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
Iris D. Tommelein

Constr. Engrg. & Mgmt. Program
Department of Civil and Environmental Engineering
215 McLaughlin Hall
University of California
Berkeley, CA 94720-1712, U.S.A

Robert Kirkendall

Industrial Design Corporation
2020 S.W. Fourth Avenue, 3rd Floor
Portland, OR 97201, U.S.A

ABSTRACT

This paper presents a 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, or changes 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 due to upstream changes of design criteria. Simulation illustrates the effects of adopting different postponement strategies. The results show that postponing the start of concept development consistently reduces the average resources spent in concept development and increases process reliability, but it augments the average design duration. The judicious choice of a postponement lag can thus yield gains in terms of cost versus time.

1 INTRODUCTION

Design processes are intrinsically complex in nature. Complexity stems from diverse factors, such as interdependencies and coordination needs between tasks carried out by distinct design disciplines, the iterative nature of the design process during which designers search for satisfying solutions, the criticality of compressing development time, and the unpredictability of design criteria (e.g., Crichton 1966, Simon 1969, Gebala and Eppinger 1991, Conklin and Weil 1997, Iansiti 1995, Eisenhardt and Tabrizi 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 that the 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. These frequently result in slippage of promised project schedule 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 delaying tasks to the last possible moment that would still let them meet the project delivery dates.

This paper fills two purposes. First, it describes the use of an event-graph simulation environment to model complex design development processes. Second, 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 to develop tools for managing such processes. For instance, Gebala and Eppinger (1991) used the Design Structure Matrix (DSM) to model design tasks and respective interdependencies, assuming a sequential evolution of the design process. 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) followed another approach. They developed the 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 being composed of two distinct phases: an initial conceptualization effort followed by a concept development phase.

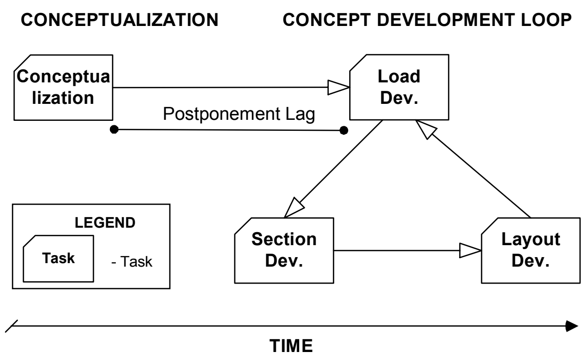

Figure 1: Design Development Model

During conceptualization, designers take a first pass at the design parameters, with the help of empirical rules and historical data. During concept development they refine the decisions made earlier during conceptualization in light of updated design criteria, using sophisticated analytical tools. The model expresses concept development as a loop of three tasks: load-, section-, and layout development. Load development expresses the 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. Layout development expresses their effort to decide the routing of the utility systems in three-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 loop in their search for a satisfying solution if time allows (Simon 1969). This may happen even when designers possess all the information they need and they know that this information would never change. For simplicity, we assume in our simulation that designers only perform the tasks once to find a satisfying solution, provided that design criteria never 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 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 that 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 cleanroom dimensions are not frequent and typically result from a need to increase the fab capacity. We assume that a 10% increase 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 are needed to serve the tools. We assume that each tool list change increases the design load by 10%. Such an increase 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 throughout that process to accommodate changes. We also assume that changes in cleanroom dimensions and tool lists are stochastically independent. This assumption can, however, be relaxed easily if other uncertainty patterns ought to be implemented.

Figures 2 represents the probability density curves of changes in cleanroom dimensions 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). Gil (2001) shows similar curves that were developed for changes in the tool list. We used rescaled and relocated beta random variables [a+(b-a).beta(α_1=2,α_2=2)] to express the time variability when a change can occur.

