A FRAMEWORK FOR EXCHANGING PRELIMINARY INFORMATION IN CONCURRENT DEVELOPMENT PROCESSES

by

C. TERWIESCH*
C. H. LOCH**
and
A. DE MEYER†

98/53/TM
(Revised Version of 97/117/TM)

* The Wharton School, Philadelphia, USA.

** Associate Professor of Operations Management at INSEAD, Boulevard de Constance, 77305 Fontainebleau Cedex, France.

† Akzo Nobel Fellow in Strategic Management, Professor of Technology Management and Asian Business at INSEAD, Boulevard de Constance, 77305 Fontainebleau Cedex, France.

Printed at INSEAD, Fontainebleau, France.

A FRAMEWORK FOR EXCHANGING PRELIMINARY INFORMATION IN CONCURRENT DEVELOPMENT PROCESSES

Christian Terwiesch

The Wharton School, Philadelphia, USA

Christoph H. Loch

INSEAD, Fontainebleau, France

Arnoud De Meyer

INSEAD, Fontainebleau, France

The usage of preliminary information is frequently referred to as an important requirement for a successful application of concurrent development processes (concurrent engineering). Interestingly, although concurrent engineering requires exchanging preliminary information, little is known about the concept, both from a managerial and a theoretical perspective. Building on fieldwork in the automotive industry, we present a framework which operationalizes preliminary information in the form of two dimensions, information precision and information stability. Information precision refers to the accuracy of the information exchanged. Information stability defines with what likelihood a piece of information is going to be changed later in the process. Our framework is grounded on detailed longitudinal data collection from five engineering decisions that we traced as on-site observers in a vehicle development project. We discuss the managerial decisions that a development organization faces in the exchange of preliminary information, and we explain the trade-offs under what circumstances which of the two dimensions should be emphasized.

Introduction

Managing and understanding the complex information exchanges in today's highly concurrent development processes is one of the biggest challenges for practitioners and scholars in the field of new product development. One key characteristic of information exchange in concurrent development processes is that it occurs at a point when the information-supplying party is still facing substantial uncertainty about the final outcomes of its problem-solving activity. Although the early release of such preliminary information creates uncertainty and ambiguity for the information receiver, it is an important enabler for high levels of process parallelity. This brings the management of preliminary information to the heart of any concurrent development process.

The importance of releasing preliminary information early in the development process has been frequently reported in academic literature. For example, Clark and Fujimoto report: "Effective management of overlapped engineering problem-solving results partly from the *early release of preliminary information from upstream and partly from downstream's efficient use of those cues to make a flying start* (Clark and Fujimoto 1991, emphasis added)." The idea of giving downstream an early start by using preliminary information also underlies other frameworks of concurrent development processes, including the ones by Ward *et al.* (1995), Krishnan *et al.* (1997), and Loch and Terwiesch (1998).

Interestingly, although most frameworks of concurrent development processes build on the concept of preliminary information release, little is known about the concept, both from a managerial and a theoretical perspective. Previous research neither supplies us with a rigorous definition of preliminary information, nor does it illuminate the underlying trade-offs inherent in its application. This lack of scientific and managerial understanding of preliminary information is described by Clark and Fujimoto (1991), who remark: "There is something of an art involved in managing the preliminary information."

The objective of the present article is twofold: First, building on in-depth fieldwork in the automotive industry, we present a framework that operationalizes the concept of preliminary information in the form of two dimensions, information precision and information stability. Second, we discuss the managerial decisions that a development organization faces in exchanging preliminary information. We illuminate underlying trade-offs that determine under what circumstances which of the two dimensions should be emphasized.

The article is organized as follows: we first review some relevant literature, and then present our research questions followed by a brief discussion of the methodology we have chosen to address them. The remaining sections develop an operational definition of preliminary information and an analysis of the managerial trade-offs involved.

Previous Research

Research on problems related to preliminary information goes back to Eastman (1980), who discusses the benefits and dangers of committing engineering resources to specifications that are not yet finalized. In their study of the world automotive industry, Clark and Fujimoto (1991) were the first to clinically research the phenomenon in the context of stamping die development. Using an information processing framework (see, e.g., Galbraith 1973), they identified intensive communication as a key driver of development performance. Instead of batching the information created by one activity and then handing over this information to the next activity, the authors find that it is more effective to release preliminary information early and to let downstream start, using this preliminary information.

The concept of preliminary information was formalized by Krishnan *et al.* (1997). The authors consider a development task creating information that can be summarized in one specific parameter, such as the size of a dashboard or the depth of a door handle. This parameter represents preliminary information if it is known only up to an interval. The

interval narrows over time, and the information becomes final when a point value is settled.

Ward *et al.* (1995) describe how Toyota uses the concept of "set-based" concurrent engineering, which is closely related to preliminary information. Set-based refers to "a deliberate effort to define, communicate about, and explore *sets* of possible solutions rather than modifying point solutions." Possible (subsystem) solutions are then explored in parallel, with multiple redundant prototypes, delaying the freezing of key design decisions. The sets of solutions are gradually narrowed over time, and a "rigorous effort is made to avoid changes that expand rather than contract, the space of possible designs." Once a single solution is established for any part of the design, it does not change unless it is "absolutely necessary". The authors contrast this approach with the US practice of iterating between single solutions in the design space and state that "truly concurrent design requires a change from this point-based paradigm to a set-based approach (Liker *et al.* 1996)".

Loch and Terwiesch (1998) examine the trade-offs of overlapping two activities. They, however, conceptualize preliminary information as modifications of point-based solutions in the form of engineering changes. They show that an overlap of upstream and downstream activities with an associated exchange of preliminary information saves time, but also causes rework for the downstream activity. Thus, point-based preliminary information creates a trade-off between gains from parallel task execution and rework in form of engineering changes. The later the information is finalized upstream, the more pronounced this trade-off becomes. These authors have empirically tested this result based on electronics industry data, and find that development projects indeed gain more time from overlap if the preliminary information can be finalized early (Terwiesch *et al.* 1996).

Research Objectives

Consider a very simple example of a development process, where two activities are overlapped to minimize development lead-time. The overlap allows the information absorbing downstream operation (e.g., stamping die development) to start before the information supplying upstream activity (e.g., product design) is completed.

