

A Framework for Design Process Specifications Management

ALI A. YASSINE & DONALD R. FALKENBURG

SUMMARY Major conflicts in design process management seem to stem from specification conflicts among the constituent coupled design tasks. Coupled tasks represent conflicts in the flow of information of a design process. Resolving these specification conflicts early on in the development process tends to reduce product development lead times and cost. In this paper, we attempt to explain the coupling or interdependency in design tasks by the specifications imposed on these tasks. Then, we formulate the fundamental equations necessary for engineering specification management and devise a graphical tool for design process improvement. The major hypothesis investigated is that any two coupled tasks can be de-coupled if the specification of one task can be designed to absorb a certain percentage of any possible variation in the output of the other task.

1. Introduction

Concurrent product design and development (or concurrent engineering) is an attempt to incorporate various constraints and requirements, related to the product life cycle, in the early design stages [1–3]. This approach tends to improve the product quality and reduce the product development lead-time and manufacturing costs [4]. However, due to concurrent consideration of many factors, the relationships and dependencies between some design tasks may be violated (i.e. tasks that are inherently sequential are performed concurrently). This will increase the complexity of the design process and make it more difficult to manage [2].

Most of the research in concurrent product development management is divided into two main streams. First, research oriented towards the improvement of methods and tools that extract product specifications from a set of customer requirements (i.e. QFD and its derivatives) [1]. Second, research devoted to the performance improvement of the design and development process given a set of product specifications to be satisfied [2,5,6,7–9]. Relatively little research attention seems to be oriented towards the process of managing design task specifications once these are attained. Major conflicts in design process management stem from specification conflicts among the constituent design tasks. Resolving these specification conflicts early on in the development process is essential for successful concurrent engineering implementations.

The study of task specifications (i.e. tolerances) is not a new concept. The literature

is full of articles discussing models and techniques of tolerance analysis, allocation and computer-aided tolerancing [10–13]. The majority of these studies address the tolerancing problem from a minimum manufacturing cost perspective [21]. Another major technique that gained a lot of attention in the area of quality engineering is Taguchi's tolerance design [14,15]. The Taguchi method calls for statistically determining the components that have a large contribution to the variance of a measured quality characteristic and imposing tight tolerances on them. Suh [16] took a completely different approach to tolerancing through his axiomatic design principles. He discusses the relationship between the tolerances imposed on the functional requirements (FRs) of a design and the coupling of the design. Our research, described in this paper, builds on the work of Suh [16] and Steward [17,18] that is introduced in the next section.

Coupled tasks represent conflicts in the flow of information of a design process. Resolving such conflicts requires engineers to iterate through trial designs until an agreement is reached. Therefore, coupled tasks are mainly responsible for design process iterations. Managing design process iterations inevitably lead us to investigate the management of task specifications. This paper presents a methodology to maximize design process concurrence through a better understanding of design task specifications. We attempt to explain the coupling or interdependency in design tasks by the specifications imposed on these tasks.

The rest of the paper proceeds as follows. In Sections 2–4, we introduce Steward's [17,18] work on design structure management, Suh's [16] work on axiomatic design, and combine them through a sensitivity design structure matrix (SDSM) representation. In Section 5, we introduce our new concept of specification management and lay down the fundamental equations. A simple example is presented in Section 6 to familiarize the reader with the application of the new method. A generalized graphical solution, for de-coupling two coupled tasks, is then proposed in Section 7. Finally, in Section 8, we present a summary and some concluding remarks.

2. The Design Structure Matrix

Steward [17,18] developed the design structure matrix (DSM) as a tool to represent and analyze task dependencies of a design project. The DSM is a binary square matrix with m rows and columns, and n non-zero elements, where m is the number of design tasks and n is the number of information flows (or dependencies) going from one design task to the other. If there exists an information dependency from task i to task j , then the value of element ij (column i , row j) is unity (or marked with an X). Otherwise, the value of the element is zero (or left empty).

