JB 95-16

Entitled:

"Decision Making Assistant for Integrated Product and Process Design Environment"

Authors:

(with R.K. Karne, M.O. Ball, S. Bashyam,
M. Townsley, F. Huang, V.S. Trichur, M. Karir,
H. Xie and A. Inbar)

Conference:

Proceedings of Symposium on
Timely Realization of Affordable Military Systems
Through Enhanced Manufacturing Technology
Albuquerque, NM
October 18-20, 1995

Sandia National Laboratories

Technical Collaborations Symposium

Timely Realization of Affordable Military Systems Through Enhanced Manufacturing Technology

October 18, 19 & 20, 1995
Holiday Inn Pyramid
Albuquerque New Mexico

Sponsored Agencies

Advanced Research Projects Agency (ARPA)

Joint Advanced Strike Technology (JAST)

Sandia National Laboratories

Decision Making Assistant for Integrated Product and Process Design Environment

Ramesh K. Karne, John S. Baras, Michael O. Ball , Sridhar Bashyam,
Mark Townsley, Feng Huang, Vinai S. Trichur, Manish Karir, Hui Xie, and Asaf Inbar

Institute for Systems Research
University of Maryland
College Park, MD 20742

Abraham Kebede, Jim Williams
Westinghouse Corporation
Electronics Systems Group
Baltimore, MD 21203

Abstract in reference to "**Symposium on Timely Realization of Affordable Military Systems Through Enhanced Manufacturing Technology**"

(**Contact**: Ramesh K. Karne, karne@src.umd.edu, 301-405-7062, fax 301-314-8586)

Abstract

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, quality, and other metric, 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 our country's economy will depend upon how thoroughly 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 product and process design in manufacturing are treated as two separate design environments. However, a given process may force to change its design if it is not manufacturable. The product and process 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 propose multiobjective optimization models 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 are suitable for virtual manufacturing.

We present a decision making assistant (DMA) tool for integrated product and process design (IPPD) 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.

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 DMA/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.

Decision Making Assistant for Integrated Product and Process Design Environment

Ramesh K. Karne[1] , John S. Baras [1], Michael O. Ball [1], Sridhar Bashyam [1],
Mark Townsley[1], Abraham Kebede[2], Jim Williams[2] , Feng Huang [1],
Vinai S. Trichur[1], Manish Karir[1], Hui Xie [1], and Asaf Inbar[1]

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[7], and computer aided simulation tools[15].

The integrated product and process design (IP/PD) is a complex systems integration problem as well as a challenging optimization problem. This is a daunting problem in manufacturing, also addressed by ARPA, and recently awarded a research contract to Rockwell International Corporation as a prime contractor. 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

---

[1] with University of Maryland, Institute for Systems Research, College Park, MD 20742.

[2] with Westinghouse Corporation, Baltimore, MD.

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 IP/PD 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 decision making assistant (DMA) 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 DMA 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 DMA tool and expects the tool to generate alternate options and present trade-off analysis with respect to all possible designs.

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

♦ Integrate product and process designs

Develop an object-oriented data models to capture the information of product designs as well as process information and make this data available to the tool through an object-oriented query interface. This integrated information is stored in an object-oriented database (Object Store) and readily available to the user through a graphical user interface.

♦ Map legacy data from relational databases to object-oriented databases

The data available for microwave modules is spread across multi-vendor databases that exist at a number of sites across the enterprise. For example, parts and material data may be in a DB2 database, and purchasing cost data may be stored in an Oracle database. Some of the data, particularly process *"instructions"*, exists as text that was intended for humans to read and comprehend and thus is not stored in a manner which is immediately useful to a computer program.

Legacy databases are predominately relational database management systems. However, manufacturing data models are clearly hierarchical in nature (assemblies, subassemblies, etc..). Object-oriented data models and programming techniques are best suited for efficient implementation of these types of applications. Thus, there is a need for mapping the relational data models to the object-oriented data models.

♦ Apply multiobjective optimization techniques to manufacturing domain

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] appear to be more along the lines of work being reported here.

Our optimization techniques used in the DMA/IPPD 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.

♦ Develop client-server software for DMA/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 DMA/IPPD system architecture is shown in Figure 2, and the following sections describe more details of this architecture.

The software architecture for the DMA/IPPD 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 is a text document, 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 interface. The PDFE is a Paradox application program which is written in ObjectPal and has custom designed forms and primitive graphical user interface (GUI) availabe to the designer. Numerous other relational DBMS (RDBMSs) can be interfaced with the system through a bridge (BR) which 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. As the ODBC query internally maps them to a given RDBMS system query, this complexity is not shown in our system architecture. However, the bridge program has to formulate and interface with PDFE and also RDBMSs.

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.

Move this paragraph to Implementation section...

We have developed the GUI interface for Microwave modules as shown in Figure 3, and 4. The Figure 3 shows the treebrowser capabilities, and the Figure 4 shows the some of the output charts that can be displayed by the DMA tool.

Hui will write a paragraph on his architecture ..

3.3 Bridge

There is a need for a bridge 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. There are numerous approaches to implement this function, however, we have taken a brute-force approach to realize it.

