

# Lead Time Modeling and Acceleration of Product Design and Development

Yadati Narahari, *Member, IEEE*, Nukala Viswanadham, *Fellow, IEEE*, and V. Kiran Kumar

**Abstract**—New product development is an important business process and constitutes a major contributor to the business excellence of any manufacturing firm. Designing an optimized new product development process is an important problem in itself and is of significant practical and research interest. Lead time is an important performance metric for a product development organization. In this paper, we develop lead time models for product development organizations that involve multiple, concurrent projects with contention for human/technical resources. The objective is to explore how the lead times can be reduced using efficient scheduling, input control, load balancing, and variability reduction. The models are based on single class and multiclass queueing networks and capture important facets of a product development organization, such as: concurrent execution of multiple projects, contention for resources, feedback and reworking of project tasks, and variability of new project initiations and task execution times. Within the new product development process, we focus attention on the product design process, which is an important subprocess. Two product design organizations, which we call Company ABC and Company XYZ, provide the real-world setting for our model-based lead time reduction. First, we present a coarse, conceptual queueing network model of Company ABC and show how rapid performance analysis can be used to explore opportunities for accelerating the design process. In particular, we show how effective input control, process control, load balancing, and cross-functional work can cut the lead times. Next, we present multiclass queueing network models (re-entrant lines) for both the companies ABC and XYZ. The re-entrant line models show up certain scheduling issues pertaining to internal flows in the product design network. Using a class of fluctuation smoothing scheduling policies, we demonstrate how lead times can be reduced appreciably, without committing additional resources. The models presented are sufficiently generic and conceptual, and will be of much value in project planning and management in product design organizations and also more generally in product development organizations.

**Index Terms**—Design cycle time, lead time reduction, multiclass queueing networks, new product development process, product design process, re-entrant lines, resource contention.

Manuscript received July 15, 1998; revised April 13, 1999. This paper was recommended for publication by Associate Editor D. Wu and Editor P. B. Luh upon evaluation of the reviewers' comments. This work was supported in part by the Office of Naval Research and the Department of Science and Technology Grant N00014-93-1017.

Y. Narahari is with the Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560 012, India.

N. Viswanadham is with the Mechanical and Production Engineering Department, National University of Singapore, 119260 Singapore.

V. K. Kumar is with Wipro Systems Limited, Bangalore 560 012, India. Publisher Item Identifier S 1042-296X(99)08169-0.

## I. INTRODUCTION

**N**EW PRODUCT development is an important business process and a major contributor to the business excellence of any manufacturing firm. The new product development process encompasses the set of all activities beginning with the perception of a market opportunity and ending in the successful production of a quality product. Ulrich and Eppinger [1] provide an overview of all activities and functions encompassed by the product development process and identify five major phases: Concept development, system-level design, detailed design, testing and refinement, and production ramp-up. The system-level design and the detailed design phases together constitute the product design process. In this paper, we are concerned with the new product development process in general and the product design process in particular.

Product development is an interdisciplinary activity. It mainly involves three broad functions: marketing, design, and manufacturing. There are many challenges involved, including numerous opportunities for trade-offs at various stages, dynamics of technology, customer preferences, competition, pressures to bring the product to market quickly, and organizational realities. A product development organization is the scheme by which individual designers and developers are linked together into groups. If the emphasis is only on product design, we call such an organization as a product design organization. A product development organization executes a product development process while a product design organization executes a product design process. As already stated, in this paper, we discuss the product development process in general and the product design process in particular. We use the acronym PDP to mean a product design process or a product development process, based on the context. Similarly, we use the acronym PDO to refer to a product design organization or a product development organization as per the context. Also, the term DLT refers to (product) development lead time or (product) design lead time as the case may be.

Designing an optimized PDP is an important problem in itself and is of significant practical and research interest. An optimal PDP involves minimizing the time and resources required to deliver an outstanding new product to the customer. Development cycle time or lead time is by far the most important performance metric for a PDO. Introducing new products faster than competition allows companies several opportunities, such as setting new product standards, being a technical pioneer, being able to respond rapidly to customer

feedback, and ultimately realizing higher profit margins [2]. Thus product design/development managers are continuously looking for techniques to shrink the lead times of new product development projects. Several articles and books in recent times have addressed this problem of PDP acceleration, for example see the articles by Adler *et al.* [3]–[5], Zirger and Hartley [2], Hauptman and Hirji [6], and the books by Smith and Reinertsen [7], and Wheelwright and Clark [8].

A typical PDO, according to many published articles and case studies [3]–[5], has the following important features:

- 1) multiple, concurrent design or development projects in progress;
- 2) contention for human/technical resources in the organization;
- 3) feedback and reworking of project tasks;
- 4) randomness in task execution times and arrivals of new projects.

Recognizing the importance of lead times or project cycle times in such PDO's, it is our objective in this paper to develop lead time models of such PDO's using single class and multiclass queueing networks [9], [10]. The project dynamics in a PDO is slow compared to that of a production process on a factory floor. In this paper, we explore and demonstrate the validity of queueing network models in capturing the dynamics of PDO's to obtain important insights into lead time reduction. There have been related efforts in this direction, notably the work by Adler *et al.* [3], [5] (a more detailed review of relevant work follows in the next subsection). However, the lead time models presented in this paper are more detailed and realistic, offer more insights into PDO dynamics, and further help synthesize the contributions of relevant literature to the specific context of PDO modeling.

#### A. Review of Relevant Work

Process modeling as a means of improving industrial new product development was reported by Cooper [11] who developed a seven-stage model for describing the activities from product conception to product launch. Such a process model is shown to be a good foundation for accelerating the product development cycle. The articles by Adler *et al.* [3], [4] formulate a single class queueing network model for a real-world product development organization and conduct a rich variety of experiments with the model to bring out several strategies for speeding up the product development process. In fact, the case study of Company ABC that we discuss in this paper is taken from the study of a plastics industry found in these papers. In these papers, the authors argue that process models are better than project models for reasoning about and improving the performance of PDO's with multiple, concurrent, nonunique projects. The authors present a process model from which they create a single class queueing network model which is parameterized using a detailed set of measurements in a real-world organization. The model so created is simulated in a variety of resource allocation and decision-making scenarios and validated against the performance measured. Many what-if type of experiments

are conducted with the model to focus on various strategies for accelerating new product development projects.

Eppinger *et al.* have looked into the modeling of a product design process using signal flow graphs [12] and explored the effect of design iterations on the distribution of design project lead time. However, they do not consider concurrent execution of multiple projects in their study.

Alexander [13] came up with the idea that a queueing network framework could be used to capture congestion and resource contention features of a multiproject PDO. In this work, essentially QN models were used to identify bottleneck resources in the system and for effective management of resources.

Harrison and Loch [14] advocate the use of simple stylized queueing network models to study the quantitative impact of input conditions on the performance of any business process, so as to develop broadly applicable intuition about the process performance. They emphasize the effect of variability on system performance. Buzacott [15] has also suggested the use of conceptual queueing models in evaluating the effects of reengineering in organizations.

In the manufacturing arena, lead time reduction is an important subject. For example, Hopp *et al.* [16] emphasize the role of variability reduction as a means of reducing cycle times, using a queueing theoretic framework. The book by Hopp and Spearman [17] contains several ideas on lead time reduction, again from a queueing theoretic perspective. Suri [18] has explored the use of queueing models in the design and analysis of quick response manufacturing systems. There are also interesting case studies on lead time reduction, see for example, Bourland and Suri [19] and Bourland [20]. Many ideas embodied in these works can be used in the product design and new product development contexts.

Zirger and Hartley [2] look at many prominent techniques that have been employed for reducing the development lead times of new products and argue using case studies of several electronics companies that fast developers had teams that were cross-functional, dedicated, included fast time to market as a development goal, and overlapped development activities wherever possible. Millson *et al.* [21] have enumerated several intuitive principles for acceleration of product development projects. The use of concurrent engineering, process concurrency, and overlapped execution of project tasks in speeding up the product development process is investigated by Handfield [22], Hauptman and Hirji [6], and Krishnan [23], respectively.

Scheduling of design or development projects in order to optimize a suitably chosen objective function is also of interest here. The use of mathematical programming techniques has been explored by Belhe and Kusiak [24], [25] and by Liu *et al.* [26].

#### B. Objectives and Outline

The aim of this paper is to propose queueing networks as a conceptual process modeling tool for PDO's having multiple, concurrent, nonunique projects involving contention for resources and uncertainties of various kinds. The focus is on lead time modeling, motivated by the preeminence of lead time in providing competitive advantage to PDO's.

We explore queueing networks [9], [17] as the lead time models, motivated by their success in factory floor modeling. However, since there are both similarities and differences between a factory floor and a PDO, the fact that queueing models are successfully used in factory floor modeling, by no means, implies trivial extension to modeling the dynamics of multiproject PDO's. For example, the project dynamics in a product design organization or in a product development organization is slow compared to that of a production process on a factory floor. Furthermore, resources in a PDP are of a totally different nature, namely design engineers, technical resources, engineering workstations, etc. A primary objective of this paper is to show that in spite of these differences, queueing network models can faithfully capture the dynamics of project execution in a PDO at a certain level of abstraction, namely the level of abstraction of a product development manager or design manager. We would like to validate the use of such models in accurately capturing the dynamics of PDO's, leading to important insights into lead time reduction.

In this paper, we first show that coarse queueing network representations of PDO's can be used in designing a quick response PDO by carrying out rapid performance analysis under a variety of alternate scenarios. This uses essentially the same approach as in the articles by Adler *et al.* [3], [4] and adapts the lead time reduction strategies employed in the production process and business process context by Harrison and Loch [14], Hopp *et al.* [16], Hopp and Spearman [17], Bourland and Suri [19], and Bourland [20]. These coarse grain models can be solved using standard algorithmic techniques from product form queueing network theory.

