Concurrent Engineering: Research and Applications (1993) **1**, 159–169

CONCURRENT ENGINEERING: Research and Applications

The Development of Virtual Concurrent Engineering and its Application to Design for Producibility

J. Schmitz and S. Desa*

Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA

Received 5 October 1992; accepted in revised form 26 March 1993

It is well known that so-called concurrent engineering is a desirable alternative to the largely sequential methods which tend to dominate most product-development methods. However, the proper implementation of a concurrent engineering method is still relatively rare, due in part to unreliable guesswork, poor knowledge structuring and utilization, and the lack of integrated, global decision making. Thus, to remedy the current shortcomings, we propose an initial, straightforward theory for product development, which is based on properly-defined concurrent engineering principles. After establishing the overall goal for product development, we formulate an objective that a product-development method must meet. This objective then leads to three fundamental criteria, which basically govern where concurrent engineering must be implemented, how consistent communication between different domains must be carried out, and how to structure and network the vast amount of expert knowledge in an effective, feasible manner. The product-development objective and the three criteria guide the establishment of a feasible computer-based implementation of concurrent engineering, called virtual concurrent engineering (VCE). The effectiveness of VCE is demonstrated by applying it to refine a method, called design for producibility (DFP), that integrates the design and manufacturing stages of product development. Two elements that are crucial to the success of DFP are the producibility cost function network, and the software package AUTOPROD (Automated Producibility). The refined DFP method has been successfully applied to concurrent product and process design in three domains: stamping, forming, and machining.

Keywords: virtual concurrent engineering, design for producibility, product development, cost function.

1. Introduction

Product development has traditionally consisted of a set of distinct stages, performed more or less in a specified sequence. One basic sequential structuring of these product development stages, called sequential engineering (SE), is shown in Figure 1. (In this structuring, the conceptual design stage represents the synthesizing of the preliminary design of a product that meets the established functional specifications; the embodiment design stage represents the specification of the detailed embodiment of a design that can directly be passed to ensuing stages, such as manufacturing.) In sequential engineering, each stage is essentially performed in isolation resulting in a costly, time-consuming, and inefficient product-development cycle.

Therefore, attempts have been made to more closely link these product-development stages, in particular to incorporate knowledge and expertise from stages that are "downstream" from the design stage so as to help the designer avoid high costs and difficulties in, say, manufacturing and assembling the product. However, as shown in Schmitz [1], such methods (e.g. so-called design for manufacturing) have similar shortcomings to sequential engineering. Thus, more recently, there have been implementations of "concurrent engineering" in which multi-disciplinary product-development teams from all stages of product development interact during early design [2]. While some success has been met with these methods, they still have many shortcomings, such as unreliable, uncertain guesswork on product-development requirements and consequences; inadequate communication that reduces the efficiency of meetings between engineers of different domains of expertise; and a lack of structure in the expert knowledge that is needed to integrate critical product-development issues and parameters.

To resolve the shortcomings of these existing methods, we have taken a broader, more general view of product development by studying the goals and the parameters that constitute successful product development, as well as the intra- and inter-relations between all the stages in product development. Based on these findings, an initial theory on product development is proposed, consisting of an overall goal for product development, a specific objective for a product-development method, properly-defined concurrent engineering, and three basic criteria (Section 3.1) to guide the structuring of a successful product-development method. These three simple and seemingly obvious criteria need to be stated because, despite being critical to successful product development, they are largely violated by most product-development methods. They can be used to both construct an ideal concurrent engineering method (Section 3.2) and also to test whether an existing development method is truly concurrent. While ideal concurrent engineering is infeasible to realize, the use of a properly-structured computer environment leads to a powerful and feasible implementation of concurrent engineering, called virtual concurrent engineering (VCE), which is described in Section 3.3.

---

*Currently at the Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, New York, USA.

