Proceedings of the 1990 Winter Simulation Conference
Osman Balci, Randall P. Sadowski, Richard E. Nance (eds.)

THE REPRESENTATION OF ACCOUNTING DATA IN INDUSTRIAL SIMULATION

Craig McLanahan

School of Business
Salem State College
Salem, Massachusetts 01970

Michael G. Ketcham

Department of Industrial Engineering
and Operations Research
University of Massachusetts, Amherst
Amherst, Massachusetts 01003

ABSTRACT

This paper examines the use of accounting data, particularly cost and investment figures, in manufacturing simulations. It briefly reviews current accounting thought and practices, as relevant to simulation, and describes a modeling approach that accommodates these practices. It describes a layered simulation where the information flow carrying cost data is overlaid on top of the dominant production flow. This layering allows a modeler to represent dynamic cost behavior, aggregation and allocation of costs, and generation of cost reports for virtually any accounting practice. It has the additional benefit, as well, of focusing attention on the information system needed to get the "correct" cost information for managing the manufacturing enterprise.

1. INTRODUCTION

This paper examines the use of accounting data, particularly cost and investment figures, in manufacturing simulations. We have two motivations for looking at this problem. First, many newer manufacturing control strategies, such as total quality control (TQC), just-in-time (JIT), and flexible manufacturing systems (FMS), affect production systems in ways that cannot be measured by analysis of capital investment or traditional cost accounting. Second, current thinking about manufacturing accounting, as proposed by H. T. Johnson, R. S. Kaplan, and others, focuses attention on dynamic system activities, rather than on allocating costs across broadly aggregated cost categories (eg. [Johnson and Kaplan 1987]). In both cases, simulation is an ideal tool for predicting the cost behavior of industrial systems.

Most simulations, however, use very simple methods of accumulating cost information which rarely represent realistic accounting methods. Moreover, accounting methods, and consequently the meaning ascribed to cost figures, changes radically from one environment to another, so that adding a single set of cost functions to a simulation language is unlikely to give satisfactory results.

To address these issues, we:

(1) look at current accounting thought and practices, as relevant to simulation;
(2) briefly review the computation of costs, return on investment (ROI), and other accounting parameters as they are commonly used in simulations;
(3) recommend a strategy for incorporating accounting parameters into simulations; and,
(4) describe a preliminary model where generation and tracking of various forms of accounting data are incorporated.

We describe a layered simulation where the information flow carrying cost data is overlaid on top of the dominant production flow. This layering allows a modeler to represent dynamic cost behavior, aggregation and allocation of costs, and generation of cost reports for virtually any accounting practice. It has the additional benefit, as well, of focusing attention on the information system needed to get the "correct" cost information for managing the manufacturing enterprise. The simulation of information systems, in itself, is potentially a powerful applications area.

2. CURRENT ACCOUNTING THOUGHT, PRACTICES, AND PROBLEMS

In a recent survey article, Kaplan [1989] makes a series of historical observations:

(1) Existing management accounting systems are obsolete, and have in fact contributed to U.S. industrial decline.
(2) Business patterns among large companies, which have tended toward the creation and sale of more and more varied products, have undermined accounting systems designed for relatively few major product streams.
(3) The philosophy of decision making and control has passed from a centralized "one best way" to more participatory judgments made at operating levels.
(4) The introduction of new manufacturing strategies, such as TQC and JIT, have expedited the decline of relevance of management accounting because they have tended to cut the historical reliance of past manufacturing activities on inventories and direct labor. Older management accounting systems focus on inventory valuation as a main activity. Direct labor components of costs used to be much larger for both the product stream and its associated inventories.
(5) The technology of information collection and management has opened up the opportunity to collect, process, and disseminate information in new ways, and people are experimenting with it. Specifically, management accounting does not have to be slaved to a monthly or annual financial accounting cycle.

He continues with observations about the major goals of the new systems he foresees:

(6) The aim of management accounting systems should always be to measure operating performance. In addition to accounting figures, the new systems must handle key physical parameters. They must present data on Q.C. results, process yields, resource consumption, and other key determinants of economic performance. There must also be direct attribution of resources consumed to value created; thereby avoiding aggregation of costs into amorphous cost pools which are redistributed by allocation back over all activities.
(7) All activity relevant to either a customer order or a "package" of product volume should be collected and

assigned to that order or product. This involves such activities as order entry, special design modifications, special processing arrangements, and any other attributable work. The main objective is to create a cost model of the enterprise and to provide a strong incentive to make minimum use of support services. This goal includes the requirement to capture as direct costs all of the "activities" that are necessary for each product or process. This, of course, requires knowledge about what drives these activity costs.

