Management of Information Technology Driven Product Development Processes

Nitin R. Joglekar [I] λ Ali A. Yassine[II]

[I] Boston University School of Management, Boston, MA 02215

[II] Massachusetts Institute of Technology, Cambridge, MA 02139

Abstract

The evolution of Information Technologies (IT) in general and Internet technologies in particular has affected all aspects of product development (PD) processes. In some instances these technologies result in speedy and more frequent communications within existing processes. However, in many more instances these technologies result in structural changes; that is, they necessitate changes in the organization and management of development processes. This chapter explores the notion of IT driven product development (IT-PD) from a process management perspective. It provides an argument for modeling IT-PD as a set of information driven productive activities. Some of the existing models for product development process management are reviewed in light of these structural changes. Emerging opportunities for IT-PD specific research issues such as supply chain coordination, design churn through synchronization, infrastructure investments, knowledge exchanges, management of productivity and learning are discussed.*

(Design Process Management; Information Processing View; Information Technology;

Product Development and Supply Chain Management Linkage)

***Acknowledgement**

Many ideas in this paper were generated during our work with Daniel E. Whitney and Steven D. Eppinger at MIT. We want to thank the following MIT programs for providing research access and funding: LFM/SDM; LAI; and the Ford-MIT Collaboration project. The first author also wishes to thank Mr. George Orzel (Air Force Research Laboratory) and the Center for Enterprise Leadership at Boston University for their support. The authors would also like to thank Daniel Whitney, Stephen Rosenthal and Durward Sobek for providing invaluable feedback on a previous revision of this chapter.

1. Introduction

The acceptance of Internet and web browsing technologies in mainstream business processes has altered the execution of product development (PD) processes. This chapter explores the notion of Information Technology driven product development (IT-PD) from a process management perspective.[1]

The Internet has become a virtual sandbox in which designers play with a host of development and information resources. Three data panels presented in Table 1 capture the degree to which these technologies have affected the electronic design industry.

Level of Access to the Internet

Year	1998	1999
US	98%	99%
Europe	94%	99%
Asia	97%	99%

Hours of Time on Internet Per Week 1999

	Business	Personal	Total
US	6	4	10
Europe	5	3	8
Asia	10	6	16

Where in Design Process is the Internet Accessed?

	Evaluate Products	Establish Specs	Develop Concept	Locating a Brand	Seeking Design Tips	Selecting Supplier	Purchaing a Product
US	70%	52%	42%	50%	48%	44%	30%
Europe	66%	52%	44%	35%	51%	45%	23%
Asia	75%	35%	33%	22%	44%	53%	50%

Table 1: Internet Usage in the Electronics Product Development Processes
(Data Source: Cahners/ EDN Survey of 2041 design engineers and engineering managers; 861 in the US, 623 in Europe, and 557 in Asia; see Mulcahy, 2000)

The top two panels in the table show that over 99% of design engineers and engineering managers have access to the Internet and they spend 15-25% of their productive time using it. The third panel in this table shows that the Internet is being used to touch every aspect of the product development process ranging from product evaluation,

[1] Some organizations use the term ePD instead of IT-PD. We use IT-PD in its broadest sense, which covers ePD and other future IT based innovations.

establishment of specifications, seeking design tips, problem solving support, to selecting a supplier and purchasing piece parts.

In another study conducted in the aerospace and the automotive industry, we have learned that design automation tools including conventional Computer Aided Design/ Manufacturing (CAx), in conjunction with web technologies, are fundamentally changing the structure of the development process (Joglekar & Whitney, 1999). For instance, Table 2 shows that the use of these technologies speeds up the existing design processes in only 30% of the cases on average. The users tend to restructure the tasks in another 32% of instances, and these technologies result in improved designs rather than a speeded up process in 18% of the cases. Most interestingly, nearly 20% of the productive effort is expended in what we have defined to be "infrastructure" development. Infrastructure is defined as supporting information technologies, and allied artifacts such as design objects, methods, and know-how needed for product realization.

	Merely Speeded Up	Restructure & Speed Up	Only Raise Quality	Only Lead to Infrastructure Growth
Overall Average	30.0	32.6	18.2	19.1
Overall Stadard Dev.	17.2	27.3	20.5	24.8
Segmentation By Industry				
Aerospace Average	29.6	34.6	19.3	16.4
Automotive Average	31.7	23.3	13.3	31.7

Table 2: Impact of Design Automation Technology on Overall Design Flow
(Averages in the cell indicate % of total resources. See Joglekar & Whitney, 1999)

These dramatic changes raise many interesting questions: Who makes these large IT infrastructure investments? Have the omnipresent software applications changed the way PD tasks get done? How have managers adapted to these changes while managing PD processes? And has the co-evolution of B2B eCommerce and supply chain management technologies brought about new forms of market mediation within PD processes?

We have taken a closer look at the aggregate data presented in Tables 1 and 2 by conducting interviews with design managers and by reviewing several graduate theses under the LFM/SDM[2] program in the electronics, automotive, and aerospace industries. We draw upon these data and cases in the body of the text when discussing specific models. The overall trend is clear: there is a new breed of IT driven product development process being put in place. This chapter is aimed at synthesizing the state of our knowledge within the IT-PD space rather than introducing new analytical results. Our argument is rather straightforward:

- **A large body of engineering and management science literature has been built around the information exchange view of product development** (Clark and Fujimoto, 1991; Whitney, 1990; Eppinger et al., 1994).
- **IT-PD is characterized by information exchange that allows for new types of market mediation. Market mediated processes ought to be modeled by introducing intermediate agents within existing models**.

We identify research issues related to design process management decisions for both conventional PD and IT-PD environments as summarized in Table 3. This table forms the framework for the chapter. Some elements in the table are marked to show that a later section of the chapter is devoted to them. The terminology used in this table is defined as follows: PD process efficiency is measured by the speed of the overall development process. Stability is described by the ability of the process to converge (i.e. finish) over time. Learning is measured in terms of the improvement in development productivity and quality over time. Collaborative environments refer to settings where all agents (principal and supplier) have congruent incentives to share information. Market mediated environments

[2] Leaders for Manufacturing/ System Design & Management graduate programs at MIT.

describe settings where the agents form a market facing supply chain, such that the incentives for information sharing cannot be taken for granted. Local tasks refer to work conducted within a small group, whereas system tasks have a global mandate. System tasks may involve market mediation. Infrastructure is defined as supporting information technologies, design object, and methodologies needed for product realization.

Model type			Conventional PD	IT-PD
Formulation	Objective	Environment		
Efficiency	Optimal Sequencing	Collaborative	Single PD process – Rearrange tasks without changing the information structure (See Sections 2.1, 3.1)	Use models for single or multiple PD processes with New tasks; Reduced durations; New interrelationships (See Section 4.1, 4.2)
		Market Mediated	Multiple development agents treated as having independent process structures (Section 2.2)	Multiple PD processes with appropriate IT linkages (See Section 5.2)
Stability (Control)	Ensure Process Convergence	Collaborative	Single PD process - analyze eigenstructure (See Section 3.2)	Multiple PD processes – Explore supply chain synchronization issues (See Section 4.1, 4.3)
		Market Mediated	Supply chain issues are typically ignored (Section 5.1)	Market mediated supply chain issues must be considered (See Section 5.2)
Learning & Productivity	Improve PD process across multiple product generations	Local	Single activity learning (Section 3.3)	Infrastructure effects are significant (Section 5.2)
		System	Multiple activity learning (Section 3.3)	Infrastructure effects are significant (Section 5.2)

Table 3: Research Themes in Conventional and IT-PD Process Management

Subsequent sections of the chapter address the following issues. Conventional models of efficiency, stability, and learning are presented in section 3. Section 4 argues that even within a collaborative setting, efficient execution of IT-PD requires that the structure of design processes be modified to account for new tasks, reduced duration, and new types of information exchanges between existing tasks. Models and case evidence on these issues are also discussed in section 4. We posit, in section 5, that there is an opportunity to develop coupled IT-PD and supply chain decision models in the context of market mediated IT-PD processes.

2.0 Information Exchange and IT-PD

In this section we draw upon a generic information exchange / execution strategy map for interconnected PD tasks, present a market mediated information exchange framework, and define the notion IT-PD. We then argue that information exchange patterns associated with the IT-PD use models endow the PD processes with unique characteristics.

2.1 Interconnected PD Tasks

Figure 1 demonstrates the mapping between the information structure and the execution strategy of interconnected PD tasks (For details see Yassine et al., 1999a; Joglekar et al., 2001).

