A Model-Based Framework to Overlap Product Development Activities

Viswanathan Krishnan • Steven D. Eppinger • Daniel E. Whitney
Management Department, The University of Texas at Austin, Austin, Texas 78712
Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139
Center for Technology, Policy and Industrial Development, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

Intense competition in many industries forces manufacturing firms to develop new, higher quality products at an increasingly rapid pace. Overlapping product development activities is an important component of concurrent product development that can help firms develop products faster. However, since product development activities may be coupled in complex ways, overlapping interrelated activities can present many difficulties. Without careful management of the overlapped product development process, the development effort and cost may increase, and product quality may worsen. This paper goes beyond the common recommendation to simply overlap activities as much as possible. We present a model-based framework to manage the overlapping of coupled product development activities. The model and framework identify conditions under which various types of overlapping are appropriate for a pair of coupled activities. We illustrate the model and framework with industrial applications involving the development of electronic pagers and automobile doors.
(*Product Development; Concurrent Engineering; Overlapping*)

1. Introduction

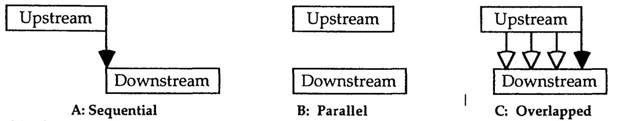
Our studies of the structures of many industrial product development processes (Eppinger et al. 1994, Krishnan 1993) suggest that the flow of information and the process execution in practice is largely sequential, with information being generated and finalized by the upstream activities before being absorbed by the downstream development activities. Figure 1A illustrates such a sequential process where a downstream activity (such as product prototyping) receives and utilizes design information only after it is finalized by the upstream activity (such as concept development). One approach to accelerating such a process involves removing the coupling between the activities, as shown in Figure 1B, thereby enabling the two activities to be executed in parallel. However, such decoupling generally requires a fundamental redefinition of the development activities and/or the product architecture, which may be undesirable, or difficult at best.

An alternative approach to acceleration is to *overlap* the activities through more frequent exchange of preliminary information (see Figure 1C). In contrast to the one-time transfer of finalized information in a sequential process, frequent exchange of design information enables the concurrent execution of coupled activities in an overlapped process. In the sequential process of Figure 1A, the downstream activity does not begin until finalized information is available at the completion of the upstream activity. In the overlapped process, the downstream activity begins earlier by using preliminary information (shown using hollow-arrows).

In order to overlap the nominally sequential activities, the pattern of information exchange between them needs to be altered. As shown in Figure 2, this may involve disaggregating the exchanged information X into

0025-1909/97/4304/0437$05.00

Figure 1 **Sequential, Parallel and Overlapped Processes**



A: Sequential
The downstream activity begins only after receiving finalized information upon completion of the upstream activity.

B: Parallel
The coupling between upstream and downstream activities is removed, resulting in a parallel process.

C: Overlapped
The coupling is preserved and the activities are overlapped through frequent information exchange.

parts X_1 and X_2, and exchanging some parts (such as X_2) multiple times in a preliminary form, and other parts (such as X_1) only once but in a finalized form at or before the completion of the upstream activity. In this paper, we seek to answer the following questions.

1) On what basis should the exchanged information be disaggregated into parts such as X_1 and X_2? Which parts should be utilized by the downstream activity only in finalized form, and which parts may be used in preliminary form?

2) How early can the parts such as X_1 be finalized and with what consequences? Under what conditions can preliminary information (as with X_2) be useful for downstream action?

3) How is product development performance, in terms of lead time, cost, and quality, affected by overlapping activities and modification of the pattern of information exchange?

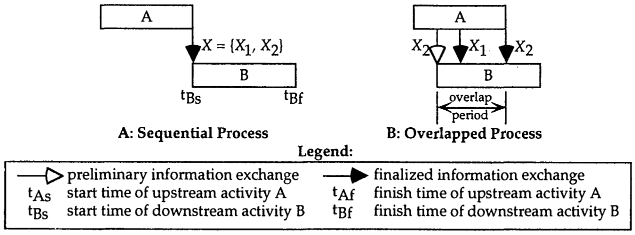
1.1. The Risks Underlying Overlapping

While a major benefit of overlapping is the potential for reducing product development lead time, the reduction in lead time (over the sequential process) does not always equal the overlap period shown in Figure 2. This is because the duration of the downstream activity may be altered in converting the sequential process into an overlapped process. When preliminary upstream information is utilized by the downstream activity too early, then future changes have to be incorporated in time-consuming subsequent iterations, resulting in an *increase in the downstream duration and effort*. On the other hand, when parts of the upstream-generated information are finalized early, the upstream activity loses the flexibility to make future changes along these dimensions. This loss of flexibility may be interpreted as a *quality loss* (QL) for the upstream activity. Any attempts at overlapping the activities—by exchanging preliminary information or by finalizing upstream information early or by a combination of both—should ensure that the adverse effects on product quality and development effort are minimal while improving the lead time performance.

In this paper, we present models to develop three insights: (i) that the effect of overlapping on development performance should be used as the basis for disaggregating the exchanged information (X into parts such as X_1 and X_2); (ii) that finalizing upstream information early may cause a quality loss, so only those parts whose early freeze would produce very little quality loss should be finalized (frozen) early; and (iii) that the upstream information (such as X_2) exchanged in a preliminary form should be chosen such that changes in its value may be absorbed without a substantial increase in the downstream effort. Such overlapping policies can ensure that development time is reduced without a large adverse impact on product quality or development effort.

To determine which parts of the exchanged information to finalize early and which parts to use in preliminary form, we consider two characteristics of the product development process. These are (upstream information) *evolution* and (downstream iteration) *sensitivity*, which are developed in the following sections. Using evolution and sensitivity, we formulate a simple mathematical model of the overlapped process to help decide when and how the development activities should be overlapped. The model is illustrated with an example of door panel development at a United States automotive company. A conceptual framework is derived from the mathematical model to provide managerial insights. The framework is illustrated using an electronic pager development process at Motorola Inc.

Figure 2 **Transforming a Sequential Process into an Overlapped Process**


t_{As}
t_{Af}
A
$X = \{X_1, X_2\}$
B
t_{Bs}
t_{Bf}
A: Sequential Process
t_{As}
t_{Af}
A
X_2
X_1
X_2
B
overlap
period
B: Overlapped Process
Legend:
preliminary information exchange
finalized information exchange
t_{As} start time of upstream activity A
t_{Af} finish time of upstream activity A
t_{Bs} start time of downstream activity B
t_{Bf} finish time of downstream activity B

In the next section, we discuss how our research relates to other work in the literature.

2. Related Work

The effect of overlapping on product innovation and noncommercial product development has been investigated by several researchers, including Mansfield et al. (1971) and Eastman (1980). Based on their field studies, these researchers observed that overlapping stages led to an increase in development costs, which helps to motivate the need for careful overlapping.

