

# THE IMPLICATIONS OF ALTERNATIVE MULTISCALE DESIGN METHODS FOR DESIGN PROCESS MANAGEMENT

David Shahan<sup>1</sup>, Kaarthic Madhavan<sup>1</sup> and Carolyn Conner Seepersad<sup>2</sup>

<sup>1</sup>Graduate Research Assistant, Mechanical Engineering Department, The University of Texas at Austin

<sup>2</sup>Assistant Professor, Mechanical Engineering Department, The University of Texas at Austin

## ABSTRACT

Multiscale design problems are typically decomposed into a hierarchy of distributed and strongly coupled sub-problems, each solved by design teams with specialized knowledge and tools. There are two contrasting approaches to formulating and solving such problems: (1) highly iterative exchanges of point solutions, as in most optimization-based approaches, and (2) minimally iterative exchanges of richer solutions, as in set-based approaches. The flexibility-based method is a representative set-based approach, developed by the authors, in which designers exchange targets and sets of Pareto solutions. In this paper, we explore the effect of these alternative approaches on overall lead time and lead time variability in a noisy, non-ideal design environment. Discrete event simulations are utilized to evaluate multiscale design lead time and its sensitivity to the level of noise in the design environment, including distracting interruptions and secondary design jobs. The results of the simulations indicate that the lead times of minimally iterative multiscale design processes are more robust to noisy design environments than highly iterative alternatives. Accordingly, the results suggest that it may be advantageous to favor richer, but less frequent, exchanges of information in a multiscale design process, even if more processing time is required within each sub-problem to generate those sets of information.

*Keywords: Multiscale design, multidisciplinary design, flexibility-based method, queuing networks, discrete event simulation*

## 1 INTRODUCTION

Multiscale design processes are decomposed hierarchically across a range of physical scales that can be as large as the overall system configuration and as small as material microstructure and composition. A multiscale design problem may involve not only multiple scales but also multiple disciplines. To solve such problems, a multiscale design team decomposes a system-level problem into a set of interconnected and distributed design tasks, each solved by an individual or team of designers with very specialized knowledge and tools. These tasks often cannot be automated entirely, but require design team intervention for problem formulation, concept generation and analysis, and solution validation. The challenge is to manage interdependencies between the teams and negotiate a satisfactory system-level solution. The challenge is formidable because interactions between design teams are often complex and costly iterations ensue.

Project management techniques, such as the design structure matrix and derivative tools, are available for modeling and restructuring design processes to reduce the impact of coupling and iteration on solution lead time [1, 2]. These project management techniques manipulate the task sequences and interdependencies in order to arrive at the shortest process lead time. These techniques do not look closely at how the tasks are executed and only characterize their behavior in high level terms that capture their affect on the process lead time. In contrast, design exploration techniques seek to

mathematically formalize the mechanisms at work within and between each design task in order to arrive at the highest quality solution.

Several optimization-based approaches have been proposed for solving multidisciplinary and hierarchical design problems, including analytical target cascading [3], simultaneous analysis and design [4], concurrent sub-space optimization [5, 6], collaborative optimization [7], and BLISS [8]. The difficulty with these techniques is that they require extensive iteration between design teams—one of the primary obstacles that project management tools are intended to diminish. The need for extensive iteration is driven partly by the exchange of single point solutions in optimization-based approaches, many of which are intended for fully automated execution by computers with little or no human intervention during the solution process.

Some researchers have sought to reduce iteration among design teams by exchanging richer collections of information. For example, set-based approaches have been advocated in the automotive industry [9]. Multiobjective genetic algorithms have been applied to multidisciplinary problems as a means of generating a variety of solutions for each subproblem, overcoming the convergence difficulties of gradient-based approaches, and incorporating discrete variables; however, extensive iterations between subproblems are still required [10-14]. In a different set-based approach, robust design techniques have been utilized to generate ranged sets or intervals of design specifications that can be shared with collaborating designers [15-17]. Alternatively, some authors [18] approach the issue as a negotiation problem and formalize the negotiations by using fuzzy set theory (as part of a Method for Imprecision [19]) for modeling uncertain or imprecise parameters such as preferences for performance variables. In game theoretic approaches, design teams reduce iteration by exchanging metamodels of their individual design spaces [20-24]. In recent work [25], we have proposed a flexibility-based approach in which multiscale design teams exchange targets and Pareto sets of solutions. Like other set-based approaches, the solutions tend to be *satisficing* [26] or approximate solutions that are ‘good enough’ but not necessarily optimal.

When compared with the optimization-based approaches, the set- and range-based approaches tend to reduce the number of iterations between design teams. However, the set-based approaches typically require more extensive human and computing resources (per iteration) to generate a richer set of solutions. Although satisfying the demands of a richer interface may increase processing time per iteration, it may have a substantial, positive impact on the viability of the design process and the robustness of it to variations in the design environment. For example, design processes with higher levels of iteration may be subject to more significant queuing effects in the design process as iterative jobs compete for design team attention with other secondary jobs and interruptions such as meetings and administrative work. Intuitively, one would expect these delays and lead time variability to increase with the number of iterations as the incomplete design is repeatedly passed from design team to design team. Clearly, the architecture of a multiscale design approach (e.g., optimization-based versus set-based) has an impact on the processing time per iteration and the number of iterations. *Does the architecture of a multiscale design approach also affect important metrics such as the mean lead time for the entire design process and the lead time variability? How sensitive are these metrics to the level of noise in the design environment (e.g., distracting interruptions and secondary design jobs)? In other words, is one type of multiscale design approach more robust than another and under what conditions?*