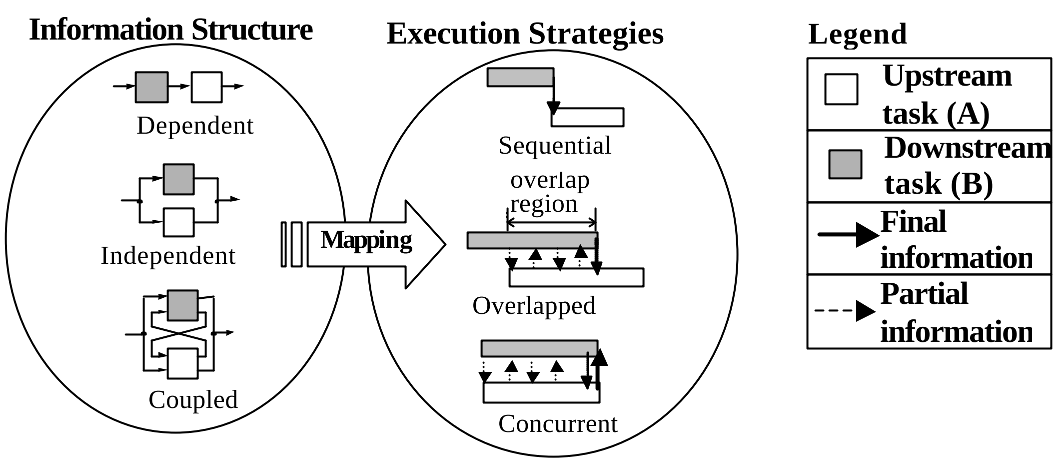


Figure 1: Mapping between Information Structure and Execution Strategies for Interconnected PD Tasks

The information dependencies between development tasks constitute the structure of the development process. In this domain, development activities are classified into three types (Eppinger et al., 1994): dependent, interdependent, and coupled. Two tasks are said to be dependent if one task depends on the other for input information. On the other hand, if both tasks depend on each other for input information, then the two tasks are coupled. Finally, if there is no information dependency between both tasks, then they are independent.

The execution strategies employed in the development process determine the development process schedule. The information structure could be mapped to three different execution strategies via different rework (i.e. development iteration) risk levels. The sequential execution of development tasks requires that upstream tasks completely finish before downstream tasks can be started. In the overlapped execution strategy, upstream tasks are scheduled to start first but downstream tasks start before the completion of upstream tasks. Finally, the simultaneous start and finish of tasks characterize the concurrent execution strategy. One such mapping model is due to Krishnan et al. (1997). They construct a model for overlapping nominally sequential activities using the notion of information evolution and PD task sensitivity.

These mappings do not address the issue of market mediated coordination between a principal (i.e. a product development organization) and its suppliers explicitly.

2.2 Market Mediated PD Tasks

Figure 2 shows a map of information exchange linkages between various agents involved in a market mediated product development process (Joglekar, 2000). This figure classifies these agents into in-house, contract design, and contract manufacturing categories. Coordination between these agents takes place through IT driven market mediation. Within this context, we define the Information Technology driven Product Development (IT-PD) as productive processes aimed at developing products by exploiting the information exchanges either within or across organizational boundaries. We emphasize that the essential impact of the Internet has been in the facilitation of both communication and provisioning of market clearance mechanisms through dis-intermediation and re-intermediation across organizational boundaries.

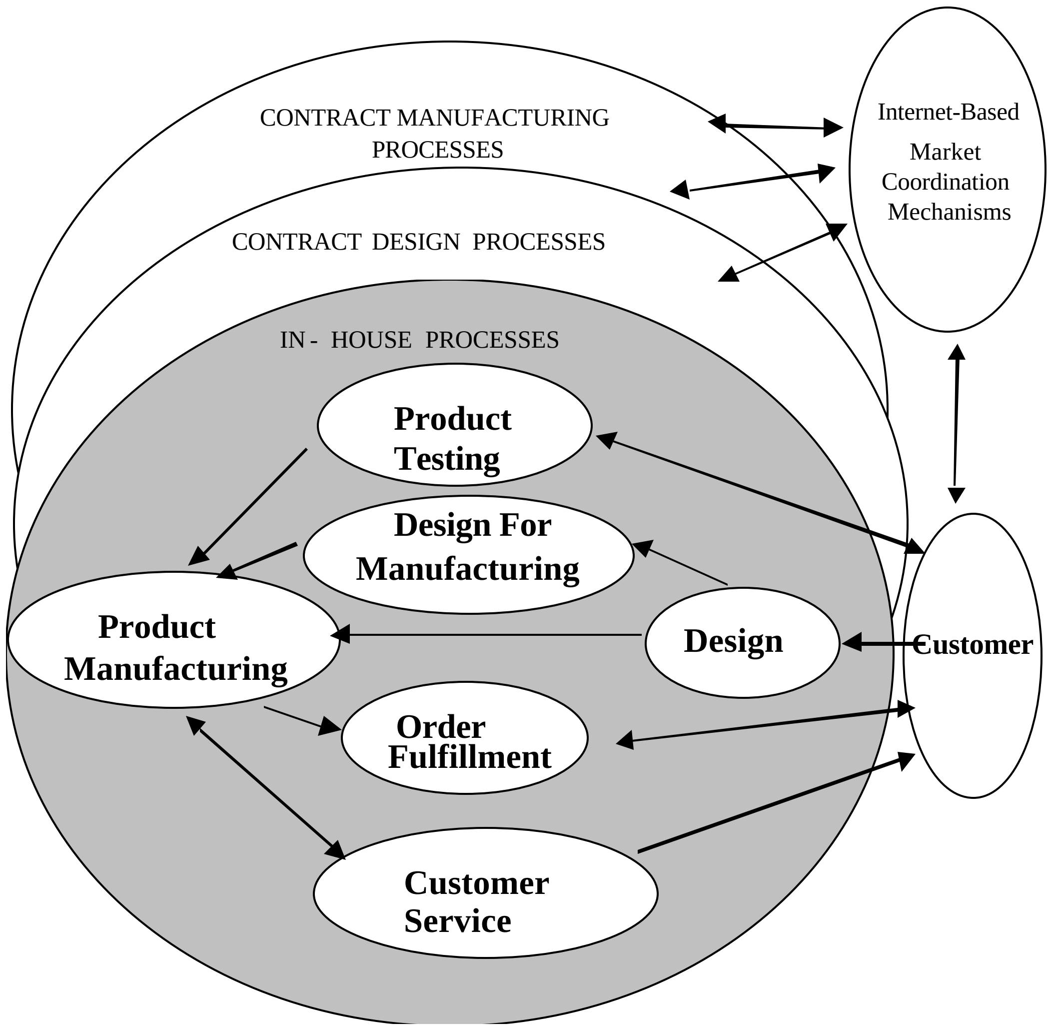


Figure 2: Market Mediated Coordination (From Joglekar, 2000)

Inter-organizational nature of product development processes is explored in some recent management science literature (Fleischer and Liker, 1997; Ulrich and Ellison, 1998; Novak and Eppinger, 2001). A new dimension is being added into this space by web driven market mediation decisions. Web driven market mediation technologies have affected PD communication processes in terms of increased speed and frequency of usage, ease of release and access, and the standardized format of the content. Interestingly, these speed-ups have been accompanied by new type of market mediated transactions. Some effects of these technologies on PD practice are discussed below.

The first impact of web technologies on the PD process has been the unbundling of the ownership of design databases and the management of bill-of-materials (BOM) related processes. Till the rise of the web, the design data and the engineering BOMs had been linked through proprietary databases under the name Product Data Management (PDM) system in the electronic and mechanical design industries. The single goal for using PDM systems had been improvement of design process efficiency through easy information access and formats translation of material and part databases. More recently, a new genre of tools, known collectively as Product Information Management (PIM) systems, has gained wide acceptance in the PDM space. PIM systems track the information exchanges, release, and version management data. Their goal is to make the design process more efficient while linking the parts data into supply chain management processes. The underlying IT architecture is typically non-proprietary and all the interfaces are web driven. Examples of software applications that support PIM are shown in Figure 3 (Collier, 1998).

This diagram shows different types of product abstraction tasks along with a variety of in-house and commercially available software applications that support these tasks. We have distinguished between two types of mediations within this figure. The boxes with white background suggest either an in-house or collaborative ownership of data. The shaded boxes emphasize that the application may be accessed through a market-mediated arrangement.