1063–293X/93/030159 + 11 $08.00/0

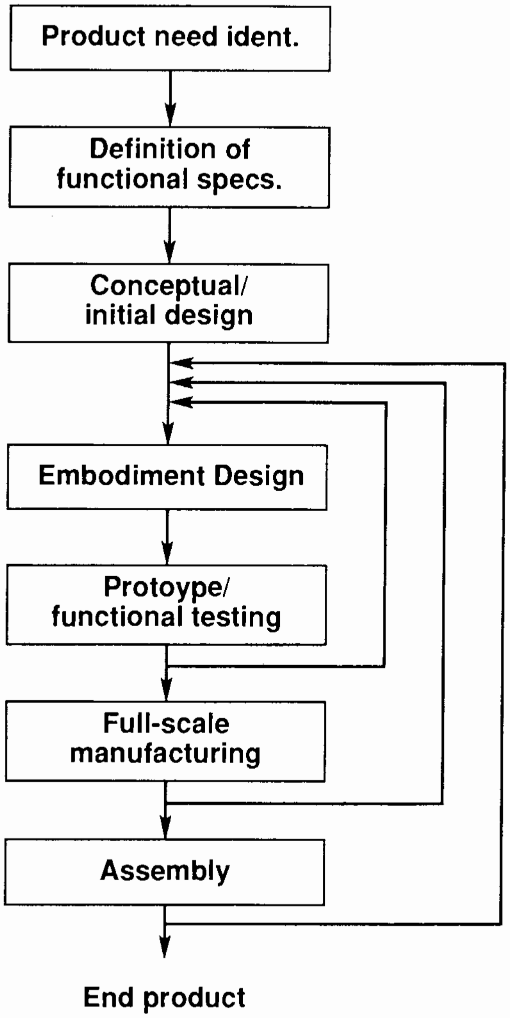


Figure 1. Typical sequential engineering.

To demonstrate the concepts and the applicability of virtual concurrent engineering, we used it to refine our earlier work [3], [4] in the design and manufacture of stamped products. The application of VCE to the integration of the design and manufacturing stages of product development is termed design for producibility (see Section 4.1). The producibility cost function network (see Section 4.2), which represents the basic structuring of the manufacturing process from the viewpoint of design, is the crucial element of design for producibility (DFP). Using the automated producibility package AUTOPROD (see Section 4.3), the refined DFP has been successfully applied to complex, precision, planar stamped products (see Section 4.4) as well as to 3-D formed and machined products [12].

2. A brief research review

In this section we first briefly survey relevant product-development methods that address concurrent engineering and/or the integration of design and manufacturing, then enumerate certain critical limitations of these methods from the standpoint of true concurrent engineering, and finally set the stage for the establishment of a concurrent engineering product-development method capable of resolving these limitations.

A concurrent engineering system that attempts to integrate design and manufacturing is described in Wei *et al.* [5]. In this system, a CAD/CAE/CAM network uses design parameterization techniques to compare design alternatives and arrive at a "better rationalized design" [5]. Production time and cost estimates are performed and made available to the product designer. In addition, during the course of design evaluation, a process plan is derived for possible use by the manufacturing engineer. The computer-based concurrent engineering method described in Altenhof *et al.* [6] permits the designer to vary design parameters and specifications in order to optimize a design with respect to an evaluation function based on costs, producibility, and maintainability. Computer-automated design methods and tools, based on "designing-with-features", have been developed to integrate design and manufacturing [7], [8]. Design for manufacturability tools have been developed in several domains, in particular sheet-metal stamping [9], forming [10], and injection molding [11].

While the above product-development methods have made valuable contributions to advancing the state of the art and knowledge in concurrent engineering, they posses one or more of the following major limitations:

1. The designed artifact has to be created in terms of special primitives and/or features that restrict the generality and complexity of the geometry that can be dealt with using these methods.
2. Modeling systems used by methods that require the designer to create and input designs using restrictive primitives and features are very tedious and error-prone in use.
3. There is a lack of the explicit knowledge structuring that is necessary in order to relate product-design parameters and specifications to the factors (like manufacturing costs, reliability, time) that govern successful product development.

The above drawbacks, in addition to those mentioned in the previous section, are also present in most concurrent engineering and design-for-manufacturability methods currently employed in actual practice [1], [3], [12].

The limitations of the above product-development methods and tools dictated what research activities were needed to define and establish a truly effective product-development methodology. Using precision planar stampings as our domain of application, we focused on the embodiment design and manufacturing stages, and how to gather, interpret, and structure the limitations and activities of the manufacturing process in such a way as to help the designer successfully develop products. The resultant structuring of design and manufacturing in the context of stamped products [1], [3], [4], [12] led to the ideas, methods, and tools discussed in the following sections.

