Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

AN INTEGRATED ENVIRONMENT FOR MODELING LARGE SCALE ELECTRONICS MANUFACTURING

Michael G. Ketcham
331 Newman Springs Rd., NVC-2Z210
Red Bank, New Jersey 07701

ABSTRACT

Using simulation to model large-scale manufacturing systems raises issues regarding simulation design, data modeling, and software integration that have been investigated during the development of a design environment for continuous flow manufacturing, called SimCFM. SimCFM combines simulation with optimization and cost analysis to design large-scale facilities for electronics manufacturing. The simulation is a modular, parametric model that shares a common data model with mathematical and cost analysis models.

1 INTRODUCTION

Using simulation to model large-scale manufacturing systems raises issues regarding simulation design, data modeling, and software integration that have been investigated during the development of a design environment for continuous flow manufacturing (CFM), called SimCFM.

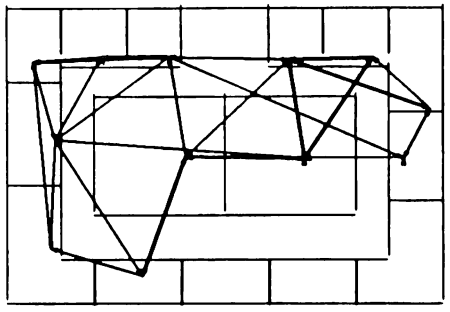
Recently, CFM has been advocated as a way of reducing delays due to product batching and reducing the complexity of production flow [Beal, 1988]. Scheduling CFM systems has led to development of models to optimize job release decisions [Gershwin, Akella, and Choong, 1985] or work-in-process levels [Hopp and Spearman, 1990]. The cost implications and expected behavior of alternative CFM designs, however, are not well understood. Wherever possible, CFM uses work centers dedicated to a single operation, as in a flow-shop; but with expensive or specialized tooling, it uses multiple-purpose, flexible work centers. An important design problem, therefore, is deciding which operations should be allocated to which facilities and how facilities can be configured.

As an example, Figures 1.A and 1.B show two alternative layouts for the same hypothetical sequence of operations. This is a small-scale system, with 40 operations for 1 product type. In Figure 1.A operations are routed between 12 work centers. In Figure 1.B they are routed between 14 work centers. In both layouts, pieces may revisit work centers for multiple operations on the same machine. Alternatively, machines of the same type may be placed at more than one location to improve product flow. The layout in Figure 1.B uses more work centers with lower machine utilization in order to limit the amount of work in process and more closely approximate true continuous flow.

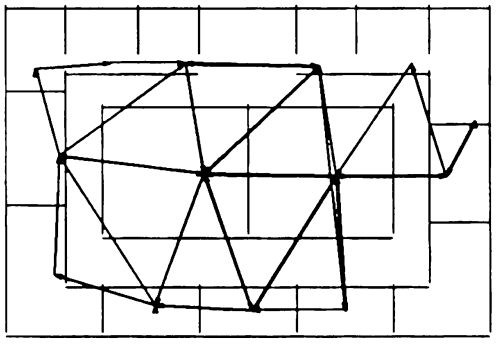
The overall goal of CFM design is to minimize capital investment in machines, which can be accomplished (approximately) by mathematical optimization, together with inventory holding costs for work-in-process, which can be best evaluated by simulation. Typically, capital costs can be minimized using a job-shop layout, whereas inventory levels and product flow can be best controlled with continuous flow layouts.

CFM systems are large enough that a good layout cannot be obtained without extensive experimentation, and an optimal layout cannot be guaranteed for any realistically sized system. The actual systems being modeled include hundreds of operations for multiple products, with additional requirements for setup, batching, and rework between operations. The system designs themselves are volatile. The designs for prototype facilities may change weekly due to changes in production processes. For these systems, engineers need to evaluate quickly the capacity of alternative designs and their capital and operating costs. For existing facilities, product mix, capacity requirements, and processes may change dramatically between planning periods. For existing systems, planners continue to evaluate capacities and costs and must also track discrepancies between planned production (as predicted by the SimCFM simulations) and actual production.