If one interprets the task ordering, in the matrix, as a time sequence, then marks below the diagonal represent information transferred to later tasks, and marks above the diagonal depict information fed back to earlier tasks. Feedback marks represent a difficult situation for engineering design managers because they imply that upstream tasks require information from downstream tasks. In an attempt to eliminate such marks, tasks within the DSM are rearranged such that the flow of information will be solely in the forward direction. This process of task rearrangement is called partitioning. Several partitioning algorithms have been cited in the literature; however, a discussion of partitioning algorithms is outside the scope of this paper. The interested reader can be referred to Refs. [2,17–19].

Three different task dependencies (i.e. relationships) can be readily observed when inspecting a partitioned DSM: sequential tasks, parallel tasks and coupled tasks. A task is part of a sequential group if its row contains a mark just below the diagonal. Two or

Task	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Receive and accept specification	A													
Concept generation / selection	X	B												
Design beta cartridges	X	X	C											
Produce beta cartridges			X	D										
Develop testing program	X	X	X		E									
Test beta cartridges			X	X	X	F								
Design production cartridge	X	X	X			X	G	X	X					
Design mold	X	X				X	X	H	X					
Design assembly tooling							X	X	I					
Purchase assembly equipment						X	X	X	J					
Fabricate molds							X				K			
Debug molds							X	X			X	L		
Certify cartridge					X						X		M	
Initial production run											X	X	X	N

FIG. 1. Simplified DSM for the Kodak Cheetah project (source: Ulrich & Eppinger [20]).

more tasks are parallel if there are no marks linking them. Finally, two or more tasks are coupled if there is no ordering of the DSM that will render the matrix lower triangular. Therefore, the existence of a mark above the diagonal, after partitioning the DSM, indicates coupling between tasks [19]. A sample DSM with the three different types of relationship is shown in Fig. 1. Note that tasks B and C are considered sequential, tasks D and E are parallel, and tasks G, H and I are coupled.

The DSM representation was used and proved, by several researchers [2,18,19], to be a useful tool in concurrent engineering management and implementation. Its ability to display all the design steps involved in the design process, allowing for direct coupling between them, in a compact form is the basic strength of this tool.

3. Axiomatic Design and the Design Matrix

Suh [16] describes the design process as the mapping between the FRs, in the functional domain, and the design parameters (DPs), in the physical domain. The FRs are defined as the minimum set of independent requirements that completely characterize the functional needs of the product. The DPs are the key variables that characterize the physical entity created by the design process to fulfill the FRs [16]. The design matrix (DM) relates a component of the FR vector to a component of the DP vector [19]. The mapping between a given FR vector and a DP vector having a design matrix [A] is given by the design equation as:

$$\{FR\} = [A]\{DP\} \tag{1}$$

For the two-dimensional case, [A] may be represented as:

$$[A] = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{2}$$

In general, the elements of the DM may be expressed by: $A_{ij} = \partial FR_i / \partial DP_j$. Analysis of the DM, as described by Suh [16], is extremely useful in establishing the characteristics of a 'good' design. That is, a designer will be able to discriminate between a good

design and a bad one by inspecting and comparing the different DMs corresponding to the different design alternatives.

According to Suh [16], a design is coupled if it violates design Axiom 1. Axiom 1 states that for an uncoupled design, each FR is strictly satisfied/mapped to one DP. That is, a change in one FR cannot be accomplished by simply changing the corresponding DP. Changing one DP will affect more than one FR. Simply stated, Axiom 1 describes an optimal design as that which maintains the independence of FRs. In order to do so, the design activities that constitute this design process must, then, be uncoupled. It is worth noting that a diagonal DM represents an uncoupled design, while a DM containing non-diagonal elements represents a coupled design. Suh [16] suggests that a coupled design may be de-coupled by rendering the DM lower triangular. In doing so, we will be able to adjust the DPs in a particular order; thus, satisfying Axiom 1.

4. The Sensitivity Design Structure Matrix (SDSM)

Steward's [17,18] DSM is different from Suh's [16] DM in that it describes the system only in the physical domain and it relates the DPs to each other. Therefore, Suh's uncoupled diagonal DM becomes a 0 DSM. In addition, Suh [16] recommends that we seek a de-coupled design if an uncoupled one does not exist. A de-coupled design is a design solution where the DM is a triangular matrix. This description of a de-coupled design is in agreement with Steward's partitioning algorithm that seeks a lower triangular form of the DSM.