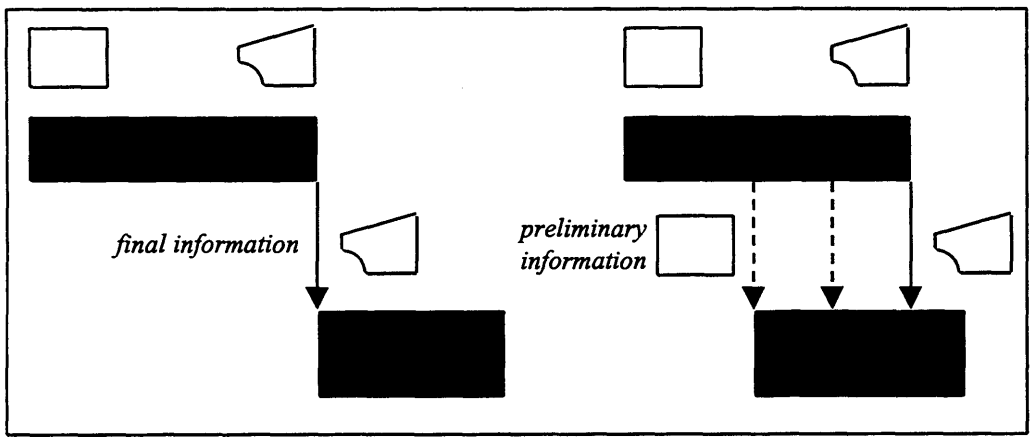

Figure 1: Overlapping Activities Requires the Use of Preliminary Information

Although the overlap creates a direct time gain through parallelity, it is not without drawbacks. In a fully sequential process (Figure 1, left), no information is released to downstream until upstream has gained a relatively high level of knowledge about its task. When downstream finally starts, it can rely on finalized information from upstream. An overlapped process (Figure 1, right) forces downstream to give up this quality of information and to go ahead, based on preliminary information. It is in the nature of such preliminary information to be based on a low to medium level of upstream knowledge, which creates high risks of future changes. Change risks are especially great if the outcome of the upstream activity is so uncertain that an accurate prediction of the final outcome is difficult, if not impossible. In this case, overlapping activities creates downstream uncertainty that does not exist in the sequential process.

Note that by labeling one activity upstream and the other downstream, we are assuming that one activity supplies information to the other during a given information exchange. This does not imply that information flows are unidirectional over the length of the activity, i.e. that results from upstream are "thrown over the wall" to downstream. The direction of the information flow may be different at each exchange, which is consistent with bi-directional information flows, or interdependent activities.

There are two ways of dealing with the above explained uncertainty. One possibility consists of upstream committing to a specification early, eliminating the uncertainty while sacrificing opportunities to fine-tune the design. However, enforcing an early upstream solution freeze is often technically impossible (e.g., the solution does not work) or unacceptable (e.g., the market requirements may change). The second possibility is that downstream accepts the uncertainty and starts with information in the form of intervals (set-based) or adapts to changes in information created by upstream (point-based).

The importance of dealing with preliminary information is widely recognized. However, despite numerous studies encouraging practitioners to "use" the concept, research to date has not provided a clear operational definition. The complexity and the vagueness of the preliminary information concept is summarized by Clark and Fujimoto (1991), who report: "There is something of an art involved, for example, in leaving a slightly thicker cutting margin in an area of the die which is more likely to be affected by design changes, or in recognizing that the location of holes in a door panel changes more frequently than the anticipated shape does."

The lack of an operational definition in existing literature, together with the practical need to exchange preliminary information in today's highly parallel engineering processes, prompts our first research question:

How can we conceptualize preliminary information? In what formats can one communicate the uncertainty, which is inherent in preliminary information?

The second objective of this study is to improve our understanding of how to manage the exchange of preliminary information. Most previous studies have emphasized the importance of frequent cross-functional communication (e.g., Hauptman and Hirji 1996, Wheelright and Clark 1992). Although this line of research has substantially improved current understanding of development processes, it is not sufficient to address the managerial challenges of dealing with preliminary information for three reasons. First, most companies have implemented many aspects of cross-functional integration over the last decade, so it is already included in the known state-of-the-art. Second, technology management literature has focused on communication *frequency* as opposed to the *format* of an information exchange. This is especially true of preliminary information, where the information quality has to be communicated in addition to the message itself.

Third, the costs/disadvantages of information exchange are not well understood. Is it always beneficial to exchange (preliminary) information, or is there a trade-off involved? The presence of trade-offs in using preliminary information was observed by Clark and Fujimoto (1991) in their discussion of preliminary information in die development. The authors report: " ... the key is to make subtle trade-offs between the cost and the benefit of starting die-making early." However, their study does not explore these trade-offs in more detail.

Some of the recent operations management work on product development (e.g., Ha and Porteus 1995, Loch and Terwiesch 1998) has addressed these trade-offs more formally. In this stream of research, communication is viewed as a fixed set-up penalty describing the meeting time, which has to be balanced with time gains resulting from communication. The limitation of this approach is, as mentioned above, that it focuses on frequency only, excluding any other issues of information exchange other than meeting times.

The observation that the trade-offs in releasing preliminary information have not been previously addressed in product development literature prompts our second research question:

How does preliminary information influence the work of downstream activities? What are the trade-offs resulting from the usage of preliminary information?

Methodology

Our study uses a multiple case design logic, investigating a total of ten components of a vehicle development project at a global automotive manufacturer. As our objective was to gain a detailed understanding of problem-solving and information exchange, we focused on one particular subsystem of the overall car, the climate control system (CCS). All ten components investigated belong to the CCS module. It contains all components and development activities related to the climate environment for the passengers, including air ventilation, air cleaning, warm-up, and cool-down. The CCS was chosen, after extensive discussion with practitioners and industry experts, because of its enormous need for coordination and information exchange.

Referring to the air intake of the car (filter box), one of our interviewees explained: "Here at the filter box you find all the problems we have in the development of new vehicles: coordination with other components (e.g., fire-wall, engine) and information release to tooling." Together with the dashboard, the CCS is the subsystem of the car with the most interfaces to other activities (see Figure 2 for the CCS architecture). This, of course, makes information management absolutely crucial.