These design requirements imply that engineers



A. Alternative Configuration A



B. Alternative Configuration B

Figure 1. Alternative CFM Layouts

will work with a "library" of alternative configurations, all intended to meet similar production requirements.

2 THE SIMCFM MODELING ENVIRONMENT

SimCFM combines simulation with optimization and cost analysis as part of a complete design cycle for large-scale facilities for electronics manufacturing Given the requirements for CFM system design, the implementation of SimCFM has addressed several issues in using simulation for large-scale factory design. These include:

1. The design of a modular simulation that can be rapidly reconfigured in response to changing system specifications.
2. The development of a mathematical model to generate near-optimal configurations for new manufacturing requirements.
3. The development of a cost analysis module to provide estimated capital and operating costs.
4. The design of manufacturing database to support SimCFM simulations and related software.

An important feature of the database is that it must manage what we have called "multi-dimensional" problems. These are problems that must represent multiple configurations, with alternative production requirements, across multiple time periods.

The overall SimCFM architecture and principal data flows are shown in Figure 2. A typical design cycle follows these steps:

1. The user enters production requirements and machine specifications into the database.
2. The mathematical model for resource assignment reads production requirements and generates a configuration that minimizes the combined cost of acquiring machines and transporting pieces between work centers. The resulting machine assignments and product routings are returned to the database.
3. The user evaluates and modifies candidate solutions. Several possible solutions, including user-generated solutions, may be held in the database, with different design parameters.
4. To simulate one design, the user chooses a candidate solution. A simulation "translator" extracts production specifications from the database and generates a SIMAN experimental frame.
5. The experimental frame is linked with a SIMAN model and the simulation is run. (The model itself uses generic, modular elements that are never altered—as will be explained.)
6. The simulation output report is parsed and statistical results are stored in the database.
7. The user can review simulation results using various pre-programmed displays. These may include, for example, bar charts of the top ten queue lengths (of possibly hundreds of queues) or correlated displays showing maximum machine utilization with a list of operations using those machines.
8. The user enters additional cost data, such as internal rate of return, inventory holding costs, and expected system life.
9. The cost model extracts system specifications and simulation output results to generate an integrated cost estimate.

The rest of the paper will discuss the major modules shown in Figure 2 and their coordination with the simulation. Each section will review the structure of each model, with an emphasis on the