In this paper, we endeavor to answer these questions by modeling multiscale design processes using discrete event simulations. Discrete event simulation has been used to model and analyze design processes, including the effects of iteration, rework, individual task duration, and task structuring on lead time (e.g., [27-30]). In this paper, we use similar techniques to evaluate and compare the robustness of alternative multiscale design approaches in a realistically noisy design environment. This approach is similar to the processing network used by Adler, Mandelbaum, Nguyen, and Schwerer [29] to study project completion times and capture queuing effects. However, we are interested in the study of particular task structures associated with alternative multiscale design approaches, rather than just the high-level management of a complex project. Accordingly, our focus is on using project lead time prediction tools to evaluate the performance of alternative multiscale design approaches over a range of design environment conditions. In the following section, we discuss the two alternative multiscale design approaches to be compared in this paper—a highly iterative point-based approach and a minimally iterative set-based approach, patterned from the flexibility-

based method developed by the authors. In Section 3, we describe the discrete event simulation of the alternative multiscale design process architectures, followed by a discussion of the results in Section 4.

## 2 OVERVIEW OF TWO ALTERNATIVE MULTISCALE DESIGN APPROACHES

In this paper, we consider two simplified extremes: (1) a highly iterative, point-based approach, intended to represent optimization-based approaches, and (2) a minimally iterative set-based approach, intended to represent approaches similar to the flexibility-based method developed by the authors. To model these two simplified extremes, we assume a basic two-level problem with two collaborating engineers or design teams. The higher level scale is called the system and the lower level scale the subsystem. For the point-based approach, we assume that a system-level optimizer controls the design process and makes repeated requests for point solutions from the subsystem-level designer. All of the exchanges involve single values for coupled parameters. For the set-based approach, analogous to the flexibility-based method illustrated in Figure 1, the designers exchange sets of coupled parameter values. We assume that the system level designer begins the design process by exploring the system level design space and exchanging multiple targets or ranges of coupled parameter values with the subsystem. In return, the subsystem designer provides the system level engineer with a set and/or range of alternatives that meet those targets as closely as possible.

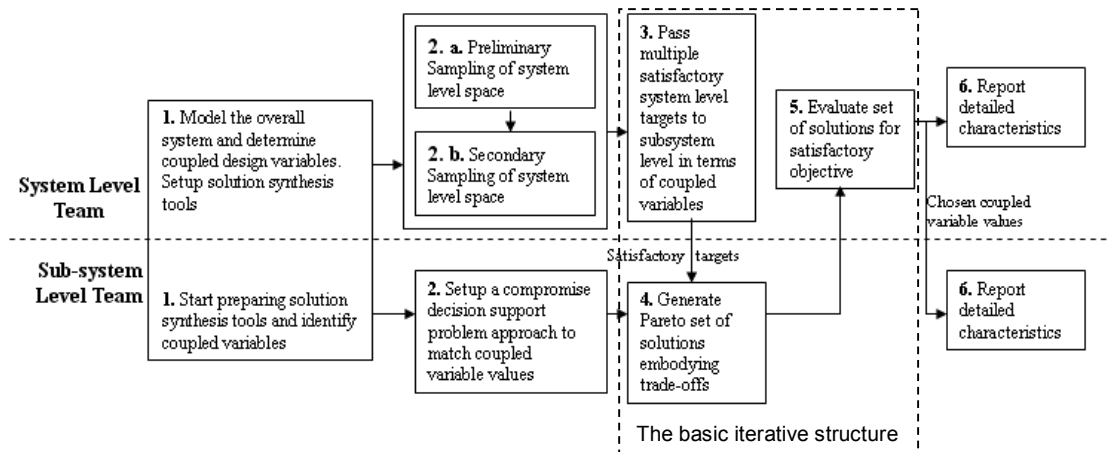


Figure 1. Schematic representation of a flexibility-based approach to multiscale design [31]

A simple representation of a collaborative multiscale design process is outlined in Figure 2. The figure highlights the total number of iterations as one of the fundamental architectural differences between alternative multiscale design approaches. Steps 3, 4, and 5 of the flexibility-based method in Figure 1 correspond to Tasks 1, 2, and 3 in Figure 2. Accordingly, a set-based method may end with Task 3. If the design method requires more iteration, as in the point-based approaches, Tasks 2 and 3 repeat as required. In the following discussion of the discrete event simulation of the design environment we refer to the entire design process as a job and the individual steps of a design process as tasks.