3. Virtual concurrent engineering

In this section, virtual concurrent engineering (VCE) is developed in a logical progression which starts with the statement of the goal of successful product development. This leads naturally to a careful definition of concurrent engineering and the derivation of three simple criteria which dictate and guide the formation of a truly concurrent engineering method. These criteria when incorporated in a properly-structured computer environment result in a feasible realization of concurrent engineering called virtual concurrent engineering. The application of virtual concurrent engineering to the integration of the design and manufacturing stages of the product-development cycle leads to the formation of the design for producibility (DFP) method. A crucial element of the DFP method is the producibility cost function network (PCFN), a structured network that organizes the manufacturing process from the standpoint of producibility, i.e. the product-development parameters that are controllable in the manufacturing stage.

3.1. Three simple criteria for establishing a successful product-development method

The desired overall *goal* of successful product development is to develop a high-quality and high-performance product reliably and rapidly, while minimizing labor, materials, and overall cost. For a product-development method to meet this goal, costly and time-consuming re-iterations, due to unforeseen problems (in manufacturing, testing, assembly) after full-scale production begins, are strictly to be avoided. Therefore, any development method which realizes the overall goal of product development must satisfy the following *requirement*: the product-development method must be *structured* such that it allows for the global (qualitative) optimization of the combination of product performance and quality, development reliability, costs, labor, materials, and time at the *embodiment design* stage, *before* full-scale commitments in further product development and production are made.

As discussed in Sections 1 and 2, current sequential engineering and concurrent engineering (CE) methods do not meet this objective. In order to develop a concurrent engineering method which does satisfy the requirement for the product-development method, we first define concurrent engineering as follows.

Concurrent engineering is the integration of the activities of engineers from each stage of product development who, working together, perform simultaneous, interactive, real engineering (in-depth reasoning, designing, testing, building, computing) in order to establish a rational basis for global qualitative optimization of the product design.

Again, it must be emphasized that guessing, assuming, and projecting on future development results and requirements do not constitute concurrent engineering as defined above.

We now discuss how the concept of concurrent engineering, as just defined, can be used as the basis for the construction of an ideal product-development method. We have determined that there are three straightforward but basic criteria for successful product development, which in combination serve two fundamental purposes: they can determine whether a proposed product-development method is a concurrent engineering method, and they can also guide the establishment and structuring of an ideal concurrent engineering method.

About 70 to 80% of product-development costs are affected by decisions made at the product embodiment design stage [7], [13]; therefore, this is the natural stage at which to draw in and integrate diverse product-development expertise for the purpose of attaining an optimal balance between all the product-development parameters. Based on this observation, Criterion 1 can be stated as follows.

Criterion 1 (C1): for concurrent engineering of a product, all resources necessary for globally (qualitatively) optimizing the product design should be concentrated and integrated at the product embodiment design stage. (These resources include manufacturing, performance testing, and assembly engineering expertise.)

Because Criterion 1 requires that engineering from different product-development stages be performed simultaneously at the embodiment design stage, there needs to be clear means of communicating between these stages. Each stage of product development has its own set of primitives, which constitute its activities, methods, and, most importantly, language. For clear communication between the various stages, there needs to be a rich, common language with which each stage of product development can communicate. Thus, in the first part of Criterion 2 (stated below) we specify the common language base which should be used for all inter-stage communication. The second part of Criterion 2, which addresses the actual process of communicating between development stages, calls for the establishment of mapping schemes that relate the common language base to the primitives specific to each development stage or domain.

Criterion 2 (C2a): the product design (embodiment) drawings (i.e. the primary specifications) and secondary specifications (materials, tolerances, etc.) provide the common basis language for consistent, unambiguous communication between development stages.

Criterion 2b (C2b): engineers from each product-development stage must derive design interpretation and domain-mapping schemes to relate the primary and secondary design specifications to their respective domains of reasoning.