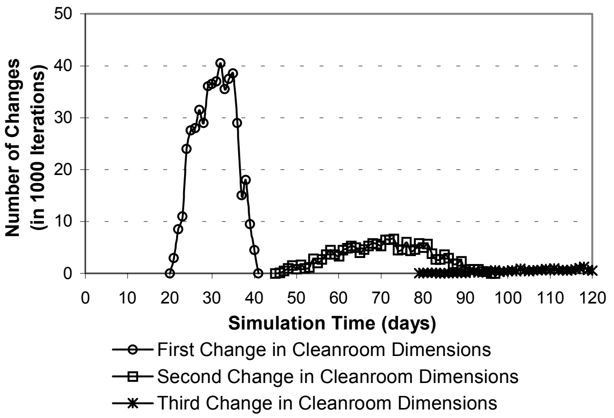

Figure 2: Histograms of Design Changes in Cleanroom Dimensions Criteria for 1000 Runs

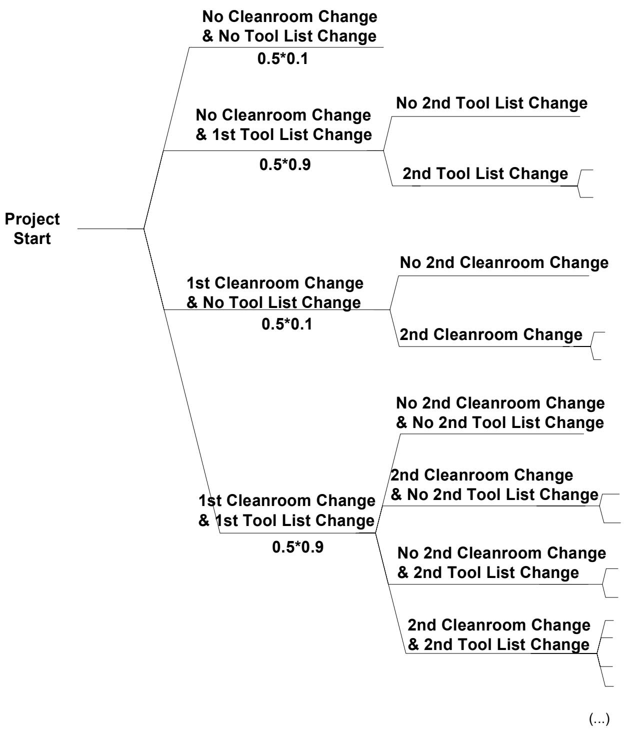

Figure 3a: Excerpt of Overall Probabilistic Tree for Cleanroom and Tool List Changes

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

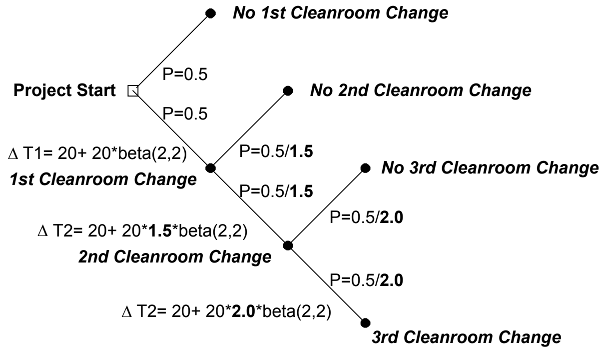

Figure 3b: Excerpt of Detailed Probabilistic Tree for Cleanroom Dimensions Change

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 'event graph' concept. An event graph models a system as it evolves over time by representing state variables that change instantaneously at discrete points in 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. Diamonds represent Decision Points—[weekly coordination] MEETINGS and changes of design criteria (CLEANROOM CHANGE and TOOL LIST CHANGE). As each event occurs, the corresponding state variables that store the design parameters get updated according to decision rules. The arrows represent relationships between the events they connect. Each arrow is associated with a set of conditions subject to which the event from which the arrow emanates schedules the event to which the arrow points after a time delay ($\Delta t \geq 0$) (solid line), or the emanating event cancels the event to which the arrow points (dashed line).