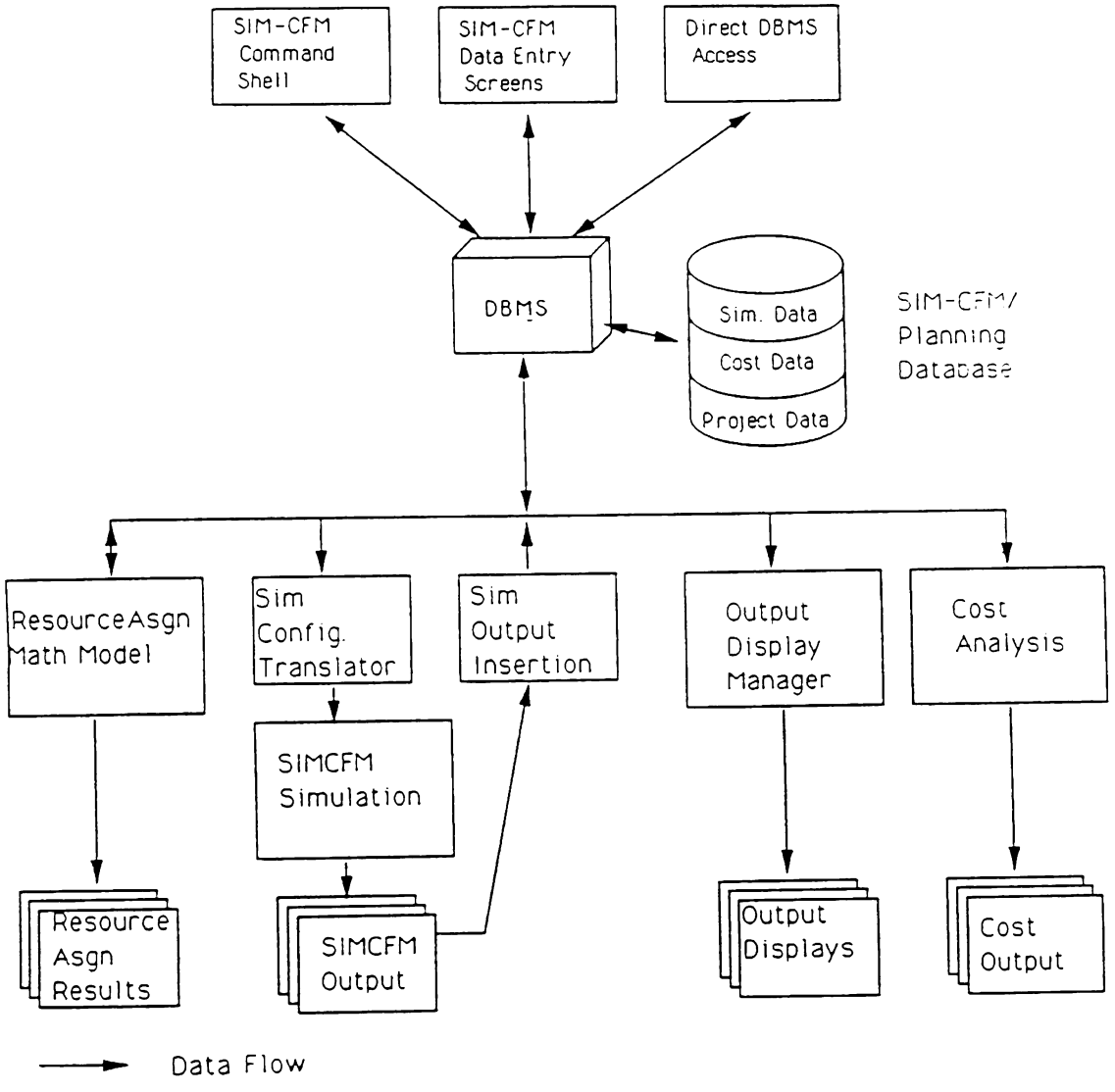


Figure 2. SimCFM Architecture

data requirements that integrate these models through the central database.

3 SIMULATION AND OPTIMIZATION

Figures 1.A and 1.B are alternative solutions to the mathematical model given in Figure 3. The optimization problem takes the form of a large mixed-integer program, which we have called the "configuration problem" (CP) [Ketcham, 1992]. The CP consists of four subproblems:

1. How many work centers are needed to serve a particular sequence of operations?
2. Which operations should be assigned to which work centers?
3. For each work center, which combination of machines will minimize machine costs?
4. Where should work centers be located to minimize cost of transporting pieces?

These subproblems interact in the following way: If the assignment for operation o is changed from facility i to facility j, this change affects the machine acquisition costs at both facilities and affects transport costs with respect to operation $o-1$ and operation $o+1$. At one extreme, all operations belonging to the same "set" of operations will be assigned to a single facility. This yields a job shop layout. At the opposite extreme, each opera-

Min:

$$CP= \sum_{m=1}^{N(m)} \sum_{l=1}^{N(l)} x(m)_{m,l} CM_m +$$
(machine acquisition costs)
$$\sum_{o=1}^{N(o)-1} \sum_{i=1}^{N(l)} \sum_{j=1, j\neq i}^{N(l)} x(o)_{o,i} x(o)_{o+1,j} CT_{i,j} +$$
(transport costs)
$$C_s(N(o)-\sum_{l}\sum_{s} x(l)_{s,l})$$
(operation setup cost)

ST:

$\sum_{o} x_{o,m,l} \lambda_o t_{o,m} \leq x(m)_{m,l} \rho_{max}$, all m, all l
(mchn. capacity cannot be exceeded)

$\sum_{m} x_{o,m,l} = x(o)_{o,l} \lambda_o$, all o, all l
(ops. must be performed completely)

$\sum_{l} x(o)_{o,l} = 1$, all o
(ops. must be performed at one loc.)

$\sum_{s} x(l)_{s,l} \leq 1$, all l
(only one "set" of ops. allowed per loc.)

