

A Decision Analytic Framework for Evaluating Concurrent Engineering

Ali A. Yassine, Kenneth R. Chelst, and Donald R. Falkenburg

Abstract— This paper quantifies key issues with regard to concurrent engineering through the use of risk and decision analysis techniques that enable us to better understand, structure, and manage the design process. In concurrent engineering, the information structure of a design process does not usually imply the execution patterns of the corresponding design tasks. On the contrary, this gap between the information structure and execution patterns is the essence of concurrent engineering and its basic advantage over traditional sequential design. In this paper, we relate the structure of information flow in a design process to three different execution strategies: sequential, partial overlapping, and concurrent. The risks of excessive task iterations or redesigns associated with each execution pattern are probabilistically modeled. Risk and decision analysis methodology is used to determine the best execution strategy and the optimal overlapping policy for a set of activities given their information structure. Applying this theoretical framework to a real-world design application of an automotive cylinder block suggested a potential 18% reduction in development cycle time.

Index Terms— Concurrent engineering, design process structure, information requirements, partial overlapping, probabilistic modeling, risk and decision analysis.

I. INTRODUCTION

SIMPLY stated, concurrent engineering (CE) is the incorporation of downstream factors and concerns into the upstream phase of product development [14]. This should lead to a shorter product development time, better product quality, and lower manufacturing costs [13]. It is concerned with the availability of information to all agents involved in the design of a product. Its basic concern is to make available all relevant information to an agent involved in the design process before the design task is begun. The ability (of the design agent) to act on the information as soon as it becomes available is yet another important dimension of CE. Unfortunately, for most engineering tasks all relevant information required by a specific task cannot be made available at the start of that task. Therefore, CE requires the maximization of such information and the ability to share and communicate useful information on a timely basis. In this paper, we discuss an analytical approach

to CE based on the modeling of information requirements among design tasks.

Previous research in design process modeling [5] identified three general types of information dependencies among design tasks (also called design process information structure): independent; dependent; and interdependent. Two or more tasks are said to be independent if they could be performed or executed simultaneously with no interaction among them. On the other hand, if one task requires the output of another to be started, then the two tasks are dependent. Furthermore, if task *A* needs information from task *B* and task *B* needs information from task *A*, then the two tasks are said to be interdependent or coupled. In addition to the information structure, Krishnan *et al.* [9] identified three different execution patterns (also called design process schedule) that can be used to execute a set of related tasks: sequential; partial overlapping; and concurrent. In the sequential strategy, a downstream task begins after the completion of the upstream task. The information transfer, if required, is only transferred in its final form at the end of the upstream task. In the partial overlapping strategy, or simply overlapping, the downstream task can be started before the complete finish of the upstream task. In this case, the upstream information is transferred to the downstream task twice. The first time, incomplete or partial information is transferred at the start of the overlapping process and a second time at the end of the upstream task. The concurrent execution strategy is similar to the partial overlapping strategy in terms of commencing the downstream task before the complete finish of the upstream task, and particularly simultaneously with it. However, the information transfer is bi-directional and more frequent than the partial overlapping case. The frequent, bi-directional information exchanges between the tasks are assumed to be facilitated via a CE team or a cross-functional team [17], [23]. It is noteworthy to distinguish between parallel (or complete overlapping) and concurrent execution of tasks. While parallel execution can be treated as a special case of partial overlapping, concurrent execution does not only involve task start timing, but also the frequency and richness of information transfer between tasks.

In the first part of the paper, we consider each information dependency type separately and probabilistically model the risks involved in the design process if the tasks were executed in any one of the three execution strategies. The goal is to select the one strategy that will result in a reduction in the development lead time. The probabilistic modeling allows us to build the decision analysis model presented in the

Manuscript received November 27, 1995; revised September 30, 1997. Review of this manuscript was arranged by Department Editor D. Gerwin.

A. A. Yassine is with the Center for Technology, Policy, and Industrial Development, Massachusetts Institute of Technology, Cambridge, MA 02143 USA.

K. R. Chelst and D. R. Falkenburg are with the Industrial and Manufacturing Engineering Department, Wayne State University, Detroit, MI 48202 USA.

Publisher Item Identifier S 0018-9391(99)03069-X.

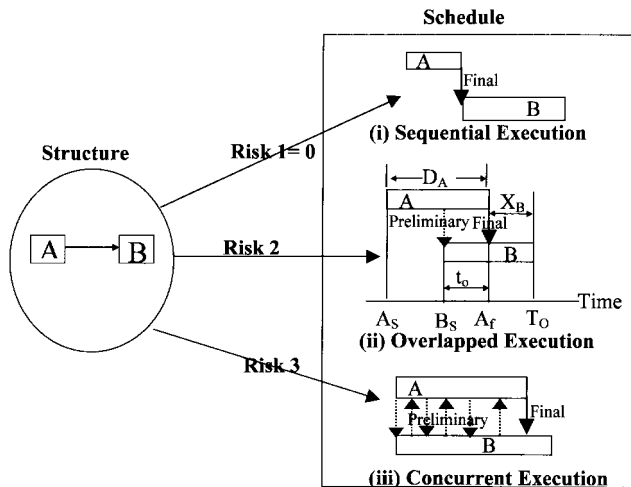


Fig. 1. Mapping between both models.

second part of the paper (Section V) where the three execution strategies are evaluated and an optimal strategy is selected.

In CE, the information structure of a design process does not usually imply the execution patterns of the corresponding design tasks. On the contrary, this gap between the information structure and execution patterns is the essence of CE and its basic advantage over traditional sequential design. For example, if two tasks are dependent, it does not necessarily mean that they have to be executed sequentially. They can be executed as sequential, partially overlapped, or even concurrent tasks depending on the amount and type of information transferred between them and the associated levels of risk underlying each execution strategy. Fig. 1 shows the mapping between the design process structure and schedule. The figure shows two dependent tasks A and B , and shows their three possible execution strategies. Each different execution strategy puts the dependent task (task B in this case) at a different level of risk depending on how much the information link between the two tasks is violated. For example, the sequential strategy shown in (i) of Fig. 1, which honors the information link between the two tasks, resulted in no risk of repetition or redesign taken on the behalf of task B . When task B is partially overlapped with task A , as demonstrated in (ii) of Fig. 1, using preliminary design information from task A , then the magnitude of risk increases, as compared to the sequential execution strategy. The magnitude of risk is determined by the dependency strength and the amount of information transfer required between task A and B . Finally, task B can also be executed in parallel with task A ; however, putting it at maximum risk of repetition [(iii) in Fig. 1]. A similar figure with different risk levels can be constructed if the two tasks were independent or interdependent in the structure domain.

A considerable body of research exists on the problem of reducing the lead time of product development [2], [5], [6], [9]–[12], [14], [15], [22], [24], [25]. Clark and Fujimoto [2] introduced the concept of “integrated problem solving” and emphasized the importance of exchanging preliminary design information in product-development situations instead of late release of complete information. They also pointed out that the cultural barriers associated with such strategy are related

to hostile environment and poor communication. Lee [14] concluded that overlapping requires more than just pushing up the start time of the downstream task. The upstream task must also absorb some of the downstream problem-solving activities directly into its own activities, resulting in an increase in its information processing load, while the downstream load will be decreased correspondingly. The net result is a synergistic reduction in the total time required for completing the two tasks combined. Both of the above mentioned works focused mainly on the organizational aspects necessary to reduce the product development lead times and provided guidelines rather than an analytical framework or tool.

On the other hand, Krishnan *et al.* [9]–[12] developed a model and a framework for overlapping, in the case of dependent tasks, which distinguished between different overlapping policies: iterative; distributive; divisive; and preemptive overlapping. The appropriateness of each overlapping policy is discussed based on two attributes: upstream information evolution and downstream sensitivity. Furthermore, the authors formulated the problem of iterative overlapping as a mathematical program whose solution provided the optimal timing of preliminary information release. Our work differs from Krishnan [9]–[12] in several respects. First, while Krishnan’s [11], [12] model included only the case of dependent tasks, our model captures the three different information dependencies and maps them to three different execution strategies (not only overlapping). Second, we conceptualize the overlapping strategy differently. Our model assumes that overlapping requires only one preliminary information exchange and Krishnan’s model allows for multiple preliminary information exchanges. Finally, while Krishnan’s model is deterministic, our model provided a purely stochastic interpretation of both overlapping and iteration.

