JB 97-13

Entitled:

"Integrated Product and Process Design Environment Tool for Manufacturing T/R Modules",

Authors:

R.K. Karne, J.S. Baras, M.O. Ball, S. Bashyam,
A. Kebede, J. Williams, V.S. Trichur, M. Karir,
H.T. Lai and S.V. Dandekar),

Conference:

ETFA'97 :
6th IEEE International Conference
on Emerging Technologies and Factory Automation
September 9-12, 1997
UCLA, Los Angeles, CA

Integrated Product and Process Design Environment Tool for Manufacturing T/R Modules [1]

Ramesh K. Karne, John S. Baras, Michael O. Ball, Sridhar Bashyam,

Abraham Kebede, Jim Williams, Vinai S. Trichur,

Manish Karir, Hsing-Tsu Lai and Swati V. Dandekar

e-mail: karne@midget.towson.edu,

baras, trichur, karir@isr.umd.edu, ball@bgmt.mail.umd.edu,

williams@amil.md.essd.northgrum.com,

lhsing1, sdande1@tiger.towson.edu

Contact: Ramesh K. Karne

karne @midget.towson.edu

Towson State University

Department of Computer Science, Stephen Hall

Towson, MD 21204 USA

Phone: (410) 830-3955

Fax: (410) 830-3868

---

[1] The research was supported by Maryland Industrial Partnership (MIPS) and Northtup Grumman, Inc.

Integrated Product and Process Design Environment Tool for Manufacturing T/R Modules

Ramesh K. Karne is a professor of Computer Science at Towson State University,

John S. Baras is the Director of Center for Satellite and Hybrid Communication Networks, Institute for Systems Research at Univetsity of Maryland,

Michael O. Ball is a professor of information management at University of Maryland,

Sridhar Bashyam is with University of Maryland,

Abraham Kebede and Jim Williams are with Northrup Grumman, Inc.,

Vinai S. Trichur is a doctoral student at University of Maryland,

Manish Karir are graduate students at University of Maryland,

Hsing-Tsu Lai and Swati Dandekar are graduate students at Towson State University

Abstract

We present a decision making assistant tool for integrated product and process design environment for manufacturing applications. Specifically, we target microwave modules which use electro-mechanical components and require optimal solutions to reduce cost, improve quality, and gain leverage in time to market the product. This tool will assist the product and process designer to improve their productivity and also enable to cooperate and coordinate their designs through a common design interface. We consider a multiobjective optimization model that determines components and processes for a given conceptual designs for microwave modules. This model outputs a set of solutions that are Pareto optimal with respect to cost, quality, and other metrics. In addition, we identify system integration issues for manufacturing applications, and propose an architecture which will serve as a building block to our continuing research in virtual manufacturing applications.

1.0. Background

Currently, the functional perspective of a product designer and the manufacturing perspective of a process designer are generally considered as separate design phases, as shown in Figure 1. Communication between these two areas are often limited by differing design philosophies, differing areas of expertise, physical location, etc. This pipeline approach results in an unnecessary long design cycle which consequently wastes time and money. This expense of resources results in a product that is less competitive in a global market. This problem is being addressed in many research institutions, resulting in a flood of publications in concurrent engineering[2,6,9], integrated product and process design methodologies[3,7,11], and computer aided simulation tools[15].

Typically, product designs are conducted in a CAD/CAM environment where the data generated is either stored as information in a CAD drawing or in a large relational database together with a variety of other company data. The process data and models are often not captured in electronic form, resulting in information that exists in the expert's mind. For example, a soldering process is very difficult to model due to its intertwined dependencies with a particular part being operated on as well as the operator's skills, state of mind, affiliation with labor unions, dexterity, and so on. The runtime (cost) and quality of a soldering process will depend upon all of the above factors. The process models must consider all of these issues to develop accurate models. Current process models use historical data and measured data in the actual manufacturing line. This data measured in the manufacturing line is often prone to errors due to inaccurate measurements, negligence of the operator who is taking the measurements, and inadequate tools to take appropriate measurements. In addition, the measuring process itself slows down the manufacturing process and thus results in an artificially longer runtime. In an IPPD system environment, the product data that does exist in legacy databases must be integrated with the more accurate process data and models which must be acquired from process design experts.

Numerous researchers have developed decision support systems for evaluating manufacturability of T/R modules[16]. Our previous research work with T/R modules[8] addressed building cost, quality, and manufacturing rating modules and further built an expert system to evaluate different design scenarios using these models. Knowledge-based systems[17] have been attempted to address assembly of printed circuit boards. The literature cited in these papers indicate that rule-based expert systems have been the predominant model for manufacturing assessment of printed board assemblies. While these tools provide significant support to product designer, they do not provide an integrated platform for product and process design and do not generate and evaluate the number of design options available. For example, a given product design has numerous options including: alternate parts and material from the same or different vendors, alternate processes for a given process, alternate implementations for a given function, and alternate assemblies due to large scale integration. For instance, a transistor of a given specification could be available as both leaded and surface mount types, and offered by a number of vendors with differing cost and quality ratings. These differences could, in turn, require different processes for assembly and electrical connection. Also, in markets where lot sizes may be relatively small, we need to consider both manual as well as automated options to carry out processes such as assembly and soldering (in contrast with high volume commercial applications where, except for odd-shaped components, the operations are almost entirely automated).

All of these factors indicate that the designers are commonly faced with a large number of options in terms of component-process configurations. Furthermore, cost and quality trade-offs are considerable between these varying choices. Consequently 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 performance metrics. We need multiobjective optimization models[1] that determine components and processes for given conceptual designs, as well as complete designs for microwave modules. These optimization techniques can in turn be used for other related application domains as they are developed as generic models that target virtual manufacturing.

2.0. Introduction

Today's manufacturing environment is becoming increasingly large, complex, mission critical and heterogeneous in several dimensions. For example, the underlying product design involves consideration of many faucets including cost, quality, process plans, electrical design, mechanical design, physical design, market analysis, testing, design cycle, optimization of design with respect to cost and quality, system platform, software tools, and so on. The design issues related to manufacturing a product span over multiple disciplines and fields of engineering and computer science. The future of manufacturing which plays a vital role in a country's economy will depend upon how well we understand manufacturing design issues and how efficiently we integrate this heterogeneous complex systems which can work together in harmony and result in a productive virtual manufacturing environment.

The Manufacturing design environment is usually treated as two isolated design environments; the product design and the process design (Figure 1). However, a given process may force to change its design if it is not manufacturable. We propose data models which closely integrate product and process data and help the designer to identify the design options before manufacturing and evaluate these options with respect to manufacturability of the product. In this paper, we will focus our attention to build a T/R Module tool for integrated product and process design (IPPD) environment, and study the design issues that are related to architecting this product. In addition, our research efforts will be confined to electro-mechanical components of the microwave module designs, but not dealing with geometric modeling, process planning, physical design, and CAD/CAM environment. We assume that the data required for the T/R Module is either already being generated by the CAD/CAM tool and exists in a database, or the designer can enter a conceptual design to the T/R Module tool and expects the tool to generate alternate options and present trade-off analysis with respect to all possible designs.

Attempts to determine optimal designs (rather than assessing a given design) based on costing mechanisms have been rather limited. Dynamic programming [2] approach to optimize the assembly processes does not appear to be practical for situations having a large search space of design alternatives. The other optimization approaches[4,5], knowledgebase techniques[10] appear to be more along the lines of work being reported here.