Second, we show that such PDO's can be more accurately described by a probabilistic re-entrant line (which is a multiclass queueing network) [27], [28] and such a model can be used in obtaining additional insights into reducing development lead times. The use of probabilistic re-entrant lines makes the models far more realistic than the coarse single class queueing network models and enables subtle, internal scheduling issues to be revealed, motivating the use of fluctuation smoothing scheduling policies [29] and other queueing techniques in achieving lead time reduction. Such detailed models cannot always be solved using algorithmic techniques but may have to be analyzed only through simulation.

The lead time models in this paper are explored in the context of two case studies. The first one is that of a product development organization studied earlier by Adler *et al.* [4]. We call this organization as Company ABC. The second one is a PCB design organization in Bangalore, India, which we call as Company XYZ.

In Section II of this paper, we describe briefly the architecture of a typical multiproject PDO (of the type studied by Adler *et al.* [4]) and present a coarse queueing network model first and then a re-entrant line model. These models are quite conceptual and serve to describe aggregately any multiproject PDO with contention for resources. We also briefly review fluctuation smoothing policies in the context of multiclass queueing networks and bring out their relevance for multiproject PDO's. Section III focuses on lead time reduction applied to models of the company ABC. First, we bring

out how rapid performance analysis of the queueing network model can be used to estimate development lead times under a variety of resource allocation scenarios and to identify and exploit opportunities for reducing the lead times. We then focus on the probabilistic re-entrant line models and fluctuation smoothing policies. We show that by intelligently selecting the next task class to be processed, one can reduce lead times in a quite innovative way. In Section IV, we discuss the performance and lead time reduction in respect of the company XYZ, which is a multiproject design organization for PCB's. First we present a probabilistic re-entrant line model and next we explore model-based lead time reduction.

## II. QUEUEING NETWORK MODELS OF DYNAMICS OF A PDP

### A. Generic PDP

According to Ulrich and Eppinger [1], a product development process is the sequence of steps or activities that an enterprise employs to conceive, design, and commercialize a product. Some organizations define and follow a precisely defined PDP while many others may not even be able to describe their processes. Also, the processes of different organizations have some differences and unique characteristics. According to Cooper [11] who constructs a process model for a typical PDP, a well-defined process enables better control and tracking of product development or design projects. A generic PDP can be visualized as comprising five phases [1]: concept development, system-level design, detail design, testing and refinement, and production ramp-up.

In *concept development*, the needs of the target market are identified, alternative concepts are generated and evaluated, and a single concept is selected for further development. The *system-level design* phase arrives at the definition of the product architecture in terms of various subsystems and components, and typically generate a geometric layout of the product, a functional specification for the subsystems, and a process flow diagram for final assembly. The *detail design* phase includes the complete specification of the geometry, materials, and tolerances of all distinct parts in the product and preparation of a bill of materials and parts to be procured from suppliers. This phase also generates a detailed process plan and tooling for each part. In the *testing and refinement* phase, multiple prototype versions of the product are built and tested (alpha prototypes and beta prototypes). Detailed testing is done to determine whether or not the product will function according to specifications and whether or not it satisfies the customer's needs. Finally, in the *production ramp-up* phase, the product is made using the intended production system. The products produced during this phase are typically supplied to preferred customers and are carefully evaluated to identify any remaining flaws. Eventually, the product is launched into the market.

The product design process can be considered as a subprocess of the product development process. The emphasis in the product design process is on the *system design* and the *detail design* phases. In this paper, we are interested in both the development process and the design process and we use the common acronym PDP for both.

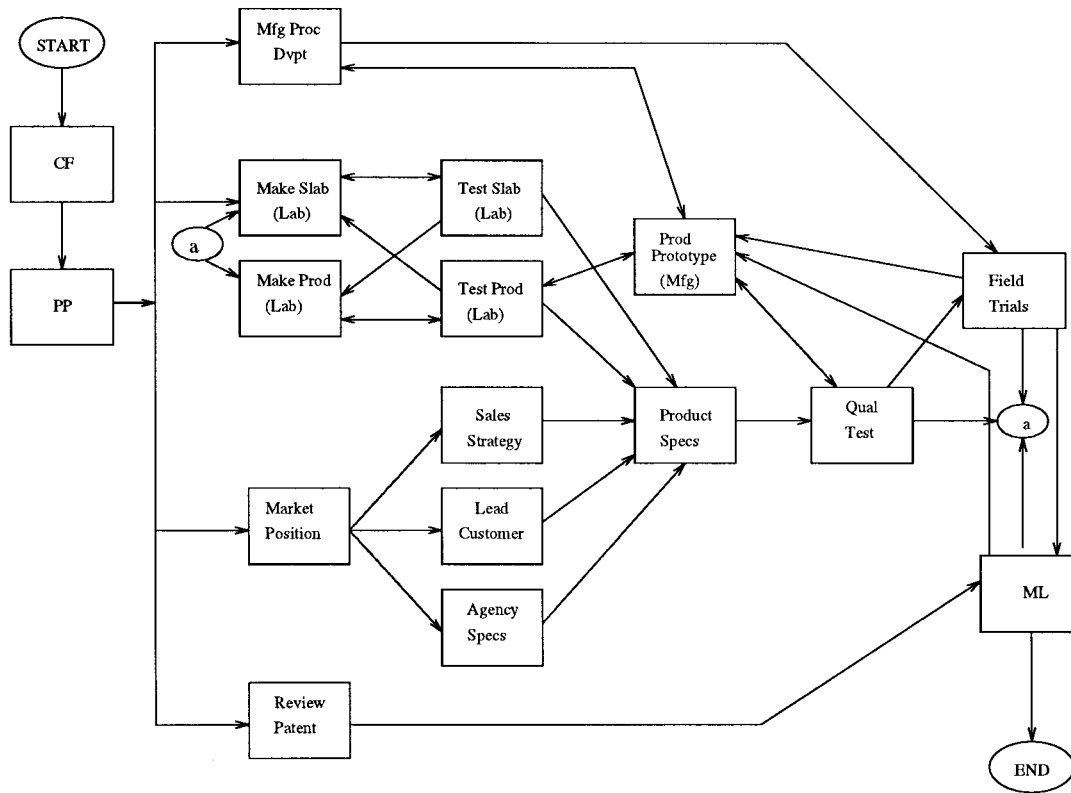


Fig. 1. Process flow diagram for the product development process in Company ABC.

**B. Multiproject PDO**

We now describe briefly the product development organization studied by Adler *et al.* [4] which serves as a typical example of a multiproject PDO. This particular PDO is involved in the development of plastics products which are either *new products* or *reformulations*. Since much of the effort is spent on new products, we shall only consider these for our modeling study here. The main resources in the PDO are the product and process engineers and technicians. Other resources are application engineers, product management personnel, manufacturing engineers, marketing and sales personnel, etc.

The activities in this organization can be broadly categorized into four phases: Phase 1 (concept/feasibility); Phase 2 (project plan/team formation); Phase 3 (product development); and Phase 4 (manufacturing standardization/product launch). Phase 3 of the process contains the bulk of the work in the PDP and as done in [4], is chosen for a detailed study here. Phase 3 involves 15 main activities. These activities are as follows.

- 1) Review patent.
- 2) *Manufacturing Process Development*: Determine process methods and equipment for all stages of production.
- 3) *Market Position*: Determine competitiveness of product and establish market position.
- 4) *Make Slabs*: Create samples in the form of slabs.
- 5) *Test Slabs*: Test slab prototype for conformance to material requirements.
- 6) *Make Product*: Make sample products from prototype materials.

- 7) *Test Product*: Test product prototype for conformance to product requirements.
- 8) *Make Product*: *Mfg*—Make product prototype in plant to uncover any manufacturing issues.
- 9) *Test Product*: *Mfg*—Test manufacturing prototype for conformance to product requirements.
- 10) *Sales Strategy*: Formulate sales strategy.
- 11) *Lead Customer*: Identify lead customers and determine their needs.
- 12) *Product Specs*: Identify product requirements and testing procedures.
- 13) *Field Trials*: Test product with lead customers.
- 14) *Agency Specs*: Determine whether product is subject to government regulations.
- 15) *Quality Testing*: Test product for conformance to all specifications.

In the above 15 activities, many are concurrent. For example, prototyping, manufacturing process development, marketing, and sales strategy can all progress at the same time. Also, for each activity, several different types of resources may be required simultaneously. For example, for the manufacturing process development activity, the following resources are required: product engineers, product technicians, process engineers, and process technicians. The precedence constraints and sequencing among these 15 tasks are shown in Fig. 1. In the above figure, CF indicates Phase 1 (concept/feasibility); PP indicates Phase 2 (project plan), and ML indicates Phase 4 (manufacturing and product launch).

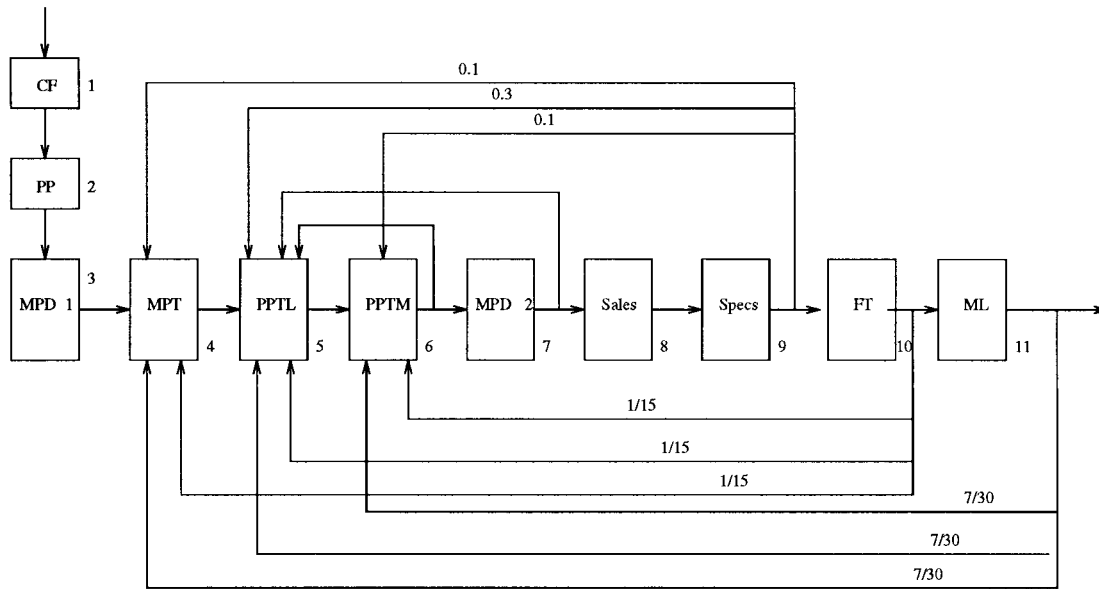


Fig. 2. Iteration structure for the product development process in Company ABC.

