Integrated Product and Process Design of Microwave Modules Using AI Planning and Integer Programming

Dana Nau, Michael Ball, John Baras, Abdur Chowdhury, Edward Lin, Jeff Meyer, Ravi Rajamani, John Splain, and Vinai Trichur
Dept. of Computer Science and Institute for Systems Research, University of Maryland
R. H. Smith School of Business and Institute for Systems Research, University of Maryland
Dept. of Electrical & Computer Engr. and Institute for Systems Research, Univ. of Maryland
IITRI
Institute for Systems Research, University of Maryland
GTE/BBN Technologies
RWD Technologies
Mitretek Systems
i2 Technologies

Key words: Integrated Product and Process Design (IPPD), Design for Manufacturability (DFM), Artificial Intelligence (AI), Integer Programming (IP), Microwave Modules (MWMs)

Abstract: A This paper describes the process planning techniques we developed for use in an Integrated Product and Process Design (IPPD) tool for the design and manufacture of microwave transmit/receive modules. Given a collection of data about the design of a microwave module, the IPPD tool uses a combination of AI planning and optimization-based tradeoff analysis to produce a collection of alternative designs and alternative process plans that have Pareto optimal values for manufacturing and purchasing lead time, process yield, cost, and number of suppliers. The IPPD tool provides facilities to enable the user to generate and examine these Pareto optimal alternatives in real time, in order to provide immediate feedback on how to modify the design to improve its cost and productivity.

1. INTRODUCTION

This paper describes an Integrated Product and Process Design (IPPD) tool for the design and manufacture of microwave transmit/receive modules. Microwave modules are complex electronic devices that operate in the 1–20 GHz range. The IPPD tool was developed as part of a contract with Northrop Grumman Corporation’s Electronic Sensors and Systems Division (ESSD) division in Baltimore. We designed it to combine high performance, ease of understandability by manufacturing personnel, ease of maintenance, and integration with other systems.

The IPPD tool uses a combination of Artificial Intelligence (AI) planning and integer programming (IP) optimization techniques to produce a collection of design alternatives. Each alternative is a collection of *design elements* (the electronic and mechanical parts to be used in the design) and *process-plan elements* (the manufacturing processes needed for the parts used in the design). The system considers the following design and manufacturing criteria: lead time (including manufacturing time and purchasing lead time); process yield; cost; and number of suppliers Each design alternative generated by the system is Pareto optimal in the sense that one design criterion cannot be improved without degrading the performance of another.. The system's GUI enables users to generate and examine the design alternatives in real time, in order to provide immediate feedback on how to modify the design to improve its cost and productivity.

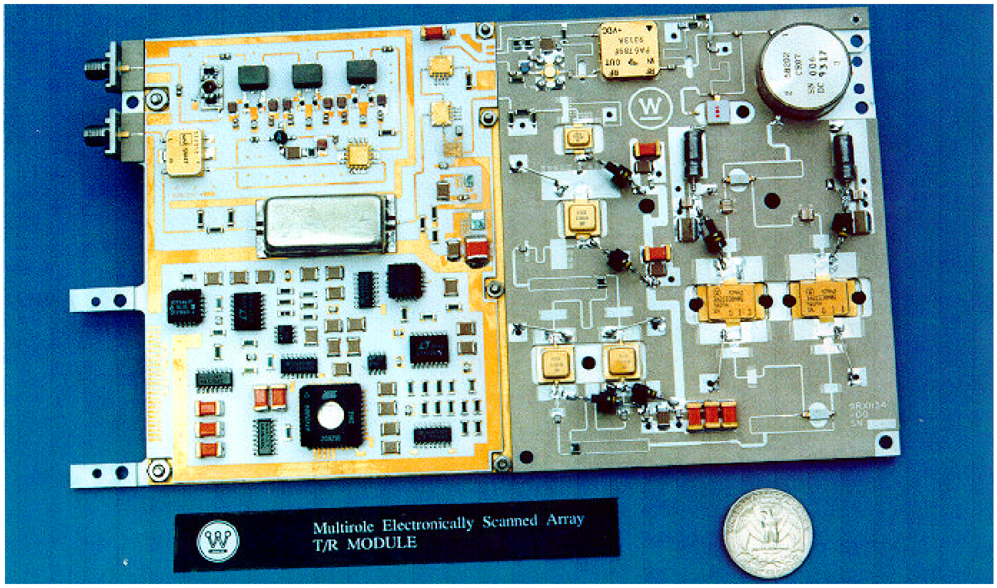


Figure 1. A typical microwave transmit/receive module.