Criterion 2 is essential for the proper construction of computer-based product-development methods. The overall network for inter-stage communication is shown in Figure 2. It should be

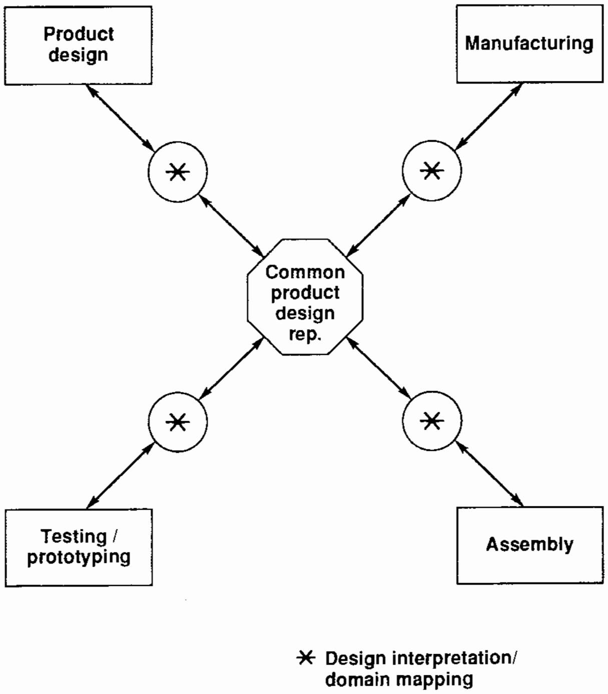


Figure 2. Inter-stage communication.

noted that each domain-mapping scheme (see Figure 2) is unique to its respective development stage. Product-development methods that require the engineer to "design with features" [5], [7], [8] essentially violate Criterion 2a because they force the designer to use very specialized features.

The first two criteria specify that concurrent engineering must be performed during the embodiment design stage, and that the embodiment design specifications and the domain-mapping schemes provide the means of inter-stage communication. The third criterion specifies what should be done after one has mapped the product embodiment into a specific development stage or domain. Recall that the requirement for a successful product-development method is that it must allow a product to be optimized with respect to various development parameters, such as manufacturing reliability and assembly costs. Thus, in order to be able to control and optimize these key parameters, Criterion 3 (stated below) dictates that each product-development stage must be *explicitly structured* and then represented by (a network of) cause–effect relations that enable one to reason about the effect of design specifications on the key product-development parameters. One of the critical shortcomings of the majority of proposed and/or actual product-development methods is the lack of proper knowledge structuring caused by inadequate understanding of the interdependency of the decisions and activities (e.g. process planning, tooling) that occur in real product development.

Criterion 3a (C3a): for successful product development, structure and link the activities and decision making within each product-development stage and establish the relevant cause–effect relations that, when integrated with the mapping schemes from Criterion C2b, yield an explicit link between design specifications and important product-development parameters (such as costs, reliability, and cycle time) associated with manufacturing, testing, and assembling the product.

Criterion 3B (C3b): in order to determine the relevance of specific product-development information or "knowledge" (i.e. in order to determine whether a specific engineering activity or decision-making step is relevant to the network of cause–effect relations), one must test whether such information can be controlled (directly or indirectly) by the product design specifications, and whether it affects the critical product-development parameters (e.g. reliability, performance, manufacturing costs).

The cause–effect relations required by Criterion C3a allow the designer to see the consequences of his design decisions in the global (qualitative) optimization process (e.g. he can weigh product performance versus manufacturing costs). Also, useful and effective re-design suggestions can be generated from these relations since an undesirable outcome can be traced back to its causes in the embodiment design specifications. The issue of determining relevance, addressed in Criterion C3b, is a key one because it resolves the problem of trying to organize and structure an overwhelming amount of information (or knowledge) in each development stage.

In combination, the three criteria just enumerated can be used to develop and test product-development methods to ensure that they are properly structured for meeting the objectives of developing successful products through extensive investigatory engineering steps performed and integrated at the design embodiment stage. We next discuss an ideal but unrealizable product-development method that can be constructed directly from the application of the three criteria; a method we call ideal concurrent engineering.

3.2. Ideal concurrent engineering

As guided by the objective for product-development methods, ideal concurrent engineering (ICE) aims to prevent re-work and re-iterations after the product design is embodied, and ICE also aims to provide a forum for the global (qualitative) optimization of the product-development parameters. This objective will be met by constructing ICE in accordance with the three criteria for product development.