Our optimization techniques used in the IPPD T/R Module tool have the following capabilities:

- It considers a set of alternative conceptual/detailed designs for a given application and for each design, the complete set of alternate options in terms of functions, parts, and processes.
- Explicit expressions for cost and quality are developed and form the basis for exploring the search space of design options. The analysis takes into account the various subassemblies that comprise the final product.
- The problem is formulated as a multiobjective integer program and a solution procedure is proposed to efficiently output a set of Pareto optimal solutions.

Some of the design issues and architectural problems addressed in this paper are listed as follows:

- Integrate product and process designs
- Map legacy data from relational databases to object-oriented databases
- Apply multiobjective optimization techniques to manufacturing domain
- Develop client-server software for IPPD environment

- Develop object-oriented modules and libraries extendible to virtual manufacturing applications
- Integrate different data models (object-oriented, relational, and OR models) and address system integration issues.

The rest of this paper is organized as follows: the system architecture is defined in section 3.0. The data models related to design and process environment are illustrated in section 4.0. The optimization problem definition is illustrated in section 5.0. The section 6.0 narrates our implementation details and software platform. Finally, the section 7.0 describes concluding remarks and our future research work in this area.

3.0 System Architecture

We integrate the product and process phases of the design into a single system environment and apply optimization techniques to achieve optimal solutions for an intended product. In effect, we systematically explore all possible alternate options that are available at each level of the product including: parts and material, processes, functions, and assemblies. Unfortunately, the data that is needed to achieve these objectives is scattered among a variety of sources. Some of these sources are database management systems, some are in CAD databases, and some are still in need of transferring into a usable form. The T/R Module system architecture is shown in Figure 2, and brief details of this architecture follows.

The software architecture for the T/R Module tool is based on a client-server architecture to accommodate the distributed nature of the design environment. We have architected the system based on a workstation (SUN/UNIX) server, and a PC client (IBM Compatible/Windows NT) environment. The object-oriented database (ObjectStore) is used as a central repository for all the design information.

3.1 Relational Databases

The manufacturing data available at our customer site is spread across many legacy databases including DB2, Tandem, and IMS. For example, the parts data may be in DB2, and purchasing information may be in IMS. In addition to the data that is spread across heterogeneous databases, designers have some local data at their workstations which is in Paradox database. The Paradox DBMS is the designers choice for local data storage and interface, and it is necessary to interface DMA with the Paradox and as well as other relational DBMS systems. Some data is not available in any of the existing databases, and it must be entered manually through the paradox GUI. For example, the process information is currently stored in ***routers*** which are text documents, and has to be manually entered by the designer.

Considering the variety of data sources, including heterogeneous databases, paradox database, text data, design data, and manually entered information, we have architected the problem with two-tier solution. All the design data, and manually entered information by the designer will be entered through the Paradox database and the Paradox Database Frontend (PDFE) provides such an interface. The PDFE is a Paradox application program which is written in ObjectPal and has custom designed forms and primitive graphical user interface (GUI) available to the designer. Numerous other relational DBMS (RDBMSs) can be interfaced with the system through a bridge (BR) which is an ODBC application program. The ODBC application maps queries to appropriate RDBMS through an ODBC driver. Thus, there is a need for ODBC driver, one for each RDBMS as required by the system. The bridge program has to formulate queries and interface with PDFE and also other relational database systems.

3.2 Graphical User Interface (GUI)

The Graphical User Interface (GUI) is the most critical part of the system. We have developed ergonomically designed user interfaces to help the user to navigate through assemblies, subassemblies, bill of material (BOM), functional block elements (FBEs), assembly hierarchies, and list of processes (LOPs). We have used ***"treebrowser"*** type of approaches to display assembly hierarchies and navigate through the elements. This GUI interacts with the object manager and issues dynamic queries to obtain data from the object-oriented database. The GUI allows the user to interact with the optimizer and perform sensitivity analysis on the design options.

3.3 Bridge

There is a need for a bridge[13,19] which will acts as a database gateway between relational and object-oriented databases. The objects in the object-oriented data models are mapped to relational table forms, and vice-versa through this bridge. E/R models are used to map the manufacturing database either to relational or object-oriented models. Optimal ODBC queries are developed to build this bridge interface.

3.4 Optimization

All the design data, including alternate scenarios, assemblies, function blocks, parts, and processes, and associated hierarchical tree structures, are stored in the ObjectStore database. We use a multiobjective binary integer programming model to explore this search space and identify designs that are Pareto optimal with respect to a cost metric and a quality metric. The tool employed for optimization is CPLEX, a state of the art linear and integer programming package. CPLEX provides a callable library of C routines that can be invoked a C/C++ program. We use this library to develop our optimization routine. We opted for a tight integration of CPLEX with the OODBMS, this eliminates the need for flat files, increases efficiency and results in transparent and easy-to-use system.

The optimization routine queries the design tree stored in the OODBMS, and constructs the associated integer program (populates the objective function and constraint coefficient matrices used by CPLEX). Next, the CPLEX callable library routines are used to generate the Pareto optimal solutions. These solutions are then transported back to the OODBMS and stored in a form accessible to the "treebrowser" GUI. In addition, a cost-quality tradeoff curve is generated, which displays the cost and quality values associated with each Pareto optimal solution. The user can click on any point on this curve and examine the corresponding design using the "treebrowser" GUI.

3.5 Object Manager

The Object Manager (OM) interacts with the object-oriented database and provides a pivotal role in the T/R MODULE tool. The main functions of the OM are listed as follows:

- interface with the user (GUI) to obtain control information to the optimizer, also provide query interface to the user to display information from the ObjectStore
- interface to the bridge to obtain appropriate data and store it in the ObjectStore
- interface to the optimizer to provide initial design and scenario data to the model, and also obtain resulting data to be stored in the optimizer.

The OM provides the above functions and also consists of implementation code for the object-oriented data models. The OM is basically an application program written in C++ and in addition consists of programs to perform OODBMS transactions. As the data required for optimizer, GUI, and bridge are stored in OODBMS, dynamic queries are written to provide and accept data from these modules. The object manager interfaces with GUI and OODBMS through a standard GUI API and OODBAPI respectively. We have architected these two APIs such that in case, the GUI and OODBMS change, there is a minimal impact on the OM code changes.

3.6 Object-oriented Database Management System

The Object-oriented database management system is a central repository for data related to designs and user data. The OODBMS in our architecture interfaces with the Object Manager as shown in Figure 2. We have chosen the ObjectStore as our OODBMS system. The ObjectStore provides a single C++ programming interface for data manipulation (queries) as well as data definition (standard data structures).

3.7 Cost and Quality Models

We have developed a cost model [8] based on labor cost, material cost at an assembly level, where each assembly is a manufacturable unit. We have developed a quality model [8] based on process yield, subassemblies, process defect rate, and material defect rate at an assembly level, where each assembly is a manufacturable unit. The cost and quality equations are used by the optimizer and refer to section 5.0 for more details.

4.0 Data Models

Data Models are abstract building blocks to understand the domain of applications. Data models capture data organization and behavior of an application which will enable us to build data structures for building application programs. In this paper, we encountered six data models; functional data model (FDM), assembly data model (ADM), AND-OR models for optimization (AOM), Entity Relationships Models (ERM), Relational data model (RDM), and Object-oriented data model (ODM). These six models serve their own purpose, and describe the manufacturing system environment and its implementation and capture the abstractions needed to understand the IPPD system.

