Proceedings of ICAD2004
The Third International Conference on Axiomatic Design
Seoul – June 21-24, 2004
Put paper number here

AN EXPLORATORY STUDY OF COST ENGINEERING IN AXIOMATIC DESIGN: CREATION OF THE COST MODEL BASED ON AN FR-DP MAP

Taesik Lee
tslee@mit.edu
Department of Mechanical Engineering
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
USA

Peter Jeziorek
peterjez@mit.edu
Department of Mechanical Engineering
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
USA

ABSTRACT

Large complex projects cost large sums of money throughout their life cycle for a variety of reasons and causes. For such large programs, the credible estimation of the project cost, a quick assessment of the cost of making changes, and the management of the project budget with effective cost reduction determine the viability of the project. Cost engineering that deals with these issues requires a rigorous method and systematic processes. This paper introduces a logical framework to achieve effective cost engineering. The framework is built upon Axiomatic Design process. The structure in the Axiomatic Design process provides a good foundation to closely tie engineering design and cost information together. The cost framework presented in this paper is a systematic link between the functional domain (FRs), physical domain (DPs), cost domain (CUs), and a task/process-based model. The FR-DP map relates a system's functional requirements to design solutions across all levels and branches of the decomposition hierarchy. DPs are mapped into CUs, which provides a means to estimate the cost of design solutions – DPs – from the cost of the physical entities in the system – CUs. The task/process model describes the iterative process of developing each of the CUs, and is used to estimate the cost of CUs. By linking the four domains, this framework provides a superior traceability from requirements to cost information.

Keywords: cost, estimation, cost engineering, traceability, cost framework

1 INTRODUCTION

Cost of a large engineering project can be huge: a vast amount of resources are consumed over a long time span. For example, the Big Dig – a nickname for the Boston Central Artery /Tunnel Project and arguably the most complex construction projects ever undertaken in the United States – cost nearly $15B in its more than decade-long project span [Axtman (2000)],[Bechtel (2003)]. Because of the huge cost of such large projects, it is critical to credibly predict and understand the cost of such projects in sustaining the financial viability.

Traditionally, cost estimation research and development have focused on "how much" will a project cost [Dean (1993)]. Cost estimation based on cost estimating relationships (CER) is very popular and widely used. CER-based cost estimation methods use statistical relationships between historical costs and a number of selected parameters that characterize a project/system. As more details about the system developed are understood, more cost information is available and the bottom-up cost estimation becomes feasible. As Dean points out, no matter how accurate the cost estimation is at the time of initial estimation, it has little or no relationship to the final cost since changes in requirements, constraints, and technical maturity are inevitable. The Big Dig project, for example, was originally estimated to cost $2.6B (in 1982 currency) at the time of preliminary concept development. With all the factors that affect the final cost such as inflation and the scope change, the final cost is expected to reach $14.6B (in 2004 currency) [Bechtel (2003)]. The central issue of the dispute between Ingalls Shipbuilding Company and the US Navy over the cost of military shipbuilding contracts – 9 amphibious assault ships (LHAs) and 30 DD963 Spruance-class destroyers – was about which party was responsible for the cost overruns of the two contracts [Cooper (1980)].

Freiman [Freiman (1983)] made an interesting observation on the relationships between the cost growth to the ratio of actual to bid, and represented in the Freiman curve (Figure 1). This curve indicates that underestimates end up creating significant cost growth, compared not only to the initial estimates but also to what it should have been.

The above discussion leads to a conclusion that a good cost estimation method should provide critical insight into 'how much' a system will cost and 'why'. Once a cost estimation method can answer the question of 'why', then it will also be able to address the problem of determining the cost impact of a design change, i.e.