$x(o)_{s,l} \leq x(l)_{s,l}$, all s, all o in s, all l
(ops. at l must belong to the "set" at l)

$x(m)_{m,l}$ integer ≥ 0

$x(o)_{o,l}$ {0,1}

$x(l)_{s,l}$ {0,1}

$x_{o,m,l} \geq 0$

N(o)	the number of ops.
N(m)	the number of different mchn. types
N(l)	the number of locs.
$x_{o,m,l}$	portion of op. o assigned to mchn. m at loc. l
$x(m)_{m,l}$	number of mchns. of type m at loc. l
$x(o)_{o,l}$	indicates op. o is assigned to loc. l
$x(l)_{s,l}$	indicates "set" s is assigned to loc. l
$CT_{i,j}$	cost of transport from loc. l_i to l_j
CM_m	annualized cost of mchn. type m
C_s	setup penality for multiple ops. at one facility
$t_{o,m}$	time reqd. for op. o on mchn.m
λ_o	arrival rate of parts into op. o
ρ_{max}	max. acceptable mchn. utilization

Figure 3. The CFM Configuration Problem

ation will have a dedicated facility, which yields a flow-shop layout. An optimal configuration usually lies somewhere in between.

This formulation assumes that operations fall into "sets" of similar operations that can be performed on the same types of machines (such as "etching" or "baking" operations). Operations belonging to the same "set" can be assigned to the same flexible facility, but operations belonging to different sets cannot be. Constraints (7) and (8) keep operations in separate sets.

The parameter ρ_{max}, in Figure 3, specifies the maximum acceptable machine utilization. The parameter C_s is a largely qualitative cost penalty for assigning several operations to the same work center. Figure 1.A solves with ρ_{max}=0.85 and C_s=0. Figure 1.B solves with ρ_{max}=0.75 and C_s=1, so the configuration in 1.B should give lower inventory levels and a continuous flow.

When these systems are simulated, the simulation represents control at the sector level and control at the operation level. A sector is a sequential grouping of operations. As an example, the first sector may represent operations that prepare a surface, and the second sector may represent operations that deposit the first conductive layer on a semiconductor wafer. Work-in-process limits control inventory levels within a sector. For example, production engineers may specify that no more than 60 pieces for product "A" can be in process in sector 2. If work-in-process reaches the specified level, new pieces must wait in sector 1 until a job-release signal allows them to move forward. The sector control logic is responsible for managing both new and reworked pieces.

When a piece arrives at a work center, processing is controlled by operation-level controls. Operations can be performed on two broad classes of machines. "Continuous" machines include ovens or cleaning baths. Pieces can enter the machine whenever the pieces are ready, up to the capacity of the machine. With "batch" machines, pieces wait in the normal way until previous pieces are finished. Despite their names, both types of machines may in fact work on batches (or lots) of products, such as a "boat" holding several pieces for a cleaning bath, or an arrangement of identical pieces in an etching machine. In all, the operation-level controls include six types of control decisions based on operation type, machine type, the loading of machines, work-in-process, and batch sizes.

In developing SimCFM, we faced several options for changing a simulation to model changing sys-

tem configurations:

1. The simulation could be re-written for each configuration. This was the approach used before SimCFM, but it required a long development time. Models typically lagged several weeks behind changes to the prototype facilities they were expected to evaluate.
2. A model generator could automatically create models. The logic for a model generator, however, was more complex than was necessary. Each model would also need to be newly translated, a time-consuming process for large models.
3. The simulation could read system specifications directly from the database as it runs. Research by Ketcham, Shannon, and Hogg [1989] has called this approach "information-based modeling." Information-based modeling is still largely a research tool and lacks the open architecture provided by a commercially available SQL database.
4. The simulation could use a parametric model, or "core model," which would represent generic work centers and generic system controls [Ketcham and Watt, 1989; Ketcham, 1988].

SimCFM uses a "core modeling" approach [Ketcham and Krishnaswamy, 1988]. The model is constructed from eight modules programmed in SIMAN which represent both the physical layout of the system and the system control logic. The "core" model does not contain any system-specific information. Instead, it models only the generic manufacturing logic common to a whole class of CFM systems. Details about machine assignments, work center locations, product routing, rework requirements, and other controls are entered in the SIMAN experimental frame, which is generated for each new layout.