4.1 Functional Data Model

Any electromechanical or electronics design stems from a functional model which satisfies the requirements of a product. The functional model is typically an hierarchical top-down structure which illustrates the decomposition of higher level functions to lower level functions and also captures the functionality of the product. A designer at this point conceptually designs functional blocks to perform a set of functions for a given design. It is possible that there can be many alternate designs possible to satisfy the same requirements of a given product. We propose a functional data model (FDM) which embraces this methodology.

A generic FDM is shown in Figure 3, and can be further described as follows. Each block in the FDM is a unique functional block element (FBE) which realizes a particular function of a design. For example, a power amplifier function in a microwave module is realized by the power amplifier FBE. Each FBE is associated with a functional bill of material (FBOM) which is a list of parts and material required for implementing this function. After the function is designed then it goes to manufacturing, and during this stage it is routed through many sequence of processes before it becomes a product. The designer can associate each FBE with a set of planned processes that are required for manufacturing. This set of processes are called functional list of processes (FLOP). For example, an amplifier function may require parts such as transistors, capacitors, resistors, and materials such as solder, cleaning solvent, and q-tips. The FBOM for an FBE thus consists of a list of parts and material and also the number of parts and material for each type. The Figure 4 shows the FDM for the power module example. The FBOM and FLOP are not shown in details to keep clarity of the figure.

4.2 Assembly Data Model

The designer starts a conceptual design through a FDM. In order to produce an end product, the FDM has to be mapped to an assembly data model (ADM), where each assembly can be considered as a manufacturable unit. The assembly captures design aspects of a product and is also subjected to the manufacturing operations. Each assembly may consist of one or more FBEs thus constituting a manufacturable unit. The mapping of FDM to ADM data models depend upon the physical characteristics of an assembly including: number of parts that can fit in a given assembly, the size of the assembly (dimensions of board, card, module, etc..), the packaging capacity on the assembly (how densely parts can be placed), and the manufacturability of the assembly (cost, yield, manufacturing rating).

The ADM is an hierarchical structure starts with a parent assembly and decomposes into children or subassemblies, similar to FDM. An abstract ADM is shown in Figure 5. The association between the parent and children assemblies is a containment relationship, that is, subassemblies are part of the parent assembly. When an assembly is manufactured, the sub-level assemblies are built first, and then they are integrated with their parent assembly. Each assembly is a manufacturable unit, either it can be manufactured internally in an enterprise, or it can be purchased from a vendor, or it may be available as an off-the shelf product. In all these cases, each assembly has manufacturing attributes which are very crucial for its production and also for its success in the market place.

The FDM can be mapped into ADM based on the FBE characteristics. One or more FBEs can be grouped and mapped onto a single assembly. Figure 6 shows such mapping for a Power Module example. The mapping process involves merging FBOMs into BOM when two or more FBEs are mapped into a single assembly, and also merging FLOPs into LOP. We need to develop mapping algorithms to map FDM to ADM. Currently, this mapping is done manually by the designer.

Each assembly is associated with a Bill of Material (BOM) which is a material and parts list for the entire assembly including the subassemblies. However, the subassemblies have their own part numbers and this part number is included in the BOM of the parent assembly. For example, a1, a2 are part numbers for the subassembly of a parent assembly, they are included in the BOM for the parent assembly. Thus, the BOM includes parts, material needed for this particular assembly, and also part numbers of the subassemblies. Each individual part or material has a part number by which it can be referenced in the database.

Each assembly is also associated with a List of Processes (LOP) or a router. A router describes the manufacturing steps required and also necessary information needed for the operator to put together the whole assembly. In particular, the router consists of a sequence of processes, setup time and runtime for each process, and a list of materials or parts that are involved with each process. For example, a router may consist of sequence of processes p1, p2, p3, ..., pn, where, a process p1 is related to material or parts c1, c2, ..,cp. To illustrate this further, a soldering process (ps) may be associated with a capacitor (c1), a resistor (c2), and a transistor (c3). The parts list (c1, c2, c3) is a process BOM (PBOM) for the soldering process (ps).

4.3 AND-OR Models For Optimization

There are always alternate choices available to perform a design, select a part or material, and to select a process. Thus, these alternate choices at product and process design phases offer variety of choices for the designer to pick and choose from and poses a critical decision making problem with complex domains of information. As shown in Figure 5 an assembly may have a router or router' to choose from for assembling the assembly. A process p3 may have alternate processes p3', and p3". Similarly, a part c1 may have alternate parts c1', and c1". It is possible to have alternate FBEs available for a given FBE. In addition to alternatives, there are also other design approaches where you can add/drop new parts, material, processes, FBEs, and assemblies. Thus, we classify these alternate options available in IPPD environment as follows:

- alternate parts
- alternate material
- alternate processes
- alternate functional block elements
- alternate assembly block elements (assemblies)

Some of the examples for the above design observations are in order. A part such as transistor may have several alternatives available based on different vendors, different cost, and different quality. A process such as "soldering" may have option of choosing from manual soldering, preformed soldering, wave soldering, or epoxy. A functional block element such as amplifier 1 may be replaced with some other amplifier. An assembly a5, and a6 can be merged to a2 thus dropping a5 and a6 assemblies. A new design for power module may add new FBE at a preamplifier level. These examples illustrate the above point and it is evident that these options are real and selecting appropriate choices is vital to the product cost and quality and to the commercial success of the product in global market place.

We captured the design options and choices available in FDM and ADM using AND/OR tree models (AOM). A typical instance of the input data for optimization problem is in the form of an AND/OR tree as shown in Figure 7. In the example being depicted by the figure, the main assembly consists of two subassemblies and a set of components (and associated processes) specific to the main assembly. As the tree evolves further, similar structures will emanate from the subassemblies.

The nodes of this tree represent assemblies, function blocks, generic part types, specific part instances and processes. Furthermore, each node may have alternate nodes, corresponding to the alternatives specified by the input data. The arcs capture both the input specifications (such as which process acts on which process) and the hierarchical structure of the design (such as the subassemblies and parts contained in each assembly).

As it can be seen from the figure, each node may have child nodes that are connected to it via 'AND-arcs' (all its children will need to be selected if this node is selected to be in the design) or via 'OR-arcs' (exactly one of the child nodes must be selected). We point out that it is always possible to construct the tree such that each node is either an 'AND-node' (all its children are joined to it via 'AND-arcs') or as an 'OR-node' (all its children are joined to it via 'OR-arcs').

Candidate designs (Pareto optimal solutions) are generated by choosing among the alternate nodes and arcs. The AND-OR tree, in addition to representing the input data in a concise and intuitive form, has some useful theoretical properties. In particular, we show that the integer programming constraints that capture the logical structure of an AND-OR tree, form an *integral polyhedron*, a fact that is crucial to the efficient solution of the multiobjective optimization problem.

4.4 Entity Relationships Models

We use entity relationships models (ERM) as our database models which are independent of underlying database system and data models. The Figure 9 shows the ERM for our manufacturing application. Attributes and other details in the figure are not shown for clarity and also protecting the proprietary nature of our customer product.

4.5 Relational Data Models

Based on the ERM as shown in Figure 9, we have developed relational data models[12] and used to implement the relational database management applications in Paradox, and other relational DBMS systems. These models are not discussed in this paper due to clarity and space problems.

4.6 Object-oriented Data Models

The object-oriented data models[14,18] are developed by using Entity Relationships models. The object-oriented data models are used to implement the ObjectStore application. The Object Manager consists of implementation of these object data models in addition to interfacing with optimizer, bridge, and GUI.