Johnson and Kaplan's critique of current accounting procedures [1987, p. 1] combines these observations in a telling phrase:

> Today's management accounting information, driven by the procedures and cycle of the organization's financial reporting system, is too late, too aggregated, and too distorted to be relevant for managers' planning and control systems.

2.1 The Goals of Accounting Systems

Accounting is, in essence, a process of sorting and summing transaction attributes that have to do with economic results. In putting together systems that will track these attributes, accountants have had to make many choices about how to represent costs. These choices have generally been guided by two influences, (1) the "fair" representation of an organization's financial situation to the outside public, and (2) the necessity to minimize the cost of the system that comes up with the figures. Lower priority has been given to the internal accuracy of costs for planning and control purposes. Before the advent of low cost computing power the second factor weighed heavily on the choice of system to use, and current systems reflect these past choices. Often in frustration, line managers have developed their own systems to produce the numbers that they need for internal planning and control, and some of these apparently work well.

There is a need for at least three different but related types of accounting data:

(1) Product Costing data which reflects the direct costs of making the product;
(2) Inventory Valuation data which allows the allocation of profits to defined time periods and tries to "fully value" each manufactured item (as for annual taxes);
(3) Financial Reporting data which is made public and used by outsiders in the valuation of the company.

Product costing should deal only with the direct resource inputs to the product, saving allocated charges for inventory valuation purposes. Many costs, such as utility costs or the costs of acquiring new technology, that are currently allocated, should be tracked more closely so that they can be directly assigned to the products that generate those costs. For example, it may be necessary to more accurately estimate the unit product use of electric power or to meter individual processes to get at real production costs.

Inventory valuation (that is, the value of units of product in inventory) will need to include allocated cost, but the allocations should be made causally to the maximum possible extent, to be as "fair" as possible to each product. Allocated costs include overhead costs that cannot be directly assigned to a specific product, but that are apportioned across all products, usually according to some ratio based on direct costs. There are significant problems with allocation formulas, however, that are made worse by the dynamic changes being made in traditional allocation bases such as direct labor or plant space. As efficiencies increase, direct labor often decreases causing unpredictable and unfair shifts in overhead absorption in current systems.

Lastly, the figures for financial reporting, while necessary outputs of accounting, should not be relied upon as indicators of internal costs since they are rarely in sufficient detail for effective analysis. that take place in current accounting methodologies.

Another problem with some manufacturing accounting systems is the way costs are aggregated and categorized between production steps. The simpler systems total all costs at each step and use the resulting figure as a material input to the following step. Thus, at the end of the processing steps only the labor cost value of the last step is recorded as labor and the material cost is grossly overstated since it contains labor, overhead, and material charges from earlier steps. The more accurate systems preserve the integrity of the direct labor and material charges as they are built up from process to process. Further, complex systems use intermediate "cost centers" to assemble costs which are then shared out among products that use those centers. To understand how they do this, it is necessary to know the exact path of cost charges and what cost elements they contain for each product. The complex cost buildup methods are illustrated in Figure 1.

2.2 Accounting Issues and Simulation Approaches

Why do these distinctions in accounting methods have a place in a discussion of simulation methods? We see three interconnections.

First, in order to put useful cost and accounting features into simulation models, it is necessary to have some appreciation of the issues faced when trying to account for expenses and the value of assets, and some knowledge of the techniques that accountants and cost analysts use. If these can be depicted in the simulation design, the results may be more realistic, more readable, and more useful.

Second, simulation provides a testbed for evaluating the impact of new costing methods. The proposals made by Kaplan and others mean a radical redesign of accounting and reporting systems. Implementation could mean that the directly attributable "direct" costs on some products will shoot upwards while others decline. Overheads will contract to unheard of low levels. A new category of "activity costs" may be introduced, causing major swings in total product costs to appear. Hitherto profitable lines and activities could become unprofitable, and vice versa. Managerial prudence would indicate that, before so radical a set of changes is introduced, any new system should be evaluated "off line" to provide experience as to how it responds and "feels." Simulation provides a way of doing this, and such a simulation would need to include proposals for new information systems designs.

Third, accurately representing accounting procedures will place demands on simulation models that cannot be effectively met by commonly used approaches to modeling costs.