In the PDO under study, at any given time, many different development projects are in progress, each possibly in a different phase. This causes contention for engineering/human resources and results in delays at various points. We would like to model the resulting congestion. Often times, different phases of the same project could be contending for a given resource. For example, product engineers are required for slab prototyping, product prototyping, manufacturing process development, and quality testing activities of the same project.

An important aspect of a typical PDO is the need for feedback and reworking at most stages of the process. This is necessitated because design/manufacturability and such other problems may get revealed at various stages and this calls for repeating a subset of PDP activities all over again. One can characterize this iteration structure using feedback probabilities. Fig. 2 shows the iteration structure of typical projects. This diagram is derived from the data available in [4]. In this diagram, Phase 3 of the PDP is aggregated into 8 activities: MPD1 (manufacturing process development—1); MPT (material prototype and testing); PPTL (product prototype and testing in laboratory); PPTM (product prototype testing and manufacturing); MPD2 (manufacturing process development—2); sales; specs; and FT (field trials).

### C. A Single Class Queueing Network Model of Company ABC

Fig. 3 shows a coarse, single class QN model of the PDO in consideration. It is an 8-node open Jackson network with each node containing one server. It is an aggregated model in many ways. The model structure is derived from the iteration structure of Fig. 2, after aggregating MPT, PPTL, and PPTM into a single stage called PT and aggregating Sales and Specs into a single stage called SST. The three stages MPT, PPTL, and PPTM of Fig. 2 involve essentially prototyping and testing of the material and the product, and involve roughly the same set of resources. Similarly, sales and specs can also be

combined into a single stage. Each node thus represents an aggregated, parallel set of activities and the single server in each node is a functional or cross-functional team executing this set of activities. Table I describes the function of these eight nodes.

Multiple, concurrent development projects that are in progress in different stages represent the customers or jobs in the network. Note that new projects enter into the network at the CF node and successfully completed projects (successfully developed new product designs) leave the network from node ML. Each project undergoes a sequence of activities in the manner shown in the network. A project can visit a node several times due to reworks. The probability of returning to a previous stage after completing service at a particular stage is also shown for all appropriate feedback possibilities. For example, after the MPD2 stage, a project will come back to the PT stage for some rework or additional work and the probability of this feedback is 0.2. This means that with probability 0.8, the project will go to the next stage (SST). It is assumed that the processing time distribution at a node for a project visiting the node for the first time is identical to that corresponding to each subsequent visit. This assumption can be relaxed in the case of the re-entrant line model discussed in the next subsection. The routing is Markovian, that is independent of the previous history of the jobs.

A server here corresponds to a functional or cross-functional team of human resources who undertake the set of activities corresponding to that node. For example, the MPD1 node (see Table I) corresponds to three activities namely: first phase of manufacturing process development, review patent, and establish market position for the proposed project. The first of these is done by a set of product engineers, product technicians, process engineers, process technicians, and manufacturing engineers. The review patent activity is done by a set of product engineers, product technicians, and application engineers. The market position activity is handled by product

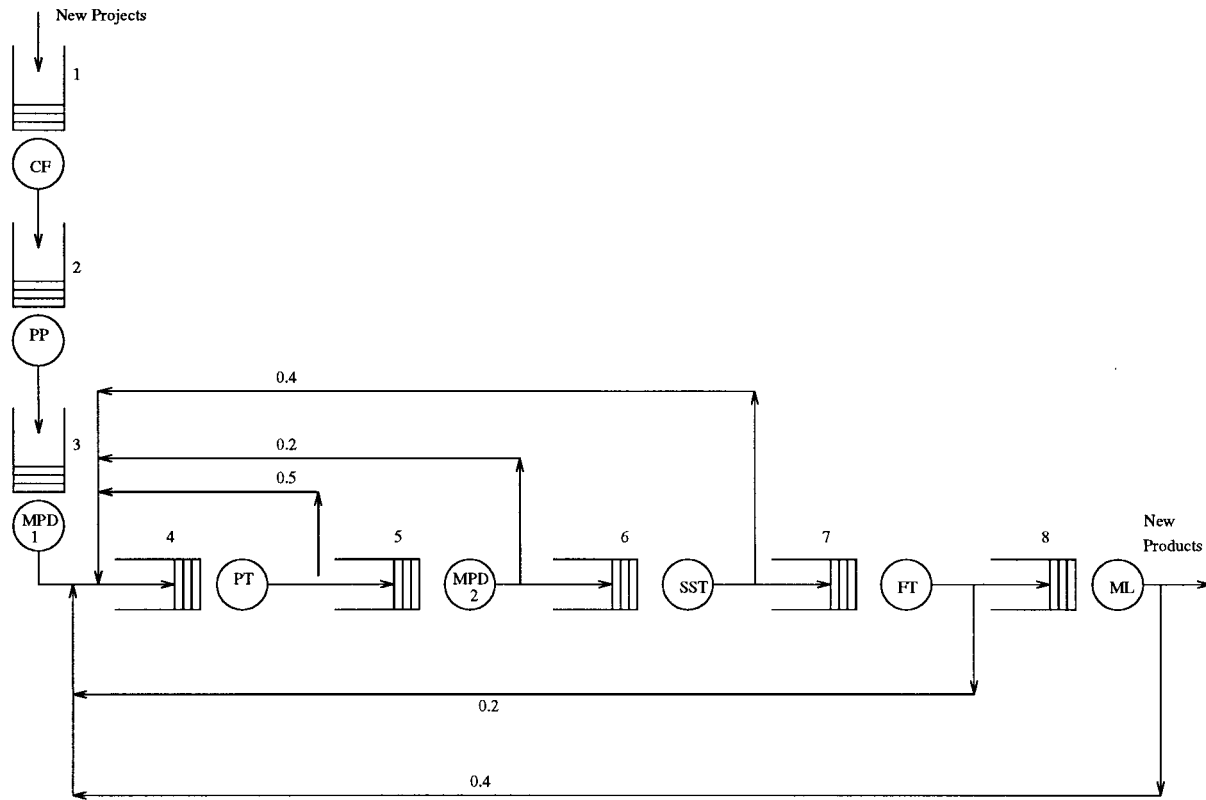


Fig. 3. Single class QN model of a multiproject PDO.

TABLE I  
DESCRIPTION OF THE NODES OF THE QN MODEL

CF	Concept and Feasibility
PP	Project Plan and Team Formation
MPD1	First Phase of Manufacturing Process Development, Review Patent, and Marketing work
PT	Prototyping and Testing of material and product
MPD2	Detailed Phase of Manufacturing Process Development
SST	Formulate sales strategy, finalize the product specs, and carry out quality testing
FT	Field trial with lead customers
ML	Manufacturing scale-up and product launch

management personnel and support staff. It is possible that a given engineer/technician is involved in two or more of the parallel activities corresponding to the given node. The server is thus a conglomerate of all these human resources and the service time corresponds to the most time-consuming activity among the parallel activities. It is also possible for the same resource, say a design engineer, to be part of two or more nodes in the queueing network model. In such a case, the percentage of time that the design engineer is known to spend in the individual nodes can be used to parameterize the capacity or the activity times of the individual nodes. This is the reason why the model is coarse and aggregates much detail. However, it is possible to parameterize such a stylized model and obtain useful insights by experimenting with the model.

For our experimentation, we assume that the interarrival time between successive new project initiations and also the

service times at various nodes are distributed according to a probability distribution whose mean and variance are known. These values can be obtained using measured data from a PDO. For example, the authors in [4] have obtained such data by interviewing the company personnel.

The QN model is thus described by the following parameters: number of nodes, mean and variance of interarrival time between successive new project initiations, mean and variance of the service time distribution at each node, routing matrix, and the scheduling policy to be followed at each node. We assume that nonpre-emptive FCFS policy is followed at each node. We also assume that the buffers at all nodes have infinite capacity.

#### D. Re-Entrant Line Model of Company ABC

The single class queueing network model just discussed, as already stated, aggregates much detail, however, as we will show in Section III, offers all the insights about lead time reduction as described in [3]. Furthermore, simple models such as these can be solved in quick time using fast queueing network solvers. To obtain more insights into PDO dynamics, we need to capture more details than in the single class model. In this section, we discuss one such model. This will be a multiclass QN and in particular can be described as a re-entrant line with probabilistic routing. Re-entrant lines [27] are appropriate for modeling queueing systems with distinct multiple job visits to service centers.

