Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

MODELING DESIGN DEVELOPMENT IN UNPREDICTABLE ENVIRONMENTS

Nuno Gil
Constr. Engrg. & Mgmt. Program
Department of Civil and Environmental Engineering
215 McLaughlin Hall
U.C. Berkeley, CA 94720-1712

Iris D. Tommelein
Constr. Engrg. & Mgmt. Program
Department of Civil and Environmental Engineering
215 McLaughlin Hall
U.C. Berkeley, CA 94720-1712

Robert Kirkendall
Industrial Design Corporation
2020 S.W. Fourth Avenue, 3rd Floor
Portland, 97201 Oregon

ABSTRACT

This paper presents a product-process simulation model representative for design development of a building system in an unpredictable environment. Unpredictability means that design criteria are prone to change as design development unfolds. The model was implemented with a discrete-event simulation engine based on event graphs. Events capture moments when tasks start or end and update state variables. State variables express decisions on product design parameters. Design criteria changes are modeled as stochastic events that cancel future scheduled events and schedule new design iterations. Between conceptualization and concept development, we assume that managers can impose a time lag so as to minimize rework of concept development tasks from upstream changes of design criteria. Simulation illustrates the effects of adopting different postponement strategies. The results show that postponing concept development consistently reduces the average resources spent in concept development and increases process reliability, but augments the average design duration.

1 INTRODUCTION

Design processes are intrinsically complex in nature. Complexity stems from diverse factors, such as interdependencies and coordination needs between tasks carried by distinct design disciplines, the iterative nature of the design process in its search for satisfying solutions, the criticality of compressing development time, and the unpredictability of design criteria (e.g., Chrichton 1966, Simon 1969, Gebala and Eppinger 1991, Conklin and Weil 1997, Eisenhardt and Tabrizi 1995, Iansiti 1995).

Empirical studies have shown that postponement of design decisions is a critical strategy for managing development processes unfolding in unpredictable environments (e.g., Iansiti 1995). At Toyota automotive company, Ward et al. (1995) observed that decisions on design parameters are frequently postponed until the last possible moment so designers can have more time to refine the design, understand clients' expectations, and ensure design is manufacturable. To the best of our knowledge, postponement strategies are, however, seldom used in the architecture-engineering-construction (AEC) industry. Instead, AEC practitioners typically adopt early commitment strategies that frequently result in slippage of promised project dates and extensive rework, for which construction projects are regrettably commonly known (Pietroforte 1997).

In this work, we use simulation to study the effect of postponed commitment strategies applied to the design development of semiconductor fabrication facilities (fabs). Our initial rationale was based on the intuition that, given the propensity for changes in fab design criteria, designers would be better off if they would delay tasks to the last possible moment that would still let them meet the project delivery dates.

The purpose of this paper is twofold. First, it describes the use of an event-graph simulation environment to model complex design development processes. Secondly, it illustrates a research method to explore the effects of postponed commitment strategies to design development unfolding in unpredictable environments

2 RELATED RESEARCH

Many academic studies have aimed to build theory on the nature of design processes and develop tools to help managing such processes. For instance, the Design Structure Matrix (DSM) models design tasks and respective interdependencies, assuming a sequential evolvement of the design process (Gebala and Eppinger 1991). DSM provides partitioning and tearing algorithms that order the tasks so as to minimize the information loops and the total duration of the process. DSM is, however, a static model in the sense it ignores the dynamic nature of design criteria. Jin and Levitt (1996) describe Virtual Design Team (VDT), a process-information model that implements the micro behavior of actors so as to gain insight of their influence in the performance of complex organizational systems.

Recent analytical and more abstract models of design development are closer to the work we present next. These models have yielded managerial insight on the nature of design processes unfolding in unpredictable environments. Krishnan et al. (1997), for instance, study the extent to which information exchanges between overlapped activities can be broken up to minimize project development time, if changes in preliminary information are to be expected. Wood (1998) analyses alternative development methods to deliver semiconductor facilities that can meet the needs of manufacturing firms for speed and flexibility.

3 PRODUCT-PROCESS DEVELOPMENT MODEL OF DESIGN

Figure 1 presents a generic product-process model for design development. We define design development as composed of two distinct phases: an initial conceptualization effort followed by a concept development phase.

