Optimizer: IBM's Multi-Echelon Inventory System for Managing Service Logistics

MORRIS COHEN — *Department of Decision Sciences, University of Pennsylvania, Philadelphia, Pennsylvania 19014*

PASUMARTI V. KAMESAM — *IBM T. J. Watson Research Center, Yorktown Heights, New York 10598*

PAUL KLEINDORFER — *Department of Decision Sciences, University of Pennsylvania*

HAU LEE — *Department of Industrial Engineering, Stanford University, Stanford, California 94305*

ARMEN TEKERIAN — *IBM National Service Division, Franklin Lakes, New Jersey 07417*

IBM recently implemented Optimizer, a system for flexible and optimal control of service levels and spare parts inventory, in its US network for service support. It is based upon recent research in multi-echelon inventory theory to address the IBM network. The inherent complexity and very large scale of the basic problem required IBM to develop suitable algorithms and sophisticated data structures and required large-scale systems integration. Optimizer has greatly improved IBM's US service business. The implementation of Optimizer has made it possible to make strategic changes to the configuration and control of the parts distribution network. It resulted in simultaneously reducing inventory investment and operating costs and improving service levels. Most important, however, Optimizer has proven to be a highly flexible planning and operational control system.

The information processing industry has experienced several decades of sustained, profitable growth. Recently, competition has intensified, and as a result, there have been rapid advances in computer technology, leading to a proliferation of both end-products and services. These trends, which have an important implication for all aspects of business operations, are especially relevant for after-sales service. Maintaining a service parts logistics system to support

0091-2102/90/2001/0065$01.25

INVENTORY/PRODUCTION — APPLICATIONS
INDUSTRIES — ELECTRIC/ELECTRONIC

products installed in the field is essential to competing in this industry.

Growth in both sales and the scope of products offered has dramatically increased the number of spare parts that must be maintained. Spare parts for information technology have also become more modular and more expensive and are used with increased commonality. These design changes economize on training costs of customer engineers and simplify diagnostic procedures.

For the IBM Corporation, the number of installed machines and the annual usage of spare parts have both increased. This growth has put upward pressure on the dollar value of service inventories, which are used to maintain the extremely high levels of service expected by IBM's customers. IBM has developed an extensive multiple-echelon logistic structure to provide prompt service for the vast population of installed machines, which are distributed throughout the US.

IBM's National Service Division (NSD) developed an extensive and sophisticated inventory management system to provide customers with prompt and reliable service. For many years we considered this system adequate in maintaining service and controlling inventory levels. A rapidly changing NSD business environment and the pressures to decrease inventory investment led IBM to search for improvements in its control system. The improvements it sought included (1) management flexibility in setting strategically driven service targets for different market segments and (2) improved inventory efficiency and cost control. In response to these new needs, IBM initiated the development of a new planning and control system for service parts management.

The effort resulted in the creation and implementation of a system called Optimizer.

An Overview of IBM Service Logistics

NSD, the service division of IBM, is responsible for providing high quality, after-market support to IBM's customers. It must also support IBM's marketing efforts and manage a profitable and competitive (after-sales) service business. The service marketplace is comprised of commercial and government customers and IBM internal accounts, with products installed or on order in the United States and its territories. Apart from IBM, a number of third-party maintainers vie for the service business afforded by this large customer base.

The geographic dispersion of service customers, coupled with the need for very quick response and repair, requires a large customer engineering work force. IBM also employs an efficient and sophisticated group of people and systems to support the work of the customer engineers. NSD employs over 15,000 customer engineers (CEs), who are trained to repair and maintain all of the installed systems.

When a machine does fail, a CE is dispatched in response to a customer telephone call to a dispatch center, or automatically by communication from the failed machine. Most repair calls require parts for replacement, diagnostics, or tools. Hence, quick and efficient deployment of these parts is vital to getting the customer's machine fixed and running quickly. CEs may obtain the needed parts in several ways. For example, they may

be delivered to the customer site before or after the CEs arrive, or the CEs may use parts stored on the customer premise (an outside location). They also carry a limited number of parts in their car trunks (or tool chests).

The distribution organization of NSD is responsible for procuring and storing parts and deploying them to CEs and directly to customers. Direct sales are to dealers, third party maintainers, and self servicers. It has been a mainstay of IBM's competitive strategy to deliver parts within a very short time.

Distribution consists of two suborganizations, Distribution Operations and Inventory Planning. Distribution Operations is responsible for transportation, warehousing, order entry, and other physical distribution functions, such as inventory record maintenance, quality inspection of inbound and outbound parts, and packaging parts. Inventory Planning is responsible for procuring, planning, and maintaining inventory throughout the parts network. It gets involved early in the life of a product in order to procure and maintain inventory to support the product throughout its market life. Inventory Planning also works closely with the service delivery group to gather and use data, such as the projected number of machine installations, parts failure rates, critical parts, engineering changes, upgrades, and service strategy.

About 1,000 IBM products are in service. The installed population of these products exceeds tens of millions.