According to Criterion 1, engineers from performance testing, manufacturing, and assembly join the product designer at the embodiment design stage to form a simultaneous engineering team. After detailed embodiment design specifications, as required by Criterion 2a, are determined by the product designer to satisfy the previously-established functional specifications, the rest of the engineering team first interpret and then map these general design specifications into their respective domains, as required by Criterion 2b. The engineers then perform the necessary steps to evaluate the design with respect to performance testing, manufacturing, and assembly (these steps are representative of the cause–effect relations referred to in Criterion 3a). Exactly which steps need to be performed are guided by Criterion 3b; that is, by those cause–effect relations which (directly or indirectly) affect the relevant product-development parameters. In order to estimate manufacturing reliability, for example, the manufacturing engineer may need to design and build the tools that would be used if the current design were to be manufactured (since from the cause–effect relations it has been established that tool wear and tool breakage help determine manufacturing reliability). He may then have to subject these tools to conditions similar to those that would exist upon full-scale manufacture. These manufacturing results, as well as the results from performance testing and assembly, can be relayed back to the designer through the domain mappings required by Criterion 2b. Also, the mappings combined with the cause–effect network from Criterion 3a can be used to relay re-design suggestions to the designer. The designer then interprets these suggestions in terms of the functionality that was established previously and, together with the engineering team, makes compromises and decisions regarding product functionality versus development costs, reliability, quality, and feasibility. Appropriate design changes are made, and further iterations of design evaluation with respect to the development stages are carried out until a qualitatively optimal product design is obtained. The only remaining activities in product development after design completion are the actual manufacture and assembly of the product. Thus, both parts of the development method objective are met: a qualitatively optimal product design is established, and re-work and re-design after the start of full-scale production are eliminated.

3.3. Virtual concurrent engineering

It becomes quickly obvious that the ideal development method (ICE) just described is not feasible to implement without a computer environment. For example, designing and making the assembly equipment every time a product design specification is changed (which is necessary in our proposed development method to investigate the consequences of each specification) is too costly in labor, time, and materials. Therefore, to resolve the infeasibility of ICE, we apply the same three basic criteria to develop a feasible computer environment to implement concurrent engineering. First, Criterion C1 requires that real engineering steps must be integrated at the embodiment design stage in order to achieve concurrent engineering. Thus, when a computer tool is used, these engineering steps must be captured and embedded in the computer system for use during embodiment design. Criterion C3 dictates how engineering knowledge and procedures should be structured and linked, and also helps to determine what knowledge is relevant. Finally, automated mapping schemes to traverse from the common design specifications into manufacturing, assembly, or testing domains must be developed according to Criterion C2. Thus, instead of having expert engineers predicting and conjecturing the effects of design decisions, the computer tool, given an initial design, can perform, automate, simulate, or reason about engineering steps (e.g. manufacturing tool making), estimate costs and locate problems associated with manufacturing or assembling the current version of the product design, and produce a set of design modifications to improve the product. The result is a method we call virtual concurrent engineering (VCE), which directly arises from the three criteria described above and which can be defined as follows.

Virtual concurrent engineering is a computer-based product development method that uses concurrent engineering principles (the three basic criteria) in order to optimize a product design with respect to the parameters which govern the entire product-development process: VCE is a feasible realization of concurrent engineering, where the simultaneous, investigatory engineering steps (e.g. process planning, costing, stress analysis, assembly verification) that are needed for sound design decision making are performed by a properly-structured computer environment.

The actual implementation of VCE will be discussed in the next section.

4. Example of virtual concurrent engineering: design for producibility

We next demonstrate the application of VCE to the integration of the embodiment design and manufacturing stages of the product-development cycle. The application of virtual concurrent engineering to these two stages results in the design for producibility method, which will now be developed. The stamping of complex, precision, planar products will be used to illustrate our ideas, concepts, and methods.

4.1. Design for producibility

First, we need briefly to discuss several concepts specific to manufacturing. The manufacturing stage itself can be represented by a set of primitives, which consist of activities (e.g. tooling, costing), decision making (e.g. choosing the type of manufacturing equipment), and processes (e.g. the physical process of removing material from a workpiece). These primitives are all geared towards producing a final product as designed by the engineer. The set of product-development parameters which need to be optimized with respect to manufacturing are best captured by the concept of producibility, defined as follows.