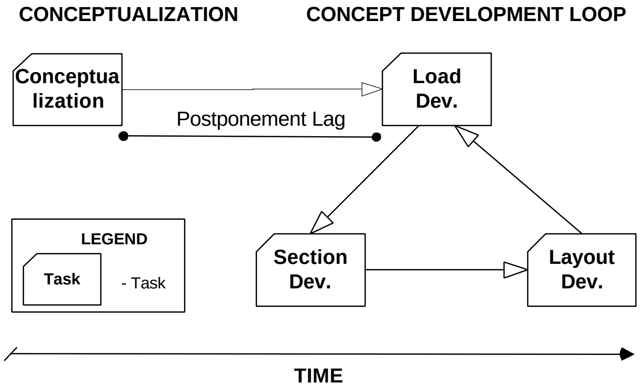

Figure 1: Design Development Model

During conceptualization, designers make a first pass of the design parameters, with the help of empirical rules and historical data. During concept development designers refine the decisions done at conceptualization in light of updated design criteria and using sophisticated analytical tools. The model expresses concept development as a loop of three tasks: load-, section-, and layout development. Load development expresses designers' effort to calculate the loads each building system should serve based on design criteria, section development expresses their effort to size the sections of the main elements in each building system based on the loads previously determined, and layout development expresses designers' effort to decide the routing of the utility systems in the tri-dimensional space and the location of major pieces of equipment.

Internal and external conditions may force designers to iterate through the aforementioned design loop. On one hand, designers may do several passes in the loop in their search for a satisfying solution if time allows (Simon 1969). Such iterations may happen even if designers possessed all the information they needed and this would not change. For simplicity, simulation assumes designers would only iterate once to find a satisfying solution, if design criteria would not change. On the other hand, task iteration may be caused by externalities such as interdependencies with other specialties or changes in design criteria. In this paper, we focus on the impacts of client-driven changes to the development process and disregard interdependencies between concurrent design processes.

4 SIMULATION

4.1 Uncertainty in Design Criteria

Uncertainty in the fab design criteria stems from diverse factors such as the concurrency of the design effort with research and development of the chip production technology the fab will house, the unknown characteristics of the production tools, and the unpredictability of market demand for the product that will be produced inside the fab. Simplistically, we assume such uncertainty affects two criteria at the core of the design process—the dimensions of the cleanroom and the list of tools to install inside.

Changes in the cleanroom dimensions are not frequent and typically result from a need to increase the fab capacity. We assume that a 20% augment of the cleanroom width and length leads AEC designers to rework conceptualization and concept development tasks. Changes in the list of process tools are more frequent than cleanroom changes. They may result from changes of the production technology or of tool suppliers. These changes may directly affect the location of tools and the utility loads that need to serve the tools. We assume each tool list change augments the design load in 20%. Such augment leads AEC designers to reiterate all concept development tasks. We neglect the impact tool list changes may have in conceptualization, given the flexibility designers have then to accommodate changes. We also assume that changes in cleanroom dimensions and tool list are stochastically independent. Such assumption can, however, be easily relaxed if other uncertainty patterns ought to be implemented.

Figure 2 represents the probability density curves of design criteria changes that we developed jointly with lead designers for research and development fabs of complex process technologies such as leading edge microprocessors and application specific integrated circuits (ASICs). We used rescaled and relocated beta random variables $[a+(b-a).beta(\alpha_1=2,\alpha_2=2)]$ to express the time variability when a change can occur.

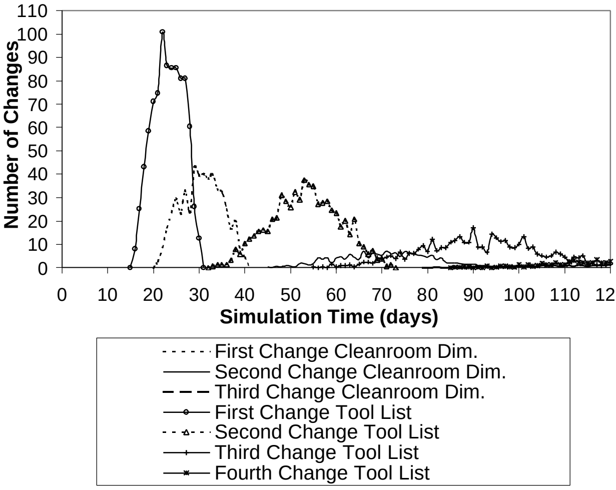

Figure 2: Histogram of Changes in Cleanroom Dimensions and Tool List for 1000 Runs