Smith and Eppinger [22] described the process of task iterations as a Markov chain where the decision maker can choose the ordering of the initial attempt at completing the interdependent tasks. The expected length of the Markov chain is then evaluated and used to determine which of the ordering is preferable. In another paper [24], Smith *et al.* used methods of feedback control theory to analyze and identify controlling features of the iteration process. The method called for the determination of the eigenvalues and eigenvectors for the work transformation matrix (WTM). The WTM contains the strength of dependency between the tasks. This will determine the rate and the nature of the convergence of the design process. The metrics produced by the model represents the work done by each of the tasks during the iteration process. Our work differs from Smith [22], [24] in two respects. First, the WTM method presented by Smith and Eppinger [24] is a deterministic model. Our model is stochastic in nature. Second, Smith’s model assumes fully parallel iterations and does not allow for overlapping, which our model included.

Loch and Terwiesch [15] conjectured that the gain from overlapping activities must be weighed against the delay from downstream rework as a result of preliminary information. Communication reduces that negative effect of rework at the expense of communication time. They modeled the rate of change in preliminary information, used during the overlap-

ping period, as a nonstationary Poisson process. The average rate of upstream changes is referred to as the “overall level of uncertainty.” The nonhomogeneity of the process represented the progress (uncertainty reduction) of the upstream task. Finally, the authors formulated the problem as an optimization problem which yields: the optimal overlapping magnitude and the optimal communication policy (how often to communicate preliminary information during the overlapping phase). While Loch’s model examined a situation with only a dependent information structure, our model, in contrast, examined all three information dependency structures. Further, Loch’s model did not include randomness in the duration of the individual tasks and only considers it in formulating the amount of rework. In this paper, we treated all elements of the design process stochastically including task durations, amount of downstream rework, and the number of iterations occurring both upstream and downstream.

Risk and decision analysis (R&DA) is a structured methodology that allows the modeling and representation of risk and uncertainty in a decision-making process [3], [18], [20]. According to Howard [7], the first step of the decision problem analysis is the formal representation of the decision maker’s real situation, which he calls a “decision basis.” The elements of a decision basis are: the alternative strategies the decision maker faces; the information pertinent to the decision situation; and the preferences of the decision maker. The alternative strategies may be readily available or may require further analysis to generate. The information establishes the connection between alternative strategies and their respective outcomes. This can be any model, relationship, or probability assignment. The uncertainty surrounding the occurrence of the different possible outcomes is represented by probability assignments. A discussion of the different probability assignment and assessment techniques is outside the scope of this paper and the interested reader is referred to [16] and [21]. Finally, the preferences of a decision maker are the consequences or values associated with choosing an alternative strategy given a state of nature [20]. After establishing the decision basis, a decision maker selects the best alternative strategy which has the highest expected gain or lowest expected loss. The first part of this paper (Sections II–IV) derives and structures the information needed for the decision analysis approach, namely, the outcomes associated with each alternative execution strategy. The second part of the paper (Section V), utilizes this framework to build a decision model for the CE decision problem. In addition, the decision analysis framework allows us to perform sensitivity analysis on all model parameters and assumptions.

The CE decision problem is modeled as a task management decision problem where the decision maker is investigating the best execution strategy for a set of related tasks. Under the concurrent strategy, there is the risk of excessive downstream iterations resulting from the violation of a dependency link between the two tasks. Under the overlapping strategy, preliminary upstream information was utilized by the downstream task and this information may change by the time the upstream task is completely finished. Therefore, the downstream task faces the risk of an additional iteration to incorporate the latest

upstream information. In general, the earlier we begin overlapping, with less reliable upstream information, the greater the risk of a downstream redesign. Therefore, the overlapping strategy consists of a range of overlapping times which reflect different levels of risk, allowing us to quantitatively determine the optimal overlapping policy. Furthermore, the decision problem becomes more complex when introducing development cost factors into the analysis. In this case, the decision problem becomes a tradeoff between development time and cost [1].

In the next three sections, we will discuss the impact of the three different execution strategies on independent, dependent, and interdependent tasks, respectively, probabilistically modeling the risk elements in each case. In Section V, we apply the framework and equations developed in Section IV for the most complex case where the tasks are interdependent. Finally, in Section VI, we present our conclusions.

II. INDEPENDENT TASKS

The following notation and definitions hold for any number of engineering tasks. However, the analysis and derivations, in this paper, will be restricted to only two tasks: an upstream task, task A , and a downstream task, task B . Extensions to this basic modeling approach that will include more than two tasks would require more assumptions and developments.

The nominal duration of any task i is a random variable D_i which is the time required to finish task i if all its requisite information is available. The nominal duration includes the setup time and all iterations internal to task i . Internal iterations reflect the randomness in the duration of a task resulting from all factors other than the availability of precedence information. On the other hand, external iterations refer to the process of task repetition due to the unavailability or change in predecessor information.

In the rest of this section we will assume that tasks A and B are independent. If both tasks A and B are executed sequentially and in any order, then the sequential cycle time (T_s) will be simply the linear sum of all task durations [1], [15]

$$T_s = D_A + D_B. \quad (1)$$

The amount of risk taken in this case by either tasks is zero since the sequential strategy did not violate any relationship or information dependency between both tasks.

The partial overlapping strategy can also be utilized. It requires, as mentioned earlier, a two-stage information transfer. Since the two tasks are independent, the information exchanged by the tasks is irrelevant and any change in one task’s information does not impact the other. The risk of a design change in the downstream task due to changes in the upstream delivered information, from its preliminary form to its final form, is nonexistent. The overlapped cycle time (T_o) will be [15]: $\text{maximum}(D_A, D_B) \leq T_o \leq (D_A + D_B)$ depending on the amount of overlap.

Finally, when both tasks start concurrently, then the concurrent cycle time (T_c) for this independent design problem becomes equal to the duration of the longest task [1]

$$T_c = \text{maximum}(D_A, D_B). \quad (2)$$

Using a similar argument to the overlapping strategy, we can conclude that the concurrent strategy does not place the two tasks at any risk. Changes in the exchanged information do not impact the other task.

Comparing the above three cycle time equations, and considering that the risk level in any case is zero, we conclude that the optimal execution strategy is the concurrent strategy which yields the minimal cycle time. It is important to note that even though the tasks are independent, information can still be shared between the tasks. However, neither task will benefit from this information sharing process. In contrast, information sharing is costly [6], [15], increases the duration of the development process, and adds no value to the tasks involved in this case. Therefore, we will assume that whenever overlapping or concurrency is used for independent tasks, they will be treated as dummy information flows, where no time is consumed in communication. This will keep the definition of overlapping and concurrency consistent with the rest of the paper and at the same time will make sure that the minimum cycle time is attained.

III. DEPENDENT TASKS

In the dependent configuration, let us assume, without loss of generality, that task B is dependent on task A .

A. Sequential Execution

In the sequential execution strategy, task B cannot be started until task A is finished completely. Therefore, the sequential cycle time (T_s) for this design problem is expressed by (1).

The cycle time expressed by (1) is maximal, but no risk is taken on the behalf of task B because the information dependency between the two tasks was not violated: finalized design information was transferred to task B .

B. Partial Overlapping

The sequential cycle time could be decreased if we allow some violation to the information link between the two tasks. That is, allowing task B to start at some time before task A is completed. However, if partial overlapping occurs, there is a chance that the work accomplished by task B , before task A is completely finalized, will be partially or completely wasted due to the possibility of a design change in upstream information during the overlapping period. If, for example, the upstream design information does not change much during partial overlapping, and that change has a small influence on the downstream task, then partial overlapping is an attractive strategy [12], [15]. However, if either the magnitude or the impact of upstream design changes during overlapping is large, then partial overlapping is a less attractive option. In the worst case, partial overlapping will result in no improvement to (1), but an increased cost as compared to the sequential strategy [1].