5.0 Optimization: Problem Definition

For ease of exposition, we will describe our approach for the following simplistic situation:

A conceptual design for the application at hand is given. The design specifies the set of required generic part types together with their functionality, and for each such part type, a number of specific component alternatives are considered. For each component, a process is to be selected from a given set of candidate processes. Key attributes such as material costs, runtimes, setup times, defect rates are known for parts, processes, and part-process combinations. It is assumed that the entire design is to be realized via a single assembly.

The problem we consider is to determine a set of parts and processes that realizes the design in a Pareto optimal manner with respect to cost and quality.

We begin by defining some notation:

$\upsilon = \{1, 2, \ldots, V\} =$ set of generic parts,

$\upsilon_j = \{1, 2, \ldots, V_j\} =$ set of alternatives for the jth generic part,

$\rho = \{1, 2, \ldots, P\} =$ set of processes,

$c_j =$ unit cost of jth part: $j \in \upsilon_k,\ k \in \upsilon,$

$s_p =$ setup time for pth process: $p \in \rho,$

$t_{pj} =$ runtime when pth process is used for jth part,

$\alpha_j =$ defect rate of jth part,

$\beta_p =$ yield rate of pth process,

$n_j =$ number of parallel components to be used for the jth part,

l = labor cost per unit time,

b = batch size.

Note that n_j represents a redundancy element that is usually considered while designing electronic devices. We assume that all the above quantities are provided as input data. We now define the following decision variables:

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

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

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

The expressions for cost (C_0) and quality (Q_0) are given as follows:

$$C_0 = \sum\nolimits_i n_i c_i x_i + l \sum\nolimits_{p,j} t_{pj} x_{pj} + \frac{l}{b} \sum\nolimits_p s_p y_p, \qquad (1)$$

$$Q_0 = \prod_p (\beta_p)^{y_p} \prod_j (1 - \alpha_j^{\,n_j})^{x_j}. \qquad (2)$$

The above equations are essentially derived on the following basis:

Cost = material cost + runtime cost + setup cost

Quality = Yield = (process yield) (material yield).

The decision variables in (1) and (2) ensure that only those elements that are selected to be in an assembly contribute to its cost and quality. We can now linearize (2) to get

$$\boxed{Q_0' = \log Q_0 = \sum_p y_p \log \beta_p + \sum_j x_j \log(1 - \alpha_j^{n_j}). \qquad (3)}$$

The problem we wish to solve is the following multiobjective 0-1 integer program:

$$\boxed{\text{minimize} \quad \left\{ \begin{matrix} C_0 \\ -Q_0 \end{matrix} \right\}}$$

subject to

$$\boxed{\sum_{j \in \upsilon_k} x_j = 1 \qquad k \in \upsilon \qquad\qquad (4)}$$

$$\boxed{\sum_{p \in \rho} x_{pj} = x_j \qquad \forall_j \qquad\qquad (5)}$$

$$\boxed{y_p \geq x_{pj} \qquad \forall_{p,j} \qquad\qquad (6)}$$

$$\boxed{x_j, y_p, x_{pj} \in \{0,1\} \qquad \forall_{j,p} \qquad\qquad (7)}$$

It is well-known that a set of efficient solutions to the above problem can be obtained by solving the following parametric problem (**P**):

$$\boxed{\text{minimize} \quad \lambda C_0 - (1-\lambda) Q_0', \qquad\qquad (8)}$$

subject to constraints (4)-(7),

where the parameter l ranges over the interval [0,1]. It is this version of the problem that we address in our project. For any given l, problem **P** directly corresponds to the well-known *incapacitated facility location* problem and, consequently, is NP-Hard. However, the problem has been relatively well studied and a number of solution strategies have been reported in the operations research literature. A tricky issue in our application here is that we would like to generate a set of Pareto optimal solutions parameterized with respect to l. Hence, in selecting a solution procedure, besides efficiency and accuracy we also need to consider the ease with which parametric analysis could be carried out. The solution approach that we propose at this time arises from the observation that the number of process alternatives involved in the T/R Module design is quite small with the key processes being

- manual or automated placement (assembly) for surface mount and leaded parts,
- manual or wave soldering for leaded (through-hole mounted) parts, and
- manual, wave or reflow soldering for surface mount parts.

The small number of possible processes implies that an approach which starts by enumerating all possible process combinations (y vectors) is computationally feasible. We then note that for a given set of selected processes, that is, a set $\boxed{\rho'}$ such that

$$\boxed{y_p = \begin{cases}1 & \text{if } p\in\rho',\\ 0 & \text{otherwise},\end{cases} \qquad (9)}$$

problem P becomes easy to solve. In fact, it is easy to show that the following greedy assignment rule produces an optimal solution.

GREEDY_SOLUTION(P')

```
for i:= 1 to V  /* for each generic part type */
  for j:= 1 to $V_i$  /* for each alternative of the ith generic part type*/
    p := $\boxed{l \in P' : t_{lj} \le t_{mj} \quad \forall m \in P' \quad /* \text{ greedy choice of process for part j } */}$
    $\boxed{cost[j] := \lambda(n_j c_j + lt_{pj}) - (1-\lambda)(1-\alpha_j^{n_j}) \quad /* \text{ cost out part j } */}$
  end for
  $\boxed{choice[i] := l \in \upsilon_i : cost[l] \le cost[m] \quad \forall m \in \upsilon_i \bullet}$
  /* greedily select alternative for ith generic part type */
end for
```

For a given set P' of selected processes, the greedy algorithm stated above first assigns the cheapest process for each part alternative. It then costs out each part alternative and subsequently selects the minimum cost alternative for each generic part type. It can easily be shown that the above algorithm generates an optimal solution.

The greedy approach however does not permit a straightforward procedure for parametric analysis with respect to 1. For this purpose, consider the following reduced problem **P**(*P*')

$$\boxed{\text{minimize} \quad \lambda C_0 - (1-\lambda)Q_0 \bullet}$$

subject to constraints (4)-(7) and (9).

It can be shown that the associated constraint matrix is totally unimodular, and hence, the feasible region for this problem is an integral polyhedron. Consequently, the linear programming (LP) relaxation to this problem is guaranteed to produce an integer solution. As such, this problem could be routinely solved using standard LP software such as LINDO or CPLEX and furthermore, full parametric analysis with respect to l could also be efficiently performed.

In summary, our approach is to solve $2^{|P|}$ subproblems, one for each choice for P'. For each subproblem, the optimal objective function value is obtained as a piece-wise linear function of l. The lower envelop of the family of these functions yields the parametric solution to the original problem **P**. The situation is shown in Figure 8.

6.0 Implementation

We have implemented the proposed system architecture in the C++ programming language environment adhering to all object-oriented programming techniques. Integrating various software applications through single programming language interface was a major challenge to the system programmers. The platform for this implementation is based on traditional client-server environment.

The object-oriented database environment supports a UNIX server and a PC client running Windows NT operating system. All application programs are written in Visual C++ and run under NT operating system. The Microsoft foundation classes (MFC) are used for writing the GUI applications and this code is completely portable to other environments. The ObjectStore database supports UNIX server and PC client environment and the OODBMS applications are also written in Visual C++.

The CPLEX optimizer supports C function libraries and we have tightly integrated the CPLEX interface with object-oriented database. The data required for the optimizer resides in the ObjectStore, and Visual C++ queries are written to provide input to the CPLEX application. The results of the CPLEX application (optimal solutions for the designs and scenarios) are further stored back in the object-oriented database through transactions. For the power module application demonstrated in this project, we have not run into any performance problems by tightly integrating CPLEX and object-oriented database systems. However, for larger applications we anticipate performance problems as the user invokes the optimizer iteratively until he/she obtains a "**good**" design solution.