At the heart of the simulation model is the use of canceling relationships between events. When a canceling relationship is executed, selected destination events that were previously scheduled get cancelled. Accordingly, a CLEANROOM CHANGE will immediately cancel 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 Coordination MEETING event turns the design

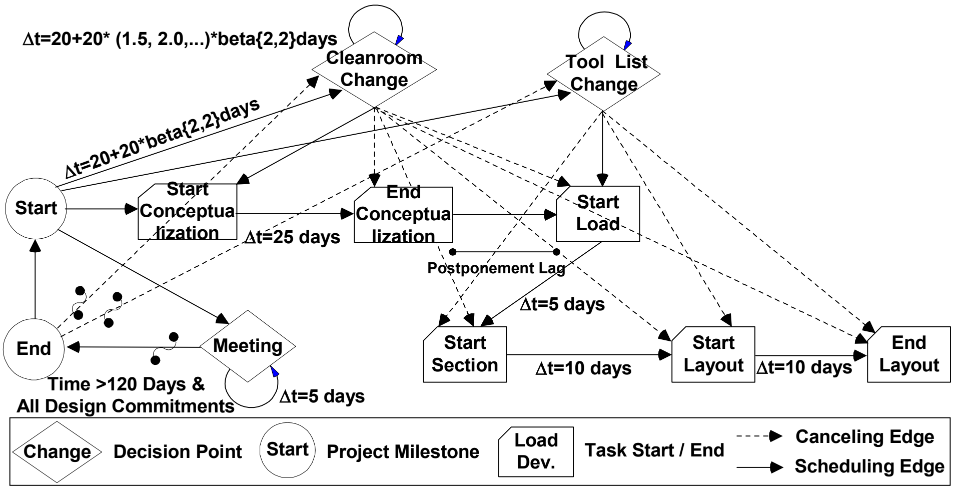

Figure 4: Event Graph Model

parameter decisions into commitments. Each MEETING self-schedules the next MEETING, according to a preset lag between consecutive meetings.

Once all the design commitments are made and the simulation time exceeds 120 days, the MEETING schedules an END event. The END event collects the values of the performance variables for the simulation run, cancels all changes scheduled to occur after day 120, that is no changes are allowed after day 120 on the condition that the design completion had already occurred (a modeling assumption further explained in section 5), and schedules a START event for a new independent run.

4.3 Product Design Rationale

We integrated a product model with the model for the design development process. The product model focuses on a selected set of design parameters that define the acid-exhaust system in a fab. The design process of the acid-exhaust system is largely representative of the process for the other 40 to 80 utility systems that may be installed in a fab. The design parameters we modeled 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. Gil (2001) presents detailed information on the design rationale of the product model.

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 stochastically schedules a subsequent CHANGE of the same type. Once designers finish CONCEPTUALIZATION, they may opt to immediately START LOAD [development] or to postpone its start date. If designers opt for the latter, they must a priori decide the last possible day by which to 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 gets underway, designers would be better off postponing the start of concept development so as to minimize rework. By the time they 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 that CONCEPTUALIZATION lasts 25 days, unless changes interrupt it, in which case designers would have to iterate that effort. One extreme scenario assumes that designers would START LOAD [development] immediately after the end of CONCEPTUALIZATION. This strategy means that designers would START LOAD [development] on day 26 if no cleanroom changes had yet occurred or on whatever day CONCEPTUALIZATION actually ended, in case one or several changes had meanwhile occurred. The other extreme scenario assumes that designers would postpone START LOAD [development] up until 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 increasing the postponed date to START LOAD [development] in intervals of 5 days, from day 25 up to day 110. For each scenario, we ran 1,000 independent simulations runs.

All models were run in SIGMA. SIGMA automatically generates source code in C, which can then be compiled into executable versions with Microsoft Visual C/C++ Version 6.0. 1,000 iterations of the compiled version take on the order of 10 seconds on a Pentium 600-MHz computer running Windows 98.

5 ASSUMPTIONS

For clarity of the model and to ease the interpretation of the results, we made the following assumptions. First, we assumed that each task has a deterministic duration, despite the fact that computer simulation lends itself to easily express stochastic durations. Given the sequential nature of the model, stochastic behavior does not influence the average results of the performance variables that were obtained with the deterministic model (a consequence of the Central Limit Theorem). Logically, though, stochastic behavior increases the variability of the performance variables.