2. MICROWAVE MODULES

Most commercial electronic products operate in the 10kHz–1GHz radio frequency spectrum. However, in the telecommunications arena, the range of operation frequency has been increasing at a tremendous pace. For scientific and commercial long-range defense applications—such as radar, satellite communications, and long-distance television and telephone signal transmissions—radio frequencies prove unsuitable, primarily due to the high noise-to-signal ratio associated with radio frequencies. Moreover, the lower-frequency bands have become overcrowded due to the overuse of these bands for commercial communications applications (Trinogga *et al.* 1991). Consequently, in contrast to other commercial electronic products, most modern telecommunications systems operate in the 1–20 GHz microwave range, and modules of such systems are called *microwave modules*.

Figure 1 shows a typical microwave module, which includes a number of *parts* mounted on a *substrate*. One factor influencing the construction of such modules is the large effect that mechanical characteristics can have on the electronic performance of microwave devices. In order to achieve proper performance, the lengths and widths of the printed wires must be chosen carefully; and to minimize vibration, the substrate is made of aluminum with a teflon coating to provide electrical insulation.

When designing a microwave module, designers and manufacturing engineers may need to choose among a large number of parts and processes in order to meet system requirements, such as cost, lead times, quality, etc. (Boothroyd 1992, Hebbar *et al.* 1996).

Parts could potentially be available in many forms (for example, a resistor could be available either with wire leads for through-hole mounting or with tabs for surface-mounting), and could be offered by a number of vendors with differing cost and quality attributes. Each of these different forms of a part could require a different set of processes in order to incorporate the part into the microwave module. The choice of these manufacturing processes depends on several factors, such as the type of dielectric material and the degree of integration of functional elements of the design.

The design task can be characterized as a problem in optimization-based tradeoff analysis. For each part that the designer specifies for use in the design, there may be several alternative parts that are suitable to be substituted for that part; and some combinations of alternatives may possibly produce better measures of overall solution "goodness." To complicate matters, some of these measures of goodness may be complimentary while others may not. For example, cost and quality are frequently directly proportional (i.e. higher quality components tend to cost more than lower quality ones), but from an optimization perspective these two attributes are at odds (generally, a designer wants to minimize cost while maximizing quality).

3. RELATED WORK

3.1 Prior Work by Others

Process planning can be defined as the act of preparing detailed operating instructions that transform an engineering design to a final part (Chang and Wysk 1985). Most work on Computer-Aided Process Planning (CAPP) has focused on the development of process plans for mechanical parts. CAPP systems have been traditionally classified as *variant* or *generative*; these are described below.

Variant process planning (which is the basis for most commercial CAPP systems) is based on the use of Group Technology (GT) coding schemes (Chang and Wysk 1985). The purpose of a GT coding scheme is to assign a fixed-lengh alphanumeric code to each design in such a way that if two designs receive the same GT code, they will require similar manufacturing processes. Given a new design, the user computes its GT code, and uses this code as a database index to retrieve a process plan for some other design having the same GT code. The user then modifies this plan by hand, to produce a process plan for the new design. In *generative* process planning, the process plan is developed automatically by the computer. The development of generative systems has been a subject of much research (for a comprehensive review, see (Shah *et al.* 1994)), but due to the difficulty of the problem, few successful commercial systems exist.