Data collection was longitudinal, with the first author staying on site for about four months on a full-time basis, from October 1996 to February 1997. The longitudinal character of our research enabled us to monitor the evolution of information and to gain

access to data sources that are usually closed to outsiders. Data were collected from multiple sources, the most important being:

- About 50 semi-structured interviews with engineers and management, including upstream (information-sending) and downstream (information-receiving) parties for every component.
- (Passive) participation in all relevant meetings on the ten components.
- Data analysis from internal quality and cost control systems.

At the detailed level of our analysis, we followed the evolution of specific design decisions and their corresponding engineering changes over time. Following the paradigm of grounded research (Miles and Huberman 1984, Eisenhardt 1989), our analysis builds on detailed field notes, including the interview notes, transcripts of engineering meetings, and company documents as mentioned above. These data were then compiled into detailed case studies for each engineering decision we followed. This process was more iterative than sequential as the emerging cases were frequently updated, based on follow-up discussions with respondents and other participating engineers. Once the cases were completed, we contrasted different paths of information evolution as described below.

In the remainder of this article, we provide a detailed description of five of these ten design decisions to illustrate our operationalization of preliminary information. The five cases are representative of the range of issues encountered; restricting the article to five cases allows us to describe thoroughly for each the information exchanged, the situation of the information-sending party, and the consequences for the information-receiving party. This enables us to at least partially share with the reader some of the richness and complexity of CCS development, and to show how our framework of preliminary information is grounded on the collected data.

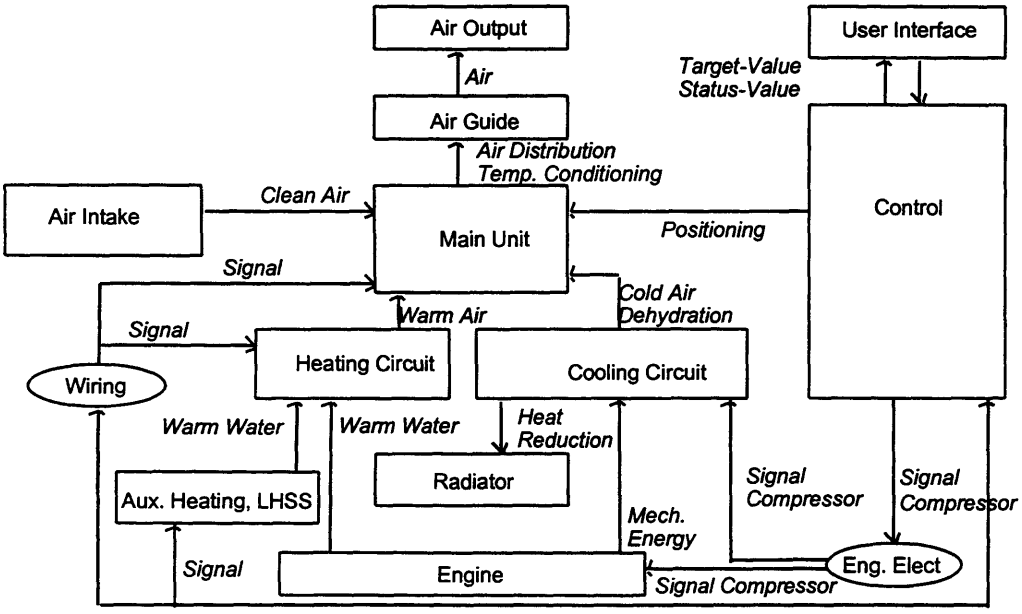

Figure 2: CCS Architecture

Five Cases of Information Evolution

Consider the situation where one development activity (upstream) has to forward information to another activity (downstream) before problem-solving has been finalized upstream, similar to the one described in Figure 1. The unit of analysis is thus an engineering problem to be resolved in the upstream activity and the associated information flows to the downstream activity. With respect to this information, the communication dyad faces residual uncertainty about the final information. How can this uncertainty be communicated to downstream? What are the consequences of the uncertainty? The following five cases from our study illustrate this situation. We first briefly describe the technical characteristics of the engineering problem for each one of the five cases. We then explain how the uncertainty was resolved upstream, and demonstrate the implications of this uncertainty resolution strategy for the downstream activity.

Technical Characteristics of the Engineering Problem

- A large part of the thermal energy used for heating the passenger cabin is supplied by the engine, the remainder coming from auxiliary heaters. Thus, crucial information in the development of the CCS is the amount of warm water supplied by the engine. This information however, is not fully available before engine development is finalized. Engine development supplies CCS development with preliminary information about the volume of warm water per second. Based on a rough estimate, the CCS development group decides on its approach to auxiliary heating.
- The geometry of the air intake/filtering system is highly dependent on the engine geometry. Packaging space between engine, fire-wall and dashboard is extremely scarce, so geometrical dependencies have to be carefully managed. This is particularly relevant for air flows because the amount of fresh air entering the cabin is a function of this geometry. For the development of the air intake/filter box, the geometric details *within* the engine are of little importance. However, the *hull* of the engine heavily influences the design solution, which aims to provide sufficient air throughput without wasting a cubic centimeter of space.
- An innovative feature of the CCS is a latent heat storage system (LHSS). Once the engine runs at its operating temperature, the LHSS can chemically store its heat for up to several days. This energy can then be used for a rapid warming-up of the engine and the passenger cabin the next time the car is started. At the time of our study, the LHSS was a highly innovative component with unproven demand, and management was not sure whether to include it or not. Given its substantial size, the packaging of the LHSS into the vehicle is a rather difficult task: having the LHSS in or not changes the complete layout of the engine compartment.
- Cars in general, and the CCS in particular, are becoming more and more sophisticated in their functionality. The CCS contains a substantial amount of software, which controls dozens of precision motors, fans and pumps to create a comfortable cabin environment for the passengers. Software development depends on testing results from CCS prototyping in order to come up with a fine-tuned control system.

- The fire-wall of the car is a solid metal sheet that separates the passenger cabin from the engine compartment. It is a key component in providing body stiffness, but at the same time it must contain a large number of holes to allow air and water from the engine compartment to enter the cabin. Thus, the fire-wall looks like a "Swiss cheese". The positions of the holes are important pieces of information as they provide the interface between fire-wall development/stiffness, climate control and stamping die development for the car body.

