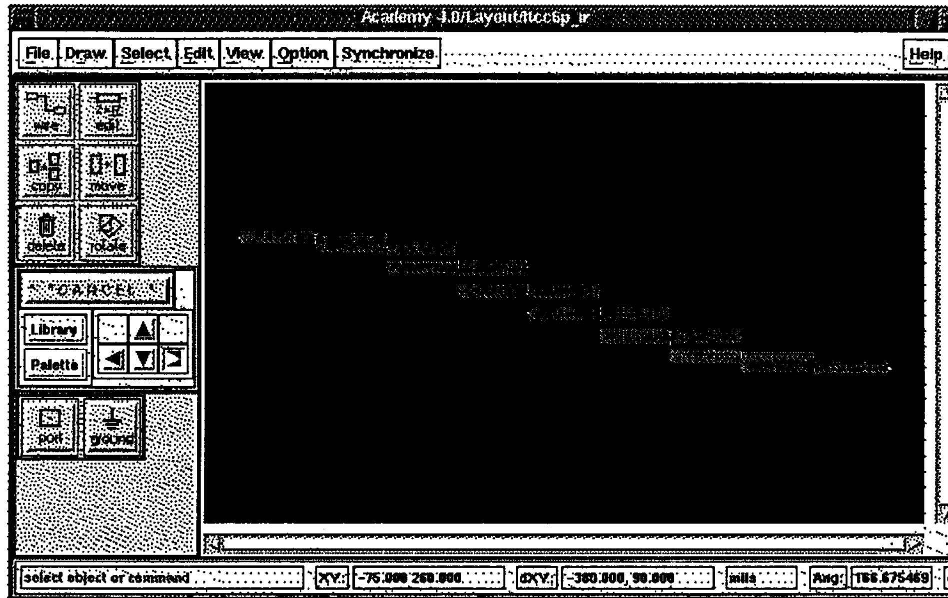


Figure 5: Layout generated by ACADEMY.

## 4.2 Substrate Design Module

The substrate design module is built using Microstation, a feature-based comprehensive CAD package supplied by Bentley Systems Inc. Features in Microstation are represented as nodes of the design tree. Nodes in the feature tree are created in the order they are built into the design by the designer. The ACIS [1] solid modeler is used internally to represent and provide methods to generate and modify solid models resulting from feature operations.

Several manufacturing applications prefer to work with *manufacturing features* that contain information helpful in reasoning about manufacturing feasibility, cost, time, and so on. The manufacturing features that are most relevant to process planning and plan evaluation in our application domain are:

- *Dielectric* : The dielectric is assumed to have prismatic geometry with designer specified corner radii. The feature parameters contain dimensions, corner radii, location, orientation, and electronic related data such as dielectric constant and dielectric material.

- *Heat Sink*: The geometry of this feature is again assumed to be prismatic with corner radii. The feature parameters include material, length, width, height and corner radius. An additional constraint specifies that the widths and lengths of the heat sink and dielectric be equal.

- *Component Mounting Pockets*: For each packaged component that requires an appropriate recess in the substrate for mounting and grounding, we have provided component mounting pocket features; their geometry corresponds to an end-milling feature. By default, the dimensions of this feature are a function of the dimensions of the packaged component, and its location is the same as that of the packaged component. This generic end-milling feature is used to construct all other cutouts, pockets, and grooves in the dielectric and heat sink.

- *Vias*: Vias directly correspond to the material removal volumes of drilling features. In addition to diameter, location, orientation, and length of the vias, useful manufacturing information is

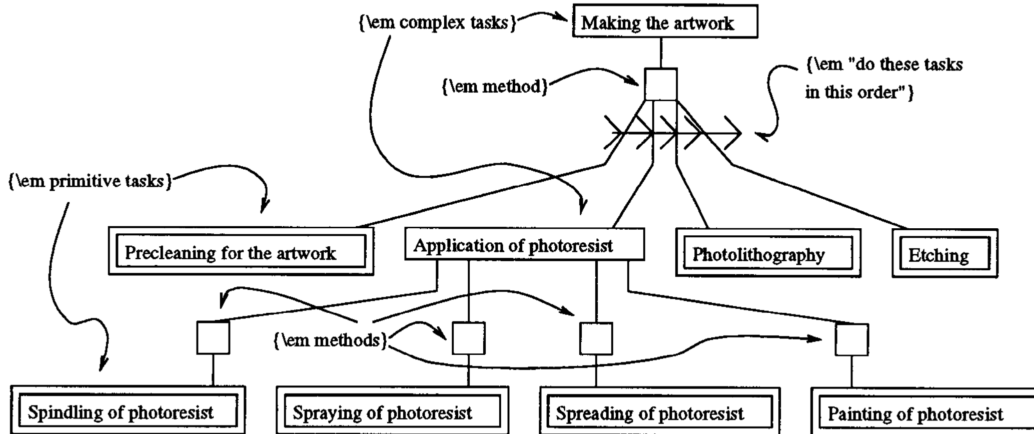


Figure 6: Part of the task network for microwave module manufacture.

provided, including electroplating thickness if electroplated, and pitch and nominal diameter if threaded.

## 4.3 Process Planning and Plan Evaluation Module

To build our process planner for microwave modules, we use *hierarchical task-network* (HTN) planning [13, 36, 43, 48].

HTN planning proceeds by taking a complex *task* to be performed and considering various *methods* for accomplishing the task. Each method provides a way to decompose the task into a set of smaller tasks. By applying other methods to decompose these tasks into even smaller tasks, the planner will eventually produce a set of primitive tasks that it can perform directly.

As an example, one method for making the artwork for the microwave substrate is to do the following series of tasks: precleaning for the artwork, followed by application of photoresist, followed by photolithography for the artwork, followed by etching. There are several alternative methods for applying photoresist: spindling the photoresist, spraying on the photoresist, painting on the photoresist, and spreading out the photoresist from a spinner. This relationship between tasks and methods results in a *task network*, part of which is shown in Figure 6.