3. REPRESENTING ACCOUNTING DATA IN CURRENT SIMULATIONS

While traditional simulations have dealt with physical quantities (utilizations, counts, flow rates), they have not dealt in a major way with costs. Some models have had cost parameters backfitted onto them, but the ability to handle the necessary transaction processing and output displays associated with accounting systems is not readily available. For a simulation to be helpful

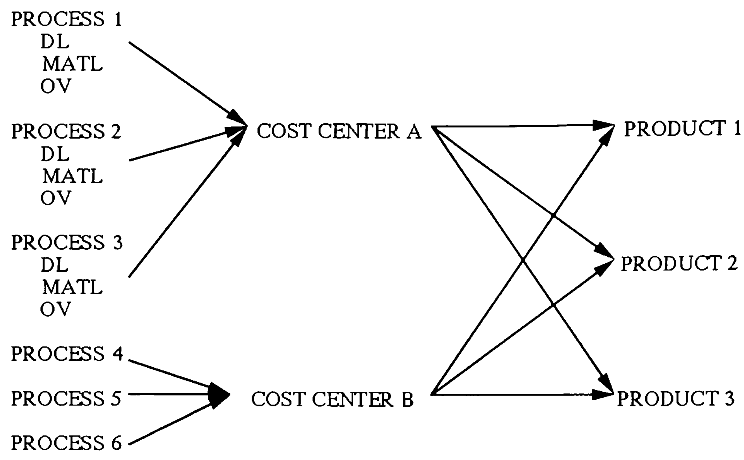

Figure 1. Complex Cost Buildup

in the sense that Kaplan suggests, it must trace information flows, mimic the data processing activities (including any time delays), be configurable in a wide variety of ways to reflect alternative proposals, and present output data that resembles the output of real systems.

Few simulation languages have made specific provisions for including cost and investment parameters in the language itself, and those that do (for example, GEMS II [GEMS-II 1987], XCELL+ [Conway et al. 1987], and MIC-SIM [Strack 1988]) have rudimentary computations that rarely represent realistic accounting practices.

General purpose simulation languages (including GPSS, SLAM II, and SIMAN, among others) provide a number of alternatives for representing cost information. One of the simplest approaches is to overlay cost figures onto the operating results, after the fact. That is, costs associated with machine utilization, inventory costs associated with work-in-process, and revenues from production throughput are allocated to products based on aggregated statistical results found at the end of the simulation run.

Alternatively, costs can be carried and accumulated by entity attributes that are adjusted each time a cost is incurred, and revenues from sales can be accumulated as products pass through a Sales station. At the end of the product flow, these attributes are processed to collect aggregated cost figures The accumulation of cost information is shown schematically in Figure 2. Although global variables can be used to record the investment in assets, and can be incremented, when necessary, by any process, it is rarely easy to assign investment asset values individually to any particular product, station, or cost category. Further, the computations of cost figures are typically tied to the production flow for the main product or service rather than having separate flows representing the collection and aggregation of cost information, without complex coding.

Where a simulation language does not provide routines that can be impressed into generating and accumulating accounting figures, it may be able to call subroutines written in some underlying language, such as Fortran, but this requires programming skill that may not be available, and may make it difficult to redesign the simulation for changing configurations.

Watson (1981) describes competitive gaming simulations, such as (i) financial models used to project financial implications of decisions, actions, and events; (ii) marketing models used to demonstrate the effects of price elasticity, forecasting methods, etc; and (iii) production models, used to generate operating costs and costs of goods sold, and for exercising scheduling skills. These training simulations are not different in concept from analytical simulations used to evaluate manufacturing systems and policies, except that more branching is needed to accommodate the variety of decisions that can be made by competing players, the models seldom include detailed system configurations or control policies, and less use is made of stochastically generated values, such as delays.

Since cost and other accounting data are routinely used in gaming simulations, the problem with using accounting values in predictive simulations seems to be more one of figuring out how best to implement them. What has possibly held back capabilities for representing costs, investment, and other accounting values is more likely the abundance of alternatives for formulating these values and the lack of an overall strategy for processing and presenting them. As we have seen, the accounting profession and its authors do not help with this dilemma because ac-

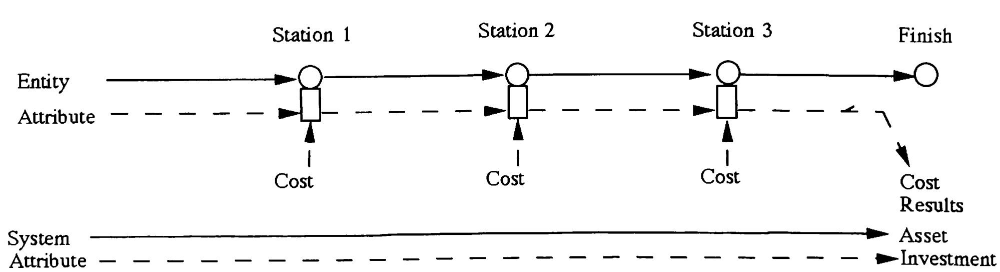

Figure 2. Accounting Data: Current Method

counting is itself undergoing some introspection and change in its approaches to cost and managerial accounting.

4. SUGGESTED METHODOLOGY

It seems apparent from the complexity of accounting data collection systems, from the current lively debate among accountants themselves, and from the many possible uses to which simulation analyses may be put that there is no single best way to structure the generation and analyses of accounting data in simulations. Instead of pointing in one single direction, the data seems to suggest that as flexible a structure as possible should be used.