Case	Information Piece	Sender	Receiver
Auxiliary Heating Concept	*How much warm water is supplied by the engine?*	*Engine Development*	*Development of CCS in determining an auxiliary heating concept*
Packaging Air Intake/ Engine	*Where to find space in the engine compartment for air intake and filter box?*	*Engine Development, Packaging Team*	*Development of CCS in determining air flow concepts*
LHSS	*Will there be a LHSS for the car?*	*Management, based on market experience for a predecessor*	*Development of CCS and all package-critical activities*
CCS Software	*What are the software specifications for the control unit?*	*Development of CCS and testing teams*	*Development of control unit*
Fire-wall Holes	*Where to put the holes in the fire-wall for air and water throughput?*	*Packaging team, Development of CCS*	*Fire-wall development, die development*

Table 1: Summary of the five cases

Nature of the Information Passed from Upstream to Downstream

The five cases share the same problem: a downstream development activity has to start, using preliminary, or uncertain, information from an upstream activity. We now describe for each of the five cases in what format the preliminary information was passed on, how this uncertainty was resolved over time, and what format the final information took.

- The key information needed for an auxiliary heating concept is the amount of water supplied by the engine. This information can be captured in one single number. While the first information transfer from the engine group occurred already in the form of a number, there was a tacit agreement that this number should be understood more as an interval. This corresponds to the Krishnan *et al.* (1997) framework and to Ward *et al.*'s (1995) set-based strategy, with the uncertainty being resolved in the form of an interval narrowed down over a series of engine tests.
- The information required for the package coordination between engine and air intake is more complex. The interface is defined not by a single number, but through a complex three-dimensional geometric structure that evolves with the development of the engine. However, not all of this information is relevant for constructing the air intake. This allowed the engine development team to provide the geometry of the hull around the engine as preliminary information to the CCS developers.
- For the LHSS case, the required information is extremely concise: One bit suffices to capture the "yes/no" answer. This information becomes available only after market input from a predecessor project, about one year before the launch. As one year is too short to redesign the whole packaging of such a large component, the information was explicitly transferred as "yes *and* no," which is mathematically similar to the hull of the engine or the confidence interval for warm water. Therefore, the development team developed two engine compartments in parallel, one with and one without the LHSS.
- The CCS control software needs detailed data from the CCS prototyping as to how changes in certain control variables (e.g., rotation speed of a fan) translate into measures perceived by the passengers (e.g., fresh air, noise from fan). Testing and SW development proceed in parallel, relying on a series of SW prototypes, each of which was developed, based on the most recent testing results.
- Finally, the relevant information concerning the fire-wall holes can be described by about 50-100 numbers describing their geometric positions and sizes. In contrast to the interval approach of cases 1 to 3, we observed that these data in most cases were given with high precision initially and then iteratively modified. An additional way of exchanging preliminary information on holes was to work with "locking zones". If,

for example, for reasons regarding stiffness, it was undesirable to have holes in some areas of the fire-wall, these zones could be "locked", thus prohibiting definition of holes in the CAD system.

Case	Uncertainty Resolved Through:	Form of Preliminary Information	Form of Final Information
Auxiliary Heating Concept	*Engine testing*	*Rough guess with tacit understanding of a confidence interval*	*One number*
Packaging Air Intake/ Engine	*Engine development*	*CAD models of the hull*	*Detailed geometry (almost impossible to capture in numbers)*
LHSS	*Market observation*	*Explicit inclusion of both alternatives*	*Yes/No*
CCS Software	*Evolutionary prototyping; adjusting parameters based on test results*	*Prototype*	*One program and a set of parameters*
Fire-wall Holes	*Iterative modification*	*Intermediate solutions*	*50-100 numbers describing position and diameters*

Table 2: Upstream uncertainty resolution

Downstream Adjustments to Changes in Preliminary Information

The upstream uncertainty resolution and the release of preliminary information have consequences for the downstream activity. On the one hand, new information available upstream may force adjustments of downstream work, which can be costly and time-consuming. On the other hand, the downstream activity might become starved, if at some point too little upstream information is available. In the five cases, downstream adjustments occurred as follows.

- The auxiliary heating concept does not require highly accurate information. A flow deviation of less than 5% creates no major problem. However, larger deviations require a new auxiliary heating concept, which is very costly and time-consuming to develop. In addition, downstream adjustment becomes more difficult if modifications occur later. Even minor modifications, if they occur late, require changes in many

fine-tuning parameters (e.g., in the control software). Late and major changes may even delay the launch of the overall project.

- The only information that matters for air intake development is the hull of the engine. As long as information changes do not affect this interface, downstream is insensitive to the changes. Hull modifications become more costly over time because tools for the air intake have very long lead-times (special plastic tools) and are difficult to change.
- The information on the LHSS contains only two bits, but it affects the packaging of the entire engine compartment. Thus, any deviation (e.g., saying "no" first and then changing to "yes") has major consequences for all development activities involved. A late change may delay the launch of the car. The consequence of saying "yes and no" is that the development had to prepare two complete packaging concepts, one with and one without LHSS.
- CCS software has a modular product architecture: most parts of the system can be developed independently from the actual test results. The parts of the system that do depend on the test results are isolated in a set of parameters that can be changed without major effort, even late in the process, and an adjustment to customer needs is even possible during after-sales service. This makes software development almost independent from upstream.
- Downstream sensitivity for the fire-wall varies widely depending on the positions of the holes to be changed. Similar to the filter box tools, the stamping dies for the fire-wall are expensive to change late in the project. Repositioning holes becomes much more costly once tools have been developed: downstream's flexibility to adjust to new information thus decreases over time.

Case	How flexible is downstream with respect to upstream deviations?	Does downstream flexibility change over time?	Can downstream work based on preliminary information?
Auxiliary Heating Concept	*A bit more or less does not matter*	*Yes, long lead time for heating concepts*	*Within certain ranges, downstream can start*
Packaging Air Intake/ Engine	*All that matters is the hull; deviations from that can be expensive*	*Once the filter box SMC tools are started, adjustments are very difficult*	*Yes, but at the expense of high rework costs if the space allocation changes*
LHSS	*Packaging of the entire engine compartment depends on this information*	*Sharp increases of adjustment costs over time*	*No. Explicit need to do both*
CCS Software	*Modular, program remains stable, only parameter adjustments*	*Very flexible, SW adjustments possible even in after sales service*	*Perfect Modularity. SW can be developed and all adjustments done by parameters*
Fire-wall Holes	*Varies, depends on hole position*	*Sharp increases of adjustment costs over time (stamping dies)*	*Yes, but at the expense of high rework costs*

Table 3: Downstream implications of changes in preliminary information

A Framework For Preliminary Information Exchange

As a first step toward our framework for preliminary information exchange, we now use the previously introduced cases to derive four constructs. We first define downstream dependence via the concepts of rework and starvation. We then conceptualize two types of preliminary information, information precision and information stability.