The paradox frontend GUI sample is shown in Figure 10 and the ObjectStore GUI interface is shown in Figure 11.

7.0 Conclusions & Further Research

The integrated product and process design automation tool development is crucial to the success of microwave module product or any other avionics (electronics) product manufacturing. As illustrated in this paper, there are numerous research and technical issues to be clearly understood and resolved in order to attempt to build such a tool. For example, integration of databases (relational and object-oriented) with real world applications is a challenging issue alone. Integrating product design and process design are crucial objectives to achieve low cost, high quality, short lead time, and fast development cycle time. In addition, integrating multiobjective optimization techniques in a software system is a major technical challenge. As challenging as it may be, this approach will result in optimal design choices and thus saving substantial cost to the manufacturer.

We have demonstrated the T/R Module tool in our laboratory and currently working on enhancing the tool to interface with variety of legacy databases. The efficiency and performance issues for the bridge are currently under investigation and we need further research in this area. In addition, we plan to demonstrate this prototype at our customer site where the tool will be closely integrated with the manufacturing environment.

References

[1] Michael O. Ball, John S. Baras, Sridhar Bashyam, Ramesh K. Karne, and Vinai S. Trichur, On the Selection of Parts and Processes during Design of Printed Circuit Board Assemblies, submitted to INRIA/IEEE Conference, October 1995.

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

[3] C.A.S. Heng and R.K.L.Gay, 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.

[4] G.A.Russell, 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.

[5] A. Kusiak, E. Szczerbicki, and K. Park, A novel approach to decomposition of design specifications and search for solutions, International Journal of Production Research, 29(7):1391-1406, 1991.

[6] Mark R. Cutkosky, Robert S. Engelmore, Richard E. Fikes, Michael R. Genesereth, Thomas R. Gruber, William S. Mark, Jay M. Tenenbaum, and Jay C. Weber, PACT: An Experiment in Integrating Concurrent Engineering Systems, Computer, January 1993, pages 28-37.

[7] Dan L. Shunk, Integrated Process Design and Development, Business One Irwin, 1992.

[8] Mike McMahaon, R. Prasannappa, John S. Baras, G. Zhang, Abraham Kebede, and P. Mendicino, Evaluation of Product and Process Design in Electronics, Technical Research Report, TR92-100, Institute for Systems Research, University of Maryland, College Park, MD 20742.

[9] Jong-Shin Liau, Robert E. Young, A Process Planning and Concurrent Engineering System for PCBs, Manufacturing Review, Volume 6, No.1, March 1993.

[10] Vilma M. Irizarry-Gaskins, Tien-chien Chang, A Knowledge Based Approach for Automatic Process Plan Generation in an Electronic Assembly Environment, International Journal of Production Research 28 (1990): 1673-1693.

[11] Fumihiko Kimura, Product and Process Modeling as a Kernel for Virtual Manufacturing Environment, CIRP Annals 1993, Manufacturing Technology, Volume 42/1/1993, pages 147-150.

[12] S. Kastelic, J. Kopac, and J. Peklenik, Conceptual Design of a Relational Database for Manufacturing Processes, CIRP Annals 1993, Manufacturing Technology, Volume 42/1/1993, pages 493-496.

[13] A. Nestler, System Integration and Technological Database, IFAC Workshop on Intelligent Manufacturing Systems 1994, pages 109-113.

[14] KAJ A. Jorgensen, and Ole Madsen, Object-oriented Modeling of Active Systems, IFAC Workshop on Intelligent Manufacturing Systems 1994, pages 129-133.

[15] Klaus Feldmann, and Jorg Franke, Computer-Aided Planning Systems for Integrated Electronic and Mechanical Design, 1992 International Electronics Manufacturing Symposium, Baltimore, MD, September 28-30, 1992.

[16] G. Harhalakis, I. Minis, and H. Rathbun, Manufacturability Evaluation of Electronic Products Using Group Technology, In Proceedings of the 1993 NSF Design and Manufacturing Systems Conference, pages 1353-1360, Charlotte, NC, 1993.

[17] S.K.Padhy, and S.N.Dwivedi, 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.

[18] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., Object-oriented Modeling and Design, Prentice Hall.

[19] Burleson, D. K., Practical Applications of Object-Oriented Techniques to Relational Databases, A Wiley-QED Publication, John Wiley & Sons, Inc., 1994, Chapter 7.

Figure 1: Integrated Product and Process Design Environment

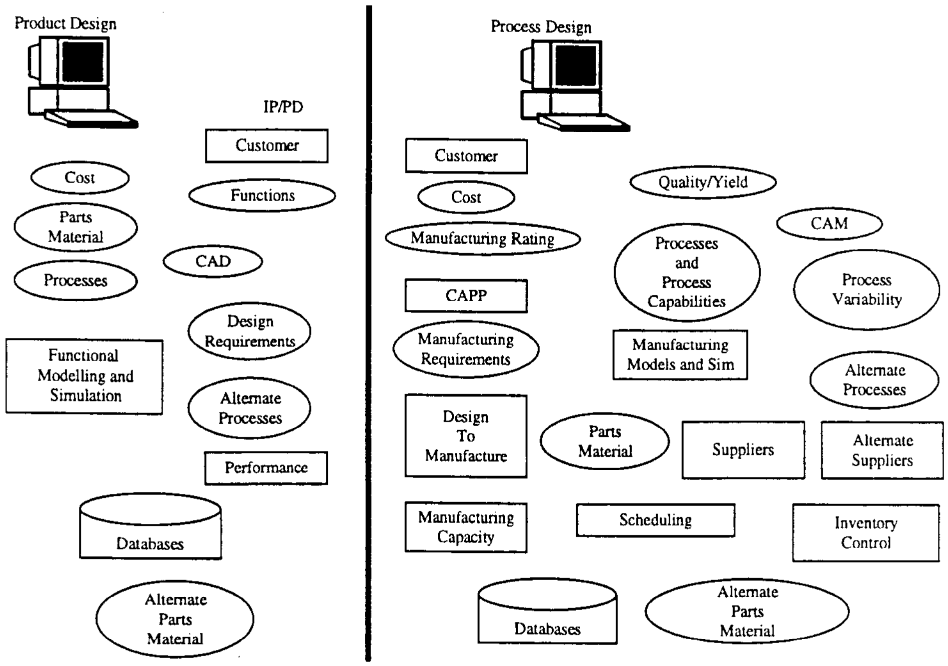


Figure 2: System Architecture for T/R MODULE

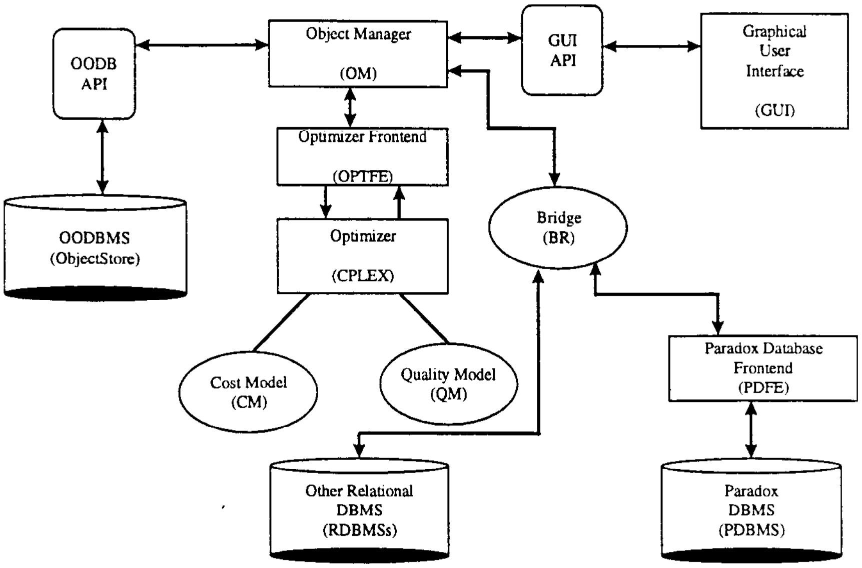


Figure 3: Functional Data Model (FDM)