---- Mark will write this section (only architecture point of view)

3.4 Optimization

The designer enters initial designs and scenarios and they are stored in the Paradox DBMS. After the designer enters the data, he/she can invoke the optimizer and monitor the results and also perform numerous sensitivity analysis. The tool used for the optimizer is the CPLEX optimizer, which offers C programming library interface and the data required for the optimizer (coefficients, data values, etc..) are provided by the optimizer frontend (). The OPTFE is C++ application program which provides data to the optimizer by performing appropriate queies to the object-oriented DBMS (OODBMS).

The CPLEX provides a C function calls as API and we integrated the CPLEX libraries tightly with the OODBMS. The OPTFE provides inputs to the optimizer and also takes the outputs from the optimzier and stores the data in the OODBMS. The input to the optimizer is initial designs and scenarios and the output of the optimizer is one or more optimal designs and scenarios with corresponding cost and quality attributes. The designer can browse this information and perform trade-off analysis provided by the GUI.

-- Sridhar will write this section (only architecture point of view if needed)

3.5 Object Manager

The Object Manager (OM) interacts with the object-oriented database and provides a pivotal role in the DMA/IPPD 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 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.

The OODBMS must support the critical requirements of DMA/IPPD. Some of the requirements are listed below:

- **Graphical interface** (allow the operator to view the designs at any level of detail, that is, to graphically navigate the Database)

- **Scalable design** (add new designs, or increase the number of designs without restructuring the database)

- **Real-time response** (store and process data in real-time)

- **Temporal views** (provide a *"snapshot"* of the hierarchy as of some real-time instant)

- **Distributed** (allow distributed data and management)

- **Object-oriented** (benefits of object-oriented programming and development)

Considering the above database requirements, we have chosen the ObjectStore as our OODBMS system. The ObjectStore OODBMS addresses most of the above issues, and provides a single C++ programming interface for data manipulation (queries) as well as data definition (standard data structures).

3.7 Cost Model

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

3.8 Quality Model

We have developed a quality model based on process yield, subassemblies, process defect rate, and material defect rate at an assembly level, where each assembly is a manufacturable unit. The 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 an object-oriented programming environment. In this paper, we describe two basic data models that are applicable to manufacturing applications: functional data models, and assembly data models.

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 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 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 x. 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 x 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 merging algorithms for tree structures as the FDM and ADM are tree data structures and the knowledge has to be input by the designer at this point. As the DMA tool does not have knowledge of physical layout of the parts and processes, the designer input to mapping FDM to ADM is required.

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 Design, data, and alternatives

A product design and development, from concept to product can be grouped into two major categories. A product design, and a process design. For a given product, each one of these areas are designed and developed independently by different people, data models, and tools. The current research in integrated product and process design (IP/PD) are focusing on breaking this wall between product and process design phases and closely integrating the design issues.

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 x 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 IP/PD environment as follows.

- alternate parts
- alternate material
- alternate processes
- alternate functional block elements
- add/drop parts
- add/drop material
- add/drop processes
- add/drop FBEs
- add/drop 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.

The FDM, and ADM models clearly demonstrate that the complexity of the problem in several dimensions for a possible designs. These alternate designs possible are extremely large and there is a need for optimization in order to select a best possible design. This problem is formulated as a multi-objective optimization problem as later described in this paper.

4.4 AND/OR Models for Optimization

Sridhar....................

4.5 Relational Data Models

Mark.................

4.6 Object-oriented Data Models

The data required for the DMA tool is stored in object-oriented database and thus the object-oriented data models are needed for efficiently manipulating data. We have studied the electro-mechanical applications in detail and analyzed the data models that are required for these applications. A complete object-oriented analysis, and design are not provided in this paper, however, object classes, and their relationships are described in this section..

The key object classes required for electro-mechanical manufacturing applications and their hierarchy are shown in Figure 7 and their relationships are shown in Figure 8. Rumbaugh[20] notation is used representing the object models.

4.7 Mapping Relational to Object-oriented Data Models

There are numerous ways of mapping relational to object-oriented data models[21]. We have chosen the most primitive approach where each object class attributes are obtained from the relational tables and forms by performing queries to the RDBMS. The queries are based on ODBC queries so that the our application program does not have to worry about tailoring the query for a particular DBMS. This is hidden by the ODBC as there will be a separate ODBC driver required for each DBMS environment. The operating system together with the ODBC support will help hiding this complexity.

In addition to the straight forward mapping as described above, we have also optimized the mapping by closely designing the relational tables and forms that can be easily mapped into our object data model. To achieve best performance, during initialization, we invoke an application program to do this mapping and transfer the data from relational database to object-oriented database (ObejctStore) and updates to the RDBMSs are done when exiting the application. This may cause storage inconsistencies when the DMA application is running, in order to solve this problem, we plan to do frequent updates to the RDBMSs.

5.0 Optimization: Problem Definition

We consider a multiobjective optimization model that determines components and processes for given conceptual designs of microwave modules. Specifically, our model creates a set of solutions that are Pareto optimal with respect to a cost, quality, and manufacturing rating metric. We will focus this problem on an integer programming based solution strategy, and apply this technique to microwave modules as our first application. Furthermore, we would like to generalize our solution to other application domains and several performance metrics.

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 where the database stores are at the servers, and the clients access the data from server as needed. The databases used in the system support and also the applications written in C++ will adhere to client-server environment and allow the DMA tool to be used in a distributed environment.