The SDSM is a link between Suh's axiomatic approach to design and Steward's representation of design process dependency. The off-diagonal entries, of a SDSM, in row i and column j represent the partial derivative for the output of task i to the output of task j : $\partial O_i / \partial O_j$. Extending the classical DSM representation by the inclusion of the sensitivity coefficients allows us to understand and capture the design process task interactions more accurately (as compared with the binary DSM). The improved DSM representation allows us to investigate a new approach for improving the design process; namely, management of task specifications, as will be discussed in the rest of this paper.

In order to evaluate the partial derivatives, we need to be able to express the system and its tasks analytically, i.e. in terms of mathematical expressions. In fact, we are seldom able to express complex engineering systems and the interaction among its subsystems and components analytically. Often, these relationships are not known and must be determined experimentally [16]. Classical statistical design of experiments [21] or Taguchi's approach [14,15] can be used to define polynomial relationships between independently varied parameters and measured outputs, therefore experimentally defining sensitivities between the tasks.

5. A Framework for Design Specifications Management

A specification hierarchy is a map showing the impact of a change in the specifications of a task, in the design process, on other tasks within the same design process. Inspecting the DSM and tracing the marks associated with a certain task develops this hierarchy. Any disturbance or change in the specification at one point in the design network will be propagated through the specification hierarchy and impact downstream tasks. The effect of the disturbance can be tolerated if the task is independent of the excited task or if the downstream task specification can absorb it. Evaluating the partial derivatives

of these tasks determines whether the later condition exists. Therefore, the SDSM serves both purposes of developing the hierarchy and investigating task sensitivities.

The first step in specification management is to determine the contribution of each feedback mark (in the DSM) to the output of a certain task impacted by these feedback marks. In order to select the principal feedback marks that dominate the output of a task i , the variance σ_i^2 of O_i can be determined by varying the output of all the tasks representing these feedback marks (i.e. O_j values). The O_j that gives the largest value of σ_i^2 has the largest effect on O_i and, hence, is the dominant (i.e. principal) feedback mark [16]. For the feedback marks that contribute only a small amount to the task's output, consider de-coupling the two tasks by allowing wider specification limits on the dependent task. For input streams that contribute a large amount to the output of a task, consider virtually de-coupling the two tasks. Two coupled tasks A and B are virtually de-coupled if the specification of task A is designed to absorb a certain percentage (say 95%) of any possible variation in the output of task B, and vice versa.

5.1 A Motivational Example

Although tasks in real product development processes can be complex, consider a simple example where task A depends on three other tasks. In this example, task A receives design information, from its predecessors, about parameters X_1 , X_2 and X_3 . Task A performs its analysis using function $Z = 0.01(X_1) + 100(X_2) + 20(X_3)$ and delivers the design parameter Z to its successors. Clearly, task A is most sensitive to changes in X_2 and least sensitive to changes in X_1 because

$$\frac{\partial Z}{\partial X_1} < \frac{\partial Z}{\partial X_3} < \frac{\partial Z}{\partial X_2}.$$

If the teams computing values for X_2 and X_3 can accomplish their job in 1 week, but the team responsible for computing X_1 will take 3 weeks to accomplish this task, then is it important to wait for the input from team 1? There is no simple answer for this question; however, if we can establish a range of values for X_1 , we can begin to answer the question. Assuming that, based on similar design experiences, X_1 is known to be uniformly distributed between 1000 and 1002, then $Z = 0.01E(X_1) + 100(X_2) + 20(X_3) \pm \text{error}$. $E(X_1)$ is the expected value of X_1 and the error term is equal to

$$\frac{\partial Z}{\partial X_1} \frac{\Delta X_1}{2} = 0.01 \frac{(1002 - 1000)}{2} = 0.01.$$

If the error term lies within the specification of task A, then task A can definitely proceed without precise information on X_1 design data. On the other hand, if the specification of task A is smaller than the error term, then there is a chance that task A will be repeated due to the approximation of the value of X_1 . Fig. 2 shows the relation between the error and specification probability distributions, where the shaded area represents the resultant repetition probability.