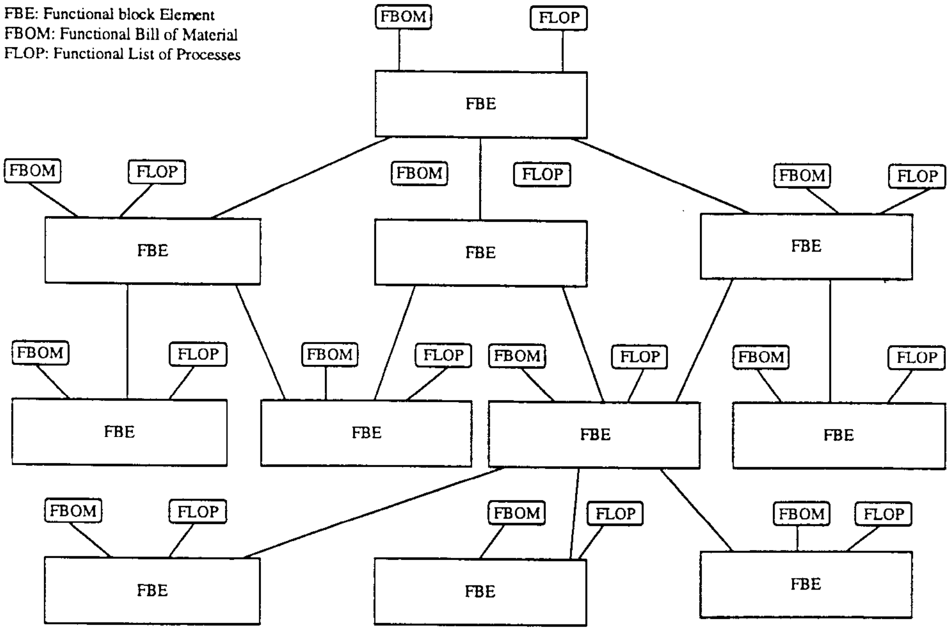


Figure 4: Power Module FDM

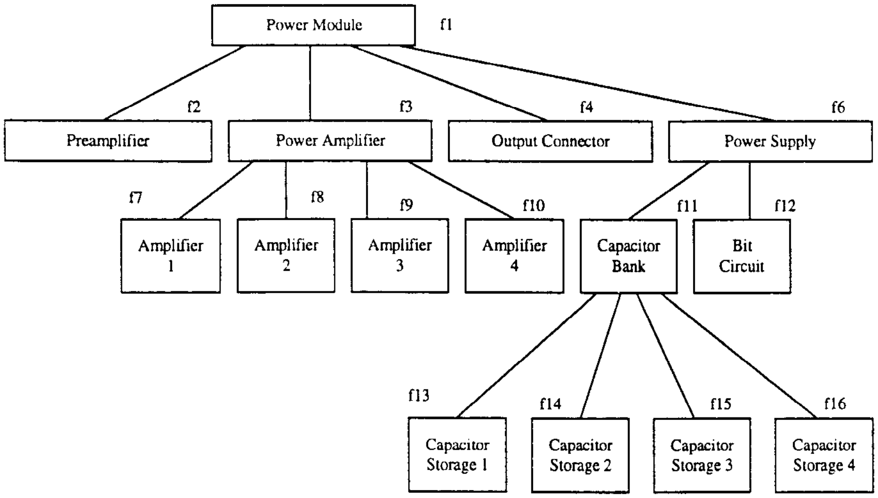
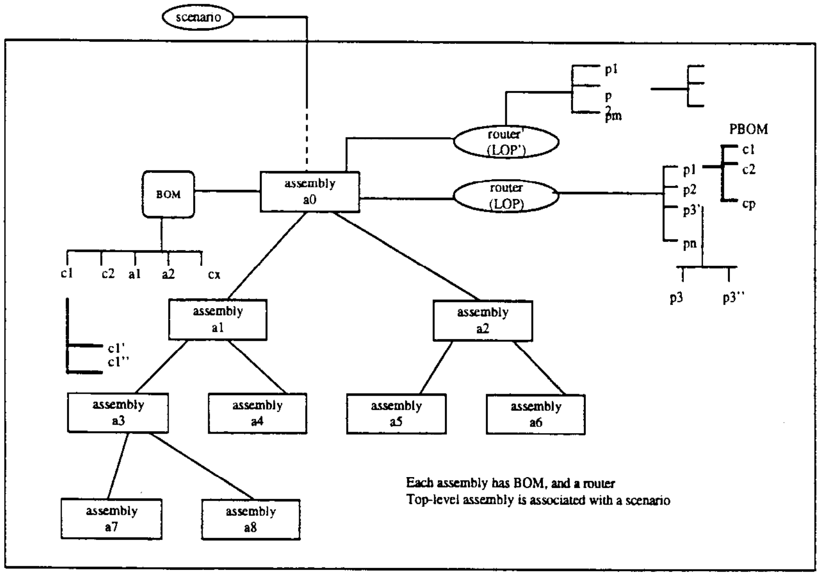


Figure 5: Assembly Data Model (ADM)



Each assembly has BOM, and a router
Top-level assembly is associated with a scenario

