Process Planning in Microwave Module Production

Jeff Meyer
Dept. of Computer Science and
Institute for Systems Research
jmeyer@cs.umd.edu

Michael Ball
School of Business and
Institute for Systems Research
mball@rhsmith.umd.edu

John Baras
Dept. of Electrical Engr. and
Institute for Systems Research
baras@isr.umd.edu

Abdur Chowdhury
Institute for Systems Research
abdur@isr.umd.edu

Edward Lin
Institute for Systems Research
lin@isr.umd.edu

Dana Nau
Dept. of Computer Science and
Institute for Systems Research
nau@cs.umd.edu

Ravi Rajamani
Institute for Systems Research
ravis@isr.umd.edu

Vinai Trichur
School of Business and
Institute for Systems Research
vtrichur@isr.umd.edu

University of Maryland, College Park, Maryland 20742

Abstract

This paper describes the development of a process planning module for an integrated product and process design (IPPD) tool which automates the design and manufacture of microwave modules. Specifically, this paper presents a process template that is used to interface with multi-objective optimization techniques to create a planner that can consider alternative designs based on process yields, cost, and manufacturing and purchasing lead times. The integration of the planner and the tradeoff optimization modules provides the designer immediate feedback on cost and productivity through generation and evaluation of alternative designs early in the design stage.

Introduction

In order to achieve the full potential of AI planning techniques for manufacturing problems, it is necessary not only to address issues of performance and scalability (Nau *et al.* 1995, 1998), but also issues such as ease of use and ease of maintenance. In AI research environments, planning mechanisms typically are represented in complex ways are difficult to understand by non-specialists. To be useful for real-world applications, AI planning schemes need to be made easily understandable and modifiable by "computer laymen", so that they will be able to modify the system's knowledge base in response to current needs and changes in the manufacturing environment.

This paper describes our work to accomplish the above goal. Our specific application task is the development of an integrated design and planning system for the development of microwave transmit-receive modules. The system is intended for use in Northrop Grumman Corporation's Baltimore facility.

In the design of a microwave module, designers and manufacturing engineers may require to choose among a large number of components and processes in order to meet system requirements, such as cost, lead times, quality, etc. (Boothroyd 1992, Hebbar *et al.* 1996). Components could potentially be available in many forms; for example, a resistor could be available as both leaded and surface mount, and offered by a number of vendors with differing cost and quality attributes. Each of these different forms of components could require different processes. 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. It is apparent that designers are faced with a large number of choices and could go through a large number of iterations between alternative designs and process plans in order to finalize a product design. Therefore, it is necessary to have a tool that will help designers generate alternative designs, generate process plans, and evaluate the cost and quality of a design. In this paper, we present the development of a process planning module in an IPPD tool.

In creating an AI planning system that is easily updated by a computer layman, we have developed a planning system that could assist in the manufacturing of microwave modules. The goal was to produce a system that could end up on the shop floor and assist engineers in creating microwave module designs that would be efficiently and correctly manufactured.

Related Research

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). Computer-Aided Process Planning (CAPP) systems have been traditionally classified as variant or generative. In the variant approach, new 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 for plan generation. 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. For comprehensive review of numerous CAPP systems in the mechanical domain, see (Shah *et al.* 1994).

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 components on placement machines. Sanii and Liau (1993) and others have used Artificial Intelligence (AI) approaches to develop plans for assembling PCBs. 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. 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. Smith developed the EDAPS Process Planner which incorporates electronic and mechanical manufacturing processes, works concurrently with an electronic CAD, and provides feedback about manufacturability and lead time to the designers based on actual process plans for the manufacture of the device (Smith 1997, Smith *et al.* 1997).

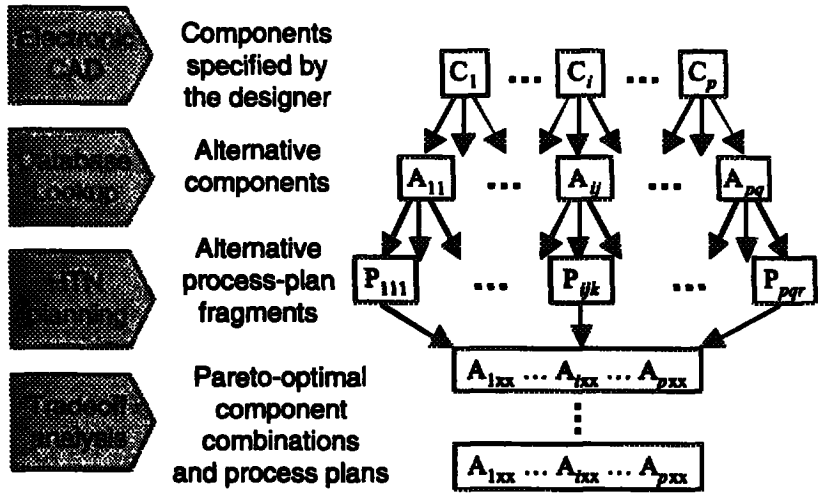

Figure 1: Steps performed by the IPPD tool.