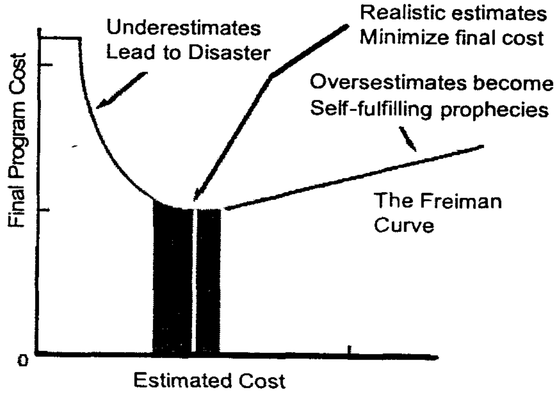

Figure 1. The Freiman Curve

how much will the cost increase in the event of a requirement change? These questions can only be answered when the system design knowledge/information is closely linked with relevant cost information. Currently, the two domains are separated, and "throw-it-over-the-wall" behavior between development group and cost group isn't uncommon. Real-time feedback/feedforward between design and cost is rarely found in practice. The goal of this paper is to lay out a basic framework in which different domains of information – requirements, design solutions, physical artifacts, tasks/processes, and cost – are systematically linked.

2 MODEL STRUCTURE

The developed framework utilizes Axiomatic Design as a basis for building the cost model. The objective of this framework is to facilitate and improve the task of cost engineering by enhancing the credibility of cost estimates and increasing the utility of the cost information. In particular, it accomplishes the following tasks:

- Enhance the credibility of cost estimation by creating the cost model based on the FR/DP map
- Quickly estimate the cost impact of changes introduced to a system
- Identify key cost drivers

This is done by systematically linking three branches of information: a system architecture map (FR/DP map), a costing-unit interaction model, and a process model.

This section discusses the creation of the cost model based on the FR-DP map, and then explains the detailed mechanisms in the model to assess the cost impact of changes. More details can be found in [Jeziorek (2004)]

2.1 CREATING THE COST MODEL BASED ON THE FR-DP MAP

The first order of credibility in an estimate should come from the *completeness* of the scope of estimation. Typically, completeness of the scope of estimation is described in terms of the comprehensiveness of a list of physical artifacts being estimated. In aerospace industry, for example, work-breakdown-structure (WBS) is a standard protocol for such purpose. WBS resides mostly in the physical domain and functions of a system or subsystems are implicit. As a result, the satisfaction of all the necessary functional requirements is considered a separate activity from cost estimation.

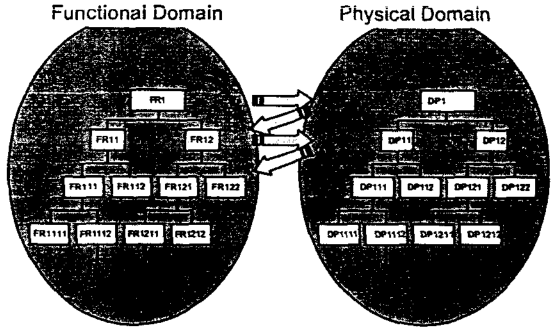

Figure 2. Axiomatic Design process creates a hierarchical description of a system by zigzagging between the functional and physical domains.

Completeness, hence, has two meanings: completeness of functional requirement addressed by a system, and completeness of WBS of a system. From a customer's (internal and external) perspective, what is most valuable is the integrated information. The Axiomatic Design process provides a good framework to integrate design information with cost information.

The fundamental concept of Axiomatic Design process is the creation of hierarchical decomposition that intertwines the functional and physical description of a system [Suh (2001)]. Customer's requirements or customer needs are translated into high-level functional requirements (FRs) that the system has to satisfy. Those FRs are mapped into a physical domain where the engineering solutions to satisfy the FRs are identified. The engineering solutions are called design parameters (DPs). This process – called mapping – continues to repeat at the subsequent levels until the detail design is completed (Figure 2).