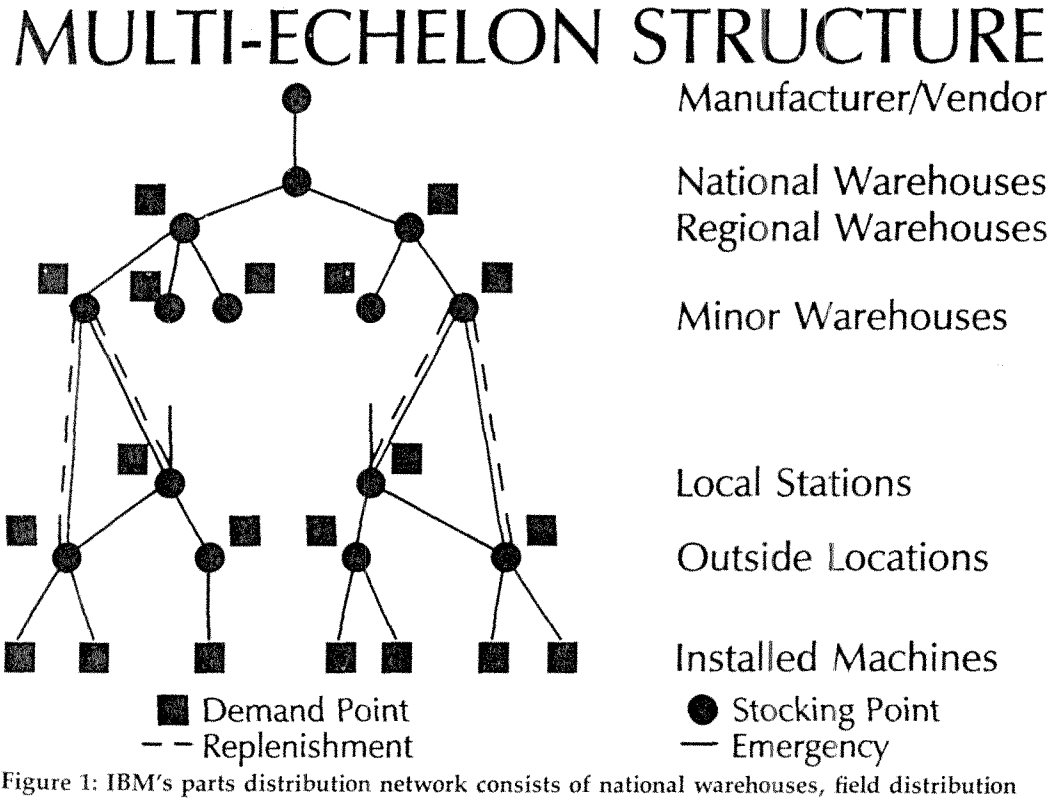

Figure 1: IBM's parts distribution network consists of national warehouses, field distribution centers, emergency parts support centers and outside locations.

Inventory Planning controls over 200,000 part numbers to support the operation of these products throughout the installed customer machine population. The major task of this group is to manage the flow of material in a manner that keeps the system supplied, reduces costs, and achieves IBM's stringent performance targets.

The IBM parts distribution system is a large, complex, multi-echelon network (Figure 1). Prior to the implementation of the Optimizer system, the locations were organized into four echelons as follows:
(1) Two central automated warehouses,
(2) 21 field distribution centers (FDCs),
(3) 64 parts stations (PSs), and
(4) 15,000 outside locations (OLs).

The central warehouses are located in Mechanicsburg, Pennsylvania and Lexington, Kentucky. These two sites receive parts from IBM plants and vendors and resupply the rest of the network. The central sites replenish the 21 FDCs and provide emergency backup for demands not filled by the FDCs. They are also responsible for shipping domestically manufactured parts to non-US IBM field service organizations and for providing parts to tens of thousands of authorized dealers and external customers.

Inventory Planning controls over 200,000 part numbers to support the operation of about 1,000 IBM products in service.

Located in the major populated metropolitan areas, the FDCs act as regional distribution centers. They provide emergency support to their assigned regions of the country, and they also replenish all of the PSs and OLs located in their territories. Each FDC can support other FDCs within the network.

PSs are stocking stations located in the service branch offices. They are responsible for filling failure induced (emergency) orders only. The PSs are usually located in medium to small metropolitan areas. Outside locations are stocking locations that are not staffed. There are three basic types of OLs: on-site customer stock locations, CE car trunks or tool chests, and shared parts sets stocked at local branches.

The inventory level for each part number is tracked at all stocking locations (except at OLs where only parts costing more than a certain threshold are tracked). Each tracked part/location combination is called a stock-keeping unit (SKU). There are currently several million SKUs in the NSD parts distribution network.

The parts inventory maintained in this network is both large and diverse. It is valued in the billions of dollars. Most of this inventory is carried at noncentral site locations. The network is extremely active, especially in the upper echelons. Millions of parts transactions are processed through the network annually. These include returns from CEs as well as disbursements to CEs.

The logistics of this network are managed by sophisticated information and control system called the parts inventory management system (PIMS). PIMS is a large complex system containing millions of lines of code (Figure 2).

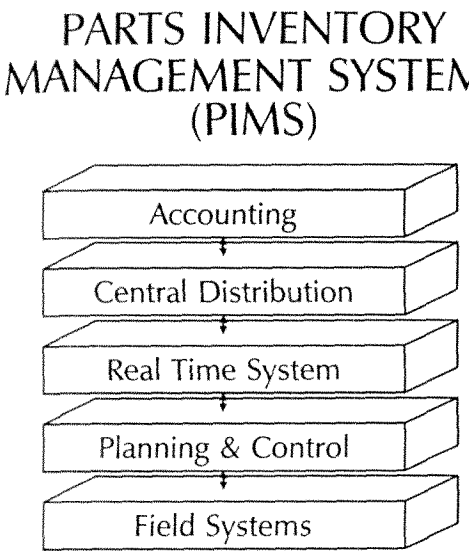

Figure 2: The PIMS system depicted as 5 major modules.

PIMS accounting module maintains all the accounting and costing information. Central distribution plans and controls the flow of parts from the plants and vendors into the spare parts distribution system. In order to process the large volume of transactions in a timely manner, IBM implemented a real-time order-processing system (RTS) in the FDC and PS locations. The CEs are in constant communication with the real-time system through hand-held terminals. This sophisticated information and communication system makes the job of filling CE parts orders a highly efficient, paperless, and phoneless operation.

The field systems of PIMS consist of dedicated information systems used to track the direct support and backup locations for each machine installed in the field identified by machine type, model, and serial number. Through the field systems, each unique customer machine is linked to a list of part numbers contained in that machine. IBM must maintain machine records down to the serial number because of the level of customization used in the production of high-end systems. This information allows the inventory planning and control system to ascertain all of the parts that each stocking location may need. PIMS also enables IBM to deploy parts to support its introduction of new products.