Downstream Dependence: Rework and Starvation / Duplication

Table 3 suggests two types of couplings between the information-supplying and the information-absorbing activities. First, *rework* occurs if downstream commits resources based on upstream information that later turns out to be wrong. There are two sub-dimensions in this coupling. (a) *Sensitivity* describes the costs/duration of rework as a function of the size of a change between the new information and the previously released information (Krishnan *et al.* 1997). For example, in the auxiliary heating system case, a small change in water supply creates little rework, whereas larger changes require a completely new auxiliary heating concept. (b) *Time dependent impact* represents the change in sensitivity over downstream work progress. Almost universally, it is more expensive to make changes late, after a lot of progress, than to make them early (e.g., Terwiesch and Loch 1998). In the fire-wall case, modifications become very time-consuming and costly after volume dies have been cut. In contrast, the impact of changing specifications in the CCS software case remains small, even after market introduction. This corresponds to the impact function in Loch and Terwiesch (1998).

Second, *starvation* occurs if the downstream activity runs out of work because there is too little information available to start or to continue the task. Starvation is similar to the concept of work authorization in Ha & Porteus (1995). The first cost of starvation is a time loss: Valuable project time passes with the downstream activity being idle, waiting for additional information.

Sometimes, starvation can be avoided by *duplication* or by working with an "educated guess". In the LHSS case, starvation was avoided by developing two alternative packaging concepts in parallel. In the warm water case, downstream was able to go ahead, working with an expected outcome of the upstream process. If starvation is avoided by making an educated guess, there is an increased likelihood of rework, especially if there are many possible solutions. If starvation is avoided by pursuing multiple alternatives in parallel, a second cost results from starvation, namely that of redundancy. In the LHSS case, starvation was avoided by developing two alternative

packaging concepts in parallel, which was very costly in terms of engineering resources and prototyping hardware.

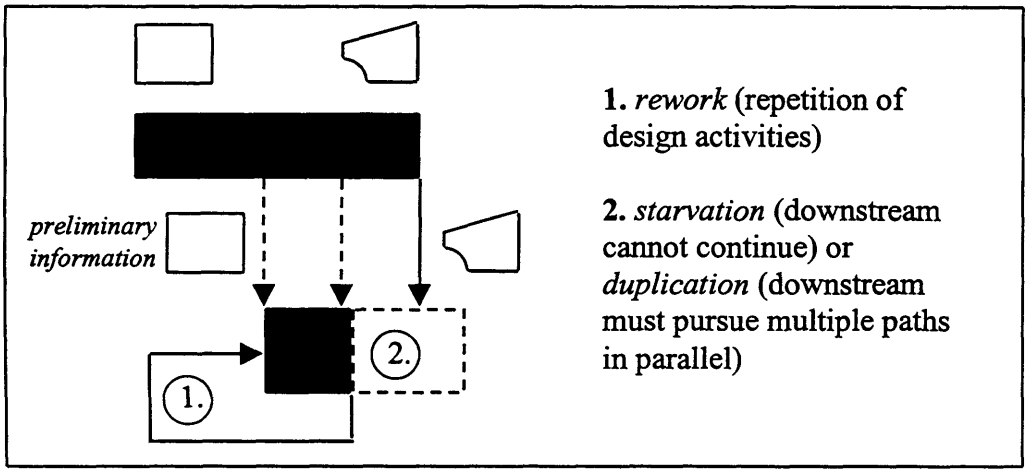

Figure 3: Dimensions of Downstream Dependence

Two Dimensions of Preliminary Information

Comparing the five cases, we see substantial differences in how preliminary information was managed. In the software and fire-wall cases, the project teams relied on intermediate solutions that were communicated in the same format as the final values. In the other cases, uncertainty was explicitly communicated, in the form of an interval (auxiliary heating concept) or as the set of all possible outcomes (LHSS).

Based on our observations in the cases, we identify two dimensions of preliminary information. *Information precision* defines the accuracy of a piece of information, that is, whether the information has the same format as a final value. For example, the information "hole at x = 123mm, y = 34mm, diameter = 5mm" is precise, whereas the information "between 19 and 21 l/h" is not. Thus, conceptually, a measure of precision can be derived by comparing the range of outcomes that is communicated in the information exchange with the range of all possible outcomes. Providing an accurate technical measure of precision is, however, beyond the scope of this article.

The other dimension of preliminary information is *information stability*. It defines with what likelihood a piece of information turns out to be wrong and thus will be changed later in the process. For example, the information for the LHSS is fully stable ("yes" and "no" provide all possible outcomes), while the fire-wall hole positions are almost certainly subject to change.

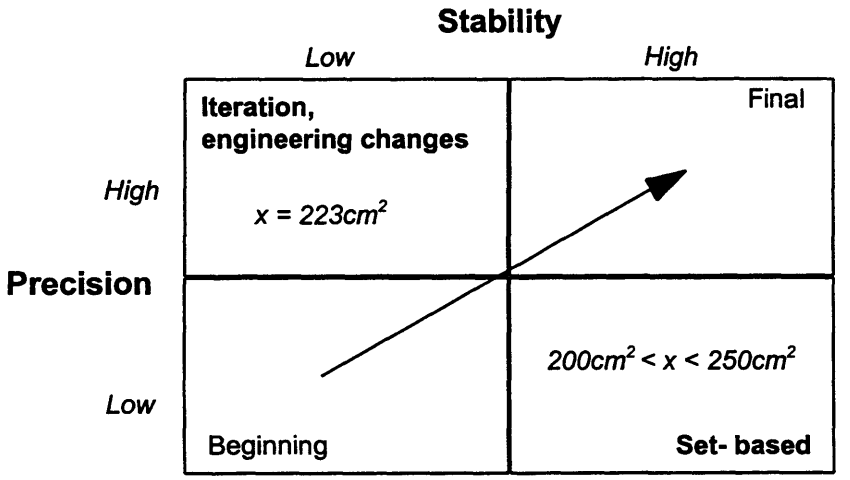

Figure 4: A Framework of Preliminary Information

Figure 4 allows a classification of information associated with an engineering problem (our unit of analysis) along two dimensions. As the problem-solving status of a design decision evolves, e.g., on a component, a sequence of information exchanges typically occurs. Initially, little information on the resolution of the design decision is available, and information is neither stable nor precise. As problem-solving on the design decision progresses, information is repeatedly communicated with changing levels of precision and stability. At the end of the problem-solving process, the design is in place and functional both upstream and downstream. Now, information is both stable and precise.