5.2 Representation of a Design Task

At the lowest level of detail, an engineering design process is divisible into a number (usually large) of individual design tasks that collectively define a complete project/product [2,17,18]. Each task is defined by a specification S . Specifications are usually

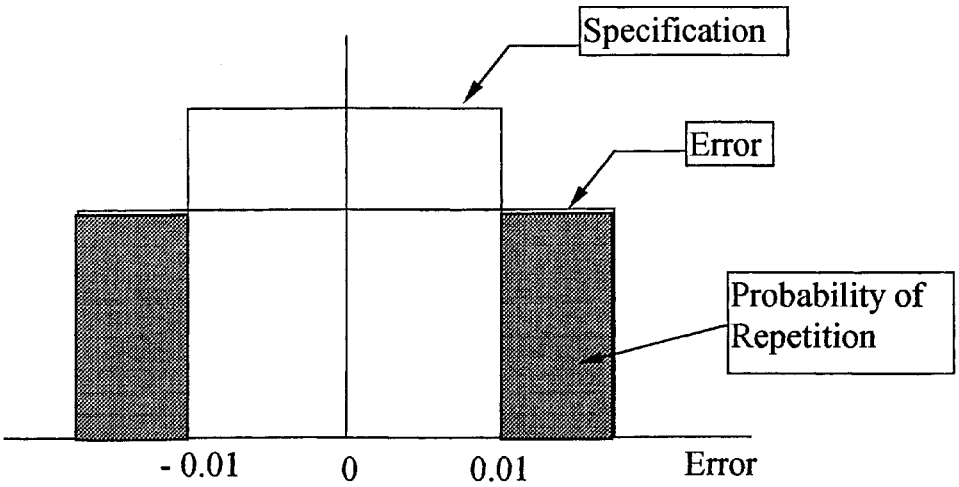


FIG. 2. Probability of task A repetition assuming a rectangular specification distribution.

given as nominal values plus/minus a tolerance value. In order to perform the task, an input I is required. There are two types of inputs: design variables (or controllable input) and interface variables (or uncontrollable input). Design variables are the output of the task itself. Furthermore, they can be decided and varied by the task itself. The interface variables constitute the outputs of some other tasks. The result of the task is an output O , which is a function of both design and interface variables. The performance Ω is a measure of the conformance of the output to the specification. Namely, it is the probability that a task's output falls within the specified specification limits for that task. A block diagram representation of a task is shown in Fig. 3 with the four different

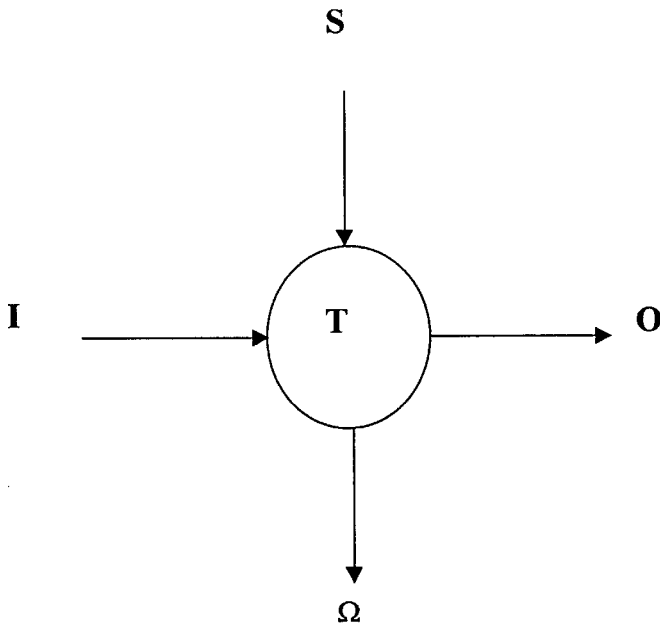


FIG. 3. Block diagram of a task.