This decomposition of tasks into various subtasks is important for planning the manufacture of microwave modules for two reasons. First, the decomposition in an HTN naturally corresponds to the decomposition of a microwave substrate into the parts and processes required to manufacture it. Second, the ability to include the complex tasks "make drilling and milling features", "make artwork", "assembly and soldering", and "testing and inspection" in sequence provides a uniform framework that can naturally accommodate all processes in mechanical and electronic manufacturing.

Sometimes a particular method can always be used to perform a particular task. For example, because spreading out the photoresist from a spinner is so accurate, this method can always be used to perform the task of applying the photoresist. Othertimes a particular method can only be used to perform a particular task. For example, because spraying on the photoresist is only somewhat accurate, this method cannot be used to apply the photoresist if a coupler in the artwork has a gap of less than or equal to 0.010 inches.

Certain tasks are *primitive*, meaning that they do not break down into any other tasks. We consider a task to be primitive if it is considered to be a single small step in the manufacturing

process. For example, precleaning of the artwork is a primitive task. Once the complex task of making the entire microwave substrate has been broken down into a series of primitive tasks, a process plan has been created; carrying out the steps of the process plan will result in the creation of the MIC.

The planning module constructs a set of process plans, and evaluates them to see which takes the least amount of time. In some cases, it evaluates a set of incomplete process plans and discards all but the one which takes the least amount of time. For example, because the method of application for photoresist does not affect the method of application for solder paste, if the quickest method of applying photoresist is spraying it on, then there is no need to generate process plans in which some other method of application is used. If no process plans can manufacture the device—because some manufacturability constraint, such as achievable tolerance, is violated—EDAPS's planner reports the failure and the reason for the failure to the designers.

This generative process planning approach allows us to provide feedback about manufacturability, cost, and lead time to the designers, based on actual process plans for the manufacture of the device. Because manufacturing engineers are accustomed to a standard format for the specification of process plans, EDAPS's planner outputs the process plan in this format. See Figure 7.

# 5 Tradeoff Analysis Module

This module helps the designer choose among the alternate parts and processes. We have enhanced the capabilities of this module considerably since its original implementation as a part of the EXTRA system. Earlier, we were primarily interested in trading off cost and manufacturing yield. Now, the tradeoff is performed with respect to five metrics - cost, manufacturing yield, number of suppliers, supplier lead time, and quantity discounts.

Figure 8 illustrates the tradeoff analysis module. The module is typically invoked after the schematic is complete and prior to designing the layout (see Section 6). Thus, the module's inputs include the list of components and their alternatives, together with those processes (and their alternatives, if any) that are related to the choice of each component. For example, if through-hole mount components are selected, then wave soldering and automated assembly, or manual assembly and soldering are possible. On the other hand, if surface mount components are selected, then reflow soldering and automated assembly, or manual assembly and soldering are possible. The component and process characteristics are needed to construct the expressions for cost and yield. The component characteristics are obtained from EEsof's library of components; i.e., for each component selected by the designer during the construction of the schematic, all alternative components that satisfy the designer's functional intent will be considered. The process alternatives are generated by a modified version of the HTN process planner which is currently being enhanced. Finally, supplier data (also obtained from EEsof) are used to construct the expressions for supplier lead time, number of suppliers, and quantity discounts.

The problem is formulated as a multi-objective integer program that is interactively solved to optimality. In the rest of this section we develop the integer programming formulation and describe the solution procedure.

## 5.1 Problem Definition

We assume that a conceptual design for the microwave module (board) is given, and is to be realized via a single assembly. The design specifies the set of required *generic* component types, and for each such component type, a number of specific component alternatives. For each specific component, we are given the list of processes that are related to the component, and the alternatives (if any)

```
Parts:
Block
 Dimensions: 7,4,1
 Ground material: Aluminum
 Substrate: Teflon
 Substrate thickness: 30 mils
 Metallized layer: Copper
 Metallized layer thickness: 7 mils
 Part number: 80280SA/2
Resistor
 Name: P1
 Part number: RNC55H2370FS
 Description: Motorola SS163
 Specification: MIL-R-55182
[...]
Processes:
Opn A  BC/WW  Setup  Run   LN  Description
001 A   VMC1    2.0  0.0   01  Hold substrate with
                               flat vise jaws at
                               3.5,4,0.5 and
                               3.5,0,0.5
                           02  Establish datum point
                               at 0,0,1
001 B   VMC1    0.0  0.6   01  Drill hole: 1,4,0
                               depth: 1 using
                               0.25 radius bit
                           02  Drill hole: 3,4,0
                               depth: 1 using
                               0.25 radius bit
001 C   VMC1    0.0  0.3   01  Drill hole: 3.5,6.5,0
                               depth: 1 using
                               0.125 radius bit
001 D   VMC1    0.0  5.0   01  Mill slot: 0.5,1,0
                               dimensions 3,1,1
                               using 0.5 radius
                               end-milling tool
001 T   VMC1    2.0  5.9   01  Total time on VMC1
```

Figure 7: Part of a process plan in a standard format.

for each such process. This defines the basic 'AND/OR tree' that captures the structure of the design [3, 4].

Key attributes such as material costs, run times, setup times, process yields, and material defect rates are assumed to be known for components, processes, and component-process combinations. In addition, we assume that supplier information is available for each component. This information consists of the lead time of the supplier and the quantity discount structure employed by the supplier. The problem is to determine a set of components (and implicitly, suppliers) and processes that are 'efficient' with respect to the five objectives mentioned earlier.