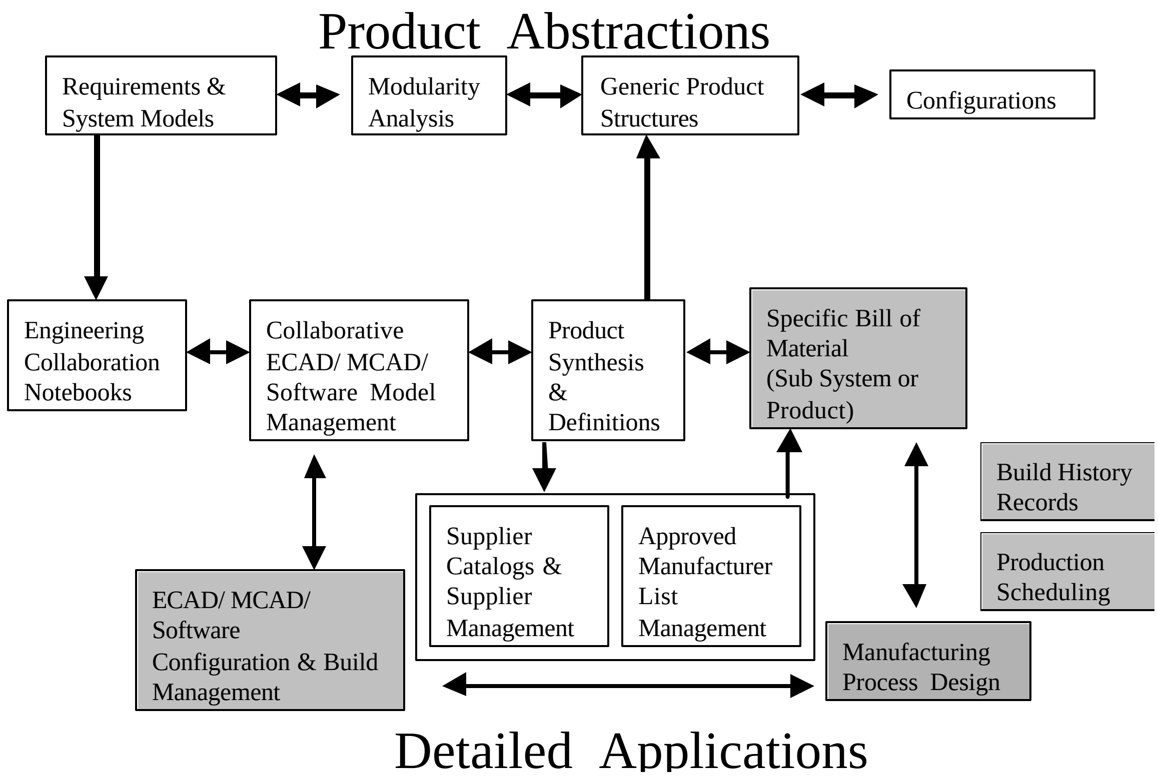


Figure 3: Product Information Management System Roadmap
(Arrows indicate the direction of information exchanges)

The second important aspect of IT-PD evolution has been the development of digital market places. In many industries, designers not only access piece part data through the web but also may quickly figure out an appropriate economic valuation for these parts. An example of this type of mediation is TabWare's RFQ site that contains part specifications and information regarding how much a supplier is willing to bid on a project. It also provides an auction capability to potentially open up a supply chain that might ordinarily rely on sealed bids (Moozakis, 2000). Other well known examples of digital market places are: Verticalnet and MetalSite. One key benefit of digital market places is the rapid reduction in transaction costs associated with market oriented activities (Verity, 2000).

Other web technology examples of note that are beginning to influence IT-PD processes are Enterprise Resource Planning (ERP) interfaces, Customer Relation Management (CRM) systems, requirement generation (Dahan & Srinivasan, 2000), and intellectual property (IP) exchanges.

In summary, some important effects of the spread of market mediation through web technologies in PD processes are:

(i) Segregation between the ownership of design and BOM databases

(ii) Creation of linkages between BOM management and ERP/ CRM processes

(iii) Creation of market clearance mechanisms for existing parts at low transaction costs

(iv) Creation of market clearance mechanisms for existing IP at low transaction costs

(v) Creation of after-markets for new IP and Parts

(vi) Signaling of the component valuation to designers and suppliers

3.0 Models of Conventional Development Processes

In this section we review some core models that address management of efficiency, stability, and learning in conventional product development processes. These models will be used as a basis for the discussion of IT-PD management issues in sections 4 and 5.

3.1 Efficiency Models

An efficient process sequence optimizes the flow of information among activities resulting in a faster product development process. One of the formalisms in process management modeling that allows studying the flow of information among activities is the Design Structure Matrix (DSM).

In the DSM representation, the PD process is modeled by a collection of interdependent development tasks. Each task receives information from other tasks, processes the information, and delivers information to subsequent development tasks. A sample DSM is shown in Figure 4. The simplest DSM model of a design process is a binary, square matrix, where the "X" marks show task interdependencies. Lower diagonal elements in the DSM represent feed forward information flows among the tasks and upper diagonal elements represent feedback. A feedback information flow captures the iteration and rework potential in a design process.

DSM representation readily reveals the existence of the three information dependency types, discussed in section 2, which allow for more efficient execution strategies. For example, the DSM in Figure 4 shows that tasks A and B are dependent since there is a dependency mark at the intersection o A's column and B's row, specifying this information dependency. Similarly, tasks C and D are shown to be independent since there is no mark at the intersection o C's column and D's row. Coupling between tasks E, F, G, and H is evident by the existence of a block indicating that each task in the block depends on the other tasks for information. Blocks in the DSM can be identified through partitioning.

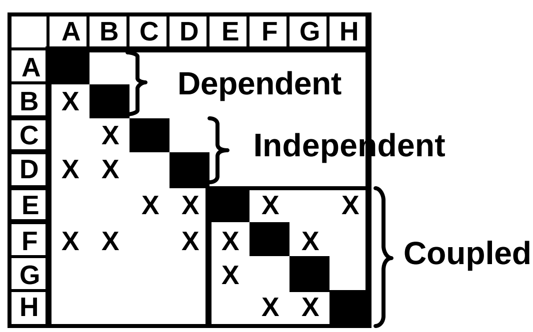


Figure 4: A representative DSM (Adapted from Eppinger et al. 1994)

Sequencing algorithms allow for the reorganization of tasks in the matrix to provide an improved sequence (Yassine et al., 1999b). This new sequence increases the efficiency of the design process, reduces product development lead time, and manages PD process risk (Browning & Eppinger, 1998).

3.2 Control Models

Control of the PD process is associated with the ability to describe and measure the convergence properties of the process. A converging PD process is a process in which the interconnected PD tasks result in a technically feasible design within a specified time frame. Alternatively, a development process that does not converge would be one where no feasible solution exists to the given specifications. Control policies may be developed either to improve the convergence process or to stabilize a non-converging development process.

Based on the DSM model of the development process, Smith and Eppinger (1997) developed a method using linear systems theory to analyze and identify controlling features of the iteration process within individual DSM blocks. The marks within a DSM block are replaced by a numerical value that describes the strength of dependencies between the coupled tasks. The measure used for dependency strength is the percentage of rework created for a task (in comparison to the amount of work required to complete the task the first time) by work performed by other tasks.

For each coupled task, the amount of work performed in a current iteration (i.e. time step) is a linear combination of the amount of work done on other tasks in a previous iteration. Working on a task reduces the workload for that task, but creates future rework for other coupled tasks. Mathematically, this is expressed as:

$$\mathbf{x(k) = A\, x(k-1) = A^k\, x(0)} \quad \text{......................(Eq.1)}$$

Where: x is the fraction of work that remains to be done on a task
k is the iteration number (i.e. time period)
A is the DSM
x(0) = 1

In the above model, the eigenvalues of matrix (A) determine the rate and nature of convergence of the development process. For convergence, the magnitude of the maximum eigenvalue for matrix (A) should be less than one[3]. An eigenvalue greater than one corresponds to a process where doing development work on a task during an iteration stage will result in a larger amount of rework in a subsequent iteration stage. A specific work policy is embedded in this model. Namely, design teams perform, at each iteration period, all the rework that was created by the other coupled tasks in the previous iteration.