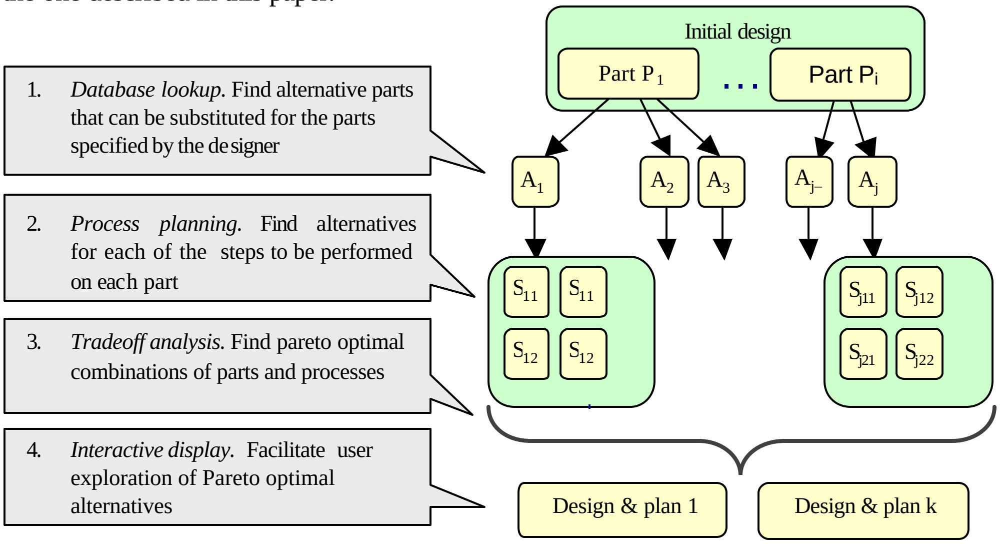
Some efforts have focused on CAPP for electronic applications (for a review, see Maria and Srihari (1992)). The PWA Planner (Chang and Terwilliger 1987) is a rule-based system that performs planning for assembly of parts on placement machines. Sanii and Liau (1993) and others have used AI approaches to develop plans for assembling PCBs; and Liau and Young (1993) have developed a process planning and concurrent engineering system for PCBs that represents process knowledge as constraints and provides manufacturability feedback on the design.

3.2 Our Prior Work

The IPPD tool described in this paper grew out of the merger of two previous projects at the University of Maryland: the EDAPS project (Hebbar *et al.* 1996, Smith *et al.* 1997) and the EXTRA project (Karne, *et al.* 1998).

EDAPS (Electro-Mechanical Design And Planning System) was an integrated design and process-planning system for microwave modules, which incorporated interfaces to electronic and commercial CAD tools, generated process plans, and provided feedback about manufacturability, cost, and lead time. EDAPS's process-planning module is a predecessor of the one described in this paper.

EXTRA (EXpert T/R module Analyst) was intended to provide an integration of enterprise-wide product database management with a tradeoff analysis optimization mechanism (Ball *et al.* 1995). EXTRA's tradeoff-analysis mechanism is a predecessor of the one described in this paper.



1. *Database lookup.* Find alternative parts that can be substituted for the parts specified by the designer
2. *Process planning.* Find alternatives for each of the steps to be performed on each part
3. *Tradeoff analysis.* Find pareto optimal combinations of parts and processes
4. *Interactive display.* Facilitate user exploration of Pareto optimal alternatives

Figure 2 Generation and evaluation of alternative designs and plans.

4. THE IPPD TOOL

Our objective in developing the IPPD tool described in this paper was to create a system to help the user perform the following tasks, which are illustrated in Figure 2:

- For each part specified by the designer, find alternative parts that might be suitable for substitution into the design in place of the original part (Nau *et al.*, 2000).
- For each alternative part, generate alternative "plan fragments", i.e., alternative collections of manufacturing processes to use on that part (Meyer *et al.*, 1998; Nau *et al*, 2000).
- Find Pareto optimal designs, i.e., combinations of parts and plan fragments that produce Pareto optimal values for the following criteria: cost, lead time, yield, and number of suppliers (Trichur and Ball, 1998).
- Select a design and a process plan from among the Pareto optimal alternatives (Splain, 1998).

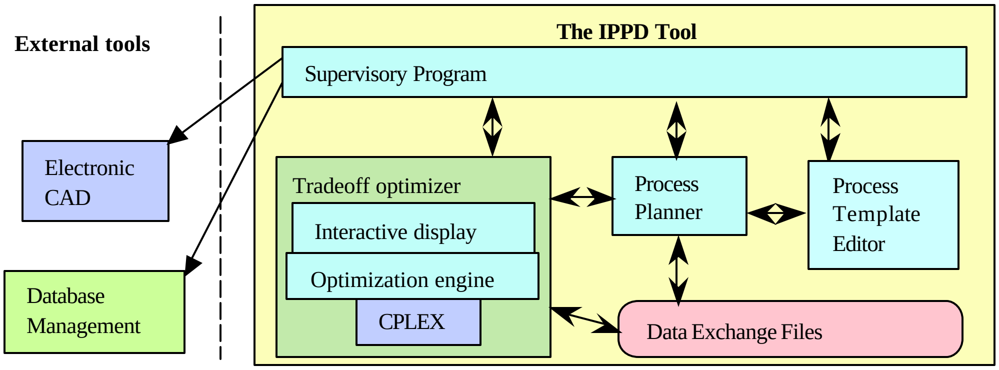


Figure 3. Overall architecture of the IPPD tool.