1) *Re-Entrant Lines*: In a re-entrant line, the parts visit the same server several times, at different stages of processing,

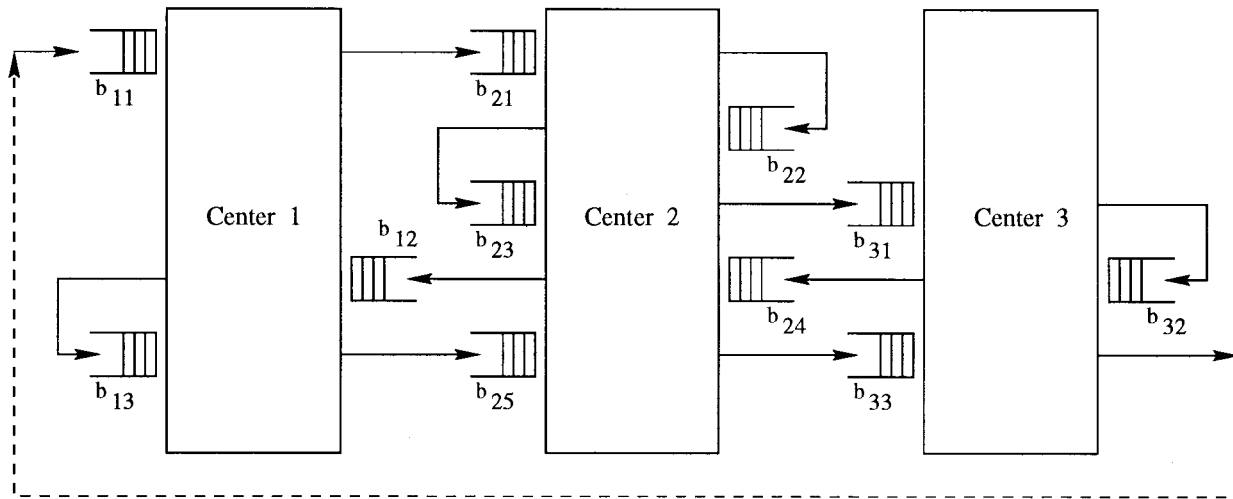


Fig. 4. Re-entrant line with three stations and 11 buffers.

before exiting the system, thus making the flow *re-entrant*. A re-entrant line can be described as follows. There is set of *service centers*  $\{1, 2, \dots, m\}$ . Service center  $i \in \{1, 2, \dots, m\}$  has  $n_i$  logical or physical buffers,  $b_{i1}, b_{i2}, \dots, b_{in_i}$ . For  $j \in \{1, 2, \dots, n_i\}$ , the buffer  $b_{ij}$  contains parts visiting service center  $i$  for the  $j$ th stage of processing. A part visits these buffers in a given sequence and any service center is typically visited several times in the route of a part.

Fig. 4 shows a typical re-entrant line with three service centers and 11 buffers. Parts enter the system at buffer  $b_{11}$  and visit the centers according to a deterministic route as shown. Finished parts emerge from center 3 after undergoing processing following a wait in  $b_{33}$ . Note that every part in this example line visits center 1 three times, center 2 five times, and center 3 three times.

In the re-entrant line shown in Fig. 4, the route of a job is deterministic. On the other hand, we can have re-entrant lines with probabilistic or Markovian routing [28], where we specify for each pair of buffers, say Buffer  $x$  and Buffer  $y$ , the probability  $P(x, y)$  which gives the probability that a job goes to buffer  $y$  next, after finishing its stay and service in buffer  $x$ . The model that we develop for a multiproject PDO will be of this type.

There are two important decisions that have significant effect on the performance of a re-entrant line. These are: *input release policies*, that specify when to release fresh jobs into the system; and *scheduling policies*, that specify which job to process next when a server becomes available.

2) *Scheduling in Re-Entrant Lines*: The scheduling problem in a re-entrant line becomes interesting because several parts at different stages of processing may be in contention with one another for service at the same service center. Several researchers have studied the issue of scheduling in re-entrant lines [27], [29]. Distributed scheduling policies based on buffer priorities and due dates have been formulated and investigated by Kumar [27], and Lu *et al.* [29]. Kumar [27] has investigated, among others, the following fixed buffer priority policies: FBFS (first buffer first serve) and LBFS (last buffer first serve). In a buffer priority policy, the buffers are assigned

different static priorities. After finishing the service on a part, the service center will pick up a part from the buffer having the highest priority (if one is available, of course). For example, in the case of LBFS, we order the  $n_i$  buffers of processing center  $i$  as  $b_{in_i}, b_{i,(n_i-1)}, \dots, b_{i2}, b_{i1}$  in decreasing order of priority. The next part selected for processing is the one that has finished most of its processing, and hence one with the least amount of processing remaining. Thus we may say that each processing center *myopically* tries to clear parts from the system as fast as possible. Other popular policies are due-date based policies such as EDD (earliest due date first) and LS (least slack first).

Fluctuation smoothing policies [29] are a special class of *least slack* scheduling policies [27]. In the least slack policies, for every job  $\pi$  that enters the network, there is an associated real number  $\beta(\pi)$ . Also to each buffer  $b_{ij}$ ,  $\forall i = 1, \dots, m$ ,  $j = 1, \dots, n_i$  there is associated a real number  $\gamma_{ij}$ , which is usually an estimate of the mean time a job in buffer  $b_{ij}$  will spend in the network before leaving the network. If a job is located in buffer  $b_{ij}$ , the slack  $s(\pi)$ , is defined by

$$s(\pi) := \beta(\pi) - \gamma_{ij}.$$

A least slack scheduling policy gives highest priority to the job  $\pi$  for which the slack is minimum. Whenever the server is to choose the next part after a service completion, it selects a part with the least slack. Now a particular choice of  $\beta(\pi)$  and  $\gamma_{ij}$  will give the particular least slack policy a unique capability. See [29] for a good overview of fluctuation smoothing policies. We will look at three such fluctuation smoothing policies.

*Reducing the Variance of Lateness*: Suppose each job  $\pi$  (project in our case) has an associated due-date  $d(\pi)$  (delivery time promised for the project). If we choose  $\beta(\pi) = d(\pi)$ , the resulting scheduling policy is found to reduce the variance of lateness of jobs and is called the fluctuation smoothing policy for variance of lateness (FSVL).

*Reducing the Variance of Cycle-Time*: If we make  $\beta(\pi)$  as the time at which the resulting scheduling policy is found to reduce the variance of cycle time of jobs and is called

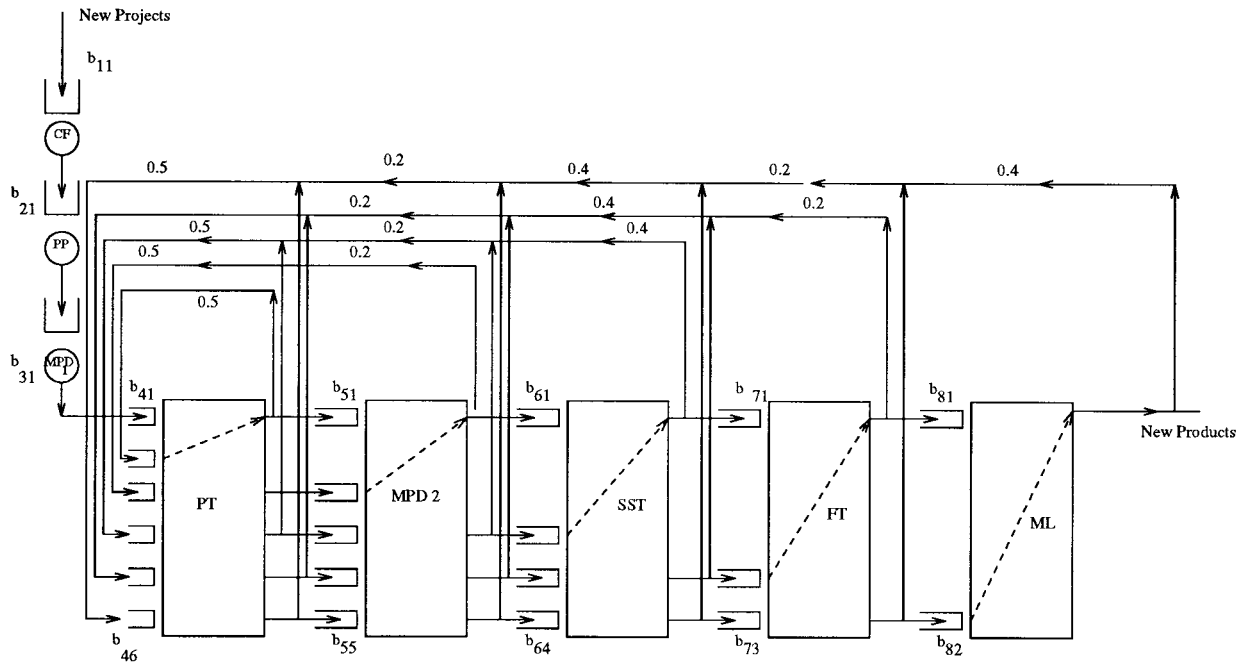


Fig. 5. Re-entrant line model of a multiproject PDO.

the fluctuation smoothing policy for variance of cycle time (FSVCT).

*Reducing the Mean Cycle-Time:* Suppose  $\pi$  is the  $n$ th job entering the network and  $\lambda$  is the average arrival rate of jobs into the network. If we choose  $\beta(\pi) = n/\lambda$ , the resulting scheduling policy is found to reduce the mean cycle time of jobs and is called the fluctuation smoothing policy for mean cycle time (FSMCT).

3) *Re-Entrant Line Model:* The re-entrant line model, shown in Fig. 5, contains eight stations (nodes) and 23 buffers. The multiple buffers at a given station contain projects which are revisiting that station for rework from different stages. For example, Station 4 (PT) has six buffers,  $b_{41}, \dots, b_{46}$ . Buffer  $b_{41}$  contains projects which are visiting this node for the first time. Buffer  $b_{42}$  contains projects which are visiting this station for rework after having undergone an operation or rework at this stage and before going to the MPD2 node. The jobs in buffer  $b_{43}$  correspond to those projects which are visiting this station for rework after having undergone an operation or rework at the MPD2 node. Likewise, we can describe the jobs in the other buffers also. The routing probabilities are now defined from one buffer to another. In this case, we have chosen the routing probability from a buffer  $b_{ij}$  to buffer  $b_{kl}$  as the routing probability in the single class QN model (Fig. 4), from the node  $i$  to node  $k$ .