In the context of improving commercial product development performance, Imai, Takeuchi, and Nonaka (Imai et al. 1985, Takeuchi and Nonaka 1986) observed that faster development processes use a more overlapped approach. They likened the sequential process to a relay race where one group of functional specialists passes the baton to the next group, and the overlapped approach to a rugby game in which a hand-picked, multidisciplinary team travels the distance as a unit. Although the analogies are powerful, these authors (i) imply that all activities in the development process can be carried out concurrently, and (ii) do not indicate *how* a sequential or phased process may be overlapped. As we noted earlier, careful consideration is required to minimize the risk associated with overlapping. Further, the technical complexity of the products developed imposes certain precedence relations among development activities due to which not all activities can be done concurrently. For instance, making major resource commitments for the development of tooling and fixtures of an automobile while the vehicle concept is still under development can increase the costs inordinately—as concept changes can require substantial rework. We believe that the technical structure and strength of interactions between tasks need to be well understood for activities to be overlapped successfully.

In their comparative study of product development practices in the world automotive industry, Clark and Fujimoto (1987b) developed a measure called "overlap ratio", and showed that faster development processes are more overlapped. They also recognized the organizational barriers to overlapping activities, such as hostile environment, poor communication, and the lack of consistency and balance in managing the critical linkages (Clark and Fujimoto 1989). To facilitate overlapping, they recommended frequent, face-to-face, bilateral communication of preliminary information instead of late release of complete information (Clark and Fujimoto 1991).

We believe these recommendations should be useful to initiate organizational changes, but to operationalize overlapping the generic reference made to information exchange needs to be augmented with more specific understanding of the characteristics of the exchanged information. Close examination of industrial product development suggests that not all information exchanges are equal. Some types of product information can be more readily used in their preliminary form than others because of their lower impact on the downstream activities. Also, some product information can be frozen early, some need to be frozen early and some others cannot be or should not be frozen early. Our work builds on previous research by Clark and Fujimoto in the following ways: a) we argue that there are limits to concurrency and develop a methodology to overlap activities based on the specific characteristics of the information exchanged between them; b) we address the risk associated with overlapping by examining the effect of overlapping on development performance; and c) we offer specific methods to disaggregate exchanged product information.

Other researchers, notably Blackburn (1991) and Smith and Reinertsen (1991), have recognized the merits of overlapping, but do not offer detailed information on how overlapping may be implemented in a coupled, nominally sequential process. Further, the effects of design interactions and the risks involved in overlapping strongly coupled activities have not been considered. Smith and Reinertsen (1991) recommend that downstream designers search for ways to start early with partial information but do not warn of considerable iterations that may result from hasty starts. Our work with industry suggests that overlapping involves not only the downstream activities starting early with preliminary information but also (i) the determination of which design parameters are of high importance to the downstream activity and (ii) upstream finalizing certain parameters earlier, ahead of normal course of action.

Ha and Porteus (1995) studied another important benefit of overlapping, which involves the early detection of flaws in the product design during design reviews. In their work, each design review is modeled to consume a certain amount of setup time, corresponding to the time needed to coordinate and conduct a review. If the product design is reviewed too frequently, then too much time would be spent on the reviews that could otherwise be spent on product and process design. Infrequent reviews, however, would increase the risk of a design flaw going undetected, leading to extensive redesign (iterations). Given these tradeoffs, they seek to determine the optimal timing of design reviews that minimizes the expected project completion time. Our paper complements the work of Ha and Porteus by focusing on processes where the downstream iterations are primarily due to changes in preliminary product information rather than due to the flaws in the product design.

Nihtila (1992) studied product development in five industries and noted the risk associated with overlapping—" . . . excessive overlapping of too many interdependent phases seemed to result in quite heavy iteration in determining the engineering parameters and constraints . . ."—but offered no specific solutions. We believe that the risk of overlapping activities can be better managed through a proper understanding of the properties of the exchanged information. In an earlier paper (Krishnan et al. 1995), we presented a model of one form of overlapping, called iterative overlapping, which we further develop and expand into a conceptual framework in this paper. The model presented in this paper is an expansion of the earlier model in that it explicitly considers early finalization of upstream information, and thereby forms the basis for other types of overlapping presented in this paper using a framework. Before discussing the model, we develop two important properties of the exchanged information in the next section.

3. Evolution and Sensitivity

From our field study of industrial product development, we find that two properties of the design process, which we call evolution and sensitivity, determine how activities must be overlapped. In this section, we develop the notions of evolution and sensitivity. In the next section, we use evolution and sensitivity to construct a model of overlapped design processes. Later, we capture the insights from the model as a framework for managing design processes.

In developing the ideas in this paper, we adopt the "information processing" view of product development (Clark and Fujimoto 1987b, Utterback 1982). From this perspective, product development is considered to be a process of transformation of input information about customer needs and market opportunities into output information which corresponds to manufacturable designs, and functional tooling for volume production. Individual development activities are themselves viewed as information processors, receiving input information from their preceding activities, and transforming this input information into a form suitable for subsequent activities. The timing of this process is illustrated by Figure 2, in which for the nominally sequential case, the upstream activity begins its information processing at t_{As} and continues processing until t_{Af}, at which point the information is transferred to the downstream activity B. To enhance our understanding of this situation, we will model the exchanged information in greater detail.

In practice, the information exchanged between activities takes various forms such as: customer specifications, part dimensions, prototypes, etc. In this paper, we focus on exchanged information that can be described as a collection of parameters (where we define parameter as a scalar piece of information that can be represented within a compact set). Examples of parametric information include part dimensions and customer specifications. In practice, exchanged information in the engineering stages of product development can often be represented as a collection of parameters. We develop the following ideas for a scalar parameter (X) and later discuss extensions to a vector of parameters.

3.1. Upstream Information Evolution

The sequential process assumes that the upstream-generated information is available for downstream use only at the completion of the upstream activity. Overlapping may be facilitated, however, by realizing in fact that the upstream activity may continually narrow and refine the generated information. We use the term evolution to refer to the refinement of the upstream generated information from its preliminary form to a final value. To more precisely define evolution, we develop a model of preliminary upstream information.

We represent preliminary upstream design information as an interval value. Such a model of preliminary product information has also been used by other researchers in design methodology. See, for example, Finger et al. (1992) and Ward and Seering (1993). In the preliminary stages, we model that the exact value of parameter X at t_i ($<t_{Af}$), X_i is unknown, but an interval $\{a_i, b_i\}$ is known (to the upstream activity and shared with the downstream activity) such that $a_i \le X_i \le b_i$. At the beginning of the upstream activity, let this interval be $\{a_{in}, b_{in}\}$, also called the initial interval. During the course of the upstream activity, as product development is carried out, and as external inputs are received, the parameter X is gradually narrowed to its final value X_f, a point within the initial interval. The interval within which the parameter is known changes continuously until when the parameter attains its final value at t_{Af} (see Figure 3).