We begin by defining some notation:

$\mathcal{V}$ = set of generic components,

$\mathcal{V}_j$ = set of alternatives for the $j$th generic component,

$\mathcal{P}$ = set of all processes (including alternate processes),

$\mathcal{P}_j$ = set of 'generic' processes related to the $j$th component,

$\mathcal{P}_{ji}$ = set of alternatives for the $i$th generic process related to the $j$th component: $i \in \mathcal{P}_j$,

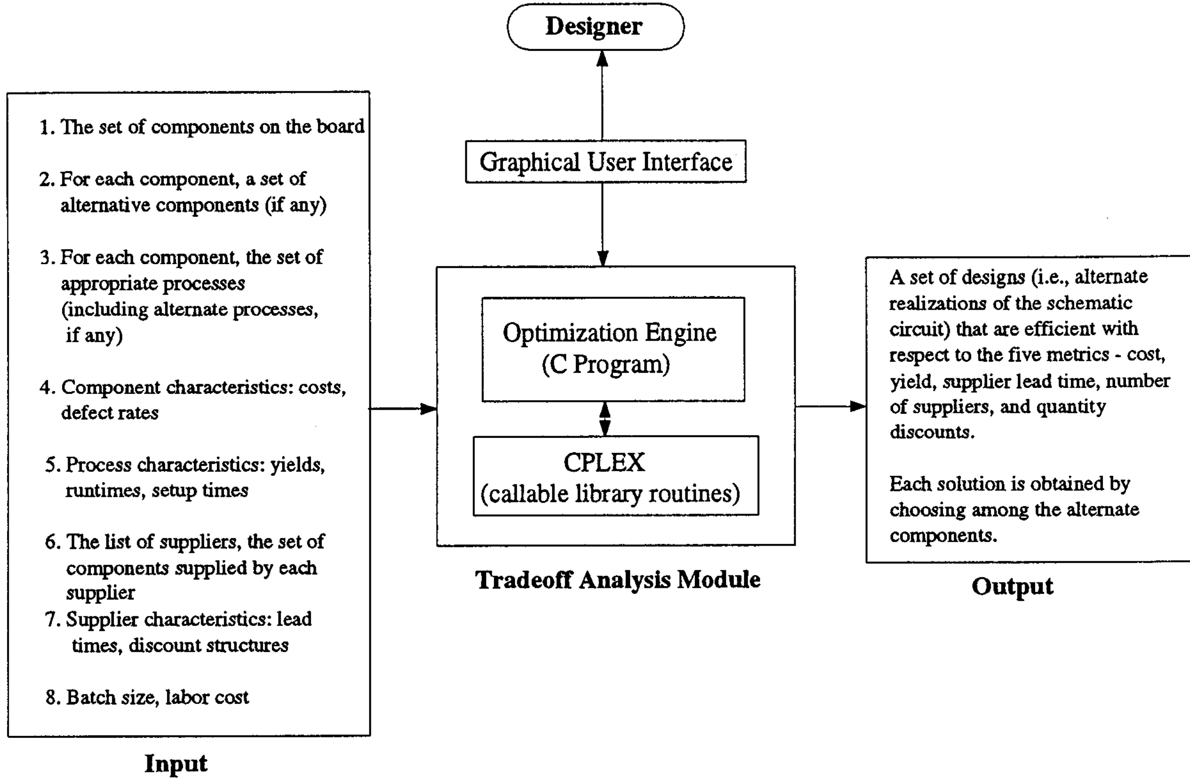


Figure 8: Tradeoff analysis module.

$\mathcal{S}$ = set of suppliers,
$\mathcal{S}_j$ = set of components supplied by $j$th supplier,
$c_j$ = unit cost of $j$th component: $j \in \mathcal{V}_k,\ k \in \mathcal{V}$,
$t_p$ = setup time of $p$th process: $p \in \mathcal{P}$,
$t_{pj}$ = runtime when $p$th process is used for $j$th component: $j \in \mathcal{V}_k,\ k \in \mathcal{V},\ p \in \mathcal{P}_{ji},\ i \in \mathcal{P}_j$,
$d_s$ = delivery lead time of $s$th supplier: $s \in \mathcal{S}$,
$\alpha_j$ = defect rate of $j$th component: $j \in \mathcal{V}_k,\ k \in \mathcal{V}$,
$\beta_p$ = yield rate of $p$th process: $p \in \mathcal{P}$,
$n_j$ = number of units of the $j$th component on the board (the redundancy element): $j \in \mathcal{V}_k,\ k \in \mathcal{V}$,
$l$ = labor cost per unit time,
$b$ = batch size.

We assume that all the above quantities are provided as input data. We now define the following decision variables:

$$x_j = \begin{cases} 1 & \text{if component } j \text{ is selected,} \\ 0 & \text{otherwise.} \end{cases}$$

$$y_p = \begin{cases} 1 & \text{if process } p \text{ is used in the assembly,} \\ 0 & \text{otherwise.} \end{cases}$$

$$
x_{pj} = \begin{cases} 1 & \text{if process } p \text{ is selected for component } j, \\ 0 & \text{otherwise.} \end{cases}
$$
$$
w_j = \begin{cases} 1 & \text{if supplier } j \text{ is selected,} \\ 0 & \text{otherwise.} \end{cases}
$$

### 5.1.1 Cost and Yield

The expressions for cost and yield are given as follows:

$$
\begin{aligned}
\text{Material cost} &= C_m = \sum_i n_i c_i x_i, && (1)\\
\text{Runtime cost} &= C_r = l \sum_{p,j} t_{pj} x_{pj}, && (2)\\
\text{Setup cost} &= C_p = \frac{l}{b} \sum_p t_p y_p, && (3)\\
\text{Total Cost} &= C = C_m + C_r + C_p, && (4)\\
\text{Yield} &= Q = \prod_p (\beta_p)^{y_p} \prod_j \left(1 - \alpha_j^{n_j}\right)^{x_j}. && (5)
\end{aligned}
$$