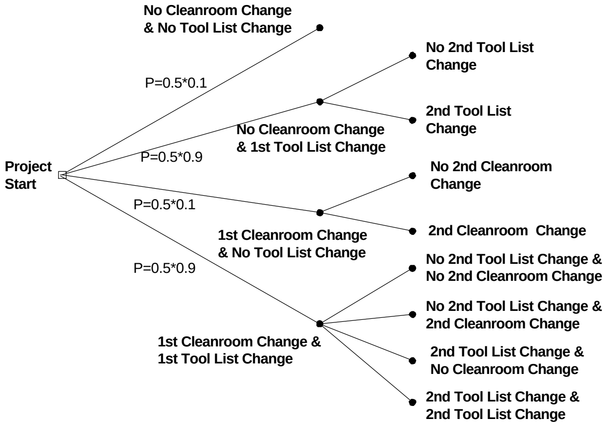

Figure 3a: Excerpt of Overall Probabilistic Tree for Cleanroom and Tool List Changes

The probabilities of first change occurrences in cleanroom dimensions and tool list are respectively 0.5 and 0.9 (Fig. 3a). The probability of occurrence of a subsequent change is conditioned to the occurrence of a previous change of the same type. We decreased the probabilities of subsequent changes by gradually dividing the probabilities of the first change in the cleanroom and tool list respectively by the terms of the sequences 1.5, 2.0, 2.5, … and 1.25,1.50, 1.75, … (Figures 3b). In addition, we gradually increased the rescaled interval of the beta distributions (b-a) between subsequent changes by multiplying them by the terms of the same sequences.

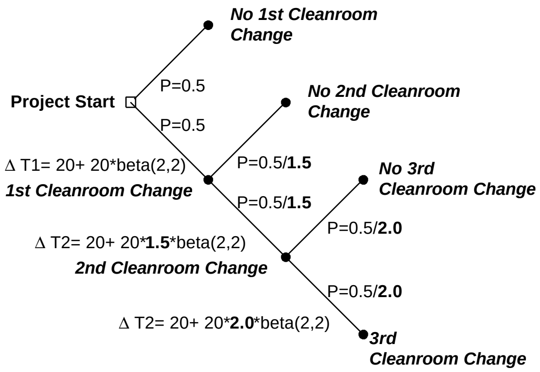

Figure 3b: Excerpt of Detailed Probabilistic Tree for Cleanroom Dimensions Change

4.2 Event Scheduling Simulation

The model was implemented with the simulation engine SIGMA (Schruben and Schruben 1999). SIGMA is a discrete-event simulation environment based on the concept of event graph. Event graphs model the systems as they evolve over time by a representation of which the state variables change instantaneously at separate points of time (Law and Kelton 2000).

Figure 4 illustrates the event graph model used in this work. The geometric figures represent events. Specifically, rectangles with a cut-off corner represent the beginning or end of design tasks, circles represent the START and END of the simulation project, and the diamonds represent Decision Points—[weekly coordination] MEETINGS and changes of design criteria (CLEANROOM CHANGE and TOOL LIST CHANGE). As each event gets executed the state variables that store the design parameters get updated according to decision rules. The arrows represent relationships between the events they connect. To each arrow is associated a set of conditions under which one event causes another event to occur (solid arrows) or to be cancelled (dashed arrows), eventually after a time delay.

At the heart of the simulation model is the use of canceling relationships between events. When a canceling relationship is executed it cancels the destination events that were previously scheduled. Accordingly, a CLEANROOM CHANGE will cancel immediately all the scheduled task events and schedule a new START CONCEPTUALIZATION. Similarly, a TOOL LIST CHANGE will cancel all the scheduled concept development tasks and schedule a new START LOAD [development]. The END event will cancel all changes scheduled to occur after day 120. The Coordination MEETING event turns the design parameter decisions into commitments. Each MEETING self schedules the next MEETING, according to a preset lag between consecutive meetings.