In this paper, the partial overlapping model relates the expected value of the downstream task duration to the percentage of completion of the upstream task. Our hypothesis is that there exists a nonlinear relationship between the amount of overlapping and the duration of the dependent downstream

task. The overlapping strategy is shown in (ii) of Fig. 1, where A_s and A_f are the start and finish times of task A . Similarly, B_s and B_f are the start and finish times of task B . If the downstream task B is to start prior to the finish time of upstream task A , then the duration of the upstream task, D_A , will not be affected but the duration of the downstream task is likely to vary depending on B_s . Therefore, the total duration of task B consists of the overlapping magnitude (t_o) and a random variable X_B , where $(D_B - t_o) \leq X_B \leq D_B$. The random variable, X_B , reflects the amount of rework in the downstream task after receiving finalized upstream information. This rework is due to the possibility of upstream design changes during partial overlapping. Equation (3) describes the random variable X_B and (4) expresses the cycle time for the overlapped dependent design problem

$$\begin{aligned} X_B &= f(B_s) \quad \text{if } B_s < A_f \quad \text{i.e. } t_o > 0 \\ &= D_B \quad \text{if } B_s \geq A_f \quad \text{i.e. } t_o = 0 \end{aligned} \quad (3)$$

$$T_o = D_A + X_B. \quad (4)$$

If all the work accomplished by the downstream task, before time A_f , is completely wasted, then $X_B = D_B$ and (4) becomes $(D_A + D_B)$. In this case, the overlapped cycle time is equal to the purely sequential cycle time. If some of the downstream work (performed before time A_f) was useful, then $X_B < D_B$ and partial overlapping resulted in an improvement to (1). In this model, the finish time of the downstream task is greater than or equal to the finish time of the upstream task: $B_f \geq A_f$. This will guarantee two things. First, the downstream task has to receive final upstream information (to verify the preliminary information used earlier) before it terminates. Second, no more than one downstream iteration will be performed after the first planned downstream iteration.

Given that task B starts at time ($B_s < A_f$), it will be at a disadvantage because of incomplete upstream information. The magnitude of this disadvantage is captured by the expected value of X_B : $E(X_B)$. The expected value for the cycle time, in this case, is

$$E(T_o) = E(D_A) + E(X_B). \quad (5)$$

$E(X_B)$ can be determined by using the concept of knowledge accumulation. Knowledge accumulation curves were first introduced by Sobieski [25] and then utilized by Krishnan [9], [10] and Yassine [26]. The knowledge accumulation curve of the upstream task A describes the percentage of knowledge accumulated by task A as a function of D_A . This allows the estimation of how much the upstream information is close to its final form and the probability of an upstream design change. The probability of a later design change in upstream information is calculated as $1 - p$, where p is the percent of knowledge accumulated by the upstream task at time B_s .

Fig. 2 (Curve A) shows a sample knowledge curve for an automotive engine cylinder block design using estimates from the designers at our study company. Details of this case study are discussed in Section V. For now, consider the two main activities involved in this design process and assume that they are dependent (in reality, they are interdependent as it will be discussed in Section V): cylinder block design (upstream

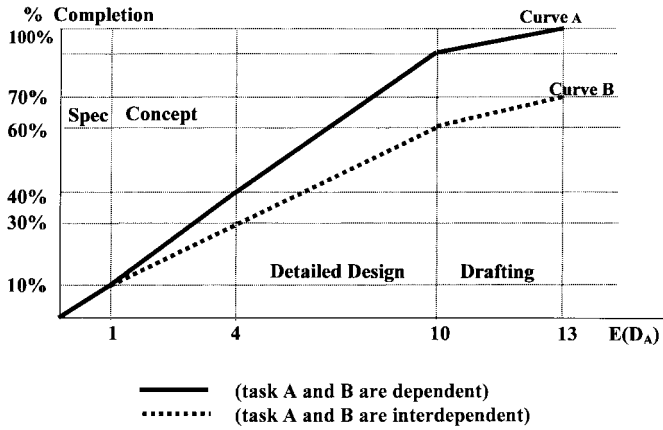


Fig. 2. Sample knowledge accumulation curve for task A.

task) followed by casting operations (downstream task). Fig. 2 shows that if casting operations are started at the fourth month into the design process, then only 40% of the information on task A is accumulated and there is 60% chance of a design change. Therefore, the probability of X_B being equal to $(D_B - t_o)$ is 40% and similarly $P(X_B \geq D_B - t_o) = 60\%$.

Once the probability of a design change is estimated from Fig. 2, we decompose it further into several elements. For example, the design change probability can simply be divided into a drastic change and a small change. A drastic change in upstream information, during overlapping, means that the downstream task has to start from scratch after receiving final upstream information at time A_f . In this case, the magnitude of X_B is equal to D_B . On the other hand, a small change in upstream information can occur during overlapping, resulting in the waste of a fraction of the work completed by task B before time A_f . In this case, there is a small increase in project cost, as compared to the sequential strategy, but overlapping succeeded in reducing the magnitude of (1). The probabilities and magnitudes for each design change type can be estimated from the designers of the downstream task. If we assume that a small change in upstream information results in the waste of 50% of the work accomplished by task B before time A_f , then the expected value of X_B becomes

$$E(X_B) = p * (D_B - t_o) + (1 - p) * [X_B P(X_B)] \quad (6)$$

where $X_B P(X_B) = P(\text{drastic change})(D_B) + P(\text{small change})(D_B - 0.5t_o)$.

C. Concurrent Execution

As mentioned earlier, the main feature of the concurrent execution strategy is the frequent bi-directional communication between the two tasks. In the case of dependent tasks, the bi-directional information exchanges are not necessary because only one task depends on the other and this requires one way information flow. In this respect, dependent tasks do not really benefit from concurrency. However, one can argue that the higher frequency of information exchanges, found in the concurrent execution strategy, might improve the value of X_B because the dependent, downstream task is instantaneously informed of upstream design changes that happen during

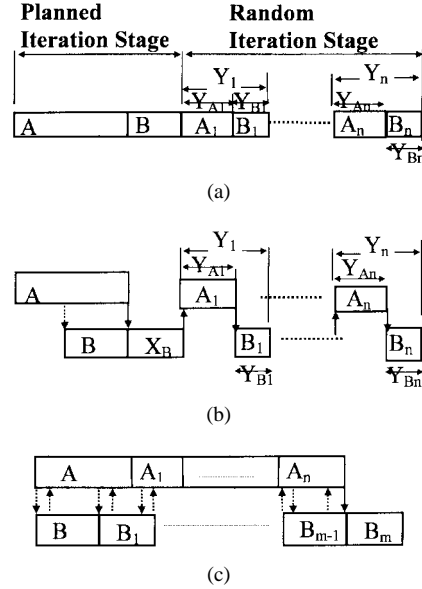


Fig. 3. Execution patterns for two interdependent tasks: (a) sequential; (b) overlapped; and (c) concurrent. Solid arrows represent final information transfer and dotted arrows represent preliminary information transfer.

partial overlapping. Ha and Porteus [6] conceptualize this process of information sharing as progress reviews during the product design phase. Then they argue that reviewing too frequently might hinder, rather than improve, the progress of the downstream task because of the time consumed to evaluate and process the new partial information every time it is received. Therefore, we will assume that if the tasks are dependent, and they are executed concurrently, then this process will not require a CE team and hence will be called parallel and not concurrent execution strategy. The information flow and cycle time calculations, in the parallel strategy, are similar to the partial overlapping strategy but the downstream task commences simultaneously with the upstream task.

IV. INTERDEPENDENT TASKS

When the development process involves interdependent tasks, the development process is divided into two stages: a planned design stage and a random iteration stage. The first stage contains only the initial attempts of both tasks with a nominal duration: $D_A + D_B$. The second stage contains all subsequent design iterations of both tasks. The two stages are shown in Fig. 3(a). There is usually no confusion on what task to start first in the interdependent tasks case. In most product development processes, interdependency between tasks is essentially a one-way dependency between an upstream task and a downstream task with a feedback loop from the downstream task to the upstream task [9].

A. Sequential Execution

Interdependent or coupled tasks can be executed sequentially as a series of alternating sequential tasks until a final agreement is reached [as shown in Fig. 3(a)]. In the planned iteration stage, the upstream task A starts the design process and the results are delivered to the downstream task B. Then,

the downstream task operates on the upstream information and may suggest changes in the upstream design parameters. The random iteration stage starts with the upstream task evaluating the suggested changes and reacting to the suggestions by performing an extra iteration, where the duration of the iteration is proportional to the magnitude of the suggested design change [10]. The latest upstream data are passed downstream and the downstream task re-operates on them and re-evaluates their feasibility. The magnitude of both upstream and downstream task iterations are described by the two random variables Y_A and Y_B , respectively. This process is repeated n times until the design is refined and both tasks agree, where n is the random variable describing the number of iterations of both tasks. The design process cycle time for the sequential execution of coupled tasks becomes

$$T_s = D_A + D_B + \sum_{i=1}^n [Y_{Ai} + Y_{Bi}]. \quad (7)$$

Since both Y_A and Y_B always occur as a couple, we will combine them into one random variable Y . Then (7) becomes

$$T_s = D_A + D_B + \sum_{i=1}^n Y_i. \quad (8)$$

The infeasibilities detected by the downstream task can be classified as minor, moderate, and major. The probability distribution can be assessed from the expertise of the designers (based on similar and previous design situations) involved in the downstream process. Similarly, the combined duration of an upstream iteration and a downstream iteration can be assessed from experts in the design process. For the cylinder block example, we interviewed experts from both engineering design and casting operations and were able to assess these probabilities and values. The probabilities of minor, moderate, and major infeasibilities are 10%, 30%, and 60%, respectively. The duration of an extra design iteration resulting from a minor, moderate, or major infeasibility is one week, two weeks, and one month, respectively. Then, the expected value of Y is $E(Y) = \sum YP(Y) = 0.775$ months.