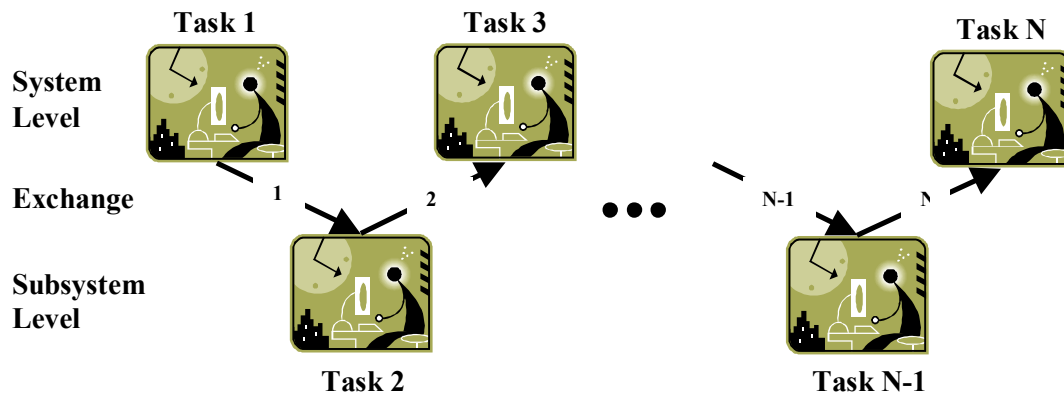


Figure 2: Flow Chart of two designers iterating in series

### 3 DESIGN ENVIRONMENT CHARACTERISTICS AND SIMULATION

In an ideal design environment, system- and subsystem-level designers complete each iteration at a very predictable rate, with little variability due to unexpected rework or setup times and with few distractions from other projects, scheduled interruptions (e.g., staff meetings) or unscheduled disruptions (e.g., illnesses). However, a realistic design environment is very noisy. For example, one essential feature of an actual design environment that complicates the process in Figure 2 is competition for design resources from other jobs. In its simplest form, as illustrated in Figure 3, this competition can be represented by *secondary* jobs being executed by both the system level and the subsystem level. These secondary jobs could represent for example the day to day responsibilities of the engineers such as attending to manufacturing, assembly or field issues. As primary (i.e., multiscale) and secondary jobs arrive, they enter each designer's inbox. A queue develops if the arrival rate of the jobs is faster than the time required to process them. We assume that a designer will not maintain an infinite queue with the hopes of returning to all of the jobs that arrive. Rather the queue will grow to a maximum size and then stop growing because jobs will be routed to other designers or cancelled. In a work environment, a designer may allow some requests to accumulate in a queue and handle other requests immediately. These latter requests are called *interruptions*. Interruptions to work extend the mean process time and increase the process time standard deviation of the interrupted job [32]. Interruptions could represent, for example, consultations for other jobs, training, staff meetings and communications, etc. Finally, another complication of an actual design environment is the variability in the arrival times of all of these jobs and the variability in the duration of each job task. Accordingly, probability distributions are assigned to the arrival rates for each job and to the process times for each job task.

The inputs, outputs and noise factors for simulating a multiscale design environment are illustrated in a P-diagram in Figure 4. From the perspective of designing the design process, input factors include the number of iterations for the multiscale design job and the mean process time for each task. The approach taken in this paper is to couple the mean process time of the multiscale job tasks with the number of iterations by choosing a constant lead time for the entire multiscale job under deterministic conditions and dividing it equally among the subtasks' mean process times. For example, if the total lead time for a multiscale job is 45 hours and it is executed in a single global iteration (system to subsystem to system), then the mean process time for each task will be 15 hours (i.e. 45/3). If it is completed in 10 global iterations, then each task will require 4.5 hours (i.e., 45/10). This assumption allows us to look exclusively at the effect of iterations on lead time. The output factors include job lead time mean and standard deviation. The remaining factors are environmental noise factors that include the variability in multiscale task process times, the mean and coefficients of variation of process times for secondary and interruption jobs, and the arrival rates for each of the jobs.

A discrete event simulation framework was created to capture the complexities of the design environment and their effect on the lead time of multiscale design jobs. The simulation has the following structure:

1. Initialize the simulation with job arrival times and task process times for both designers. Set the time equal to zero. Enter the simulation main loop.
2. Determine the smallest time to the next event(s), subtract this time from the remaining times for all other events, and increment the simulation time.
3. Update the state of each designer and their queues according to which event(s) took place.
4. Repeat steps 2 and 3 until the collaborative design job has been completed and its lead time can be calculated.

There are some subtleties to consider. First, the simulation is stochastic in order to capture the variation. All the jobs have arrival time means and coefficients of variation that are sampled from lognormal probability distributions. Similarly, all the tasks have processing time means and standard deviations that are also sampled from lognormal probability distributions. The metric of interest, the total multiscale job lead time, will consequently have a mean and an associated distribution. Since the goal is to study the lead time of the multiscale design job, the simulation starts without any multiscale jobs and runs for some time in order to approach a steady state environment at which point we introduce the multiscale design job. The multiscale design job is executed with as many iterations as desired. The simulation ends when the multiscale design job ends. The simulation is repeated in batches of one hundred simulations and the mean and standard deviation of the multiscale design job's lead time is calculated. The batch runs are terminated after changes in the cumulative mean and standard deviation converge to within a specified convergence limit [27]. Using this procedure, we calculate lead times as a function of arrival rates of secondary jobs and numbers of multiscale design iterations. In the next section, we report the results of a series of experimental simulations.