Second, we assumed that learning and efficiency gains occur between consecutive iterations of CONCEPTUALIZATION. To determine the duration of CONCEPTUALIZATION in a rework cycle, we prorate its duration from the preceding cycle using the following equations:

1) if designers had concluded the task when the change occurred:

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

if the change interrupted the execution of the task:

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

where

i , n –number of times designers have started to perform the task (i=1,2,3,…), given a previous number of times they already completed the task (n=1,2,…)
$\mathbf{D_{i,n}}$ –expected duration for the task in iteration i given that designers already completed the task n times, and provided that a design change will not interrupt its execution
$\mathbf{T_{i,n}}$ – period of time designers spent working on iteration i, given that they already completed the task n times.
$\mathbf{D_{1,1}}$ – expected duration for the task the first time designers execute it, provided that a design change will not interrupt its execution.

We assumed that designers do not learn or otherwise gain in efficiency in concept development tasks. Accordingly, the expected duration for a task in an iteration is equal to the task's initial expected duration, whether or not the change interrupted the execution of the task or the task had already been executed. Our sole purpose in opting for such a 'dumb' algorithm was to amplify the results from different postponement strategies.

Third, we assumed that the needed resources are always available to execute any or all tasks, whether or not managers decide to postpone concept development.

Fourth, we assumed that the client would not consider any changes occurring after day 120, on the condition that the design had already been completed by that day (this is, END LAYOUT had already occurred, and no change occurred between END LAYOUT and day 120). The client would also not consider any changes occurring after design completion, if this event would take place after day 120. Changes occurring after day 120 would only be considered if by then the design was not yet completed—however, this scenario is unfeasible for the particular set of inputs in this paper because time delays between successive changes become so large by then that sooner rather than later the whole design process will end. In theory, the model could run for infinitely long time.

6 PERFORMANCE VARIABLES

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

(1) Total project duration: the period of time elapsed between START CONCEPTUALIZATION and END LAYOUT [development] for the last iteration of the latter task in the simulation run.
(2) Total man-hours spent in concept development: total added time spent in (repeatedly) performing concept development tasks, assuming one unitary resource is allocated to each task in concept development (the reader can imagine that this resource can be a lead designer, the pacesetting, or the most critical resource executing the task).
(3) Number of design iterations of each task: this performance variable includes all iterations for each design task, regardless of the design's 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 the curves in this figure 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 average durations for these tasks for the design process of the acid-exhaust system, according to empirical research.

Fig. 5a illustrates 3 curves, one for each of the following postponement strategies: (1) no postponement, (2) concept development shall not start before day 70 (thus corresponding to a postponement lag of 45 days, if conceptualization completes in a single pass), and (3) concept development shall not start before day 100. The simulation time is charted on the (X) axis. The progression of design tasks is charted on the (Y) axis. 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 sequential order and would only be executed once. A postponement strategy, therefore, does not bring any value in terms of resource savings. The effect of postponement is thus exclusively to proportionally delay the date of conclusion of concept development.

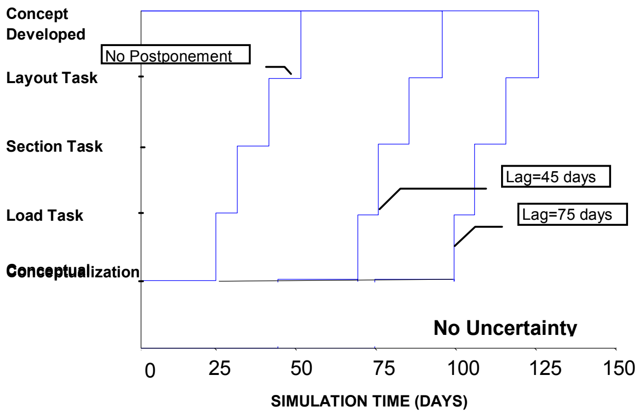

Figure 5a: Design Development Process with Fixed Criteria

7.2 Design Development with Dynamic Design Criteria

As we implement the uncertainty pattern, the design development simulation exhibits random behavior. Each simulation run tends to evolve differently according to when and how frequently changes occur. For each scenario, we ran 1,000 iterations. We then calculated the sample mean and variance with its unbiased estimators (Law and Kelton 2000).