3.2.1 Local Stability

McDaniel (1996) expanded Smith's model by decomposing Equation (1) into two parts: one for the dependency structure and another for an explicit work policy. In doing so, McDaniel was able to test the impact of different work policies on reducing the process lead time. In addition, McDaniel incorporated a feedback loop into the analysis, instead of the open loop system developed by Smith. The resultant model is:

$$\mathbf{x(k) = x(k-1) + A\ F\ x(k-1)} \quad \text{……………….…….(Eq.2)}$$

Where: **F** is a workload policy matrix.

The block-diagram representation of the resultant system is shown in Figure 5.

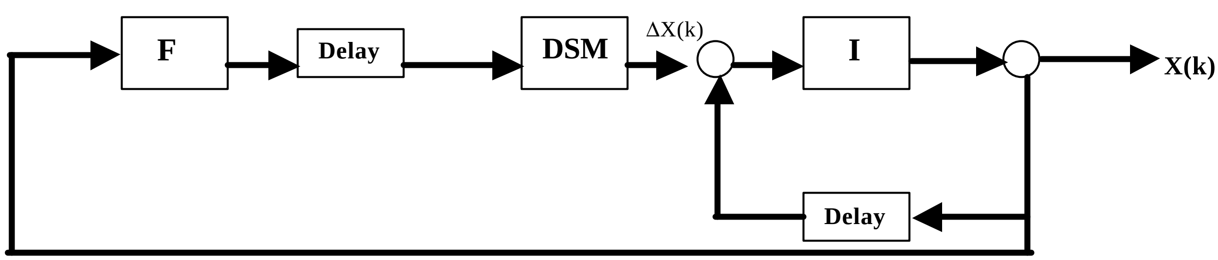


Figure 5: Closed-Loop Model (Adapted from McDaniel, 1996)

The formulation of Equation (2) considers interdependencies and work policies within the same development group (a single DSM model was assumed).[4] As such, stability becomes an intrinsic property of a single development team without regard to external

[3] In this model, there is no guarantee that the stable process is feasible.

intervention such as other development teams or management policy changes. For this reason, we refer to the stability analysis, discussed up till now, as "Local Stability."

3.2.2 System Stability

When PD occurs in a distributed environment, different development groups work concurrently on multiple aspects of the process. In his study of project management practices at a large automotive company, Mar (1999) reported that "design churning" was cited by many managers to be a major cause of project delay. Design churning was defined as "avoidable iteration and rework". Avoidable in the sense that program management has some degree of control over the factors that cause iteration and rework. Further, he discussed several reasons for design churning: poor planning, poor communication, changing requirements, and weak design freezes. As a result of design churning, project progress becomes harder to assess or measure. Progress oscillates between being on schedule (or ahead of schedule) to falling behind. Development tasks are repeated and no one knows why.

It is possible to model this phenomenon by extending the local stability loop shown in the previous section. Figure 6 shows a schematic of the exchanges within the PD process that accounts for system feedback. Matrix B, in the figure, represents a system DSM relating the sensitivity of local tasks to changes/recommendations from a system level perspective.[5] In a distributed development environment these can be system-level testing results for system integration purposes. The figure shows that the synchronization between the two workgroups happen at periodic intervals of time (T). Mathematically, the above system can be written as:

[4] Equation (2) can be analyzed for stability and matrix F can be designed to enhance stability (Joglekar and Ford, 2000; Yassine et al., 2000b).

$$\mathbf{x(k) = x(k\text{-}1) + A\ F\ x(k\text{-}1) + d(k)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\text{(Eq.3)}$$

Where $\Delta x(k)$ represents the convergence rate of the local development process, and d(k) is the system level input to the local development team – it might amount to a periodic system feedback.

Insights drawn from the analysis of Equation (3) for collaborative and market-mediated settings will be discussed in sections 4.3 and 5.2 respectively.

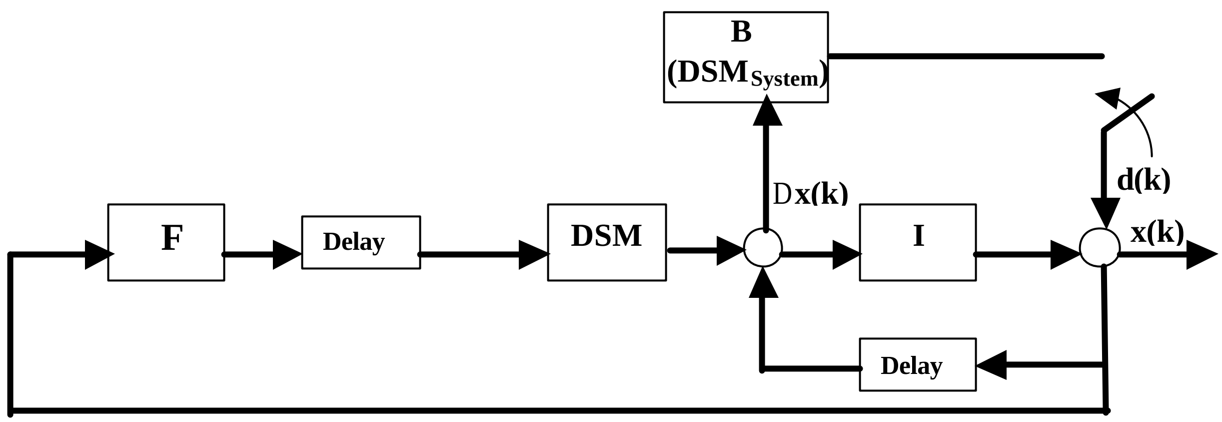


Figure 6: Closed-Loop Model with System Feedback

3.3 Learning Effects

Experimentation is becoming a central theme in modeling learning effects during product development. Key decisions in this regard revolve around the tradeoffs between the cost and benefit of testing and optimal timing for testing (Thomke, 1998; Thomke and Bell, 2001). In conventional PD processes, productivity has been measured while assuming that the technology is stable (Clark and Fujimoto, 1991). Both experimentation and productivity measurement formulations can be explained in terms of local and system feedback loops described in Sections 3.1 and 3.2. We will further discuss these issues in Section 4.4.

[5] Matrix B amounts to changing the scope of the work performed by individual groups. Example of an existing model addressing these issues is set based concurrent engineering (Sobek et al., 1999).

4. Management of Collaborative IT-PD Processes

PD organizations realize that improving PD process efficiency and convergence require high levels of collaboration among development teams. In large organizations, these teams are likely to be dispersed across several geographical locations (Vedapudi, 2000; Sosa et al., 2000). In such a distributed environment, collaboration among development teams becomes more difficult and a real challenge for IT tools. These tools should be able to support sharing of evolving (i.e. incomplete, partial and preliminary) design information, maintained in distributed databases and in different formats. Another level of complexity is added when some of these teams belong to different organizations (i.e. suppliers). Facilitating PD collaboration across several enterprises is yet another major IT concern in product development. In this section, we review the use models within collaborative IT-PD and discuss the corresponding efficiency, stability, and learning issues.

4.1 Use Models

Creating an efficient IT system requires the development of PD process models capable of capturing not only the development steps in the process, but also the different interactions that exist at the local and system levels. Whitney and Patil (2000) suggested the "use models" schema in order to address several important issues in developing IT solutions for product development process management. A use model consists of a set of development tasks, methods, tools, and a recipe (i.e. process map) for their use by PD teams.

- Tasks: actions that need to be performed for completing the PD project.
- Methods: how the actions should be performed.
- Tools: software applications needed to perform the actions.
- Process map: interactions and interrelationships between the different tasks.

Figure 7 shows the critical parts of a use model. It includes the methods and tools that appear as rows and columns in the DSM. It is important to know not only that there are dependencies between tasks but also their magnitude, the way they are handled, and who is responsible for them (Yassine et al., 1999b). The use model changes over time, as better modeling techniques and tools evolve. For example, the introduction of CAD workstation eliminated the old paper-pencil use model of creating 2-D drawings.