The number of iterations required to refine the design can be approximated from the experience of the designers with similar development projects. For the cylinder block design example, our interviews with experts from engineering design and casting operations resulted in the following probability distribution: $n = 1, 2, 3$ with 30%, 40%, and 30% probability, respectively. The expected value of n is $E(n) = \sum nP(n) = 2$ iterations.

It is not obvious how the probabilities, associated with the different types of infeasibilities, change with subsequent iterations. Rectifying an infeasibility could trigger another, which had not existed earlier, and that infeasibility could be smaller or larger than the previous one. What exactly will happen is difficult to determine, and for simplicity we assumed that the probabilities will remain constant [22]. The above discussion motivates two assumptions about Y_i ($1 \leq i \leq n$), the duration of each iteration, and n , the number of iterations. First, Y_1, \dots, Y_n are independent, identically distributed, and have a finite mean $E(Y)$. Second, n is

assumed to be independent of Y_i and have a finite mean $E(n)$. Then, using Wald's equation [19] we find the expectation of (8) to be

$$\begin{aligned} E(T_s) &= E(D_A) + E(D_B) + E(Y_i), \quad (1 \leq i \leq n) \\ E(T_s) &= D_A + D_B + E(n)E(Y). \end{aligned} \quad (9)$$

B. Partial Overlapping

Partial overlapping of interdependent tasks can be modeled as an extension to the partial overlapping model for dependent tasks. In addition to the risk of infeasibility detection, discussed earlier, the risk of a redesign in the upstream task during the overlapping period will be included. Similar to overlapping dependent tasks, if a change in the design parameters of the upstream task occurs during overlapping, then the downstream task increases by an amount X_B to incorporate the design change. The magnitude of the downstream task duration increase is a function of the magnitude of the design change, as discussed in Section III. Then, the cycle time for the overlapped interdependent design problem becomes

$$T_o = D_A + X_B + \sum_{i=1}^n Y_i. \quad (10)$$

In order to calculate X_B , we need to estimate the probabilities of upstream design changes during the overlapping period. This can be done by constructing the knowledge accumulation curve for the upstream task. The procedure is the same as described in Section III. However, the curve will never reach the 100% knowledge point even by time A_f tasks A and B are interdependent and no one task is completely finalized without input from the other. The knowledge accumulation curve for the cylinder block example, introduced earlier, when both tasks are interdependent is shown in Fig. 2. This figure shows that by the time task A finishes its first design iteration, it has accumulated only 70% of the design knowledge. The other 30% will be accumulated with time through additional design iterations. These additional design iterations are what was called earlier the random iteration stage. The overlapping execution strategy is shown in Fig. 3(b).

Finally, we will consider overlapping only for the planned design stage (i.e., initial attempt of both tasks). After the initial attempts, both tasks will alternate sequentially. The basic reason for this latest restriction is that the occurrence of any subsequent iteration is uncertain. That is, the first planned design iteration might be successful.

C. Concurrent Execution

This section discusses the impact of bi-directional and frequent communication patterns, which characterizes the concurrent execution strategy, on the cycle time of interdependent tasks. The desired communication patterns in the concurrent strategy are accomplished through a CE team that is formed in the early stages of the upstream task and consists of representatives from both upstream and downstream tasks. The main function of the CE team is to enhance the knowledge accumulation rate for the upstream task by including and discussing downstream concerns early in the development

process. This will allow the downstream task to start concurrently with the upstream task. The upstream task duration might increase, when a CE team is involved, because more points of view are aired and more concerns are considered.

The concurrent execution of interdependent tasks [Fig. 3(c)] is similar to the partial overlapping model (discussed in the previous subsection) but with some modifications. First, we push the start time of the downstream task to start simultaneously with the upstream task, and then we do not allow for subsequent iterations of both tasks to alternate. All subsequent task iterations (within the same task) proceed one after the other with no idle time between them. The number of iterations of both tasks and the magnitude of iterations are not necessarily equal. We will assume that task A will perform n iterations and task B will perform m iterations before the two tasks terminate. After all the iterations of the upstream task terminate at most one downstream iteration will be necessary. This will guarantee that $B_f \geq A_f$ for the same reasons discussed in Section III-B. The magnitude of the i th downstream iteration Y_{Bi} ($0 \leq Y_{Bi} \leq D_B$) can be assessed from the expert of the downstream designers. The resultant cycle time for the concurrent case becomes

$$T_c = D_A + c + \sum_{i=1}^n Y_{Ai} + Y_{Bm} \quad (11)$$

where

- c the increase in the upstream task duration due to CE team involvement;
- n the number of upstream iterations subsequent to the planned one;
- Y_{Ai} the magnitude of the i th upstream iteration in the random iteration stage;
- m the number of downstream iterations subsequent to the planned one;
- Y_{Bm} the magnitude of the m th downstream iteration.

V. CYLINDER BLOCK DESIGN APPLICATION

This section discusses a real-world example that comes from the development of a cylinder block at U.S. automotive firm. The data were collected over a five-month period from January to May 1994. The methodology was not implemented by the firm due to lack of management support for probabilistic thinking in a decision-making context. Although the method suggests potential cycle time reduction, it was not used to reduce the cycle time in this instance.

The development of a cylinder block (from concept to prototyping) involves three different activities: engineering design; casting operations; and machining operations [26]. In this paper, we will examine the interaction between engineering design and casting operations. These two tasks are essentially dependent with engineering design starting first, then the design information is transferred downstream to casting operations for further processing. However, these two tasks are also interdependent due to the fact that no engineering design for the block will be complete without casting input, and at the same time casting engineers need to have a complete engineering design before they can comment on its casting

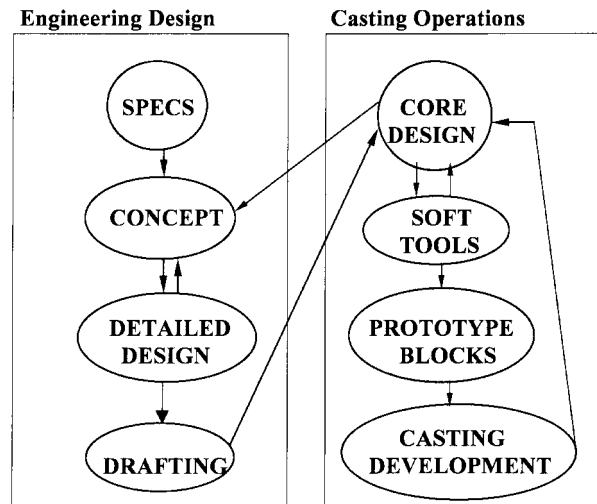


Fig. 4. Cylinder block design process.

feasibility. Therefore, the equations developed in Section IV will be used to evaluate the execution patterns. In this case study, we intend to demonstrate analytically the magnitude of the impact of preliminary information sharing on the cylinder block development lead time.

The following is a brief description of the cylinder block development process. The process flow is shown in Fig. 4. Engine block designers (the upstream task) receive "high-level" description of the engine specifications (i.e., number of cylinders, power and torque requirements, and space available) which they convert into a detailed cylinder block design. Once the block design is detailed and documented, it is passed to the casting department to evaluate its castability (i.e., feasibility from a casting view point) and extract from it the design of the sand cores. Finally, the designs of the sand cores are passed to a supplier to build the soft casting tools (core boxes) necessary. The first set of soft tools are used to produce prototype block castings. After layout and inspection of the prototypes, design changes to the block might occur to rectify any structural or mechanical defects. This task is usually referred to as casting development. In the sequential engineering approach, both engineering design and casting operations will alternate iterations and design changes until an agreement is reached. This extensive iterative process is costly and time consuming.

To reduce the development cycle time, casting operations are considering to start core designs before the engineering design phase is completely finished. The decision model built for the cylinder block development process evaluates the three different approaches of product development: sequential; partial overlapping; and concurrent. In developing the framework for this decision problem, two main decisions are considered:

- 1) execution strategy—sequential versus partial overlapping versus concurrent approach;
- 2) magnitude of overlapping between engineering design and casting operations if the partial overlapping strategy was selected.