Those simulation projects that can benefit from the use of accounting data on costs, investments, sales revenues, etc. will certainly need it for different purposes, from project to project. For new product lines, the generation of sales revenue and cost allocations for inventory valuation are likely to be useful. For projects aimed at evaluating the investment of capital in new machinery, the use of asset figures and variable unit costs will be called for - to construct IRR or ROI estimates. For service organizations, the generation of required investments and operating costs to achieve targeted service levels may be necessary.

This section lays out a scheme for adding the ability to generate and analyze accounting data in a wide range of formats within a simulation, rather than merely overlay the dollar figures after the analysis of operating characteristics is complete. The emphasis is on flexibility of choice for the accounting parameters chosen and ease of programming and presentation of the data once it is available.

In principle, the concepts presented are based upon the existence of both a product flow and an information flow which, as noted by Schmenner [1987], are separate from each other in virtually every organization. What is most desirable is the ability of the simulation to represent any accounting system and present figures that might be encountered in the corresponding real world situation. This would include, at the very least, the ability to:

(1) generate and accumulate unit product costs as products flow through the simulation, using either fixed per unit charges or time variable costs for each product;
(2) segregate costs of two or more products or product lines, if they are included in the simulation;
(3) carry the investment in assets for each asset and for the whole group included in the simulation, together with the ability to increase or decrease the asset valuations during the simulation runs;
(4) accommodate a wide range of possible bases for the collection and subsequent allocation of charges such as utility costs, depreciation, rents, interest;
(5) follow closely virtually any cost accounting scheme in current use, in order to "match" figures with it for comparative purposes.

To do this it seems necessary to rely on only two major design principles. First, it is necessary to design the simulation so that a companion but separate "information system" collects data while the product or service flows through. Second, the information system needs to be able to sort, analyze, and present the results provided by the information system. Because cost collection is not dependent upon the flow of the product, it can be tailored to produce almost any result desired to better reflect the "real world" environment being simulated.

4.1 A Representative Simulation Program with Accounting Data

The model used to test this approach is a simple production system with one work station, surrounded by additional information processing stations designed to collect and process cost and inventory data. This model is not intended to replicate any specific "real world" system, although its similarity to a small repair shop, medical office, fast food restaurant, or other service enterprise is nevertheless apparent. The model is intended primarily to provide a simple prototype production system which can be used to demonstrate the feasibility of working with cost accounting data in a simulation.

The model is implemented in the IBIS simulation language developed at the University of Massachusetts, Amherst [Ketcham et al. 1989].

The sequence of operations is straight-forward. A Customer places an order at a customer service counter called the OrderDesk, and the resulting WorkOrder is then transmitted to the Assembly station where the single assembler is assigned as soon as possible. Once the assembler finishes, a Product is sent on to the Checkout station where it is delivered to the waiting customer, once the associated paperwork and customer can be matched up.

A schematic diagram appears in Figure 3. The IBIS input data file is shown in the Appendix. Both stations and entities have descriptive attributes attached to them, which may be used for statistics collection. Attributes can also be initialized based on a computational formula. IBIS route statements specify the flow of entities between work centers, and the ProcessStep statements specify the sequence of operations within a work center. In addition to specifying the sequence of operations, ProcessStep entries may contain expressions for calculating mathematical formulas, as can be seen in the Controller station, where most of the cost and other accounting data is processed. For example, the expression

'FADD("STATION:*;FinGdsVal",FVAL("*;BillTime")*0.4)'

says that, to the value of the Finished Goods at the current station should be added the value of BillTime carried by the current information entity, after it is multiplied by the costing rate of 0.40 (or, in this simulation, $0.40 per minute of Billing clerk time).

The information system collects and processes information from the OrderEntry, Assembly, and Checkout operations. First, the time required for OrderInput is sent to the Controller, by way of an information entity called OrderAdminData. The Controller uses this information to increment the administration costs (AdminCost) due to OrderInput. Next, the time required for assembly is recorded and sent to Billing as an attribute of the information entity PdCostData. At Billing, product cost information is processed in several ways. An Invoice is created that prices out the product and also calculates its cost, based upon assembly time plus a fixed factor. The formulas for cost and price are incorporated directly into the IBIS model. This billing process takes a time delay. When billing is complete, an invoice is dispatched to the Checkout station. Concurrently with the creation of the invoice, the time required for billing is recorded and an information entity is created to carry both BillTime and PdCost to the Controller for further processing. At the Controller station BillTime is converted to a second increment for AdminCost; and PdCost (which has already been computed at Billing) is added to the Finished Goods Inventory (FinGdsVal), indicating the completion, but not delivery, of another Product.