The object-oriented database environment supports a UNIX server and a PC client running Windows NT operating system. Our 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 can be written in Visual C++. The Paradox DBMS applications, PDFE is written in ObjectPal and this is different from other application programs.

The Object Manager, and Bridge 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 are optimal solutions for the designs and scenarios and they are further stored in the object-oriented database through transactions. For the power module application demonstrated in this project, we have not run into a performance problems with respect to integrating CPLEX and object-oriented database tightly in the system. However, for larger applications we anticipate performance problems as the user invokes the optimizer iteratively until he/she satisfies with "good" solutions.

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. 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.

During Phase I of our research, we have focused on the following issues:

- Identified requirements for an IP/PD environment
- Developed functional and assembly data models and acquire data for microwave modules
- Developed relational data models for the Relational Database Front End and collect data from real environment
- Developed Bridge requirements and write ODBC interface
- Design, code and test software modules Object Manager, GUI, Relational Database Frontend, and Optimizer Frontend
- Formulated multiobjective optimization based trade-offs for microwave modules (cost, and quality)
- Tightly integrated CPLEX optimization tool with the IP/PD Design Automation Tool

During Phase II which is already underway, we plan to focus on the following design issues:

- Extend the bridge interface to dynamically query information from the remote or distributed databases
- Enhance the multiobjective optimization models for manufacturing rating and other metrics
- Investigate performance problems with respect to relational and object-oriented databases, and CPLEX tight coupling
- Define object class libraries for IP/PD environment
- Enhance Graphical User Interface to suit the design environment
- Evaluate the tool to apply to other application domains
- Develop and implement tradeoff analysis techniques for microwave modules
- Test the prototype with other domains and customers

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, submotted 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] Integrated Product and Process Design, Book..

[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 Modelling 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 Modelling 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] S.J. Schwartz, and M.T. Shreve, GT AND CAPP - Productivity Tools for Hybrids, The International Journal for Hybrid Microelectronics, Vol. 5, No. 2, November 1982.

[19] C.C.Gallagher, and W.A.Knight, Group Technology, (London, Butterworth and Co., 1973) p.3.