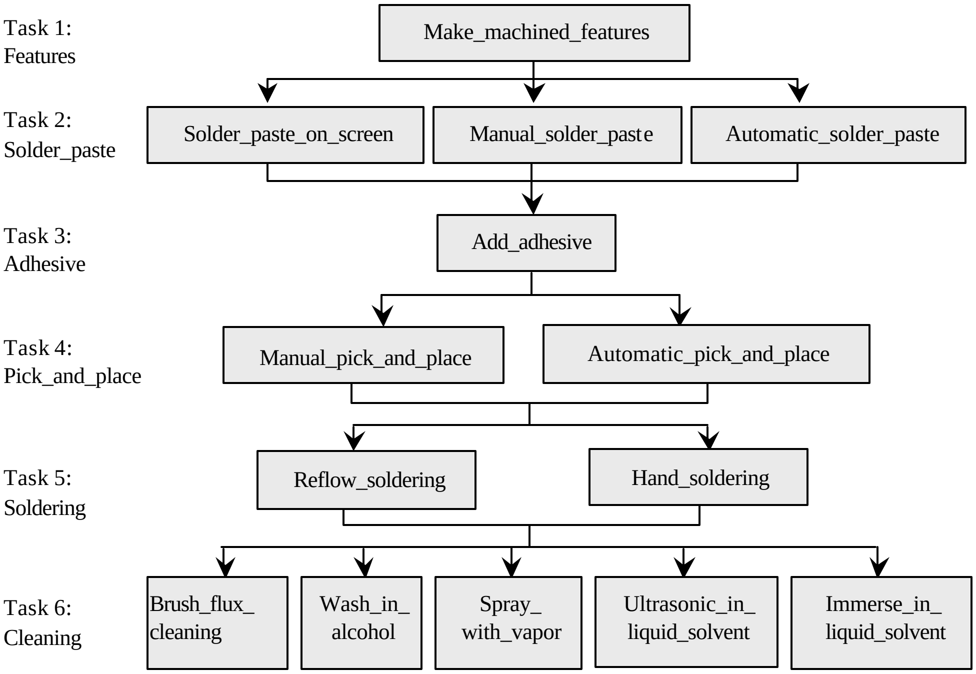


Figure 4. An example of a process template. As described in the text, the template includes formulas for each process's setup time, run time, yield, applicability conditions, and preclusion conditions.

Figure 3 shows the system architecture that we developed to accomplish these tasks. The system includes:

- A *process planning module* (the "Process Planner" in Figure 3). This module uses AI planning techniques to generate "plan fragments" for each part in a design. The process planning module is based on the use of a *process template*. For each part, the process planning module determines (1) alternative processes for each of the processing steps that need to be performed on this part (2) the yields, setup times, and run times of each of those processes, and (3) what processes might damage this part (and are thus precluded from being used on other parts in the design).
- A *process template editor*. This module is used to create and store the domain-specific knowledge base used by the process planning module. The