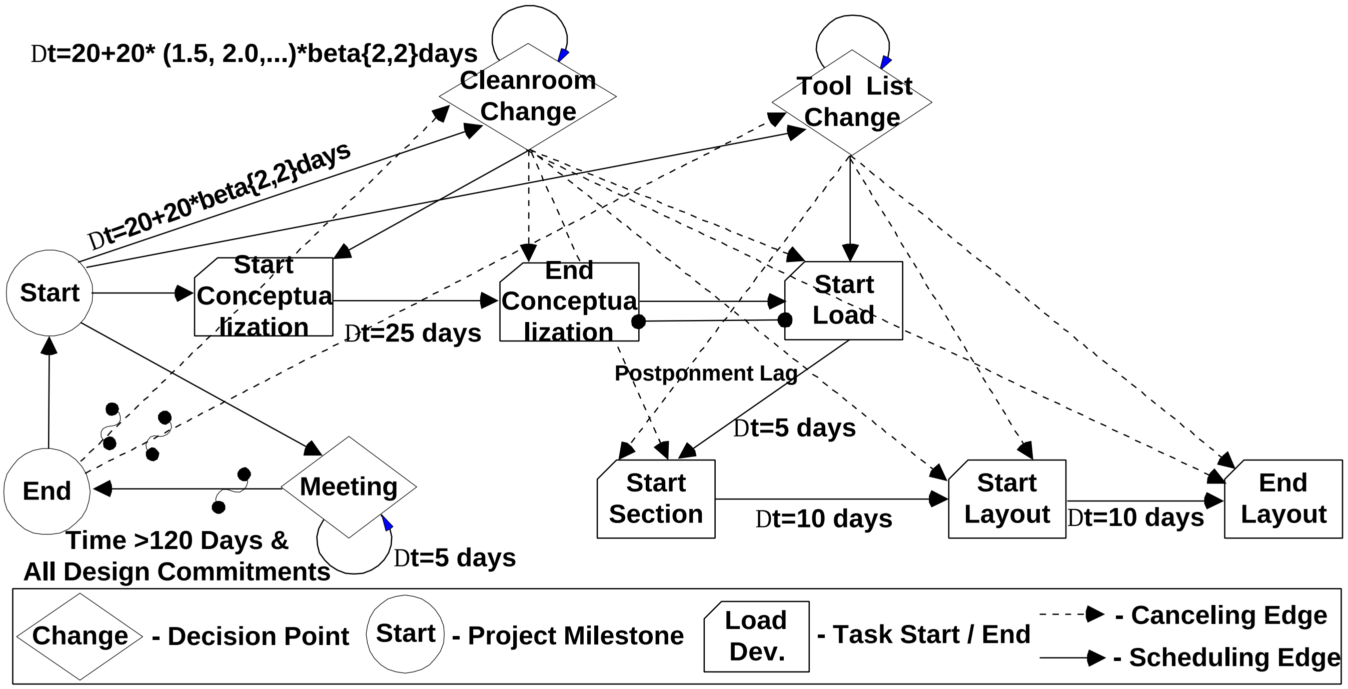

Figure 4: Event Graph Model

Once all the design commitments are made and the simulation time is above 120 days, the MEETING schedules an END event. The END event collects the values of the performance variables for the simulation run, cancels any changes that are expected to occur after day 120, and schedules a START event for a new independent simulation run.

4.3 Product Design Rationale

The product model focuses on a selected set of design parameters that define the acid-exhaust system in a fab. The acid-exhaust system is typically designed by dry-mechanical designers. The design parameters we considered are: acid-exhaust load, minimum and commercial diameter of a critical cross-section of the acid-exhaust system, and length and number of lateral routings.

Table 1 shows the rules of thumb we implemented in the model to update the design decisions in function of the design criteria changes. These are rules used in practice by dry-mechanical designers to initially estimate the design parameters. The maximum flow speed, and the width and length of the cleanroom are design criteria set in the conceptualization phase. The model assumes the maximum flow speed remains fixed. The acid-exhaust load per sq.ft and the width of subfab bays result from historical data and are also assumed fixed. Commercial diameters for acid-exhaust duct vary significantly, starting at 8 inch and increasing in intervals of 4 inch up to more than 56 inch.

Table 1 – Rules of Thumb to Estimate Design Parameters for the Acid-Exhaust System

Load	Acid exhaust load (cfm)	L	3.0 to 3.5 cfm/sq.ft of cleanroom x cleanroom width x cleanroom length
Section	Minimum diameter of upstream cross-section of acid-exhaust lateral (inch)	D	$\sqrt{\frac{L.l.4.w}{V.p}}$
	Commercial diameter of upstream cross-section of acid-exhaust lateral (inch)	D	D_c =16*(D^2 <1.78)+20*(1.78<= D^2 <2.78)+24*(2.78<= D^2 <4.0)+ 28*(4.0<= D^2 <5.44)+32*(5.44<= D^2 <7.11)+36*(7.11<=D<9.0)+40*(9.0<= D^2 <11.11)+44*(11.11<= D^2 <13.44)+48*(13.44<= D^2 <16.0)+52*(16.0<= D^2 <18.78)+56*(18.78<= D^2)
Layout	Length of lateral line (feet)	l	$\frac{WidthoftheCleanroom}{2}$
	Number of acid-exhaust laterals in the subfab	n	$\frac{2.LengthoftheCleanroom}{w}$

w – width of subfab bays (feet)
V – maximum flow speed in the lateral routing (fpm)