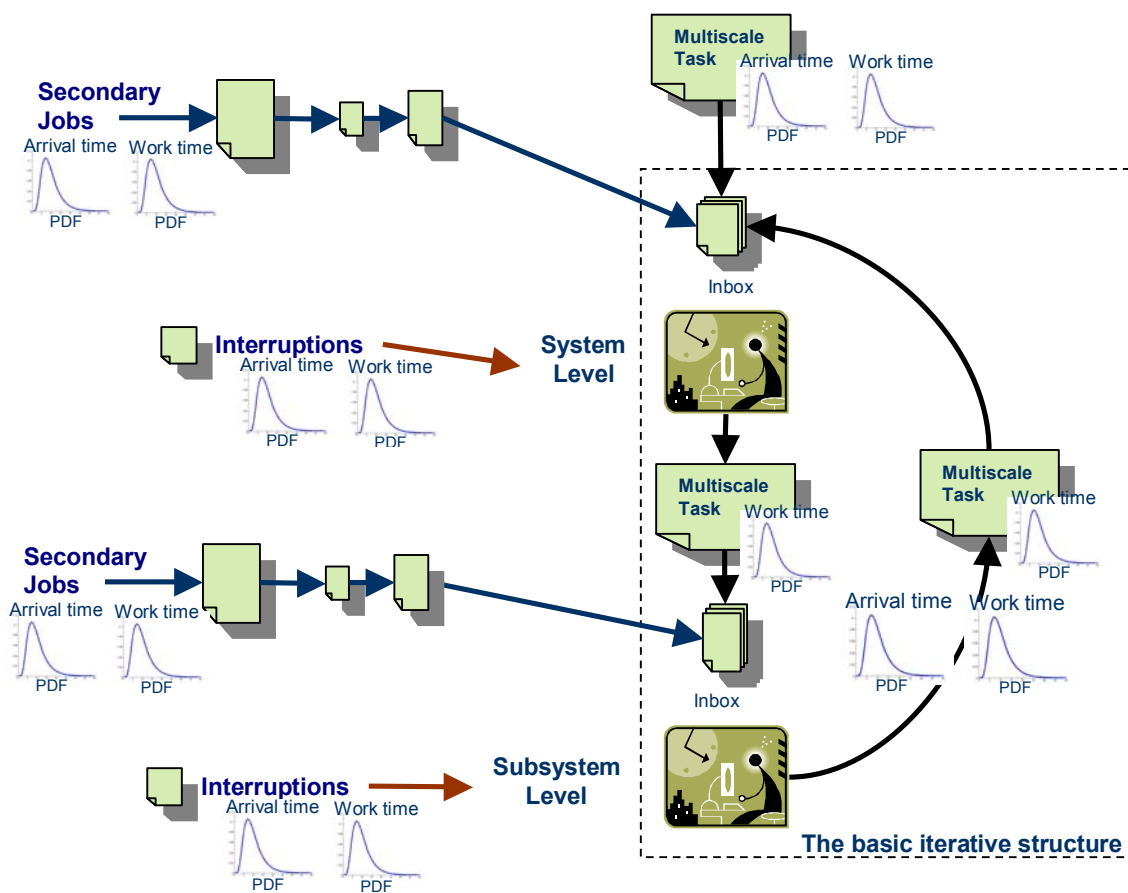


Figure 3: Flow Chart of the design environment

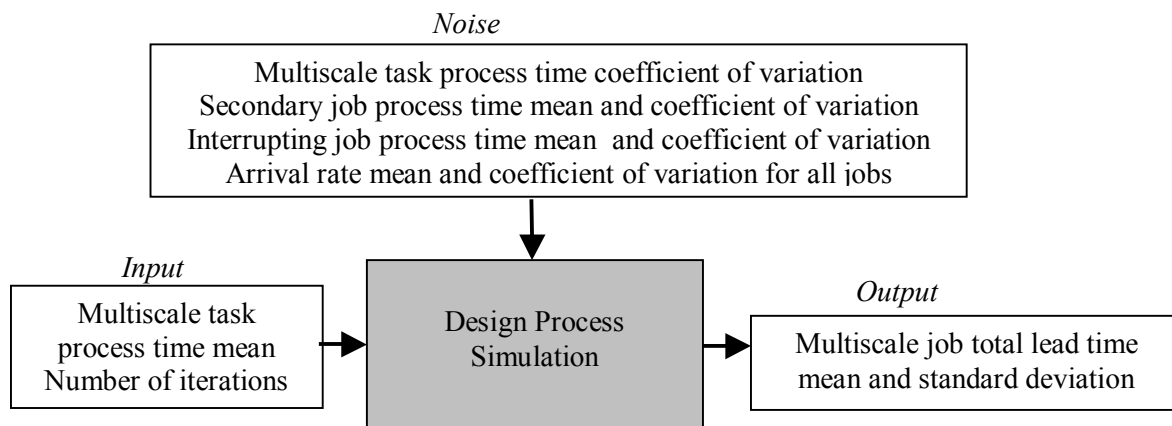


Figure 4: P-diagram representation of the design process simulation

## 4 RESULTS

The discrete event simulation is utilized to investigate three different multiscale design scenarios. In the first scenario, the emphasis is on the effect of secondary job arrival rates on the total lead time for the multiscale design job. In the second scenario, the emphasis is on the effect of increased variation in secondary job arrival rates and task processing times on total lead time. In the third scenario, the emphasis is on the effect of interruptions on the total lead time.

The input parameters for the first scenario are documented in Table 1. The total lead time under deterministic conditions is set at 45 hours, and the mean process times for the multiscale job subtasks are expressed as the appropriate fraction of the total lead time to enforce this constraint. Time is expressed in hours, although the choice of units is arbitrary.