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

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

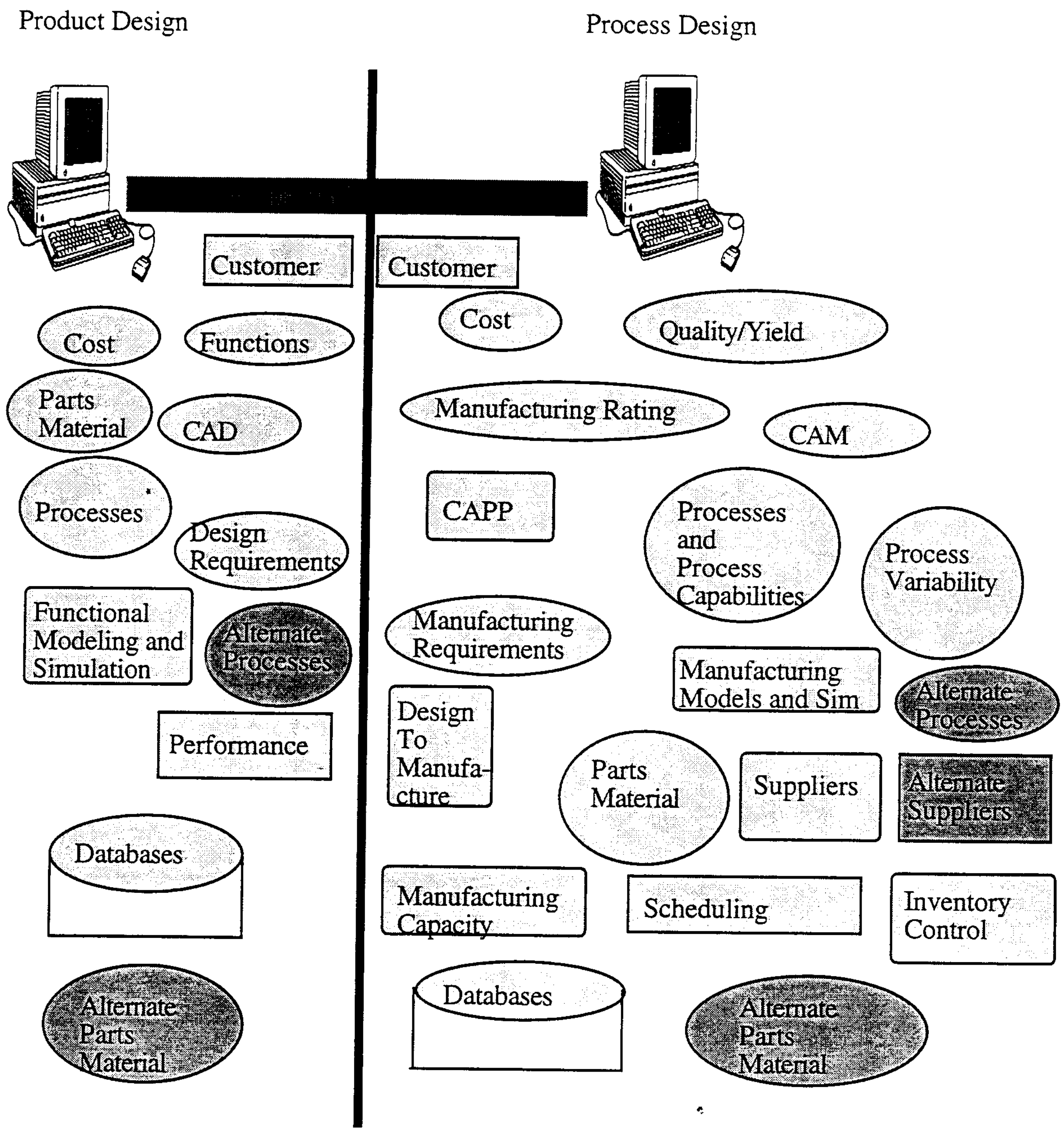