The processing time distributions could be different for customers in different buffers. Also, the scheduling policy to be used at each station is now more complex and more interesting than in the case of the single class QN model. Thus, the present model refines the single class QN model in the following ways.

- 1) The jobs (projects) are now distinguished based on their history of progress in the network. At each node, we have buffers which will indicate whether the job is

coming there for the first time, which stage are they coming from, etc. The processing time distributions for jobs in different buffers at the same station could be different.

- 2) The routing probabilities now correspond to pairs of buffers instead of pairs of stations. This makes the routing more general and flexible.
- 3) Sophisticated scheduling policies can be defined to select the buffer and job to process next at a given station. The policies include: buffer priority policies, such as FBFS or LBFS [27], due-date based policies [27], and fluctuation smoothing policies [29]. This issue will be discussed in detail in Section IV.

### III. ANALYSIS AND LEAD TIME REDUCTION IN A PRODUCT DEVELOPMENT ORGANIZATION

Our aim here is to evaluate the performance of the QN model of Fig. 3 and the re-entrant line model of Fig. 5, and explore several opportunities for lead time reduction in Company ABC, which is a representative product development organization. The performance measure of interest will be the mean of product development lead time (DLT). Other performance measures such as variance of DLT, mean number of projects in progress, utilization of resources, can also be computed. The utilization of resources gives a highly aggregate picture of the utilization of functional or cross-functional teams and for this reason, is not considered here. In order to estimate the mean DLT, we need to analyze the model of Fig. 3. Since the model is a single class open Jackson network with single server stations, Markovian routing, general interarrival times, and general service times, we can use a package such as QNA [30]. A software tool has been built at the Indian Institute of Science for this purpose. The inputs to the package are:



TABLE II  
MEAN SERVICE TIMES FOR THE QN MODEL

Station	CF	PP	MPD1	PT	MPD2	SST	FT	ML
Mean service time (weeks)	2.0	1.0	2.5	1.0	2.5	1.5	1.5	4.0

TABLE III  
ROUTING PROBABILITIES FOR THE QN MODEL

Station	CF	PP	MPD1	PT	MPD2	SST	FT	ML
CF	0	1	0	0	0	0	0	0
PP	0	0	1	0	0	0	0	0
MPD1	0	0	0	1	0	0	0	0
PT	0	0	0	0.5	0.5	0	0	0
MPD2	0	0	0	0.2	0	0.8	0	0
SST	0	0	0	0.4	0	0	0.6	0
FT	0	0	0	0.2	0	0	0	0.8
ML	0	0	0	0.4	0	0	0	0

- 1) number of stations and number of servers at each station;
- 2) routing matrix;
- 3) mean and SCV (squared coefficient of variation) of interarrival times into the network;
- 4) mean and SCV of service times at each station.

The performance measures are computed using analytical formulae and for the models of the type in Fig. 3, it takes less than one tenth of a second even on a primitive PC to evaluate the model. Thus one can use this model to do rapid performance analysis; one can evaluate the performance over a wide range of input parameters and get a comprehensive idea of the system performance. Also, aggregate level decision alternatives can be evaluated and compared, leading to an optimal system configuration.

Based on the extensive data available in [4], we have chosen the mean service times for the eight servers in our model as shown in Table II. The units are in weeks. Also, based on the available data from [4], we have chosen the routing probabilities as in Table III. A typical entry  $P(i, j)$  in the routing table gives the probability of going next to station  $j$  after finishing service at station  $i$ . The aim is not to exactly mimic the model there but to create a credible enough model to experiment with. Assuming a Poisson arrival process with a mean interarrival time (IAT) of 16 weeks (that is, one new project initiated every four months on an average), the above base model gives a mean development lead time of 82.04 weeks if the processing times are assumed to be exponentially distributed with means as in Table II. On the other hand, if the processing times are assumed to be deterministic with the above values, the mean lead time drops to 57.68 weeks.

We now investigate different lead time reduction strategies using the model of Fig. 3. These strategies are process control; input control [3], [14], [16], [19]; load balancing [3], [14], [16], [19]; and cross-functional work [3]. We then use the more realistic model of Fig. 5 and explore the use of innovative scheduling to achieve lead time reduction.

TABLE IV  
LEAD TIMES WITH DIFFERENT SERVICE TIME VARIABILITIES. COV:  
COEFFICIENT OF VARIATION; MDLT: MEAN DEVELOPMENT LEAD TIME

		COV = 1.0		COV = 0.144		Const. proc. times	
IAT in weeks	MDLT weeks	Av. no. projects	MDLT weeks	Av. no. projects	MDLT weeks	Av. no. projects	
24	58.89	2.48	47.65	1.98	46.46	1.93	
22	61.84	2.81	50.62	2.31	49.50	2.25	
20	65.93	3.26	53.21	2.66	49.59	2.48	
18	71.995	4.08	57.95	3.21	55.91	2.92	
16	82.04	5.115	64.72	4.04	57.68	3.61	
15	90.135	5.84	68.07	4.53	64.69	4.312	
14	102.58	7.18	72.18	5.15	66.26	4.732	
13	124.76	8.31	84.26	6.48	79.43	6.12	
12	179.27	15.72	157.44	13.13	139.7	11.64	

#### A. Lead Time Reduction Through Process Control

Table IV shows the mean development lead times and mean number of ongoing projects in the organization assuming Poisson arrivals over a range of arrival rates, for three different service time scenarios. In the first, the service times are all assumed to have a coefficient of variation of 1.0 (coefficient of variation is the ratio of standard deviation to the mean of the random variable; a value of 1.0 corresponds to the exponential distribution); in the second, the processing times have a coefficient of variation of 0.144; and finally in the third, they are all assumed to be constant (coefficient of variation equal to zero). The values in Table II give the mean service times in all the cases. The results here indicate the increase in the mean DLT as a function of the arrival rates of new projects. What is of interest for us here is the reduction in the development lead time when the coefficient of variation of the service times reduces from 1.0 to 0.144 to 0 (deterministic service times). This is a direct consequence of the variability law of queueing theory [17], [29] according to which the mean waiting times in a queueing system are positively correlated to the variabilities of the interarrival times and service times. What we have done here is to reduce the variability in the service times, which is a consequence of tighter control over the service process. In the product development setting, this translates to systematic planning, efficient project management, and a well-defined and well-understood PDP.

#### B. Lead Time Reduction Through Input Control

Since waiting times are positively correlated with the variability of the arrival process, deterministic arrivals provide another opportunity for lead time reduction. However, since new product project initiations are often motivated by market opportunities, the arrival process here is subject to the market fluctuations. So, it is not feasible to have purely deterministic new project initiations. One way of reducing the arrival fluctuations is to operate the organization in a closed network mode, i.e., initiate a new project only when an existing one finishes. This ensures a constant population of projects inside the PDO and has a decrementing effect on the variance of arrivals. This constant population model is identical to the CONWIP (constant work-in-process) strategy that is popular in manufac-

TABLE V  
LEAD TIME REDUCTION THROUGH INPUT CONTROL.  
MLT: MEAN LEAD TIME; TR: THROUGHPUT RATE

Population of closed model	MLT in weeks	Associated Mean IAT	TR (per year)	MLT for open model
1	40.031	40.031	1.299	49.15
2	47.168	23.584	2.201	59.14
3	54.712	18.237	2.851	71.324
4	62.633	15.6582	3.32	84.54
5	70.897	14.1794	3.667	99.85
6	79.472	13.245	3.926	117.92
7	88.326	12.618	4.121	139.01
8	97.431	12.1788	4.271	164.27
9	106.76	11.862	4.384	195.01
10	116.29	11.629	4.472	232.61
11	125.995	11.454	4.54	278.12
12	135.86	11.321	4.593	333.14
13	145.86	11.219	4.634	405.62
14	155.97	11.141	4.667	493.46
15	166.19	11.08	4.693	601.32

turing control [14], [17]. Another way is to admit a new project only when the number of projects is below a threshold or the total workload in the system is below a particular threshold.

Table V shows the effect of operating the PDO in a fixed population mode. The routing probabilities are as in Table III and the mean service times are as in Table II. The service times are assumed to be exponentially distributed. The first column gives the current population of the network in the closed network mode; the second column gives the mean DLT for the corresponding population; the third column gives the mean interarrival times that are consequent on having the corresponding population in the closed network; the fourth column provides the corresponding throughput rate of successful projects completed per year; the final column gives the mean DLT if the PDO is operated as an open network with these throughput rates (note that in a stable open network, the arrival rate is the same as the throughput rate of the network). On close observation, the virtues of operating in the fixed-population mode become clear. For example, with a population of five, the mean DLT is 70.897 weeks; the throughput rate is 3.667 completed projects per year; to obtain this throughput rate using an open mode of operation will entail a mean DLT of 99.85 which is more than 30% higher. Thus lower lead times are achieved for a specified throughput rate and conversely higher throughput rates can be obtained for specified cycle time. However, a closed mode will entail rejection of some projects and also continuous availability of fresh projects for initiation. This calls for close coordination between the PDO and the customers (could be internal from within the same business unit or external) who generate the fresh product development projects. With proper coordination, sharing of information, and upfront planning between the PDO and potential customers, input control of the fixed population type can be implemented, fairly accurately, if not exactly.