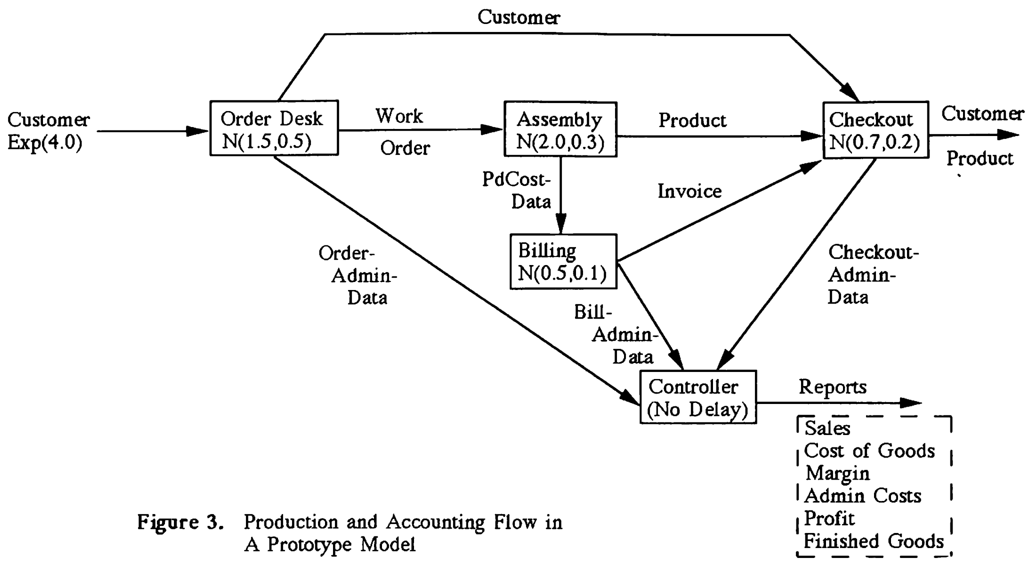

Figure 3. Production and Accounting Flow in A Prototype Model

When a product is delivered, an information entity called CkoutAdminData, holding CkoutTime, PdCost, and PdPrice, is sent to the Controller to complete the accounting cycle. At the Controller, the checkout processing time is converted to a cost and added to the AdminCost; the value of finished goods inventory is decremented by the value of PdCost to reflect delivery of the Product; and the value of Sales is incremented by the value of PdPrice to reflect the sale. Sales and Cost of Goods are thus running totals of the transactions during each replication. Accounting is completed when the Controller computes the margin (that is, the difference between the accumulated Sales and the accumulated cost of goods) and the Profit (the difference between the Margin and the AdminCost).

4.2 The Simulated Accounting System

Certain features stand out in this model.

First, information flow is separated from the production flow. New entities, called information entities in this example, carry accounting information to its proper destination in the information system. This leads to a schematic representation of information flow like that shown in Figure 4.

By allowing the information on costs and other accounting data to flow separately from the product flow, the model can be adapted to model the information system and costing categories that match a current "real world" accounting system.

Second, the actual income statement from operations can be constructed directly from values collected and reported by the simulation run. The balance sheet item for Finished Goods Value is also available and there would be nothing to prevent this simulation, with appropriate modifications, from creating the other balance sheet items necessary to present a complete financial picture of the enterprise. Sales, Cost of Goods, and Administrative Costs are accumulated as each replication develops, while Margin and Profit are derived figures based upon the running totals of the others. Finished Goods is the current inventory of completed product awaiting delivery, kept on a running basis as the replication develops. Thus, embedded in the running of this model, is a simplified accounting system that operates in "real time."

Third, attributes have been assigned locally to the stations to record asset valuation data and to allow the incrementing or decrementing of the accounts used in the system. Local attributes can be similarly attached to resources, to queues, and other objects in the system to more completely represent the economic features of a "real world" system.

More than one information entity or accounting attribute can be used, when necessary, to carry accounting data for each station or entity involved. For example, if it were desirable to track material, labor, and utility costs, separately for each production entity; or if it were desirable to segregate asset installation costs from their purchase costs, this could be done by providing additional attributes or information entities for each category. Thus, intermediate cost collection points can be simulated, costs can be shared between product lines according to some allocation scheme, and virtually any of the accounting alternatives described in Section 2 can be replicated.