Prior to the implementation of Optimizer, the inventory planning and control functions of PIMS were performed by two systems working in conjunction, a recommended spare parts (RSP) system and a demand system. An RSP list was associated with each machine type. It specified a minimum complement of parts that should be stocked at each echelon. The RSP lists were prepared initially by the product designers and then revised periodically based on the history of part usage and on judgment. PIMS used the RSP list to set minimum stock levels for each part number in order to provide an acceptable level of customer service. The demand system in PIMS used standard, single-location order point, order-up-to-level logic in conjunction with an exponential smoothing forecast based on site-specific, locally observed consumption patterns.

Potential for Improvement

The principal requirements for effective after-sales service support are service level flexibility and inventory efficiency. The challenge facing NSD was to achieve lower costs while maintaining or improving customer service.

A variety of service measures can be used to support information system

technology products. PIMS used the parts availability level (PAL) as its basic performance measure. PAL is equal to the fraction of a part's demand that is filled immediately from on-hand stock at a stocking location. Since multiple parts make up a machine, the PALs for these many parts affect service performance when machines fail. In the newly developed Optimizer system, machine service objectives are specified for each echelon level. These service objectives then determine target PAL values for each part used in each machine. Many different PAL-combinations yield a given machine service objective. The old PIMS stocking procedures did not consider all of the cost consequences of alternative PAL combinations.

This problem is compounded by the proliferation of high technology products. For such products IBM has identified specific groupings of parts, called technology component groups (TCGs). A failure of a TCG part leads to the complete failure of the machine. We wanted to provide the flexibility to specify service requirements not only for products, but for TCG's within products as well. Also management wants to be able to set different service goals for different market requirements (products/locations) in order to respond to changes in the competitive environment.

The existing process used by service delivery and inventory planning analysts in assuring service performance was complex and arduous. They used on-line parts-ranking models and judgment to define RSP lists. These lists were input to PIMS to establish the parameters of the stocking policy. It was difficult to change an RSP list and its associated stocking policy parameters in response to changes in service requirements.

Concurrent with its need to achieve service flexibility, IBM is also committed to containing the growth of its parts-inventory investment and other inventory-related costs. We determined that PIMS could be improved in several ways:

— by improving the demand forecasting method,
— by accounting for the multi-echelon structure,
— by accounting for part commonality, and
— by enhancing cost-service trade-offs.

Information on the number of installed machines (each with a unique complement of parts) in each geographical region, together with part-failure rates, can be used to improve the forecast for regional demand of parts. Moreover, the local-consumption-based exponential-smoothing procedure used in PIMS had difficulty in dealing with parts for which demand was erratic and infrequent.

The inventory control algorithms in PIMS used single location logic in determining parameters for stocking control. In a multi-echelon setting, the stocking control decisions at lower echelon locations affect the patterns of demand observed (quantity, timing, and priority) at higher echelon supply points. Moreover, the demand faced by a given higher echelon location can be classified into different priority types (1) requirements generated by part failures in customer machines directly supported by this location; (2) emergency (expedited) requirements for

customer demands not filled at lower echelon locations, which use this location as an emergency backup supply; and (3) replenishment requirements to restock lower echelon locations that use this location as their source. These linkages and demand priorities should be considered in setting stocking control parameters for all locations in the network.

Products of the same family have many parts in common. Hence, the stocking policies for a common part affect the customer service of all machines that use that part. Significant savings in inventory can be achieved by incorporating the commonality relationship into stock control procedures.

The existing (PIMS/RSP) system used a simple cost classification in setting service priority targets for parts. It also used an EOQ formula to set replenishment batch sizes. In order to improve the efficiency of the inventory control system, we had to evaluate total costs, which also include emergency (expediting) transportation costs. Cost trade-offs also had to be considered over all locations and parts.

The challenge facing NSD was to address the escalating need for increased service flexibility and inventory efficiency by creating a new inventory control system.

Model Development

An initial review of the PIMS capabilities indicated that it would not be able to address IBM's concerns for increased service flexibility and inventory efficiency. In 1983-84, researchers from IBM and the academic team (working as consultants) started to develop a model formulation and a solution algorithm. The magnitude of the project was complicated by the following factors:

- There are over 15 million part-location combinations;
- There are over 50,000 product-location combinations;
- System control parameters must be updated frequently (weekly) in response to changes in the service environment and installed base;
- The success of the system is vital to IBM's daily operations and can have a major impact on its future sales and revenues; and
- Employees within the organization could be expected to resist any change; the existing control system was functioning, and sophisticated and the overall parts logistics problem was complex.

A review of the literature showed that most of the relevant existing multi-echelon theory is based on a one-for-one replenishment model structure [Feeney and Sherbrooke 1966]. The one-for-one replenishment policy is the basis of military logistics control, and represents the state of the art for low demand spare stocking control systems [Sherbrooke 1968; Muckstadt and Thomas 1980; and Graves 1985]. While appropriate for low-demand items, this policy does not provide adequate cost and service performance for the wide range of demand rates present in IBM's

The parts inventory maintained in this network is valued in the billions of dollars.

parts environment. Moreover, these models treat items independently and hence product and part service interactions are not captured. Finally, these models are restricted to one demand priority class; IBM's parts inventory system contains multiple demand priority classes and their explicit treatment is necessary.

We began to solve the problem. Our objective was to determine a stock control policy for each location and for each part that would minimize the expected costs for the whole system and satisfy service constraints for products and TCGs. The cost function that we used included (1) replenishment cost, (2) emergency (expediting) cost, and (3) inventory holding cost. Replenishment cost contains transportation, handling, and order setup components. A comprehensive set of inventory management decisions include

(1) Reorder-point, order-up-to (s,S) stock-control parameters for each part at each location and echelon in the system,
(2) Alternative sourcing network structures for each part, for both emergency backup and stock replenishment requirements, and
(3) Issuing policy that determines the priority attached to different classes of demand when shortages arise.