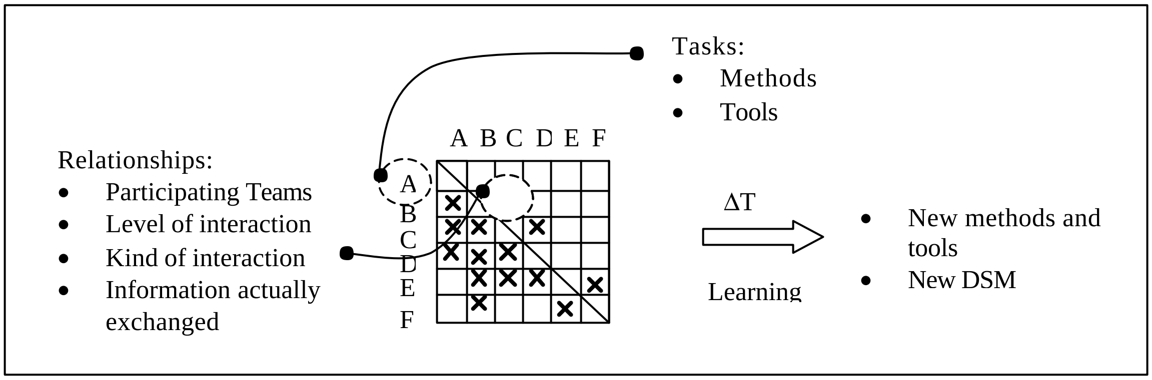


Figure 7: DSM and Use Models (Adapted from Whitney and Patil, 2000)

4.2 Efficiency

An efficient IT system should be able to deliver on several dimensions.

1. Faster execution of development tasks: using IT based design and analysis tools speeds up individual tasks in the PD process.
2. Faster information transfer: using IT communication capabilities result in faster and better information access and release.
3. More concurrency and new task structures: using IT to change the information structure of the PD process allows for more concurrency in executing development tasks; thus, reducing total development time.

The introduction of IT systems into PD usually changes the structure of the PD process. Aoshima et al. (1999) found that the impact of introducing 3D-CAD into engineering organizations is not limited to changes in individual design tasks. Advanced

3D-CAD systems also change existing task boundaries for individual design tasks and the relationships between multiple design tasks. Process restructuring due to IT include the elimination of some old tasks, the addition of new tasks, and the revision of some task dependencies. For example, Yassine et al. (2000a) reported two `case studies where the development process was restructured/reengineered through the incorporation of IT solutions. In both cases, the DSMs for the "as-is" and the "re-engineered" development processes show the impact of IT on the process tasks and information structure. In the first case study, a development process for an automotive seat belt assembly was reengineered by introducing an electronic catalog that allows organizationally and geographically dispersed design participants to use one source of design information, eliminating the need to check for inconsistencies downstream.

		2	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Select Material for all system components	2						6	6						6			4						4
Freeze proportions and selected hardpoints	4			6	4																		
Verify that hardpoints and structural joints are	5				4																		
Approve master sections	6		6	6					9														
Develop initial design concept (Prel. CAD mo	7			6	6				6		6	4				4			6				6
Estimate blank size	8				4	6						4				4			3				
Estimate efforts	9	6																					
Develop initial attachment scheme	10				6	6						6				4			3				4
Estimate latch loads	11				6	9																	4
Cheat out panel surface	12	6				6				3						4							
Define hinge concept	13				4														3				
Get preliminary mfg & asmbly feasibility	14				4	6	6		6			4							3				
Perform cost analysis	15	6			6	6	4		3			4	4			4	4		3				
Perform swing study	16				4							4							6				
Theme approval for appearance	17				4								4						3				
Marketing commits to net revenue	18															4							
Program DVPs and FMEAs complete	19	4				6			3			4											
Approved theme refined for craftsmanship exe	20	4														4							
PDN0 - Class 1A surfaces to engineering	21	4				3									3	6			6		6	4	
Conduct cube review and get surface buyoff	22	4			4														3	6		4	
Verify mfg and asmbly feasibility	23				6	9	4		6			4	6			6			3	4	3		
Evaluate functional performance (analytically)	24	6				9			6			4	4							4			

(a) The original process

		2	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	21	22	24	20	23	25	26	27
Select Material for all system components	2										6	6						6		4						
Freeze proportions and selected hardpoints	4			6	4								6													
Verify that hardpoints and structural joints are c	5		4		4					6																
Approve master sections	6		6	6				4																		
Generate structural requirements (analytically)	7												3						4							
Develop conceptual design strategy	8					3																				
Develop structural CAD model	9		6				4										4		4							
Verify functional performance (analytically)	10	6						4																		
Develop prelimin. Design intent CAD model	11				6				3				6		6	4			4		6					
Estimate blank size	12				4					4						4			4		3					
Estimate efforts	13	6																								
Develop initial attachment scheme	14				6					4						6			4		3					
Estimate latch loads	15				6					6																
Cheat out panel surface	16	6								4				3					4							
Define hinge concept	17				4																3					
Get preliminary mfg & asmbly feasibility	18				4					4	6		6			4					3					
Perform cost analysis	19	6			6					4	4		3			4	4		4	4	3					
Theme approval for appearance	21				4												4				3					
Marketing commits to net revenue	22																		4							
Approved theme refined for craftsmanship execu	24	4																	4							
Perform swing study	20				4											4					6					
Program DVPs and FMEAs complete	23	4								4			3			4										
PDN0 - Class 1A surfaces to engineering	25	4																	6		6	3			6	4
Conduct cube review and get surface buyoff	26	4			4																3			6		4
Verify mfg and asmbly feasibility	27				6					6	4		6			4	6		6		3			4	3	

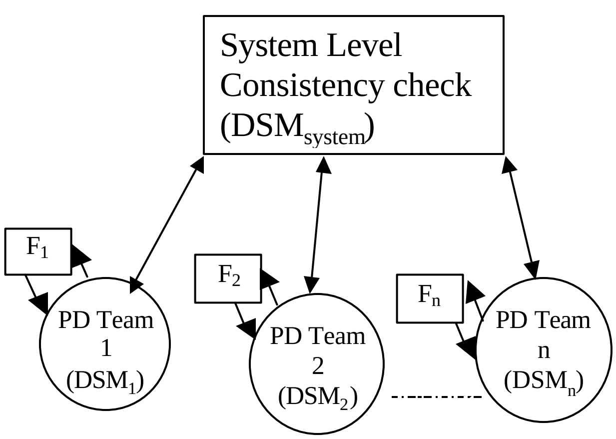
(b) The re-engineered process

Figure 8: DSM for the Hood Development Process
(Source: Yassine et al. , 2000a)

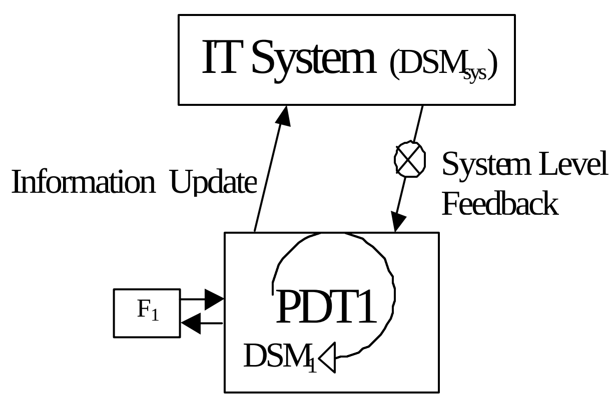
In the second case study, an automotive hood development process, shown in Figure 8a, was reengineered by incorporating a recent IT solution in body development called "hood inner panel generator". The re-engineered process is shown in Figure 8b, where the shaded tasks in the DSM represent the new IT system. Once the generator is given the locations of structural beams and the outer panel surface, it can generate a rough, but representative CAD model. While it provides only a crude CAD model that is unsuitable for developing tooling cutter paths, for example, it is suitable for structural analyses. It is interesting to note that the number of iterations in both process configurations is the same; however the reengineered process finishes sooner.

4.3 Stability

The frequency of system level feedback might depend on either exogenous considerations (such as suppliers ability to provide updates) or endogenous considerations (such as system level test requiring a minimal turn around time for a desired fidelity). If the synchronization is instantaneous, for example during daily builds of Microsoft's Development Cycles (Figure 5.1 in Cusumano & Selby, 1995), then in effect we can think about the whole process in terms of a unified (combining local and system level) DSM. It is also possible to set up the system level updates in a sequential manner. That is, the system provides feedback only when individual groups solve their currently assigned set of problems. Successive cycles will result in fewer open issues provided that the system is stable.