The IPPD Tool

Northrop Grumman has created an initiative called the "Design-to-Cost" project, which was created to provide immediate feedback to a designer about cost and productivity information. The University of Maryland and the Institute for Systems Research (ISR) have positioned themselves in support of this project by creating a system that can generate and evaluate alternate microwave module designs. This new system, called the Integrated Product and Process Design (IPPD) system, is being developed through a contract with Northrop Grumman's ESSD facility in Baltimore, Maryland, in support of their "Design-to-Cost" project.

The current IPPD project grew out of the merger of two previous projects, the EDAPS and the EXTRA projects. EDAPS (Electro-mechanical Design And Planning System), which was developed under NSF funding, was created to take designs and constraints on designs as input and produce plans that supplied information about functionality, manufacturability, cost, lead time, and quality (Smith *et al.* 1997). EXTRA (EXpert T/R module Analyst) was created in a prior contract with Northrop Grumman, and was intended to provide an integration of enterprise-wide product database management with a tradeoff analysis optimization mechanism (Ball *et al.* 1995).

As illustrated in Figure 1, the objective of the current IPPD project is to create a more sophisticated system capable of performing the following tasks:

- find alternative components that are suitable for substitution into the design in place of the components specified by the designer;
- generate alternatives for those portions of the process plan that would be needed by each alternative component;
- find pareto-optimal designs, by taking combinations of the alternative components and alternative plan fragments that produce pareto-optimal values for the following criteria: cost, lead time, yield, and number of suppliers;
- provide a GUI to aid the user in selecting a design and a process plan from among the pareto-optimal alternatives.

This system builds upon the ideas previously used in the EDAPS project, by breaking the functionality of EDAPS's process planner into two distinct units: a simplified planner and a tradeoff analysis module. The purpose of the simplified planner is to determine valid process options for a given set of components, while the tradeoff analysis module would then be able to determine which of these process options are pareto-optimal.

The tradeoff analysis module expects to receive as input a list of parts, a list of alternates for those parts, a list of all the processes which can be used to place these parts on the circuit board, and a list of processes precluded by these parts. For each of these valid processes, the tradeoff analysis module expects to receive the fixed cost setup time, the incremental run time, and the yield associated with the process. The fixed cost setup time is the one time cost of starting up the process, while the incremental run time is the cost that occurs for each component that is put on the board because of this process. The yield describes the basic effectiveness of that process to accurately