We decomposed the model development process into three stages:

(1) A one-part, one-location model,
(2) A multi-product, one-location model, and
(3) A multi-product, multi-echelon model.

For the one-part, one-location problem, we formulated a periodic review, stochastic model. It included prioritized demand classes, shortage expediting, fixed lead times, and two supply sourcing network structures (one for emergency backup of failure-induced demands, and the other for replenishment purposes). We developed a general (s,S) stochastic inventory control model for a single item with prioritized demand classes to support the solution to the location-specific service allocation problem. The replenishment lead times are based on the assumption that the replenishment sources have ample supplies of the required parts. An exact representation of the stochastic processes describing material flows in such systems requires the solution of a large-scale Markov chain for each policy alternative. For computational efficiency, we developed renewal theory-based approximations for both the expected cost and service level functions [Cohen, Kleindorfer, and Lee 1988].

Separate models can be formulated for the case where at most one order is outstanding $(S-s>s)$ and where there is one-for-one replenishment $(S-1,S)$. An important refinement to the basic single part model involved the interpolation of these two extreme cases for the computation of part fill rates [Kamesam and Tekerian 1986]. For each location within an echelon, we used the model to consider the problem of minimizing expected inventory-related costs for all parts used by machines supported by that location (directly and indirectly), subject to product service constraints at that location. The outputs of this problem are (s,S) values for all relevant parts. We refer to this

multi-product, one-location model as the service allocation problem.

We used a greedy heuristic solution procedure in solving the service allocation problem. At each iteration of the algorithm, increments to part stocking levels are selected on the basis of their marginal contributions to improving the objective function and to meeting the service constraints. The effectiveness of this heuristic is reported in Cohen, Kleindorfer, and Lee [1988 and forthcoming], and Cohen et al. [1989].

The multi-product, one-location model solutions are then embedded into a solution algorithm for the overall multi-echelon problem. This solution algorithm is based on a level-by-level decomposition method. The decomposition starts with the locations at the lowest echelon (OLs), where all demands are generated by

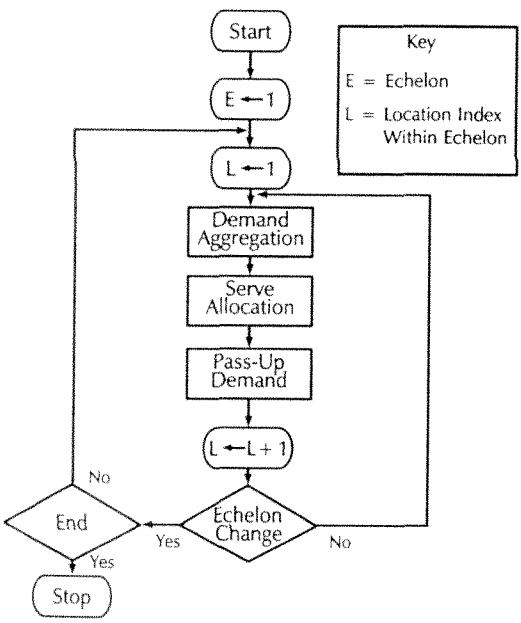

Figure 3: The level by level decomposition algorithm links the stocking policies at higher echelons with those of the lower echelons.

failures of machines installed at customer sites. At each location in that echelon, a location-specific service allocation problem is solved. The solutions to these problems are then used to derive a characterization of the "passed-up" demands that are experienced at the next echelon locations. We obtained probability distributions for both emergency backup and replenishment demands using a method based on nonlinear regression techniques [Cohen et al. 1986]. Given these passed-up demand distributions, the set of location-specific service-allocation problems for all locations at the next higher echelon can be solved. The procedure is then repeated until all echelons have been considered (Figure 3).

The level-by-level decomposition algorithm does not, in general, give truly optimal solutions to the multi-echelon problem. In reality, the replenishment lead time at each location is a random variable and depends on the stocking policies of higher echelons. By treating the echelons one at a time, we use the assumption that this lead time is constant, that is; there is always an ample supply of parts at the replenishment sources. Such a solution procedure is likely to be closer to optimality in cases where the service requirements at all sites are high. The appendix contains the details of the mathematical formulation and solution algorithms.

System Development and Implementation

Soon after we completed the initial research efforts, we wrote prototype computer programs to test the algorithms. We developed an event-driven Monte-Carlo simulator to validate the approximations

for fill rates and costs incorporated in the multi-echelon algorithm. These experiments showed that the PAL estimates generated by this algorithm are slightly conservative. We compared the performance of the policies generated by the algorithm against those generated by PIMS and by the one-for-one replenishment model of Feeney and Sherbrooke [1966] to understand the relative cost/service efficiencies. We made numerous runs based on subsets of the logistics system locations, product lines, and parts population. By the end of 1984, we were convinced that these methodologies held promise (that is; that they could be adapted into an operational system) but knew they would require a great effort to develop and implement. We recognized that integrating such a system with PIMS would be an equally big challenge. During the 1985-88 period we worked on and completed the design, testing, and installation of the new system, Optimizer.

An important first step in the Optimizer project was to form a multi-functional project group. The project group members were divided into the following teams:

— The User Team: Members of the user team were drawn from each functional area that would be affected by the introduction of a new parts-control system (including information systems, inventory and distribution, product serviceability, and customer engineering). These team members worked on the project on a part-time basis by attending system-design sessions, reviewing user-acceptance tests, and helping to write design specifications.

— The Information Systems (I/S) Team: This team consisted of the I/S programmers who developed code to feed data to the model and to interface the model output into PIMS.

— Model Development Team: This team included an operations researcher and computer scientist, who was also one of the project leaders, and a mathematician-scientific programmer. The former reported to the user organization and the latter to the I/S group. Their roles were to define the system architecture, develop highly efficient and numerically stable implementations of the algorithms, and integrate them into a full-scale operational control system to be written in PL/1.

Filling customer engineer parts orders is a highly efficient, paperless, and phoneless operation.