attributes that characterize any task. Interface variables ensure that certain decisions are reached in conjunction with previous decisions, and are subject to change based on later results. In particular, further analysis may show that any previous decision is unworkable and needs to be redone. It is important to distinguish between the input of a task (I) and its specification (S). Inputs are information supplied to the task based on its request: they could be a parameter, a document, a computer-aided design drawing, or a process/product data. On the other hand, specifications are constraints imposed on the design task and they represent the bounds on a feasible solution. Task specifications could be time or cost limitations, space constraints or power/torque requirements.

5.3 Basic Notation

If a set of tasks j feed task i , then the output of task i is dependent on its input (I_i) and the outputs of tasks $j(O_j)$:

$$O_i = f(I_i, O_j) \tag{3}$$

The change in O_i is explained by:

$$\Delta O_i = \frac{\partial O_i}{\partial I_i} \Delta I_i + \sum_{\forall j} \frac{\partial O_i}{\partial O_j} \Delta O_j \tag{4}$$

An optimal design (i.e. uncoupled) is a design solution where the change in the output of task $i(O_i)$ is only explained by changes in I_i , thus forcing the second term of the right-hand side of equation (4) to be equal to zero, as in equation (5).

$$\sum_{\forall j} \frac{\partial O_i}{\partial O_j} \Delta O_j = 0 \tag{5}$$

There are four different design scenarios that satisfy equation (5).

- (1) A design environment where ΔO_j is always zero. That is, there is no uncertainty or changes in the values of O_j throughout the whole design process. This situation is referred to as a static design environment.
- (2) The determination of a design (O_i) such that the sensitivity of O_i with respect to O_j ($\forall j$) is zero, as in equation (6). This is referred to as an uncoupled design. If two tasks i and j satisfy equation (6), then the two tasks are said to be uncoupled. Satisfying equation (6) is a purely design issue, where the system is designed in such a way that its components do not interact or influence (i.e. are independent of) each other. This is in agreement with Suh's [16] Axiom 1 describing an optimal design as an uncoupled one where the design tasks are completely independent from each other.

$$\frac{\partial O_i}{\partial O_j} = 0, \forall j \tag{6}$$

- (3) In the absence of a design luxury allowing us to utilize equation (6), a designer must seek a de-coupled design where the design tasks are weakly dependent on each other. The designer seeks a design (O_i) such that variations in O_j are

absorbed by the specification of task i , as in equation (7). This is referred to as a de-coupled design where task i is weakly dependent on a set of tasks j .

$$LSL_i \leq \sum_{vj} \frac{\partial O_i}{\partial O_j} \Delta O_j \leq USL_i \tag{7}$$

Here, LSL and USL denote the lower and upper specification limits of task i , respectively. If two tasks i and j satisfy equation (7), then the two tasks are said to be de-coupled.

- (4) In some design situations, equation (7) cannot always be satisfied. However, it can be satisfied with some probability, as in equation (8). In this case, the design is referred to as a virtually de-coupled design, and tasks i and j are virtually de-coupled tasks. The performance of task i (Ω_i) is defined as in equation (8).

$$\Omega_i = \text{Probability} (LSL_i \leq \sum_{vj} \frac{\partial O_i}{\partial O_j} \Delta O_j \leq USL_i) \tag{8}$$

where Ω_i is the probability that the cumulative sum of all design changes of the predecessor tasks (relative to task i) is within the tolerance limit of task i .

6. An Illustrative Example

Consider the design of the cylindrical part (shaft) explained in Ref. [3]. There are two stages (i.e. tasks) in this design problem, as shown in Fig. 4. First, the design stage involves specifying tolerances on the shaft. Then, the manufacturing stage specifies a manufacturing process to transform a bar stock into a finished shaft meeting specifications.

The two tasks are uncoupled if the choice of a manufacturing process is irrelevant to the design alternative selected by the design engineers. On the other hand, the two tasks are coupled if the number of rejects produced by any manufacturing process, chosen by the manufacturing engineers, are not always acceptable (i.e. within manufacturing specification).