The modular structure for the simulation is shown in Figure 4. Figure 4 shows sector-level and operation-level controls. Additional modules for *scrap* and *exit* manage the disposition of finished pieces. By permanently retaining the generic logic for CFM systems, the "core" model eliminates the need for re-programming when modeling new configurations. Instead, the user can define new configurations simply by entering specifications in the SimCFM database, following the data requirements shown in Table 1. Table 1 shows values as inputs into or outputs from the three principal planning models.

The simulation and optimization models share overlapping data requirements, as we see in Table 1, but they provide complementary capabilities in evaluating system designs. The simulation, of course, is not an optimizing tool. Layouts such as those in Figures 1.A and 1.B represent only two of several hundred feasible layouts, each of which may take hours to evaluate through simulation. The mathematical model is solved based on mean values, without considering distributions or the dynamic interactions introduced by the system control logic. Inventory policies, operation-level control policies, probabilistic interactions between parts at different stages of production, rework, and machine failures, and the effects of variable parts arrivals and variable processing times can only be captured by the simulation.

4 SIMULATION AND COST ANALYSIS

Like the optimization model, the cost analysis compensates for limitations in the simulation. Conventional simulation outputs, such as queue lengths, resource utilization, and product cycle times are inadequate for comparing system designs. The volume of data is too large to easily assimilate, and it is extremely difficult for designers to translate raw performance data, such as queue lengths, into a common basis of comparison: The systems under study may have hundreds of queues; the causes of excessive queue lengths may include inadequate machine capacity, poorly planned parts routings, inefficient parts batching rules, or inadequate policies for managing rework; and the consequences of excessive inventory will differ at different stages of production.

The SimCFM database manages large data sets through preprogrammed database queries and displays. Database queries can easily make correlations that are difficult for an analyst. Queueing data, for example, can be tabulated together with information about the machines that serve that queue and the (possibly many) operations that feed it.

Like these preprogrammed data displays, the cost analysis reads simulation results from the database, together with system specifications from the mathematical optimization. It combines results into a single cost calculation based on capital and operating costs. The structure of the cost model is shown in Figure 5. In computing inventory holding costs, the cost model reconstructs the value added per piece, per sector, and computes the inventory holding cost, per sector, as: $hT(v_e+v_a/2)$, where h is the inventory holding cost,

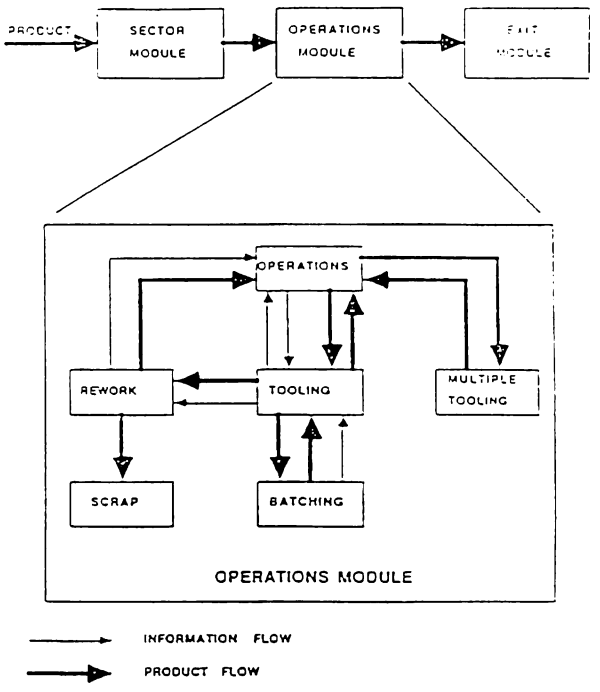


Figure 4. Structure of the "Core" Model

T is the sector cycle time, v_e is the value of the piece when it enters the sector, and v_a is value added within the sector. Value added, including losses due to reworked and scrapped parts, is converted into "equivalent finished units" to give a common basis for throughput and cost comparisons [Dalo, 1989].