Even though the project had the strong support of NSD senior management, it immediately became clear that people within the organization had serious reservations about its potential for actual improvement. There were also concerns based on the complexity of the task. For these reasons all the team leaders decided to conduct a pre-implementation test to quickly demonstrate the feasibility of the multi-echelon-based algorithm and its potential for improving system performance and to identify all design problems early enough to fix them. The project team leaders were determined that the system would be successfully implemented. Their

optimism was reflected in the names given to the three project phases well before implementation was assured. The belief was instilled in all project teammembers that any problem that emerged would have an acceptable solution, and it was their job to find it. The system development for Optimizer was organized into the following overlapping phases:

— The pre-implementation test,
— The field implementation test, and
— National implementation.

Each phase began with joint application design (JAD) sessions. In the JAD sessions, the user and information systems groups worked together to develop design specifications. Each phase entailed developing a system that would provide code useable for the next phase. The objective was to minimize the amount of throwaway code and to use the system developed for each phase as a foundation for the next. System development involved sizing the design specification to obtain the required I/S resources, developing a detailed design, coding, testing, and integrating the systems.

After developing a system in each phase, the user group and the I/S group performed tests of various modules, as well as of the entire system. Test cases were developed ahead of time, with programmer instructions and expected results. Each test case was run, and the results were documented. Test cases that did not achieve expected results were kept open until a user and programmer team found the root cause and a fix.

We used the results of the user acceptance tests to redefine the specifications for the next phase. To accelerate final implementation, we carried out the phases in a partially parallel manner. The user and I/S project members conducted reviews at each major milestone to redefine the specifications, designs, and approach. These reviews were conducted often, and open discussion was strongly encouraged.

The Pre-Implementation Test

The system we developed in this phase contained a minimal interface to provide data inputs and the multi-echelon algorithm without any enhancements. We did not develop an output interface to PIMS. Most of the major changes from the original design occurred in this phase. Some of these changes are of particular interest.

During the JAD sessions, we discovered that the echelon structure was actually more complex than the structure used in the analytic model. In particular we discovered that two installations of the same machine type and model could share a first-level support location and have different backup support locations. This was due to the fact that in the PIMS system the backup support structure for each machine installation was established independently of the other machine installations. The user team members were adamant that such flexibility in assigning initial and back up support locations be maintained. Consequently, we had to develop extensions to the demand pass-up methodology and incorporate them into the model.

The pre-implementation test was conducted in early 1986. This test led to the discovery that the value of the total inventory generated by the new system was much smaller than expected. Test team

members poured through the output runs until they discovered that the problem was due to differences in criticality of parts. Failure of some parts can cause a machine to fail or run in a degraded state. Other parts, such as screws, washers, covers, supplies, and filters are consumed in great volumes but have no direct impact on machine performance. The algorithm made extensive use of these inexpensive, nonfunctional parts to achieve product service objectives. It was clear that only functional parts should be counted in computing the impact of PALs on product service performance. The model development team modified the model and data structures to account for this dichotomy in part criticality. We established a parts classification process to group parts on an ongoing basis.

Another problem discovered at this stage was the churn (instability) in the recommended stock levels from week to week. Although stock levels are expected to change from time to time in response to changing failure rates and to changes in the installed base, it is desirable to keep the stock levels quasi-static in order to avoid serious logistic and supply problems. We developed suitable control procedures and modified the model to take care of the churn problem.

Field Implementation Test

In this phase, we completed all of the functions required for implementation, including code to interface the algorithm into PIMS. We also developed an extensive measurement system to monitor the field implementation test. It was interesting to note that by this stage all of the project members had become ardent supporters of Optimizer. They all had hands-on experience with the system and were comfortable with it. They also had contributed to shaping the system design. As a result, all project members had a strong sense of ownership. They began to sell Optimizer to their functional areas.

After completing system coding for this phase, we conducted a very extensive user acceptance test. Every program module was tested both individually and jointly. About 400 test cases were run for this purpose. Finally, an extensive field implementation test went live on seven machine types in one FDC cluster. This test began in early 1987. Needless to say there was much celebrating when we discovered that the system worked as expected. The scope of the field test was gradually expanded. The results were monitored on a weekly (and then a monthly) basis by the measurement system.

National Implementation

In this phase we completed the development and installation of all the functions currently in place in Optimizer. The system was able to provide the specified service performance for all parts and locations by machine, model, and TCG. We also completed a variety of additional enhancements in this stage. User acceptance testing and final system integration went smoothly.

Most of the project team was kept intact throughout the three stages. This helped to facilitate delivery of Optimizer in the final stage. The project staging helped sustain support for the project by demonstrating concrete progress throughout the implementation process. It also

helped to flush out formulation and algorithm problems and programming bugs early on. As a result, very few problems occurred when the system went live nationally.

The final Optimizer system for national implementation was a PL/1 based application system consisting of four major modules:

— A forecasting system module that consists of a few programs that estimate the failure rates of individual part numbers in each product, and programs that combine these failure rates with information on the machine installation base to estimate the first two moments of the part failure probability distributions;
— A data delivery system module that contains approximately 100 PL/1 programs that process over 15 gigabytes of data to provide the basic data inputs for Optimizer;
— A decision system that solves the multi-echelon stock-control problem. It is designed to handle the enormous and varying internal memory requirements of the algorithms, as well as to provide computational efficiency. The module has its own dynamic memory management scheme to control the allocation and release of storage for all the data structures. The tailored dynamic memory management scheme had a dramatic impact on processing time. Today, the decision system generates a solution in under 75 minutes of CPU time and less than five hours of elapsed time on a IBM 3084 CPU (which is well under the desired processing requirement); and
— The PIMS Interface System, which consists of six PL/1 programs that serve as interfaces for the output of the decision system and PIMS.

Optimizer is now an integral part of PIMS and is run each week.

Impact

The implementation of Optimizer yielded a variety of benefits.