We introduce the notion of degree of evolution (denoted by ϵ) to measure how close the unfinalized design parameter is to its final value. With the interval model of preliminary information, the smaller the interval width, the closer the parameter is to its final value. For simplicity, the degree of evolution is defined to be a linear function of the width of the design interval. The evolution is said to be *monotonic* when for $t_j > t_i$, the interval $\{a_j, b_j\}$ lies within $\{a_i, b_i\}$. In this paper, we will restrict ourselves to monotonic evolution. Models of processes for which the upstream design evolution is not monotonic are presented in (Krishnan 1993). The following expression defines the degree of evolution ϵ_i, at time t_i, such that $\epsilon_{As} = 0$ and $\epsilon_{Af} = 1$:

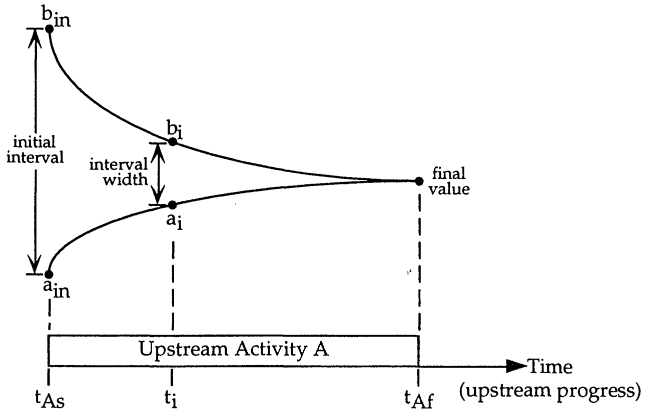
$$\epsilon_i = 1 - \frac{b_i - a_i}{b_{in} - a_{in}}.$$

A plot of ϵ_i is called the *evolution function* of the parameter X. Since ϵ_i depends on the width of the interval at time t_i, the evolution function indicates how (at what rate) the upstream activity narrows the width of the parameter interval to zero. With the evolution function and the initial interval as input, it is possible to determine the interval width at various points in time during the design process, but not the exact interval endpoints of the parameter X.

3.2. Model of Downstream Iterations and Downstream Iteration Sensitivity

In the overlapped process, the upstream activity shares preliminary upstream information available in the form of interval endpoints with the downstream activity. In many practical situations, however, the downstream development activities (such as prototyping and tooling design) require a point value of the parameter X to perform development iterations. In this paper, we assume that when the downstream activity begins to perform a development iteration at time t_i ($<t_{Af}$), it uses the expected value, x_i, of the exchanged information X_i. Further, downstream iterations in the overlapped process are used to accommodate the changes in the upstream information in the downstream development. Specifically, downstream iteration i, beginning at time t_i, incorporates the change in the exchanged information value (used by the downstream activity) from the start time of the previous iteration, t_{i-1}, to the start time of the current iteration, t_i. This change in the exchanged information value from t_{i-1} to t_i is denoted by $\Delta x(t_{i-1}, t_i)$.

Figure 3 Upstream Activity Narrows the Exchanged Parameter to Final Value



By *downstream sensitivity*, we mean the relationship between the duration of downstream iteration i, d_i, and the magnitude of the change in the upstream information value, $\Delta x(t_{i-1}, t_i)$. The function relating d_i and $\Delta x(t_{i-1}, t_i)$ is called the *sensitivity function*, Φ: $d_i = \Phi(\Delta x(t_{i-1}, t_i))$. Except for a few special cases, the sensitivity function for product development processes is usually nondecreasing; i.e., larger changes in the value of the exchanged information require longer iterations to process those changes.

When the downstream activity uses the expected value, x_i, to perform iterations, the magnitude of $\Delta x(t_{i-1}, t_i)$ equals the change in the expected value from

t_{i-1} to t_i. This change in the expected value, however, depends on the design interval at t_{i-1} and t_i. Note that it is not possible to predict what the design interval endpoints might be at various points in time in the design process. The given evolution function is only indicative of what the width of the interval would be at these times. It is, however, possible to derive a conservative estimate for Δx when the evolution function is monotonic. In Krishnan et al. (1995) we show that the maximum change in the expected value of X between t_{i-1} and t_i is given by:

$$\Delta x(t_{i-1}, t_i) = w_{in}(\epsilon_i - \epsilon_{i-1}),$$

where $w_{in} = (b_{in} - a_{in})/2$ is half the initial interval width of X. In other words, the change in the value of the exchanged information used by the downstream activity at two points in time can be estimated as the difference in degree of evolution at these points in time. Such an estimate obviates the need for any adhoc assumptions about the process of arrival of changes at the downstream activity.

3.3. Implications of Upstream Evolution and Downstream Sensitivity

Understanding of upstream evolution and downstream sensitivity has several implications for managing the design process. If the evolution of the upstream information is such that large changes were to happen near the completion of the upstream activity, then it would be difficult to reduce the lead time by overlapping—as the subsequent design iterations would be of significantly large duration. If, however, the evolution of the design information is such that major changes occur during the initial period of the upstream activity, and only minor changes happen during the final period of the upstream activity, then overlapping is more likely to help reduce lead time.

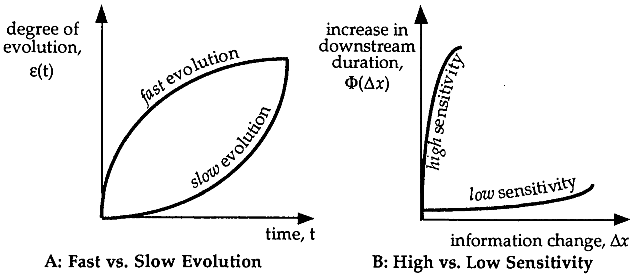
From the expression for $\Delta x(t_{i-1}, t_i)$ derived in the previous subsection, we note that the change in the exchanged information value between two points in time is proportional to the difference in degrees of evolution at these points in time. If the design information evolves such that as the upstream activity progresses the increase in the degree of evolution during a given interval of time gets progressively smaller, then the amount of change decreases as the upstream activity progresses. We term this *fast evolution* as major changes happen early (Figure 4A). Also, in this case, the exchanged information gets close to its final form rapidly, and is capable of being frozen and passed downstream early in the upstream process without much quality penalty for the upstream activity. When the increase in the degree of evolution gets progressively larger as the upstream activity progresses, the change in the exchanged information also increases as the upstream activity progresses, and the evolution is said to be *slow* (Figure 4A). In this case, finalizing upstream information early in the upstream process either would be impossible or would entail a huge quality penalty for the upstream activity. The evolution is slow for example, when the information pertains to a component which (i) interfaces with several change-prone components, (ii) requires information supplied by an extraneous factor that is not available until late in the upstream process, or (iii) involves solving a complex technical problem. Overlapping activities is generally easier when the upstream evolution is fast than when it is slow.

Similarly, we consider the two extreme cases of downstream sensitivity, *low and high sensitivity*. Low sensitivity describes the case when substantial changes (in the exchanged information value used by the downstream activity) can be accommodated quickly by the downstream activity (Figure 4B). Downstream sensitivity is said to be high when even small changes require design iterations of a large magnitude.

3.4. Measuring and Modifying Evolution and Sensitivity

In practice, evolution and sensitivity functions should be obtained as input from product development professionals based on their experience. This is particularly

Figure 4 **Extreme Values of Evolution and Sensitivity**



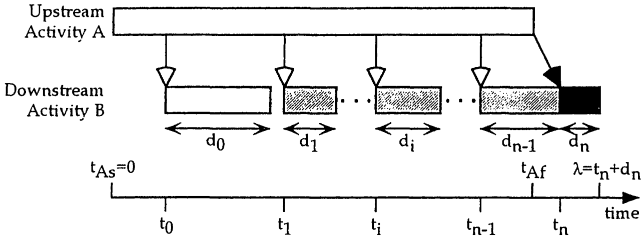
true for redesigned products, which form a significant portion of products designed in many companies (McGrath et al. 1992, Whitney 1990). The evolution function can be constructed based on how much the intermediate upstream tasks help refine the exchanged design parameter. The sensitivity function is constructed by examining the impact of upstream design changes on the downstream activity. Practitioners have observed that the process of designing a new product is often not radically different from that of an earlier generation product (Albano and Keska 1989, McGrath et al. 1992). In such cases, the documentation and understanding of the established process can be used to describe the evolution and sensitivity functions. If the change in the development process is substantial or the product is entirely new, then forecasts of evolution and sensitivity need to be made based on the particular design concept, product architecture, and underlying technologies. For situations where evolution and sensitivity are difficult to measure, we develop a conceptual framework (later in this paper) that uses qualitative evolution and sensitivity information to determine how to overlap the activities.

Evolution and sensitivity can to some extent be altered by better communication practices and information technology. As we discuss later with reference to pager development, improved communication between the upstream and downstream activities in the studied process helped reduce the downstream sensitivity. Also, computer-based tools can help ensure that the upstream design evolves faster and the changes are absorbed by the downstream activity more easily.

4.0. Model of the Overlapped Process

In this section, we use evolution and sensitivity to formulate a model of the overlapped process. Figure 5 illustrates the features of the process we are modeling.

Figure 5 Model of Interaction for Overlapping
Shaded blocks indicate downstream iterations in progress. White space between shaded blocks denotes wait time between iterations. Dots between blocks (· · ·) denote a series of iterations not shown in an expanded form. Arrows indicate information exchange between upstream and downstream activity.



Notation

t_i = start time of ith iteration (decision variable) $i = 0, 1, 2, \ldots, n$,

d_i = duration of ith iteration (variable) $i = 1, 2, \ldots, n$,

d_0 = duration of planned iteration (input),

t_{As} = start time of upstream activity (taken as origin of the time scale),

t_{Af} = nominal finish time of upstream activity (input),

t_F = advance freeze time (decision variable, not shown in Figure 5),

n = number of iterations subsequent to planned iteration (decision variable), and

λ = product development lead time (output measure).

Upstream activity A begins at time t_{As} to generate design information X that will be used by downstream activity B. (For the rest of this paper, $t_{As} = 0$ unless stated otherwise). The downstream activity is modeled to begin with a planned downstream iteration of given duration d_0, which is the nominal time required to complete the downstream activity in the absence of upstream design changes.

In the sequential process, the upstream-generated information X is not finalized until time t_{Af}. The downstream activity B begins the planned iteration upon receipt of finalized information from A at time t_{Af}, resulting in a lead time of $t_{Af} + d_0$. No subsequent downstream iterations exist in this approach, as there are no changes in the value of the exchanged information ($n = 0$ and $t_F = t_{Af}$). Any internal downstream iterations are assumed to be included in the planned iteration.

In the overlapped process, there are two important differences: (i) the upstream information may be finalized early thereby allowing earlier access to final information for the downstream activity, and (ii) the downstream activity may also begin with preliminary upstream information (incorporating changes in subsequent iterations). Our model helps to decide how many downstream iterations to perform, when to start these iterations, and how early the upstream information should be finalized. We model the practice at leading firms where preliminary information can be transferred (electronically or through other means) to the downstream activity without incurring significant costs.

In the case where the upstream generated information is not let to evolve until t_{Af}, it is finalized at an earlier point in time, t_F ($\leq t_{Af}$). This freeze time is chosen by the design team. The associated quality loss will be modeled in the next subsection.

In the case where the upstream activity A exchanges unfinalized information, changes in the exchanged information must be incorporated in *subsequent* downstream iterations which are indexed from 1 to n. Downstream activity once again starts with a planned iteration of duration d_0, beginning at time t_0, which is assumed to start no earlier than the upstream activity, $t_0 \geq t_{As}$.

Each subsequent downstream iteration i, has duration d_i, starts at t_i, is performed with the expected value of the design parameter at time t_i, and incorporates the change in the value of the exchanged information, $\Delta x(t_{i-1}, t_i)$. To facilitate learning from iterations, all changes occurring *during* a downstream iteration will be batched together and incorporated in the next downstream iteration.

Because successive iterations are performed by the same downstream activity (resource), no two downstream iterations may overlap. This leads to the constraint: $t_i \geq t_{i-1} + d_{i-1}$. If this constraint is not satisfied as an equality, it means that there is some idle time between iterations.

Because the value of X is finalized at t_F, exactly one downstream iteration is required after information is frozen to incorporate the final change that happened from the previous iteration ($t_n \geq t_F$). All iterations except the last start before t_F; hence, $t_{n-1} < t_F$.

Collecting these constraints and modeling assumptions together, we have:

$$t_F \leq t_{Af} \text{ (information may be frozen before } t_{Af}\text{)}, \quad (1)$$

$$t_i \geq t_{i-1} + d_{i-1}, \quad i = 1, 2, \ldots, n \quad (2)$$

(idle time between iterations may not equal 0),

$$t_0 \geq t_{As} = 0 \text{ (planned iteration starts after } t_{As}\text{)}, \quad (3)$$

$$t_{n-1} < t_F \text{ (all iterations, but last, start before } t_F\text{)}, \quad (4)$$

$$t_n \geq t_F \text{ (last iteration cannot start before } t_F\text{)}, \quad (5)$$

$$\lambda = t_n + d_n \text{ (}\lambda\text{ is the development lead time)}, \quad (6)$$

$$\Delta x(t_{i-1}, t_i) = w_{in}(\epsilon_i - \epsilon_{i-1})$$

(amount of change is related to evolution), (7)

$$d_i = \Phi(\Delta x(t_{i-1}, t_i))$$

(duration of iterations is a function of change). (8)

In the above model, the decision variables are the number of iterations, n, and the start timing of the iterations, $t_0, t_1, \ldots, t_n$. The number of constraints therefore depends on the decision variable n. In order to make the dependency of the model on n explicit, we decompose the above model into two subproblems; the first subproblem $P(n)$, determines the iteration starting times for a given value of n. The second subproblem determines the value of n for the optimal overlapping strategy. This decomposition is illustrated in the next section for the door panel example. Alternatively, the problem could be solved by dynamic programming, but due to the small values of n encountered in practice, this approach was not deemed necessary.

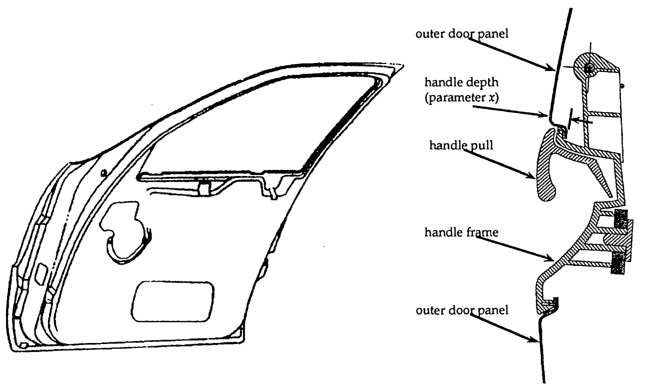
4.1. Model Objectives

As noted in the beginning of this paper, overlapping the development activities affects all three dimensions of product development performance: quality, lead time, and development effort. If the upstream generated information were to be frozen at t_F, ahead of the normal time of freeze t_{Af}, then the upstream activity foregoes the flexibility to make changes in the design until t_{Af}. The amount of change expected in this period is:

$$\Delta x(t_F, t_{Af}) = w_{in}(\epsilon_{Af} - \epsilon_F) = w_{in}(1 - \epsilon_F).$$

The quality loss due to this loss of flexibility is modeled as: QL $= \rho(w_{in}(1 - \epsilon_F))$, where $\rho(\cdot)$ is the quality loss function determined by the product and its market. The development effort is a function of the duration for

Figure 6 An Automobile Door Panel and Door Handle



which the upstream and downstream resources are committed.

Based on the product development situation at hand, each of these performance dimensions may become more or less important than the others. For example, in a product development organization using dedicated platform teams (as is the case with the door panel development process in the next section), the human resources are committed throughout the development process, so there may not be an added cost for using the engineering resources to start downstream iterations early. Thus quality and lead time are more important than effort in this case. (There may be other costs associated with more iterations such as cost of materials, machining etc. which must be considered.) Note that the objective and/or the constraints (corresponding to the evolution and sensitivity functions) above may be nonconvex leading to multiple local optima. This is illustrated with the following door panel development process.

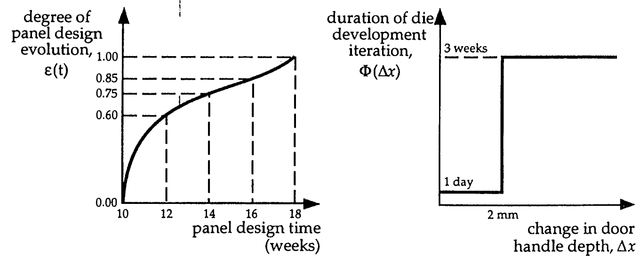
5.0. Overlapping Model Illustration: Door Outer Panel Development Process

We illustrate the model with application to the design process of an automobile door (see Figure 6). We focus specifically on one of the long lead time phases in the design process: prototyping of the door (outer) panel stamping dies. Prototypes of the door panel dies are constructed using (relatively) inexpensive soft steel to verify form and function, before machining them from expensive hard steel. The prototyping process in our study company, however, is often delayed for want of the door handle design information. We turn our attention to overlapping door handle engineering design (upstream activity A) and die prototype development (downstream activity B) which exchange the door handle design information.

Figure 6 also shows a door handle positioned in an outer door panel. Interviews with door designers suggest that early freeze of the door handle design can entail a high quality loss as the door handle may not meet the customer specifications adequately. Also, a handle design frozen too early may resemble some recently introduced competitor product, and with door handles being very visible parts of a car, differentiation is important. The door handle depth into the outer panel (x) is an important parameter used in the panel die development process, and delays in the door handle design decisions tend to delay the die development process (and to some extent the entire door development process).

The evolution (ϵ) of the door handle depth estimated by the door handle designers at our study company is given in Figure 7A. The design process (with $t_{As} = 10$, and $t_{Af} = 18$) includes several intermediate tasks, analyses, and consultations occurring at approximately two-week time intervals, which help to refine and finalize the door handle depth. Until the 10th week, the handle concept is not known, so we define the evolution to be zero ($\epsilon = 0.0$). The styling engineers provide input on the pocket profile in the 12th week ($\epsilon = 0.60$). Finite element analysis takes place in the 14th week ($\epsilon = 0.75$). Vendor feedback becomes available in the 16th week (ϵ

Figure 7 Evolution and Sensitivity Functions for Door Handle Depth Information



A: Evolution Function

B: Sensitivity Function

= 0.85). Finally the stamping engineers provide feedback which confirms the door handle depth in the 18th week ($\epsilon = 1.00$). Note how the evolution function has both concave and convex segments.

The sensitivity function (Φ) relating the change in door handle depth to the duration of die prototype development iterations is shown in Figure 7B. Small changes in the door handle depth require mere updates (small iterations) of the appropriate engineering drawings and requisite approvals. Changes larger than 2 mm in the depth would exceed the accommodation (safety factor) planned in die development, and the die design must be reworked in large iterations which can take as much as three weeks. (See (Krishnan 1993) for a detailed description of this change implementation process.)

In overlapping panel design and die development, freezing the door handle before all external feedback is available may result in a large quality loss. Because quality is of paramount importance to our study company, we rule out the possibility of the early freeze of the door handle design (thereby setting $t_F = t_{Af}$), and explore the possibility of overlapping by exchanging preliminary information about the handle depth. Our study company uses dedicated platform teams for the vehicle development, so there may not be any added cost for committing the downstream resources (engineers) early; however, the larger die development iterations requiring rework at the die shop are expensive. To adhere to the die development budget, no more than two large iterations can be done subsequent to the planned iteration.

The nominal duration for die model development is estimated by the engineers to equal four weeks ($d_0 = 4$). The initial design interval for the depth, $\{a_{in}, b_{in}\}$, is equal to $\{2, 12\}$ mm, based on previous experience in designing door handles, so $w_{in} = 5$ mm. Later we will examine the sensitivity of the optimal solution to the changes in some of these parameters.

The dominant performance term in this case is the development lead time, λ. The marginal cost of development lead time for an entire automobile is estimated to be at least \$1 million per day (Clark et al. 1987a). Perhaps, the delay cost in door panel development time would not be quite as high, but still substantial because sheet metal panel development is generally on the critical path.

The Formulation. As noted earlier, the number of constraints in the model is a function of the decision variable, n, so we formulate the model as a two-stage problem. The first stage, $P(n)$, is a nonlinear program as given below.

$$P(n) \quad \text{Minimize } \lambda = t_n + d_n \text{ subject to:}$$

$$t_i \geq t_{i-1} + d_{i-1}, \quad i = 1, 2, \ldots, n, \tag{9}$$

$$t_0 \geq t_{As} = 10, \tag{10}$$

$$10 < t_i < t_F, \quad i = 1, 2, \ldots, n-1, \tag{11}$$

$$t_n \geq t_F = t_{Af} = 18, \tag{12}$$

$$\Delta x(t_{i-1}, t_i) = w_{in}(\epsilon_i - \epsilon_{i-1}), \tag{13}$$

$$d_i = \Phi(\Delta x(t_{i-1}, t_i)), \quad i = 1, 2, \ldots, n. \tag{14}$$

Now we find the overall optimal strategy by solving $P(n)$ for all realistic values of n and picking the smallest value of n that leads to the lowest lead time. Our study of industrial product development processes suggests that the number of iterations in practice is not very large (typically $n < 10$), so the above approach remains computationally feasible. Also, we can often identify lower bounds on lead time that greatly simplify the solution procedure as shown below for the door handle example.

Let $d_{\min}$ denote the time consumed by the iteration with the smallest duration, ($d_i \geq d_{\min}$). For the door handle, $d_{\min} = 1$ day (from Figure 7B). From the above constraints, it is seen that a lower bound on lead time is given by $\lambda = t_n + d_n \geq t_{Af} + d_{\min}$. Using the values for the door handle process, we have $\lambda \geq 18\frac{1}{7}$ weeks.

For $n = 0$, the process is sequential with lead time $= t_{Af} + d_0 = 22$ weeks.

For $n = 1$, the solution is $t_0 = 14$; $d_1 = \frac{1}{7}$; $t_1 = 18$; $\lambda = 18\frac{1}{7}$ weeks. (There are multiple optima yielding the same lead time, we identify only one of them here.)

Since the optimal solution with $n = 1$ equals the lower bound, we need not consider larger values of n. Thus, overlapping activities by exchanging preliminary information helps reduce the development time from 22 weeks (for the sequential process) to 18 and one-seventh weeks for the overlapped process.

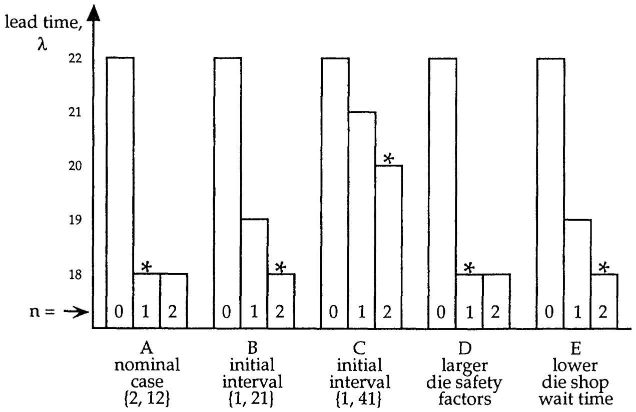
5.1. Model Discussion

Since the model inputs (evolution function, sensitivity function, initial interval, and duration of the planned iteration) are estimates, we now examine the effect of variations in these inputs on the lead time advantage gained by overlapping. Specifically, we analyze how the development lead time varies with changes in one of the inputs, initial interval width. Suppose that the handle width did not lie between 2 mm and 12 mm, but in an interval twice as wide: between 1 mm and 21 mm. Figure 8B shows that for this case, the optimal lead time (shown with a "*") is once again $18\frac{1}{7}$ weeks, but requires two subsequent iterations. (With one iteration, the lead time is $19\frac{1}{7}$ weeks). If the handle used is so radical that the initial depth is known only within a much wider interval say, {1, 41} mm, the resulting policy is shown in Figure 8C. In this case, two subsequent iterations can reduce the lead time only to 20 weeks. As the design becomes more unpredictable, the lead time advantage gained from overlapping decreases.

What can the product development organization do to accelerate such radically new processes? In Figure 8D we see that (for the initial depth interval of {1, 41}) an improvement in the die development safety factor to 5 mm (such that all changes in Figure 7B of up to 5 mm in depth can be accommodated in small iterations) can help reduce the optimal lead time to $18\frac{1}{7}$ weeks. Similarly as shown in Figure 8E, for the same initial depth interval of {1, 41}, a decrease in the timing of large iterations to one week (by reducing the die shop wait time or processing time) can help reduce optimal development time to $18\frac{1}{7}$ weeks. Improving the die development safety factor is a more effective strategy, as it achieves the optimal lead time with fewer iterations, but is probably a harder capability to acquire in practice. (This involves substantial research in the area of metal forming; one work in this direction is a computer program developed by Toyota to evaluate press forming severity that helps detect stamping infeasibilities quickly (Okamoto et al. 1988).)

The above calculations show that when the initial interval width increases, the development time increases, but in a relatively small way. For a two-fold increase in the width of the initial interval, the optimal development time does not change, and for a four-fold increase in the width of the initial interval, the optimal development time increases and the lead time advantage decreases by two weeks.

Figure 8 **Effect of Inputs on the Overlapping Policy**



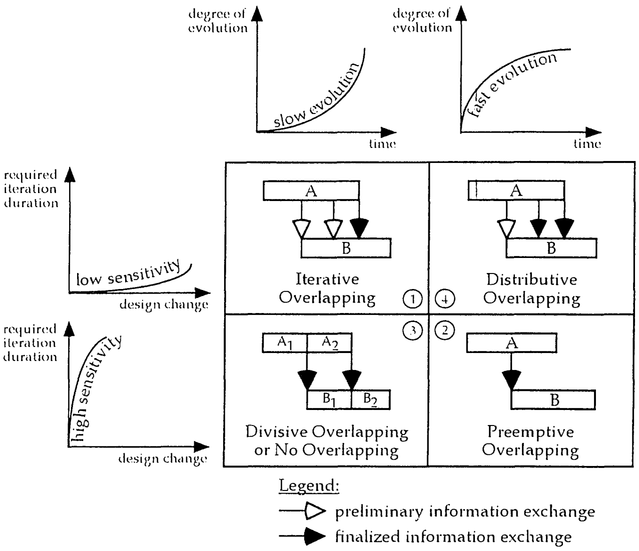
6. A Conceptual Framework for Overlapping Activities

The model uses some detailed inputs which may be difficult to obtain in practice. For example, in the absence of a good understanding of the design process, it may be difficult to collect detailed and accurate evolution and sensitivity data. In this section, we derive a conceptual framework to provide managerial insights using qualitative evolution and sensitivity inputs. The framework considers the extreme values of evolution and sensitivity and offers insights on how the coupled product development activities may be overlapped.

6.1. The Evolution-Sensitivity Framework

In overlapping product development activities, managers face the same questions which motivated the model development: When should downstream act on upstream information? How should the activities be overlapped when the downstream activity cannot work on preliminary information? When is it better to freeze the upstream information rather than transfer it in a preliminary form to the downstream activity? The combination of the extreme values of evolution and sensitivity offers answers to these questions. In Figure 9, we show the four extreme situations which can arise—when the upstream evolution is fast or slow combined with when the downstream sensitivity is high or low. The mechanisms used for overlapping (preliminary information exchange/early finalization) differ for each combination of evolution and sensitivity.

Figure 9 Types of Overlapping Based on Evolution and Sensitivity



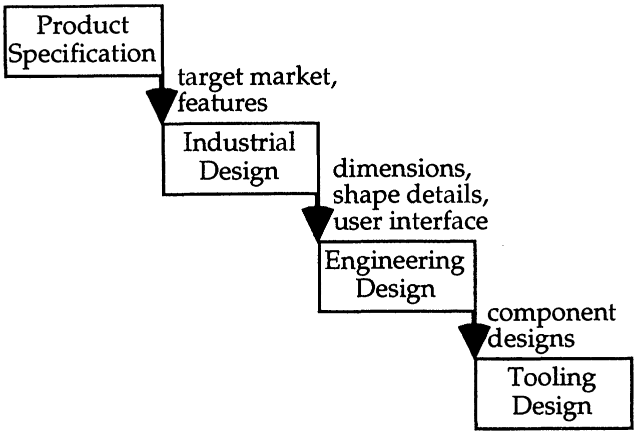
Case 1: When the sensitivity is *low*, it is possible to commit downstream resources based on preliminary upstream information. In this case, even if the changes in the exchanged information are large, their effects on the downstream activity are not. Further, when the upstream evolution is *slow*—major changes happen until late into the upstream process before which information cannot be finalized—then the overlapping is said to be *iterative*. In this case, the activities are overlapped by beginning downstream activity with preliminary information, and incorporating design changes in subsequent downstream iterations ($n > 0$). The design information is not finalized until the nominal completion of the upstream activity, because doing so may result in a large quality penalty for the upstream activity due to the slow evolution ($t_F = t_{Af}$).

Case 2: The opposite case is when the downstream sensitivity is *high*, but the upstream information evolution is *fast* (information can be finalized early in the upstream activity without much quality loss). In such a case, the exchanged information is to be preempted by taking its final value/form at an earlier point in time. In other words, the upstream problem solving is accelerated and information frozen ahead of the normal time of freeze ($t_F < t_{Af}$). This is called *preemptive* overlapping and would help reduce development time by starting the downstream activity earlier- but with *finalized* upstream information. Note that there are no subsequent downstream iterations ($n = 0$). It may result in some quality loss to the upstream activity because the upstream activity loses the opportunity to make changes until its original completion time.

Case 3: Consider the case when the downstream sensitivity is *high* and the upstream evolution is *slow*. Here it is neither desirable to start downstream activity with preliminary information nor feasible to preempt further changes in the exchanged information at an earlier point in time ($t_F = t_{Af}$; $n = 0$). In such a case, the exchanged information is disaggregated into components to see if any of the components evolve faster or if transferring any of the components in their preliminary form to the downstream activity is practical. Often the evolution and sensitivity of the components may be different from the aggregated information (Krishnan 1996). Because the disaggregation is based on physical or functional division of the upstream and downstream activities, this approach is called divisive overlapping. If no information evolves quickly, nor is it of any use to the downstream activity in preliminary form, then no overlapping is recommended with the current evolution and sensitivities.

Case 4: The last scenario occurs when both the upstream information evolves *rapidly* and the downstream sensitivity is *low*. In such a case, it is possible to both start downstream activity with advance information and to preempt later changes in the exchanged upstream information ($t_F < t_{Af}$; $n > 0$). Because the impact

Figure 10 Pager Development Process Phases



of overlapping is distributed between the upstream and downstream activities (unlike in other cases), this situation is called *distributive* overlapping.

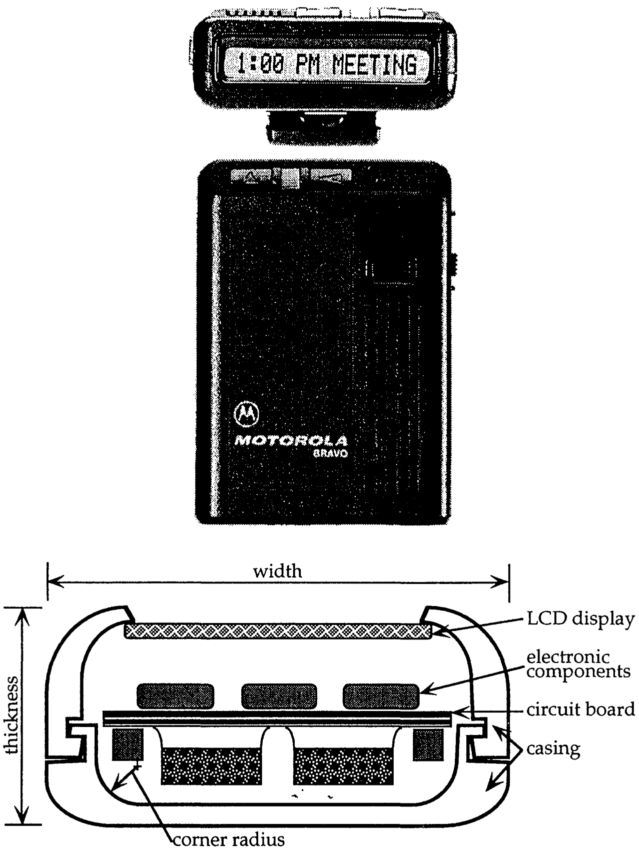
It is noteworthy that the different types of overlapping result in different tradeoffs among the performance parameters. In iterative overlapping for instance, downstream effort is traded-off against lead time, while in preemptive overlapping, upstream quality is traded-off against lead time. Also, while the above framework presents overlapping mechanisms for different combinations of evolution and sensitivity, it does not make generalizations about the amount or degree of overlap for the following reason. It is generally the case that a fast evolution, low sensitivity situation is more favorable for overlapping (and a high sensitivity, slow evolution process is less favorable). Under very low sensitivity, however, it may be optimal for a slow evolution process to involve greater overlap than a faster evolution process. This is due to the nonlinear form of our model, and can be explained by the fact that a small amount of wait (during the initial part of the upstream activity) can help the downstream activity avoid a period of intense change in the fast evolution case, while the same is not true for the slow evolution case.

6.2. Illustration of the Conceptual Framework: Pager Development at Motorola

In this section, we illustrate the above framework with application to the process of development of an electronic pager at Motorola Inc., our second study company. As shown in Figure 10, the process of development of a pager involves four stages: product specification, industrial design (styling), engineering design, and tooling design. During the product specification stage, the target market, pager geometric form and carrying method are defined. During the industrial design stage, the external dimensions, shape details and the user interface are defined. The engineering design stage involves design of the mechanical components and the electrical circuitry. Finally, tooling design involves the development of molds, assembly tooling, and test fixtures.

Our focus here is on the overlapping of the engineering design and industrial design phases, since together these consume a large fraction of the total development time. Figure 11 shows a representative pager and a pager cross section, which allows us to discuss these two stages in more detail.

Figure 11 Representative Pager and Schematic of a Pager Cross Section

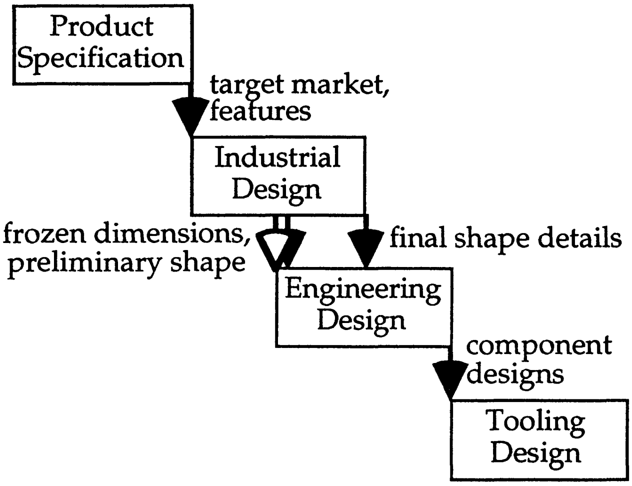


Engineering design of the pager requires several important inputs from the industrial designers, including the pager external dimensions and shape details. Experience with designing pagers indicates that the pager dimensions evolve quickly. These dimensions are constrained by the competition, target market and technology, and are included in the product specification, but must be confirmed by volume studies and human factor studies done by the industrial designers. For instance, if the pager is to fit into a shirt pocket, its dimensions are immediately constrained by standard pocket sizes and by sizes of recent competitive products. On the other hand, the evolution of the shape details is much slower as industrial designers change shape information such as the corner radii to ensure consistency of shape with details of the other features. This relatively slow evolution is confirmed by data from a recently developed pager where the radius had changed as much as 30% near the end of the industrial design phase, while the corresponding change in the pager dimensions was small enough to not require a subsequent engineering iteration.

The engineering activity is very sensitive to changes in product dimensions. Not only is the layout of the components affected but also the choices of components, as smaller components (manufactured with more recent or not yet available technology) may be needed to accommodate a shrinkage in size. Changes in the shape are slightly different. Changes in corner radii may indeed affect the layout of components as the casing wall could interfere with some components. However, for the process studied, effective communication within the team allowed engineers to anticipate potential changes in the radii and to place the taller electronic components near the center of the circuit board. Since changes in the corner radii do not affect the center of the board, the sensitivity of engineering design to changes in shape details was reduced.

Thus engineering requires two types of information from industrial design: pager dimensions (fast evolution, high sensitivity), and shape details (slow evolution, low sensitivity). To overlap the engineering and industrial design functions, engineering design may start with preliminary shape information, but frozen values of the pager dimensions (see Figure 12). Overlapping engineering and industrial design by early freeze of the highly sensitive, fast evolving pager dimensions falls under preemptive overlapping in which the pager dimensions were frozen to their final value to obviate engineering rework. Changes in the shape details, on the other hand, are incorporated in future iterations due to their lower impact; this exemplifies iterative overlapping. Exchanged information is therefore disaggregated based on its evolution and sensitivity.

Figure 12 **Overlapping of Engineering Design and Industrial Design**



7. Conclusions

The framework presented above helps determine how to disaggregate design information and overlap consecutive stages of a development process based on the (evolution and sensitivity) properties of the information exchanged between these stages. The framework and the mathematical model presented in §4 have also been found to be applicable to the development of an automobile instrument panel development process, the results of which are reported in (Krishnan 1996).

Classification of the exchanged information on the two dimensions, evolution and sensitivity, is useful to model the effect of overlapping on product development performance and enables the management of the risk associated with overlapping. Note that the conventional project management paradigm, which assumes the one-shot transfer of finalized information, corresponds to only one of the four possible cases—that of slow evolution and high sensitivity requiring finalized information release. The framework presented here expands the domain of project management by considering three other combinations of evolution and sensitivity. In developing the model and the framework, we have made a conscious effort to preserve simplicity and keep the attention focused on the design issues observed in practice. Specifically, the cost of transferring information to the downstream activity is not included in our model, because of the emerging trend of information technology making this task inexpensive. While making more modeling assumptions (such as the functional forms of the evolution and sensitivity) can help enrich the model of the overlapped process, care must be exercised to ensure that the model continues to be a good abstraction of practice.

Several aspects of the model and framework presented in this paper merit further examination. First, extension of the model to processes with multiple design activities needs attention. This can perhaps be accomplished by adding more variables corresponding to the information exchanges between each pair of activities. Second, the model may be augmented to include the detection of upstream infeasibilities by the downstream activities. This would involve the possibility of downstream feedback leading to upstream iterations. Also, reviews performed to detect flaws/infeasibilities may involve substantial time, so the cost of these reviews must be included (Ha and Porteus 1995). A third direction for future research is to study the overlapping problem when the upstream and downstream activities involve internal iterations as modeled by Smith and Eppinger (1995).

In summary, few analytical methods exist to model and analyze the product development process. Existing project management tools are inadequate because product development processes are inherently iterative, and information in product development processes can be profitably exchanged multiple times in preliminary form. This paper contributes to the research on the management of the design process by (i) highlighting the limits to concurrency and developing a simple model of the overlapped development process, (ii) presenting a conceptual framework to facilitate managerial decision making about overlapping a currently sequential process, and (iii) illustrating both the model and the framework with industrial applications thereby demonstrating their potential value.[1]

[1] The authors gratefully acknowledge the support for this research received from the Massachusetts Institute of Technology Leaders for Manufacturing Program, a partnership between MIT's management and engineering schools, and 13 major United States manufacturing firms.

References

Albano, R. E. and J. P. Keska, "Is Design Realization a Process? A Case Study," in *The Seventh International IEEE Electronic Manufacturing Technology Symposium*, IEEE, New York, 1989.

Blackburn, J. D., "New Product Development: The New Time Wars," in *Time-Based Competition*, J. D. Blackburn (Ed.), Business One Irwin Publishers, Homewood, IL, 1991.

Clark, K. B., B. Chew, and T. Fujimoto, "Product Development in the World Auto Industry: Strategy, Organization and Performance," *Brookings Papers on Economic Activity*, 3 (1987a), 729–771.

—— and T. Fujimoto, "Overlapping Problem Solving in Product Development," Harvard Business School Working Paper # 87–048, Cambridge, MA, 1987b.

—— and ——, "Reducing the Time to Market: The Case of the World Auto Industry," *Design Management J.*, 1, 1 (1989), 49–57.

—— and ——, *Product Development Performance: Strategy, Organization, and Management in the World Auto Industry*, Harvard Business School Press, Boston, MA, 1991.

Eastman, R. M., "Engineering Information Release Prior to Final Design Freeze," *IEEE Trans. Engineering Management*, 27, 2 (1980), 37–41.

Eppinger, S. D., D. E. Whitney, R. P. Smith, and D. A. Gebala, "A Model-Based Method for Organizing Tasks in Product Development," *Research in Engineering Design*, 6 (1994), 1–13.

Finger, S., M. S. Fox, F. B. Prinz, and J. R. Rinderle, "Concurrent Design," *Applied AI*, 6 (1992), 257–283.

Ha, A. and E. Porteus, "Optimal Timing of Reviews in Concurrent Design for Manufacturability," *Management Sci.*, 41 (September 1995), 1431–1447.

Imai, K., I. Nonaka, and H. Takeuchi, "Managing the New Product Development Process: How the Japanese Companies Learn and Unlearn," in *The Uneasy Alliance*, K. B. Clark, R. H. Hayes, and C. Lorenz (Eds.), Harvard Business School Press, Boston, MA, 1985.

Krishnan, V., *Design Process Improvement: Sequencing and Overlapping Activities in Product Development*, Doctoral Thesis, Massachusetts Institute of Technology, Cambridge, MA, 1993.

——, S. D. Eppinger, and D. E. Whitney, "Accelerating Product Development by the Exchange of Preliminary Product Design Information," *J. Mechanical Design*, 117 (December 1995), 491–498.

——, "Managing the Simultaneous Execution of Coupled Phases in Concurrent Product Development," *IEEE Trans. Engineering Management*, 43 (May 1996), 210–217.

Mansfield, E., J. Rapoport, J. Schnee, S. Wagner, and M. Hamburger, *Research and Innovation in the Modern Corporation*, First Ed., W. W. Norton & Company, New York, 1971.

McGrath, M. E., M. T. Anthony, and A. R. Shapiro, *Product Development: Success Through Product and Cycle-Time Excellence*, Butterworth-Heinemann Publishers, Boston, MA, 1992.

Nihtila, J., *Simultaneous Engineering and its Applications in Project Oriented Production—A Case Study in the European Project Oriented Industry*, Licentiate in Technology Thesis, Helsinki University of Technology, Finland, 1992.

Okamoto, I., A. Takahas, H. Sugiura, T. Hiramatsu, N. Yamada, and T. Mori, "Computer Aided Design and Evaluation System for Stamping Dies at Toyota," SAE 880524, Society of Automotive Engineers, Warrendale, PA, 1988.

Smith, P. G. and D. G. Reinertsen, *Developing Products in Half the Time*, Van Nostrand Reinhold, New York, 1991.

Smith, R. P. and S. D. Eppinger, "Identifying Controlling Features of Engineering Design Iteration," *Management Sci.*, 43(3) 1997, 276–293.

Takeuchi, H. and I. Nonaka, "The New New Product Development Game," *Harvard Business Rev.*, January-February (1986), 137–146.

Utterback, J. M., "Innovation in Industry and Diffusion of Technology," in *Readings in the Management of Innovation*, M. L. Tushman and W. L. Moore, Eds., Pitman Books, Marshfield, MA, 1982.

Ward, A. C. and W. Seering, "The Performance of a Mechanical Design Compiler," *J. Mechanical Design*, 115, 3 (1993), 341–345.

Whitney, D. E., "Designing the Design Process," *Research in Engineering Design*, 2 (1990), 3–13.

Accepted by Stephen Graves; received March 11, 1994. This paper has been with the authors 7 months for 2 revisions.