4.4 Simulation Rationale

The design process simulation starts with the CONCEPTUALIZATION task. The START event also stochastically schedules the first TOOL LIST CHANGE and CLEANROOM CHANGE after some time delay. When a CHANGE event occurs it then stochastically schedules a subsequent CHANGE of the same type. Once designers finish CONCEPTUALIZATION, they may opt to immediately START LOAD [development] or postpone its start date. If designers opt for a postponement strategy, they a priori decide the last possible day after which designers should start concept development. Our initial intuition for a postponement strategy was the following. Given designers common belief in the propensity of criteria to change as the design process is underway, designers would be better off in postponing the start of concept development so as to minimize rework. By the time designers would then start concept development hopefully no more changes would occur and they could develop the design in a single pass. This intuition translates in the following simulation rationale.

We assume CONCEPTUALIZATION lasts 25 days, if no changes interrupt it, in which case designers would have to iterate that effort. One extreme scenario assumes designers would START LOAD [development] immediately after the end of CONCEPTUALIZATION. This strategy means designers would START LOAD [development] on day 25 if no cleanroom changes had yet occurred or on whatever day CONCEPTUALIZATION would end, in case a change had meanwhile occurred. The other extreme scenario assumes designers would postpone START LOAD [development] up to day 110 (corresponding to a lag of 85 days if CONCEPTUALIZATION had finished on day 25) so as to maximize the probability of executing concept development in a single pass. In between, we tested alternative strategies by gradually increasing the postponed date to START LOAD [development] in intervals of 5 days, from day 25 up to day 110. For each scenario, we run 1000 independent simulations runs.

All models were run in SIGMA. SIGMA automatically generates source code in C, which can than be compiled into executable versions with Microsoft Visual C/C++ Version 6.0. One thousand iterations of the compiled version take on the order of 10 seconds, on a Pentium 600-MHz computer running Windows 98.

5 SIMPLIFYING ASSUMPTIONS

For clarity, we made some simplifying assumptions that we pass to address:

- We assumed each task has a deterministic duration, despite the fact that computer simulation lends itself to easily express a stochastic duration. From our experience in experimenting with the model, such stochastic behavior caused an insignificant difference in terms of the average results for the performance variables if compared with the results of the deterministic model. Logically though, the stochastic behavior increases the variability of the performance variables.
- We assumed learning and efficiency gains exist between consecutive iterations of CONCEPTUALIZATION. To determine the duration of CONCEPTUALIZATION in a rework cycle, we prorate its duration in the precedent cycle with the following equation: 1) if designers had concluded the task when the change occurred:

$$D_{1,n+1} = \frac{n.D_{1,n}}{n+1} = \frac{D_{1,1}}{n+1}, \forall n \qquad (1)$$

2) if the change interrupted the execution of the task:

$$D_{i+1,n} = D_{i,n} - T_{i,n} + \frac{n.T_{i,n}}{n+1} = D_{i,n} - \frac{T_{i,n}}{n+1}, \forall n, \forall i \qquad (2)$$

- ***i* , *n*** - *number of times designers have started to iterate the task (i=1,2,3,…), given a previous number of times designers have already completely executed the task (n=1,2,…)*
- $D_{i,n}$ – *expected duration for the task in iteration i given they have already completely executed the task n times, if a design change will not interrupt its execution*
- $T_{i,n}$ – *period of time the design team spent working on iteration i, given they have already completely executed the task n times.*
- $D_{1,1}$ – *expected duration for the task the first time designers will execute it, if a design change will not interrupt its execution.*

- We assumed there are no efficiency gains or learning in concept development tasks. Accordingly, the expected duration for a task in the subsequent iteration is equal to its initial expected duration, whether the change interrupted the execution of the task or the task had already been executed. Our sole purpose in opting for such algorithm was to make more prominent the results from different postponement strategies.
- We assumed availability of resources to execute the tasks, whether or not managers would decide to postpone concept development.

6 PERFORMANCE VARIABLES

To evaluate the effect of postponement on design development we defined the following performance variables:

- <u>Total project duration</u>: the period of time elapsed between START CONCEPTUALIZATION and END LAYOUT [development] for the last iteration.

- Total man-hours spent in concept development: total added time spent in iterating concept development tasks, assuming one unitary resource is allocated to each task in concept development (the reader can imagine it as if a lead designer would be the sole person executing the task).
- Number of design iterations of each task. This metric includes all iterations for each design task, regardless of its state of progression when the changes interrupted the task.