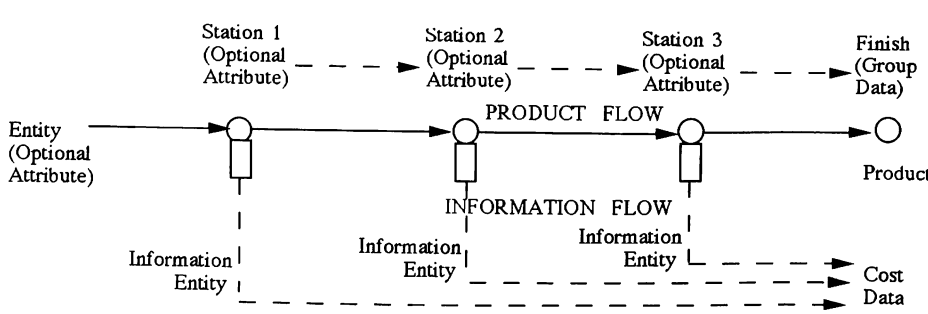

Figure 4. Accounting Data: Proposed Method

4.3 Simulation Results

The "Financial Values" calculated across several replications of this model are shown in Table 1.

A careful examination of the total output reveals a discrepancy related to accounting for work in process. Comparing the count of entering customers passing through Order Entry to the count of Delivered Products leaving Checkout shows that there were nearly always some customers and as yet unbuilt orders left in the system. Less frequently, a completed order was also left in the delivery queue. Completed orders are properly accounted for in the final Finished Goods Inventory, and no Margin errors (other than rounding) seem to have occurred for any replication. The incomplete orders left in the system represent a Work in Process inventory that may have accrued small amounts of AdminCosts prior to completion of the replication run, and these work in process values are causing the discrepancy, since these are costs incurred that are not properly balanced by recorded assets held in the system. This discrepancy points to a need for improving the reporting mechanism that accounts for work in process, so that even this simple model guides the design of the information system. It is significant, however, that although the model makes no explicit provision for reporting this work in process, it is being tracked properly and could be specifically picked up with some modifications to the model.

5. SUMMARY AND IMPLICATIONS FOR THE FUTURE

This model is one where Cost Accounting and Financial Accounting transactions have been incorporated into a simulated information system running alongside a production system. Although both the production and the accounting functions are comparatively simple, the demonstrated ability of the simulation program, IBIS, to carry out this set of functions simultaneously is important to simulation capability, in general.

Because of the modular nature of of IBIS models, the IBIS simulation program is capable of expanding the single production station into a series of production stations without rewriting any existing data entries or redesigning the model, so that more complex models can be developed from this preliminary model. Figure 5, for example, shows how two products, using a single facility, might be simulated with cost collection, using the complex

Table 1. Data Summary – 50 Replications

	AVE'GS	SDEVS	95% CONFID. INTERVAL MAX.	MIN.
Assem'r Util.	.51	.04	.52	.50
Bill Util.	.14	.04	.15	.12
Ckout Util.	.18	.02	.18	.17
Cust. Time	5.84	.36	5.94	5.74
Cust. Util.	.38	.02	.39	.38
Order Util.	.39	.04	.40	.38
Prod. Time	1.19	.05	1.21	1.18
Prod. Util.	.59	.02	.59	.58
Order #inQ	.16	.09	.18	.13
Order TinQ	.52	.14	.56	.48
Ckout #inQ	.67	.11	.70	.64
Ckout TinQ	2.59	.29	2.67	2.51
Average Financial Values				
Sales	400.50	53.93	415.45	385.56
CofGds	192.69	23.56	199.22	186.16
Margin	207.81	30.45	216.25	199.37
AdminCost	156.78	13.65	160.56	152.99
Profit	51.33	19.89	56.84	45.82
FinGdsVal	.79	.04	.80	.78
Logic check				
Margin	207.81	30.45	216.25	199.37
Profit	51.04	19.88	56.55	45.53
ErrMargin	.00	.00	.00	-.00
ErrProf	.30	.44	.42	.18
OrderCount	122.80	10.46	125.70	119.90
SalesCount	121.46	10.23	124.30	118.62
WIPCount	1.34	1.02	1.62	1.06