Figure 1: **Integrated Product and Process Design Environment**

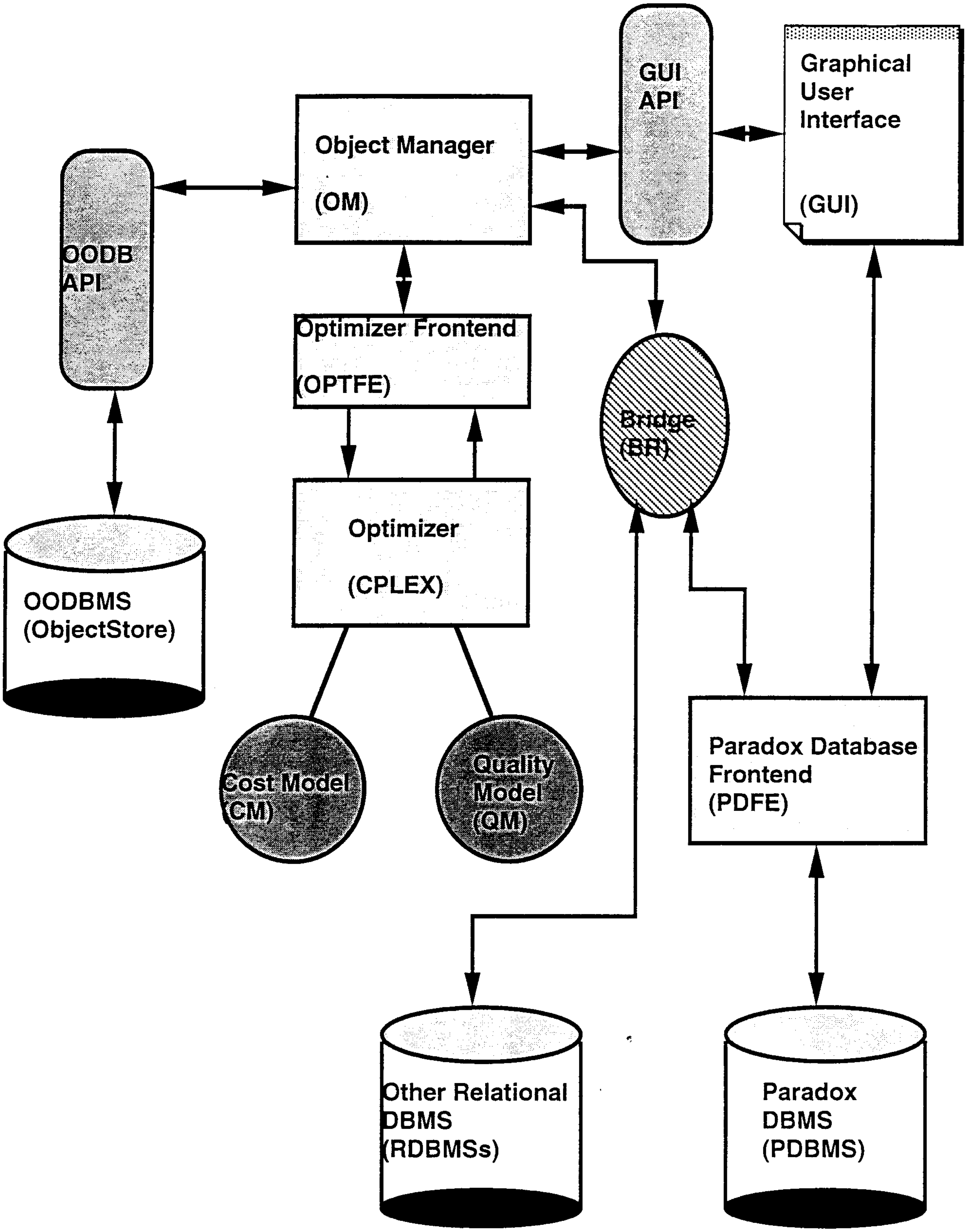
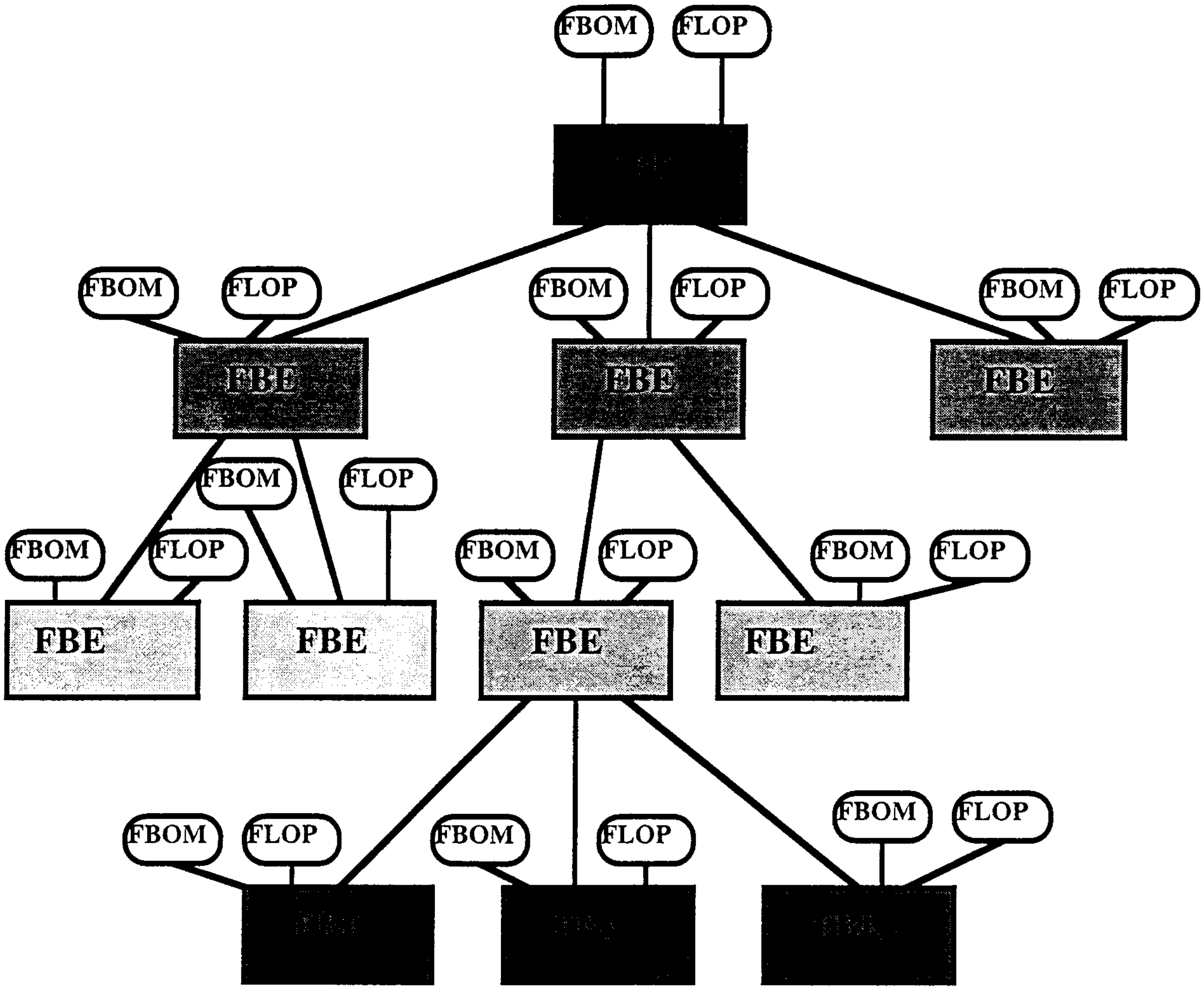