Figure 6: Power Module ADM

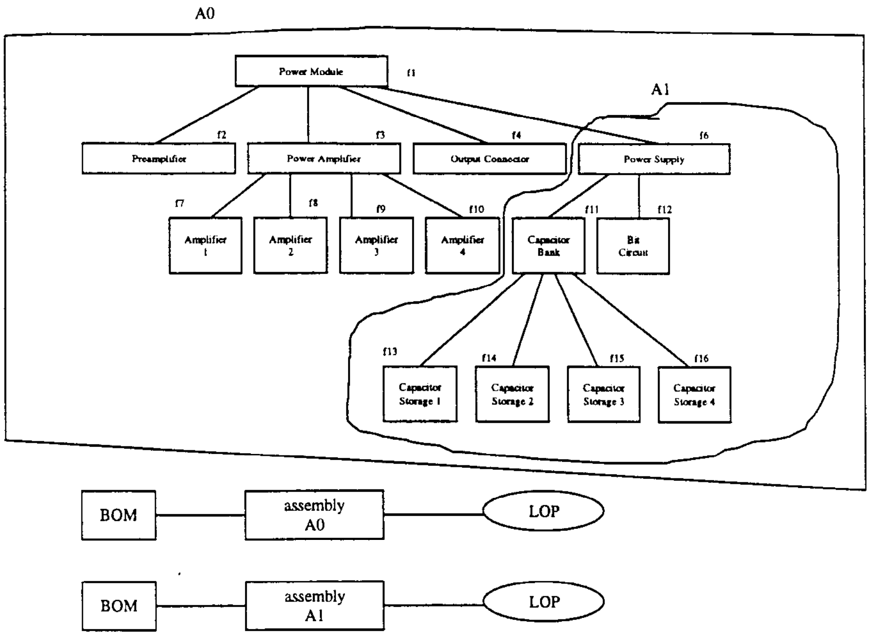


Figure 7: AND/OR Model

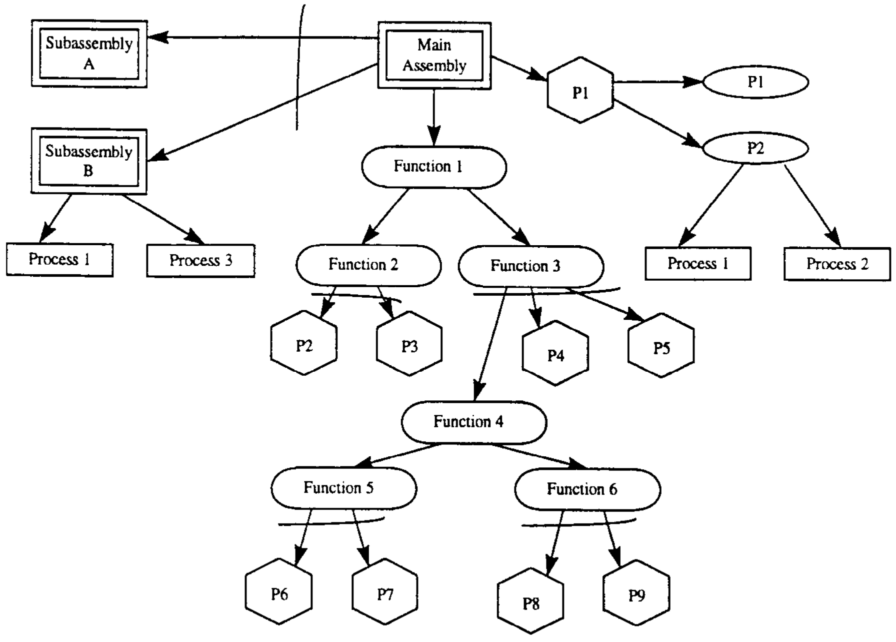
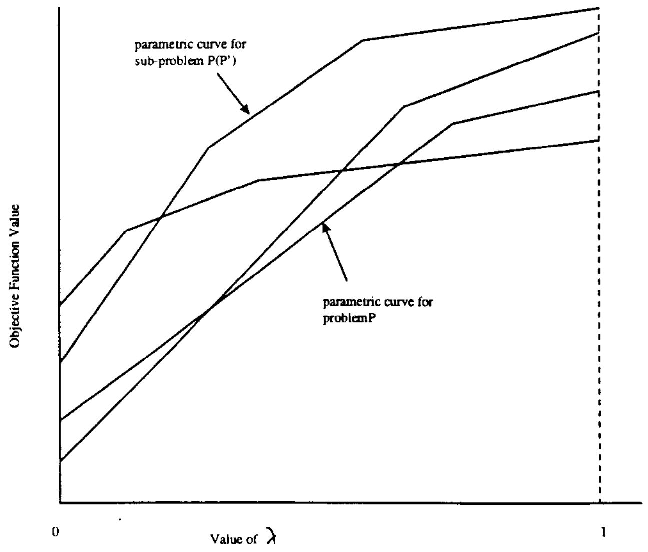


Figure 8: Parametric Analysis with respect to λ



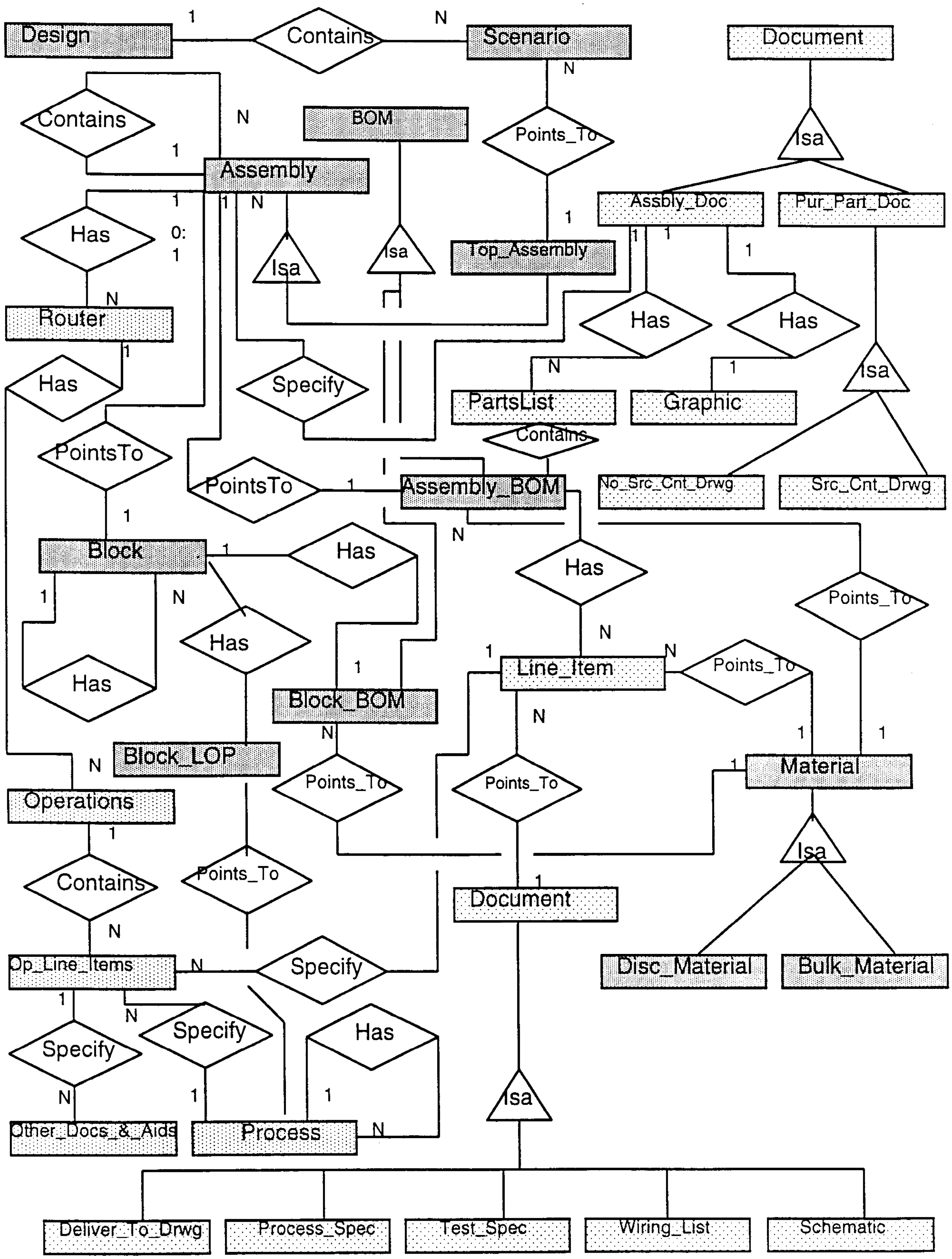
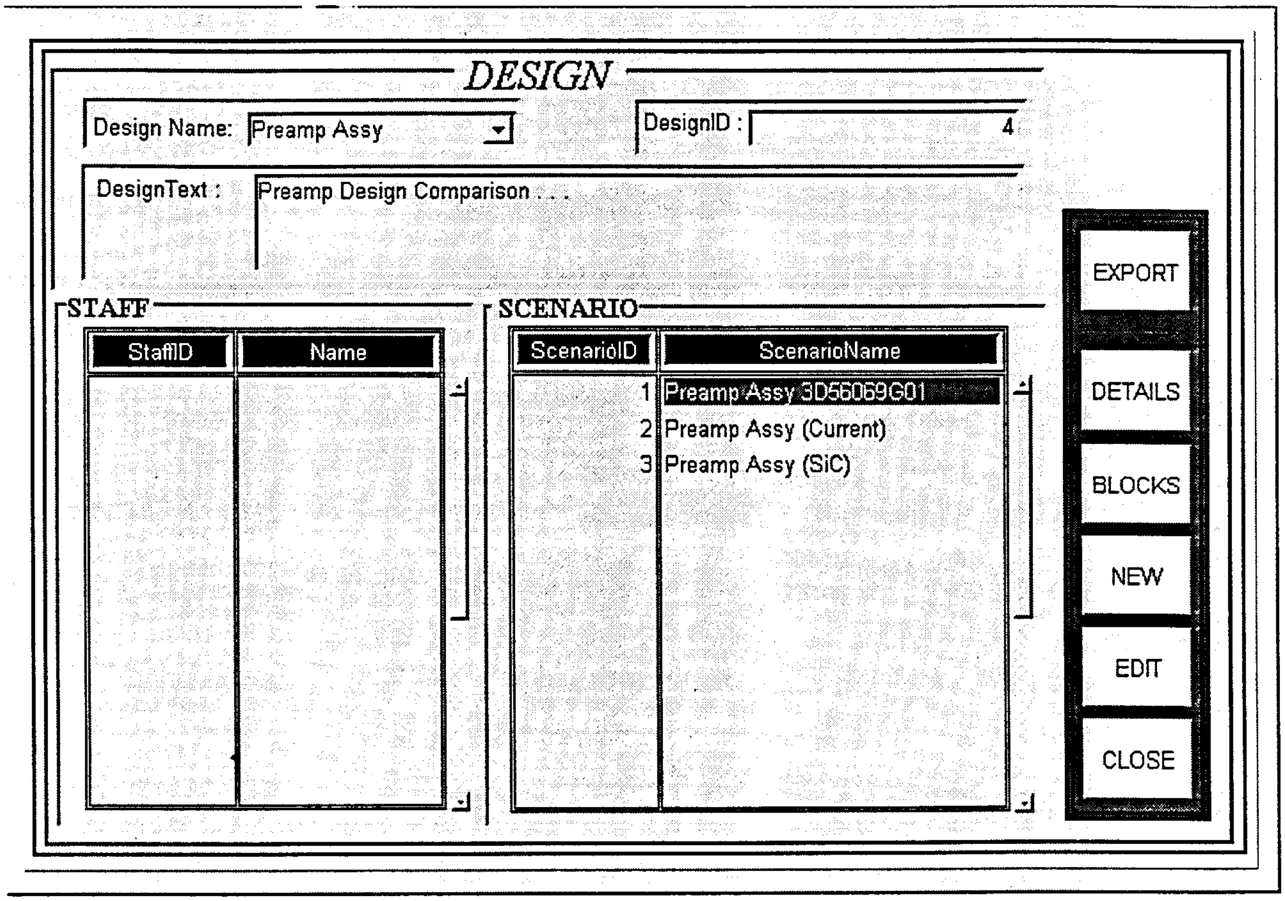