### C. Lead Time Reduction Through Load Balancing

Another oft-used technique for reducing the congestion inside a queueing system is to identify the bottleneck resources

TABLE VI  
LEAD TIME REDUCTION THROUGH LOAD BALANCING.  
MDLT: MEAN DEVELOPMENT LEAD TIME

Without Load Balancing				With Load Balancing		
IAT weeks	Utili. of MPD2	Utili. of SST	MDLT weeks	Utili. of MPD2	Utili. of SST	MDLT weeks
16	0.6667	0.319	64.72	0.607	0.540	61.31
15	0.705	0.337	68.07	0.657	0.588	64.95
14	0.750	0.360	72.18	0.683	0.604	65.61
13	0.811	0.389	84.26	0.730	0.648	75.15
12	0.917	0.441	157.2	0.778	0.696	78.02

and ease the congestion by providing additional capacity to the bottleneck. In the PDO setting, this might mean taking away some capacity from lightly loaded stations if the technical/human resources involved are interchangeable. This happens, for example with stations 5 and 6, in our model (see Table I). Station 5 (manufacturing process development—2) involves the work of product engineers, process engineers, product technicians, process technicians, manufacturing engineers, etc. Station 6 (sales, specs, and testing) involves the participation of product engineers, process engineers, product management people, and technical services personnel. Since some of these resources are interchangeable between MPD2 and SST functions, the load across the two can be balanced. Table VI presents the results of an experiment to explore the effect of such bottleneck analysis and load balancing through resource sharing and reallocation. For five different arrival rates, the open queueing model with uniformly distributed service times (with a coefficient of variation of 0.144 each) is evaluated without and with this load balancing applied to stations 5 and 6. For example, with mean interarrival time (IAT) equal to 12 weeks, the utilization of station 5 is 0.917 while that of station 6 is 0.441. This results in a mean DLT of 157.14 weeks. By allocating some capacity of station 6 to station 5, the mean service time of station 5 is reduced whereas that of station 6 gets increased. We have for instance assumed an increase in capacity of station 6 by 10% when the capacity of station 6 is decreased by 20%. For such a reallocation, the utilizations are found to be 0.778 and 0.696, respectively, and the mean DLT dramatically decreases to 78.02 weeks.

### D. Lead Time Reduction Through Cross-Functional Work

Next we present how by reducing feedbacks and rework through increased cross-functional participation, the lead times can be brought down considerably. Table VII shows some results. It is quite a standard argument in concurrent engineering literature [31] that by making product development work more cross-functional, the rework loops are reduced. At the same time, each individual activity will need more resources, more discussion, more interaction, and consequently more time. In our experiment, we have increased the processing times of stages 4–8 by 20% to account for the additional time entailed by increased cross-functional work and assumed that the associated feedback probabilities are reduced by 10%. This is only an empirical experiment that shows the effect

TABLE VII  
LEAD TIME REDUCTION THROUGH CROSS-FUNCTIONAL WORK

Less cross-functional work			More cross-functional work			
IAT weeks	Utili. of MPD2	Utili. of SST	MDLT weeks	Utili. of MPD2	Utili. of SST	MDLT weeks
16	0.6667	0.319	64.72	0.661	0.325	60.99
15	0.705	0.337	68.07	0.691	0.342	64.53
14	0.750	0.360	72.18	0.799	0.391	76.58
13	0.811	0.389	84.26	0.829	0.406	87.94
12	0.917	0.441	157.1	0.912	0.447	141.1

of cross-functional work. The experiment can be repeated easily if actual data is available. Table VII shows the mean DLT before and after this adjustment in the model, again assuming that the processing times before adjusting the model are uniformly distributed with a coefficient of variation of 0.144 each. The gains in lead time performance are quite clear, except possibly in one case. We can also use the model to predict the minimum reduction in feedback probabilities required to justify a concurrent engineering effort (modeled in terms of specifies increase in processing times) to improve lead time performance. If we are able to estimate credibly the change in the input parameters, we can predict lead time improvements in an accurate way.

#### E. Lead Time Reduction Through Effective Scheduling

The coarse-grain, single class models discussed so far are innovative in two ways.

- 1) They are much simpler than the simulation model presented by Adler *et al.* but offer virtually the same insights.
- 2) This model can be solved using fast queueing network solvers like QNA and so will enable rapid performance analysis.

However, the models aggregate much detail and in order to obtain deeper insights, we need to develop more detailed model.

Now we consider the re-entrant line model of company ABC, shown in Fig. 5, and demonstrate that better performance can be achieved by intelligently scheduling internal work in the product development network. We look at four indices of performance: mean lead time, variance of lead time, average lateness, and variance of lateness. We consider six different scheduling strategies: FCFS, FBFS, LBFS, FSMCT, FSVCT, and FSVL.

Table VIII shows the results for a typical scenario. We have assumed Poisson arrivals and exponential processing times. The processing times of a job on its first visit to a station are assumed to be the same as in the single class QN model discussed earlier. On subsequent visits to a station, the processing time means are assumed to half of their original values. The results in Table VIII are obtained by a detailed simulation under each policy, where each simulation is run to complete about 10 000 projects and the performance measures are computed after deleting an appropriate number of initial

TABLE VIII  
PERFORMANCE OF COMPANY ABC UNDER FLUCTUATION SMOOTHING POLICIES

	IAT = 16 weeks	IAT = 15 weeks	IAT = 14 weeks
<b>FCFS Policy</b>			
Mean DLT (Weeks)	126.62	141.24	148.22
Variance of DLT	130.04	185.21	194.12
Average Lateness	-64.81	-72.15	-79.26
Variance of Lateness	135.12	173.12	189.63
<b>FBFS Policy</b>			
Mean DLT (Weeks)	141.04	155.12	179.23
Variance of DLT	149.12	155.12	198.12
Average Lateness	-81.41	-93.12	-100.66
Variance of Lateness	169.66	191.43	214.15
<b>LBFS Policy</b>			
Mean DLT (Weeks)	109.6	152.5	168.34
Variance of DLT	132.4	186.7	198.12
Average Lateness	-66.12	-74.41	-83.62
Variance of Lateness	132.6	175.34	191.45
<b>FSMCT Policy</b>			
Mean DLT (Weeks)	99.50	110.78	124.58
Variance of DLT	117.2	129.45	139.34
Average Lateness	-49.12	-57.61	-68.12
Variance of Lateness	110.23	123.43	143.22
<b>FSVCT Policy</b>			
Mean DLT (Weeks)	125.57	140.55	144.85
Variance of DLT	92.29	108.18	106.08
Average Lateness	-42.14	-48.66	-64.12
Variance of Lateness	101.45	123.56	140.43
<b>FSVL Policy</b>			
Mean DLT (Weeks)	125.58	140.612	144.63
Variance of DLT	117.23	138.93	144.91
Average Lateness	-78.42	-89.33	-95.66
Variance of Lateness	83.12	90.15	99.89

transients. A confidence level of 0.95 is considered for these results. In the simulation, arriving projects (arriving jobs) are assigned due dates which are randomly drawn from a carefully chosen window. Lateness of a completed project is the completion time subtracted from the due date and is usually negative. The lateness values obtained have only a relative significance and do not have any absolute significance (due to the random due dates).

Applying fluctuation smoothing to improve lead time performance is a very attractive alternative to some of the techniques explored earlier in this paper. Here, we do not need to add any additional capacity to the system resources, or incur any overheads such as rejecting some projects. We only choose the way in which to prioritize work corresponding to internal flows or internal processes. In the model, at any given station, we distinguish between work by the history of sojourn in the network and due dates that different jobs are carrying and schedule in one of three ways depending on what is required to be minimized.

The results show the positive influence of the fluctuation smoothing policies. In the context of a PDO, the scheduling strategies have implications on how the internal subprocesses of various current projects are scheduled. The attractive aspect of employing these policies arises by virtue of not having to commit any additional resources or personnel for various project tasks. We are only prioritizing individual work elements in an appropriate way.

#### IV. CASE STUDY OF A PCB DESIGN ORGANIZATION

Here, we consider a PCB (printed circuit boards) design company located in Bangalore, India, which we call Company XYZ, and model its lead time performance through a re-entrant line model. Using this case study, we demonstrate the applicability of simple lead time reduction strategies based on the re-entrant line model. The emphasis will be on the use of clever scheduling of internal work in the design organization.

##### A. Description of the Design Organization

The Company XYZ is a medium scale PCB design company, involved in the design of single sided, double sided, and multilayer boards. The design of a PCB varies in complexity depending on the complexity of the circuit (simple, less dense, medium dense, and highly dense); types of components used (through hole or surface mount); and the technology used for fabrication (analog or digital). It is found that almost 90% of the time, the PCB's designed have medium complexity and the study here focuses on the performance with respect to this class of PCB's. Thus a job in the organization will refer to a design project for such a PCB.

The design process for a typical design project in Company XYZ consists of the following phases.

1) *Order Processing (OP)*: When a customer arrives with a requirement for designing a specific PCB, the order processing department will prepare the quotation and initiate the acceptance of the order. It is found that on an average, an order is received every two days but there is substantial variability in the interorder arrival times. Order processing typically takes half a person-day (about 4 h time). Order processing typically involves collecting some additional information concerning the PCB type to be designed.

2) *Net List Generation (NLG)*: This involves generating a detailed layout and schematic drawing for the PCB. This is done either automatically (NLG-A) through CAD tools (if the design is complex) or manually (NLG-M) by writing the netlist (for simple designs). Usually one design engineer is assigned this task. It is found that 95% of the jobs are done through automatic means and only about 5% of the jobs go through manual netlist generation. It takes about 8 h (1 person-day) to generate the netlist automatically (i.e., for complex designs) and about 12 h (1.5 person-days) to generate it manually (i.e., for simple designs). There is some variability in the processing times here. The output of this stage is immediately conveyed to the customers for their feedback and it is found that about 30% of the automatically done layouts are returned for reworking and about 40% of manually done layouts are returned for reworking. The rework times

are about half of the original processing times. The reworked designs are immediately checked by concerned customers and it is possible a few of these (5% on an average) come back for a second rework (which takes much less time). It is found that no layout comes back for a rework for a third time.