The objective of the model is to reduce the development time to a minimum at a reasonable cost increase. Management will decide what is a reasonable cost increase depending on the

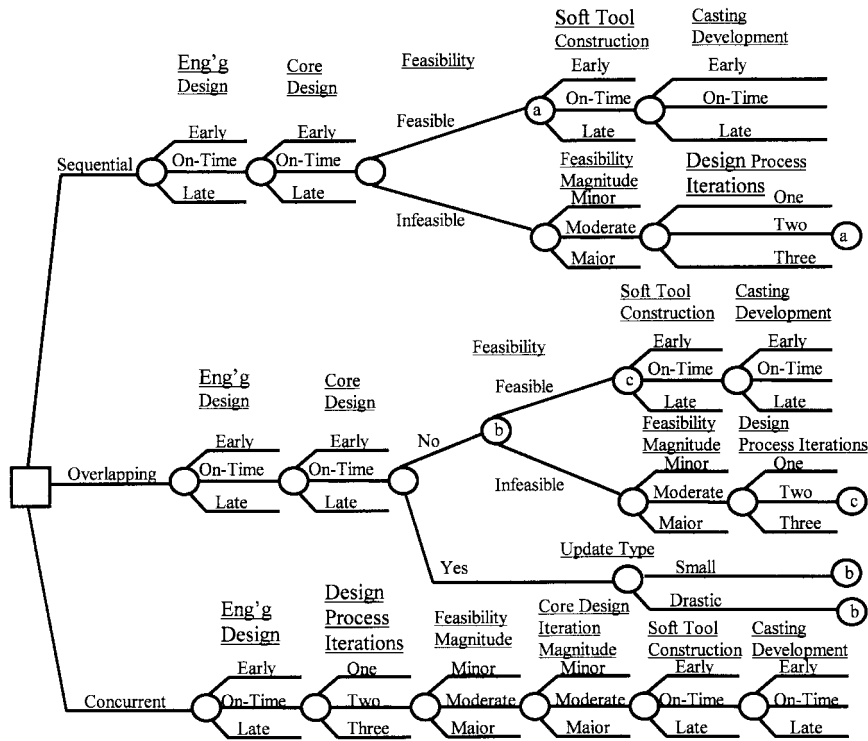


Fig. 5. Cylinder block decision tree.

nature of the product at hand and the relative importance of reducing the lead time when compared to development cost.

A. Randomness

Several internal iterations occur in this design process which reflect the randomness in the duration and cost of design activities. In the engineering design phase, internal iterations occur as a result of the inability of the detailed designers to fulfill all conceptual requirements. Therefore, they negotiate some deviation from the initial concept. Analogously in the casting phase, core designers negotiate the design of cores with casting tool suppliers whenever the tool builders are incapable of producing a certain core shape. All of these internal iterations are modeled by the random variable D_i and captured in the decision model by a chance node for each internal iteration process (see Fig. 5). The chance nodes have three levels each.

Early: no internal iterations (probability of occurrence = 25%).

On-Time: only one extra iteration (probability of occurrence = 50%).

Late: two or more extra internal iterations (probability of occurrence = 25%).

External iterations capture the impact of information interdependency or coupling between two or more activities. In this case study, external iterations occur between engineering design and casting operations when a casting infeasibility in the design is detected. This process is included in our decision model, shown in Fig. 5, by using two chance nodes: infeasibility magnitude and design process iterations. One node represents the number of possible external iterations and has

TABLE I
LEVELS OF DESIGN INFORMATION CHANGE DURING THE OVERLAPPING PERIOD BASED ON OVERLAPPING START TIMES: $7 < t < 10$ AND $10 \leq t < 13$

Design Change	Duration - X_B (months)	Probability: $P(X_B)$ $7 \leq t < 10$	Probability: $P(X_B)$ $10 \leq t < 13$
Small	$D_B - (50\%) t_0$	30%	70%
Drastic	D_B	70%	30%

three levels: one; two; and three, with probabilities equal to 30%, 40%, and 30%, respectively. The other is for the duration of this iterative process and has three levels: minor; moderate; and major. Representative verbal description of the three different levels is as follows.

Minor: a dimensionally infeasible feature that requires a change in location or size.

Moderate: a mechanical problem that is easy to get rid of, such as problems requiring the addition of ribs or texture.

Major: a mechanical problem that is hard to get rid of. It usually requires a change in the engineering design, such as some types of metal shrinks.

The next chance node is related to the upstream design changes during the overlapping period. Core designers have to incorporate these design changes through a “core design update” task. This chance node is the random variable X_B , discussed in Subsection III-B. We will discretize it into two levels of core design updates.

Small: a change that requires simple modifications to the core designs (deleting/adding a feature to an established core box) such as a change in the number of head bolts.

TABLE II
TIME AND MAN-HOUR REQUIREMENTS

	Months	Participants #	% Time	Man-Hour
ENGINEERING DESIGN PHASE				
Specifications	1	1	100 (full)	176
Concept design	3	1	100	528
Detailed design (early)	5	5	100	4400
" (on-time)	6	5	100	5280
" (Late)	7	5	100	6160
Drafting	3	1	100	528
CASTING OPERATIONS PHASE				
Core design (early)	5	5	100	4400
" (on-time)	6	5	100	5280
" (late)	7	5	100	6160
Tool Construction (early)	1.5	2	30	158
" (on-time)	2	2	30	211
" (late)	2.5	2	30	264
Casting Development (early)	1.5	4	3(100%) & 1(60%)	950
" (on-time)	2	4	3(100%) & 1(60%)	1267
" (late)	2.5	4	3(100%) & 1(60%)	1584

Drastic: a change that requires a complete redesign of one or more cores; this is usually related to a change in one of the major block parameters such as changing the number or size of cylinders.

The probability and duration of each design update type depends upon the "Overlapping_Start_Time," t , which is the time at which overlapping is started. For example, if overlapping starts between seven and ten months into the engineering design activity and a design change occurs, then the probability of this change being small is 30%. The time required by casting to incorporate such a change in design information is equal to $[D_B - 50\%(t_o)]$. The rest of the design change levels with their (expert estimated) probabilities and casting response durations are listed in Table I. The last column of the table is used when overlapping times range between ten and 13 months. The reason for having two different sets of probabilities for different overlapping time periods is to reflect higher levels of risk for earlier overlapping start times.

The last chance node is the random variable Y_{B_i} (discussed in Subsection IV-C). In this decision model, we called it "core design iteration magnitude" and was discretized into three levels: minor; moderate; and major. The probability of occurrence of each level is assessed from casting experts to be 30%, 30%, and 40% for minor, moderate, and major, respectively. The magnitude of each is also assessed to be one, two, and three weeks, respectively.

B. Consequences Assessment

Two attributes were used to evaluate the consequences of each decision and random event. The optimal decision will be a tradeoff between both of these two attributes [1]. The first attribute used in the decision model is "calendar time" (in months). It reflects the duration of the overall development process. The other attribute is the "man-hour" (in hours) consumed. The man-hour attribute is used to reflect the cost ramification of the alternative courses of action. Calculations for the man-hour values are based on 22 days

per month and 40 h per week. Table II is a summary of all relevant data for both attributes. For example, the specifications activity, which is part of the engineering design phase, takes one month to finish using one designer working full time.

A similar table can be constructed for the CE team approach, but the only change would be the addition of the CE team man-hours throughout the design process. In order to calculate the man-hours requirement for the CE team, we need to know its formation. The CE team consists of one block engineer, a casting design representative, and a machining representative. The CE team involvement during the early stages of the design process is higher than its involvement toward the end due to the resolution of most serious problems and concerns upstream. Therefore, we averaged the contribution of the CE team members throughout the design process to 15% of the time. This will result in 80 man-hours increase in every month during the design process.

C. Decision Analysis

The decision tree is shown in Fig. 5. Note that the tree and all the forthcoming expected values and sensitivity analysis calculations were produced using the Decision Programming Language (DPL) software [4]. For the overlapping strategy, no specific value for the overlapping start time was initially specified in the decision model. Instead, value sensitivity analysis on the decision variable "Overlapping_Start_Time" (t) was performed to determine the optimal overlapping start time. Fig. 6 shows the optimal decision profile when varying the overlapping start time from $t = 7$ months to $t = 13$ months. The lower limit was selected as seven months to guarantee that the downstream task does not finish before the upstream task.