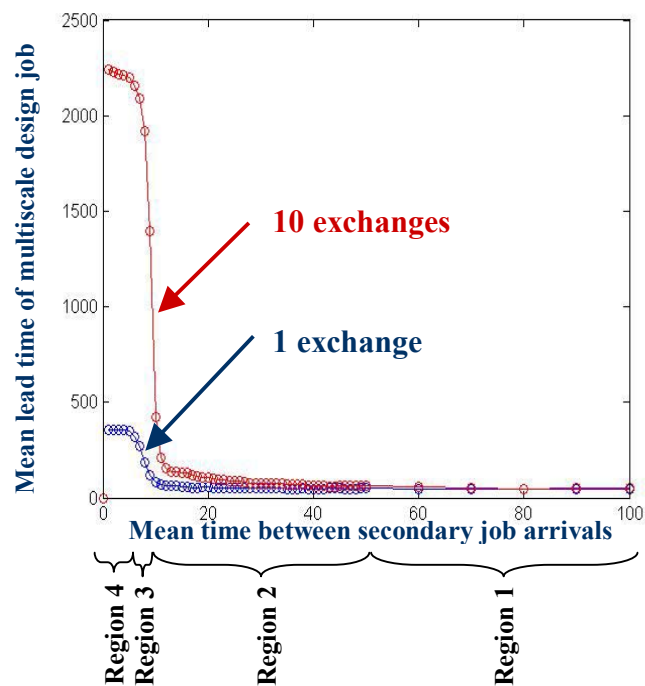
The results of the discrete event simulation with these input parameters are illustrated in Figures 5 and 6. In Figure 5, the mean lead time for a multiscale design job is plotted versus the arrival rates of secondary jobs. Results are plotted for a highly iterative (point-based) multiscale job with 10 global iterations and a minimally iterative (set-based) multiscale job with 1 global iteration. The results can be segmented into 4 regions, as indicated in Figures 5 and 6. For Region 1 with mean arrival times greater than roughly 45 hours, the secondary jobs do not interfere with the execution of the multiscale design job enough to change the lead time for different exchange counts. For this region, the mean multiscale design lead time is 45 hours as expected. For mean arrival rates of roughly 10 hours to 45 hours, in Region 2, the jobs begin to be affected appreciably by waiting times in queues. The more exchanges there are, the more waiting time in queues, and the bigger the effect on lead time. Region 3 is a jump transition that occurs around mean arrival rates of 10 hours. This is where the queues begin to grow rapidly until they saturate at the specified limit of 10 jobs. In region 4 we find the maximum lead time for the collaborative job. In this region, for the single exchange multiscale job there will be three separate times in queue with 10 jobs ahead of it for a total of about 300 hours of waiting in queues. For the multiscale job with ten exchanges there will be about 2100 hours of waiting in queues. Since the average lead times are close to these values, this gives a good indication that the simulation results are valid. Most designers seem to spend their time operating in regions 2, 3 and 4. As designers increase in value to the company, they spend more of their time in region 4. In this region, a flexibility-based method is most valuable.

The lead time standard deviation provides a measure of the lead time variation. The impact of secondary job arrival rates and multiscale design job iterations on lead time standard deviation is illustrated in Figure 6. In general, the variation in the lead times increases with the amount of interaction with other stochastic jobs. Since our metric is time, there is a cumulative increase in variation as time progresses, and each task interacts in a historical way with all of the events that occur after the multiscale job's arrival. In region 3 there is a lot of variation in queue lengths because the system is transitioning to a less variable state of full queues and is affected by variation in conditions when the multiscale job enters the system. In region 4 the queue lengths saturate fast enough to

produce more consistency, and hence the variation reduces significantly from the spike in variation seen in region 3. Multiscale jobs with fewer exchanges lead to more robust design processes (with lower standard deviations) under these conditions.

*Table 1. Simulation Parameters*

Secondary job mean arrival time for each designer	Varied from 1 to 100
Secondary job arrival time coefficient of variation for each designer	0.1
Secondary job mean process time for each designer	10
Secondary job process time coefficient of variation for each designer	0.1
Multiscale job mean arrival time to system level designer	100
Multiscale job arrival time coefficient of variation for the system level designer	0.1
Multiscale job mean process time for each designer, all tasks	$45/(2 \cdot \text{iterations} + 1)$
Multiscale job process time coefficient of variation for each designer, all tasks	0.1



*Figure 5: Mean Lead Times for the First Simulation with Coefficients of Variation = 0.1*

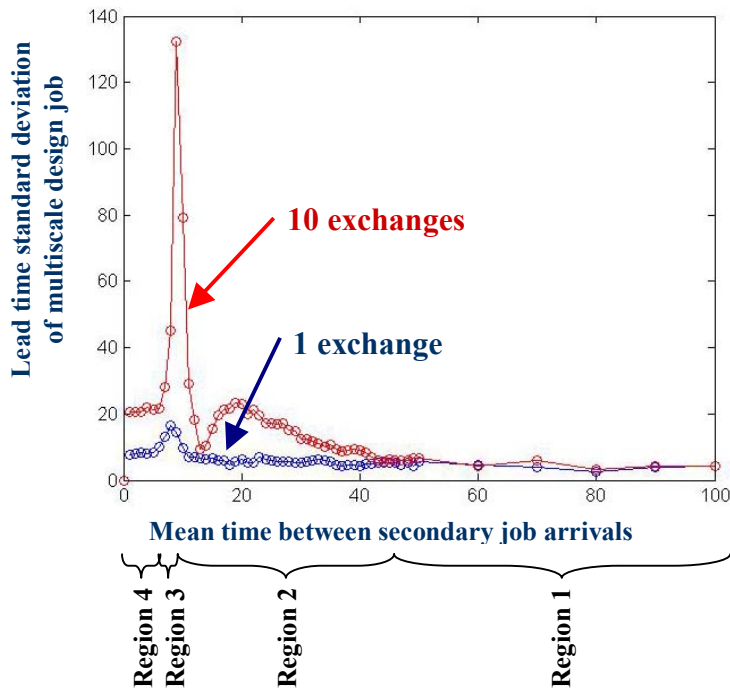


Figure 6: Lead Time Standard Deviations for the First Simulation with Coefficients of Variation = 0.1

For the second simulation scenario, the procedure was repeated with the changes shown in Table 2 to investigate the effect of more variation in the system. Figures 7 and 8 show the new results. The results in Figures 7 and 8 are very similar to the previous results. The transitions are smoother between each region, which means that the job with ten exchanges fares even worse in region 2 with this increase in variation. The tremendous amount of variation in regions 3 and 4 for the ten exchange job compared to the relatively stable, single exchange job provides further support for the conclusion that the single exchange job is more robust in terms of lead time variation. The results obtained for this simulation are very similar to results obtained when using exponential distributions for all of the process parameters.