Figure 2: System Architecture for DMA/IPPD



FBE: Functional Block Element
FBOM: Functional Bill of Material
FLOP : Functional List of Processes

Figure 3: Functional Data Model (FDM)

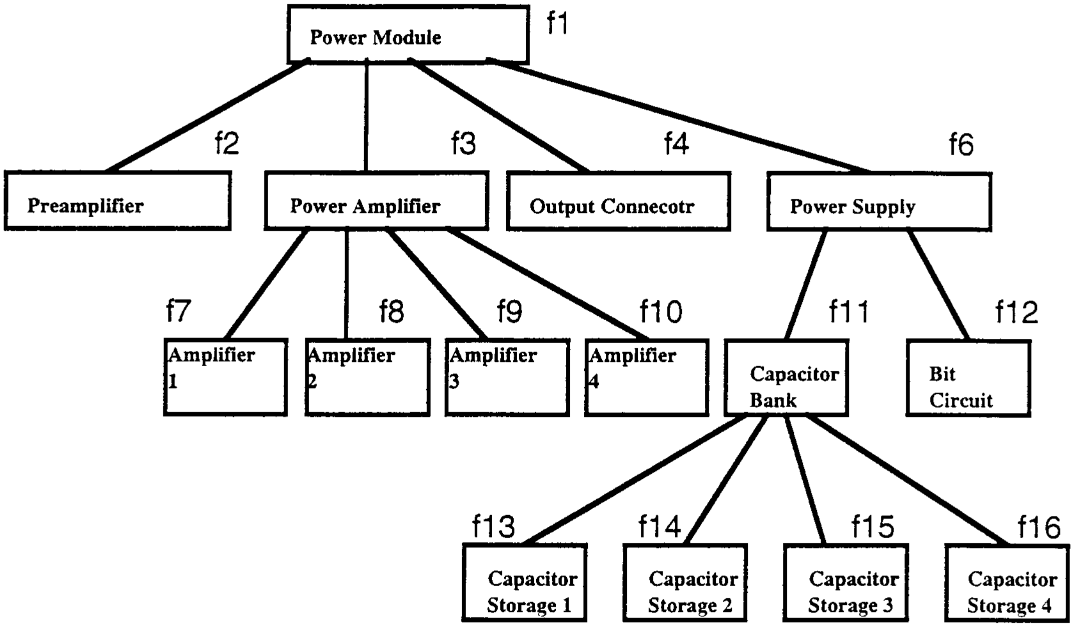
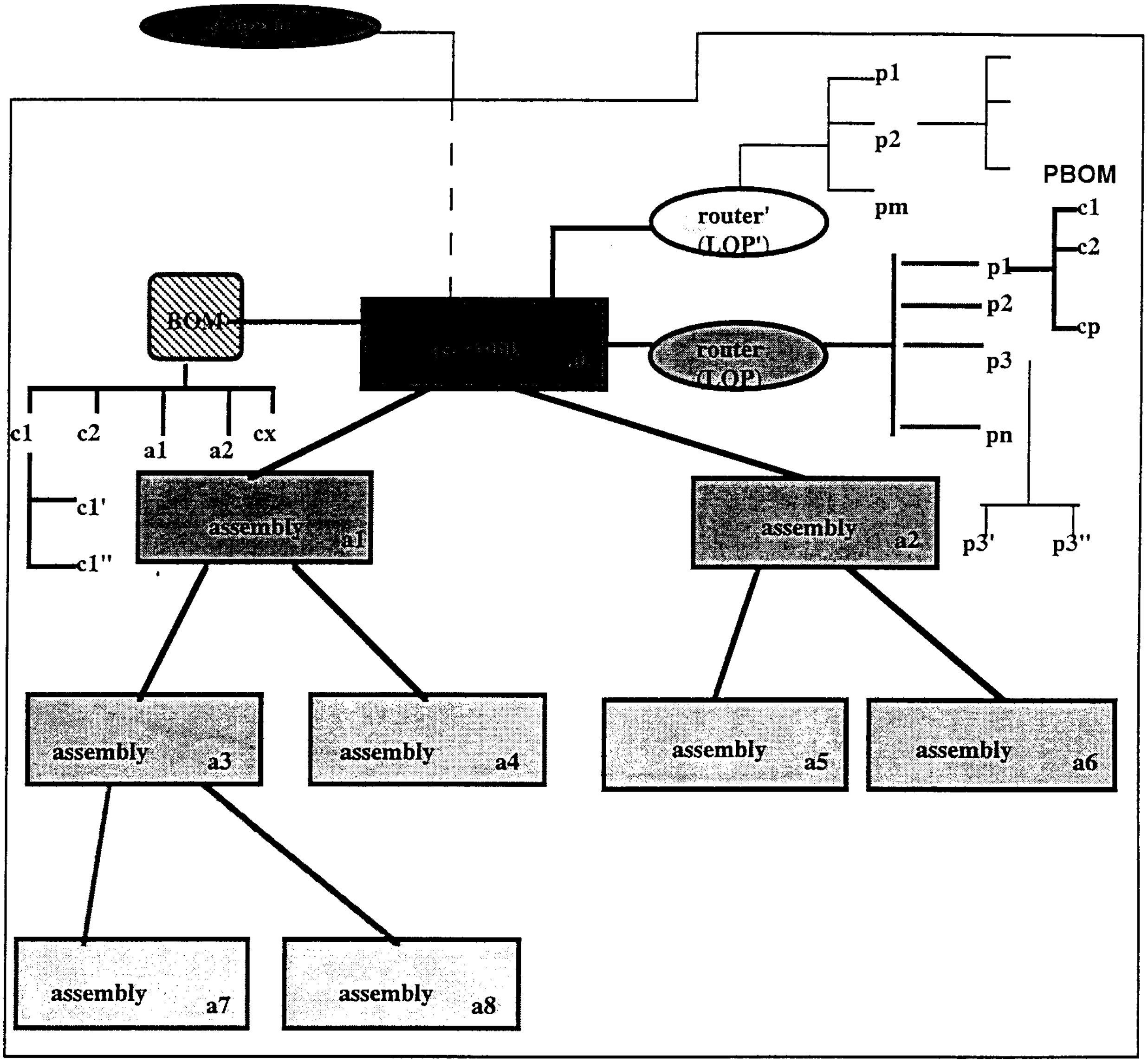


Figure 4: **Power Module FDM**



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

Figure5: Assembly Data Model (ADM)