At each time of an information exchange, a decision must be made about the degree of precision and the degree of stability. Taking as fixed the level of knowledge about the problem (the design progress status), there are two extreme strategies. The lower right-hand corner in Figure 4 corresponds to communicating the information in a way that is very stable (e.g., it will change with a probability of less than 1%). The advantage of this strategy is that downstream can commit resources based on the information without

running the risk of rework. The disadvantage is that, given the low precision, downstream may not be able to continue at all (starvation because of lack of precision) or must perform duplication (redundant work), as in the LHSS case. The upper left-hand corner in Figure 4 corresponds to communicating the information in a way that is very precise. The advantage is that downstream can proceed with operational information, but at the cost of having to iterate and re-do some part of the work if modifications occur.

We refer to the first of these two extreme strategies as "set-based." This strategy sacrifices precision, emphasizing information stability at the risk of possible starvation or duplication of downstream effort. In contrast, "iteration" or "management by engineering changes," (Ward *et al.*'s "point based" approach) emphasizes precision at the expense of increasing rework. These extreme choices are described for the sake of illustration; intermediate choices are often available.

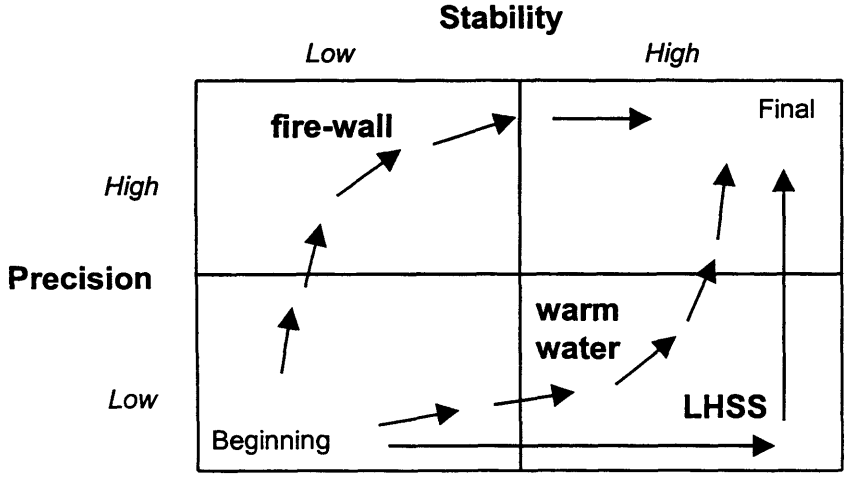

Figure 5: The Evolution of Information as a Result of a Decision Sequence

Making the choice between stable and precise communication repeatedly over time results in a path through the matrix defined in Figure 4. This path describes the way the organization resolves uncertainty over time. This is illustrated in Figure 5. In the fire-wall case, an iterative strategy was used first. Hole sizes and positions were communicated precisely, but changed repeatedly over time. The auxiliary heating concepts case is an example of a more set-based approach, where an interval was

communicated first, then narrowed over time. The LHSS provides the most extreme case. Full stability was maintained from the beginning, since the set of possibilities ("yes" or "no") did not change. Precision remained low until the engine compartment packaging was about 2/3 complete, and then jumped to 100% on the day of the management decision.

In choosing between an iterative and a set-based approach, the organization has to consider multiple factors, including the size of the residual search space, its current knowledge concerning the final outcome of the problem-solving and the cost of making a wrong decision or no decision at all. We now investigate the costs and benefits of the two approaches and show how the trade-off between the two is influenced by technical characteristics of the task and the organization's process capabilities.

Trade-offs Between Iterative and Set-Based Communication of Preliminary Information

With the cost of delay and knowledge level held constant, the choice between iterative and set-based communication depends on the relative magnitude of the costs of starvation to the costs of rework. As the relative cost of rework grows, stability becomes more attractive. This is depicted in Figure 6. Consider, for example, point A in Figure 6, which is characterized by high cost of rework and relatively low cost of starvation. Our framework recommends a focus on stability. In contrast, point B is characterized by low rework cost and a relatively high cost of starvation. The framework recommends to focus on precision.

Whereas the left-hand part of Figure 6 prescribes a communication strategy for a given cost structure, the right-hand part of Figure 6 illustrates how a change in the cost structure affects the trade-off between precision and stability. Consider point C in the right-hand part of Figure 6. *Set-based communication* becomes more attractive as point C moves toward the upper left (i.e. cost of rework are increasing relative to the cost of starvation).

Iterative communication becomes more attractive as point C moves toward the lower right.

In general, we can say that having a large residual search space makes an effective duplication (as proposed by set-based communication) more expensive and thus favors the iterative strategy. Set-based communication becomes more attractive if the information format favors "natural sets", that is, information can be communicated with a few bits (e.g., the LHSS with 0/1, or the warm water supply with liters/minute).

Based on our observations and previous research in product development, we now provide several examples of how a development organization can alter the relative cost structure between rework and starvation and what consequences this has for the communication strategy.