(a) Interaction between system and local teams



(b) An IT system facilitating the interaction

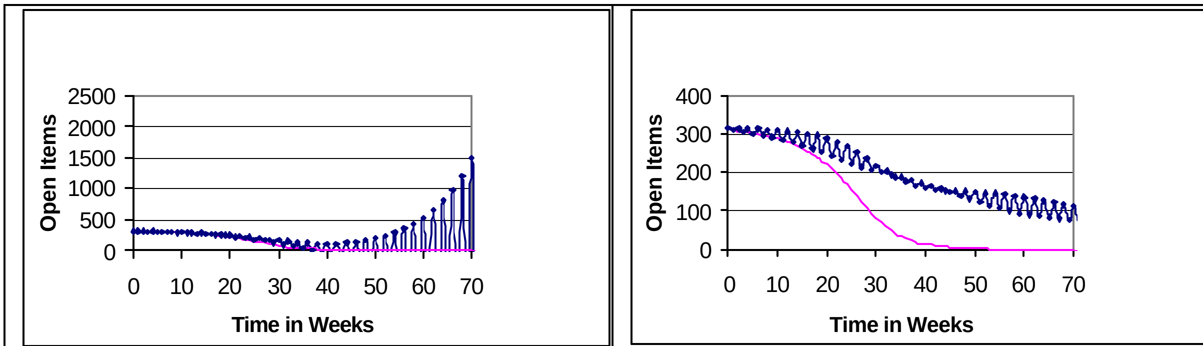
Figure 9: System and Local DSMs Facilitated by an IT system

One major source of design churn, described in section 3.2, is the frequency of synchronization between PD teams and system level considerations/testing. Coordination between groups takes place through an information system. Individual groups provide status updates to the information system. This information is processed based on global considerations, which may result in rework for some of the individual groups. Figure 9 shows a schematic of the information exchanges within the PD process described above.

Information updates from various PD teams take place on a regular basis. The system level feedback is provided on periodic basis as shown in Figure 9 (b). An interesting research question is: how might multiple DSMs that are mediated by an IT system be managed? That is:

- How often should or can we provide useful system level feedback?
- Given this set up, how might work policies at the local level be structured?

To answer these questions, system stability can be investigated by the incorporation of system feedback elements as shown earlier in Figure 6. Figure 10 shows two simulated scenarios for an automotive PD process modeled using the structure described in Figure 6 (Joglekar et al., 1999). Part (a) of the figure shows an unstable process. Part (b) of the figure shows that the same process can become stable by merely changing the system feedback. It is worth noting that Figure 10 (b) shows a slow drift (beat like) behavior that might be viewed by managers as a design churn leading into an unstable resource allocation policy.



(a) Unstable

(b) Beat

Figure 10: Stability Assessment of PD Process

4.4 Learning Effects

Figure 11 (Joglekar & Whitney, 1999) captures the design process flow in a collaborative IT enabled design environment. This flowchart shows that the development of complex products takes place within a context provided by the design automation/ web infrastructure. Based on this idea Joglekar and Whitney have proposed a scale-scope framework that classifies product development tasks as (i) core operation or communication (ii) knowledge capture (iii) infrastructure development. We argue that in IT-PD environments, firms need to invest significant amount of their development resources to continually update their infrastructure. Core operation productivity is predicated on the effective management of the knowledge capture processes for each infrastructure generation.

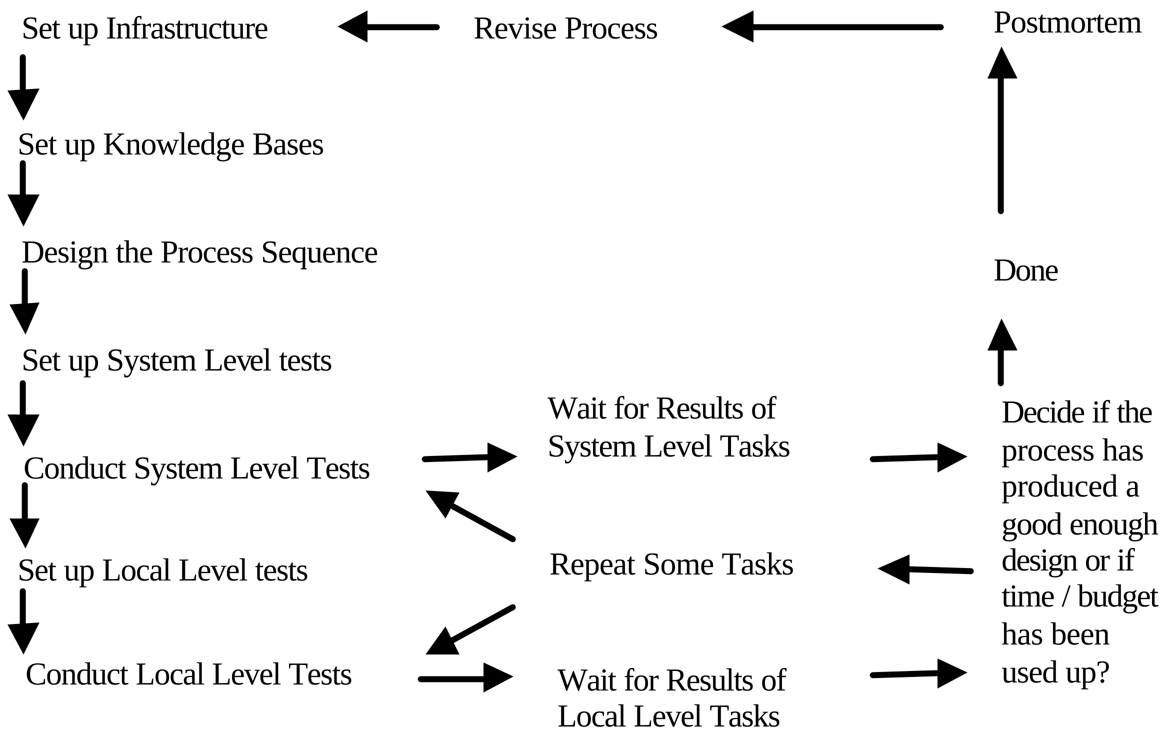


Figure 11: Infrastructure and IT-PD Process

The scale-scope matrix yields six cells as shown in Table 4. These have been named: Execution, Coordination, Single Activity Learning, Multiple Activity Learning, Generic Solutions, and Platforms.

		Scale of Design Automation	
		Local	System
Scope of DA	Core Operation or Communication	Execution *Atomic Design Tasks *Information Releases *Lateral Control	Coordination *Sequence *Interdependencies * Top Down Control
	Knowledge Capture	Single Activity Learning *Observation *Analysis *Experimentation	Multiple Activity Learning *Alternatives *Multiple Knowledge Bases *Aggregation
	Develop Infrastructure	Develop Generic Solutions *Parts/Objects for Reuse *Processes for Reuse *Knowledge Codification	Develop Platforms *Decomposition *Interoperability *Standardization

Table 4: Design Automation Assessment Framework

“Local” is defined as a task within the control of one geographic site, or a design group with networked tools. “Systemic” is defined to be pertaining to the overall context or system performance. Core Operation or Communication refers to atomic tasks involving information generation via design transformation or information exchange. Knowledge Acquisition tasks allow a designer to learn about the behavior of artifacts/objects within the context of an infrastructure. Infrastructure is defined as supporting technologies that are needed for product realization. A detailed discussion of individual cells within this framework is available in (Joglekar & Whitney, 1999).

In conventional PD processes, productivity has been measured while assuming that the technology is stable (Clark and Fujimoto, 1991). This approach might be justified if only a small amount of productive effort during PD is devoted to infrastructure development. However, in IT-PD settings nearly 20% of the overall design task are linked with the development of IT related infrastructure artifacts. Significant amount of time might be spent in experimentation though the use of simulation environments (Thomke, 1998). We also postulate that these experimentation deal with both local (i.e. single activity learning) and system (multiple activity learning) behavior. There are significant set up times associated with these iterations even within collaborative settings that lead to congestion effects within the PD cycle (Loch & Terwiesch, 1999; Pekoz and Joglekar, 2000).

This view of IT-PD processes suggests that the infrastructure governs the effectiveness of single activity and multiple activity learning, and becomes a significant variable while measuring the efficacy of core operating tasks. We end this discussion by suggesting that measurement of IT-PD productivity not proceed by assuming a stable technology environment. Another issue of interest within this space is the amount of infrastructure investment and complementary nature of the relation between the size of infrastructure and productivity of the technology supply chain.

5.0 Management of Market Mediated IT-PD Processes

We begin this section with a discussion of market mediation issues and describe an architecture to implement supply chain agents within an IT-PD setup. We provide a temporal framework linking PD and Supply Chain Management decisions. Finally, we discuss how the PD process models, described earlier, can be augmented to accommodate market-mediated PD processes.