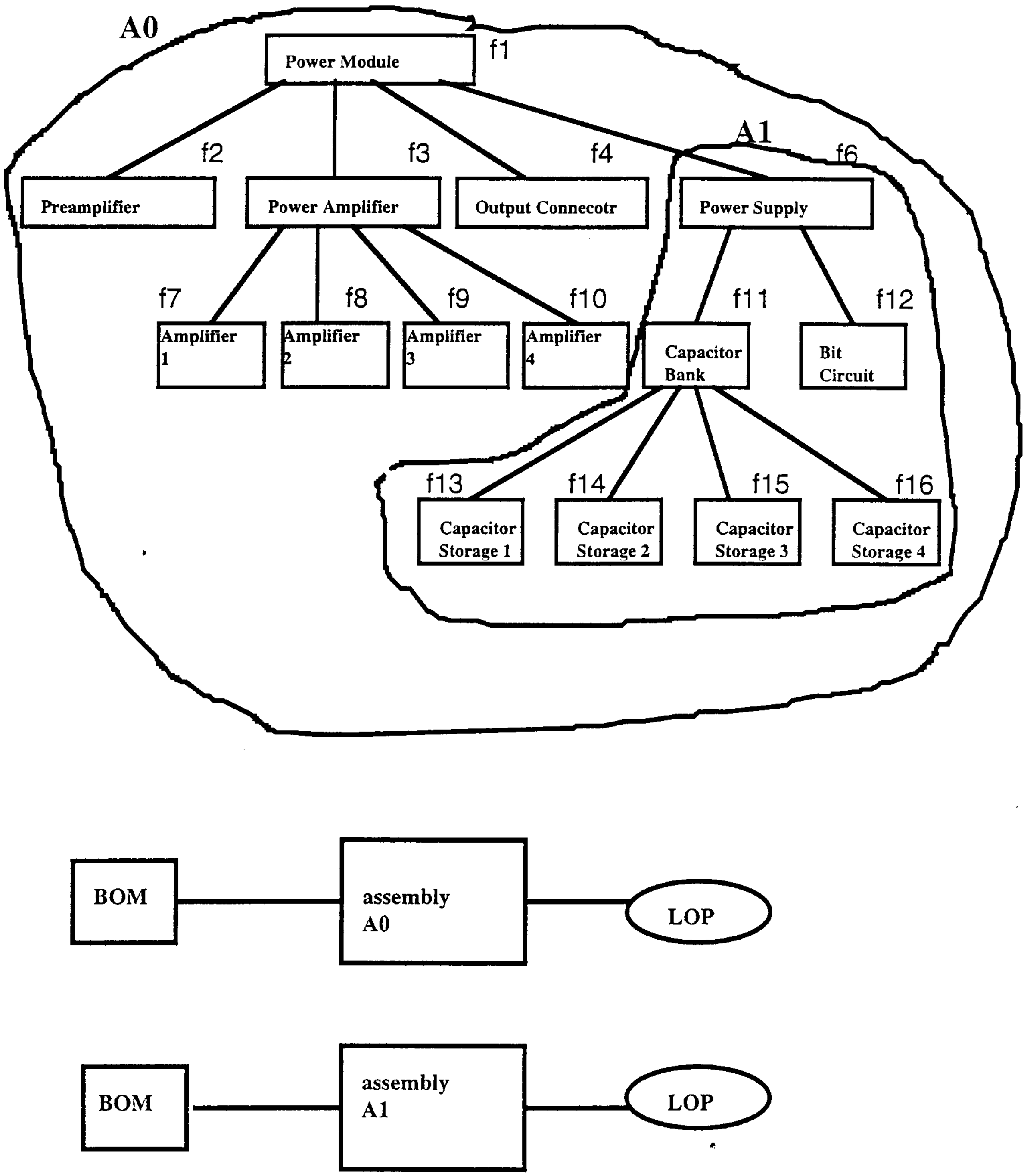


Figure 6: Power Module ADM

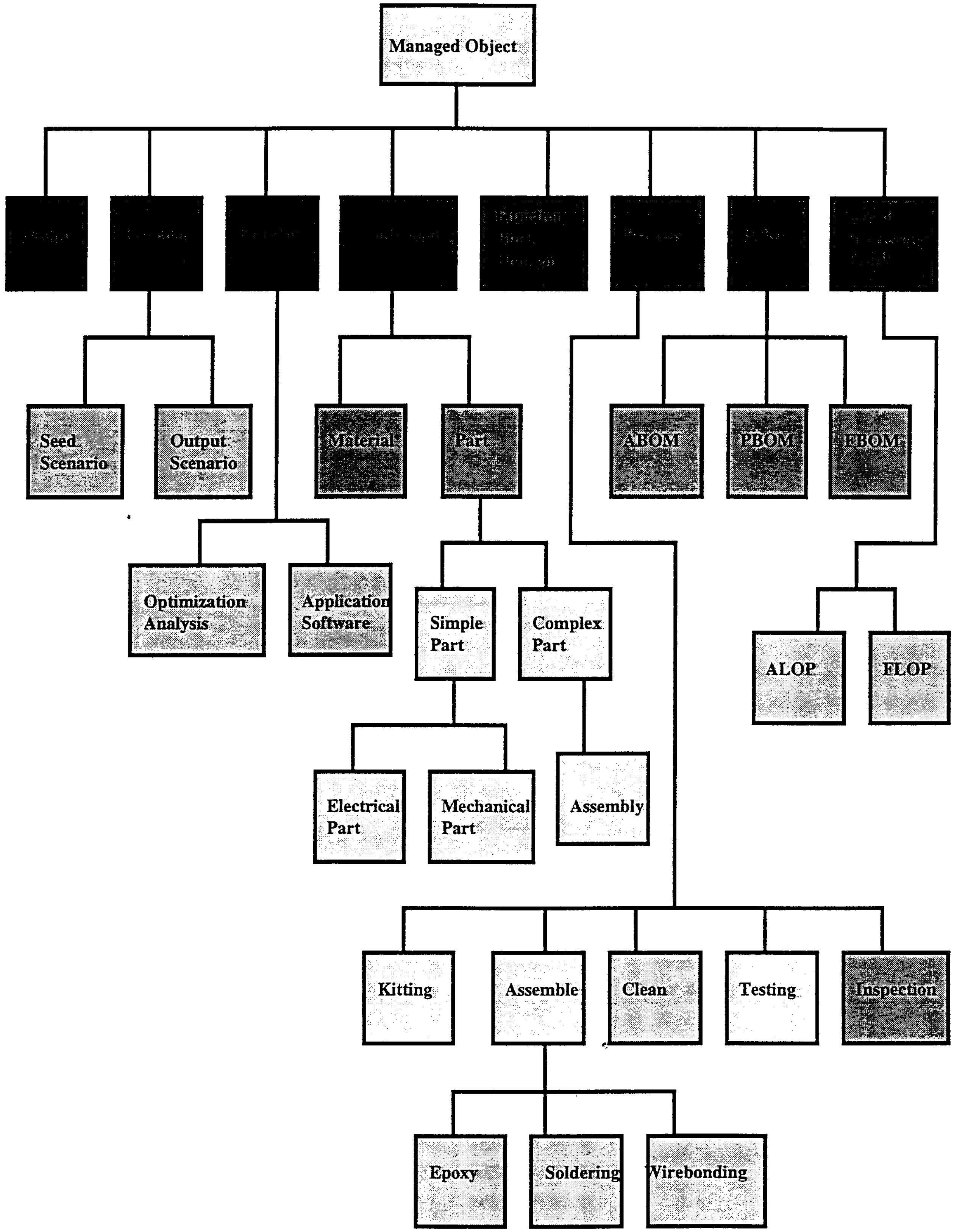