The producibility of a design is a measure of the feasibility of manufacturing the designed part reliably, at low machine cost, with minimal labor and maintenance costs, and with a short design-to-manufacturing cycle time.

For products to be developed successfully, the aim is to maximize producibility, while still maintaining the necessary product functionality. From a thorough study of a given manufacturing process, one first establishes the constituents, or determinants, of producibility for that manufacturing process [3], [12]. Since collectively the values of these constituents yield the costs associated with manufacturing that part, we call each of these constituents a *manufacturing cost factor* (MCF). A list of the important manufacturing cost factors for planar stampings is given in Figure 4b. In order to assess the producibility of a designed part, one must simply compute the value of each manufacturing cost factor. In theory, the design with the highest (i.e. best) producibility is the one for which the sum of the (normalized) manufacturing cost factors is the minimum. A detailed discussion of producibility is given in Schmitz [12].

Design for Producibility is the part of virtual concurrent engineering that enables the engineer to develop a product that is (qualitatively) optimal with respect to producibility, and that still maintains the required functionality. In order to optimize producibility, the MCFs must be related to the product design specifications. To this end, the first step is to structure the manufacturing stage in a higher-level manner that establishes the main steps within the manufacturing stage, and that provides the basic, overall sequence of the activities, engineering, and governing principles within the manufacturing stage. After study of real-life manufacturing environments, we concluded that the manufacturing stage (generally) consists of the following five steps: design interpretation (in order to perform manufacturing reasoning), process planning and tool design, tool fabrication, product manufacturing, and final costing (see Figure 3). In the design interpretation step, the manufacturing engineer interprets the design drawings in terms of data structures that allow manufacturing reasoning to be performed. Only after clearly under-

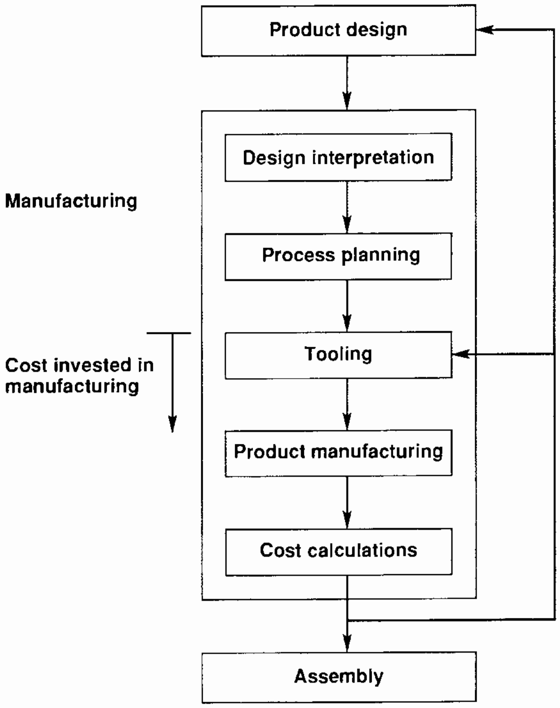
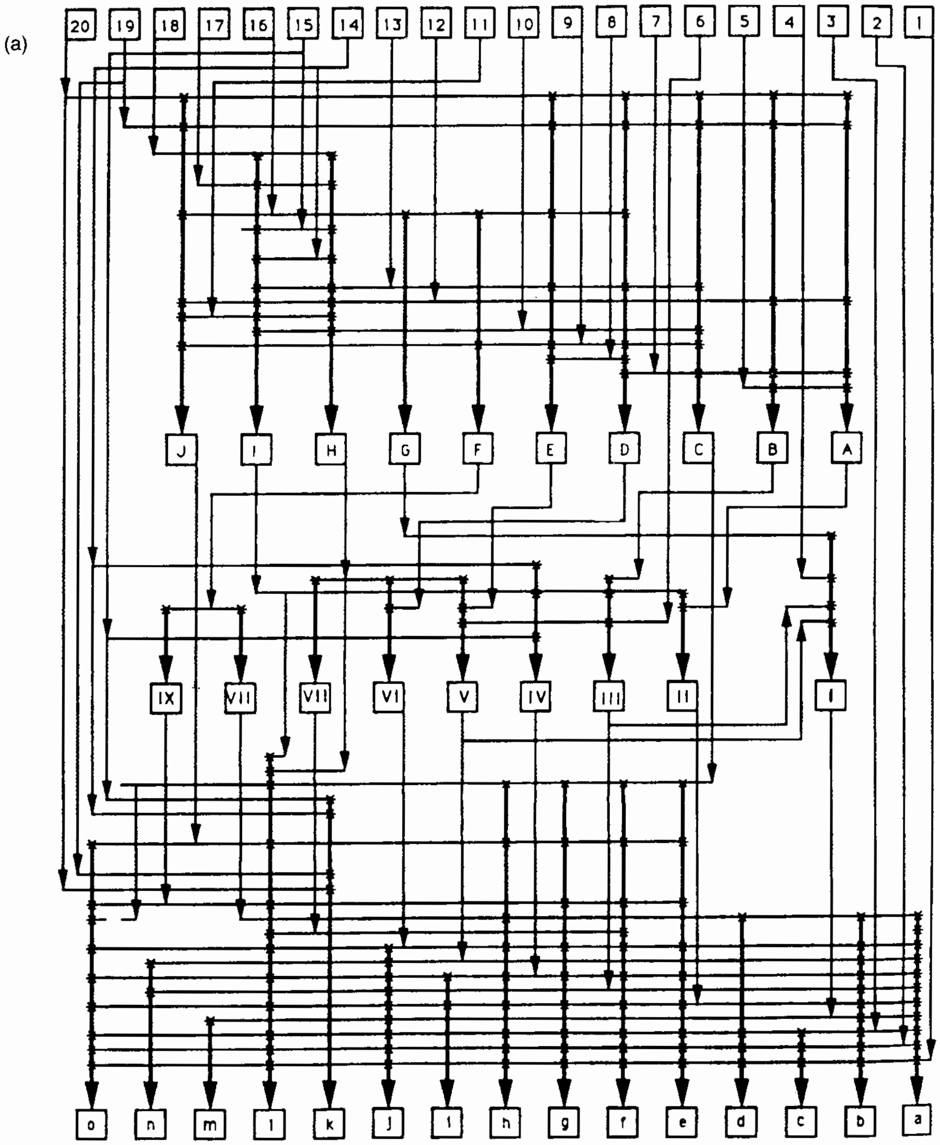