5.1 Supply Chains and IT-PD

The growth of web in B2B settings has been accompanied by a rising interest in supply chain management. Conventional analyses within the supply chain literature is centered on optimizing the placement of inventories (Ganeshan et al, 1998). Some of the recent web related developments build linkages between supply chain management and IT-PD processes. An example of this type of linkage in the electronics design industry is the Surelink Supply Chain Integration program announcement by Cadence Design services (Cadence, 2000). Their service brings together disparate partners such as Intellectual Property (IP) /virtual component providers, software IP/ component providers, Integrated Circuit (IC) manufacturing firms, packaging and test services providers, contract assembly/ manufacturing service providers, and logistics/ distribution providers into a loosely coupled coalition. In this industry, product life cycles are rather short. Supply chain and PD processes must simultaneously account for inventory, product configurations, and intellectual property considerations. As an example, recent research coupling some of these choices in the electronics industry is the work by Graves and Willems (2000).

The choice of IT infrastructure provides a context within which market mediated information exchanges take place across the supply chain. One architecture for market mediated information exchange infrastructure has been proposed by (Salminen et al., 2000) as shown in Figure 12.

This type of architecture extends the previous literature on collaborative product development (See Tong and Sriram, 1992) by modeling the market mediation through a brokering mechanism.

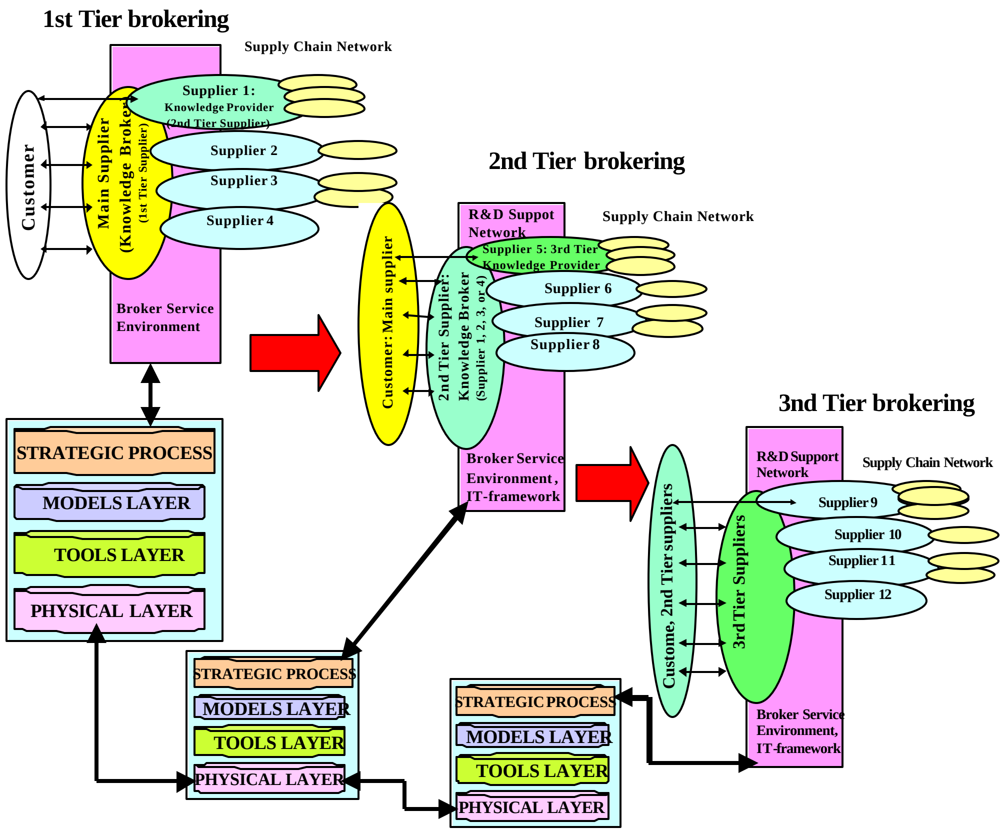


Figure 12: Product Development with Supply Chain Network

The degree to which the systems are either loosely coupled or hardwired is a strategic choice for the PD firm. In industries with a short PD life cycle (one to three years), the infrastructure is typically updated at the beginning of a major PD program. Typically firms are reluctant to change their infrastructure halfway through a product development cycle. For larger product life cycles, infrastructure updates and supply chain redesigns are being aligned with mid-life kicker product upgrades. Table 4 shows the types of coupled decisions that must be made. There is an inverse relation between the degree of coupling and the decision time horizon. Therefore, there exists a need to augment PD process management models to accommodate market mediation effects, especially with longer decision horizons.

Decision Time Horizon	IT-PD Process Management Issues	Supply Chain Management (SCM) Issues	Possible Areas of Overlap between IT-PD and SCM
Multiple Generations; (Several Years)	Learning; Level of Design Outsourcing; Design Flexibility	Learning; Level of Outsourcing; Production / Logistics Flexibility	Level of Outsourcing; Learning; Design- Logistics Flexibility
Single Generation (Typically one year +)	IT/ Infrastructure investments, IP outsourcing; Part Choices (configuration, postponement; ramp ups); Game theoretic Analysis of IP pay off and information asymmetry	IT/Infrastructure investments; Product Configuration, Variety Management; System Design; Postponement; Ramp-ups; Game theoretic analysis of inventory clearance;	Configuration, Variety Management; ramp-up; Postponement; Game theoretic Analysis of IP; inventory clearance and information asymmetry
Quarterly/ Monthly	System Optimization; System level Synchronization	System Optimization; Forecasting; Demand Smoothing; Information Delays	Some; Particularly pertaining to stability issues
Monthly/ Weekly	Flow planning and Sequencing; Local learning; Synchronization	Flow Planning	PIM / ERP Interfaces
Daily	Task set up and Queuing Effects; Local Learning and synchronization	Transactions	Minimal

Table 5: IT- PD and SCM Decisions of Interest [6]

5.2 Models of Market Mediated IT-PD Processes

To the best of our knowledge, not many PD models attempt to capture the nuances of the market mediated PD decision space. In this section we discuss how conventional models, discussed earlier, may be augmented to accommodate market mediation mechanisms.

As noted in the previous section the coupling of supply chain considerations with IT-PD results in different types of augmentations to the PD process models. There can be a variety of inter-organizational efficiency models under market-mediation involving sequencing of PD tasks, optimal frequency of information exchanges, and design of product components with the supply chain considerations. Table 6 lists potential augmentation opportunities for these models. Similarly, the table lists potential augmentation opportunities for stability and learning models referenced in the previous sections. An example augmentation scenario is discussed next.

Type of Design Process Model	Decisions within collaborative IT-PD	Augmented Model for Market Mediated IT-PD
Efficiency	DSM based task sequencing (e.g. Smith & Eppinger, 1997; Browning & Eppinger, 1998)	Each firm has an independent DSM; Linkages involve information hiding and delays
	Degree of overlap based on evolution and sensitivity (Krishnan et al, 1997)	Exogenous constraints on task overlapping and frequency of information exchange across organizational boundaries
	Simultaneous design of components & supply chains (Graves & Willems, 2000)	Search mechanism for alternatives; gaming behavior by the actors
Stability	Minimization of design churn caused by rework (McDaniel 1996, Yassine et al, 2000b)	Market clearance mechanisms may govern design convergence
	Amount of design release based level of task completion (Sobek et al. 1999)	Amount of design release may be guided by valuation of components/ intellectual property
Learning	Timing and amount of experimentation (Thomke, 1998, Thomke and Bell 2001)	Timing & amount of experimentation is based on market valuation of the design options (Baldwin and Clark, 2000)

Table 6: Augmentation Opportunities for Market Mediated PD Process Models

[6] We have drawn on recent papers from the Supply Chain Management literature to think about the timing issues associated with some of SCM decisions (Ganeshan et al., 1998).

In most conventional development projects, the transaction costs for getting market feedback are rather large and developers tend to use rough estimates for costs and market feedback to expedite the process. Arguably, if there is an incentive for the developers to get information about component valuation in a market mediated setting, it may be possible to get this information through an on-line auction at a low cost. However, these developers may have to restructure their tasks, and possibly add new ones, to take full advantage of these auctions. For instance, in the hood development example, described in Figure 8, one of the PD tasks is "perform cost analysis." If this task involves a market auction, then the existing DSM structure must be augmented to account for it. That is, an early, but preliminary, CAD model can result in a low transaction cost bid with a large variance on the part cost, while a detailed/late model might lead to a lower variance. Thus a process management model dealing with risk has to account for the market-mediation options. Supplier qualification is another aspect of market mediated IT-PD. Supplier performance and reputation are routinely quantified in electronic transactions. These performance measures could be used to establish the probability of rework used in DSM analysis when evaluating efficiency. Similarly, the variability in the supplier's ability to synchronize with the system DSM can be used to establish the stability of the process.