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 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. Fig. 5c illustrates the results of 50 iterations for this same scenario.

Fig. 6 charts the relationship between the average overall design duration and the average total resources spent in concept development that results as the postponement lag increases. Each data point in the chart and its respective one standard deviation along the (X) and (Y) axis were calculated with the unbiased estimators applied to the results of 1,000 independent runs.

Finally, Fig. 7 charts the variation of the following variables in function of the postponement lag: (1) average numbers of iterations for each task, (2) average numbers of changes falling within the postponement lag, and (3) average numbers of changes falling between the end of concept development and day 120.

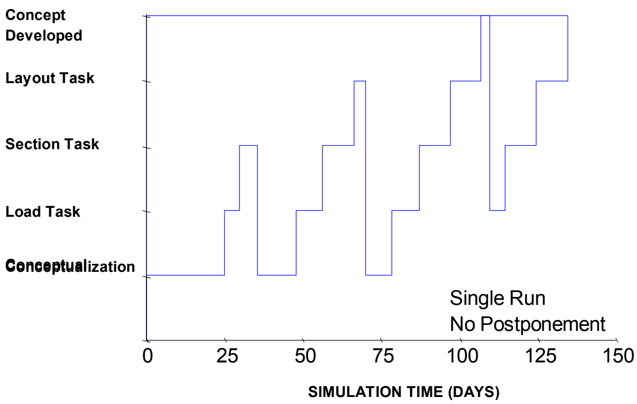

Figure 5b: Design Development Process with Dynamic Criteria (Single Run)

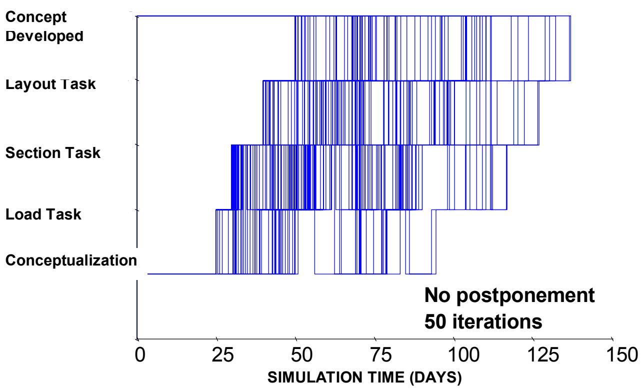

Figure 5c: Design Development Process with Dynamic Criteria (50 Runs)

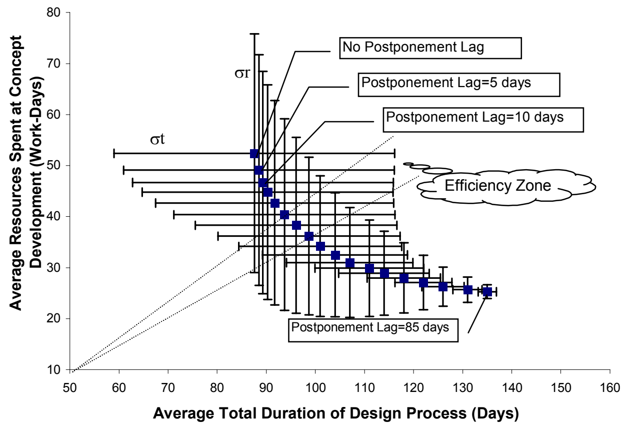

Figure 6: Total Duration of Design Process vs. Resources Spent at Concept Development in Function of Postponement Strategy

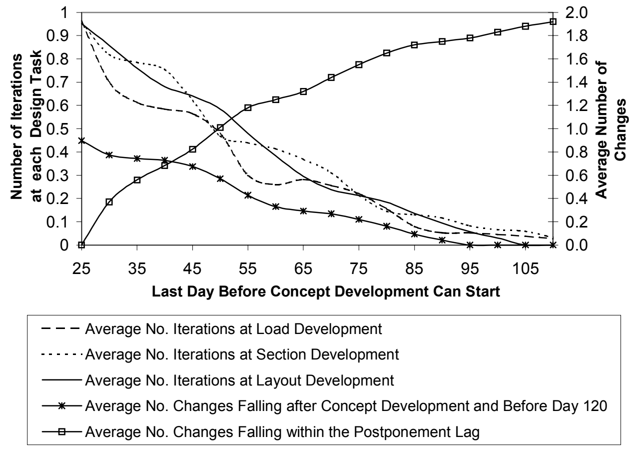

Figure 7: Variation of Mean Number of Task Iterations and Changes in Function of Postponement Strategy