(1) A reduction in inventory investment required,
(2) Improved service,
(3) Enhanced flexibility in responding to changing service requirements,
(4) The provision of a planning capability,
(5) Improved understanding of the impact of parts operations on a customer service,
(6) Increased responsiveness of the control system, and
(7) Increased efficiency of NSD human resources.

These benefits can be traced to the key advantages of the Optimizer methodology: optimization, improved forecasting, multi-echelon linkages, and product-part interactions. We believe that much of the savings resulted from the improved forecasting and optimization of stocking levels.

The measurement system installed during the field implementation test phase was used to quantify a number of these benefits relative to a baseline derived from the existing PIMS stocking logic. The time-averaged value of inventory recommended by the stocking policies of Optimizer was 20 to 25 percent below that of the existing system. This difference was obtained along with equal or improved levels of service. This difference is in excess of a half a billion dollars of

inventory investment. NSD management, however, decided to redeploy part of the inventory reduction to improve the service levels and to reduce operating costs. Nevertheless, a conservative estimate of the annual total inventory reduction from operating Optimizer throughout the entire network is in excess of a quarter of a billion dollars.

In September 1988, NSD began to implement strategic network changes involving facility location, sourcing of emergency and replenishment material flows, and service targets for critical locations. These changes included decreasing the number of field distribution centers, increasing the number of parts stations, and increasing the fill rates at the parts stations and outside locations. Concurrent with the overall reduction in inventory investment, these strategic changes have yielded

— a 10 percent improvement in the parts availability at the lower echelons while maintaining the parts availability levels at the higher echelons, and
— operational efficiency on the order of 20 million dollars a year.

Optimizer was instrumental in developing and implementing these strategies. It did so by providing a planning tool to evaluate the potential impact of service policy changes for different network configurations. Moreover, Optimizer also drives the operating system used to adjust stocking policies in response to these policy changes.

Identifying the role of functional parts in providing product service is an example of the benefits derived from the implementation of Optimizer. Classifying parts according to their functionality has led to more effective management and measurement of product service.

The ability to run Optimizer on a weekly basis has increased the responsiveness of the entire parts inventory system. Stocking list updates, which used to be performed monthly, are now recomputed weekly with each OPTIMIZER run.

Finally, for machines controlled by Optimizer, inventory analysts no longer have to specify parts stocking lists for each echelon in order to assure that service objectives are attained. These analysts can now focus on other critical management issues, such as inventory deployment for new product support, and engineering changes.

Optimizer has proved to be an extremely valuable planning and operating control tool for NSD. It has enhanced the effectiveness of after-sales service delivery, and as a result, it supports IBM's competitiveness. The full impact of Optimizer has yet to be realized.

The time-averaged value of inventory recommended by the stocking policies of Optimizer was 20 to 25 percent below that of the existing system.

Acknowledgments

We acknowledge the support of Mr. David McDowell, IBM vice-president and president of NSD, who initiated the project described in this paper. We also thank the following individuals for their help: Dave Cabana, Marty Canavan, Amitabh

Dutt, Jerry Koenig, Skip Molander, Ray Nealon, David Perlmutter, Larry Readnour, Richard Smock and many others from IBM; and Professor William Pierskalla at the Wharton School, University of Pennsylvania.

APPENDIX

This appendix gives an overview of the mathematical formulation of the Optimizer problem and the associated solution algorithm. Details for the model, its properties, and the effectiveness of the solution algorithm are reported in Cohen, Kleindorfer, and Lee [1985], Cohen, Kleindorfer, Lee, and Tekerian [1986], Cohen, Kleindorfer, and Lee [1988], Cohen, Kleindorfer, and Lee [forthcoming], and Cohen, Kleindorfer, Lee, and Pyke [1989].

Let

s_{jk} = reorder point for part j of location k in the network, $j \varepsilon J$, $k \varepsilon K$;

S_{jk} = order-up-to point for part j of location k in the network, $j \varepsilon J$, $k \varepsilon K$;

$\underline{s}$ = vector of all the s_{jk}, $j \varepsilon J$, $k \varepsilon K$;

$\underline{S}$ = vector of all the S_{jk}, $j \varepsilon J$, $k \varepsilon K$;

PAL_{ik} = target parts availability level for product i at location k, $i \varepsilon I$, $k \varepsilon K$;

$\phi_{jk}(\underline{s}, \underline{S})$ = expected cost for part j at location k, when the stocking policies at all locations are given; and

$\psi_{ik}(\underline{s}, \underline{S})$ = expected fill rate for product i at location k, when the stocking policies for all parts are given.

The expected cost for a part at location k depends on the stocking policies of the same part at other locations (specifically, locations at lower echelons that are supplied by k). This is so because stocking policies at locations in lower echelons will affect the incoming demand distribution for location k. Also the expected fill rate for product i at location k is a function of the stocking policies of all the parts that support that product at all locations. The stocking policies at location k determine the parts availability levels (PALs) for all items stored at that location. Given the product structure, these PALs determine a product service level for the parts requirements generated by product repair orders. The overall problem can be stated as a mathematical program:

$$\text{Minimize} \sum_{k \varepsilon K} \sum_{j \varepsilon J} \phi_{jk}(\underline{s}, \underline{S}) \qquad \text{(I)}$$

subject to $\psi_{ik}(\underline{s}, \underline{S}) \geq PAL_{ik}$, for all $i \varepsilon I$, $k \varepsilon K$.

To specify $\phi_{jk}(\underline{s}, \underline{S})$ and $\psi_{ik}(\underline{s}, \underline{S})$, we define the following for each location k. Subscripts j and k are dropped for convenience.