The graph in Fig. 6 shows that the expected cycle time is an increasing function of t , the overlapping start time. The decrease in total development time, when overlapping is used, is explained by the fact that the mean incremental increase in

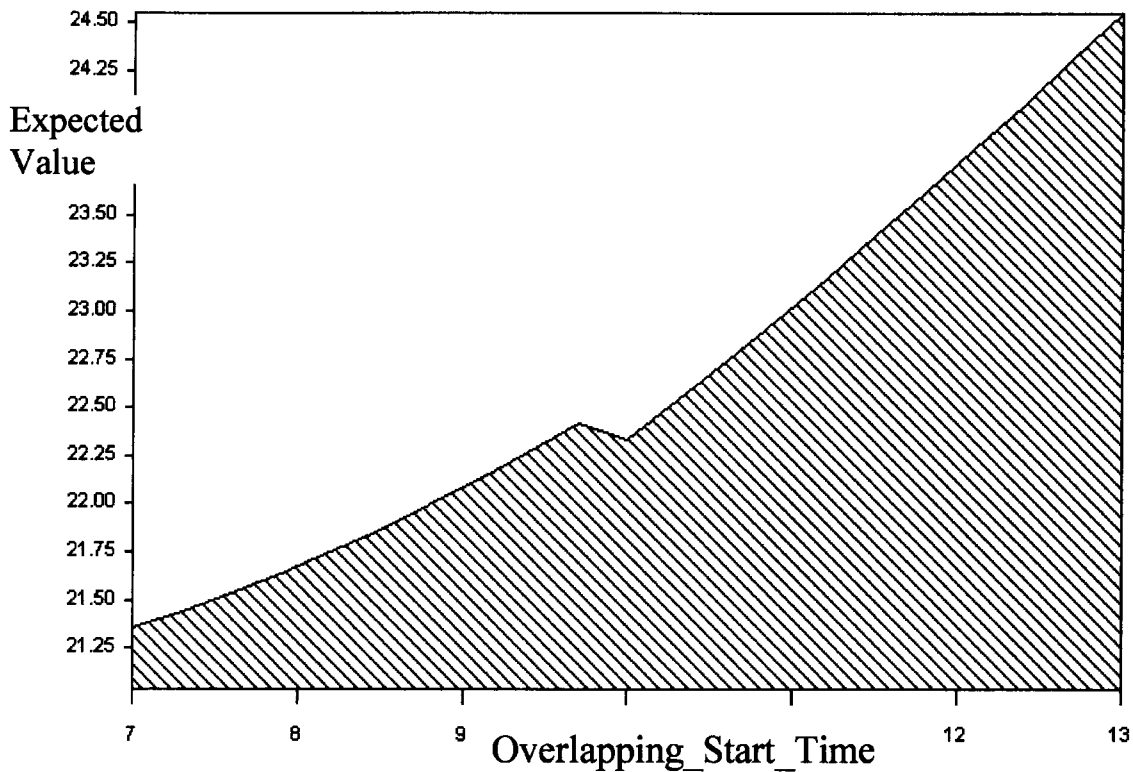


Fig. 6. Sensitivity to overlapping start time.

development time due to core design updates is smaller than the mean decrease in development time due to overlapping. Therefore, we conclude from the figure that when partial overlapping is used, the minimum cycle time is attained by starting the overlapping process at $t = 7$ months.

Fixing the overlapping start time at seven months (for the partial overlapping strategy) and running the decision model gives 24.55 months for the sequential strategy, 21.85 months for the partial overlapping strategy, and 20.15 months for the concurrent strategy. Therefore, the optimal policy is the concurrent strategy.

Running the model with the man-hour attribute, we reached a different optimal policy. The sequential strategy was preferred with an expected value of 14 634 man-hours. The result is not surprising because partial overlapping increased the man-hours used in casting operations through the core design update task. Further, the concurrent approach increased the man-hours due to the participation of the CE team in the design process. Performing sensitivity analysis on the “overlapping start time” variable showed that the expected man-hours requirement is a decreasing function of t , the overlapping start time. That is, as t varies from seven months to 13 months, the expected man-hours requirement for the design process varies from 17 102 h to 14 634 h respectively. The expected man-hour requirement for the concurrent strategy is 18 147 h with a difference of 3513 h (or an increase of 24%) as compared to the sequential strategy. Consulting with design participants on this issue, it was found out that the management of the cylinder block design process would prefer to take this cost increase in order to attain the expected minimum

development time which resulted from the concurrent strategy. Therefore, the optimal solution is to execute the tasks concurrently.

D. Cumulative Risk Profiles

Risk profiles provide the decision maker with insight into why one decision policy might be better than another. They show the probability distribution of the value of the optimal decision policy and are usually represented by a cumulative probability distribution. By a quick look at the risk profile of a decision policy, one can see the range of possible outcomes of that policy.

The cumulative risk profiles for both the sequential and concurrent strategies are shown in Fig. 7. The figure shows that the concurrent and the partial overlapping strategies dominate the sequential strategy. That is, under concurrent or partial overlapping, there is always a higher probability of achieving a certain cycle time as compared to the sequential strategy. When comparing the curves for concurrent versus partial overlapping, we see that the partial overlapping strategy has a bigger spread than the concurrent strategy. This means that there is a bigger variability in the partial overlapping strategy and also there is some chance in achieving higher and lower cycle times than the concurrent strategy. We can also note that there is almost a 40% chance that the development time will be 20 months or less if either the concurrent or the partial overlapping strategy is used. Beyond this point, the concurrent strategy yields a higher probability for achieving a certain development time. The intersection of both curves also

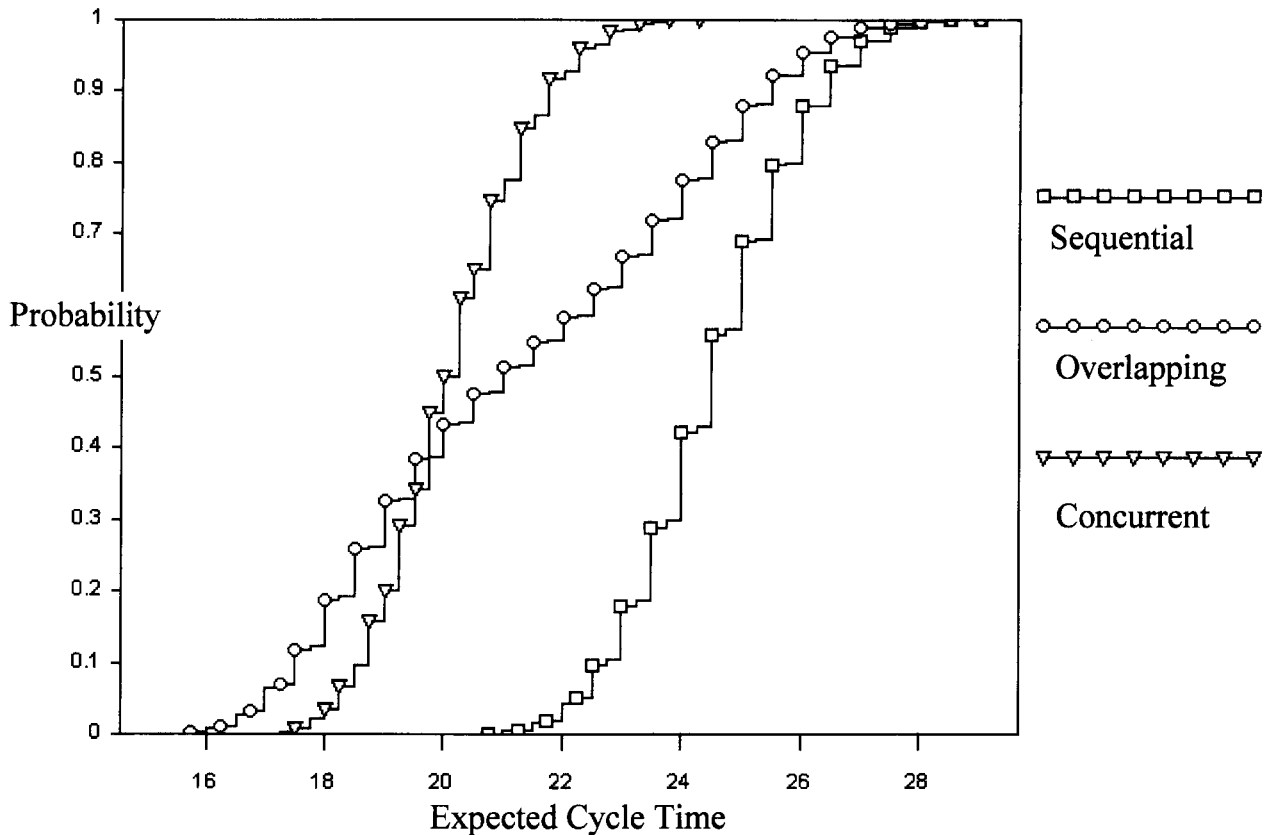


Fig. 7. Cumulative risk profiles.

mean that non of the strategies dominate the other, and there is some region where one is better than the other.