Table 2. New Simulation Parameters

Secondary job arrival time coefficient of variation for each designer	1
Secondary job process time coefficient of variation for each designer	1

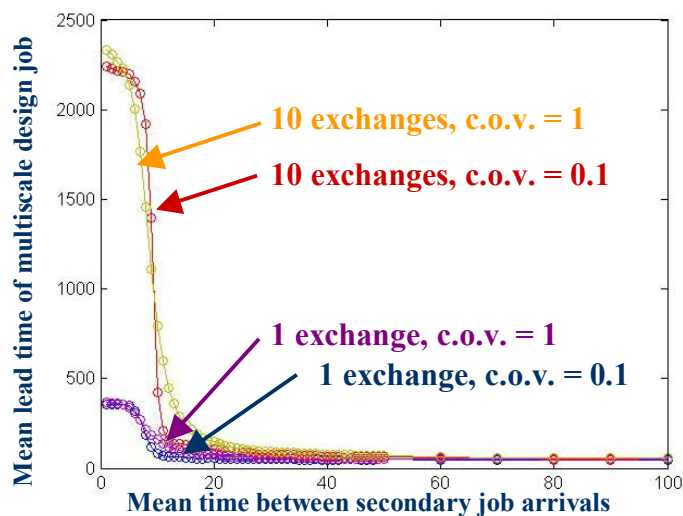


Figure 7: Mean Lead Times for the First Simulation with Coefficients of Variation = 0.1 Versus the Second Simulation with Coefficients of Variation = 1

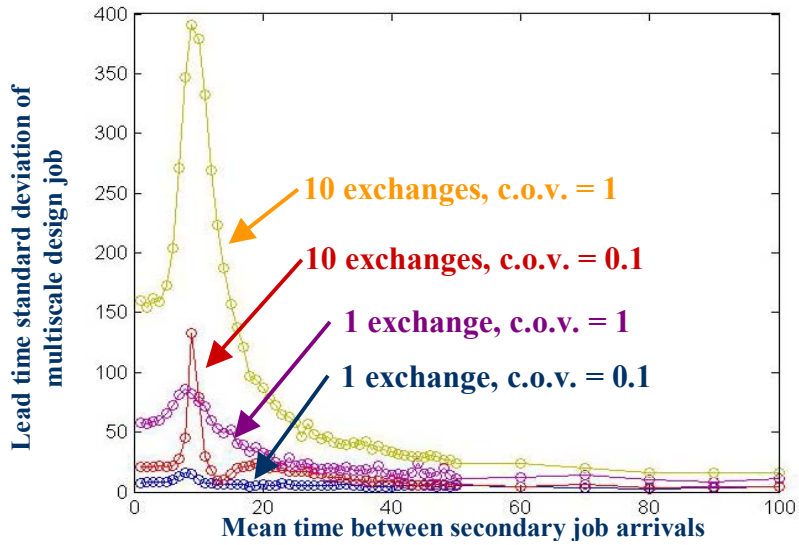


Figure 8: Lead Time Standard Deviations for the First Simulation with Coefficients of Variation = 0.1 Versus the Second Simulation with Coefficients of Variation = 1

For the third simulation scenario, we introduce the effects of interruptions. Accordingly, we repeat the simulation with the parameter values listed in Table 1, but with the changes indicated in Table 3. As illustrated in Figures 9 and 10, interruptions in general tend to increase both the mean and the standard deviation of the lead time. The increase is noticeable even though variation in the interruption arrival and processing times is small. Notice that the effect of interruptions on the mean lead time is even more adverse in the busy design regions for the 10-iteration job compared to the single-iteration job. This effect can be conceptualized in terms of a percent increase in lead time having a larger effect on larger lead times. Again, the job with one exchange proves to be more robust to noisy interruptions than the job with ten exchanges.

Table 3. New Simulation Parameters

Secondary job arrival time coefficient of variation for each designer	0.1
Secondary job process time coefficient of variation for each designer	0.1
Interruption job mean arrival time for each designer	5
Interruption job arrival time coefficient of variation for each designer	0.1
Interruption job mean process time for each designer	0.5
Interruption job process time coefficient of variation for each designer	0.1

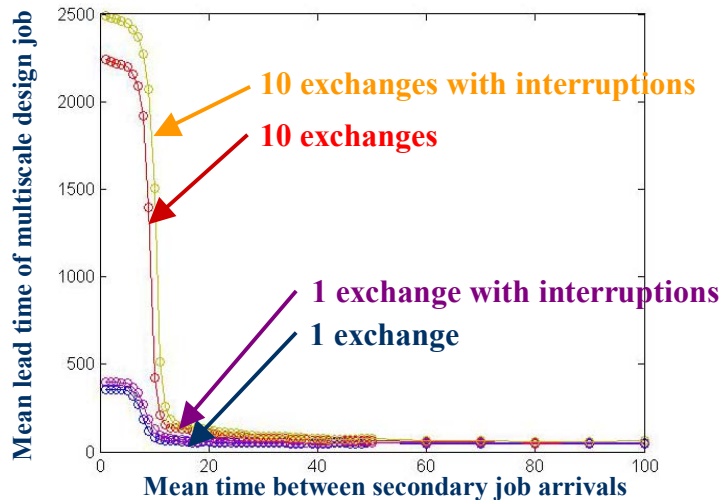


Figure 9: Mean Lead Times for the First Simulation without Interruptions Versus the Third Simulation with Interruptions