C_E^W = unit cost for each unit shipped from location k to satisfy emergency demand or direct customer demand;

C_R^W = similar to above for replenishment demand within the region supported by location k;

C_E^O = similar to above for emergency demand of one unit from the next higher echelon in support of location k;

C_R^O = average cost, excluding setup, for each unit of replenishment orders shipped to location k from higher echelons;

K = setup cost for normal replenishment;

C_H = inventory holding cost per unit per period;

D_E^L = emergency shipment and direct customer demand to location k during lead time L, with mean μ_E and variance σ_E^2;

D_R^L = replenishment demand to location k during lead time L, with mean μ_R;

D_T^L = total demand to location k during lead time, L, that is, $D_T^L = D_E^L + D_R^L$, with mean μ, variance σ^2, and probability density function $f_T(.)$;

$C_E = C_E^O - C_E^W$;
$C_R = C_R^O - C_R^W$;
$Y_T = Z + D_T^L$;
$Y_E = Z + D_E^L$;
Z = undershoot variable of stock at the time a replenishment order is placed,

$$\text{Prob}\{Z=u\} = \frac{1}{\mu}\sum_{x\geq u+1} f_T(x) \text{ for } u<s, \text{ and}$$

$$\text{Prob}\{Z=s\} = \frac{1}{\mu}\sum_{v\geq s}\sum_{x\geq v+1} f_T(x);$$

$$Z_M = \lim_{s\to\infty} E(Z) = (\mu - 1 + \sigma^2/\mu)/2.$$

Given the demand distributions to location k (which are functions of the stocking policies of lower echelons), we can write $\phi_{jk}(\underline{s}, \underline{S})$ as a function of the (s,S) value for part j at location k only. For a particular part j, $\phi_{jk}(s_{jk}, S_{jk})$ can be approximated by the following function, where the subscripts j and k are again dropped for convenience [Cohen, Kleindorfer, and Lee 1988].

$$\mu\{K + (C_E - C_R)[E(Y_E - s)^+ + (\frac{\mu_E}{\mu})(Z_{M-E}(Z))] + C_R[E(Y_{T-S})^+ + Z_M - E(Z)]\}$$
$$\text{---------- (1)}$$
$$S - s + Z_M + E(Y_T - s)^+ + \frac{C_H}{2}[S - s + E(Z) + 2E(s - Y_T)^+]$$

The notation X^+ denotes Max$(0,X)$. The PAL (fill rate) for this part at location k is given by

$$\beta_{jk}(s,S) = \frac{[S - s + E(Z)]}{[S - s + Z_M + E(Y_T - s)^+]} \quad (2)$$

The service level constraint for product i, $\psi_{ik}(\underline{s},\underline{S})$ is thus given by

$$\psi_{ik}(\underline{s},\underline{S}) = \sum_{j\in M_i} \gamma_{ijk}(s_{jk}, S_{jk}) \geq PAL_{ik} \text{ ; where (3)}$$

$$\gamma_{ijk}(s_{jk},S_{jk}) = \frac{\mu_{ij}}{\sum_{k\in M_i}\mu_{ik}}\beta_{jk}(s_{jk},S_{jk}); \text{ that is}$$

γ_{ijk} = the contribution to the service of product i from part j at location k;
μ_{ij} = mean demand per unit time of part j in support of product i usage at location k; and
M_i = set of all parts that are used in product i.

A level-by-level decomposition algorithm is used to solve problem (I). The algorithm starts with locations at the lowest echelon, which face demands generated by part failures in customer installed machines. Hence there are no replenishment demands at this echelon. At each location k, the following problem is solved:

$$\text{Minimize} \sum_{j\in J}\phi_{jk}(s_{jk},S_{jk}) \quad \text{(II)}$$

subject to (3), for all $i\varepsilon I$.

The solution to problem (II) utilizes a greedy heuristic [Cohen, Kleindorfer, Lee 1988]. It is not necessary to search simultaneously with respect to both s and S. This follows since the order-up-to point can be expressed as a function of the reorder point in the following manner (again, the subscripts j and k are dropped for convenience).

$$D(s) = S - s$$
$$= \{\frac{2\mu}{C_H}[K + (C_E - C_R)[E(Y_E - s)^+ + \frac{\mu_E}{\mu}(Z_M - E(Z))] + C_R[E(Y_T - s)^+ + Z_M - E(Z)] + \Delta(s)]\}^{1/2} - Z_M - E(Y_T - s)^+, \text{ where}$$

$$\Delta(s) = [-(C_E - C_R)[E(Y_T - s)^+ + (\frac{\mu_E}{\mu})(Z_M - E(Z))] + (\frac{C_E}{2} - C_R)[E(Y_T - s)^+ + Z_M - E(Z)] - K]^+.$$

The idea behind this specification of S is that, for a given s, $D(s)$ minimizes ϕ_{jk}. The adjustment factor $\Delta(s)$ ensures the fill rate β to be nondecreasing in s. These results allow us to use a one dimensional greedy heuristic to solve problem (II). The algorithm uses the following steps.

(i) Set $s_{jk} \leftarrow 0$, $S_{jk} \leftarrow s_{jk} + \text{Max}\{1, D_{jk}(s_{jk})\}$, for all $j\varepsilon J$.
(ii) $j \leftarrow 1$.

(iii) Set $s_{jk} \leftarrow s_{jk} + 1$, $S_{jk} \leftarrow s_{jk} + 1 + \text{Max}\{1, D_{jk}(s_{jk} + 1)\}$.
Compute $\gamma_{jk} = \phi_{jk}(s_{jk} + 1, S_{jk}(s_{jk} + 1)) - \phi_{jk}(s_{jk}, S_{jk})$.
If $\gamma_{jk} \leq 0$, then go to (iii), otherwise, set $j \leftarrow j + 1$.
If $j \leq$ the number of parts in J, then go to (iii), otherwise, set $j \leftarrow j + 1$.

(iv) Define $I' = \{i \mid (3) \text{ is not satisfied}\}$. If $I' = \phi$, go to (vi), otherwise continue.

(v) Compute $\gamma_{jk} = \phi_{jk}(s_{jk} + 1, S_{jk}(s_{jk} + 1)) - \phi_{jk}(s_{jk}, S_{jk})$;
$\delta_{ijk} = \psi_{ijk}(s_{jk} + 1, S_{jk}(s_{jk} + 1)) - \psi_{ijk}(s_{jk}, S_{jk})$, for all $j \varepsilon J$; and
$\delta_{jk} = \sum_{i \varepsilon I'} \delta_{ijk}$.