The decision variables in (4) and (5) ensure that only those elements that are selected to be in an assembly contribute to its cost and quality. This is the only feasible consideration at this preliminary design stage. Also, note that the labor cost, $l$, might be different for manual and automatic processes; thus, (2) may consist of two terms, one for manual processes and another for automatic processes. The yield expression, (5), essentially consists of the product of the material and process yields. We can now linearize (5) to get

$$
Q' = \log Q = \sum_p y_p \log \beta_p + \sum_j x_j \log\left(1 - \alpha_j^{n_j}\right). \tag{6}
$$

It should be noted that the yield expression, (5), can actually be incorporated into the cost equation. This is because, in practice, a module that is incorrectly manufactured is rarely, if ever, scrapped; instead, it is *reworked*, and this rework cost can form a significant component of the total cost if the manufacturing yield is low. Thus, we could define rework cost as follows:

$$
\text{Rework cost} = R = b * (1 - Q) * \text{rework cost per board} \tag{7}
$$

and add (7) to the expression for total cost. While this makes sense intuitively, it complicates the formulation—it is not clear how we would linearize the total cost expression in this case. Moreover, yield should be visible in any practical application. Hence, we feel that it is useful to retain the expression for yield as a separate objective.

### 5.1.2 Lead Time

Another objective that we consider is supplier lead time. We would like to choose components (and hence suppliers) such that the delivery lead time is as low as possible. The overall delivery lead time is given by the following expression:

$$
\text{Lead Time} = L = Max(d_1 w_1, d_2 w_2, \ldots, d_S w_S) \tag{8}
$$

This equation says that the delivery lead time is equal to the maximum of the lead times of the selected suppliers. It should be noted that L, as defined in (8), is not a linear expression. However,

it is easily linearized by replacing it with auxiliary continuous variable, $L'$, in the objective function, and adding the following constraints:

$$L' \geq d_i w_i \quad \forall i \in \mathcal{S} \tag{9}$$

Since we are minimizing the lead time, $L'$ will be automatically set equal to the maximum supplier lead time.

### 5.1.3 Number of Suppliers

We would also like to minimize the total number of suppliers. Typically there are quantity discount advantages associated with ordering more components from the same supplier; also using a smaller number of suppliers reduces overhead costs related to inventory management and component tracking. Minimizing the total number of suppliers captures these effects. The number of suppliers is given by the following expression:

$$\text{Number of suppliers} = N = \sum_{i=1}^{S} w_i \tag{10}$$

### 5.1.4 Quantity Discounts

We could also explicitly model the quantity discounts associated with placing more orders with the same supplier. One simple approach would be as follows: we assume that we receive a 'pay-back' (in dollars) when we place additional orders with a supplier; the pay-back will capture quantity discounts and other intangible benefits, such as better (on-time) performance, etc. For instance, we might receive a certain payback if we place between 1000 and 2000 orders, a greater payback if we place between 2000 and 4000 orders, and so forth.

The pay-back would increase for a while and then flatten out. It is a piecewise linear function, $d(u)$, where $u$ stands for the number of orders, and is specified by the points $\{u_i, d(u_i)\}$ for $i = 1, \ldots, k$, where $d(0) = d(u_1) = 0, d(u_i) > d(u_{i-1})$, and $d(u) = d(u_k)$ for $u > u_k$. $u_k$ represents the number of orders beyond which no additional pay-back accrues. The pay-back structure (defined by the function, $d(u)$, and $k$) could be different for different suppliers— we would thus try to place more orders with suppliers who have more lucrative pay-back structures.

We can incorporate these pay-back structures into the IP formulation as follows. First, we note that the number of orders placed with the $j$th supplier is given by:

$$\text{Number of Orders} = \lambda_j = b \sum_{i \in \mathcal{S}_j} x_i \tag{11}$$

We now define the auxiliary variables, $v_{1j}, v_{2j}, \ldots, v_{kj}, v_{k+1,j}$, where:

$$\begin{aligned} v_{1j} &= \begin{cases} 1 & \text{if } 0 \leq \lambda_j < u_1, \\ 0 & \text{otherwise.} \end{cases} \\ v_{ij} &= \begin{cases} 1 & \text{if } u_{i-1} \leq \lambda_j < u_i, \\ 0 & \text{otherwise.} \end{cases} \\ v_{k+1,j} &= \begin{cases} 1 & \text{if } \lambda_j \geq u_k, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The payback from the $j$th supplier is given by:

$$D_j = \sum_{i=1}^{k+1} d_i v_{ij} \tag{12}$$

where $d_1 = 0, d_{k+1} = d_k > d_{k-1} > \ldots > d_1$. If the pay-back structure is supplier dependent, we would replace $k$ by $k_j$, and $d_i$ by $d_{ij}$. We would also need to add the following two constraints (for each supplier) to the formulation:

$$\lambda_j \geq \sum_{i=1}^{k} u_i v_{i+1,j} \tag{13}$$

$$\sum_{i=1}^{k+1} v_{ij} = 1 \tag{14}$$

These constraints, together with the fact that we are maximizing $D_j$, and $d_i > d_{i-1}$, ensure that the appropriate $v_{ij}$ is set equal to 1.