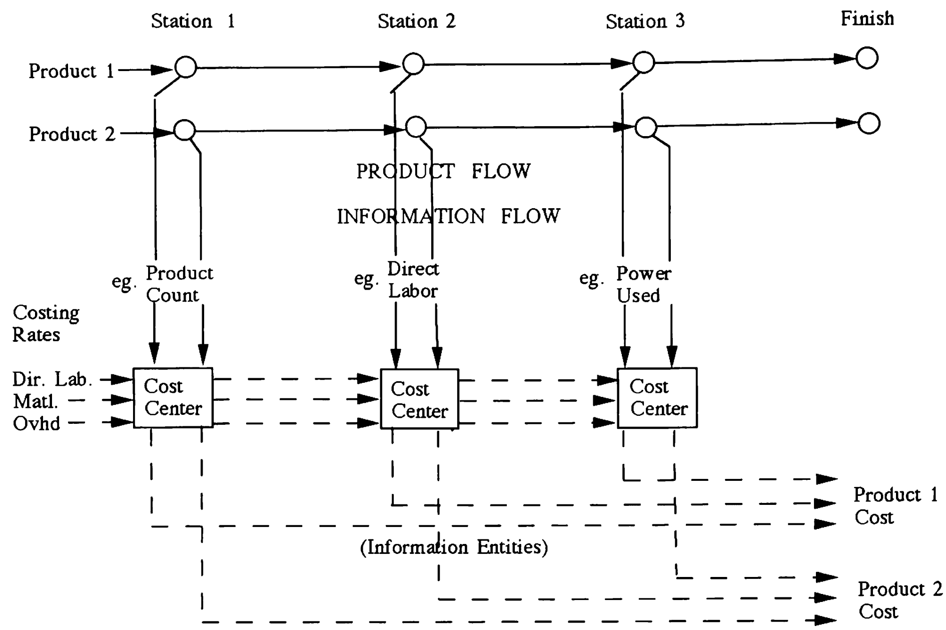

Figure 5. Two Products With Separated Information Flow System

accounting scheme of Figure 1, in which costs are aggregated at cost centers and then shared by the two products. Adding new work stations, products, and information entities are all straightforward modifications to the IBIS input specifications.

In addition to the ability to work with accounting data, this model and the IBIS program have provided an interesting small laboratory in which to experiment with alternative reporting arrangements. This model has required (1) the setup of a data gathering set of operations, (2) the integration of these operations into a small information system, (3) the specification of what information to carry to what points of the organization, and (4) the specification of the process by which that information should be reduced to accounting values and presented to the outside world. Although these steps for this model consisted mostly of single formula arithmetic expressions, it was not always obvious where costing rates should be applied, where information should be aggregated, which stations needed what information, and so on. These issues appear to be surfacing also in the current manufacturing and accounting literature for "real world" applications.

With the growth in the Just In Time production concept and the availability of relatively cheap computer power for gathering and processing large amounts of data, many are beginning to re-examine the data gathering and processing operations that accompany production operations. Higher priorities are being placed upon accuracy, and timeliness of these systems just as new processing concepts, such as JIT, are changing production floor layout and operations.

Using this methodology, the ability of various information networks to gather and present accurate and timely information can be tested. Possibly more valuable, the ability to experimentally devise improved measures that accurately and concisely predict performance could be addressed. Just as the creation of this model has revealed ways in which cost data is handled that are different from the ways product entities are treated, the expansion of this model will probably reveal more about how effective information systems are or should be constructed.