5 SIMULATION AND PRODUCTION DATABASE DESIGN

Table 1 shows the principal data items and their role in system design. Figures 6 and 7 show two portions of the same data model in the form of a data-dependency diagram, which indicates the logical relationships among data objects. Figure 6 shows the SimCFM data model used to represent system configurations. Figure 7 shows the data model that supports alternative models for "multi-dimensional" problems. In Figure 7, the *System* table holds comparatively static cost parameters, such as direct labor costs, costs for floor space, and internal rate of return. The *System* can be manifested in multiple *Production Plans*, where plans differ based on either planning parameters that will change from period to period, or policy and control alternatives. *Production Plans* are also used to represent alternative process plans, particularly when SimCFM is used to design prototype facilities. Each *Production Plan* can be modeled by several alternative models. Results from each model, such as statistical results from multiple replications, are all managed by the central database.

This data model satisfies the requirements for supporting the CFM planning cycle that we found in the Introduction:

1. Data can be managed and queried for very large databases.
2. Choosing a relational database provides standardized SQL commands for data queries, and provides forms-based applications for rapid data entry.
3. Data are maintained for multiple alternative designs.
4. Data supports the complete design cycle, even though the SimCFM simulation by itself uses only a small portion of the data model.
5. Data are defined in a single "canonical" form so that models can be maintained independently but still share data. The database provides a software linkage between models of different types, which have different data requirements and which may be written in different languages. The database, therefore, isolates the other planning models from the limited data structures which the SIMAN simulation would otherwise impose.

6 CONCLUSION

Integrating models in this way requires the software designers to consider three new issues, in addition to the design of simulation itself.

1. The designers need to identify shared data requirements for multiple models and need to develop a data model which can be converted into a working database.
2. They must think of models in terms of reusable, generic modules, keeping in mind that simulations may be mused (within a few months, or less) for new system configurations that will be very different from those the designers originally encountered.
3. Model integration requires writing supporting programs that will extract and reformat data from a shared database.

The benefits of integration have proven to be well worth the effort, in the case of SimCFM and similar integrated modeling environments.

Table 1. Data Requirements

Variable	Opt.	Sim.	Cost
Machine costs	→ in		→ in
Maximum batch size per machine	→ in	→ in	
Machine time-to-failure and time-to-repair	→ in	→ in	
Machine floor space requirements	→ in		
Machine life and salvage value			→ in
Worker support required per machine	→ in		
Sector to which operations are assigned		→ in	
Processing time required, per operation, per machine	→ in	→ in	
X- and Y- coordinates for floor space locations	→ in		
Floor space available at each location	→ in		→ in
Material handling costs per unit distance	→ in		→ in
Material handling time per unit distance		→ in	
Work center locations	out →		
Product routings between work centers	out →	→ in	
Machine assignments to work centers	out →	→ in	→ in
Workers required	out →	→ in	→ in
Arrival rate of jobs	→ in	→ in	
Thoughput of jobs		out →	→ in
Work in process, per sector		out →	→ in
Scrap, per sector		out →	→ in
Shift schedules	→ in	→ in	→ in
Labor costs	→ in		in
Floor space costs	→ in		→ in
Cost of setting up a work center	→ in		→ in
Annualized capital costs			out →
Annualized operating costs			out →

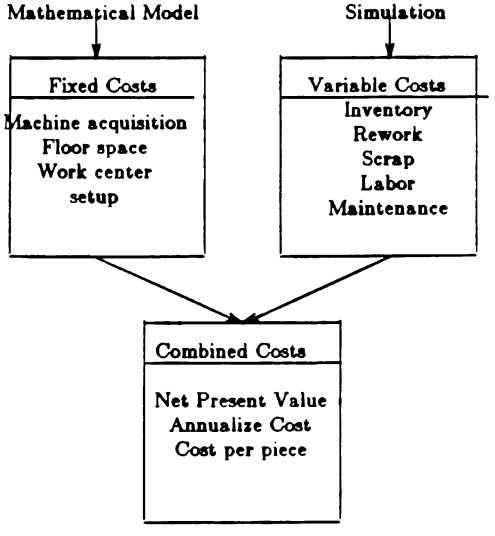


Figure 5. Structure of the Cost Model

1. Users are able to "leverage" model results: they can use the results from one model as inputs to other models, and can use models jointly to examine different characteristics of system performance.
2. In an integrated environment, analysts can use one model as a verification for other models.
3. Most importantly, integrated environments provide a platform from which both system designs and models can evolve. The data independence provided by the database allows the data model to evolve and new models to be added to the planning environment with only minimum changes to existing models.