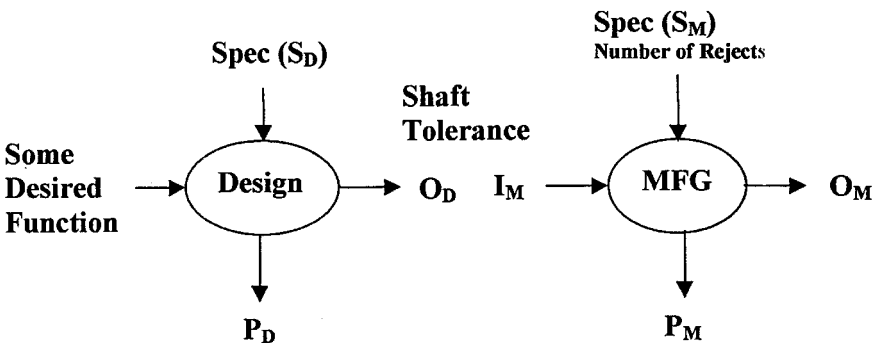


FIG. 4. Shaft design.

Design Specifications			
MFG Process	± 0.003	± 0.004	± 0.005
Turret Lathe	465	184	105
Engine Lathe	155	46	12
ASM	3	00	00

The entries in the above table are the number of rejects produced using a certain manufacturing process at a different set of design specifications.

FIG. 5. Shaft design data.

Fig. 5 is taken from Singh [3], but we also added the (± 0.005) as an added design specification level. The corresponding reject calculations for this added design level can easily be attained using equations described in Ref. [3]. Note that it is assumed that the cost of producing a part increases as we move from a turret lathe to an engine lathe to an automatic screw machine (ASM).

If the design specification, $S_D = \pm 0.003$, and the manufacturing specification, $S_M = 0$, rejects, then the two specifications on the design task and the manufacturing task conflict and S_M is unattainable with the available manufacturing processes. Also, Fig. 5 indicates that there exist some, but not all, design points at which both specifications can be satisfied simultaneously. Recognizing such phenomena, we can say that the two tasks are coupled. In order to de-couple the two tasks at $S_D = \pm 0.003$, S_M should be ≥ 465 rejects. This would make the choice of any manufacturing process irrelevant, from the perspective of a number of defects. If we cannot tolerate such a high number of rejects, then we either change S_D or we have to couple the two tasks. If we choose to change the design specification (S_D), then S_D should be equal to ± 0.004 or ± 0.005 . Furthermore, if we choose to couple the two tasks, then a turret lathe cannot be considered as a manufacturing option (at $S_D = \pm 0.003$).

The equations that govern the interaction between design and manufacturing from the manufacturing point of view are:

$$O_M = f(O_D) \tag{9}$$

$$\Delta O_M = \frac{\partial O_M}{\partial O_D} \Delta O_D \tag{10}$$

If we let $\Delta O_D = \pm 0.001$, then from Fig. 5 it is obvious that $\Delta O_M \neq 0$ and

$$\frac{\partial O_M}{\partial O_D} \neq 0.$$

Consequently, the two tasks are coupled. In order to de-couple them, we must set S_M such that $S_M = \geq 465$ rejects.

$$0 \leq \frac{\partial O_M}{\partial O_D} \Delta O_D \leq 465 \tag{11}$$

If we choose to virtually de-couple the two tasks, then manufacturing should be able to allocate probabilities for the likelihood of choosing any one of the three design

alternatives (i.e. ± 0.03 , ± 0.004 or ± 0.005). Assuming that any one design has an equal probability of being selected by the design engineers (i.e. any one design has a probability of 33.33% of being selected), then setting S_M to be between 0 and 184 rejects virtually de-couples the manufacturing task from the design task 66.67% of the time. That is, 66.67% of the time, the manufacturing process choice is independent of the design specification selected by the design engineers. The resulting performance of the manufacturing task will be $P_M = \text{Probability}(0 < S_M < 184) = 66.67\%$.