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 adopting a postponement strategy would jeopardize 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 they therefore act accordingly. The opportunity costs associated with a delayed start-up of a fab are huge. Designers also acknowledge that they frequently reiterate the same tasks several times because of criteria changes. Yet, they seem resigned to accept iteration as an intrinsic characteristic of the design process. We agree that iteration is part of the exploration process so common in the search for a good design solution. Nevertheless we question whether or not all iteration is equally valuable.

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. The presented simulation work enabled us to gain more insight into this intuition. As Fig. 6 illustrates, a strategy of postponement consistently increases the average design duration while it decreases that duration's variability at the same time. Postponement strategies also decrease the average number of 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 downside risk of increasing the overall design duration ($\mu_t+\sigma_t$) remains approximately steady for small postponement lags. However, as the lag continues to increase the marginal reduction of resources spent becomes less significant, and the gains in process reliability are insufficient to counterbalance the steep marginal increase of the average overall design duration.

In Fig. 6, we schematically graphed two rays that bound what we call an "*efficiency zone*" for the design process. The efficiency zone defines a set of postponement strategies that significantly decrease the variability as well as the average number of resources spent at concept development without jeopardizing the ability of designers to deliver the project before a specific milestone date, within a predictability interval. In Fig. 6, the *efficiency zone* corresponds to a set of postponement strategies with a lag varying approximately from 25 to 35 days. The reader may have observed that within the efficiency zone the lower limit of the resources spent at concept development ($\mu_r-\sigma_r$) is close to the lowest value it can assume, another process benefit associated with this set of strategies.

Fig. 7 shows that, as the duration of the postponement lag increases, the average numbers of task iterations decrease, the average numbers of changes falling within the postponement lag increase, and the average numbers of changes falling after concept development (but before day 120, as later changes get cancelled) decrease. The graph also shows that the number of iterations for any one task does not decrease steadily but rather fluctuates up- and downward along a trend line, ultimately reaching zero. Because design criteria changes occur around time-dependent means, each postponement lag shields differently the concept development tasks from design criteria changes.

This fluctuation would have been hard to anticipate without conducting a simulation, even for a simple design process as the one we have presented here. For more complex design processes, the effect of a postponement strategy will be even more difficult to gauge, since each specific lag appears to lead to unequal benefits for the various tasks. Given the design process structure and the actual circumstances (including the durations of tasks and frequency of changes), one discipline may be forced into doing a lot of rework, even though this rework does not reflect their own skills and capabilities. One discipline may also benefit less from postponement than another, and therefore may be less eager to buy into this strategy. Design managers must be made aware of such phenomena so that they will reward team performance and not exclusively individual work.

9 FINAL CONSIDERATIONS

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

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 the project duration within a predictability interval.

10 ACKNOWLEDGMENTS

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 and from the Luso-American Foundation for Development (FLAD), through scholarships 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.

REFERENCES

Conklin, E. J. and Weil, W. (1997). “Wicked Problems: Naming the Pain in Organizations.” Group Decisions Support Systems, Inc. Reprinted with permission on <http://www.3m.com/meetingnetwork/readingroom/gdss_wicked.html, accessed on July 19,2001>.

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.

Gil, N. (2001). *Product-Process Development Simulation to Support Specialty Contractor Involvement in Early Design.* Ph. D. Diss., Civil & Envir. Engrg., Univ. of California, Berkeley.

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.” *Comp. & Math. Org. 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 McGraw-Hill Co., 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. Press (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 Management 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.

AUTHOR BIOGRAPHIES

NUNO GIL is a Doctoral Candidate of Construction Engineering and Management 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 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 address is <robert.kirkendall@idc-ch2m.com>.