In order to connect the design description of the system with the available cost data for the purpose of cost estimation, another domain is created. This new domain is termed as a *costing unit* domain. The costing units (CUs) are entities, cost of which can be estimated. To some degree, it is equivalent to the BOM (bill of material). CUs are not necessarily same as DPs since DPs can be variables or characteristics that are not physical parts. For example, a beverage can has 12 FRs and 12 corresponding DPs, but only three

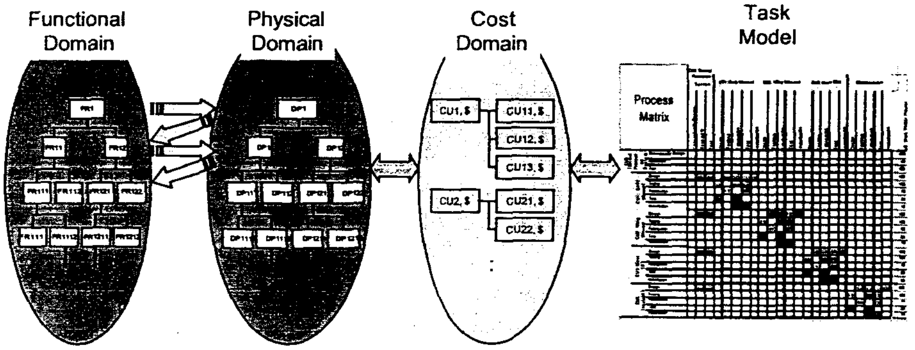

Figure 3. The cost model begins with the functional description of the system in the functional domain. FRs are mapped into DPs, which are then mapped into CUs. Task model captures the iterative process of CU development.

CUs – integrated physical components [Suh (2001)]. The design of the main body of the can should satisfy the functional requirements of containing radial and axial pressure and withstanding impact from a 2m height by modifying design parameters such as the material, thickness, radius, length, and convex shape of the bottom of the can. In our model, FRs, driven by customer's needs, are mapped into DPs, and DPs are mapped into CUs. This chain of information enables the assessment of the cost of FRs. Thus, information that resides in two separate domains is now integrated together. With this structure, the completeness of cost estimation is readily attainable. The scope of estimation can be examined in terms of the functional requirements, and thus ultimately the customer needs, as well as the physical entities. With the functions of a system (FRs), design solutions to the functional requirements (DPs), and the physical entities that embody the design solutions (CUs) linked together, the last piece of information necessary is the process model. It models the iterative nature of the development process by using a task-based iteration model [Eppinger/Smith (1999)]. The task-based iteration model captures all the tasks and processes required to develop and produce each of the CUs. It uses information on task interactions to model the iteration process. The interaction captured in the model is essentially the design information exchange. Using the model, the expected development effort can be estimated. Figure 3 shows the overall structure of the cost model. It encompasses four domains of information: FRs, DPs, CUs and tasks. The model provides a means to systematically organize customer's requirements, engineering design, and cost information.

The FR-DP structure in this model is hierarchical. As the design matures from conceptual solutions to detailed solutions, the FR-DP map develops deeper levels of hierarchical structure. Since the cost information is closely tied to the FR-DP map in our model, it naturally produces a system design cost breakdown from the top-level to leaf-levels. This cost breakdown, along with the hierarchical design description, offers a good way to manage cost information.

2.2 QUICKLY ESTIMATING THE COST IMPACT OF CHANGES INTRODUCED TO A SYSTEM

The life-cycle-cost is the sum of the development, production and operation costs. This relationship can be seen in Equation (1). Development cost is incurred by the developers during the design phase and includes costs of engineering design, management, documentation, prototyping, testing, and evaluation. Production cost is incurred during the manufacturing of the system and is closely related to time to produce each unit and the number of units to produce. Each fully manufactured system takes a certain amount of time to machine and deliver parts, assemble, and test. The production cost is the total cost of manufacturing the desired number of units. Operation cost is incurred by the user of the system during usage and is closely related to the cost of maximizing the satisfaction of all FRs and the number of units in operation. For example, the cost of operation of a Boeing 747 jet includes the fuel, salaries of pilots, stewards, maintenance crews, cost of leasing gates of terminals in airports, and the cost of replacement parts for used or broken aircraft components.

$$\begin{aligned} LifeCycleCost &= \$Development \\ &+ \$Production + \$Operation \end{aligned} \tag{1}$$

In order to analyze the benefit of making a change to the design of a system, the cost advantages and disadvantages for each of these three categories must be analyzed. A developer could spend extra money in the development phase to simplify the system in order to save money in the production phase. On the other hand, saving money in the development or production phase could lead to poor performance and an increase in operational cost. Our method aims to provide this insight to designers when evaluating a proposed design change. With this method, design changes can be evaluated in the light of cost advantages and disadvantages so that the optimal decision can be made.