APPENDIX: IBIS INPUTS

```
> File: Station
StaName            Class               Parent
/* ========================================= */

Firm               -                   -

Assembly           -                   Firm
Billing            -                   Firm
  PdCost:        -        StatType: Observation
  PdPrice:       -        StatType: Observation
Checkout           -                   Firm
  PdCost:        -
  PdPrice:       -
Controller         -                   Firm
  AdminCost:     -        StatType: Observation
  CofGds:        -        StatType: Observation
  FinGdsVal:     -        StatType: Observation
  Margin:        -        StatType: Observation
  Profit:        -        StatType: Observation
  Sales:         -        StatType: Observation
OrderDesk          -                   Firm

> File: Entity
EntName            Class               Type       Fixed
/* ================================================== */

Assembler          -                   Permanent  Yes
Clerk              -                   Permanent  Yes
BillClerk          Clerk               Permanent  Yes
CheckoutClerk      Clerk               Permanent  Yes
OrderClerk         Clerk               Permanent  Yes

Customer           -                   Temporary  No
  StartServTime: -
WorkOrder          -                   Temporary  No
  startServTime: -
Product            -                   Temporary  No

Info               -                   Temporary  No
BillAdminData      Info                Temporary  No
  BillTime:      TIME-FVAL("*;StartServTime")
CkoutAdminData     Info                Temporary  No
  CkoutTime:     TIME-FVAL("*;StartServTime")
  PdCost:        FVAL("*,Invoice;PdCost")
  PdPrice:       FVAL("*,Invoice;PdPrice")
```

```
OrderAdminData     Info                 Temporary  No
  OrderTime:      TIME-FVAL("*;StartServTime")
PdCostData         Info                 Temporary  No
  PdTime:         TIME-FVAL("*;StartServTime")
  StartServTime: -

Invoice            -                    Temporary  No
  PdCost:         FVAL("*;PdTime")*0.3+0.8
  PdPrice:        FVAL("*;PdTime")*0.8+1.2

> File: Route
Entity             Step      Source       Destination
/* ============================================== */

Customer           0.0000    -            OrderDesk
  InterArrRate:   'EXPON(4.0)'
Customer           0.0000    OrderDesk    Checkout
WorkOrder          0.0000    OrderDesk    Assembly
Product            0.0000    Assembly     Checkout

OrderAdminData     0.0000    OrderDesk    Controller
BillAdminData      0.0000    Billing      Controller
CkoutAdminData     0.0000    Checkout     Controller
Invoice            0.0000    Billing      Checkout
PdCostData         0.0000    Assembly     Billing

> File: OpTime
Operation          Entity            ProcTime
/* ====================================== */

Assemble           WorkOrder        'NORMAL(2.0,.3)'
Bill               PdCostData       'NORMAL(.5,.1)'
Delivery           Product          'NORMAL(.7,.2)'
GetClerk           -                '0.0'
OrderInput         Customer         'NORMAL(1.5,.5)'

> File: ProcessSteps
Station            Entity          StepNumbr Command
/* ============================================== */

OrderDesk           Customer        1.0000     OPERATION
  Queue:          CustomerIn
  Operation:      GetClerk
OrderDesk           Customer        2.0000     -
  Operand:        'FASSIGN("*;StartServTime",TIME)'
OrderDesk           Customer        3.0000     OPERATION
  Operation:      OrderInput

Assembly            WorkOrder       1.0000     OPERATION
  Queue:          WorkWaiting
  Operation:      GetClerk
Assembly            WorkOrder       2.0000     -
  Operand:        'FASSIGN("*;StartServTime",TIME)'
Assembly            WorkOrder       3.0000     OPERATION
  Operation:      Assemble

Checkout            Customer        0.0000     QUEUE
  Queue:          CustomerOut
Checkout            Invoice         0.0000     QUEUE
  Queue:          InvoiceOut
Checkout            Product         1.0000     OPERATION
  Queue:          FinishJobs
  Operation:      Matchup
Checkout            Product         2.0000     -
  Operand:        'FASSIGN("*;StartServTime",TIME)'
Checkout            Product         3.0000     OPERATION
  Operation:      Delivery

Billing             PdCostData      1.0000     OPERATION
  Queue:          BillsWaiting
  Operation:      GetClerk
Billing             PdCostData      2.0000     -
  Operand:        'FASSIGN("*;StartServTime",TIME)'
Billing             PdCostData      3.0000     OPERATION
  Operation:      Bill

Controller          BillAdminData  0.1000     -
  Operand:        'FADD("STATION:*;FinGdsVal",
                     FVAL("*;PdCost"))'
Controller          BillAdminData  0.5000     -
  Operand:        'FADD("STATION:*;AdminCost",
                     FVAL("*;BillTime")*0.4)'
Controller          CkoutAdminData 0.7500     -
  Operand:        'FADD("STATION:*;AdminCost",
                     FVAL("*;CkoutTime")*0.4)'
Controller          CkoutAdminData 1.0000     -
  Operand:        'FADD("STATION:*;FinGdsVal",
                     -FVAL("*;PdCost"))'
Controller          CkoutAdminData 2.0000     -
  Operand:        'FADD("STATION:*;Sales",
                     FVAL("*;PdPrice"))'
Controller          CkoutAdminData 3.0000     -
  Operand:        'FADD("STATION:*;CofGds",
                     FVAL("*;PdCost"))'
Controller          CkoutAdminData 4.0000     -
  Operand:        'FASSIGN("STATION:*;Margin",
                     FVAL("STATION:*;Sales")-
                     FVAL("STATION:*;CofGds"))'
Controller          CkoutAdminData  5.0000     -
  Operand:        'FASSIGN("STATION:*;Profit",
                     FVAL("STATION:*;Margin")-
                     FVAL("STATION:*;AdminCost"))'
Controller          OrderAdminData 0.0000     -
  Operand:        'FADD("STATION:*;AdminCost",
                     FVAL("*;OrderTime")*0.4)'
```

REFERENCES

Conway, R., W.L. Maxwell, J.O. McClain and S.L. Worona (1987), *User's Guide to XCELL+ Factory Modeling System*, Scientific Press, Redwood City, CA.

GEMS-II User's Manual (1987), Revision 3, Lodestone, Inc., Bryan, TX.

Johnson, H.T. and R.S. Kaplan (1987), *Relevance Lost: The Rise and Fall of Management Accounting*, Harvard Business School, Cambridge, MA.

Kaplan, R. S (1989) "Management Accounting for Advanced Technological Environments," *Science*, Aug. 25,819-823.

Ketcham, M. G., R. E. Shannon and G. L. Hogg (1989), "Information Structures for Simulation Modeling of Manufacturing Systems", *Simulation* 52, 2, 59-67.

Schmenner, R.W. (1987), *Production/Operations Management: Concepts and Situations*, SRA, New York, NY.

Strack, D.E. (1988), Personal correspondence and promotional brochure, to the authors. November 30, 1988.

Watson, H.J. (1981), *Computer Simulation in Business*, John Wiley, New York.