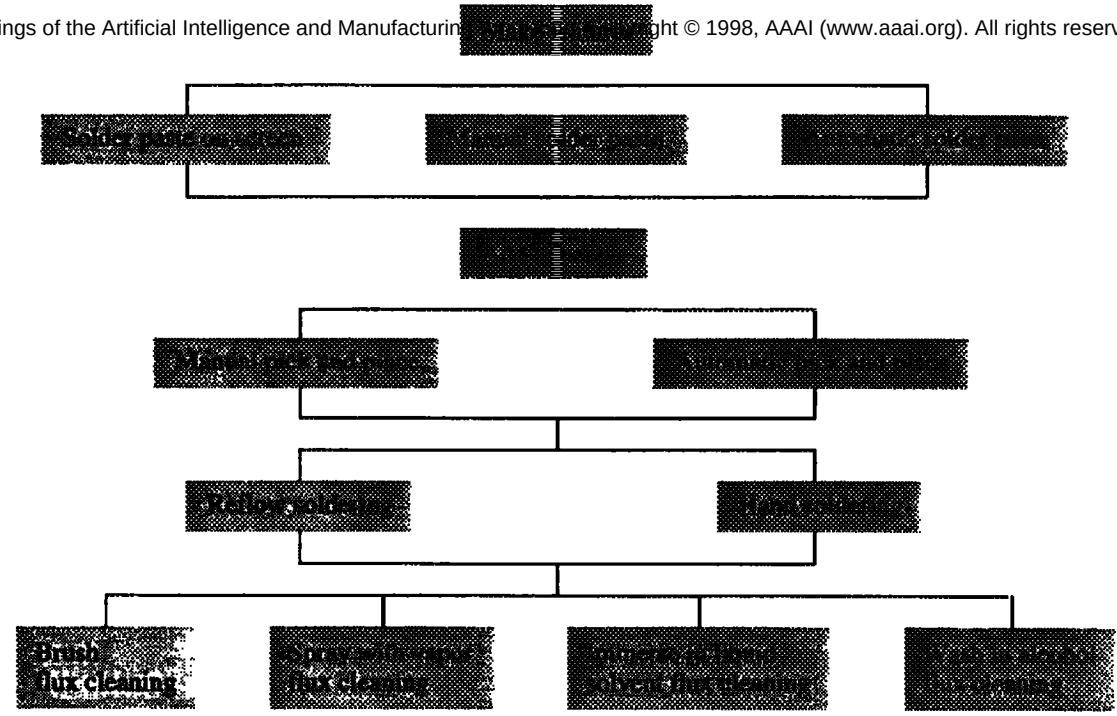

Figure 2: Sample process planning template.

perform or complete its job on a component on the board. The tradeoff analysis module will take this information and perform a global optimization on the data and decide which of the processes and components are most cost effective.

Process Planning

The primary purpose of the process planner is to feed information into the tradeoff analysis module. For this purpose, we initially considered modifying the process planning module we had developed earlier in the EDAPS project. However, we decided that this would not meet our objectives, because the EDAPS process planning module had all of its process information hard-coded into the program. (This process information included both the breakdown of the process plan hierarchy and all of the fixed and incremental costs of each process.) For the IPPD project, we wanted to remove the necessity of having a software engineer maintain the system after delivery, so the information tied into this planner (which was written in C++) needed to be placed somewhere that would be easy for a non-computer programmer to update. Below we describe the design decisions that we made in order to meet these objectives.

In order to provide an easy way for the various information pertinent to a manufacturing process to be updated by non-computer science personnel, we used Microsoft Foundation Class (MFC) to develop a GUI which anyone familiar with using Windows 95 should be able to easily use. Using this GUI, the user can create a process template that details the various processes that a component would go through to be placed on a circuit board. This GUI will allow the user to be able to effectively update the flow of the planner without having to modify any source code, which most likely would be a discipline that they are not very familiar with.

A graphical example of the type of templates that can be generated is shown in Figure 2. This template describes the steps that a component must go through before it can be placed on the board. (It is important to note that this template does not reflect a complete plan. It is merely provided here to give an example of what a process template conceptually looks like).

Each of the boxes in Figure 2 represents an option (or process) that can be used on a component. Each of the processes at the same vertical level of this figure can be considered to be alternates of the same step. More than one process at a given step may be valid for a component,

Process name:	Solder_paste_on_screen
Setup time:	60.0
Run time:	num_pins * 1.5
Yield:	0.99
Conditionals:	num_pins > 10; length * width > 20;
Precluded conditionals:	num_pins == 0;

Figure 3: Process specifics.