As a first step toward the complete life cycle cost assessment tool, we created a method of determining the cost impact of a change to the development of a system. The main goal of the method is to identify the components that would be affected by a change made in the functional domain and then determine the change in cost of the development labor. The development cost is the sum of the labor and material costs and investment into infrastructure, as seen in Equation (2). Note that the cost of infrastructure is out of the scope of this cost estimation effort. We employ a simple proportionality rule of labor and material to estimate the total development cost.

$$\begin{aligned} \$Development &= \$Labor_{developmen} + \$Material_{developmen} \\ &+ \$Infrastructure_{developmen} \end{aligned} \tag{2}$$

2.2.1 IDENTIFYING THE COMPONENTS AFFECTED BY A FUNCTIONAL CHANGE

The Axiomatic design framework provides a mapping from customer needs into functional requirements (FRs), or the set of functions that the product must perform in order to satisfy the needs of the customer. Functional requirements are then mapped into design parameters (DPs), or specific engineering parameters that are varied in order to perform the desired functions. By doing this, a clear connection is established between the customer's needs and the actual product being designed. The relationship between FRs and DPs, as seen in Figure 4 is of special interest. There are four functional requirements and four design parameters that are related to each other by this matrix. FR1 is affected by DP1, FR2 is affected by DP1 and DP2, FR3 is affected by DP2 and DP3, and FR4 is affected by DP4. Or in other words, if DP1 is changed, FR1 and FR2 will be affected, if DP2 is changed, FR2 and FR3 will be affected, if DP3 is changed, only FR3 will be affected, and if DP4 is changed, only FR4 will be affected. Later, this information will be used to identify which DPs will be necessary to change in order to satisfy a change to a functional requirement.

	DP1	DP2	DP3	DP4
FR1	X			
FR2	X	X		
FR3		X	X	
FR4				X

Figure 4. FR-DP relationship

A designed system is the sum of all its components, which are directly related to the design parameters used in designing the system. Each DP can correspond to many physical components, and vice versa as in the beverage can example discussed in the above section. The design of each component corresponds to the adjustment of various design parameters. Therefore, there is a direct relationship between design parameters and components, also termed costing unit (CU). CUs are the objects of the cost estimation, and the physical artifacts generated from the design. For the beverage can, the three physical components of the can would be the three costing units. The relationship between DPs and CUs can be seen in the example in Figure 5. CU1 contains DP1 and DP2; CU2 contains DP1, DP2 and DP3; CU3 contains only DP4. Each DP can be embedded in multiple CUs and each CU can embody multiple DPs.

	CU1	CU2	CU3
DP1	X	X	
DP2	X	X	
DP3		X	
DP4			X

Figure 5. DP-CU relationship

With these two matrices, we can determine the list of components that are affected by a functional change. Suppose that the following design matrix, in Figure 4, was under consideration for a significant design change. In order to best satisfy the customer needs, a manager decides that FR1 will have to change. Consequently, in order to satisfy FR1, DP1 will also have to change. Because DP1 also affects FR2, DP2 will have to be changed in order to compensate for the effect from the change in DP1 and still satisfy FR2. Likewise, DP3 will need to be modified to reflect the change impact from DP2. The result of the change to FR1 is that DP1, DP2 and DP3 will have to change. From the DP-CU relationship, seen in Figure 5, we can find the CUs (components) that will be affected by the changes to DP1, DP2 and DP3. Reading from left to right and then up, we can identify CU1 and CU2 as the components that will need to be changed as a result of the change to FR1. Thus the output from the DP-CU matrix is a list of components affected by the functional changes. Figure 6 summarizes the process.