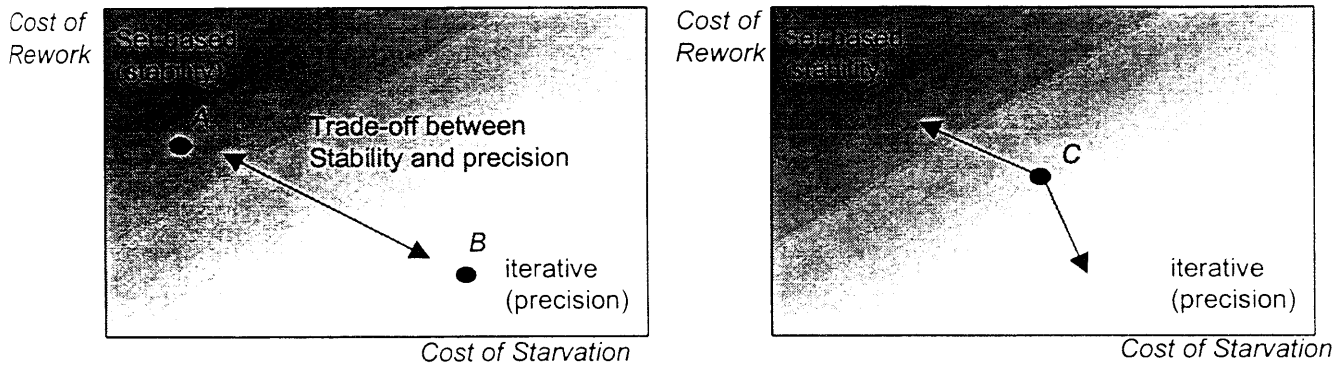

Figure 6: Trade-offs between set-based and iterative communication

Iteration Strategy, or Management by Engineering Changes

We see three ways to reduce the cost of rework (corresponding to point C in Figure 6 moving toward the lower right). Every one of them makes engineering changes (ECs), and the corresponding iterative strategy of resolving the uncertainty of preliminary information, more attractive.

First, rework cost can be reduced by loosening the coupling (dependence) between development activities. One way of achieving this is by increasing the level of product modularity (Ulrich 1995, Baldwin and Clark 1997). Product modularity prevents ECs from "snowballing" to other components (Terwiesch and Loch 1998), and thus keeps ECs local and restricted in number. Such modularity can be found in the CCS software that separates the coding from the parameterization. Thus, changes triggered by new prototyping results require a redefinition of parameters but leave the software code unchanged. This reduces the amount of time and effort necessary for an iteration and thus the relative cost of rework.

Second, rework cost can be reduced by increasing the flexibility in the downstream activity, i.e. by making downstream more responsive to upstream changes. One example of this is to use so called "soft dies" for the fire-wall stamping, which are less costly to modify. Flexibility can also be increased by relying on virtual prototypes (CAD representation, see, e.g., Baba and Nobeoka 1998) or rapid prototypes (e.g., stereolithography) rather than on full-scale physical prototypes.

Third, the problem-solving process used by the development organization is also influencing the cost of rework and thus the trade-off between the set-based and the iterative strategy. We want to discuss three aspects of this, (a) the quality of an educated guess, (b) the EC-support process, and (c) the exchange of meta-information.

(a) The iterative approach becomes more attractive if the organization is able to achieve high precision without sacrifying too much stability. In other words, if developers are capable of making an "educated guess" about the final outcome, there is a strong likelihood that the initial high precision will not cause too much rework. In the CCS development project we studied, engineers used a number of sophisticated simulation tools to analyze the warming-up of the passenger cabin, the defrosting of the windscreen, or air turbulence caused by a blower. These tools allow for an early and highly accurate prediction of the final problem-solving results (see Thomke 1998 for a more detailed

discussion), reduce the likelihood of future changes, and, ultimately, reduce the cost of rework.

(b) Effective iteration strategy requires a fast process of problem detection, problem-solving and engineering change implementation. In our study, we found lead-times of this process ranging from ten days to over ten months. Long EC lead-times typically result from complicated formal approval processes, extensive paper work, problem-solving responsibilities dispersed across different groups, and congestion effects. In these cases, substantial waiting times result, increasing the cost of rework (Terwiesch and Loch 1998). The importance of fast processes is also documented in Cusumano (1997) on the example of software development and Iansiti and West (1996) on the example of semiconductor development and experimental turnaround.

(c) In addition to exchanging the day-to-day problem-related information, an organization following an iterative problem-solving strategy should also exchange "meta-information" (Krishnan 1996). That is, the organization should make explicit to all parties involved what important interdependencies exist among which components, and what types of changes are likely to cause substantial rework. If all development engineers have this information available, they are able to anticipate problems and rework costs, which avoids bad surprises during the process.

Set-Based Strategy

Set-based concurrency has recently been described as an attractive strategy of exchanging preliminary information (Ward *et al.* 1995). In this section, we examine ways of reducing the cost of starvation and thus of making the set-based communication strategy more favorable (corresponding to point C in Figure 6 moving toward the upper left).

First, starvation can be avoided, if downstream pursues all possible outcomes in form of redundant prototyping. In order to make this approach economically feasible, the organization must be able to create prototypes at low cost. Over the past few years, many

new technologies have enabled vehicle development teams in general, and CCS development teams in particular, to reduce the cost of prototyping. For example, photolithography allows an engineer to physically create a three-dimensional geometry of a component overnight, based only on CAD data. Similarly, advances in CAx technologies[1] have reduced the costs of prototyping by offering "virtual prototypes" with short lead times and, together with simulation tools, high functionality. Such new problem-solving tools shift the trade-off curve, depicted in Figure 6, in favor of the set-based approach (Thomke 1997, Baba and Nobeoka 1998).

Second, similar to the iterative approach, there are *process capabilities* supporting a set-based approach. The set-based approach requires a deep understanding of how much accuracy the downstream activity needs to continue its work. If too little accuracy is provided, downstream starvation and delay will result. If, however, premature accuracy is provided, stability may not be maintainable, and the approach degenerates into unintended iterations. Thus, the set-based approach also requires the exchange of meta-information.

One key benefit of the set-based approach is that uncertainty becomes explicit in the information exchange. It is better to have information labeled as uncertain than to wait for information or to commit resources to information that later turns out to be wrong. As one manager in our host organization commented: "Many problems associated with concurrency have their source in attitudes: many engineers want the same information as in a sequential process, only earlier. They do not appreciate that the quality of the information has changed."

From the discussion above, we see that new prototyping technologies (such as 3-dimensional CAD, rapid prototyping, or complex system simulation) as well as system modularity tend to make *both* communication strategies more attractive and thus broaden the application of process concurrency in *all* combinations of rework and starvation costs.

[1] We use the acronym CAx to include technologies such as Computer Aided Design (CAD), Computer Aided Engineering (CAE), and other similar engineering tools.

This is consistent with Baba and Nobeoka (1998, p. 655) who state that new CAD systems may facilitate concurrent engineering, and with the observation by Ulrich (1995) that modularity allows a more parallel product development process.