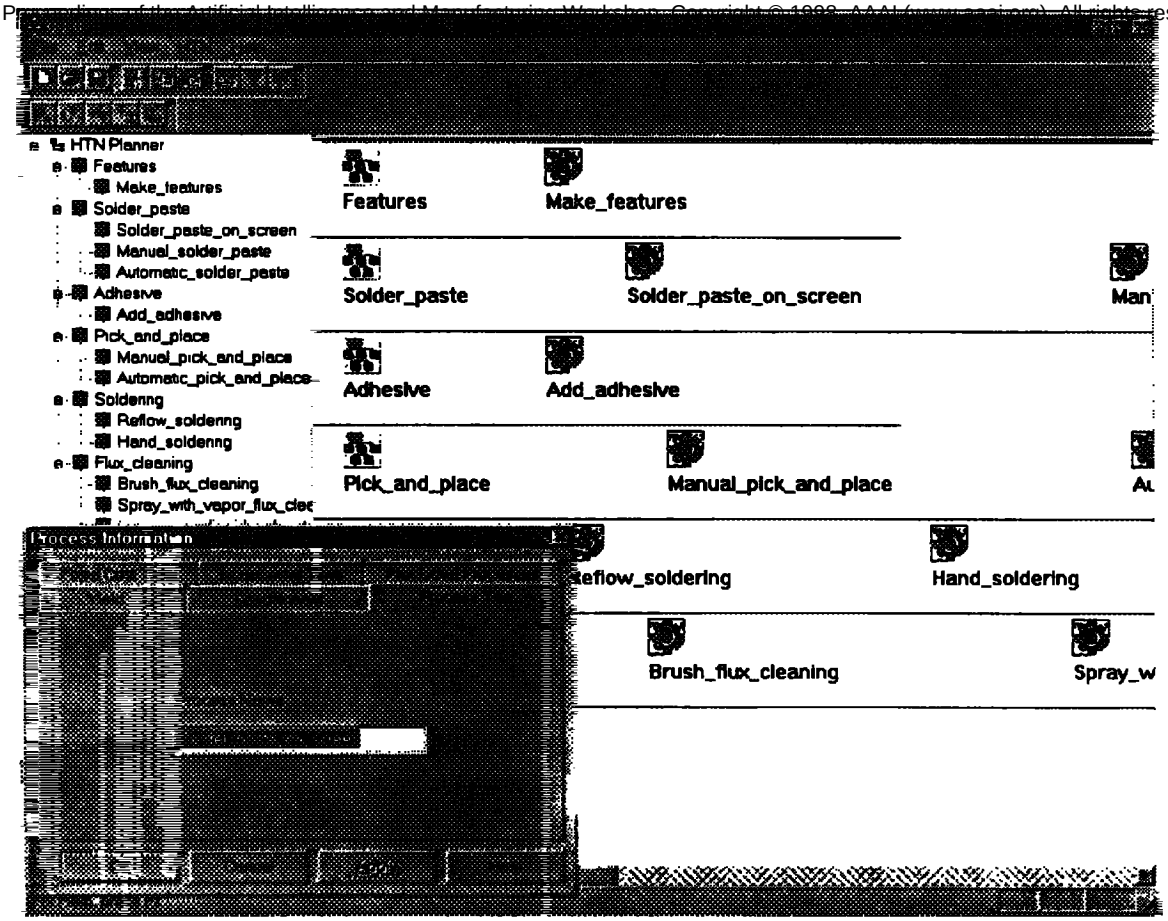

Figure 4: Process planning GUI.

for instance, "Solder paste on screen", "Manual solder paste", and "Automatic Solder Paste" are all methods for getting solder paste on the component. Each of these options (or processes) will have conditions associated with it that must be true for this option to be able to be run (These options will be dependent on various characteristics of the component such as number of pins, size of component, etc.).

In addition to these conditions, each option will have a fixed time, an incremental time, a yield, and a list of precluded process conditionals associated with that process. The fixed time represents the one time fixed cost associated with running that process, while the incremental time represents the incremental time that is associated with each iteration of that process. The yield represents the effectiveness of that process. Finally, the precluded process conditionals are a list of conditionals that describe when this process could not be run on a board. (See Figure 3) Typically, including a certain part of a circuit board might preclude certain processes from being run on that board, since they might damage the component. For instance, a non-immersible part would not be able to run the process "immerse in liquid solvent flux cleaning", since that process would surely destroy the component.

After the user has created the template on the PC, they can have the template installed into the IPPD system so it will be used on subsequent runs of microwave module designs. The actual template planner code resides on a UNIX platform; the PC will create a connection to a server which is set up on a Sun Sparc, and it will send the template so that it can be stored for use on the Sun. Once the template is installed, the user can input a list of components for a board (the board will be designed by the user using commercial software like EESOF), which will be used on this board, into the process planner server. A front-end interface will take this list of parts, access a parts database, and extract all of the attributes known about this part. This list of information will then be sent to the template planner server (on the Sun) so it can use this list of parts and their attributes to instantiate the process template.

The planner will determine which of the options can be run for this component by evaluating the conditionals for that process against the list of component attributes and their values provided. For each of the processes selected, the planner will instantiate the run times to provide numeric results based strictly on that component. The planner will then determine which (if any) processes are precluded from running because of this component based

```
-Solder_paste 3
Solder_paste_on_screen2 1
60.0
num_pins * 1.5
0.99
num_pins > 10
length * width > 20
num_pins == 0
Manual_solder_paste 0  0
30.0
0.50
0.990
Automatic_solder_paste 0  0
60.0
0.083
0.999
```

Figure 5: Partial listing for process plan

upon the precluded process conditionals. It will place these results in a file that will eventually be passed to the tradeoff analysis module.