7 SIMULATION RESULTS

7.1 Design Development with Fixed Design Criteria

Fig. 5a shows the results of the design process simulation for a baseline scenario with fixed design criteria. The shape of curves in Fig. 5 reflects the deterministic duration we consider for each task, respectively 25 days for CONCEPTUALIZATION, and then 5, 10, and another 10 days respectively for LOAD, SECTION, and LAYOUT [development]. These are the average duration of these tasks for the design process of the acid-exhaust system, according to empirical research.

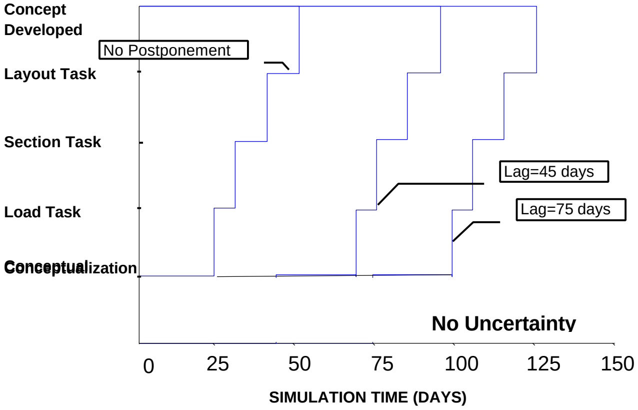

Figure 5a: Design Development Process with Fixed Criteria

Fig. 5a illustrates 3 curves, one for each following postponement strategy: 1) no postponement, 2) concept development shall not start before day 70 (corresponding thus to a postponement lag of 45 days, if conceptualization evolves in a single pass), and 3) concept development shall not start before day 100. On the (X) axis we chart the simulation time. On the (Y) axis we chart the progression of design tasks. Each specific curve connects the points corresponding to the start and finish dates of conceptualization and the three concept development tasks. In this scenario with fixed design criteria, the tasks would unfold in a sequential order and would only be executed once. A postponement strategy does not therefore bring any value in terms of resource savings. The effect of postponement is thus exclusively to delay proportionally the date of conclusion of concept development.

7.2 Design Development with Dynamic Design Criteria

As we implement the uncertainty pattern shown in Fig. 2, the design development simulation exhibits a random behavior. Each simulation run tends to evolve differently according to when changes occur and their frequency of occurrence. For each scenario, we run 1000 iterations. We then calculate the sample mean and variance with its unbiased estimators (Law and Kelton 2000).

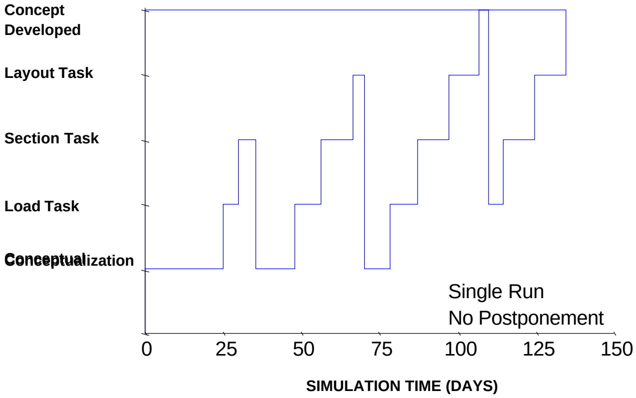

Figure 5b: Design Development Process with Dynamic Criteria (Single Run)

Fig. 5b illustrates an instance of a single simulation run from a scenario without postponement. In this specific run, the design process was interrupted three times: first by a change in the cleanroom dimensions during section development, second by a change in cleanroom dimensions during layout development, and third by a tool list change after completion of concept development. Figs. 5c illustrates the results of 50 iterations for a scenario without postponement.

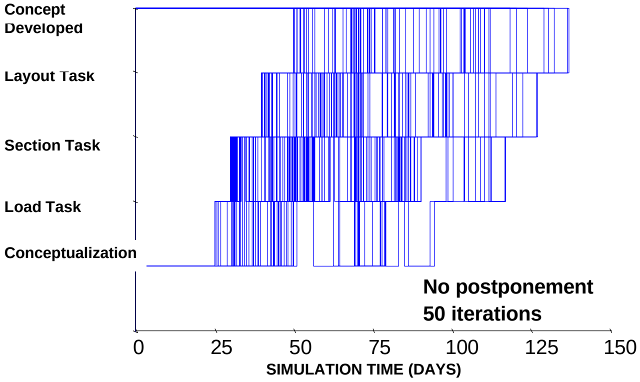

Figure 5c: Design Development Process with Dynamic Criteria (50 Runs)

Fig. 6 charts the relationship between the average overall design duration and the average total resources spent in concept development that results as we increase the postponement lag. Each data point in the chart and its respective one standard deviations in the (X) and (Y) axis were calculated with the unbiased estimators applied to the results of 1000 independent runs.