7. A Graphical Solution for Coupled Tasks

The graphical interpretation and representation of the presented discussion is shown in Fig. 6. The area above the line $S_M = 465$ represents the uncoupled tasks region because all the manufacturing processes fall below it. If manufacturing choose to operate in the region ($S_M < 465$), then the two tasks become coupled. How much coupling between the tasks is decided by how low the S_M line can go. This question is also directly related to the manufacturing belief in the chances of $S_D = \pm 0.003$ occurrence. If manufacturing believe that $S_D = \pm 0.003$ is less likely to happen (say less than 5%), then lowering the S_M line to 184 (see Fig. 5) will ensure that S_M is met 95% of the time for any choice of S_D , therefore virtually de-coupling the two tasks 95% of the time.

Fig. 7 shows a generic template for the graphical solution of two interacting tasks.

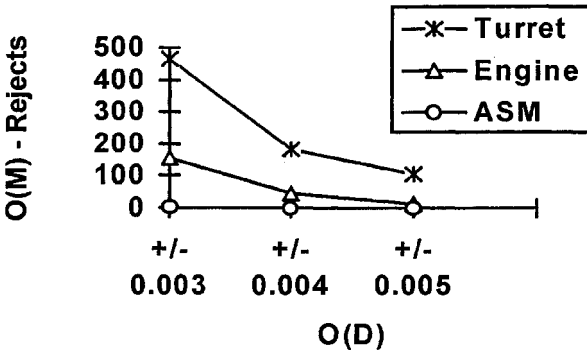


FIG. 6. Interaction between design and manufacturing.

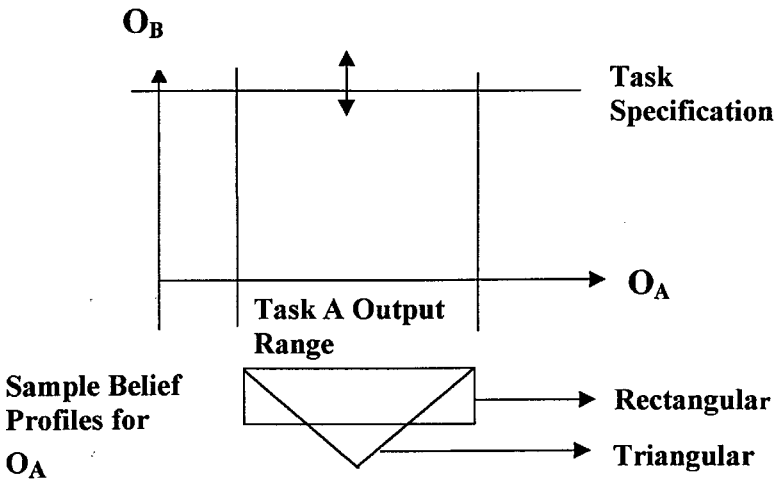


FIG. 7. A generic task interaction graph.

We call this graph a task interaction graph. The steps involved in graphically de-coupling two coupled tasks, A and B, are as follows. Without loss of generality, we are assuming that task B depends on information from task A.

- (1) Calculate the output of task B based on different possible values of task's A output.
- (2) Associate a belief function (or assign probabilities) to the occurrence of possible O_A outcomes.
- (3) Plot the task interaction graph (i.e. values of O_A versus O_B).
- (4) Plot the S_B line on the same graph.
- (5) Check the de-coupling region and see if it satisfies the specifications of task B. If it does, then the two tasks are uncoupled. If it does not, then adjust the specification line of task B to achieve de-coupling or virtual de-coupling.

Further research will be required to generalize the method for multiple tasks (i.e. more than two interacting tasks). In such cases, optimization techniques instead of a graphical solution might be explored.

8. Summary and Conclusion

The objective of this research paper was to investigate new methods for de-coupling coupled or interdependent design tasks. Engineering specifications management, a new tool that helps in design task de-coupling, was developed. The purpose of engineering specification management is to involve and utilize the specifications imposed on design tasks in the de-coupling process.