Next, it will create a file that contains the list of processes and their respective setup times and yields. The setup times and yields are not component specific, so they will be placed in a separate file to reduce space overhead as well as speed up the processing for the tradeoff analysis module. Both of these files will be sent back to the PC (where the original run time calls came through the front-end software). The front-end interface will forward those files on to the tradeoff analysis module so it can determine which processes would be most cost effective for producing the board design that is desired. Figure 3 shows an example of a specific process.

The "Solder_paste_on_screen" process can only be chosen by the process planner if the conditionals are satisfied by the component that is currently being instantiated against the template. This component would have to have more than 10 pins on it, and it would have to have a surface area of greater than 20 square centimeters. If the component satisfied these constraints, it would then have it's run time determined by multiplying the number of pins on the component by the constant 1.5. This number, along with the process that was chosen for this component, would be output to the resulting component file for input into the tradeoff analysis module. The yield of 99% and the fixed setup time of 60 seconds would be written to a separate list of processes file. Finally, if the component did not satisfy the conditionals on this process, the planner will check against the precluded process conditionals to see if this process would destroy the component if it were used on a board that contained this component. In this contrived example, any component without pins would be destroyed by the process "Solder_paste_on_screen", so "Solder_paste_on_screen" would be included in the precluded process list for any components with no pins.

```
2
-P2
immersible = 1
length = 10
width = 15
num_pins = 8
preformed = 0
adhered = 1
-P4
immersible = 0
length = 18
width = 24
num_pins = 2
preformed = 1
adhered = 0
```

Figure 6: Component file for first example.

Sample Run

This section details a sample run of two sets of parts on a template that is modeled after the one shown in Figure 2. This section will show the information that is stored at the server, the information that is supplied as input to the server, and the corresponding output from the server that will then be passed on to the trade/off analysis module. Figure 4 shows a snapshot of the creation of a template using the PC's GUI. After the template has been created, the user needs to install the template on the Sun server using the control features on the GUI. The template will be validated and then stored in a compact form on the server so that any list of parts can then be run against that template to generate a list of alternative processes that can place those parts on a microwave module board. The form that the server stores the information in is partially listed in Figure 5. For brevity, only a portion of the entire plan is displayed: the portion involving the the "Solder_paste" step in Figure 2. The "Solder_paste" step contains three different options: Solder_paste_on_screen, Manual_solder_paste, and Automatic_solder_paste. The last two processes do not have any conditionals associated with them, while the "Solder_paste_on_screen" has two conditionals that must be satisfied in order for this process to be valid. This is important to note for the two example runs that will be shown.

In the first example, the list of components found in Figure 6 is provided to the server. The instantiation of this list of components created the two output files shown in Figure 7. One of the output files contains information about all of the processes listed in the template, along with their setup times and yields. The other output file contains information about all the alternatives that are valid for the given components, and it provides instantiated run time values (i.e. all the run time values in the file are numeric, since the variables in any formulas were replaced with their actual numeric values).

It is important to note that for the part P2, the two valid alternatives for solder pasting are "Manual_solder_paste" and "Automatic_solder_paste". The third possible alternative "Solder_paste_on_screen" (shown in Figure 2) is not valid for P2 according to the conditionals, so it was not listed.

The second example run modifies the input component file shown in figure 6 by increasing the num_pins field of P2 from 8 to 12 and provides it to the server. The results of the second run are shown in figure 8. The process "Solder_paste_on_screen" is now an option for "Solder_paste" on part P2, since the number of pins in part P2 now exceeds 10. (Note: the file with the process names and their setup times and yields has not changed at all, and it should not change unless the actual process plan is modified. The setup times and yields are assumed to be constants for each process.) These two examples show how different alternatives can be selected because of different attributes of the parts.

Conclusions

The template planner created was able to provide a smart and efficient way to aid engineers in the development of microwave modules. The information contained in the template will compactly display and store the steps necessary for placing each component in a microwave module design on a circuit board. The parameters found in the template are easily modifiable without the necessity of knowing the kind of confusing and often unintuitive syntax normally found in most database systems. The template was created with ease of update and minimal amount of training for the user in mind, and it succeeded by creating a very simple and intuitive interface.