Figure 7: Object Data Model Hierarchy

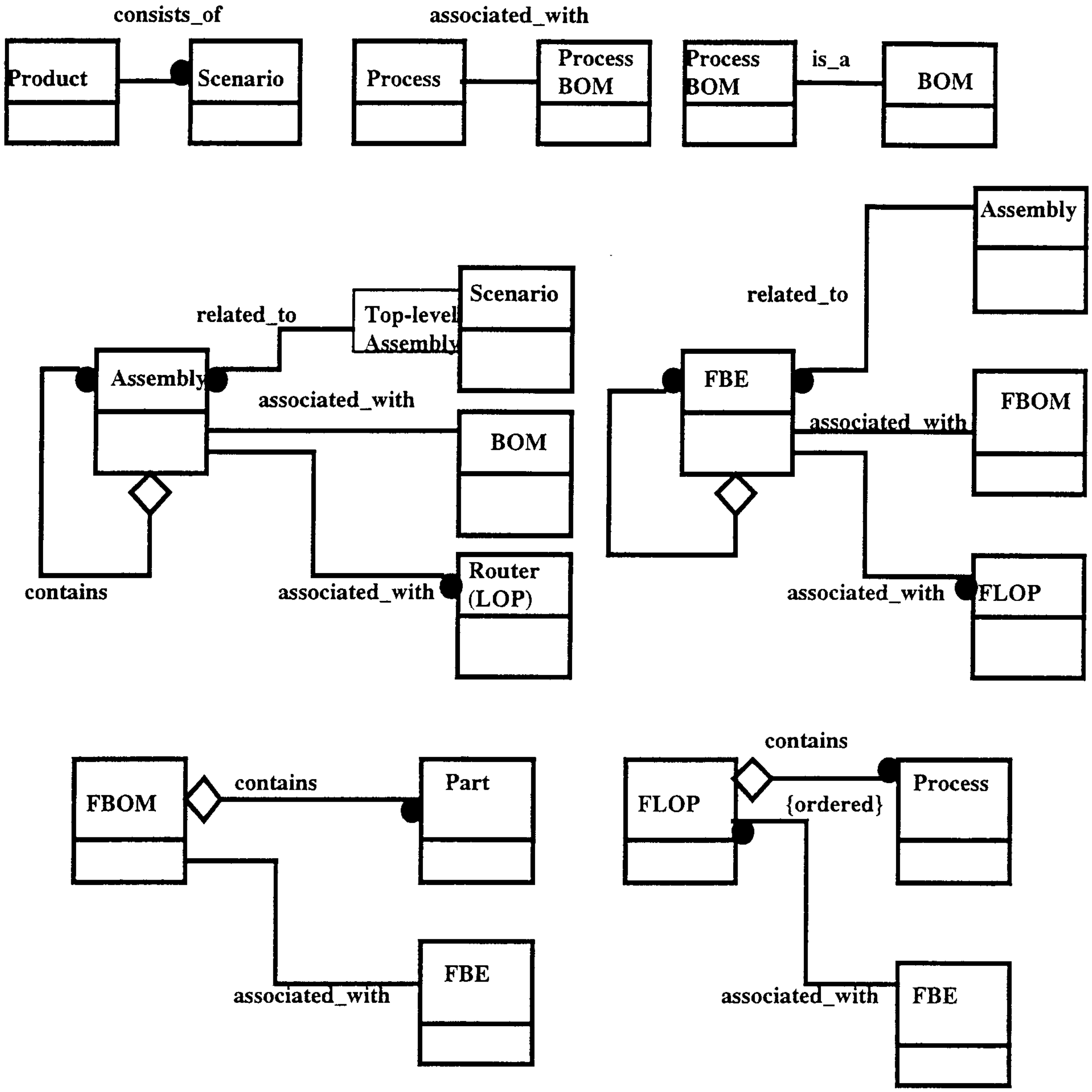


Figure 8: Object Data Model Relationships