3) *Component Placement (CP)*: This phase involves generation of placement of components and generation of check plot and drill data report. This is again entrusted to a design engineer. On an average, this activity takes about 10 h (1.25 person-days) and the document that is generated is reviewed by the concerned customer. It is approved first time in about 75% of the cases. In the rest of the cases, reworking is done the average rework time being 3 h. 94% of reworked jobs are approved by the customers and about 6% of jobs go for a second rework, which takes hardly an hour. All jobs that are twice reworked are found to be approved by the concerned customers.

4) *Component Routing (RT)*: This phase involves generation of routing to provide the required interconnections among the components. A designated design engineer will handle this task. On an average, this activity takes about 12 h (1.5 person-days) and the document that is generated is reviewed by the concerned customer. It is approved first time in about 90% of the cases. In the rest of the cases, reworking is done the average rework time being 4 h. All the reworked jobs are found to be approved by the customers.

5) *Generation of CAM Data (CAM)*: Here, using the placement and routing information, and drill data, the CAM related data is generated for facilitating optimal fabrication sequence. A design engineer will take about 4 h for this activity and it is found that 95% of jobs are approved by the customers first time. The reworking usually takes about 2 h and all reworked jobs are found to be approved. The design is now ready for fabrication.

##### B. Re-Entrant Line Model for Company XYZ

Since rework or design iteration is a prominent feature of each design project in the above organization, a re-entrant line model is a natural choice for capturing the dynamics of project flow. Fig. 6 depicts a re-entrant line model for the above design organization. The model has 6 stations and 14 buffers. The reentrancy models in a faithful way the reworking that happens at various stages and also allows distinct processing times for different rework stages. Each station corresponds to human and technical resources assigned that task. For example, station NLG-A corresponds to the activity of automatic generation of netlist. The server here is a design engineer entrusted this task. It is to be noted that this server will need some passive resources such as a computer workstation, generation tools, etc. It is assumed that such technical resources are always available whenever needed.

The parameters of this model are: distribution of interarrival times of successive design projects; distribution of processing times at each one of the 14 buffers; and routing probabilities. Fig. 6 shows all the routing probabilities and also the mean values of all the processing time distributions. The mean values are in person-days.

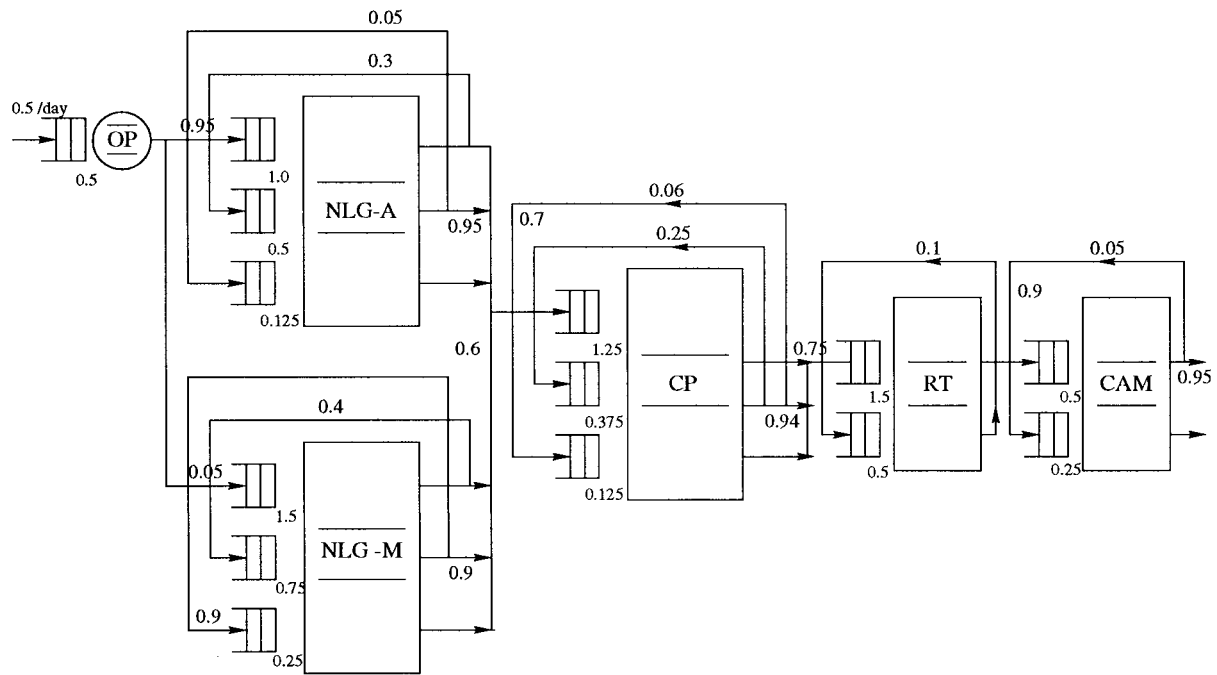


Fig. 6. Re-entrant line model of a multiproject design organization.

TABLE IX  
PERFORMANCE OF AN OPEN RE-ENTRANT LINE MODEL OF COMPANY XYZ UNDER DIFFERENT SCHEDULING POLICIES. MLT: MEAN LEAD TIME

Scheduling Policy	MLT (person-days)	Lead Time (variance)	Lateness (average)	Lateness (variance)
FCFS	15.21	9.28	-16.29	28.43
FBFS	17.78	17.68	-20.85	38.177
LBFS	14.303	9.236	-16.474	28.07
FSMCT	11.989	9.17	-14.71	26.95
FSVCT	15.09	7.14	-12.78	20.188
FSVL	15.56	10.332	-23.16	14.57

TABLE X  
PERFORMANCE OF A CLOSED RE-ENTRANT LINE MODEL OF COMPANY XYZ UNDER DIFFERENT SCHEDULING POLICIES (POPULATION = 5). MLT: MEAN LEAD TIME

Scheduling Policy	MLT (person-days)	Lead time (variance)	Lateness (average)	Lateness (variance)
FCFS	10.51	4.389	-10.305	18.705
FBFS	10.67	5.404	-10.28	18.984
LBFS	10.02	3.484	-10.875	17.983
FSMCT	8.06	4.176	-6.176	14.181
FSVCT	10.08	2.78	-10.69	13.61
FSVL	10.334	4.174	-10.85	8.05

We have to also specify the scheduling policy followed at each station. We consider the following six policies as in Section III-E: FCFS, FBFS, LBFS, FSMCT, FSVCT, and FSVL.

C. Analysis and Lead Time Reduction

Table IX shows the performance measures: mean cycle time, variance of cycle times, average lateness, and variance of lateness for the base model of the design organization XYZ, shown in Fig. 6, under six scheduling policies of interest. It is assumed that the interarrival times and the processing times are independent exponential random variables. The simulation is carried out for about 100 000 events and 95% confidence intervals are considered after initial transient deletion. Each design project that enters the network is assigned a random due date that is uniformly distributed around its predicted exit time from the network. Note that FSMCT and LBFS perform better than the other policies in reducing mean cycle times. The mean cycle times obtained here assume that customer feedback at intermediate stages of design (NLG-A, NLG-M, CP, RT, and CAM) is instantaneous. In actual practice, this is

not true and one needs to add an expected customer feedback time in order to arrive at the correct picture. It often happens that the customer reviews a design document overnight and the feedback is ready by the beginning of next day. In this latter case, the feedback time is to be counted as zero since the cycle times are expressed in terms of person-days. The mean cycle time values that were obtained here for the FCFS policy were found to be within 10% of those usually observed for a typical project in the Company XYZ.

The main observation that one can make from Table IX is that scheduling plays an important role in lead time reduction. Clever scheduling of internal work can accelerate the flow of design projects without committing additional human or technical resources.

Table X shows how by exercising simple input control, one can accelerate the projects in Company XYZ further. The results correspond to a closed mode of operation, wherein a new design project is taken up only when an earlier design project has just been completed. The processing times are identical to the ones in the base model (exponentially distributed). A

TABLE XI  
PERFORMANCE OF COMPANY XYZ UNDER THE FSMCT POLICY

	Expo. proc. times	Uniform proc. times	Constant proc. times
<b>OPEN MODEL</b>			
Mean DLT (days)	11.989	8.567	6.682
Variance of DLT	9.17	6.53	3.97
Throughput Rate	3.5	3.5	3.5
<b>CLOSED MODEL</b>			
Mean DLT (days)	8.02	7.43	6.58
Variance of DLT	4.176	3.754	2.48
Throughput Rate	3.37	3.84	4.25

population of five design projects is considered here since it corresponds approximately to the throughput rate of the base model (which is an open model). Also, it was found that there were usually about five concurrent design projects in progress in the Company XYZ. The closed mode of input control leads to all-round improvement in the performance of the system. In particular, the FSMCT policy leads to a cycle time that is very low compared to that in the other policies and the FSVCT policy leads to very low variance of cycle time.

The performance of the company XYZ can be improved further, as shown in Table XI, by tighter control on the processing times. Since the FSMCT policy is the policy of choice for lead time reduction, we investigate the performance of the network under the FSMCT policy, assuming exponential processing times, uniform processing times (with 20% variability around the mean), and deterministic processing times (zero variability). As expected, a closed mode of operation with five concurrent projects scheduled according to the FSMCT policy with 100% control over processing times leads to the best lead time performance. In fact, the mean lead time of 6.58 days achieved in this setting is less than half of what one can achieve with the common FCFS policy with open input control (See Table IX also).

## V. CONCLUSION

In this paper, we have treated the product development process and one of its subprocesses, the product design process, in the same manner as a production process and modeled a multiproject PDO as a queueing network. The project dynamics in a product design organization such as Company XYZ or in a product development organization such as Company ABC are slow compared to that of a production process or a factory floor process. However, we have demonstrated the validity of queueing network models in capturing the slow dynamics adequately enough to reveal insights into lead time reduction. One limitation of the single class model is its coarseness but our aim is aggregate performance analysis and lead time reduction. The re-entrant line model is more detailed and captures interesting scheduling issues that can arise in a multiproject PDO.