	DP1	DP2	DP3	DP4
FR1	X			
FR2	X	X		
FR3		X	X	
FR4				X

	CU1	CU2	CU3
DP1	X	X	
DP2	X	X	
DP3		X	
DP4			X

Figure 6. A change in FR1 necessitates a change in CU1 and CU2.

This list of affected components only takes into account functional interactions between design parameters. Many components interact with each other physically as well as functionally. However, this information is typically not captured by a design matrix. Instead, a new CU-CU matrix was created in order to capture physical interactions between CUs. Imagine a set of components that physically interact with each other, as in Figure 7. There are five attributes of components that can interact with other components: physical, spatial, thermal, information and electromagnetic. The physical attribute indicates that this component physically integrates with another component, for example, by a mount or tubing. The spatial attribute specifies size or location of the component. The thermal attribute marks the presence of heat exchange. The information attribute is the flow of information into and out of a component. The electromagnetic attribute is the transmission of an electromagnetic signal into or out of a component. A component can interact with its neighbors through any combination these five attributes. For example, In Figure 7, the physical attribute of Component 1 interacts with the physical attribute of Component 3 and the information attribute of Component 1 interacts with the electromagnetic attribute of Component 2. These interactions are two way because of the question that we ask: "Does the information attribute of Component 1 interact with the electromagnetic attribute of Component 2?" "Does the electromagnetic attribute of Component 2 interact with the information attribute of Component 1?" is the same question. The answer is yes for both questions, since they are the same question. Even an attribute within component 1 could interact with another attribute of component 1. In the example in Figure 7, the information and electromagnetic aspects of component 1 interact with each other.

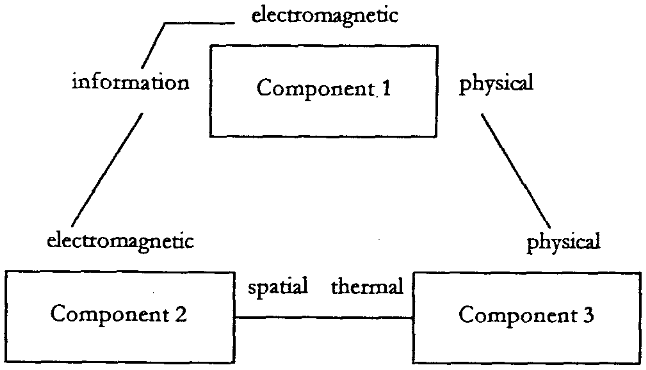

Figure 7. Component-Component interactions

2.2.2 DEVELOP THE DEVELOPMENT LABOR COST

Now that the complete list of CUs has been identified and the amount of rework required for each CU has been calculated, the amount of development time required to implement those changes can be determined. By measuring the impact of a change on the development time of a project, the cost of labor can be determined. This is accomplished using a task-based model [1] [Eppinger/Smith (1999)].

For each CU, a set of tasks/processes are identified. At the early conceptual design stage, the set can be as general as 'preliminary design,' 'detailed design,' 'tooling,' and 'testing.' To alleviate the workload to specify detailed sets of tasks for each and every CU, one might group CUs to several major categories within which common processes are assumed to apply. The task-based model employs the concept of a work transformation (WT) matrix developed by [Eppinger/Smith (1999)]. The work transformation matrix, shown in Figure 8, captures the interactions among individual tasks, thereby computing the amount of rework for the subsequent iterations due to the interactions.

	A	B
A		0.5
B	0.3	

Figure 8. A work transformation matrix, WT, for two tasks, A and B. When performed in parallel, task A creates 30% rework for task B in the next iteration. Likewise, task B creates 50% rework for task A in the next iteration.

[1] In this paper, two terms, 'task-based model' and 'process-based model' are used interchangeably.

For example, in Figure 8, task A creates 30% of rework for task B in the next iteration, and task B generates 50% of rework for task in the next iteration. The amount of rework for task A and task B for n^{th} iteration is, thus, the amount of work done in $(n-1)^{th}$ iteration times the matrix, **WT**:

$$u_n = WT \cdot u_{n-1} \qquad (3)$$

where u_n is a column vector, called a *rework* vector, whose components represent the amount of rework at n^{th} iteration step.