The SimCFM environment begins to address the problem of multi-dimensional systems designs that we saw in the Introduction. Multi-dimensional planning systems must manage alternative designs for multiple products, across several planning periods. In addition to multiple plans, however, the modeling tools themselves will have multiple dimensions. The last ten years have seen a rapid evolution in manufacturing control strategies (such

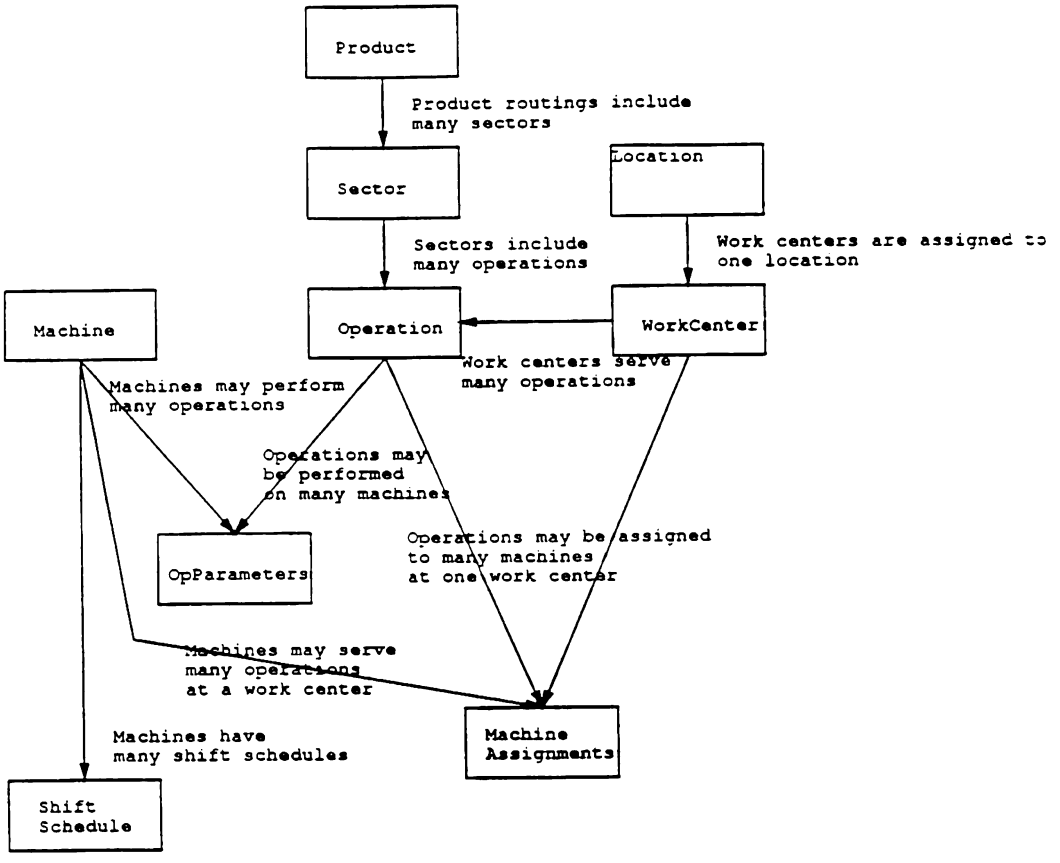


Figure 6. Manufacturing Data Model

as CFM) and a rapid evolution in modeling techniques (such as generic simulations). We can expect to see this evolution continue, with models communicating by way of shared databases and integrated software environments. Rather than writing and rewriting new models to accommodate each change in the manufacturing environment, flexible software environments comprised of multiple models will allow users to select models with different solution characteristics, range of applicability, solution speed and accuracy, or measures of performance.