Finally, we want to maximize the total pay-back, given by

$$D = \sum_{j \in \mathcal{S}} D_j \tag{15}$$

## 5.2 The IP Formulation

The problem we wish to solve is the following multi-objective integer program:

$$\text{minimize} \quad \begin{Bmatrix} C \\ -Q' \\ L' \\ N \\ -D \end{Bmatrix}$$

subject to

$$\sum_{j \in \mathcal{V}_k} x_j = 1 \quad k \in \mathcal{V} \tag{16}$$

$$\sum_{p \in \mathcal{P}_{ji}} x_{pj} = x_j \quad \forall j, \;\; i \in \mathcal{P}_j \tag{17}$$

$$y_p \geq x_{pj} \quad \forall p, j \tag{18}$$

$$w_i \geq x_j \quad \forall i \in \mathcal{S}, j \in \mathcal{S}_i \tag{19}$$

$$x_j,\, y_p,\, x_{pj},\, w_s \in \{0,\, 1\} \;\forall\, j,\, p,\, s \tag{20}$$

Constraints (16) and (17) capture the AND/OR tree structure of the problem. Constraints (18) and (19) tell us which processes and suppliers have been selected. We would also need to include the set of constraints (9), and, for each supplier, the constraints (11), (12), (13) and (14).

## 5.3 Discussion

It can be shown that constraints (16) and (17) define a totally unimodular matrix, implying that the associated polyhedron has integer extreme points [30]. If our formulation had only these two sets of constraints, we could have solved the UP relaxation to the IP and obtained an (integer) optimal solution. The original implementation of EXTRA involved just these constraints—consequently, we were able to develop a very fast, linear programming based algorithm that generated a set of designs that were Pareto optimal with respect to the cost and quality metrics. However, constraints (18) and (19) disrupt this unimodular structure; it is easy to show that these constraints correspond to the

constraints that define an uncapacitated facility location problem. Hence we can no longer solve the LP relaxation and be guaranteed of integer solutions—we need to solve the IP itself now. However, this formulation of the uncapacitated facility location problem is known to be "strong", meaning that IP solvers generally perform well on it, at least for problems of moderate size. Preliminary experiments with the above model show that it can be solved quite efficiently.

Since we can no longer use the LP formulation to generate a set of Pareto optimal solutions, we propose the following solution procedure. We are interested in finding *non-dominated* or 'efficient' solutions—those for which the value of any metric cannot be improved without an accompanying degradation in the value of another metric. Also, since the set of efficient solutions may be very large, we feel that it would make more sense for the optimization to proceed in an interactive manner, with the designer controlling the 'search direction.' The following approach, which we are currently implementing, is motivated by the architecture used by the CONSOL system, which performs nonlinear, multiobjective optimization [44].

First, the user specifies a set of *good* and *bad* values for each of the objectives. The *good* and *bad* values can be thought of as defining a range within which the objective is required to lie—for instance, we might want the yield to lie between 95% and 100%. The objective is not permitted to exceed its *bad* value (assuming that we are minimizing the objective.) Clearly, the narrower the range, the more difficult it becomes to restrict the objective within the range—objectives with narrow ranges are consequently given more weight during the optimization process. Thus, the user can alter the relative weights of the objectives by changing their *good* and *bad* values.

Next we *normalize* the objectives, as follows:

$$K_{norm} = \frac{K - K_{good}}{K_{bad} - K_{good}} \tag{21}$$

where $K$ stands for $C, Q', L', N$, or $D$. Thus, a value of 0 for a normalized objective corresponds to the (user-specified) best value, while a value of 1 corresponds to the (user specified) worst value. Next, we define a new variable, $Z = max(C_{norm}, Q'_{norm}, L'_{norm}, N_{norm}, D_{norm})$, and minimize $Z$. Thus, we would be minimizing the maximum deviation of an objective from its *good* value. However, this is just one of the possible approaches. For instance, we might not mind one of the objectives being close to its *bad* value, if another objective were to be correspondingly close to its *good* value. Of course, we would have to ensure that one of the objectives does not become better than its *good* value at the expense of another objective. These considerations can be taken into account by first defining the following auxiliary variables:

$$\begin{aligned}
c_n &= max(C_{norm}, 0) && (22)\\
q'_n &= max(Q'_{norm}, 0) && (23)\\
l'_n &= max(L'_{norm}, 0) && (24)\\
n_n &= max(N_{norm}, 0) && (25)\\
d_n &= max(D_{norm}, 0) && (26)\\
& && (27)
\end{aligned}$$

Now we minimize $z = c_n + q'_n + l'_n + n_n + d_n$, subject to $Z \leq 1$.

In either case, the optimization will proceed in an interactive fashion, with the user modifying the good and bad values at each stage, to yield a set of solutions.

## 5.4 Computational Experience

In order to test the strength of the above integer programming formulation, we constructed a sample problem involving 75 components (a typical microwave module board contains between 50 and 100

components.) Each of these components had between three and six alternatives. We modeled ten processes. Two of these processes were 'board specific', i.e., they had to be done irrespective of the choice of components; however, the first of these (application of photoresist) had four alternatives, while the second (flux cleaning) had five alternatives. Furthermore, some of these alternatives were applicable only to certain classes of components; for instance, washing in alcohol, one of the alternatives for flux cleaning, can be used only if the board contains no non-immersible components. The remaining eight processes were 'component specific', i.e., they were specific to particular components or sets of components. Of these three processes had alternatives—the first (application of solder paste) had three alternatives, while the second (pick and place) and third (hand soldering) had two alternatives each. Both the component and process data were based on figures supplied by a manufacturer of microwave modules. The components were supplied by ten vendors, each having a different delivery lead time and quantity discount structure.

The resultant integer program had around 450 variables and 1200 constraints. We interactively solved the IP and obtained six nondominated solutions. Each of these solutions required on the order of 10 seconds to generate, on a Sun SPARC 10 workstation running CPLEX version 3.0.

# 6 Using the Proposed System

This section describes the interaction between different modules of the system, and steps in using the system. A more detailed example of using the EDAPS system is given in [21]. Figure 9 details the usual way in which this system will be used.

**Step 1: Schematic Circuit—Circuit Schematic and Layout Module.** The designer chooses the Circuit Schematic and Layout module from the user interface. Subsequently, Libra is invoked. The designer generates an initial network of circuitry based on device specifications, choosing hybrid packaged elements such as resistors, capacitors, Wets, inductors, and diodes, and distributed elements, such as transmission lines from the library of components supplied with Libra. The designer then simulates the schematic circuit, subjecting the circuit to various forms of time domain and frequency domain inputs. The design is tested for functionality and noise levels. Upon completion of the schematic circuit, the AEL routines extract the list of components used in the circuit.