By summing up the rework vector at each iteration step until it converges – i.e. each component of the rework vector ~ 0 –, one can estimate the total amount of work required to complete task A and B, given the task interaction structure:

$$\begin{aligned} U &= u_0 + u_1 + \cdots u_N \\ &= u_0 + WT \cdot u_0 + WT \cdot (WT \cdot u_0) \cdots \\ &= (1 + WT + WT^2 + \cdots + WT^N) \cdot u_0 \end{aligned} \qquad (4)$$

where U is a column vector that represents the total amount of work during N iterations, and u_0 is the initial amount of work to complete each task (each component of u_0 is normalized to 1). Our model assumes that the work transformation matrix is constant through each iteration step (i.e. that WT does not depend on time). The curve in Figure 9 shows the tendencies of the total work amount – for simplicity, a sum of elements in U. A fast progress is made during early iteration steps, the rate of progress becomes small toward the finish of the work, and finally it converges – no more rework is required.

Once U is computed, it is converted to the total development time by multiplying U with the actual initial amount of work. Then, a standard labor rate for each task is multiplied to the total development time, generating a monetary value for doing the required change implementation.

2.2.3 COMPUTING THE CHANGE COST

The last step toward estimating the cost impact of a change is comparing the current projected cost with as-is design to the adjusted cost projection given the change. As described in the foregoing sections, a change in FRs is translated into a necessary change in relevant DPs. Then, the list of such DPs identifies the CUs that need to be modified to accommodate the DP changes. The CU-CU interactions are also considered to account for any physical implementation issues. Once a complete set of CUs that need to have been adjusted, the task-based iteration model estimates the effort to implement such changes in terms of labor hours.

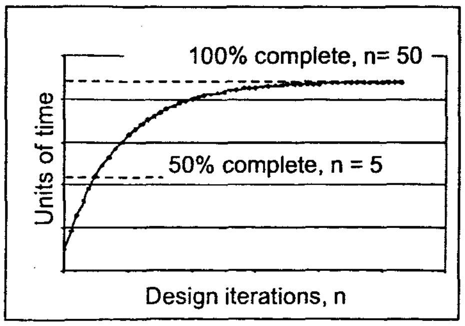

Figure 9. The total amount of work increases rapidly during the early iterations , and the progress is slowed down until it eventually converges.

Two input parameters are used in the current method of the change cost calculation: the severity of the change introduced to the system , and % completion of a current development process at the time the change is made. The severity of a change is an approximate measure of the degree to which the identified CUs have to be redesigned. For example, the implementation effort for a minor design change for an isolated component will be quite different from that for a major redesign work. In addition to the severity of a change, when the particular change is introduced to a system has a significant impact on the final prediction. Early in the development phase, there is a relatively little cost penalty for making changes while a redesign at the end of the development phase will cause a significant waste of resources and, thus, incur a larger cost penalty.

Our method takes these factors into account by taking the % completion of a current development process and adjusting the rework vector using a parameter, α. In other words, the model determines an interrupt point, say k^{th} step, during the iteration that corresponds to % completion input, and replaces the current rework vector, u_k, with a new rework vector, u_k', that reflects the change. For example, if the current development phase for as-is design follows the curve shown in Figure 9, and a change is introduced at about 50% development completion, then the model replaces u_5 with u_5^* where u_5^* can be calculated as:

$$u_5^*(i) = (1 - \alpha_i) \times u_5(i) + \alpha_i \qquad (5)$$

where α_i is a severity parameter for task *i*. α_i ranges from 0 to 1, 0 for no change and 1 for the most severe change, e.g. complete re-implementation required. The total work vector can then be calculated, seen in Equation 6, in the style of Equation 4.