This project proved the necessity for inter-disciplinary research. This project shows how a useable system can be creating by merging the information present in the computer science, business, systems engineering, and mechanical engineering disciplines.

```
P2 6 0
- 1
Make_features 0.25
- 2
Manual_solder_paste 0.5
Automatic_solder_paste 0.083
- 1
No_Adhesive 0
- 2
Manual 0.8
Automatic 0.01
- 2
Reflow_soldering 0.1
Hand_soldering 0
- 3
Brush_flux_cleaning 0
Immersion_in_liquid_solvent_flux
Wash_in_alcohol_flux_cleaning 0
P4 6 2
- 1
No_Features 0
- 2
Manual_solder_paste 0.5
Automatic_solder_paste 0.083
- 1
Add_adhesive 0
- 2
Manual 0.2
Automatic 0.01
- 2
Reflow_soldering 0.1
Hand_soldering 0
- 1
Brush_flux_cleaning 0
Immersion_in_liquid_solvent_flu
Wash_in_alcohol_flux_cleaning
```

```
Make_features 30 0.985
No_Features 0 1
Solder_paste_on_screen 60 0.99
Manual_solder_paste 30 0.99
Automatic_solder_paste 60 0.999
No_Solder_paste 0 1
Add_adhesive 30.5 0.99
No_Adhesive 0 1
Manual 0 0.995
Automatic 60 0.999
No_Pick_and_place 0 1
Reflow_soldering 5 0.99
Hand_soldering 15 1
No_Soldering 0 1
Brush_flux_cleaning 10 0.995
Immersion_in_liquid_solvent_flux 5 0.993
Spray_with_vapor_flux_cleaning 10 0.995
Wash_in_alcohol_flux_cleaning 20 0.999
No_Flux_cleaning 0 1
```

Figure 7: Output of first example run.

```
P2 6 0
- 1
Make_features 0.25
- 3
Solder_paste_on_screen 18
Manual_solder_paste 0.5
Automatic_solder_paste 0.083
- 1
No_Adhesive 0
- 2
Manual 1.2
Automatic 0.01
- 2
Reflow_soldering 0.1
Hand_soldering 0
- 3
Brush_flux_cleaning 0
Immersion_in_liquid_solvent_flux
Wash_in_alcohol_flux_cleaning 0
P4 6 0
- 1
No_Features 0
- 2
Manual_solder_paste 0.5
Automatic_solder_paste 0.083
- 1
Add_adhesive 0
- 2
Manual 0.2
Automatic 0.01
- 2
Reflow_soldering 0.1
Hand_soldering 0
- 3
Brush_flux_cleaning 0
Immersion_in_liquid_solvent_flux
Wash_in_alcohol_flux_cleaning 0
```

```
Make_features 30 0.985
No_Features 0 1
Solder_paste_on_screen 60 0.99
Manual_solder_paste 30 0.99
Automatic_solder_paste 60 0.999
No_Solder_paste 0 1
Add_adhesive 30.5 0.99
No_Adhesive 0 1
Manual 0 0.995
Automatic 60 0.999
No_Pick_and_place 0 1
Reflow_soldering 5 0.99
Hand_soldering 15 1
No_Soldering 0 1
Brush_flux_cleaning 10 0.995
Immersion_in_liquid_solvent_flux 5 0.993
Spray_with_vapor_flux_cleaning 10 0.995
Wash_in_alcohol_flux_cleaning 20 0.999
No_Flux_cleaning 0 1
```

Figure 8: Output of second example run.

One future direction for our work is to integrate the plan generation and the tradeoff analysis into one step instead of keeping them separated into two distinct steps. The reason for this is that it is not really possible to do the plan generation as well as we would like without taking into account various tradeoff considerations inherent in cost and yield information. Also, there are quite possibly interactions between the processes that have been ignored for simplicity's sake in this project, and an integration of these two modules would allow future researchers and designers to produce a complete system.

As a first step toward integrating plan generation and tradeoff analysis, we have begun work on integrating AI planning techniques into an Integer Programming formulation. Our preliminary experimental results (Vossen *et al.* 1998) are quite promising.

Acknowledgements:

This work is supported by the NSF Grant EEC-9402384, Maryland Industrial Partnerships contract MIPS 1705.17, and the Northrop Grumann ESSD design to cost program. 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.

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

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

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.

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.

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.

Nau, D.; Gupta, S.K.; and Regli, W.C. 1995 AI planning versus manufacturing-operation planning: a case study. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1670-1676. Morgan Kaufmann, San Mateo, California.

Nau, D., Smith, S. J., and Erol, K. 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.

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

Smith, S. J. 1997. Task-Network Planning using Total-Order Forward Search, and Applications to Bridge and to Microwave Module Manufacture. Ph.D. Dissertation, University of Maryland at College Park.

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.

Vossen, T., Ball, M., Lotem, A., and Nau, D. 1998. Integer Programming Models in AI Planning: Preliminary Experimental Results. *AIPS'98 workshop on Planning as Combinatorial Search*, 111–113.