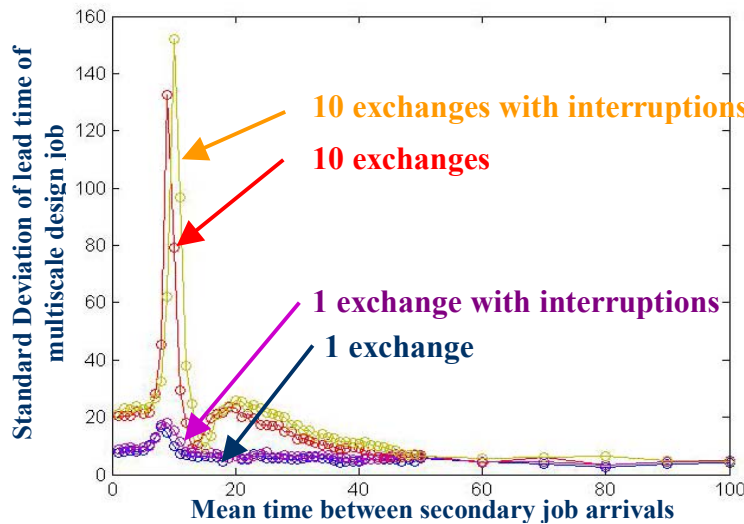


Figure 10: Lead Time Standard Deviations for the First Simulation without Interruptions Versus the Second Simulation with Interruptions

As an illustration of the need to consider the number of global iterations when designing a collaborative design process, consider the following example. A two-level, micro unmanned aerial vehicle design process was solved in our research laboratory using typical optimization based algorithms (a genetic algorithm followed by sequential quadratic programming) and a flexibility-based design process [31]. The flexibility-based design method required 1 global iteration (versus 1660 iterations for the optimization-based design process) and approximately 50 times more computing time per subtask, to obtain a solution within 5% of the Pareto optimal curve. Assuming mean process times of 1 hour per subtask of the optimization-based process and 50 hours per subtask of the flexibility-based design method, a value of 15 for the mean arrival time of secondary design jobs and all other parameters as recorded in Table 1, we can expect the highly iterative optimization-based design process to exhibit a lead time 83 times longer than the flexibility-based design method. The standard deviation of the lead time also increases by 3100%. More detailed simulations are needed with multiple design teams and more accurate process times, but the results suggest that a minimally iterative, set-based approach is a more viable alternative for a realistic design environment.

## 5 CLOSURE

The results of our simulations support the conclusion that minimally iterative multiscale design processes are more robust than highly iterative approaches in noisy design environments. Robustness is measured in terms of the decreased mean and standard deviation of total lead times for minimally iterative approaches. Under ideal, deterministic conditions the two extreme approaches are designed to deliver results with identical lead times. The mean and standard deviation of the overall lead time increase dramatically for highly iterative processes under noisy conditions. Noise factors here include the presence of secondary jobs and interruptions with durations of specified mean and coefficients of variation. The results suggest that it may be advantageous to favor richer, but less frequent, exchanges of information in a multiscale design process, even if more processing time is required per iteration within each sub-problem to generate those sets of information.

There are many opportunities for extending this line of research and relaxing some of the assumptions underlying the present work. The simulation presented in this paper is limited to a simple sequential back-and-forth iteration between two designers at two scales and could readily be extended to consider more complex collaborative architectures with larger numbers of sub-problems and more complex interactions such as overlapping and parallel work flows. A particularly intriguing direction for further work would incorporate alternative process flows as determined by task dependant decisions, further integrating the flexibility-based design problem formulation with process management planning. In the present paper only two types of design environment effects have been considered: secondary jobs and interrupting jobs. The simulation could be expanded to include other types of effects such as set up times, communication times, learning curves, rework, and more. The presented simulation uses lognormal probability distributions and other distributions should be investigated. Other assumed process parameters should be explored such as the mean arrival time of the flexibility-based design job. The simulations could be useful for comparative studies of alternative multiscale design approaches, executed on common test problems. In such applications, the simulations could provide estimates of lead time and lead time robustness with respect to noise in the design environment; these metrics could complement typical metrics such as total number of iterations, computing time, and solution quality. Finally, the simulations could be coupled with data from actual design problems and processes to calibrate and validate the results.

## REFERENCES

- 1 **Steward, D.V.** *Systems Analysis and Management: Structure, Strategy, and Design.* (Petrocelli Books, New York, 1981).
- 2 **Eppinger, S.D., Whitney, D.E., Smith, R.P. and Gebala, D.A.** A Model-Based Method for Organizing Tasks in Product Development. *Research in Engineering Design*, 1994, **6**, 1-13.
- 3 **Kim, H.M., Michelena, N.F., Papalambros, P.Y. and Jiang, T.** Target Cascading in Optimal System Design. *ASME Journal of Mechanical Design*, 2003, **125**, 474-480.
- 4 **Haftka, R.T.** Simultaneous Analysis and Design. *AIAA Journal*, 1985, **23**(7), 1099-1103.
- 5 **Sobieszczanski-Sobieski, J.** Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems. *Second NASA/Air Force Symposium on Recent Advances in Multidisciplinary Analysis and Optimization* (NASA TM-101494. NASA CP-3031., Hampton, VA, 1988).
- 6 **Wujek, B.A., Renaud, J.E., Batill, S.M. and Brockman, J.B.** Concurrent Subspace Optimization Using Design Variable Sharing in a Distributed Computing Environment. *Concurrent Engineering: Research and Applications*, 1996, **4**(4), 361-377.
- 7 **Kroo, I., Altus, S., Braun, R., Gage, P. and Sobieski, I.** Multidisciplinary Optimization Methods for Aircraft Preliminary Design. *5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* (Paper No. AIAA-94-4325. Vol. 1, pp.697-707, Panama City, FL, 1994).
- 8 **Sobieszczanski-Sobieski, J. and Kodiyalam, S.** BLISS/S: A New Method for Two-Level Structural Optimization. *Structural and Multidisciplinary Optimization*, 2001, **21**(1), 1-13.
- 9 **Sobek, D.K., Ward, A. and Liker, J.K.** Toyota's Principles of Set-Based Concurrent Engineering. *Sloan Management Review*, 1999, 67-83.
- 10 **Hajela, P.** Nongradient Methods in Multidisciplinary Design Optimization--Status and Potential. *Journal of Aircraft*, 1999, **36**(1), 255-265.
- 11 **Rowell, L.F., Braun, R.D., Olds, J.R. and Unal, R.** Multidisciplinary Conceptual Design Optimization of Space Transportation Systems. *Journal of Aircraft*, 1999, **36**(1), 218-226.