Suppose that some changes are introduced to a system when about 60% of the current development is complete. In Figure 10, a curve plotted by square symbols () represents a work progress profile with the original design. Without the change, total development time will be 220 units. The change introduced system increases the development workload as some parts of the system need to be re-worked to accommodate the change. A curve marked by (x) in Figure 10 shows the new workload profile. As expected, the change implementation requires additional effort and results in larger total workload (290 units). It is the difference between the two final workload estimates that should be considered since that difference is the penalty caused by the change. In this case, Figure 10 shows an increase of 70 units in the final total workload.

The difference of 70 units of workload is converted to the actual development hours by multiplying each of them by the corresponding initial workload hours, and then interpreted in terms of monetary value given the relevant labor rates. As mentioned earlier, material cost during development phase is approximated to be a certain fraction of the labor cost based on historical data for each set of tasks. The cost for acquiring new infrastructure has to be assessed separately with some accounting standard established in the organization. Three factors of development cost are summed to generate a final estimate of the cost impact of a change to the system.

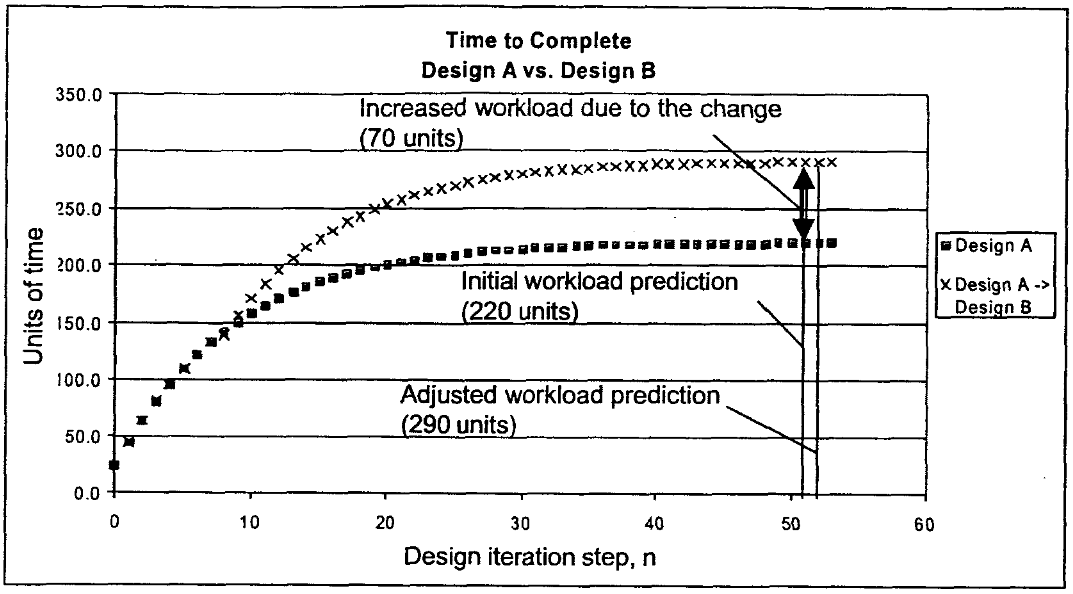

Figure 10. A change is introduced when 60% of the development has been completed. A plot marked by square symbols (▪) shows a work progress profile with the original design, and a plot by (x) shows a new trajectory with the design change.

3 DISCUSSION

The model presented in this paper is at its preliminary stage. It has much room for improvement. First of all, it only addresses the development phase of the program life cycle. It needs to be expanded to obtain capability to manage and manipulate the cost information in the subsequent phases, production and operation. As the model expands to those phases, the dynamic nature of the project cost becomes more important. Only after the complete life cycle cost can be assessed can the model truly contribute to the task of cost engineering.