Discussion and Conclusion

In this article, we provide an operational definition of preliminary information, based on the concepts of information precision and information stability. This complements the rapidly growing literature base of concurrent engineering, which focuses on information exchange frequency and often refers to the construct of preliminary information without providing a clear definition.

The managerial contribution of the article is twofold. First, the framework and the corresponding problem-solving paths enable practitioners to make a conscious choice of when and how they want to deal with preliminary information. The framework illustrates that communication frequency does not sufficiently capture the management of information flows in concurrent development processes, but the information content in terms of stability and precision must be addressed.

Second, by shedding light on the trade-offs linked to the usage of preliminary information, we provide a simple tool in order to choose between an iterative and a set-based information exchange strategy. In contrast to Ward *et al.* (1995), we argue that in communicating preliminary information, neither of the strategies, iterative or set-based, is superior as such. The key to successful product development lies in mastering both, which gives the organization the flexibility to choose its communication strategy in accordance with the technical characteristics of the engineering task and the problem-solving capabilities of the organization. We have also discussed several opportunities to reduce the cost of rework and the cost of starvation, as well as their consequences on the trade-off between information precision and stability.

By grounding our study on detailed data of engineering communication and problem-solving, collected based on a participant observer approach, we also complement the existing literature from a methodological perspective. Over the past years, most studies have relied on interviewing, survey data, or mathematical modeling, methodologies that tend to lose at least some of the richness of the underlying phenomena.

However, while our approach has allowed us to further examine the concept of preliminary information in concurrent engineering, it is not without drawbacks. Single-company studies always raise questions about biases, reproducibility, and, most importantly, generalizability. By thoroughly following methodological guidelines in our data collection, we have tried to minimize these problems, however, future research is needed to ultimately overcome them.

Several additional opportunities for future research remain. First, our exploratory investigation provides us with emerging variables and trade-offs that are important in the management of preliminary information. Our framework implicitly states a number of hypotheses that need to be tested by future research. This includes the definition of measurements and the collection of additional data, which would also overcome the limited generalizability mentioned above. Second, the trade-offs are qualitatively discussed in this article, but they may also be quantitatively formulated in analytical models or empirically tested in future studies. This would require a more detailed modeling of engineering problem-solving than provided by the existing models of concurrent engineering. Finally, it is important to explore the impact of the growing use of information technologies in product development processes. New support technologies such as CAD or rapid prototyping do substantially influence the trade-offs discussed in this article, and open up many promising avenues for future research.

Only the surface has been scratched, and much work lies ahead. This article provides a first attempt to move the management of preliminary information, in Clark and Fujimoto's (1991) words, one step from "something of an art" to a science.

References

Baba, Y., and K. Nobeoka, "Towards Knowledge-Based Product Development: the 3D-CAD Model of Knowledge Creation," *Research Policy* 26, 1998, 643 – 659.

Baldwin, C. Y., and K. B. Clark, "Managing in an Age of Modularity," *Harvard Business Review*, September - October 1997, 84 - 93.

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

Cusumano, M. A., "How Microsoft Makes Large Teams Work Like Small Teams", *Sloan Management Review*, Fall 1997, 9 - 20.

Eastman, R. M., "Engineering Information Release Prior to Final Design Freeze", *IEEE Transactions on Engineering Management* EM-27, 1980, 37-41.

Eisenhardt, K.M., "Building Theories from Case Study Research", *Academy of Management Review* 14, 1989, 532-550.

Eisenhardt, K.M. and B. N. Tabrizi, "Accelerating Adaptive Processes: Product Innovation in the Global Computer Industry", *Administrative Science Quarterly* 40, 1995, 84-110.

Flynn, B. B., Sakakibara, S., Schroeder, R. G., Bates, K. A., and E. J. Flynn, "Empirical Research Methods in Operations Management", *Journal of Operations Management* 9, 1990, 250-284.

Galbraith, J., *Designing Complex Organizations*, Addison Wesley, 1973.

Ha, A. Y., and E. L. Porteus, "Optimal Timing of Reviews in Concurrent Design for Manufacturability," *Management Science* 41, 1995, 1431 - 1447.

Hauptman, O., and K. K. Hirji, "The Influence of Process Concurrency on Project Outcomes in Product Development: An Empirical Study With Cross-Functional Teams," *IEEE Transactions in Engineering Management* 43, 1996, 153 - 164.

Iansiti, M. and J West, "Learning, Experimentation, and Technology Integration: The Evolution of R&D in the Semiconductor Industry," Harvard Business School Working Paper 96-032, 1996, forthcoming in *Research Policy*.

Krishnan, V., "Managing the Simultaneous Execution of Coupled Phases in Concurrent Product Development", *IEEE Transactions on Engineering Management* 43, 1996, 210 - 217.

Krishnan, V., S. D. Eppinger, and D. E. Whitney, "A Model-Based Framework to Overlap Product Development Activities", *Management Science* 43, 1997, 437 - 451.

Liker, J. K., D. K. Sobek II, A. C. Ward, and J. J. Cristiano, "Involving Suppliers in Product Development in the United States and Japan: Evidence for Set-Based Concurrent Engineering", *IEEE Transactions on Engineering Management* 43, 1996, 165 - 178.

Loch, C. H., and C. Terwiesch, "Communication and Uncertainty in Concurrent Engineering", *Management Science* 44(8), August 1998.

Miles, M. and A. M. Huberman, *Qualitative Data Analysis*, Sage Publications, 1984.

Terwiesch, C., C. H. Loch, and M. Niederkofler, "Managing Uncertainty in Concurrent Engineering", Proceedings of the 3rd EIASM International Product Development Conference, 1996, 693 - 706.

Terwiesch, C., C. H. Loch, "Managing the Process of Engineering Change Orders: The Case of the Climate Control System in Automobile Development," *Journal of Product Innovation Management* 15, 1998, forthcoming.

Thomke, S. H., "The Role of Flexibility in the Design of New Products: An Empirical Study," *Research Policy* 26, 1997, 105 - 119.

Thomke, S. H., "Simulation, Learning and R&D Performance: Evidence From Automotive Development," *Research Policy* 27, 1998, 55 - 74.

Ulrich, K., "The Role of Product Architecture in the Manufacturing Firm", *Research Policy* 24, 1995, 419 - 440.

Ward, A., J. K. Liker, J. J. Cristiano, D. K. Sobek II, "The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster", *Sloan Management Review*, Spring 1995, 43 - 61.

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