**Step 2: Extraction of alternative components—External C++ routines.** We are developing routines that, using a look up table, generate a list of alternatives for components in the design, if any such alternatives exist. All functionally similar components are considered as potential alternatives to each other.

**Step 3: Selection of processes for components—Rough Process Planning & Plan Evaluation Module.** In the domain of electronics manufacturing, components can be associated with processes to a certain extent. For example, surface mount components are always assembled using reflow soldering process, while through hole components are usually wave soldered. Next, for each component, the system determines the appropriate processes and calculates (rough) estimates of runtime and setup time, as well as the yield of each such process. This is done by a modified version of the existing HTN process planner in the Process Planning & Plan Evaluation module. Efforts for modifying the existing process planner are in progress. It should be noted that the evaluation metrics will only be rough estimates; however, these rough estimates will suffice for a 'global' tradeoff analysis the principal aim of which is to decrease the size of the search space. These process data will then be combined with the component characteristics (cost, defect rate, and supplier information) extracted in Step 2, and a file containing these data will be passed to the tradeoff analysis module.

**Step 4: Choosing among alternate components—Tradeoff Analysis Module.** The module begins by constructing a multi-objective integer program using the data from Steps 2 and 3. The user now interacts with the tradeoff analysis module to generate a set of designs that are

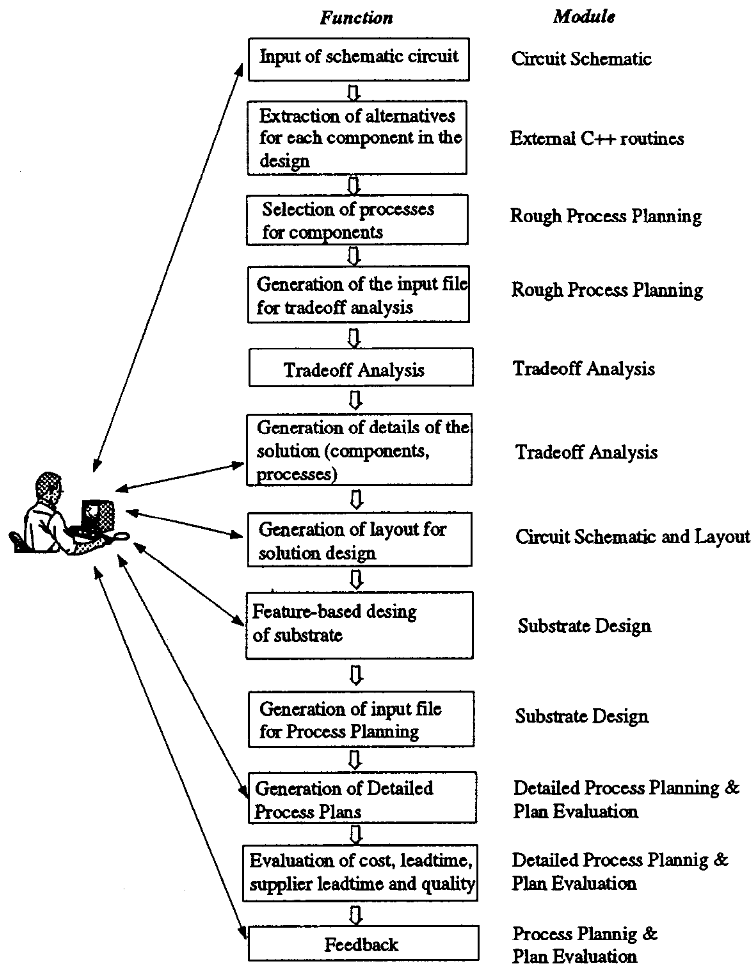


Figure 9: Using the integrated system.

'efficient' with respect to the five metrics mentioned earlier. The user generates different solutions by (effectively) changing the relative weights of the metrics. The designer then chooses one of these 'efficient' designs, for which artwork is developed in the next stage.

We are currently developing the input routines (that automatically generate the integer program) and the user interface (that permits interactive solution of the problem) for this module. The optimization 'engine' that solves the integer program has been implemented—it consists of a C program that calls the CPLEX callable library of functions.

**Step 5 : Artwork Layout—Circuit Schematic and Layout Module.** Routines written in EESof's AEL language examine a selected design from the efficient ones provided by the tradeoff analysis and input the design to ACADEMY, the artwork layout generation module of EESof. As mentioned earlier, artwork can either be automatically generated (e.g. for transmission lines) or manually specified by the designer (text items, datum elements, and so on). The artwork elements usually associated with hybrid packaged components are rectangular pads on which the components are soldered. Alternatively, the user can override default pad shape generation with user-defined shapes.

Once artwork generation is complete, the system calls an AEL routine that extracts the product

information relevant for manufacturing, such as the substrate material, dielectric thickness, conductor thickness, location and dimensions of components, and so on, from the design database. Finally, the system translates the layout into an IGES file, and exports it to Microstation for substrate designing in the next stage.

**Step 6: Mechanical design—Substrate Design Module.** The designer invokes Microstation from the user interface. The Microstation kernel reads the layout from the IGES file and re-generates the artwork as a Microstation design file. It then reads the location and dimensions of the components from the file generated in the previous step, and using this information, creates solid models of the packaged components at their correct locations.

Routines written in Microstation's development language (MDL) automatically generate milled pockets to mount components which have to be directly connected to the ground plane/heat sink. The designer has the choice to override the mounting pocket generation feature of the system at any time, and can either specify custom mounting pockets or not generate any mounting pockets at all. All other features—such as via holes, holes for clamping the MIC substrate, cutouts, and so forth—are generated by the user by boolean subtraction of the appropriate features. As explained previously, milled pockets and drilled holes have already been constructed and stored as manufacturing features.