E. Sensitivity Analysis

Sensitivity analysis is an important part of R&DA because it involves checking the effects of the model assumptions and parameters (values and probabilities) on the model solution. Therefore, our initial estimates, for the model parameters, need not be very accurate. After performing sensitivity analysis on all the values and probabilities used in the decision model, we will be able to determine which parameters in the model impact the optimal solution and which ones are insignificant. Then, these significant parameters can be revisited and carefully re-assessed.

Sensitivity analysis was performed on the optimal strategy to see the impact of variation in some of the probabilities and values on the optimal decision. First, we will determine the impact of varying c , the increase in the engineering design (upstream task) duration due to the CE team involvement (initially set to 1.5 months). The value of c was varied between one and four months, as shown in Fig. 8. From the figure we see that the optimal policy (concurrent execution) does not hold once the value of c is larger than almost 2.5 months. In this case the optimal policy becomes the partial overlapping strategy and the cycle time becomes 21.85 months when overlapping starts at $t = 7$ months. Second, the value of "No_Change_Probability" (at $t = 7$ months) was varied from

0.4 to 0.8 (initially set to 0.45). Its sensitivity graph is shown in Fig. 9. The figure shows that the optimal strategy changes from concurrent to partial overlapping when the probability of no change in preliminary engineering design information increases to 70% or more. We have also performed sensitivity analysis on the probabilities and duration of small and drastic design changes (during the partial overlapping period). The optimal policy was insensitive to reasonable changes in those values.

VI. CONCLUDING REMARKS

In this paper, we have introduced the notion of design structure versus design schedule and showed how can CE benefit from the discrepancy between both models. Then, we developed a probabilistic model of the design process that allowed us to investigate the benefits and risks of partial overlapping and CE practices as means to improve the design process. This, in turn, allowed us to determine analytically the optimal overlapping magnitude if partial overlapping is to be used.

Applying this probabilistic modeling approach to the cylinder block development process suggested 18% reduction in development time when compared to the sequential duration of 24.5 months. The reduction in development time came at a substantial increase in development cost as reflected by the 24% increase in the man-hour requirement. This suggests that CE can be an expensive option and it should be used when

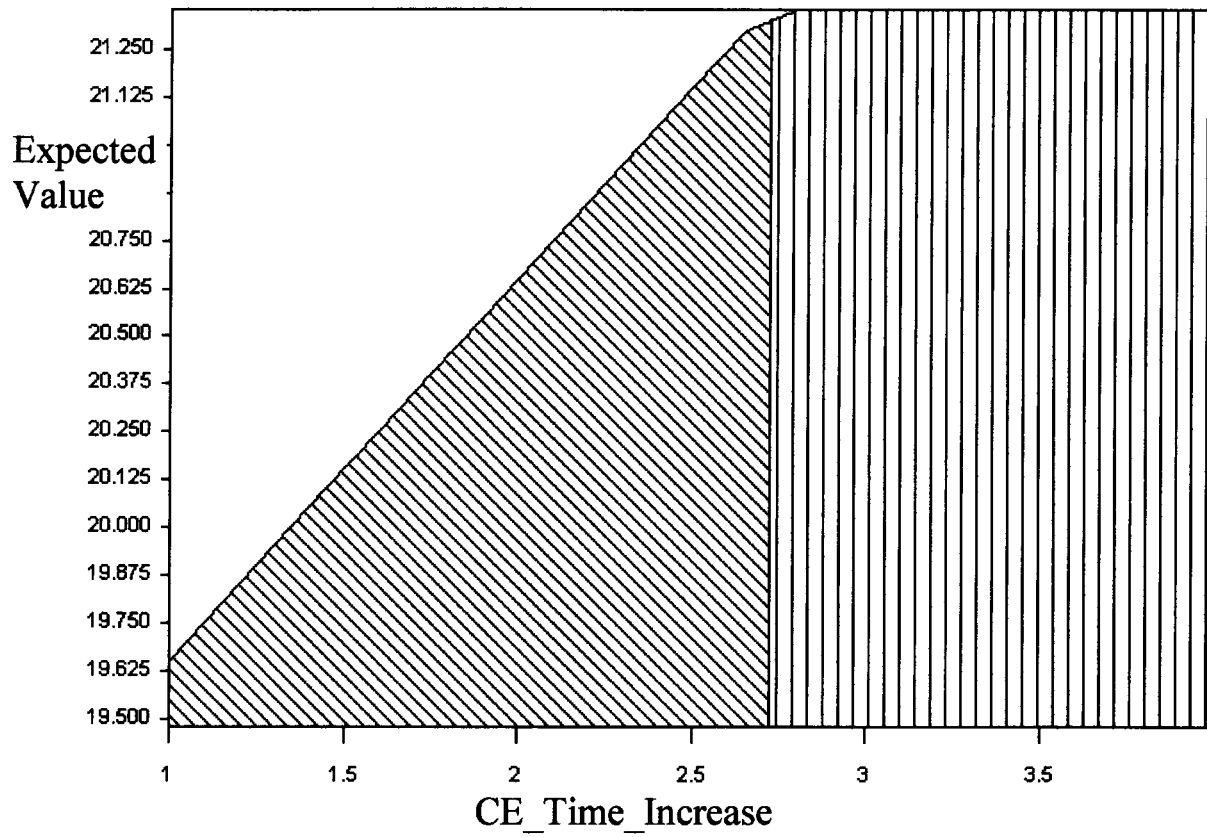


Fig. 8. Sensitivity to c value.

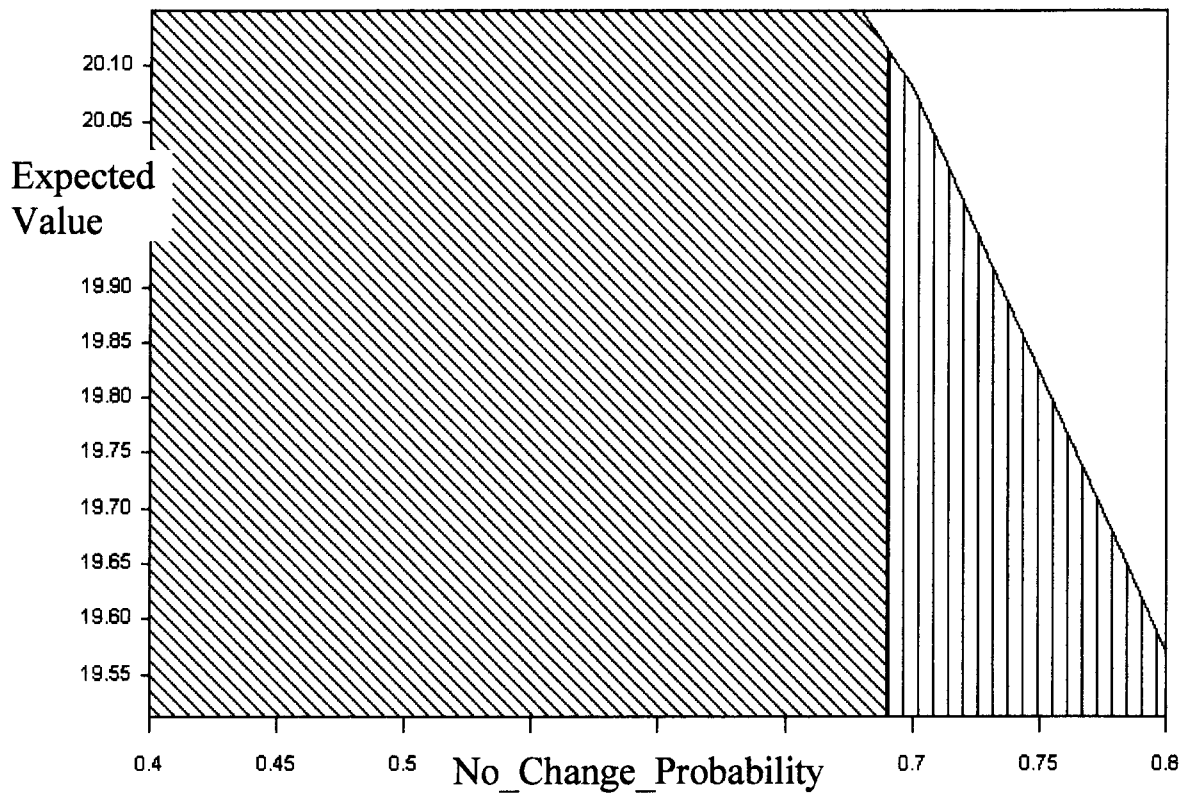


Fig. 9. Sensitivity to No_Change_Probability.

the reduction in development time has much higher priority than cost. Otherwise, management should consider the partial overlapping alternative that can also reduce (but to a lesser extent than the concurrent strategy) the sequential development time at a smaller cost increase. For example, they can select the level of acceptable cost increase and find the overlapping start time that corresponds to it.