Figure 9: **ER Model Diagram**

Figure 10: Paradox Interface T/R Module



Design Name: Preamp Assy

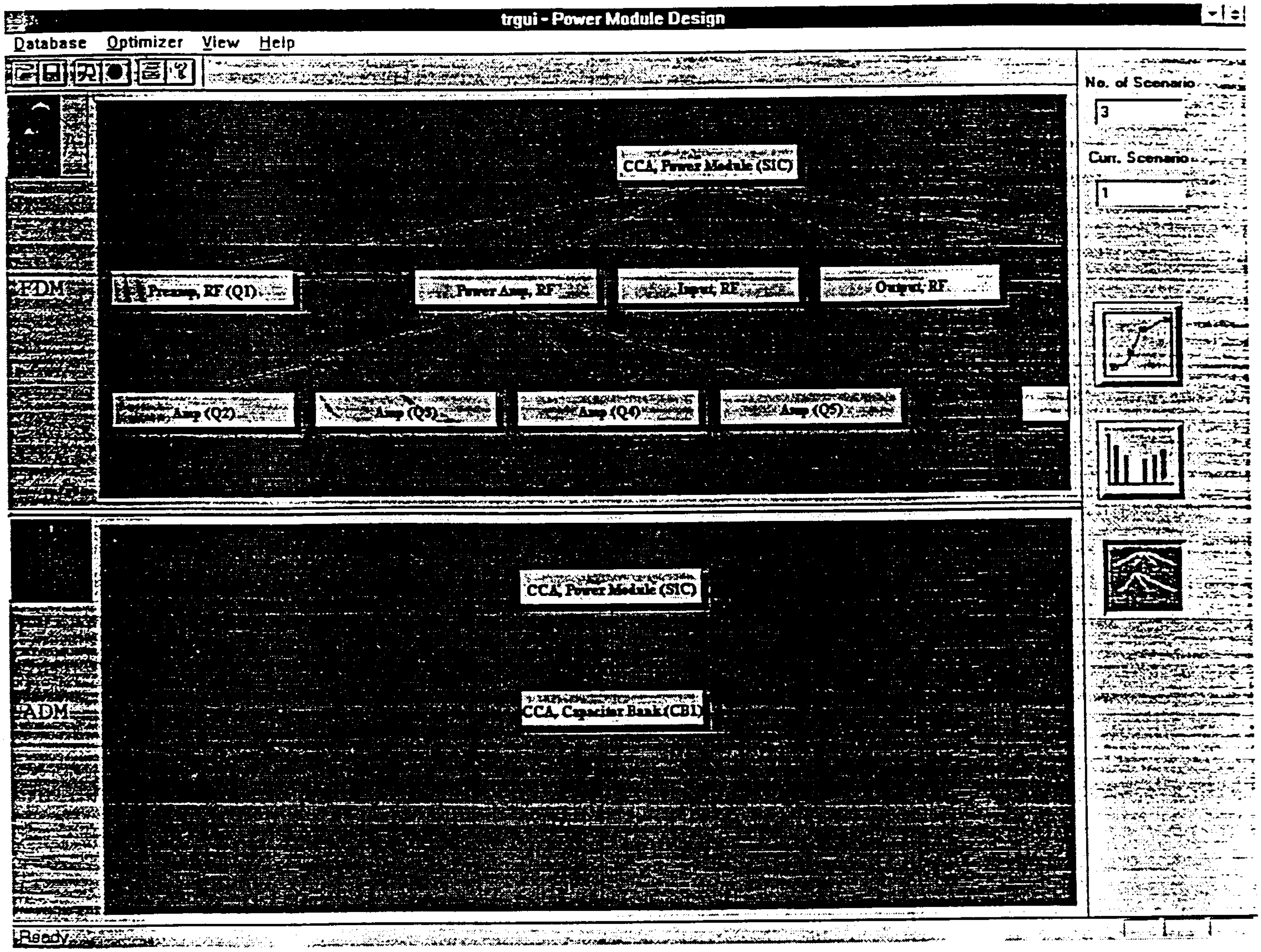
DesignID : 4

DesignText : Preamp Design Comparison . . .

StaffID	Name

ScenarioID	ScenarioName
1	Preamp Assy 3D56069G01
2	Preamp Assy (Current)
3	Preamp Assy (SiC)

Figure 11: **ObjectStore Interface T/R Module**



No. of Scenario: 3

Curr. Scenario: 1