Market mediation influences learning due to transaction delays, congestion effects, and information hiding. Furthermore, it may provide a hedge against technological obsolescence effects associated with learning. Thus, proper incentives and infrastructure investments that facilitate smooth information transfers have direct bearing on PD process learning.

6.0 Conclusion

Information technology plays a central role in determining the efficiency, stability and learning associated PD processes. This chapter discusses ways in which information technology is leading to structural changes within PD processes. We have described some models that can be used to make these processes more efficient and stable.

This chapter is focused on a process management view of the IT-PD phenomenon. It may be worthwhile to look beyond the process view. Some alternative perspectives within of this domain are: 3D Concurrent Engineering (Fine, 1998), PD process implementation in the AI/Computer Science literature (Tong & Sriram, 1992), organizational behavior (Fleischer & Liker, 1997), and the role of web in product definition (Dahan and Srinivasan, 2000). It might also be interesting to think about who owns the oversight of IT-PD in an organization: is it the CEO, CTO or CIO?

In conclusion, the chapter argues for coupling IT-PD and supply chain management decision models in market mediated settings and posits that IT/infrastructure investments set up the context for learning across multiple generations. Productivity of technology supply chains is predicated on effective management of IT-PD processes.

References

Aoshima, Y., Nobeoka, K. and Y. Takeda, "Benefiting from Task-Redefining Process Technology: The Influence of an Organizational Process on the Introduction of 3D-CAD." Working Paper #99-15, Institute of Innovation Research, Hitotsubashi University, 1999.

Baldwin, C. and K. Clark, *Design Rules: The Power of Modularity.* MIT Press: Cambridge, MA, 2000.

Browning, T. and S. D. Eppinger, "A Model for Development Project Cost and Schedule Planning." MIT Sloan School of Management Working Paper Number, 1998.

Cadence Design services, http://www.cadence.com/company/pr/05_24_00ds.html, 2000.

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

Cusumano, M. and R. Selby, *Microsoft Secrets*. Free Press: New York, 1995.

Collier, W., "Models, Abstractions, and Transformations: The Essence of Product Information Management." *Proceedings of Implementation Road Map*. D. H. Brown Associate: Dearborn, MI, 1998.

Dahan, E. and V. Srinivasan, "The Predictive Power of Internet-Based Product Concept Testing Using Visual Depiction and Animation." Journal of Product Innovation Management, March 2000.

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

Fine, C., *Clock Speed: Winning Industry Control in the Age of Temporary Advantage*. Perseus Books: Cambridge, MA 1998.

Fleischer, M., and J. Liker, *Concurrent Engineering Effectiveness: Integrating Product Development across Organizations*. Hanser Gardner: Cincinnati, OH, 1997.

Ganeshan, R., Jack, E., Magazine, M.J. and P. Stephens, "A Taxonomic Review Of supply Chain Management Research" in *Quantitative Models for Supply Chain Management*. Tayur, Ganeshan, and Magazine (editors), Kluwer Press: Boston, 1998.

Graves, S. C. and S. P. Willems, "Optimizing the Supply-Chain Configuration for New Products," Proceedings of the 2000 Manufacturing and Service Operations Management Conference, Ann Arbor, MI, 2000.

Joglekar N.R. and D.E. Whitney, "Automation Usage Pattern During Complex Electro Mechanical Product Development," Center for Technology Policy and Industrial Development Report, Massachusetts Institute of Technology, Cambridge, MA, 1999.

Joglekar, N.R., "Web Driven Development Workshop Report," Center for Enterprise Leadership Working Paper, Boston University School of Management, 2000.

Joglekar, N. and D. Ford, "Resource Allocation During Product Development Projects," Presentation at the annual meeting of the Institute for Operations Research and Management Sciences (INFORMS), San Antonio, November 2000.

Joglekar, N., Patil, S., Whitney, D., and A. Yassine, “Design Automation Use Models: A Framework for Controlling Design Releases,” Presentation at the annual meeting of the Institute for Operations Research and Management Sciences (INFORMS), Philadelphia, November 1999.

Joglekar, N.R., Yassine, A., Eppinger, S.D. and D.E. Whitney, “Performance of Coupled Product Development Activities with a Deadline.” Management Science, 47 (12), 2001.

Krishnan, V., Eppinger, S. D., and D. E. Whitney, “A Model-Based Framework to Overlap Product Development Activities.” Management Science, 43 (4), 437-451, 1997.

Loch, C. and C. Terwiesch, “Accelerating the process of engineering change orders: capacity and congestion effects.” Journal of Product Innovation Management, 16 (2), 145-159, 1999.

Mar, C., “Process Improvement Applied to Product Development.” Unpublished MS Thesis, MIT Leaders for Manufacturing Program 1999.

McDaniel, C.D., “A Linear Systems Framework for Analyzing the Automotive Appearance Design Process.” Unpublished Thesis, MIT Leaders for Manufacturing Program 1996.

Moozakis, C., “Asset Management Vendors extend into E-Procurement.” Internetweek, June 19, 2000.

Mulcahy, S., “A Whole New World.” Electronics Business, Special Annual Issue 2000.

Novak, S. and S. Eppinger, "Sourcing By Design: Product Architecture and the Supply Chain.” Management Science, 47 (1), 189-204, 2001.

Pekoz E. and N. Joglekar, “Queues with Highly Delayed Feedback.” Boston University School of Management Working Paper, 2000.

Salminen, V., Yassine, A. and A. Riitahuhta, "A Strategic Management Framework for Collaborative Product Development." Proceedings of the 4th International Conference on Engineering Design and Automation, Orlando, Florida, USA, July 30 - Aug 2, 2000.

Smith, R.P. and S.D. Eppinger, “Identifying Controlling Features of Engineering Design Iteration.” Management Science, 43(3), 276-293, 1997.

Sobek, II, D.K., Ward, A.C. and J.K. Liker, "Toyota's Principles of Set-Based Concurrent Engineering." Sloan Management Review, 40 (2), 67-83, 1999.

Sosa, M., Pich, M. and S. Eppinger, "Factors that Influence Technical Communication in Distributed Product Development: An Empirical Study in the Telecommunications Industry." MIT Sloan School of Management Working Paper, 2000.

Thomke, S. "Managing Experimentation in the Design of New Products." Management Science, 44(6), 743-762, 1998.

Thomke, S. and D. Bell "Sequential Testing in Product Development." Management Science, 47(2), 308-323, 2001.

Tong, C. and R. Sriram. *Artificial Intelligence in Engineering Design*. Academic Press: Boston, 1992.

Ulrich, K. and D. Ellison, "Beyond Make-Buy: Internalization and Integration of Design and Production." Working Paper, The Wharton School, 1998.

Vedapudi, M., "Product Information Management (PIM) Infrastructure for Partner Programs." Unpublished Thesis MIT System Design and Management Program 2000.

Verity, J., "Digital Marketplaces," http://www.time0.com/Press/Whitepapers/digital_marketplaces.pdf, 2000

Whitney, D.E., Dong, Q., Judson, J. and G. Mascoli, "Introducing Knowledge-Based Engineering into an Interconnected Product Development Process." M.I.T. Center for Innovation in Product Development, Cambridge, MA, White Paper Jan. 27, 1999.

Whitney, D., "Designing the Design Process." Research in Engineering Design, 2(30), 1990.

Whitney, D. and S. Patil, "Mapping the Product Development Process to Information Technology Solutions through Use Models." Proceedings of ASME International Design Engineering Technical Conferences, September 10-13, 2000.

Yassine A., Chelst, K. and D. Falkenburg, "A Decision Analytic Framework for Evaluating Concurrent Engineering." IEEE Transactions on Engineering Management, 46 (2), 144-157, 1999a.

Yassine A., Falkenburg, D. and K. Chelst, "Engineering Design Management: an Information Structure Approach," International Journal of Production Research, 37 (13),

Yassine, A., Whitney, D., Lavine, J. and T. Zambito, "DO-IT-RIGHT-FIRST-TIME (DRFT) Approach to Design Structure Matrix (DSM) Restructuring." Proceedings of the 12th International Conference on Design Theory and Methodology, 2000a.

Yassine, A., Joglekar, N., Whitney, D. and S. Eppinger, "Convergence of Product Development Processes." Presentation at the MIT Design Structure Matrix (DSM) Workshop, 2000b.