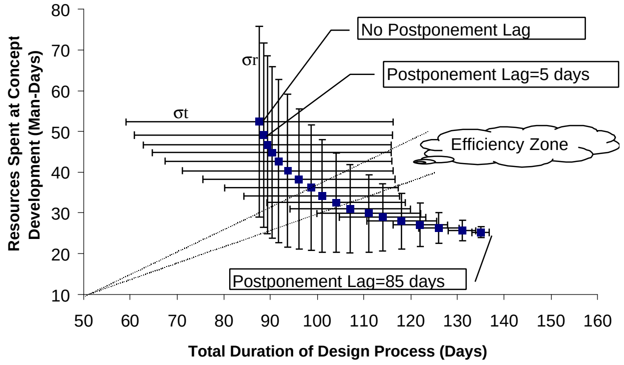

Figure 6: Total Duration of Design Process vs. Resources Spent at Concept Development in Function of Postponement Strategy

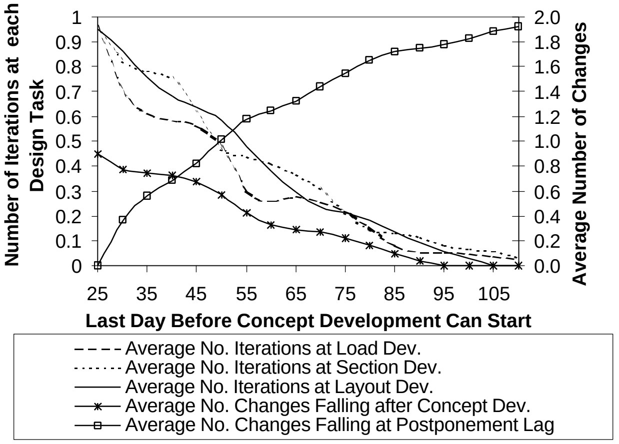

Figure 7: Variation of Mean Number of Task Iterations and Changes in Function of Postponement Strategy

Finally, Fig. 7 charts the variation of the average number of iterations for each task and changes falling at the postponement lag and after completion of concept development in function of the postponement strategy.

8 ANALYSIS AND DISCUSSION OF RESULTS

Postponement strategies are seldom used in current practice in the design development of fabs, at least to the extent we observed and discussed with practitioners during empirical research. The common argument invoked by practitioners is that if they adopted a postponement strategy they would be jeopardizing their ability to meet the project milestone dates clients impose. In other words, designers believe that every possible day of work counts in order to meet the deadline and therefore act accordingly. Designers also acknowledge that they frequently iterate several times the same tasks because of criteria changes but seem resigned to accept iteration as an intrinsic characteristic of the design process. We started this work with the intuition that many of these iterations were needless and could be prevented without compromising the project deadlines if designers would adopt a postponement strategy.

This work brought a more meaningful insight to such intuition. As Figs. 6 illustrates, a strategy of postponement consistently increases the average design duration decreasing however its variability. Postponement strategies also decrease the average resources spent in concept development and its variability. As the postponement lag increases, the marginal reduction of the spent resources is very significant, without significantly augmenting the average overall design duration. In addition, the upper bound of the overall design duration $(\mu_t+\sigma_t)$ remains approximately steady for the initial postponement lags. However, as the lag continues to increase the marginal reduction of resources spent gets less significant, and the gains in process reliability are not enough to counterbalance the steep marginal increase of the average overall design duration.

In Fig. 6, we schematically graph two rays that define in-between what we call an "*efficiency zone*" for the design process. The efficiency zone defines a set of postponement strategies that significantly decrease the variability and average of resources spent at concept development without jeopardizing the ability of designers to deliver the project before a specific milestone date, within a variability interval. In Fig. 6, the *efficiency zone* corresponds to a set of postponement strategies with a lag varying approximately between 30 to 45 days. The attentive reader may have well observed that within the efficiency zone the lower bound of resources spent $(\mu_r-\sigma_r)$ achieves its absolute lowest point.