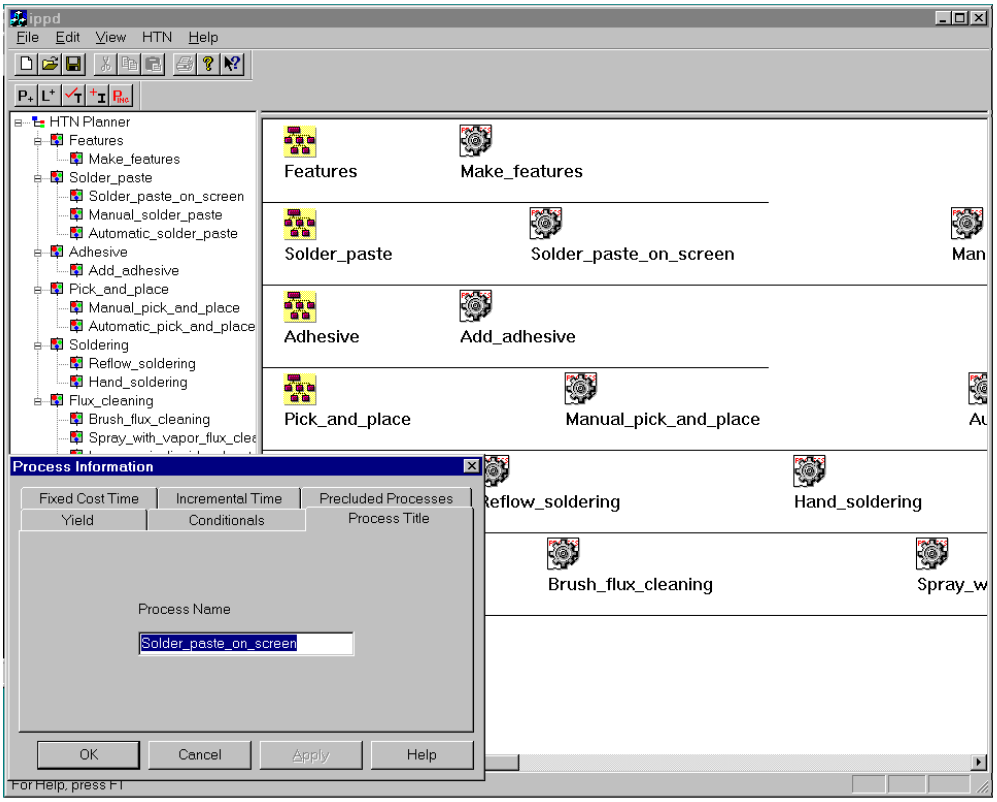
knowledge base consists of a "process template" such as the one shown in Figure 4. This process template contains information about all of the possible processes that might be used on the part. For each process, the process template contains formulas for computing the process's setup time, run time, and yield, applicability conditions (i.e., whether or not the process is applicable to this part), and preclusion conditions (i.e., whether or not this part will preclude the process from being used on other parts in the design).

- A *tradeoff analysis module* (the "Tradeoff Optimizer" in Figure 3). This module uses Integer Programming to generate alternate realizations of the circuit schematic; all these realizations are Pareto optimal with respect to the four criteria mentioned above. Each realization is obtained by making specific choices among the available alternatives for parts and processes. This module includes a GUI (shown in Figure 5) whereby the user can interactively explore Pareto optimal alternatives, and an optimization 'engine' written in C++, which makes calls to the Cplex integer programming solver library. In order to obtain the process requirements and cost estimates associated with the individual parts, the tradeoff optimizer directly interfaces with the process planning module.
- A *supervisory program*. This module, written in Visual Basic, permits the designer to smoothly interact with the heterogeneous collection of modules described above. It also provides an interface between those modules and tools external to the system, such as the *data management* software written by Northrop Grumman personnel using Microsoft Access, and an electronic CAD package such as Hewlett Packard's Advanced Design System (formerly known as EEsof Series IV).
- *Data exchange files*, for exchanging information among the above modules. The data exchange is done in a manner that is transparent to the user.

5. IMPLEMENTATION AND USAGE

5.1 Creating a Knowledge Base

In order to provide an easy way for design and manufacturing engineers to create and maintain process templates, we used Microsoft Foundation Class (MFC) to develop a user-friendly GUI running under Windows NT. This GUI allows the user to update the planner's knowledge base without having to modify any source code (as was necessary in EDAPS's process-planning module). Using the GUI, the user can create each of the levels of the template (corresponding to tasks to be performed), create alternative processes at each level, and create the formulas for computing the process parameters (the applicability conditions, setup time, run time, yield, and preclusion conditions). After the user has created the process template, the template can be installed into the IPPD system, and used on subsequent runs of microwave module designs.



Process Name: Solder_paste_on_screen

Figure 5. The GUI for Process template creation.

As an example, Figure 5 is a screen shot of the GUI for editing the process templates. In this screen shot, the user is editing a process called “solder paste on screen,” which is one of the alternative ways to apply solder paste step at level 2 of the process template in Figure 4. By selecting the appropriate tab of the dialog box shown in Figure 11, the user can enter the process’s formulas for setup time, run time, yield time, applicability conditions, and preclusion conditions.

5.2 Running the IPPD System

The architecture for the IPPD system is shown in Figure 4. Most of the modules run on a PC under Windows NT. However, the process planning module runs as a server on a Sun workstation, and the rest of the system communicates with this module by exchanging files and commands over an ethernet connection.