Determine j' such that $\frac{\gamma_{j'k}}{\delta_{j'k}} = \text{Min}_{j} \left(\frac{\gamma_{jk}}{\delta_{jk}}\right)$.
Set $s_{j'k} \leftarrow s_{j'k} + 1$, $S_{j'k} \leftarrow s_{j'k} + 1 + \text{Max}\{1, D_{j'k}(s_{j'k} + 1)\}$. Go to (iv).

(vi) End.

This algorithm computes "near-optimal" (s,S) values for all parts at all locations in the first echelon. The algorithm then proceeds to the second echelon and repeats the calculations. At the second echelon locations, however, the incoming demand distributions are no longer exogenously determined. These distributions are consequences of the stocking policies computed for the first echelon, and their parameters must be computed explicitly. Three kinds of incoming demands can be seen at each location in the second echelon:

— Replenishment demands from locations at the lower echelons that are sourced by the second echelon location;
— Emergency demands generated by parts failure from locations in the lower echelon that are supported by the second echelon location;
— Failure-induced demands from customer machines that are supported directly by the location.

While it is possible, theoretically, to compute the exact distributions of these two types of "passed-up" demands via Markov chains analysis, approximation methods are used in Optimizer for computational efficiency. The central idea is to estimate the first two moments of the passed-up demand distributions for the respective locations from the lower echelon, based on parameters specific to the respective locations. Consider a part at a location of a lower echelon. Given the (s,S) policy used at this location, the part fill-rate β can be computed from (2). Let
η_E, V_E = mean and variance of emergency passed-up demands, respectively;
η_R, V_R = mean and variance of normal replenishment passed-up demands, respectively;
V_T = variance of total passed-up demands;
L = lead time of normal replenishment for this location.

The means and variances of the passed-up demands can be estimated as

$\eta_E = (1-\beta)\mu_E$;
$\eta_R = \mu - \eta_E$;
$V_E = \alpha_0(s+1)^{\alpha 1} L^{\alpha 2}(S-s)^{\alpha 3}(\mu_E)^{\alpha 4}(\sigma_E{}^2)^{\alpha 5}(\eta_E)^{\alpha 6}$;
$V_R = \lambda_0(s+1)^{\lambda 1} L^{\lambda 2}(S-s)^{\lambda 3}(\mu_E)^{\lambda 4}(\sigma_E{}^2)^{\lambda 5}(\eta_E)^{\lambda 6}$; and
$V_T = \kappa_1(V_E)^{\theta 1} + \kappa_2(V_R)^{\theta 2} + \kappa_3(V_E V_R)^{\theta 3}$.

where the parameters a_l, b_l, c_l and d_l, were estimated by regression techniques from a large number of numerical simulation runs [Cohen, Kleindorfer, Lee, and Tekerian 1986].

The incoming demand distributions to a particular location at the second echelon are then obtained by aggregating the first two moments from all the locations at lower echelons that are supported by this current location, as well as those from the direct customer demands at the location. The first two moments of the aggregated demands are then used to fit a Compound Poisson distribution with logarithmic compounding density. In this way, the incoming demand distributions for

the second echelon are specified.

Similar logic is then used in progressing from the second to next echelon and so on (Figure 3).

References

Cohen, M. A.; Kleindorfer, P. R.; and Lee, H. L. 1985, "Optimal stocking policies for low usage items in multi-echelon inventory systems," *Naval Research Logistics Quarterly*, Vol. 33, No. 1, pp. 17–38.

Cohen, M. A.; Kleindorfer, P. R.; and Lee, H. L. 1988, "Service constrained (s,S) inventory systems with priority demand classes and lost sales," *Management Science*, Vol. 34, No. 4, pp. 482–499.

Cohen, M. A.; Kleindorfer, P. R.; and Lee, H. L. forthcoming, "Near-optimal stocking policies for service-constrained single-echelon repair facilities," *Operations Research*.

Cohen, M. A.; Kleindorfer, P. R.; Lee, H. L.; and Pyke, D. F. 1989, "Multi-item service-constrained (s,S) policies for spare parts logistics systems," working paper.

Cohen, M. A.; Kleindorfer, P. R.; Lee, H. L.; and Tekerian, A. P. 1986, "Excess demand distributions for MEsS stocking policies in multi-echelon logistics systems," in *Inventories in Theory and Practice*, ed. A. Chikan, Elsevier Science Publishers, Amsterdam, pp. 655–667.

Feeney, G. J. and Sherbrooke, C. C. 1966, "The $(S-1,S)$ inventory policy under compound poisson demand," *Management Science*, Vol. 12, No. 5, pp. 391–411.

Graves, S. C. 1985, "A multi-echelon inventory model for a repairable item with one-for-one replenishment," *Management Science*, Vol. 31, No. 10, pp. 1247–1256.

Kamesam, P. V. and Tekerian, A. P. 1986, "OPTIMIZER: A multi-echelon inventory management system," internal paper, IBM Corporation, Greencastle.

Muckstadt, J. A. and Thomas, L. J. 1980, "Are multi-echelon inventory methods worth implementing in systems with low-demand rate items?" *Management Science*, Vol. 26, No. 5, pp. 483–494.

Sherbrooke, C. C. 1968, "METRIC: A multi-echelon technique for recoverable items control," *Operations Research*, Vol. 16, No. 1, pp. 122–141.

R. L. Sullivan, Director-Distribution, National Service Division, IBM, 400 Parson's Pond Drive, Franklin Lakes, New Jersey 07417, writes: "We could have implemented Optimizer in a manner that addressed only the inventory investment. We, however, applied the strengths of Optimizer along with its decision support capability to make fundamental changes to our network. This implementation resulted in improvements in service level, operating expenses and inventory investment. That is why we sometimes refer to Optimizer as the cornerstone of our parts logistics capability."