When the mechanical design phase is complete, the system stores information about the location, type, and dimensions of all machined features and packaged components in a file that is input to the the process planner.

**Step 7: Process Planning—Process Planning and Plan Evaluation Module.** As we have mentioned, the system's HTN planner works by decomposing complex tasks into simpler tasks. The initial task, which decomposes into all other required tasks, is simply called "Make board". The rest of the plan is generated in a similar manner, and output in the format is shown in Figure 7. The output of the planner includes:

- A totally ordered sequence of process specifications that can be used to produce the finished substrate from the materials given;
- Process parameters of all the processes that are required to manufacture the device;

The output is fed back to the designers, and they can make modifications to the process plans. The process planner will also evaluate the plans for manufacturing the designs, and estimates costs, quality, manufacturing lead times, suppliers for the parts, and supplier lead times the designs. Implementation of this evaluation mechanism, described here and in the next step, is not yet complete.

**Step 8: Feedback—Process Planning and Detailed Plan Evaluation Module** When the implementation of the evaluation routines are completed, the system will feed back the above mentioned metrics, along with the cost, quality, and lead-time (manufacturing, suppliers) "hot spots" indicated. The designer can then choose to change the design by selecting alternate components for the design and rerun the entire process, until a satisfactory design is obtained.

When the designers and manufacturing engineers are satisfied with the design, the artwork elements will be extracted from Microstation in the substrate design module, and the equivalent IGES file will be generated and sent to ACADEMY in the circuit schematic and layout module. ACADEMY can then export the design file in either IGES format or Gerber format for manufacturing.

# 7 Conclusions

In this paper we have described our current work on building a design and process planning environment which integrates mechanical and electronic design tools in a single platform, supports the

choice of efficient designs among alternatives and evaluates designs based on their manufacturing plans. The system thus supports concurrent engineering by incorporating both manufacturing metrics (material cost, manufacturing yield and manufacturing lead time) and strategic metrics (number of suppliers, quantity discounts and delivery time).

Our future work will focus on completing the implementation of parts of the tradeoff analysis and the rough process planning and plan evaluation modules, integrating the modules into one system, and incorporating some more downstream considerations into our optimization model. In particular, we will investigate how the interaction between product components affects the final product quality, and then try to incorporate this quality metric into our formulation.

# References


[1] ACIS Geometric Modeler. 1993. Spatial Technology, Inc., Boulder, Colorado.

[2] Bakerjian, R, editor, 1992. *Design for Manufacturability,* volume 6 of *Tool and Manuf. Engineers Handbook.* SME.

[3] Ball, M. O.; Baras, J. S.; Bashyam, S.; Karne, R. K.; and Trichur, V. 1995. On the selection of parts and processes during design of printed circuit board assemblies. In *Proc. of the INRIA/IEEE Symp. on Emerging Technologies and Factory Automation, vol. 3*, 241–249. IEEE Computer Society Press, Los Alamitos, CA.

[4] Ball, M. O.; Baras, J. S.; Bashyam, S.; Karne, R. K.; and Trichur, V. 1995. A Cost-Yield Model for the Design of Printed Circuit Assemblies. *INFORMS Conference, Fall 1995, New Orleans.*

[5] Boothroyd, G. *Assembly Automation and Product Design.* Marcel Dekker, Inc., 270 Madison Avenue, New York, NY 10016, 1992.

[6] Boothroyd, G. and Dewhurst, P. 1983. *Design for Assembly—A Designer's Handbook.* Dept. of Mech. Engg., Univ. of Massachusetts at Amherst.

[7] Bralla, J., editor 1986. Handbook of Product Design for Manufacturing. McGraw Hill.

[8] Brindley, K. 1990. *Newnes electronics assembly handbook.* Heinemann Newnes, Oxford, England.

[9] Candadai, A.; Herrmann, J.W.; Minis, I.; and Ramachandran, V. 1994. Product and process information models for microwave modules. In *Proc. of the ASME Intl. Mech. Engineering Conf. and Exp.*

[10] Chang, T. C. and Wysk, R. A. 1985. *An Introduction to Automated Process Planning Systems.* Prentice Hall, Englewood Cliffs, CA.

[11] Chang, T. C. and Terwilliger, J., Jr. 1987. PWA Planner—a rule based system for printed wiring assembly process planning. *Comp. in Industrial Engineering* **13:1–4**, 34–38.

[12] Chenu, J. P. and Müller, H. 1989. Manufacture of Microwave Telecommunication Equipment. *Electrical Communication* **63:2**, 159–167.

[13] Currie, K. and Tate, A. 1985. O-Plan—control in the open planner architecture. BCS Expert Systems Conf., Cambridge University Press, UK.

[14] Dong, J. 1994. Feature-based manufacturing process planning for integrated product and process development. In *Proc. of the Fourth Intl. Conf. on Comp. Integ. Manuf. and Autom. Techn.*, 9–16.

[15] EEsof Series IV version 4. 1992. EEsof Inc., Westlake Village, CA.

[16] Feldmann, K. and Franke, J. 1993. Computer-Aided Planning Systems for Integrated Electronic and Mechanical Design. *IEEE Trans. on Comp., Hybrids, and Manuf. Tech.*, **16:4**, June 1993, 377–383.

[17] Gupta, S. K. 1994. *Automated Manufacturability Analysis of Machined Parts.* Ph.D dissertation, University of Maryland, College Park.

[18] Harhalakis, G.; Kinsey, A.; Minis, I. and Rathbun, H. 1993 Manufacturability evaluation of electronic products using group technology. In *NSF design and manuf. systems grantees conf.*, Charlotte, NC.