As input, the IPPD system needs a list of the parts that the designer has chosen to use in a proposed design, as well as miscellaneous information such as the batch size associated with the design and the labor cost. It also needs information about the attributes of each part, including the following:

- the alternatives for each of these parts;
- the cost (in dollars) for each part;
- the defect rate (a number between 0 and 1) for each part;
- supplier information including who the suppliers are and the lead time associated with each one.

Normally, the parts list would come from a commercial CAD tool (such as Hewlett Packard's EEsof Advanced Design System, which we used during our project), and the information about the part attributes would come from a parts database. However, rather than tying the IPPD tool to any particular design tool or database tool, we wanted it to be compatible with a wide variety of design tools and database tools. For this reason, the IPPD tool reads its input from flat files rather than querying the CAD tool and database tool directly, and it is the user's responsibility to export the part information into the flat files from the CAD tool and the database tool. Once the user has done this, the IPPD tool sends this information to its process planning module, so that the process planning module can run the information about each part through the process template to determine which processes may be applicable and what their parameters are.

The process planning module determines which processes are applicable to each part by evaluating each process's applicability conditions against the list of part attributes and their values. For each process that is applicable to a part, the planner evaluates the formula for the process's run time, to compute how much time the process will take on that part. The planner then determines which (if any) processes are precluded by this part (based upon the process's preclusion conditions). It then places these results in a file that is eventually passed to the tradeoff analysis module. The process planning module also creates a file that contains the entire list of possible processes, together with their respective setup times and yields.

The input to the tradeoff optimizer consists of the output of the process planning module, plus the other information mentioned at the beginning of this section. Using this information, the tradeoff optimizer selects combinations of parts and processes in order to generate alternative designs and process plans. Each design and plan generated by the tradeoff optimizer is *Pareto optimal* with respect to the total cost of the parts, the total delivery lead time, the total manufacturing yield, and the total number of suppliers used.

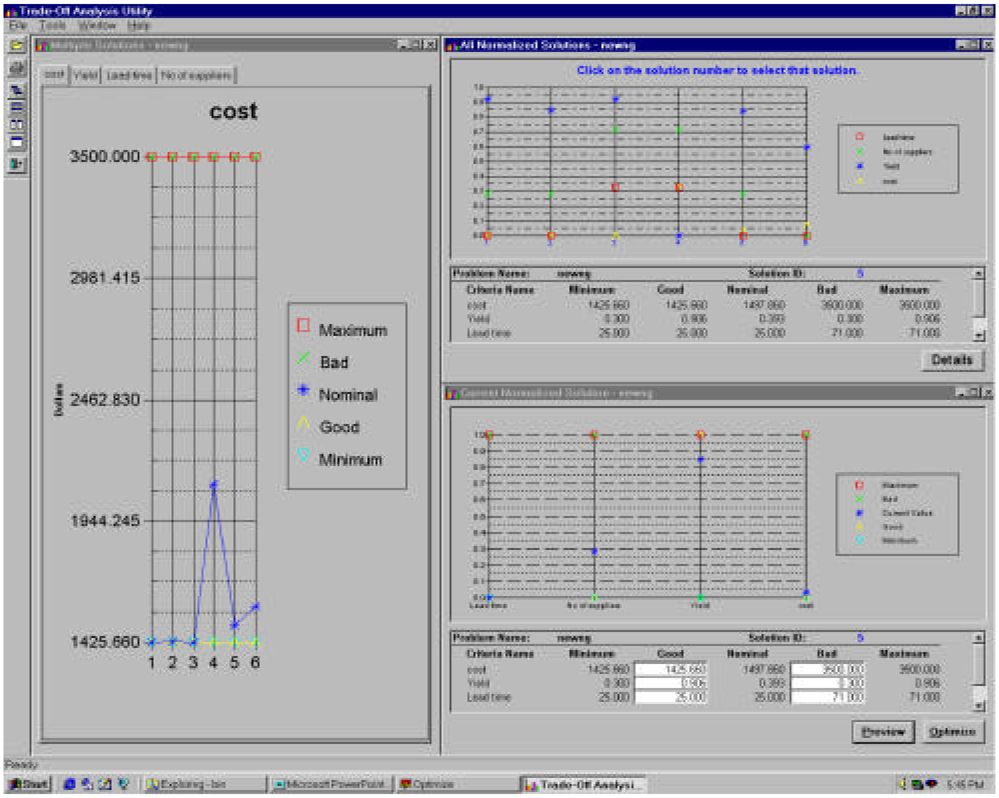
For a complex design problem with many alternative parts and process options, there may potentially be hundreds of Pareto optimal solutions, and the problem is to find one that is satisfactory to the designer. Obviously, it would not be feasible to compute all such solutions and display them—this would overload the designer with too much information. Thus, the IPPD tool provides a GUI to help designers "zero in" on the particular Pareto optimal solutions that are of interest, by enabling them to control the "search direction" via an interactive optimization procedure.