The sensitivity analysis performed on the decision model showed that the optimal solution is sensitive to the amount of increase in development time due to CE team involvement. If this increase is larger than 2.5 months, then it will not help in reducing the development cycle time and partial overlapping will be a better option. For the overlapping strategy, the probability of a design change during overlapping has the biggest impact on the optimal strategy. A change in this probability from the initially assumed 0.45 to 0.7 or higher, will make the overlapping policy a better alternative than concurrent execution.

Several extensions and improvements to the work presented in this paper are possible. First, we considered three main execution strategies, but one can devise a combination execution strategy and determine its effectiveness in reducing the cycle time. One combination strategy might be utilizing a CE team in partial overlapping or even in the sequential execution approach. Second, overlapping was mainly examined between two coupled tasks. We think that the same methodology can be expanded to include overlapping between three or more coupled tasks. Finally, more accurate methods for knowledge accumulation can be used to provide better estimates of design change probabilities and impacts, thus improving the overall model soundness.

ACKNOWLEDGMENT

The authors are grateful to the Department Editor and two anonymous reviewers for their helpful and insightful comments on previous versions of this research paper.

REFERENCES

- [1] F. AitSahlia, E. Johnson, and P. Will, "Is concurrent engineering always a sensible proposition," *IEEE Trans. Eng. Manag.*, vol. 42, pp. 166–170, May 1995.
- [2] K. Clark and T. Fujimoto, *Product Development Performance: Strategy, Organization, and Management in the World Auto Industry*. Boston, MA: Harvard Business School Press, 1991.
- [3] R. Clemen, *Making Hard Decisions*. Boston, MA: PWS Kent, 1990.
- [4] Decision Programming Language (DPL), Standard Version User Guide, ADA Decision Systems, 1992.
- [5] S. Eppinger, D. Whitney, R. Smith, and D. Gebala, "Organizing the tasks in complex design projects," in *Proc. ASME Design Theory and Methodology Conf.*, Chicago, Sept. 1990, pp. 39–46.
- [6] A. Ha and E. Porteus, "Optimal timing of reviews in concurrent design for manufacturability," *Manage. Sci.*, vol. 41, no. 9, pp. 1431–1447, Sept. 1995.
- [7] R. Howard, "Decision analysis: Practice and promise," *Manage. Sci.*, vol. 34, no. 6, pp. 679–695, 1988.
- [8] N. King and A. Majchrzak, "Concurrent engineering tools: Are the human issues being ignored?," *IEEE Trans. Eng. Manag.*, vol. 43, pp. 189–201, May 1996.
- [9] V. Krishnan, S. Eppinger, and D. Whitney, "Iterative overlapping: Accelerating product development by preliminary information exchange,"

in *Proc. ASME Design Theory and Methodology Conf.*, 1993, DE-vol. 53, pp. 223–231.

- [10] V. Krishnan, S. Eppinger, and D. Whitney, "Accelerating product development by the exchange of preliminary product design information," *J. Mechanical Design*, vol. 117, pp. 491–498, Dec. 1995.
- [11] V. Krishnan, "Managing the simultaneous execution of coupled phases in concurrent product development," *IEEE Trans. Eng. Manag.*, vol. 43, pp. 210–217, May 1996.
- [12] V. Krishnan, S. Eppinger, and D. Whitney, "A model-based framework to overlap product development activities," *Manage. Sci.*, vol. 43, no. 4, pp. 437–451, Apr. 1997.
- [13] M. Lawson and H. Karandikar, "A survey of concurrent engineering," *Concurrent Engineering: Research and Applications*, vol. 2, pp. 1–6, 1994.
- [14] D. Lee, "Management of concurrent engineering: Organization concepts and a framework of analysis," *Eng. Manage. J.*, vol. 4, no. 2, pp. 15–25, June 1992.
- [15] C. Loch and C. Terwiesch, "Communication and uncertainty in concurrent engineering," *Manage. Sci.*, to be published.
- [16] M. Merkhofer, "Quantifying judgmental uncertainty: Methodology, experiences, and insights," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-17, pp. 741–752, Sept./Oct. 1987.
- [17] M. Morelli, S. Eppinger, and R. Gulati, "Predicting technical communication in product development organizations," *IEEE Trans. Eng. Manag.*, vol. 42, pp. 414–425, Aug. 1995.
- [18] H. Raiffa, *Decision Analysis*. Reading, MA: Addison Wesley, 1968.
- [19] S. Ross, *Introduction to Probability Models*. New York: Academic, 1985.
- [20] D. Samson, *Managerial Decision Analysis*. Homewood, IL: Irwin, 1988.
- [21] G. Shephard and C. Kirkwood, "Managing the judgmental probability elicitation process: A case study of analyst/manager interaction," *IEEE Trans. Eng. Manag.*, vol. 41, pp. 414–425, Nov. 1994.
- [22] R. Smith and S. Eppinger, "Modeling design iterations," Massachusetts Inst. Technol. Sloan School Manage., Working Paper, no. 3160-90-MS, June 1990.
- [23] R. Smith, "The historical roots of concurrent engineering," *IEEE Trans. Eng. Manag.*, vol. 44, pp. 67–78, Feb. 1997.
- [24] R. Smith and S. Eppinger, "Identifying controlling features of engineering design iteration," *Manage. Sci.*, vol. 43, no. 3, pp. 276–293, Mar. 1997.
- [25] J. Sobieszcanski-Sobieski, "Multidisciplinary optimization for engineering systems: Achievements and potential," Langley Research Center, Hampton, VA, NASA TM 101566, Mar. 1989.
- [26] A. Yassine, "An information based approach to concurrent engineering implementation and management," Ph.D. Dissertation, Wayne State University, Detroit, MI, 1994.



Ali A. Yassine received the B.E. degree in mechanical engineering in 1988 from the American University of Beirut, Lebanon. He received the M.S. and Ph.D. degrees in 1989 and 1994 in industrial and manufacturing engineering from Wayne State University, Detroit, MI.

Currently, he is a Research Scientist at the Massachusetts Institute of Technology (MIT) Center for Technology, Policy, and Industrial Development, Cambridge. Previously, he was a Consultant at Chrysler International Corporation and at Ford Motor Company. His research work involves concurrent engineering enabling methodologies and design process modeling and management using risk and decision analysis.



Kenneth R. Chelst received the B.A. degree from Yeshiva College, New York, the M.S. degree in operations research from New York University, New York, and the Ph.D. degree in operations research from Massachusetts Institute of Technology, Cambridge.

He is currently a Full Professor in the Department of Industrial and Manufacturing Engineering, Wayne State University, Detroit MI. He also consults with Urban Science Applications, Inc., a worldwide leader in the planning of new car dealership networks.

His research interests include structured decision making in engineering management and the impact of globalization on the engineering and manufacturing management functions. He has also carried out extensive research in the area of emergency service planning. He is currently directing an ongoing project to develop operations research modules for use in the high school math classroom.

Dr. Chelst is a member of the Institute for Operations Research and the Management Sciences (INFORMS).



Donald R. Falkenburg received the Bachelor of Mechanical Engineering and Master of Science degrees from Clarkson University, Potsdam, NY, and the Ph.D. degree from Case Western Reserve University, Cleveland, OH.

He is Professor and Chair of Industrial and Manufacturing Engineering at Wayne State University, Detroit, MI. He was a member of a team that launched the Industrial Technology Institute, Ann Arbor, MI, which is a contract R&D organization chartered to enhance industrial productivity through

its program focus on advanced manufacturing technologies. He was one of the founders of the Information Technology Requirements Council and served as Chairman of the Board and Acting President during the start-up phase of this organization chartered to develop user requirements for a broad spectrum of information technologies.

Dr. Falkenburg's professional activities include membership on the Board of the State of Michigan Research Fund. He has served as a member of Board of Directors of the Michigan Technology Council, Chairman of the Midwest Simulation Council, a member of the Board of Directors of the Society for Computer Simulation, and Vice-President of Industries at the Institute of Industrial Engineers.