Needless to say, the models presented are still not representative of all the details and distinctive aspects of a PDO. Only a comprehensive simulation model provides a partial answer to the problem of creating a faithful replica of a given PDO. What we have attempted here is to come up with a good analytical model that captures certain important performance determinants of a PDO and use the model toward a deeper understanding of project management issues through rapid performance analysis. The models capture the effect of various lead time reduction strategies at the level of abstraction of a product development manager or a product design manager. Such models can be used by managers in aggregate project planning and project management. In order to use such models in the detailed planning of projects in a multiproject PDO, one has to enrich the models and also the analysis techniques. In this sense, the paper certainly throws open several interesting issues for further investigation.

The models described can become the foundation of a software tool that can be used by managers in multiproject PDO's. Such a product is currently under development at the Indian Institute of Science.

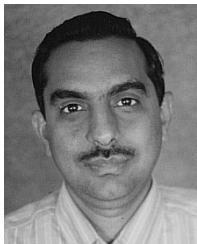
## ACKNOWLEDGMENT

The authors would like to thank S. Aithal and E. Manjunath, Central Manufacturing Technology Institute, Bangalore, India, for providing the data for Company XYZ, P. Nagaraju, for the simulation results which were obtained by a software package he developed, and K. Ravikumar and M. Azaraiah for preparing the figures.

## REFERENCES

- [1] K. T. Ulrich and S. D. Eppinger, *Product Design and Development*. New York: Mc-Graw Hill, 1995.
- [2] B. J. Zirger and J. L. Hartley, "The effect of acceleration techniques on product development time," *IEEE Trans. Eng. Manage.*, vol. 43, pp. 143–150, May 1996.
- [3] P. S. Adler, A. Mandelbaum, V. Nguyen, and E. Schwerer, "From project to process management: Strategies for improving development cycle time," Tech. Rep., Sloan Sch. Manag., Mass. Inst. Technol., Cambridge, Mar. 1994.
- [4] ———, "From project to process management: An empirically-based framework for analyzing product development time," *Manag. Sci.*, vol. 41, no. 3, pp. 458–484, Mar. 1995.
- [5] ———, "Getting the most out of your product development process," *Harvard Bus. Rev.*, vol. 22, no. 2, pp. 134–152, Mar./Apr. 1996.
- [6] O. Hauptman and K. K. Hirji, "The influence of process concurrency on project outcomes in product development: An empirical study of cross-functional teams," *IEEE Trans. Eng. Manage.*, vol. 43, pp. 153–164, May 1996.
- [7] P. Smith and D. Reinertsen, *Developing Products in Half the Time*. New York: Van Nostrand Reinhold, 1991.
- [8] S. C. Wheelwright and K. B. Clark, *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality*. New York: Free Press, 1992.
- [9] N. Viswanadham and Y. Narahari, *Performance Modeling of Automated Manufacturing Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
- [10] L. Kleinrock, *Queueing Systems: Volume 1—Theory*. New York: Wiley, 1995.
- [11] R. G. Cooper, "A process model for industrial new product development," *IEEE Trans. Eng. Manage.*, vol. 30, no. 1, pp. 2–11, Jan. 1983.
- [12] S. D. Eppinger, M. V. Nukala, and D. E. Whitney, "Generalized models of design iteration using signal flow graphs," *Res. Eng. Design*, vol. 9, pp. 112–123, 1997.
- [13] R. Alexander, "Scheduling and resource allocation methodologies for fast product development in a multi-product environment," Tech. Rep., M.S. thesis, Sloan Sch. Manag., Mass. Inst. Technol., Cambridge, 1990.

- [14] J. M. Harrison and C. H. Loch, "Operations management and reengineering," Tech. Rep., Graduate Sch. Business, Stanford Univ., Stanford, CA, Dec. 1995.
- [15] J. A. Buzacott, "Commonalities in reengineered business processes: Models and issues," *Manag. Sci.*, vol. 42, no. 5, pp. 768–782, May 1997.
- [16] W. J. Hopp, M. L. Spearman, and D. L. Woodruff, "Practical strategies for lead time reduction," *Manufact. Rev.*, vol. 3, no. 2, pp. 78–84, 1990.
- [17] W. J. Hopp and M. L. Spearman, *Factory Physics: Foundations of Manufacturing Management*. New York: McGraw-Hill, 1996.
- [18] R. Suri, "Using queueing networks to support quick response manufacturing," in *Proc. IIE Conf.*, May 1996.
- [19] K. Bourland and R. Suri, "Spartan industries," Tech. Rep., Case Study, Tuck Sch. Business, Dartmouth College, Hanover, NH, 1992.
- [20] K. Bourland, "Cellular production at Spartan," Tech. Rep., Teaching Case, Tuck Sch. Business, Dartmouth College, Hanover, NH, 1994.
- [21] M. R. Millson, S. P. Raj, and D. Wilemon, "A survey of major approaches for accelerating new product development," *J. Prod. Innovat. Manag.*, vol. 9, pp. 53–69, 1992.
- [22] R. B. Handfield, "Effects of concurrent engineering on make-to-order products," *IEEE Trans. Eng. Manage.*, vol. 41, pp. 384–393, Nov. 1994.
- [23] V. Krishnan, "Managing the simultaneous execution of coupled phases in concurrent product development," *IEEE Trans. Eng. Manage.*, vol. 43, pp. 210–217, May 1996.
- [24] U. D. Belhe and A. Kusiak, "Scheduling design activities with a pull system approach," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 15–21, Feb. 1996.
- [25] ———, "Dynamic scheduling of design activities with resource constraints," *IEEE Trans. Syst., Man Cybern. A*, vol. 27, pp. 105–111, Jan. 1997.
- [26] F. Liu, P. B. Luh, and B. Moser, "Scheduling of design projects with resource constraints and uncertain number of design iterations," in *Proc. AIM-97, Adv. Intell. Mechatron.*, Tokyo, Japan, 1997.
- [27] P. R. Kumar, "Re-entrant lines," *Queueing Syst.: Theory Appl.*, vol. 13, pp. 87–110, 1993.
- [28] Y. Narahari and L. M. Khan, "Modeling re-entrant manufacturing systems with inspections," *J. Manufact. Syst.*, vol. 16, no. 1, Feb. 1997.
- [29] S. H. Lu, D. Ramaswamy, and P. R. Kumar, "Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants," *IEEE Trans. Semiconduct. Manufact.*, vol. 7, pp. 374–388, Aug. 1994.
- [30] W. Whitt, "The queueing network analyzer," *Bell Syst. Tech. J.*, vol. 62, no. 9, pp. 2779–2815, Sept. 1983.
- [31] D. A. Gatenby, P. M. Lee, R. E. Howard, K. Hushyar, R. Layendecker, and J. Wesner, "Concurrent engineering: An enabler for fast, high quality product realization," *AT&T Tech. J.*, pp. 34–47, Jan./Feb. 1994.



**Yadati Narahari** (M'99) received the M.E. degree in computer science and the Ph.D. degree, both from the Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India, in 1984 and 1988, respectively.

He is an Associate Professor with the Department of Computer Science and Automation, Indian Institute of Science, Bangalore. In 1992, he visited the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, to work on dynamic and stochastic scheduling of manufacturing systems. In 1997, he spent a sabbatical year at the National Institute of Standards and Technology, Gaithersburg, MD, where he was instrumental in developing an integrated approach for design for quality of electromechanical products using object-oriented models and statistical tolerancing methodologies. In 1996, he completed *Lecture Notes on Data Structures and Algorithms*, which is being developed as a web-based digital textbook. He cowrote the textbook *Performance Modeling of Manufacturing Systems* (Englewood Cliffs, NJ: Prentice-Hall, 1992). His research interests are broadly in the areas of stochastic modeling and scheduling methodologies for future factories, supply chain management, enterprise resource planning, and object oriented modeling.

Dr. Narahari received the Indo-U.S. Science and Technology Fellowship in 1992. In 1997, he was a Guest Editor for two special issues of the journal *SADHANA*, on the topic of advanced manufacturing systems. He is currently an Associate Editor of the *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION* and the *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*.



**Nukala Viswanadham** (SM'86–F'93) received the Ph.D. degree in control systems from Indian Institute of Science, Bangalore, Karnataka, India, in 1970.

He is a Professor with the Department of Mechanical and Production Engineering, National University of Singapore, Singapore. He was with the Department of Computer Science and Automation, Indian Institute of Science, Bangalore, from 1967 to 1999. He has held visiting appointments at the University of New Brunswick, Fredericton, N.B., Canada, University of Waterloo, Waterloo, Ont., Canada, General Electric Corporate Research and Development Centre, University of Connecticut, Storrs, and Stanford University, Stanford, CA. He was a GE Research Fellow during 1989. He co-authored *Reliability in Computer and Control Systems* (North-Holland, 1987) and *Performance Modeling of Automated Manufacturing Systems* (Englewood Cliffs, NJ: Prentice-Hall, 1992) and authored *Analysis of Manufacturing Enterprises* (Norwell, MA: Kluwer, 1999). His current research interests include modeling, control and management of manufacturing and business enterprises, supply chain management, and new product development.

Dr. Viswanadham received the 1996 Indian Institute of Science Alumni Award for excellence in research. He is a member of the Indian National Science Academy, Indian Academy of Sciences, Indian National Academy of Engineering, and the Third World Academy of Sciences. He is an Editor of the *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*. He is also on the editorial boards of the *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, the *Journal of Manufacturing Systems*, and the *Journal of Franklin Institute*.

**V. Kiran Kumar** received the M.E. degree in computer science from the Indian Institute of Science, Bangalore, in 1994.

Since 1994, he has been working with the Wipro Systems Limited, Bangalore. He has research interests in the area of performance modeling and simulation of systems. He has participated in the development of a wide range of software products in networking and electronic commerce.