Within the development phase, it also needs to be improved in many aspects. Quantifying the relational links between FR-DP, DP-CU, and CU-CU is a challenging task both in theory and in practice. For example, in CU-CU physical interaction, how far should we allow the propagation of changes? What should be the best way to create, maintain and update such huge matrix information? Another area of consideration is the task-based development process model. Obviously, the task-based model discussed in this section can and should be improved to reflect the reality of iterative nature of a development process. For example, in many cases, performing different tasks in a parallel fashion is unlikely. It will be reasonable to assume that some of the task interactions are known a priori and will be reflected in the development process by following certain sequences. Some works have been done previously, and some of the serial process iteration models are available [Eppinger/Smith (1999)]. In plotting Figure 9, it is assumed that the work transformation matrix is constant throughout the entire iteration steps. The assumption may not be realistic since the same tasks will require less amount of effort if they have been done previously due to the learning curve effect. Determining the values of the work transformation matrix is a possible heuristic solution to take into account the learning effect [Cho/Eppinger (2001)]. Still another area for improvement is effectively quantifying the severity of a design change depending on the nature of the change being made. A trade-off has to be made between employing expert knowledge (accuracy) and producing rough estimates with minimum external input (speed).

One of the most intriguing questions about the cost estimation is "what is the absolute minimum cost of a system". Typically, the cost data from history represent the actual, final cost that accounts for all factors affecting the cost, e.g. mismanagement, and underestimation (refer to Figure 1). Thus, the estimates based on those data do not reveal the realistic minimum cost of a system. Knowing the true minimum cost of the system will prevent a waste of resources and enable better cost management for contractors to maximize their profit. This challenging task of understanding the 'minimum' cost of a system will be a very important contribution.

These issues should be carefully studied to obtain a best resolution within the practical limitations of the cost model's operating environment.

4 CONCLUSION

A framework to address the problems in cost engineering is developed. The framework provides a structure to integrate a system design knowledge/information with the cost information. In particular, as a first step toward the grand goal, a preliminary cost model for a development phase of a system life cycle is presented.

The model renders the information flow between engineering requirements, design solutions, their embodiment (artifacts), and cost data tied to tasks/processes required for the physical implementation. This model is developed based on the Axiomatic Design process. It offers an effective way to examine the completeness of the scope of estimation to ensure the first order of the credibility of the estimates. It also provides traceability between individual domains within the development phase, which is particularly useful in assessing the cost impact when a change is introduced to a system at certain point.

5 ACKNOWLEDGMENTS

The authors wish to thank Mr. Chip Woods, Mr. Ray Damaso, and professor Nam P. Suh for helpful discussions and their support. The authors would also like to thank Axiomatic Design Solutions, Inc. for their kind cooperation in developing a software prototype.

6 REFERENCES

[1] Axtman, Kris. "In Boston, a 'Big Dig' into taxpayer pockets." http://search.csmonitor.com/durable/2000/02/18/p3s1.htm. Published on WWW; accessed in 2004.

[2] Bechtel Report, February 2003, http://www.bechtel.com/PDF/Big%20Dig%20key%20facts.pdf. Published on WWW; accessed in 2004.

[3] Cho, Soo-Haeng and Eppinger, Steven D. "Product Development Process Modeling using Advanced Simulation," Proceedings of ASME Design Engineering Technical Conference (DETC) 2001.

[4] Cooper, Kenneth G. "Naval Ship Production: A Claim Settled and a Framework Built." Interfaces, Vol. 10, No. 6, pp. 20-36 December 1980.

[5] Dean, Edwin B. "Why Does It Cost How Much?," AIAA 1993 Aircraft Design, Systems, and Operations Conference, Monterey CA, 11-13 August 1993.

[6] Eppinger, Steven D and Smith, Robert P.. "Identifying Controlling Features of Engineering Design Iteration," Management Science, Vol. 43, Issue 3, pp 276-293 1999.

[7] Eppinger, Steven D., Whitney, Daniel E., Smith, Robert P. and Gebala, David A. "A Model-Based Method for Organizing Tasks in Product Development," Research in Engineering Design, 1994.

[8] Freiman, Frank R. The Fast Cost Estimating Models. AACE Transactions, G.5, AACE International, Morgantown, WV, 1983.

[9] Jeziorek, Peter N. "Cost Estimation of Functional and Physical Changes to Complex Systems", Master's Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, To Be Published 2004.

[10] Suh, Nam Pyo. *Axiomatic Design: Advances and Applications*. New York: Oxford University Press, 2001.