- 12 **Gunawan, S., Farhang-Mehr, A. and Azarm, S.** On Maximizing Solution Diversity in a Multiobjective Multidisciplinary Genetic Algorithm for Design Optimization. *Mechanics Based Design of Structures and Machines*, 2004, **32**(4), 491-514.
- 13 **Nair, P.B. and Keane, A.J.** Coevolutionary Architecture for Distributed Optimization of Complex Coupled Systems. *AIAA Journal*, 2002, **40**(7), 1434-1443.
- 14 **Altus, S.S., Kroo, I.M. and Gage, P.J.** A Genetic Algorithm for Scheduling and Decomposition of Multidisciplinary Design Problems. *ASME Journal of Mechanical Design*, 1996, **118**(4), 486-489.
- 15 **Chen, W. and Lewis, K.** A Robust Design Approach for Achieving Flexibility in Multidisciplinary Design. *AIAA Journal*, 1999, **37**(8), 982-989.
- 16 **Chang, T.S., Ward, A.C., Lee, J. and Jacox, E.H.** Conceptual Robustness in Simultaneous Engineering: An Extension of Taguchi's Parameter Design. *Research in Engineering Design*, 1994, **6**, 211-222.
- 17 **Kalsi, M., Hacker, K. and Lewis, K.** A Comprehensive Robust Design Approach for Decision Trade-Offs in Complex Systems Design. *ASME Journal of Mechanical Design*, 2001, **123**(1), 1-10.
- 18 **Scott, M.J. and Antonsson, E.K.** Formalisms for Negotiation in Engineering Design. *ASME DETC/CIE Design Theory and Methodology Conference* (Paper Number: DETC96/DTM1525, 1996).
- 19 **Antonsson, E.K. and Otto, K.N.** Imprecision in Engineering Design. *Journal of Mechanical Design*, 1995, **117**(2), 25-32.
- 20 **Lewis, K. and Mistree, F.** Collaborative, Sequential and Isolated Decisions in Design. *ASME Journal of Mechanical Design*, 1998, **120**(4), 643-652.
- 21 **Hernandez, G., Seepersad, C.C. and Mistree, F.** Designing for Maintenance: A Game Theoretic Approach. *Engineering Optimization*, 2002, **34**(6), 561-577.
- 22 **Xiao, A., Zeng, S., Allen, J.K., Rosen, D.W. and Mistree, F.** Collaborative Multidisciplinary Decision Making Using Game Theory and Design Capability Indices. *Research in Engineering Design*, 2005, **16**(1-2), 57-72.
- 23 **Fernandez, M.G.** A Framework for Agile Collaboration in Engineering. *G.W. Woodruff School of Mechanical Engineering* (Georgia Institute of Technology, Atlanta, GA, 2006).
- 24 **Panchal, J.H., Fernandez, M.G., Allen, J.K., Paredis, C.J.J. and Mistree, F.** An Interval-Based Focalization Method for Decision-Making in Decentralized, Multi-Functional Design. *ASME IDETC/CIE Advances in Design Automation Conference* (Paper Number: DETC2005-85322, Long Beach, CA, 2005).
- 25 **Carlos, S., Madhavan, K., Gupta, G., Keese, D.A., Maheshwaraa, U. and Seepersad, C.C.** A Flexibility-Based Approach to Collaboration in Multiscale Design. *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* (AIAA-2006-7063, Portsmouth, VA, 2006).
- 26 **Simon, H.A.** *The Sciences of the Artificial*. (MIT Press, Cambridge, Mass., 1996).
- 27 **Browning, T.R. and Eppinger, S.D.** Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development. *IEEE Transactions on Engineering Management*, 2002, **49**(4), 428-442.
- 28 **Cho, S.-H. and Eppinger, S.D.** A Simulation-Based Process Model for Managing Complex Design Projects. *IEEE Transactions on Engineering Management*, 2005, **52**(3), 316-328.
- 29 **Adler, P.S., Mandelbaum, A., Nguyen, V. and Schwerer, E.** From Project to Process Management: An Empirically-Based Framework for Analyzing Product Development Time. *Management Science*, 1995, **41**(3), 458-484.
- 30 **Jun, H.-B., Ahn, H.-S. and Suh, H.-W.** On Identifying and Estimating the Cycle Time of Product Development Process. *IEEE Transactions on Engineering Management*, 2005, **52**(3), 336-349.
- 31 **Madhavan, K.** A Framework for a Flexibility-based Approach to Multiscale and Multidisciplinary Design. *Mechanical Engineering* (University of Texas at Austin, Austin, 2007).
- 32 **Hopp, W.J. and Spearman, M.L.** *Factory Physics: Foundations of Manufacturing Management*. (McGraw-Hill Higher Education, New York, NY, 2001).

Contact: C.C. Seepersad  
The University of Texas at Austin  
Mechanical Engineering Department  
1 University Station C2200  
Austin, TX 78712-0292 USA  
Phone: 512-471-1985  
Fax: 512-471-7682  
Email: ccseepersad@mail.utexas.edu  
Website: <http://www.me.utexas.edu/~ppmndlab/>