This research paper makes a contribution to the concurrent engineering literature by drawing attention to the utilization of engineering specifications in design process de-coupling, in an attempt to increase the concurrency of a design. The paper introduced the concept and foundations of engineering specification management for two coupled tasks and formulated the fundamental equations needed for such analysis. We also presented a framework and a graphical tool for design process improvement through better understanding and management of task specifications.

In conclusion, two coupled tasks can be de-coupled if, and only if, the fluctuations underlying the output of task j (feeding task) can be tolerated by the specifications of the other task i (being fed). If this scenario applies, then the specifications of task i have to be modified (widened or increased) by an amount proportional to the possible deviation in task j (assuming there is no interaction between the feeding tasks). In terms of the earlier design example, we had to increase the specification of the manufacturing task to 465 rejects in order to produce an uncoupled design and to 184 rejects in order to virtually de-couple the design process.

REFERENCES

- [1] CLAUSING, D. (1994) *Total Quality Development: A Step-by-Step World-Class Concurrent Engineering* (New York, ASME Press).
- [2] KUSIAK, A. & WANG, J. (1993) Efficient organizing of design activities, *International Journal of Production Research*, 31, pp. 753–769.
- [3] SINGH, N. (1996) *Systems Approach to Computer-Integrated Design and Manufacturing* (New York, Wiley).
- [4] LAWSON, M. & KARANDIKAR, H. (1994) A survey of concurrent engineering, *Concurrent Engineering: Research and Applications*, 2, pp. 1–6.

- [5] HA, A. & PORTEUS, E. (1995) Optimal timing of reviews in concurrent design for manufacturability, *Management Science*, 41, pp. 1431–1447.
- [6] KRISHNAN, V., EPPINGER, S. & WHITNEY, D. (1991) Towards a cooperative design methodology: analysis of sequential design strategies, *Proceedings of the 1991 ASME Conference—Design Theory and Methodology* (New York, ASME).
- [7] LOCH, C. & TERWIESCH, C. (1998) Communication and uncertainty in concurrent engineering, *Management Science*, 44, pp. 1032–1048.
- [8] SMITH, R. & EPPINGER, S. (1990) Modeling design iterations, *Working Paper No. 3160-90-MS* (MIT, Sloan School of Management).
- [9] SMITH, R. & EPPINGER, S. (1997) Identifying controlling features of engineering design iteration, *Management Science*, 43, pp. 276–293.
- [10] BJORKE, O. (1989) *Computer-Aided Tolerancing* (New York, ASME Press).
- [11] CHASE, K. & GREENWOOD, W. (1988) Design issues in mechanical tolerance analysis, *Manufacturing Review*, 1, pp. 50–59.
- [12] PETERS, J. (1988) Tolerancing the components of an assembly for minimum cost, *Transactions of ASME, Journal of Engineering for Industry*, 110, pp. 232–235.
- [13] ZHANG, C. & WANG, H. (1998) Robust design of assembly and machining tolerance allocations, *IIE Transactions*, 30, pp. 17–29.
- [14] PARK, S. (1996) *Robust Design and Analysis for Quality Engineering* (London, Chapman and Hall).
- [15] PHADKE, M. (1989) *Quality Engineering Using Robust Design* (Englewood Cliffs, NJ, Prentice Hall).
- [16] SUH, N. (1990) *The Principles of Design* (New York, Oxford University Press).
- [17] STEWARD, D. (1981) *System Analysis and Management: Structure, Strategy and Design* (New York, Petrocelli Books).
- [18] STEWARD, D. (1991) Planning and managing the design of systems, *Proceedings of Portland International Conference on Management of Engineering and Technology*, Portland, OR, USA, 27–31 October 1991.
- [19] EPPINGER, S.D., WHITNEY, D.E. & GEBALA, D.A. (1990) Organizing the tasks in complex design projects, *ASME Design Theory and Methodology Conference*, Chicago, September 1990.
- [20] ULRICH, K. & EPPINGER, S. (1995) *Product Design and Development* (New York, McGraw-Hill).
- [21] MONTGOMERY, D.C. (1991) *Design and Analysis of Experiments* (New York, Wiley).