Fig. 7 shows that, as the duration of the postponement lag increases, the average numbers of task iterations and changes falling after concept development decrease and the number of changes falling in the postponement lag increase. The graph shows, however, that the number of iterations does not decrease steadily but rather fluctuates up- and downward to zero along a trend line. Because design criteria changes occur around time-dependent means, each postponement lag shields differently the concept development tasks from the design criteria changes. Given the simplified model in this work, we foresee such variability as a vicissitude in the implementation of a postponement strategy in complex design processes since each specific lag would lead to unequal benefits across different tasks.

9 FINAL CONSIDERATIONS

Clients in the AEC industry commonly synthesize their needs with the expression "fast, cheaper, and better quality". Clients are primarily concerned to get the semiconductor fabs delivered on the milestone dates they strategically set. In addition, clients also demand process flexibility. This is, clients want the freedom to change the criteria along the design process with the simultaneous reassurance designers will still meet the milestone dates instead of invoking changes as an argument to justify delays.

Simulation results show that an early commitment strategy, though efficient for compressing average project duration, comes with some costs. One cost is the maximization of the average number of task iterations designers have to go through and of resources spent in design development. A second cost is the loss of reliability in the design development process. Results show, however, that if all else would be left equal, a thoughtful postponement strategy helps to effectively decrease design iteration and resources spent without affecting project duration within a variability interval.

10 ACKNOWLEDGEMENTS

This research was funded by grant SBR-9811052 from the National Science Foundation, whose support is gratefully acknowledged. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Financial support from the Portuguese Foundation of Science and Technology, through a scholarship awarded to Mr. Nuno Gil, is also gratefully acknowledged. Last, but not least, we owe thanks to all people interviewed, for the time and knowledge they shared with us.

11 REFERENCES

Conklin, E. J. and Weil, W. (1997). “Wicked Problems: Naming the Pain in Organizations.” Group Decisions Support Systems, Inc. Reprinted with permission on the web at <http://www.3mco.fi/meetingnetwork/readingroom/gdss_wicked.html>, Oct 98.

Crichton, C. (1966). *Interdependence and Uncertainty, A Study of the Building Industry.* Tavistock Publications Limited.

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

Gebala, D. A., and Eppinger, S. D. (1991). “Methods for Analyzing Design Procedures.” Design Theory and Methodology, DE-Vol. 31, ASME.

Iansiti, M. (1995). “Shooting the Rapids: Managing Product Development in Turbulent Environments.” California Management Review, 38 (1), Fall, 37-58.

Jin, Y., and Levitt, R.E. (1996). “The Virtual Design Team: A Computational Model of Project Organizations.” *Computational & Mathematical Organization Theory*, 2 (3), 171-196.

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

Law, A. M. and Kelton, W. D. (2000). Simulation Modeling and Analysis. The McGrw-Hill Companies, Inc., 760 pp.

Pietroforte, R. (1997). "Communication and governance in the building process." Construction Management and Economics, 15, 71-82.

Schruben, D. A. and Schruben, L. W. (1999). Event Graph Modeling using SIGMA, Custom Simulations.

Simon, H. A. (1969). “The Sciences of the Artificial.” Massach. Inst. of Tech., (3rd ed 1996).

Ward, A., Liker, J. K., Cristiano, J. J., and Sobek II, D. K. (1995). “The Second Toyota Paradox: How Delaying Decisions can Make Better Cars Faster.” Sloan Mgt. Review, Spring, 43-61.

Wood , S. C. (1997). “The Impact of Tool Delivery Times on the Optimal Capacity and Value of Semiconductor Wafer Fabs.” Proc. of the IEMT Symposium, Austin, Texas, October 13-15.

12 AUTHOR BIOGRAPHIES

NUNO GIL is a Doctoral Candidate of Construction Engineering and Mgt. in the Civil and Environmental Engineering Department at the University of California, Berkeley. His research interests are in project-based operations management, systems engineering, lean management, and concurrent engineering. His work involves planning, scheduling, and simulation. His email and web addresses are <`ngil@uclink4.berkeley.edu`> and <`http://www.ce.berkeley.edu/~nunogil`>.

IRIS D. TOMMELEIN is a Professor of Construction Engineering and Management in the Civil and Environmental Engineering Department at the University of California, Berkeley. Her research interest is in developing principles of production management for the architecture-engineering-construction industry, what is termed 'lean construction. Her work involves computer-aided design, planning, scheduling, simulation, and visualization of construction processes; wireless communication, bar coding, and laser-based positioning. Her email and web addresses are `tommelein@ce.berkeley.edu` and <`http://www.ce.berkeley.edu/~tommelein`>.
ROBERT KIRKENDALL is a Senior Designer at Industrial Design Corporation, Portland, Oregon. His email is <`robert.kirkendall@idc-ch2m.com`>.