Figure 3. Basic manufacturing steps.



(b)

Manufacturing features, attributes, and secondary specifications

1. Tolerance specified for two holes with respect to each other
2. General tolerance specified for the contour of each hole
3. General tolerance specified for the blanking boundary
4. Part ductility
5. Sudden change in leg width (leg reduction factor)
6. Number of parts desired
7. Leg minimum width
8. Sudden change in notch width (notch reduction factor)
9. Notch minimum width
10. Total notch cavity angle (if more than 180 degrees, notch is concave)
11. Total leg cavity angle (if more than 180 degrees, leg is concave)
12. Convex corner radius
13. Concave corner radius
14. Hole contour
15. Outer-boundary contour
16. Web minimum width
17. Hole symmetry
18. Blank symmetry (including checking for equal holes)
19. Part ultimate strength
20. Part thickness

Process variables

A. Punch compressive stress
B. Punch cyclic stress
C. Punch manufacturability
D. Die compressive stress
E. Die cyclic stress
F. Stock in-plane stress (lateral)
G. Stock shear stress
H. Die-section complexity
I. Punch contour complexity
J. Die-section manufacturability

Intermediate cost factors

I. Burrs
II. Punching-tool breakage
III. Punching-tool wear
IV. Punching-tool buckling
V. Die-section wear
VI. Die-section breakage
VII. Die-section instability
VIII. Stock in-plane distortion (lateral)
IX. Stock in-plane breakage

Manufacturing cost factors

a. Manufacturing reliability
b. Inspection time
c. Need for piloting
d. Stock requirements
e. Number of punching tools
f. Number of die pieces
g. Number of stations
h. Stripper requirements
i. Special punching-tool holders
j. Tool replacement and tool sharpening
k. Press tonnage
l. Tooling time for initial press construction
m. Deburring
n. Finish grinding
o. Secondary operations (drilling, milling, . . .)

Figure 4. (a) Producibility cost function network (PCFN) for stamping. (b) