Communication and Uncertainty in Concurrent Engineering

Christoph H. Loch • Christian Terwiesch
INSEAD, Boulevard de Constance, 77305 Fontainebleau, France
The Wharton School, University of Pennsylvania, Philadelphia, Pennsylvania 19104-6366

We present an analytical model of concurrent engineering, where an upstream and a downstream task are overlapped to minimize time-to-market. The gain from overlapping activities must be weighed against the delay from rework that results from proceeding in parallel based on preliminary information. Communication reduces the negative effect of rework at the expense of communication time. We derive the optimal levels of concurrency combined with communication, and we analyze how these two decisions interact in the presence of uncertainty and dependence. Uncertainty is modeled via the average rate of engineering changes, and its reduction via the change of the modification rate over time. In addition, we model dependence by the impact the modifications impose on the downstream task. The model yields three main results. First, we present a dynamic decision rule for determining the optimal meeting schedule. The optimal meeting frequency follows the frequency of engineering changes over time, and it increases with the levels of uncertainty and dependence. Second, we derive the optimal concurrency between activities when communication follows the optimal pattern described by our decision rule. Uncertainty and dependence make concurrency less attractive, reducing the optimal overlap. However, the speed of uncertainty reduction may increase or decrease optimal overlap. Third, choosing communication and concurrency separately prevents achieving the optimal time-to-market, resulting in a need for coordination.

(*Product Development*; *Concurrent Engineering*; *Simultaneous Engineering*; *Overlapping*; *Communication Policy*; *Frontloading*; *Engineering Changes*; *Optimization*)

1. Introduction

In many industries, time-to-market emerged as as a key source of competitive advantage in the early 1990s (e.g., Blackburn 1991). Many tools have since been proposed to accelerate the product development process, prominent among which is the concept of concurrent engineering, whose benefits have been described in a large number of articles (e.g., Imai et al. 1985, Takeuchi and Nonaka 1986, Clark and Fujimoto 1991, Wheelwright and Clark 1992). Despite its popularity, there is recent empirical evidence that concurrency is not applicable to all product development projects (Eisenhardt and Tabrizi 1995, Terwiesch et al. 1996).

This recent conflicting evidence prompts us to investigate the applicability of concurrency in greater depth. The focus of our study can no longer be whether or not to overlap activities—overlapping has become a well-established part of best practice—but to probe more deeply. The present article develops an analytical model addressing the two questions of (1) *how much to overlap* activities depending on the project characteristics, and (2) *how to coordinate* the concurrent activities.

The model yields three main results. First, we present a dynamic decision rule for determining the optimal meeting schedule. The optimal meeting frequency follows the frequency of engineering changes (uncertainty reduction) over time, and it increases with the levels of uncertainty and dependence. Second, we derive the optimal concurrency between activities when communication follows the optimal pattern described by our de-

0025-1909/98/4408/1032$05.00

cision rule. Uncertainty and dependence make concurrency less attractive, reducing the optimal overlap. However, the speed of uncertainty reduction may increase or decrease optimal overlap. Third, the interaction of communication and concurrency may create local optima in the problem of finding the time-minimizing overlap level. In these cases, an organization would have to undertake a major process redesign to benefit from concurrency. Marginal improvement, even if targeted toward the global optimum, may increase rather than decrease development time. In addition, communication and overlap can not be determined in a decentralized way.

After reviewing the relevant literature on concurrent engineering in §2 of this article, in §3 we introduce the general mathematical model. Section 4 derives the optimal dynamic communication policy, and §5 the optimal concurrency level with optimal communication. Section 6 analyzes coordination prior to the start of the project and its impact on concurrency. We conclude with a discussion of managerial insights in §7.

2. Related Literature on Concurrent Engineering

Concurrent engineering is regarded as an important tool for reducing the time-to-market for new products. Blackburn et al. (1994) distinguish between time and information concurrency. Time concurrency refers to activities that are performed in parallel by different people or groups. Information concurrency refers to the degree to which information is shared among the involved parties.

The classical ''over the wall approach'' falls short on both counts: the development phases are performed in sequence, and information is transmitted only when the downstream phase begins. The importance of time concurrency for faster development processes was first widely publicized by Imai et al. (1985) and Takeuchi and Nonaka (1986). They also coined the metaphors ''relay race'' (one specialist passes the baton to the next as in the over the wall mode) and ''rugby team'' (a cross functional team on the project performing activities in parallel).

The importance of information sharing was emphasized in the studies by Clark and Fujimoto (1991) and Wheelwright and Clark (1992). The former observed that in their studies of the world automobile industry, companies with short development lead times not only overlapped their development activities, but complemented the overlap with frequent information transfer. Clark and Fujimoto call this combination of activity overlap and intensive communication ''integrated problem solving.''

Based on this work, concurrency has become a widely used tool for accelerating development processes (e.g., Griffin 1996). However, overlapping also involves significant risks. Eisenhardt and Tabrizi (1995) find in an empirical study that an ''experiential approach'' may be more promising than the overlapped ''compression approach'' if market uncertainty (''velocity'') is high. Similarly, Cordero (1991) recommends applying concurrency only in projects with moderate technical uncertainty. Hence, we ask the question: In which circumstances does concurrency accelerate product development, and when does it not?

Several modeling efforts have been put forward to address this question. The inherent limits to concurrency are described in a simple model by Hoedemaker et al. (1995). Ha and Porteus (1995) investigate a situation in which two development tasks are inherently *interdependent* and must be carried out in parallel to avoid quality problems. They develop the ''how frequent to meet'' problem as a dynamic program. If one design activity proceeds without incorporating information from the other, design flaws and corresponding rework result. Thus, parallel development together with design reviews save time and rework. Similar to a quality inspection problem in production, these gains have to be traded off with the time spent on review meetings. The main question is *how* to coordinate, i.e., how often to communicate. Our model, in contrast, examines a situation with a *sequential* task structure, i.e., where the tasks are logically consecutive. We show that even in this case overlapping may be beneficial to compress time-to-market, if complemented by an appropriate communication policy.

Krishnan et al. (1997) developped a framework for concurrency in case of sequentially dependent activities. It has had a strong influence on the emerging literature on modeling concurrency, and it is closely related to our work. The authors model preliminary information passed from an upstream to a downstream activity in the form of an *interval*. A parameter, e.g., the depth of a car door handle, is initially known only up to an interval, which narrows over time as the design becomes final. In this framework, two concepts determine the overlap trade-off. ''Evolution'' is defined as the speed at which the interval converges to a final upstream solution. ''Downstream sensitivity'' is defined as the duration of a downstream iteration to incorporate upstream changes associated with the narrowing of the interval. If upstream information is frozen before the interval has been reduced to a point value, a design quality loss occurs.

The authors formulate the problem as a mathematical program and show when overlapping (and thus preliminary information) should be used, and when upstream information should be frozen early (see Krishnan et al. 1997, Figure 9). With regard to our question of *how much to overlap*, the authors solve an application example numerically and suggest that ''generally, a fast evolution and low sensitivity situation is more favorable for overlapping'' (Krishnan et al. 1997, p. 11 and 22), although nonlinearities in the problem may lead to the optimal overlap being higher for a slow evolution process than for one with fast evolution.

The present paper differs from the work of Krishnan et al. in three important aspects. First, we conceptualize preliminary information differently, with it being precise from the start, but then being modified repeatedly as the design evolves. These modifications are incorporated downstream through engineering changes (ECs), a concept that is widely used in industry. One key aspect of ECs is that they virtually always become more difficult to implement the later they occur (Terwiesch and Loch 1998). This increasing impact of ECs plays an important role in deciding by how much to overlap.

Second, we extend the Krishnan et al. research by explicitly incorporating an appropriate information batching policy, addressing the question of how to coordinate the overlapped activities. The batching of ECs is frequently observed in practice and can be prompted by communication times (similar to Ha and Porteus 1995) or by setups required for tool changes.

Third, by incorporating both dimensions of Clark and Fujimoto's concept of ''integrated problem solving,'' namely concurrency and coordination, we show that they interact in a fundamental way. They cannot, therefore, be managed separately: marginally adjusting one, with the other fixed, will not lead to an overall time-to-market optimum.

Empirical support of our findings is provided by Terwiesch et al. (1996), where it is demonstrated that later uncertainty reduction reduces the time benefits of overlapping in electronics development projects.

3. The Model

Consider the duration of a project with two tasks of length T_1 and T_2, respectively. In the product development process, the reader may picture T_1 as the time of product design and T_2 the time of process design. Similarly, in software development, the first phase could be specification development and the second coding. We call the first activity upstream and the second downstream. In the ''over the wall'' approach, total completion time is $T_1 + T_2$. The objective is to minimize the total completion time. Assume that some proportion λ of T_2 can be conducted in parallel to T_1. If $T_2 \geq T_1$, no more than T_1/T_2 can be parallelized. Let $\Lambda = [0, \lambda_{\max}]$, where $\lambda_{\max} = \min\{1, T_1/T_2\}$, be the interval of all possible levels of overlapping.[1] λ provides a continuous measure of concurrency. Without taking into account any drawbacks of concurrency, the project completion time T benefits from overlapping:

$$T = T_1 + (1 - \lambda)T_2. \tag{1}$$

Although overlap creates an immediate time advantage, it is not without drawbacks. In a fully sequential process, downstream starts with finalized information from upstream, whereas in an overlapping process it has to rely on preliminary information. This approach can be risky if the upstream information may change substantially or if there exists a strong dependence between the activities. Under these conditions, engineering changes (of upstream information) may cause downstream rework delaying the whole project (Eastman 1980), cre-

[1] $\lambda_{\max}$ may actually be smaller than this if downstream cannot start at the same time as upstream, having to wait for intermediate results to start their work. For simplicity of exposition, we omit this effect of ''work authorization.''

ating a trade-off between time gains from overlapping and time rework delays.

3.1. Uncertainty and Evolution

Engineering changes arrive upstream according to a stochastic process with a time-dependent mean. These changes affect the preliminary information based on which downstream has begun work. We assume that these changes follow a nonstationary Poisson process with rate $\mu_\alpha(t^{\text{up}})$, defined over the upstream task duration $t^{\text{up}} \in [0, T_1]$. The Poisson assumption is frequently made in models of quality (e.g., Lee and Rosenblatt 1986 for a model of a machine breaking down according to a Poisson process) and reliability (e.g., Ramamoorthy and Bastani 1982 offer a model of software defects). It is justified when modifications arise from many modules or project participants, each being a potential source of requests for engineering changes.

We consider a situation of *sequential* dependence, i.e., downstream must readjust its work if upstream changes its design in an unexpected way. This corresponds to a situation where downstream engineers have to include the final upstream information, including all modifications, even if they are communicated after the downstream start. This does not exclude a close information exchange upfront (such as general design rules or downstream process limits). Such upfront understanding is, however, static and does not substitute for dynamic communication over the course of a project.

Reciprocal dependence inherently forces overlap and joint problem solving because neither task can proceed without the other (Van de Ven and Delbecq 1974). The question then is how to best coordinate, a problem analyzed by Ha and Porteus (1995). We model *sequential* dependence because it is commonly practiced even for activities that are logically consecutive, in order to compress the development cycle. Krishnan (1996) describes such a situation in the development of an instrument panel. Other examples include development and die design of an automotive door (Krishnan et al. 1997), the rudder design, including the supplier, of the Boeing 777 (Sabbagh 1996), and the flying start of software development phases (Blackburn et al. 1996). For such sequentially dependent activities, overlapping is inherently risky. Our model offers insights under which circumstances the risks may be justified by the compression time gain.

The overall level of uncertainty is denoted by μ_α, the average rate of upstream changes affecting downstream work. This overall uncertainty can be reduced through coordination prior to the start of the upstream activity, such as the definition of proven product technologies or approved parts databases for the specific project. Adler (1995) describes how such pre-communication may satisfy the overall coordination requirement between design and manufacturing if the design problem at hand is routine and can be solved using past solutions. Let α be the total number of coordination meetings before the development work starts. The modification rate reduction exhibits decreasing returns (consistent with empirical results, e.g., Adler 1995, p. 161):

$$\mu_\alpha = \mu_0 \exp\{-B\alpha\}. \tag{2}$$

μ_0 represents the inherent technical uncertainty of the project, or the rate of modifications in the absence of planning or coordination. The parameter B represents the organization's capability to reduce uncertainty during the precommunication phase: the higher the B, the better the reduction effect achieved with a given communication intensity. This parameter reflects the degree of partnership and integration, e.g., the effectiveness with which downstream engineers not only see the early outlines of the design, but may even influence it to make their own task easier. Precommunication does have a cost; we assume a linear cost $\tau_1\alpha$, which can be interpreted as the average total meeting time required for *ex ante* integration.

The nonhomogeneity of $\mu_\alpha(t^{\text{up}})$ represents the progress (uncertainty reduction) of the upstream task. If evolution to a stable design is fast, then $\mu_\alpha(t^{\text{up}})$ is high at the beginning and drops as stability is approached. If, in contrast, upstream convergence to a design solution is slow, then $\mu_\alpha(t^{\text{up}})$ begins low and rises as the design concept evolves at the end. For simplicity of exposition, we model $\mu_\alpha(t^{\text{up}})$ as a *linear* function:

$$\mu_\alpha(t^{\text{up}}) = \mu_\alpha\left[1 + e\left(2\frac{t^{\text{up}}}{T_1} - 1\right)\right]. \tag{3}$$

The integral over the modifications corresponds to Krishnan et al.'s (1997) evolution function: If e is negative (the initial rate of changes is high), then much progress occurs early over the upstream activity. The parameter $e \in [-1, 1]$ is a shape parameter for $\mu_\alpha(t^{\text{up}})$. It is called the *evolution parameter*. When $e > 0$, then $\mu_\alpha(t^{\text{up}})$ increases over t^{up}, corresponding to slow evolution. When $e < 0$, then $\mu_\alpha(t^{\text{up}})$ decreases, corresponding to fast evolution. When $e = 0$, then modifications are generated as a homogeneous Poisson process ($\mu_\alpha(t^{\text{up}}) = \mu_\alpha$). In all three cases, the total expected number of changes generated over the time of task 1 is the same, namely $\mu_\alpha T_1$. Thus, e represents the evolution of uncertainty over time, while μ_α represents the level of uncertainty after precommunication, and μ_0 the inherent level of uncertainty before precommunication.

3.2. Downstream Sensitivity

The amount of downstream rework created by a modification depends on how far downstream has already progressed in its problem solving. We define the impact function $f(t)$ over downstream duration as the time it takes to change previous downstream work, if a modification is communicated at $t \in [0, T_2]$ units of downstream time. This time delay is added to the project completion time. Our approach is based on the concept of downstream sensitivity, developed by Krishnan et al. (1997). Changes in the preliminary information received so far will delay downstream activity. The more downstream has progressed in its work, the more cumulative work must be modified. Hence, $f(t)$ is nondecreasing.

Modifications and impact are closely related to Adler's (1995) definitions of fit novelty and fit analyzability. Upstream passes on preliminary information based on its existing experience base, such as preferred parts lists reflecting the downstream cost structure. In our model, $\mu_\alpha(t)$ is the rate of deviations from this preliminary information and thus corresponds to fit novelty, the newness of the design solution chosen upstream.[2] Downstream dependence, modeled by the impact function $f(t)$, captures the idea of fit analyzability, defined by Adler as the time it takes to resolve a given fit problem. Early in downstream progress, the modules affected by a modification can be easily identified and changed. Changes become more difficult with the growing size of the product or system already developed.

In the analytical model below, we focus attention on linear impact functions ($f(t) = kt$). In practice, the impact function might be concave or convex. Learning effects or a personnel buildup may increase downstream progress speed, implying a convex impact function. Similarly, if modifications not only require rework, but make large parts of achieved progress obsolete, $f(t)$ grows faster than linearly. On the other hand, $f(t)$ might be concave if late upstream changes consist only of modifications to a production machine, which was purchased and installed earlier during the downstream activity. A linear impact function corresponds to an ''average'' situation; it also allows us to analytically derive that adding communication to the overlap problem introduces a fundamental interaction between the two.[3]

3.3. Concurrent Communication and Rework

During the overlapped phase, communication occurs according to a communication policy $\mathcal{C}(t)$ with $t \in [0, \lambda T_2]$. When product and process engineers sit together in a cross-functional team meeting at time t, they discuss the latest changes in product design for downstream incorporation, and they set a rule for calling the next meeting (the policy will be specified below). Downstream will not become aware of any new engineering changes until the next meeting. Thus, more frequent meetings reduce communication delay of modifications. If communication were costless, then the team would optimally communicate and incorporate each modification immediately.

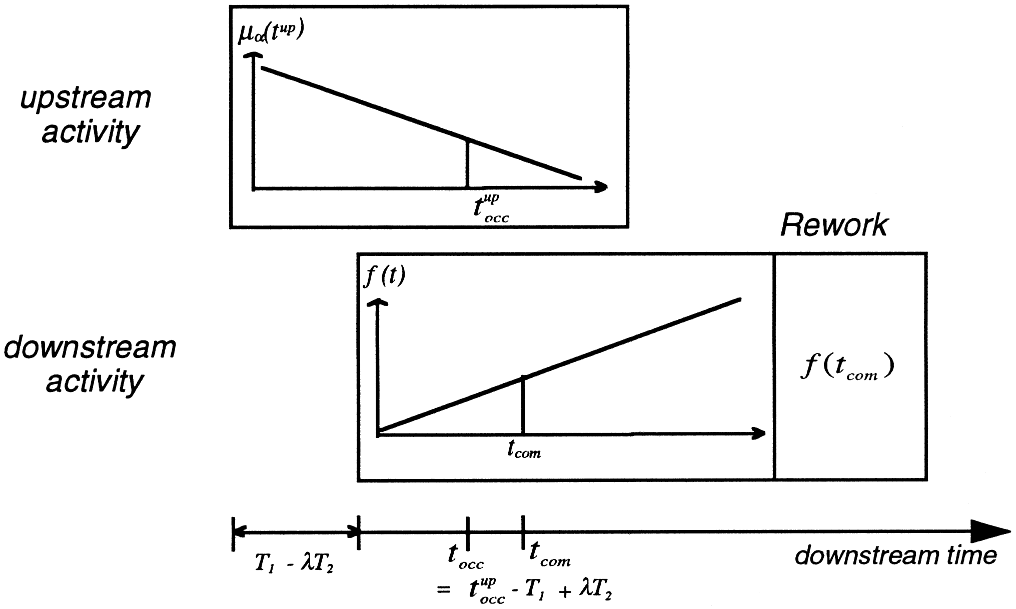
Figure 1 demonstrates the communication benefit. If a modification occurs at upstream time $t^{\text{up}}_{\text{occ}}$, according to the time-dependent Poisson process, it is communicated at the time of the next meeting, t_{com} in downstream time. The longer the delay until the meeting, the larger becomes the impact $f(t_{\text{com}})$ of the modification.

While communication reduces delays, it also carries a cost. Team meetings require valuable engineering time (spread out over the overlap time in Figure 1) pos-

[2] This newness may stem from unexpected downstream challenges, such as the fitting of components, or from unexpected upstream problems, such as lack of stability in the design.

[3] We have examined the effect of convex and concave impact functions via numerical examples (available from the authors upon request). Their effect was as expected: A convex impact function increases the optimization problem's convexity mitigating the concavity effects described below, while a concave impact function makes the problem more concave. We have also simulated non-Poisson modification arrivals and found that our results are robust.

Figure 1 Communication and Dependence for One Modification



sibly delaying project completion. For example, Iansiti (1995) quotes a senior executive of a mainframe manufacturer: ''We no longer have the luxury to spend much time communicating—the problems are too complex and time is too tight. . . .''

While communication represents one source of set-up costs, and thus one reason for batching ECs, there are several others. In die development, for example, there are substantial set-ups required before one can start recutting or welding dies (e.g., taking the die out of the press). Similarly, in software development, rewritten code has to be recompiled and tested. Such set-up costs make it easier to implement two ECs in one batch rather than to implement them individually, which corresponds to sub-additivity of the rework function. For the remainder of this article, we let τ_2 denote the fixed set-up time per batch. We label τ_2 communication costs, consistent with the terminology in Ha and Porteus (1995), but any other type of set-up time applies equally well.

The total set-up time of information batches accumulates in expectation at the rate $\beta(t)\tau_2$, where $\beta(t)$ is the expected communication rate resulting from the policy $\mathcal{C}$. This is similar to the EOQ-like structure in Ha and Porteus (1995): The ''setup cost'' corresponds to τ_2, and the ''holding cost'' of a modification to the additional impact caused by a communication delay, $f(t_{\text{com}}) - f(t_{\text{occ}})$.

3.4. Summary of Model Parameters and Assumptions

We have now defined all the elements of the model and can state the time-to-market optimization problem:

$$\min_{\alpha,\mathcal{C}(t),\lambda} : ET = T_1 + (1-\lambda)T_2 + \alpha\tau_1 + EC(\alpha, \mathcal{C}(t), \lambda) + ER(\alpha, \mathcal{C}(t), \lambda); \quad (4)$$

subject to: Equations (3), (2);

$$0 \le \alpha; \quad \lambda \in \Lambda; \quad (5)$$

$$\mathcal{C}(t) \text{ depends only on modifications up to time } t. \quad (6)$$

$ER(\alpha, \mathcal{C}(t), \lambda)$ is the expected rework[4] resulting from the combination of overlap λ, precommunication intensity α, and the concurrent communication policy $\mathcal{C}(t)$. $EC(\alpha, \mathcal{C}(t), \lambda)$ is the delay caused by meetings during the overlap period. We will derive the optimal communication

[4] We do not consider the time-to-market *variance* in this article. The variance increases with the overlap level because overlapping introduces (rework) uncertainty. Simulation results showing this can be obtained from the authors.

Table 1 Model Parameters and Decision Variables

τ_1	Communication cost for precommunication (time per meeting)
τ_2	Communication cost for concurrent communication (time per batch)
T_1	Upstream task time
T_2	Downstream task time
μ_0	Basic rate of modifications without precommunication
μ_α	Basic rate of modifications, depending on level of precommunication
$\mu_\alpha(t)$	Time-dependent rate of the Poisson process of generation of modifications by the upstream task team
B	Precommunication capability parameter: reduction in the rate of modifications from one unit of additional communication
k	Impact of a modification (time units per engineering change)
e	Evolution parameter: if -1, fast evolution; if $+1$, slow evolution
	Decision Variables:
α	Precommunication intensity
$\beta(t)$	Expected communication frequency resulting from the concurrent communication policy (time-dependent during the overlap period)
λ	Overlap, % of downstream task length

policy and the resulting expected communication rate $\beta(t)$ in the next section. All variables are summarized in Table 1.

The model focuses on minimizing total expected project completion time (time-to-market). We are not including the design quality resulting from the project. We are in effect assuming that the project teams have to work until a certain required quality standard is met, and the only question is whether they can organize themselves in such a way as to achieve this quality as quickly as possible. This focus is appropriate in light of the large differences that persist, for example, in the automotive industry: Currently (1996), the fastest Japanese producers take under 20 months for the development of a new car, while the slower U.S. companies, at 48 months, take more than twice as long without appreciable quality advantages. We are not including project costs, either. If project costs are linear for the various types of activities (see, e.g., Chakravarty 1995), they are equivalent to a set of weights on the completion times of the different tasks, and thus easy to include in our model.

The critical assumption driving the results of the model is the sequential dependence of the downstream task as discussed above. The other assumptions in our model are purely computational: the generation of modifications according to a Poisson process with a linear rate, the linear impact function, and the linear meeting costs of communication are required to allow closed form results offering structural insights. The same holds for the assumption that the basic task times T_1 and T_2 are deterministic—the source of uncertainty in the model is purely the generation of changes. The model is numerically analyzable also for more general functions and random task times (see footnote 3). Finally, engineering changes are modeled as coming in "packets" of equal size (through the parameter k). This simplification can be relaxed immediately: If modification work content varies *independently* of the arrival process, all our results remain unchanged.

In §4 we derive the optimal concurrent communication policy $\mathcal{C}(t)$. Section 5 explores the interaction between concurrent communication and overlap, and §6 that between precommunication and overlap.

4. Optimal Concurrent Communication

Precommunication α and overlap λ are already determined when concurrent communication is carried out. Therefore, the overall optimum of the decision problem (4) will be achieved by optimizing over α and λ, anticipating optimal concurrent communication. In this section, we characterize the optimal dynamic concurrent communication policy for any given level of α and λ. We assume that the overlap period is long in comparison to the time between two modification occurrences, ignoring the influence of end effects.[5] The result is stated in Theorem 1.

THEOREM 1. *The optimal dynamic communication policy $\mathcal{C}(t)$ is characterized as follows.*

1. *Any communication meeting is held directly after a modification occurs.*

2. *After a modification occurring at time s, a meeting is held if the number n of modifications pending (occurred, but*

[5] In technical terms, Theorem 1 assumes an infinite horizon. The end effect consists in a meeting being skipped if its communication cost outweighs the additional delay of modifications all the way to the end of the overlap period.

not yet communicated) is at least as large as a critical value $n^*(s)$*, and no meeting is held otherwise. Moreover,*

$$n^*(s) = \sqrt{\frac{2\tau_2 \mu_\alpha(s)}{k}}. \tag{7}$$

3. *The resulting expected communication frequency is*

$$\beta^*(s) = \sqrt{\frac{k\mu_\alpha(s)}{2\tau_2}}. \tag{8}$$

4. *The resulting value of the objective function, given* α *and* λ*, is*

$$\begin{aligned} ET(\alpha, \lambda) = {} & T_1 + (1 - \lambda)T_2 + \alpha\tau_1 \\ & + \int_0^{\lambda T_2} kt\mu_\alpha(t + T_1 - \lambda T_2)dt \\ & + \int_0^{\lambda T_2} \sqrt{2k\tau_2\mu_\alpha(t + T_1 - \lambda T_2)} - \frac{k}{2}\, dt. \end{aligned} \tag{9}$$

PROOF. For easier readability of the text, all proofs are shown in the appendix.

Theorem 1 describes the following structure. Information transfers occur only directly after modifications, because otherwise rework could be saved by holding the meeting earlier, without any other changes. The upstream team, who are informed about the rate of modifications, $\mu_\alpha(t)$, and about the rework rate k, initiate an information transfer whenever a critical number of modifications have been accumulated. Thus, upstream attempts to balance communication costs with the additional rework from waiting too long.

Equation (7) shows that the critical number of modifications triggering a meeting, $n^*(t)$, changes over time as the rate of modifications changes. If evolution is slow, that is, $\mu_\alpha(t)$ increases over time ($e > 0$), the resulting optimal communication frequency $\beta^*(t)$ increases over time, and if evolution is fast ($e < 0$), the communication frequency decreases over time. When e is zero, the optimal communication intensity stays constant. This is a structural result that Ha and Porteus (1995) also obtain in their numerical example. The optimal adaptation of communication both to e and over time t occurs at a decreasing rate, namely with the square root.

In addition, higher overall uncertainty μ_0 (pushing up $\mu_\alpha(t)$) and downstream dependence k increase the optimal frequency of communication throughout. A smaller communication cost τ_2 also leads to higher optimal communication, corresponding to a higher ''communication capacity'' of the two project teams. These findings are consistent with empirical findings in the organization literature (e.g., Tushman 1978).

Finally, the resulting time-to-market objective function in Equation (9) is *separable* into three parts: the first two summands represent the direct time gain from overlapping. The first integral describes the expected ''minimal'' rework resulting from proceeding in parallel when communication is instantaneous. The second integral represents the additional communication time and rework from delays. This second integral disappears if communication costs are zero and thus no delays are necessary.

Equations (7) and (9) are approximations, which are accurate as long as $\mu_\alpha(t^{\text{up}})$ does not change much from one information transfer to the next (this is made precise in Proposition 3 in the appendix). In the situations of interest here this is justified, as there will usually be a number of information transfers over the duration of the overlap period. In other words, the change in the rate of modifications is important over the course of several meetings, but not within one intermeeting period.

5. Concurrent Communication and Overlap

In this section, we assume the precommunication intensity α to be exogenously determined. We thus focus on the interaction between concurrent communication and overlap, given an average rate of engineering changes μ_α. The resulting simplified time-to-market problem can be written as an optimization over λ, anticipating the optimal communication policy and the resulting rework as derived in Theorem 1.

$$\begin{aligned} \min_{\lambda \in \Lambda}: ET = {} & T_1 + (1 - \lambda)T_2 \\ & + \int_{T_1 - \lambda T_2}^{T_1} k(t - T_1 + \lambda T_2)\mu_\alpha(t)dt \\ & + \int_{T_1 - \lambda T_2}^{T_1} \sqrt{2\tau_2\mu_\alpha(t)k} - \frac{k}{2}\, dt; \end{aligned} \tag{10}$$

We see the same components of time-to-market as in the general problem: overlap gain, unavoidable rework and communication delay. Before we can derive the optimal level of overlap λ we need to state a technical characterization of the shape of the objective function depending on e. This technical result is presented here, because it is interesting in its own right and reveals the structure of the trade off in question:

PROPOSITION 1. *There are $0 < \underline{e} < \bar{e}$, with $\underline{e} < 1$, such that*

1. *if $e \le \underline{e}$ then ET is convex in λ,*
2. *if $\underline{e} < e \le \bar{e}$ then ET is convex-concave in λ, and*
3. *if $e > \bar{e}$ then ET is concave in λ.*

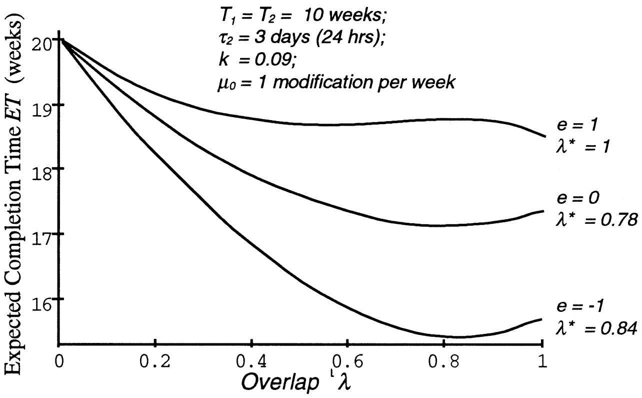
The values of $\underline{e}$ and $\bar{e}$ are derived in the proof. The reader may note in the proof that if we restrict attention to the interesting cases where more than one meeting over the overlap period is optimal ($\lambda T_2 \beta^* > 1$ in (8)), then $e > \bar{e}$ will not occur. This case is still interesting in the sense that for concave ET, the optimal overlap is either zero or $\lambda_{\max}$: if it is worthwhile to overlap at all and suffer the rework penalty of the many late changes, it is also optimal to go all the way and overlap fully.

Proposition 1 explains the impact of evolution on time-to-market. If e is negative, most modifications arise early during task 1. If task 2 overlaps only a little, it is affected only by a few changes that have to be made. As overlap increases, not only does the time period increase over which modifications arise, but so do the rate of modifications and their corresponding impact. Therefore, the rework increases at an increasing rate with overlap, which leads to convexity of ET.

If, on the other hand, e is large and positive, most of the modifications arise at the end of task 1. While the unavoidable rework is still convex in λ, this is not true for the communication delay. Since for large e the meeting frequency increases towards the end of the project, an extra percent of overlap will extend the overlap period to the earlier, less modification-prone portion of the upstream activity. Thus, for large e, the communication delay is concave, and as a result the overall objective function is convex-concave. The argument is presented more formally in the proof.

This situation is illustrated by the numerical example in Figure 2. For small values of e, ET is convex in λ. With e increasing, this convexity changes to convex-concavity as depicted in the upper two curves. This shape of ET is important from a managerial perspective. Consider an organization attempting to improve time-to-market with given parameters k, μ_α, and e across similar projects. If ET is convex, the organization can rely on marginal improvement: any change in overlap that decreases the project completion time is a step towards the globally optimal level of concurrency. If, however, e is large and ET is convex-concave, an incremental improvement policy may trap the organization in a local optimum (e.g., $\lambda = 0.5$ in the upper curve of Figure 2). Under these conditions only a large change in overlap will allow a further reduction of project completion time (e.g. moving to $\lambda = 1$). Conversely, projects with only slightly different parameters may require substantially different overlap levels.

Figure 2 **Influence of Evolution on Time-to-Market: Numerical Example**



With the result from Proposition 1 we can now characterize the optimal overlap and provide comparative statics on the parameters for impact, uncertainty, and evolution, k, μ_α and e, respectively. The result says that *higher uncertainty as well as higher dependence decrease the optimal amount of overlap, but slower evolution may increase or decrease optimal overlap*. Together with the convex-concavity discussed above, this is the second main result of this article.

THEOREM 2. *Higher impact k, or a higher uncertainty level μ_α each separately decreases the optimal degree of overlap, λ^*. This decrease is continuous when $e < \underline{e}$, and discontinuous (from $\lambda_{\max}$ to 0) when $e > \bar{e}$.*

Slower evolution (increasing e) may increase or decrease the optimal overlap.

1. *If $e < \bar{e}$, then there exists a $\bar{\lambda}$ such that: If the solution*

to the FOC λ_{FOC} lies to the left of $\bar{\lambda}$, it decreases with e. If λ_{FOC} lies to the right of $\bar{\lambda}$, it increases with e.

2. *When $\bar{e} < e$, then λ^* decreases with a discontinuous drop with e as it does with k and μ_α.*

Higher uncertainty and impact reduce the optimal overlap. However, the effect of evolution is more complicated. In many cases, slower evolution will reduce the optimal overlap, but not always. The numerical example in Figure 2 demonstrates why. ET is convex for small e, and strictly increases everywhere with growing e. But as the curve moves toward concavity, it increases more in the ''center'' $\bar{\lambda}$ (which is derived in the proof) than on either side. Thus, if the optimal overlap is to the right of this center, it is pushed further to the right. Moreover, it may jump to 1 if the curve decreases at $\lambda = \lambda_{\max}$. Figure 2 also illustrates the sensitivity of project completion time $ET(\lambda^*)$ with respect to changes in e. A slower evolution results in longer project duration and fewer benefits from overlap, the latter of which has been empirically confirmed in Terwiesch and Loch (1996).

Corollary 1 describes the special case of our time-to-market problem (10) without communication delays.

COROLLARY 1. *If the communication cost $\tau_2 = 0$, it is optimal to communicate each engineering change immediately. In this case, the objective function is convex in λ for all e, and if the optimal overlap is interior, it decreases strictly with e, k, and μ_α.*

Thus, introducing communication into the optimal overlap problem makes the impact of uncertainty reduction more complicated. Faster evolution may imply a *lower* optimal overlap level instead of a higher one. Moreover, incremental improvement no longer necessarily works. When the problem parameters change, the organization may have to take a drastic step in order to reach the globally optimal concurrency level.

6. Precommunication and Overlap

In this section we explore the interaction between overlap and precommunication, using again optimal concurrent communication once overlap is set. Therefore, we examine a specialized version of the general model (4) by setting the evolution parameter e to 0. This makes the change generation process a homogeneous Poisson process with rate μ_α. By Theorem 1, the optimal concurrent communication frequency is in expectation a constant rate $\beta = \sqrt{k\mu_\alpha/2\tau_2}$. Thus, the minimization problem (4) simplifies to:

$$\min_{\alpha,\lambda} ET = T_1 + (1-\lambda)T_2 + \alpha\tau_1 + \lambda T_2\left(\frac{1}{2}k\mu_0\exp\{-B\alpha\}\lambda T_2 + \sqrt{2k\tau_2\mu_0\exp\{-B\alpha\}} - \frac{k}{2}\right); \qquad (11)$$

$$\text{subject to: } \lambda \in \Lambda; \quad 0 \le \alpha.$$

The objective function still comprises the same components as before: overlap time, precommunication time, unavoidable rework over the overlap period, and communication time and rework from delays. As in §5, we need a technical result first, which is again included in the body of the text because it reveals the structural properties of the optimization problem we are facing.

PROPOSITION 2. *In the interior region of the minimization problem* (11), *the Hessian matrix of* (11) *is convex in the directions of λ and α separately, but indefinite overall. Let λ_{FOC} be the solution of the first order condition for λ. Then the objective function in the direction (α, given λ_{FOC}), is strictly concave in the interior region.*

Proposition 2 illuminates the structure of the problem. First, the positive second derivative in the λ direction indicates that if precommunication is exogenously determined, the decision problem for overlap is convex, and the solution is given by the first order condition (FOC), or by the nearest boundary if the solution to the FOC is infeasible.

Second, the interaction between precommunication and overlap introduces a saddle point into the problem. In particular, the objective function is concave in the direction α given λ_{FOC}. That is, the optimal precommunication level, and thus the optimal overlap, must be an extreme solution at one of the boundaries.

Where does this concavity come from? Recall that the value of precommunication lies in a priori coordination of the two task teams, reducing μ_α, the basic rate of upstream modifications and thus rework. Precommunication exhibits decreasing returns: at high levels of α, a further increase will reduce rework only very little. If the level of overlap is exogenously fixed, this results in a convex objective function, where the optimal solution balances marginal rework reduction from precommunication with the marginal communication cost. If, however, overlap is adjusted optimally, it overcompensates. Rework actually increases slightly with precommunication because the change rate reduction is more than offset by the longer concurrency period over which changes occur. This overcompensating effect of the optimal overlap causes the problem to become concave.

Thus, precommunication either has enough benefit to push overlap all the way, or it is not at all worthwhile. This is made precise in Theorem 3, the final main result of our model discussed in this article.

THEOREM 3. *The optimal solution to the problem* (11) *is to either extensively precommunicate* (*frontload*) *and then overlap fully, or to not precommunicate and then proceed sequentially. When* $\lambda_{\max} = 1$, *the solutions are* (*the case* $\lambda_{\max} = T_1/T_2$ *is analogous*):

Solution 1
(sequential):

$$\lambda_{\text{sequ}} = \left\{ \begin{array}{ll} 0 \quad \text{if } \mu_0 \geq \dfrac{(1+k/2)^2}{2k\tau_2}, & \\ \min\left\{1, \dfrac{1}{k\mu_0}\left(1-\sqrt{2k\mu_0\tau_2}\right)\right\} & \text{if } \mu_0 < \dfrac{(1+k/2)^2}{2k\tau_2}; \end{array} \right\}$$

$$\alpha_{\text{sequ}} = 0; \text{thus } \mu_1 = \mu_0;$$

$$ET_{\text{sequ}} = T_1 + T_2. \tag{12}$$

Solution 2
(parallel):

$$\lambda_{\text{paral}} = 1; \alpha_{\text{paral}} = \frac{1}{B}\ln\left(\frac{\mu_0}{\mu_{\text{paral}}}\right);$$

$$\mu_{\text{paral}} = \frac{1}{kT_2}\min\left\{ \begin{array}{l} 1 + \dfrac{k}{2} + \dfrac{\tau_2}{T_2} - \sqrt{\left(1 + \dfrac{k}{2} + \dfrac{\tau_2}{T_2}\right)^2 - \left(1 + \dfrac{k}{2}\right)^2} \\ \dfrac{\tau_1}{BT_2} + \dfrac{k\tau_2}{2T_2} - \sqrt{\left(\dfrac{\tau_1}{BT_2} + \dfrac{k\tau_2}{2T_2}\right)^2 - \left(\dfrac{2\tau_1}{BT_2}\right)^2}; \end{array} \right.$$

$$ET_{\text{paral}} = T_1 + \frac{1}{2}kT_2^2\mu_{\text{paral}} + \tau_1\alpha_{\text{paral}} + T_2\left(\sqrt{2k\tau_2\mu_{\text{paral}}} - \frac{k}{2}\right). \tag{13}$$

Which one of the two solutions is optimal depends on the specific parameter constellation. By inspecting the parameters in ET_{paral} and μ_{paral}, we can see that the fully parallel solution will tend to be preferred if: first, communication delays τ_1 and τ_2 are low, corresponding to high communication capacity; second, if the precommunication capability, B, is high; and third, if the impact parameter k is low, corresponding to low downstream dependence. In the "sequential" solution, a positive overlap may still occur if the basic project uncertainty μ_0 is so low (Equation 12), that overlap can outweigh the resulting rework, even without precommunication. This corresponds to a very low complexity project where overlap can be risked purely based on past knowledge. Even in this case, concurrent communication is required in the optimal solution.

These model results are consistent with the findings discussed in the concurrent engineering literature: activity overlap coupled with task dependence and novelty requires communication, along with its associated delays. The lower dependence and communication delays are, the lower the costs in this trade off. And with higher precommunication efficiency one can push down task novelty and thus shift the trade off toward more overlap. In addition, our model shows that an organization may not be able to set frontloading and overlap in a decentralized fashion. For example, if the engineering function sets precommunication, and project management determines overlap and concurrent communication, each adjusting marginally to parameter changes given what the other group does, an optimum may not be reached. A *coordinated* decision is necessary to set both either high or to zero.

7. Discussion and Conclusion

We have presented an analytical model of concurrent engineering, which combines the decisions of overlap and communication (before and during the overlap phase) in the presence of uncertainty and dependence between tasks, with the goal of minimizing time-to-market. In determining the optimal overlap, the gain from overlapping activities has to be weighed against the rework delay resulting from the use of preliminary information by the downstream task. Communication reduces the negative effect of rework at the expense of set-up times per information batch.

We model uncertainty as the basic rate of modifications (''fit novelty'' in the terminology of Adler 1995), and uncertainty reduction as the change of the modification rate over time. In addition, we model dependence by the delaying impact imposed by upstream modifications on the downstream task. This is similar to Adler's (1995) ''fit analyzability'' and to ''downstream sensitivity'' in Krishnan et al. (1997), with the additional characteristic that the cost of a change increases over downstream time.

We have developed three managerial results about the interaction between communication and overlap and their connection to uncertainty and dependence. First, we are able to characterize the optimal concurrent communication policy, which results in an expected communication frequency increasing over time if evolution is slow, and decreasing if evolution is fast. Moreover, the average communication level increases with uncertainty and dependence. This is consistent with Adler's (1995) empirical finding and with the numerical example in Ha and Porteus (1995).

Second, both uncertainty and dependence make concurrency less attractive, thus reducing the optimal overlap. This finding complements recent empirical studies (e.g., Eisenhardt and Tabrizi 1995) with a causal explanation. However, the speed of uncertainty reduction (evolution) has a more complicated effect: a high optimal overlap will be increased further by slower evolution, and a low optimal overlap will be decreased further. The conclusion that slower uncertainty reduction reduces the optimal overlap level is correct only in the special case of instantaneous communication.

Third, the interaction of communication and concurrency may create local optima in the problem of finding the time-minimizing overlap level. When uncertainty resolution is slow, the optimal overlap combined with concurrent communication may jump when problem parameters change. Similarly, precommunication either reduces uncertainty sufficiently to bring overlap to a high level, or it is not at all worthwhile (and thus sequential execution of the tasks is preferred). This analytical result has important managerial implications. Incremental improvement over consecutive projects may no longer work. Approaching the optimal concurrency through small trial-and-error steps may trap the organization in a local optimum; a drastic step change may be required. Conversely, a small change in the problem parameters may require a major change in overlap and communication to again minimize time-to-market. Our model illuminates how an improvement of these problem parameters (uncertainty, evolution and dependence, and communication capability) can improve optimal performance (time-to-market).

Another implication of the interaction between communication and concurrency is that an organization should not choose communication and overlap levels in a decentralized fashion. For example, consider a situation where the engineering function decides precommunication, and project management separately decides overlap and concurrent communication. We have shown that even if each anticipates the other's behavior (assuming an optimal decision by the other group), the global optimum may not be reached. Thus, communication and overlap have to be chosen in a *coordinated* way, which may limit the autonomy of project teams.

We have kept the model as simple as possible in order to focus on structural results. Refinements of the model are possible in several directions. First, the influence of overlap and communication on the task times themselves should be further investigated. Second, we have focused on time-to-market, but trade-offs between time-to-market, project costs, and design quality deserve further attention. Finally, our model can also be analyzed numerically in more complicated and realistic applications to actual cases.

We have generated empirically testable hypotheses on the effectiveness of overlap and on the impact of preliminary information in development. Empirical investigations of activity overlap in concurrent engineering are scarce (Clark and Fujimoto 1991, Eisenhardt and Tabrizi 1995, and Terwiesch et al. 1996) and offer interesting research opportunities.[6]

[6] The authors thank the editor, the associate editor and three anonymous referees for helping to substantially improve this manuscript.

Appendix

Proof of Theorem 1

We assume an infinite horizon and $\mu(t)$ constant between two meetings (which corresponds to a situation where the decision makers only estimate the modification rate when they meet). We proceed by proving three lemmas.

LEMMA 1. *Meetings are optimally held only directly after modifications occur.*

PROOF. Let a meeting be considered every Δt time units for some small Δt. With independent modification interarrival times, we can define the state of a Markov decision process by n, the number of modifications pending, t, the downstream time, and t_l, the time of the last modification. Then the dynamic program recursion becomes, with the first line corresponding to the action "meet" and the second to "not meet" (Heyman and Sobel 1984, 115 ff.):

$$V(n, t, t_l) = \min \begin{cases} \tau_2 + p(t, t_l)V(1, t + \Delta t, t_l) \\ \quad + (1 - p(t, t_l))V(0, t + \Delta t, t_l) \\ kn\Delta t + p(t, t_l)V(n + 1, t + \Delta t, t_l) \\ \quad + (1 - p(t, t_l))V(n, t + \Delta t, t_l), \end{cases} \tag{14}$$

where $p(t, t_l)$ is the probability of an arrival over the next interval Δt.[7] The recursion tells us directly that no meeting should be held when $n = 0$. Thus, there is at most one meeting between two consecutive modifications.

At t_l, the time of the last modification, we can write the value function, given that a meeting will be held at some time t before the next modification arrival at time s (a random variable): $V(n, t_l | \textit{one meeting at time } t) = kn(t - t_l) + \tau_2 + EV(1, s)$. This is minimized for $t = t_l$. Thus, a meeting should be held, if at all, at t_l, immediately after the last modification. □

LEMMA 2. *A "critical value" policy is optimal: There is an $n(t)$ such that it is optimal to meet if and only if the number of modifications pending (occurred and not communicated) is larger than $n(t)$.*

PROOF. Based on Lemma 1, we can restate the dynamic program for t, the time of a modification arrival:

$$V(n, t) = \min \begin{cases} \tau_2 + V\left(1, t + \dfrac{1}{\mu_\alpha(t + T_1 - \lambda T_2)}\right) \quad \text{(meeting)}, \\ \dfrac{kn}{\mu_\alpha(t + T_1 - \lambda T_2)} + V\left(n + 1, t + \dfrac{1}{\mu_\alpha(t + T_1 - \lambda T_2)}\right) \\ \text{(no meeting)}. \end{cases} \tag{15}$$

Here, the expression $1/(\mu_\alpha(t + T_1 - \lambda T_2))$ is the expected time until the next modification occurs (the time argument of μ is shifted because modifications are generated according to upstream time). The cost is independent of n for the action "meeting", whereas it increases in n for the action "no meeting". Therefore, there must be a critical value $n(t)$. □

LEMMA 3. *If a critical value policy $n(t)$ is used, the expected cost rate at time t is*

$$\frac{\mu_\alpha(t)}{n(t)}\tau_2 + \frac{k(n(t) - 1)}{2}. \tag{16}$$

PROOF. Modifications arrive at the rate $\mu_\alpha(t)$. At a randomly chosen time t, one out of $n(t)$ modifications will trigger a meeting, which results in an expected communication cost rate of $\tau_2\mu_\alpha(t)/n(t)$. Arrivals between two meetings follow a homogeneous Poisson process and are thus uniformly distributed. Therefore, the expected number of modifications pending at a random time between two meetings is

[7] For Poisson arrivals, $p(t, t_l) = \mu(t)\Delta t$, and t_l can be dropped from the state description. t_l is updated exogenously with every arrival.

$(n(t) - 1)/2$ (since at most $n(t) - 1$ modifications ever wait). Thus, the expected "holding cost" rate is $k(n(t) - 1)/2$. □

Only the communication-related cost rate in (16) depends on the communication policy. It can be minimized for each point in time t separately and is convex, with the optimal critical value $n^*(t)$ in (7). The resulting expected communication frequency is calculated via $\mu_\alpha(t)/n(t)$. Plugging $n^*(t)$ into (16) and combining this with the unavoidable rework produces the result. □

Proof of Proposition 1

The first and second derivatives of (10) with respect to λ are

$$\frac{\partial ET}{\partial \lambda} = -T_2 + T_2\left[\sqrt{2k\mu_\alpha\tau_2\left(1 + e - 2e\lambda\,\frac{T_2}{T_1}\right)} - \frac{k}{2}\right] + T_2^2\lambda k\mu_\alpha\left(1 + e - \frac{T_2}{T_1}\,\lambda e\right), \qquad (17)$$

$$\frac{\partial^2 ET}{\partial \lambda^2} = -\frac{2e}{T_1}\frac{T_2^2}{2}\frac{\sqrt{2k\mu_\alpha\tau_2}}{\sqrt{1 + e - 2e\lambda\,\dfrac{T_2}{T_1}}} + k\mu_\alpha T_2^2\left[1 + e - 2e\lambda\,\frac{T_2}{T_1}\right]. \qquad (18)$$

The first term in (18) represents the communication delay costs. It is negative (concave in the objective function) for $e > 0$ and positive (convex) for $e < 0$. The second term stems from the convex unavoidable rework and is always positive.

For $e < 0$, the second derivative is positive for all feasible λ, and it decreases strictly with λ. Therefore, if $\partial^2 ET/\partial\lambda^2 > 0$ at the maximum value for λ, the second derivative must be positive over the whole feasible range of λ. At the other extreme, if $\partial^2 ET/\partial\lambda^2 < 0$ at $\lambda = 0$, then the second derivative must be negative over the whole feasible range of λ. In between, the objective function is convex-concave, because the second derivative is decreasing.

Therefore, setting (18) to zero at $\lambda = 0$ and $\lambda = \min\{1, T_1/T_2\}$ yields the values of $\underline{e}$ and $\bar{e}$:

$$\frac{\underline{e}}{\left(1 + \underline{e}\left[1 - 2\lambda_{\max}\,\dfrac{T_2}{T_1}\right]\right)^{3/2}} = T_1\sqrt{\frac{k\mu_\alpha}{2\tau_2}}\,;\ \frac{\bar{e}}{(1 + \bar{e})^{3/2}} = T_1\sqrt{\frac{k\mu_\alpha}{2\tau_2}}\,.$$

Note that when $T_2 > T_1$ (the downstream task is long) then $\lambda_{\max} = T_1/T_2$. If the downstream task is short, then $\lambda_{\max} = 1$. Thus, the value of $\underline{e}$ differs between the two cases. When the right-hand side of the above conditions is >1 (that is, $\lambda T_2\beta^* > 1$), then $e > \bar{e}$ is impossible, because $e/(1 + e)^{3/2} < 1$. □

Proof of Theorem 2

We consider k and μ_α first. As the cost rate in the integral of the objective function is positive and increasing in k, we have $\forall t \in [T_1 - \lambda T_2, T_1] : k(t - T_1 + \lambda T_2)\mu(t) + \sqrt{2k\tau_2\mu_\alpha(t)} - k/2 > 0$ and increasing with k. Thus, substituting $t = T_1 - \lambda T_2$ we have $\sqrt{2k\mu_\alpha\tau_2(1 + e - 2e\lambda(T_2/T_1))} > 0$ and increasing with k. Using this result, we see that (17) is increasing in k, and thus the mixed partial $\partial^2 ET/\partial\lambda\partial k$ is positive.

When the objective function is convex in λ (in the first region of e), then the optimal overlap is either λ_{FOC} (the solution to the FOC) or at a border of Λ. By the implicit function theorem

$$\partial\lambda_{\text{FOC}}/\partial k = -\left.\frac{\partial^2 ET}{\partial\lambda\partial k}\right|_{\lambda_{\text{FOC}}} \Big/ \left.\frac{\partial^2 ET}{\partial\lambda^2}\right|_{\lambda_{\text{FOC}}}.$$

This expression is negative, implying that the optimal overlap decreases with k (strictly if λ_{FOC} is optimal). By inspecting (17) we see that the same argument holds for μ_α.

We now consider the case where the objective function is concave. We have to compare the objective function (10) at the two possible solutions $\lambda^* = 0$ and $\lambda^* = \lambda_{\max}$. For $\lambda^* = 0$, the objective function is $ET = T_1 + T_2$, independent of e, k or μ_α. For $\lambda > 0$, however, ET strictly increases with $k\mu_\alpha$, as inspection of the objective function (10) shows. Therefore, the optimal solution can only drop from $\lambda_{\max}$ to 0 as $k\mu_\alpha$ increases, but never the other way round.

Finally, when the objective function is convex-concave, the candidates for optimal overlap are 0, $\lambda_{\max}$, or λ_{FOC}. Since the mixed partial of ET with respect to λ and k (or μ_α) is positive, the objective function increases more with k (or μ_α) at full overlap than at λ_{FOC}. Thus, again, the solution can only drop from full overlap to λ_{FOC} to zero overlap, but never the other way round. This concludes the comparative statics.

We now turn to the influence of e on optimal overlap. We need to evaluate the mixed partial (inspection of (10) shows that $\partial ET/\partial e > 0$):

$$\frac{\partial^2 ET}{\partial\lambda\partial e} = T_2\left[\lambda T_2 k\mu_\alpha\left(1 - \lambda\,\frac{T_2}{T_1}\right) + \sqrt{2k\mu_\alpha\tau_2}\,\frac{1 - 2\lambda\,\dfrac{T_2}{T_1}}{2\sqrt{1 + e\left(1 - 2\lambda\,\dfrac{T_2}{T_1}\right)}}\right]. \qquad (19)$$

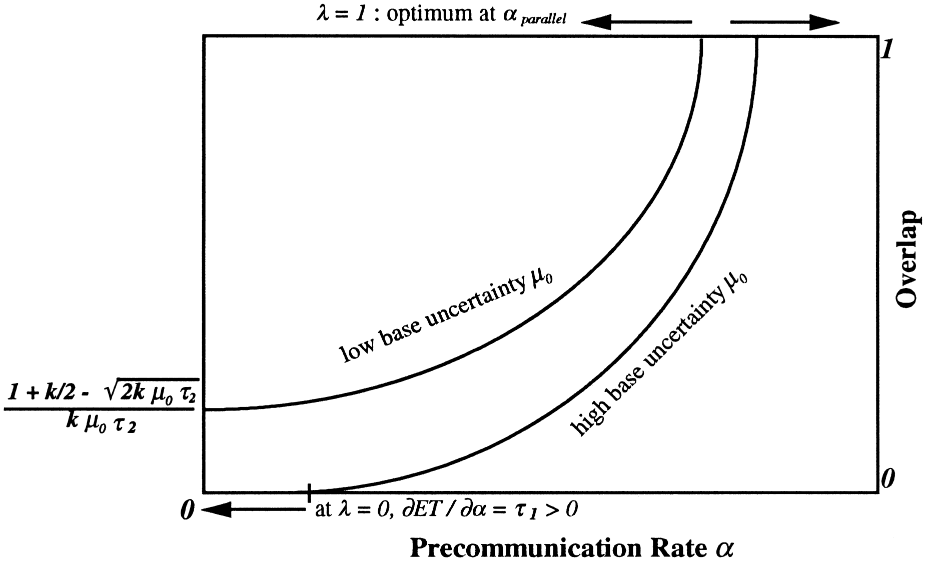
This expression is positive whenever $(1 - 2\lambda T_2/T_1) \geq 0$ (i.e., when overlap is less than 50 percent of $\lambda_{\max}$). Moreover, (19) strictly decreases in λ when $(1 - 2\lambda T_2/T_1) < 0$, so it has at most one zero in Λ. Therefore, there is at most one $\bar{\lambda}$ such that the mixed partial is positive to its left and negative to its right. The implicit function theorem used as above implies that λ_{FOC} shifts to the left as e increases if $\lambda_{\text{FOC}} < \bar{\lambda}$, and it shifts to the right if it lies to the right of $\bar{\lambda}$ (see Figure 2). Moreover, when the objective function is convex-concave, $ET(\lambda_{\max})$ may increase less with e than $ET(\lambda_{\text{FOC}})$, so the solution may jump, to full overlap, as uncertainty reduction becomes slower.

We have already observed that (19) strictly decreases at $\bar{\lambda}$. In addition,

$$\frac{\partial^3 ET}{\partial\lambda^2\partial e} = T_2 k\mu_\alpha\left(1 - 2\lambda\,\frac{T_2}{T_1}\right) + \sqrt{2k\mu_\alpha\tau_2}\,\frac{T_2}{T_1}\left[\frac{1}{\sqrt{1 + e\left(1 - 2\lambda\,\dfrac{T_2}{T_1}\right)}} + \frac{1 - 2\lambda\,\dfrac{T_2}{T_1}}{2\left(1 + e\left(1 - 2\lambda\,\dfrac{T_2}{T_1}\right)\right)}\right] > 0.$$

Therefore, by the implicit function theorem, $\partial\bar{\lambda}/\partial e < 0$.

Figure 3 **Saddle Curve $\lambda_{\text{FOC}}(\alpha)$**



When the objective function is fully concave (e is in region 3 of Proposition 1), then the same argument applies as for $k\mu_\alpha$: the optimum of (10) can only drop from full overlap to zero, but not jump from zero to full overlap. This concludes the proof. □

Proof of Proposition 2

The first derivatives of (11) with respect to the decision variables are:

$$\frac{\partial ET}{\partial \lambda} = T_2\left(-1 + k\mu_\alpha\lambda T_2 + \sqrt{2k\tau_2\mu_\alpha} - \frac{k}{2}\right);$$

$$\frac{\partial ET}{\partial \alpha} = \tau_1 - \frac{1}{2}\lambda T_2 B[k\lambda T_2\mu_\alpha + \sqrt{2k\tau_2\mu_\alpha}]. \tag{20}$$

The Hessian of Equation (11) becomes:

$$H(\lambda, \mu) = \begin{bmatrix} k\mu T_2^2 & -T_2B\left[k\lambda T_2\mu_\alpha + \sqrt{\frac{k\tau_2\mu_\alpha}{2}}\right] \\ -T_2B\left[k\lambda T_2\mu_\alpha + \sqrt{\frac{k\tau_2\mu_\alpha}{2}}\right] & \frac{\lambda T_2B^2}{2}\left[k\lambda T_2\mu_\alpha + \sqrt{\frac{k\tau_2\mu_\alpha}{2}}\right] \end{bmatrix}. \tag{21}$$

This matrix is convex in the directions of λ and α separately, but indefinite overall, since $H_{11} > 0$ and $\det(H) < 0$. In particular, the objective function (11) is concave in the direction α *given* λ_{FOC} (the solution of the FOC for λ):

$$\lambda_{\text{FOC}} = \frac{1 + k/2 - \sqrt{2k\mu_\alpha\tau_2}}{k\mu_\alpha T_2}. \tag{22}$$

The projected one-dimensional objective function in α is:

$$ET(\alpha) = T_1 + T_2 + \alpha\tau_1 - \frac{(1 + k/2 - \sqrt{2k\mu_\alpha\tau_2})^2}{2k\mu_\alpha}. \tag{23}$$

Figure 3 shows the two-dimensional shape of this one-dimensional subspace. λ_{FOC} is shown as a function of α. The shape of the curve is convex, which can be seen by taking the derivatives in (22). The maximum and minimum values of α for which the resulting overlap λ remains feasible are indicated where the curve hits the borders. Two cases are shown: The optimal overlap may be zero for small α, but if the maximum change rate μ_0 is very low, λ_{FOC} remains positive even for $\alpha = 0$ ($\mu_\alpha = \mu_0$). We restrict attention to the case where $T_2 \leq T_1$, so $\lambda_{\max} = 1$ (the other case is analogous).

We now show that the objective function is concave in the direction of the curve in Figure 3. We take the derivatives of Equation (23) with respect to α:

$$\frac{\partial ET(\alpha)}{\partial \alpha} = -\tau_1 + \frac{B(1 + k/2)}{2k\mu_\alpha}(1 + k/2 - \sqrt{2k\mu_\alpha\tau_2}); \tag{24}$$

$$\frac{\partial^2 ET(\alpha)}{\partial \alpha^2} = -\frac{B^2(1 + k/2)}{2k}\left[\sqrt{\frac{k\tau_2}{2\mu_\alpha}} + \frac{1 + k/2 - \sqrt{2k\mu_\alpha\tau_2}}{\mu_\alpha}\right] < 0. \tag{25}$$

Thus, the objective function is strictly concave in α for an interior λ.

Proof of Theorem 3

Since the objective function (11) is concave in the direction (α, $\lambda_{\text{FOC}}(\alpha)$), the optimal solution must lie on a border of the interior region. There are three possible border points, as shown in Figure 3. The first possibility, $\lambda_{\text{FOC}} = 0$, is not optimal: the first derivative (20), with $\lambda = 0$, is negative. Thus, the optimal solution "slides" along the border $\lambda = 0$ to zero precommunication, or $\alpha = 0$. This corresponds to the sequential solution in Theorem 3.

The second possibility is relevant if μ_0 is so small that the curve in Figure 3 hits the right-hand border before λ goes down to zero. This border point is optimal, because λ_{FOC} is the solution of the first order condition in the convex problem given α. Precommunication α is still zero. Thus, a ''minimum overlap'' is possible in the sequential solution. The third possibility is the border point at the top of Figure 3, where full overlap $\lambda = 1$ prevents α from being raised further. Again, this border point is not optimal. When inserting $\lambda = 1$ in the first derivative, the FOC requires

$$\frac{\tau_1}{T_2 B \mu_\alpha} = \frac{1}{2} k T_2 + \sqrt{\frac{k\tau_2}{2\mu_\alpha}},$$

yielding a quadratic equation in μ_α. The solution corresponds to α_{paral}. □

Proposition 3 and Its Proof

Let $EC(t)$ be the expected cost rate due to communication delay only, as defined in (16). Let the critical value of the communication policy, $n(t)$, be a continuous function of $\mu_\alpha(t)$. Proposition 3 shows that this cost rate, and thus Lemma 3, hold approximately when $\mu_\alpha(t)$ changes little between two consecutive meetings.

PROPOSITION 3. *If modifications follow a nonstationary Poisson process of rate $\mu_\alpha(t)$, then as $|\mu_\alpha(t) - \mu_\alpha(s)| \to 0\ \forall t$, s in the same intermeeting interval,*

$$\left| EC(t) - \frac{\mu_\alpha(t)}{n(t)} \tau_2 - \frac{k(n(t) - 1)}{2} \right| \to 0.$$

PROOF. Modifications arrive with the rate $\mu_\alpha(t)$. At a randomly chosen modification, at, say, time t, one out of $n(t)$ will trigger a meeting and its associated delay of τ_2. This yields an expected communication cost rate of $\tau_2 \mu_\alpha(t)/n(t)$, establishing the first summand.

The ''holding cost'' component remains to be established. Pick a time s randomly. Let the time of the last meeting be t_l and the time of the following meeting t_f. If s is picked randomly, then $E[s$ given $t_l, t_f] = (t_l + t_f)/2$. The number of modifications pending at time s must be $\in \{0, 1, \ldots, n(t_f) - 1\}$, because after the $n(t_f)$th arrival, a new meeting is held immediately. We seek to calculate EI, the expected number of arrivals by time s given that there are $n(t_f)$ arrivals at the time of the next meeting. By transforming the nonhomogeneous Poisson process into a homogeneous one (Taylor and Karlin 1984, p. 177), we can write the conditional expectation of arrivals by time s, given t_f, as

$$E(I(s) \text{ given } n(t_f)) = (n(t_f) - 1) \frac{\int_{t_l}^{s} \mu_\alpha(u) du}{\int_{t_l}^{t_f} \mu_\alpha(u) du}.$$

Now, using the fact that $\mu_\alpha(u) = a + bu$ has a linear form, we can evaluate the integrals as: $E(I(s)$ given

$$n(t_f)) = E_s\left[\frac{a(s - t_l) + \frac{1}{2} b(2st_l - t_l^2)}{a(t_f - t_l) + \frac{1}{2} b(2t_f t_l - t_l^2)} (n(t_f) - 1) \right].$$

Since s is independent of t_f, t_l, the expectation can be pulled into the brackets. As μ_α converges to a constant, $b \to 0$ and $n(t_f) \to n(s)$ by assumption. Thus, $\lim_{b \to 0}(EI) = (n(s) - 1)/2$ for all t_f, t_l. □

References

Adler, P. S. 1995. Interdepartmental interdependence and coordination: the case of the design/manufacturing interface. *Organization Sci.* **6** 147–167.

Blackburn, J. D. (Ed.) 1991. *Time Based Competition: The Next Battleground in American Manufacturing*. Business One Irwin, Homewood, IL.

——, G. Hoedemaker, L. N. Van Wassenhove 1996. Concurrent software engineering: prospects and pitfalls. *IEEE Trans. on Engineering Management*. **43** 179–188.

Chakravarty, A. K. 1995. Overlapping design and build cycles in product development, Working Paper, Tulane University.

Clark, K. B., T. Fujimoto 1991. *Product Development Performance: Strategy, Organization and Management in the World Auto Industry*. Harvard Business School Press, Cambridge, MA.

Cordero, R. 1991. Managing for speed to avoid product obsolescence: a survey of techniques. *J. Product Innovation Management*. **8** 289–294.

Eastman, R. M. 1980. Engineering information release prior to final design freeze,'' *IEEE Trans. on Engineering Management*. **27** 37–41.

Eisenhardt, K. M., B. N. Tabrizi 1995. Accelerating adaptive processes: product innovation in the global computer industry. *Admin. Sci. Quarterly*. **40** 84–110.

Griffin, A. 1996. The impact of engineering design tools on new product development efficiency and effectiveness. *Proc. 3rd EIASM International Product Development Conference*. Fontainebleau, France, 363–380.

Ha, A. Y., E. L. Porteus. 1995. Optimal timing of reviews in concurrent design for manufacturability. *Management Sci.* **41** 1431–1447.

Heyman, D. P., M. J. Sobel 1984. *Stochastic models in operations research volume II*. McGraw Hill, New York.

Hoedemaker, G. M., J. D. Blackburn, L. N. Van Wassenhove 1995. Limits to concurrency. Working Paper, INSEAD, France.

Iansiti, M. 1995. Technology integration: managing technological evolution in a complex environment. *Res. Policy* **24** 521–542.

Imai, K., I. Nonaka, H. Takeuchi. 1985. 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, Cambridge, MA.

Krishnan, V. 1996. Managing the simultaneous execution of coupled phases in concurrent product development. *IEEE Trans. on Engineering Management*. **43** 210–217.

Krishnan, V., S. D. Eppinger, D. E. Whitney. 1997. A model-based framework to overlap product development activities. *Management Science* **43** 437–451.

Ramamoorthy, C. V., F. B. Bastani 1982. Software reliability—status and perspectives. *IEEE Trans. on Software Engineering*. **8** 359–371.

Rosenblatt, M., H. Lee 1986. Economic production cycles with imperfect production processes. *IIE Trans*. **18** 48–55.

Sabbagh, K. 1996. *Twenty-First Century Jet*. Scribner, New York.

Takeuchi, H., I. Nonaka. 1986. The new product development game. *Harvard Business Rev*. January–February, 137–146.

Taylor, H. M., S. Karlin 1984. *An introduction to stochastic modeling*. Academic Press, New York.

Terwiesch, C., C. H. Loch, M. Niederkofler 1996. Managing tradeoffs in concurrent engineering. *Proc. 3rd EIASM International Product Development Conference*. 693–706.

Terwiesch, C., C. H. Loch 1998. Managing the process of engineering change orders: the case of the climate control system in automobile development. *J. Product Innovation Management* **15**, forthcoming.

Tushman, M. L. 1978. Technical communication in research and development laboratories: the impact of task characteristics. *Acad. Management J*. **21** 624–645.

Van de Ven, A. H., A. L. Delbecq 1974. A task contingent model of work unit structure. *Admin. Sci. Quarterly*. **19** 183–197.

Wirth, N. 1975. *Algorithmen und Datenstrukturen*. Teubner, Stuttgart, Germany.

Wheelwright, S. C., K. B. Clark 1992. *Revolutionizing Product Development*. The Free Press, New York.

Accepted by Hau Lee; received April 3, 1996. *This paper was with the authors* 4 *months for* 3 *revisions*.