Given the entire universe of possible parts and processes, and their associated attributes, the problem of selecting a subset of the parts (and implicitly, processes) can be formulated as an integer program (IP). The logical structure of the design (such as inclusion and preclusion conditionals for processes, etc.) gives rise to the constraints of the IP. Since we are interested in optimizing the values of four objectives, we must consider a multi-objective IP. While arbitrary integer programs can be difficult to solve (Integer Programming is NP-hard), the underlying structure the IP formulation of the microwave module design problem lends itself to relative fast solution by commercial-off-the-shelf IP solvers. The tradeoff optimizer includes two alternative solution procedures. Trichur and Ball (1998) gives details on the procedures.

Figure 9 gives an example of the tradeoff optimizer in action. When it starts up, the first thing that it does is to find six Pareto optimal solutions/designs (represented by the six columns in the left and top right graphs shown in the figure). The first two designs use the original set of parts specified by the designer, but with alternative process plans that optimize cost and yield, respectively. For the 3rd, 4th, 5th, and 6th alternatives, the system chooses alternatives for the parts specified by the designer, and alternative processes for those parts, in order to find the best possible values for the four objectives being optimized (cost, yield, lead time, and number of suppliers). In the left-hand pane of the figure, the optimizer is currently displaying the cost associated with each of these six solutions, but the user can click on the tabs to display the values for yield, lead time, and number of suppliers. In the right-hand panes, the system has normalized all four of the criteria to the interval [0,1], so that the user can compare them simultaneously.

In the boxes near the lower right-hand corner of the figure, the user can enter lower and upper bounds on the acceptable values of the objectives, and the system will produce additional Pareto optimal solutions (if any) that lie within these bounds. This provides a way for the user to "zero in" on solutions that provide the best balance of lead time, cost, yield, and number of suppliers.

At each major step in the tradeoff-analysis process, a discrete optimization problem is solved. The GUI enables the user to iterate through the process, changing the relative weights that the system will give to the four optimization criteria (lead time, process yield, cost, and number of suppliers). The GUI then displays the results of the optimization process graphically so the user may determine the consequences of increasing or decreasing specific weights. This proces s can be repeated until the user is satisfied he/she has reached an optimal design for the problem under consideration. When the designer is satisfied with one/more solutions, the GUI recovers the relevant data (selected parts, processes, and suppliers, together with the associated values for the four objectives) corresponding to the chosen solution(s) and saves it in a file that is then passed back to the supervisory GUI of the IPPD tool.



Problem Name: newwg Solution ID: 5

Criteria Name	Minimum	Good	Nominal	Bad	Maximum
cost	1425.660	1425.660	1497.860	3600.000	3600.000
Yield	0.300	0.906	0.393	0.300	0.906
Lead time	25.000	25.000	25.000	71.000	71.000

Problem Name: newwg Solution ID: 5

Criteria Name	Minimum	Good	Nominal	Bad	Maximum
cost	1425.660	1425.660	1497.860	3600.000	3600.000
Yield	0.300	0.906	0.393	0.300	0.906
Lead time	25.000	25.000	25.000	71.000	71.000

Figure 3: The IPPD tool's GUI for interactively generating and selecting Pareto optimal combinations of design elements and process-plan elements.

5.3 Performance Tests

To test the tradeoff optimizer, several problem test sets were generated. The number of parts ranged between 25 and 100 with 4 to 6 alternatives per part. Using a Sun Sparc 10 workstation and Cplex 4.0 as the IP solver, the time required to find an individual efficient solution ranged from a few seconds to slightly over 2 minutes. Since, for our application, these problems are of realistic size, we feel this indicates that our IP approach provides a practical problem solving tool.

6. CONCLUSIONS

This project illustrates the benefits of a strongly interdisciplinary approach to knowledge-intensive CAD. Our research team included researchers from computer science, business, electrical engineering, systems engineering, and mechanical engineering; and in addition, we interacted extensively with our industrial partner Northrop Grumman. Our technical approach combined AI planning techniques for generating process-plan elements, IP tradeoff-analysis techniques for selecting Pareto optimal combinations of design elements and plan elements, and a GUI for user control of the system's operation.