ACKNOWLEDGMENTS

This research was supported by the Glass Ceramic Substrate Manufacturing Group, IBM, East Fishkill, NY, while the author was at the Dept. of Industrial Engineering Operations Research, University of Massachusetts, Amherst.

REFERENCES

Beal, K. 1988. "Integrated material logistics and continuous flow manufacturing." *International Journal of Production Research*, 26:3, 351-373.

Dalo, C. 1989. "An Integrated Cost Model for Continuous Flow Manufacturing." Unpublished master's thesis, Department of Industrial Engineering and Operations Research, University of Massachusetts, Amherst.

Gershwin, S. B., Akella, R., and Choong, Y. F. 1985. "Short-term production scheduling of an automated manufacturing facility." *IBM Journal of Research and Development*, 29:4, 392-400.

Hopp, W. J., Spearman, M. L, and Woodruff, D. L. 1990. "Practical Strategies for Lead Time Reduction." *Manufacturing Review*, 3:2, 78-84.

Ketcham, M. G. 1988. "A Core Modeling Ap-

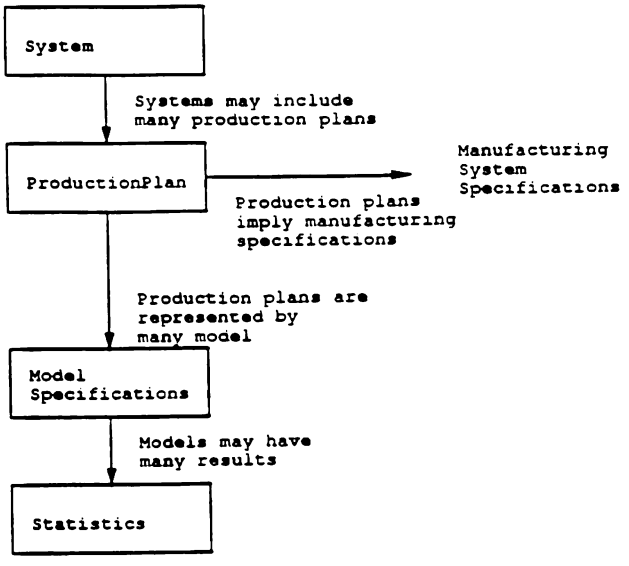


Figure 7. Planning Data Model

proach to Manufacturing Simulation." *Proceedings of the Seventh Annual Control Engineering Conference.* Des Plaines, IL: Control Engineering, 1988. Pp. XII-1 to XII-10.

Ketcham, M. G. 1992. "A Branch and Bound Approach to Facility Design for Continuous Flow Manufacturing Systems." *International Journal of Production Research*, vol 30, no 3, 573-597.

Ketcham, M. G. and Krishnaswamy, C. 1988. "A Generic Simulator for Continuous Flow Manufacturing." *Proceedings of the 1988 Winter Simulation Conference*, San Diego, CP., Dec. 1988, ed. M. A. Abrams, P. L. Haigh, and J. C. Comfort. San Diego, CP.: Society for Computer Simulation. Pp. 609-615.

Ketcham, M. G., Shannon, R. E., and Hogg, G. L. 1989. "Information Structures for Simulation Modeling of Manufacturing Systems." *Simulation*, vol. 52, no. 2 (Feb. 1989), 59-67.

Ketcham, M. G., and Watt, J. 1989. "Parametric Simulation for Flexible Assembly Systems." *Journal of Manufacturing Systems*, vol. 8, no. 2, 115-125.

AUTHOR BIOGRAPHY

MICHAEL G. KETCHAM is a member of the Network Reliability Group at Bell Communications Research, Navesink Research Center, 331 Newman Springs Rd., NVC-2Z201, Red Bank, NJ, 07701. He received his Ph.D. in 1986 from Texas A&M University, and from 1984 to 1992 was on the faculty in the Dept. of Industrial Engineering & Operations Research, University of Massachusetts, Amherst. His interests are in integrated modeling environments, combining simulation, analytical modeling, artificial intelligence, and databases for analysis of large-scale systems.