[19] Heng, C. A. S. and Gay, R. K. L. 1991 "Design For Manufacturability: Cost Analysis of Electronic Printed Circuit Board Assembly". In G.A. Gabriele, editor, *Advances in Design Automation - Volume 1*, pages 63–67. ASME, 345 East 47th Street, United Engineering Center, New York, NY 10017, 1991.

[20] Hoffmann, R. K. 1987. *Handbook of Microwave Integrated Circuits.* Artech House, Norwood, MA.

[21] Hebbar, K.; Smith, S. J. J.; Minis, I.; and Nau, D. S. 1996. Plan-based evaluation of designs for microwave modules. ASME 1996 Design Engineering Technical Conference and Computers in Engineering Conference, Irving, California.

[22] *The Initial Graphics Exchange Specification (IGES) Version 5.1.* 1991. IGES/PDES Org., Gaithersburg, MD.

[23] Jakiela, M. and Papalambros, P. 1989. Design and implementation of a prototype intelligent CAD system. ASME J. of Mech., Transm., and Autom. in Design, **111:2**, June 1989, 252–258.

[24] Konishi, Y. 1991. *Microwave Integrated Circuits.* Marcel Dekker, New York, New York.

[25] Ishii, K. 1993. Modeling of concurrent engineering design In Andrew Kusiak, editor, *Concurrent Engineering: Automation, Tools and Techniques*, pp 19–39, John Wiley & Sons, Inc.

[26] Lam, G. 1995. *Automated High Level Process Planning for Mechanical and Electro-Mechanical Components for Agile Manufacturing.* M.S. thesis, University of Maryland.

[27] Liau, J., S. and Young, R. E. 1993. A process planning and concurrent engineering system for PCBs. *Manuf. Review* **6:1**, March 1993, 25–39.

[28] Maria, A. and Srihari, K. 1992. A review of knowledge-based systems in printed circuit board assembly. *The International Journal of Advanced Manufacturing Technology* **7**:368–377.

[29] *Microstation Version 5.* 1995. *Bentley Systems, Inc.*, Exton, Pennsylvania.

[30] Nemhauser, G. L. and Wolsey, L. A. *Integer and Combinatorial Optimization.* John Wiley & Sons, Inc., 1988.

[31] O'Grady, P.; Young, R. E.; Greef, A.; and Smith, L. 1991. An advice system for concurrent engineering. *International J. of Computer Integrated Manuf.* **4:2**, March 1991, 63–70.

[32] Oh, C.J. and Park, C.S. 1993 "An Economic Evaluation Model for Product Design Decisions under Concurrent Engineering". *The Engineering Economist*, 38(4):275–297, Summer 1993.

[33] Ousterhout, J. K. 1994. *Tcl and the Tk Toolkit.* Addison-Wesley, Reading, MA.

[34] Padhy, S. K. and Dwivedi, S. N. "An Object-Oriented Knowledge-Based System for Assembly of Printed Circuit Boards". In G.A. Gabriele, editor, *Advances in Design Automation - Volume 1*, pages 43–54. ASME, 345 East 47th Street, United Engineering Center, New York, NY 10017, 1991.

[35] Russell, G. A. "Design for Assembly of Printed Circuit Boards". In *Proceedings of the First International Conference on Product Design for Assembly*, Newport, RI, April 15-17 1986.

[36] Sacerdoti, E. D. 1977. *A Structure for Plans and Behavior.* American Elsevier Publishing Co.

[37] Salomons, O.W. 1993. Review of research in feature-based design. *J. of Manuf. Systems*, **12:2**. 113–132.

[38] Sandborn, P.A. 1992. A software tool for technology tradeoff evaluation in multichip packaging. In *Eleventh IEEE/CHMT Intl. Elec. Manuf. Tech. Symp.*, 337–341. IEEE, New York, NY.

[39] Sanii, E. T. and Liau, J. S. 1993. An expert process planning system for electronics PCB assembly. *Comp. in Elec. Engg.* **19:2**, 113–127.

[40] Jami Shah, Martti Mantyla, and Dana Nau, 1994. *Advances in Feature Based Manufacturing* Elsevier/North Holland.

[41] Smith, S. J. J.; Nau, D. S.; and Throop, T. 1996. A planning approach to declarer play in contract bridge. *Comp. Intelligence*, **12:1**, Feb. 1996.

[42] Suh, N. P. 1990. *The Principle of Design.* Oxford University Press, New York, New York.

[43] Tate, A. 1977. Generating project networks. In *Proc. 5th International Joint Conf. Artificial Intelligence*, 888–893. Morgan Kaufmann, San Mateo, CA.

[44] Tits, A. L.; Ma, Z. 1986. Interaction, Specification Refinement, and Tradeoff Exploration in Optimization-Based Design of Engineering Systems. In *Proc. of the 1985 IFAC Workshop on Control Applications of Nonlinear Programming and Optimization*, 189–194. Pergammon Press.

[45] G. Toye; M.R. Cutkosky; L.J. Leifer; J.M. Tenenbaum; and J. Glicksman. 1994. SHARE: A Methodology and Environment for Collaborative Product Development. *The Intl. J. of Intelligent and Cooperative Info. Systems*, vol.3, no.2, June 1994, p. 129-53.

[46] Trinogga, L. A.; Kaizhou, G.; and Hunter, I. C. 1991. *Practical Microstrip Design.* Ellis Horwood, Chichester, UK.

[47] Whitney, D E; Nevins, J L; De Fazio, T L; and Gustavson, R. 1993 Problems and Issues in Design and Manufacture of Complex Electro-mechanical Systems. *C S Draper Laboratory Report*, R-2577, MIT, December 1993.

[48] Wilkins, D. E. 1984. Domain independent planning: representation and plan generation. *Artificial Intelligence* **22**:269–301.