As a follow-up to the work described in this paper, we currently are developing IPPD techniques for earlier stages of design (i.e., conceptual design), where the decisions made during the design process can have an even bigger impact (Ball and Fleisher 2000). That work is the subject of an ongoing contract with Northrop Grumman corporation.

This project also shows how research on topics motivated by complex real-world problems can produce benefits to the underlying theory. Our idea of combining AI planning and integer programming was initially motivated by the requirements of the practical problem at hand, but our subsequent exploration of this idea has led to significant theoretical advances. In particular, we have subsequently developed integer programming techniques to solve domain-independent AI planning problems (Vossen *et al.* 1999, Vossen *et al.* 2000), thereby answering one of the challenges proposed in a prominent IJCAI-97 "challenge paper" (Selman *et al.* 1997).

ACKNOWLEDGEMENTS

This work was supported by NSF Grant EEC-9402384, Maryland Industrial Partnerships contract MIPS 1705.17, the Northrop Grumann ESSD design to cost program, ARL contract DAAL01-97-K0135, and DARPA contract F306029910013. Hewlett Packard donated the Advanced Design System 1.0 software used in this project. We would like to thank Bob Hosier and Jim Williams of Northrop Grumman for their valuable inputs to this project.

REFERENCES

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 *Proceedings of the INRIA/IEEE Symposium on Emerging Technologies and Factory Automation*, vol. 3, 241–249.

Ball, M.O. and Fleisher, M., (2000) A Product Design System Employing Optimization-Based Tradeoff Analysis, submitted for publication.

Boothroyd, G. (1992). *Assembly Automation and Product Design*. Marcel Dekker, Inc., New York.

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.

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

Gass, S. I. (1985). *Linear Programming, 5th Edition*. International Thomson Publishing, Inc.

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.

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.

Karne, R.K.; Baras, J.S.; Ball, M.O.; Trichur, V.; Bashyam, S; Kebede, A.; Williams, J.; Huang, F.; Xie, H.; Karir, M. (1998). Integrated Product and Process Design Environment Tool for Manufacturing T/R Modules. *Journal of Intelligent Manufacturing,* Vol. 9, No. 1, 9-15.

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.

Meyer, J., Ball, M., Baras, J., Chowdhury, A., Lin, E., Nau, D., Rajamani, R. and Trichur, V. (1998). Process Planning in Microwave Module Production. In *1998 Artificial Intelligence and Manufacturing: State of the Art and State of Practice*.

Nau, D., Ball, M., Baras, J., Chowdhury, A., Lin, E., Meyer, J., Rajamani, R., Splain, J. and Trichur, V. (2000). Generating and Evaluating Designs and Plans for Microwave Modules. *AI in Engineering Design and Manufacturing*. To appear.

Nau, D., Smith, S. J., and Erol, K. (1998). Control strategies in AI planning: theory versus practice. *IAAI-98.*

Sanii, E. T. and Liau, J. S. (1993). An expert process planning system for electronics PCB assembly. *Comp. in Elec. Eng.* 19:2, 113-127.

Selman, B., Kautz, H., and McA llester, D. (1997). Ten challenges in propositional reasoning and search. In *Proc. Fifteenth International Joint Conf. Artificial Intelligence (IJCAI-97),* Nagoya, Japan.

Shah, J., Mantyla, M., and Nau, D. (1994). *Advances in Feature Based Manufacturing*, Elsevier/North Holland.

Smith, S. J., Hebbar, K., Nau, D., and Minis, I. (1997). Integrating Electrical and Mechanical Design and Process Planning. In *Knowledge Intensive CAD*, Volume 2, 269-288. Chapman and Hall, London.

Splain, J. M. (1998). *A User Interface for Discrete Optimization-Based Tradeoff Analysis*. Master's Thesis, University of Maryland, College Park.

Trichur, V. and Ball, M.O. (1998). A Multiobjective Integer Programming Framework for Product Design. Submitted for publication.

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

Vossen, T., Ball, M., Lotem, A., and Nau, D. (1999). On the Use of Integer Programming Models in AI Planning. In *IJCAI-99*.

Vossen, T., Ball, M., Lotem, A., and Nau, D. (2000). Applying Integer Programming to AI Planning. *Knowledge Engineering Review*. To appear.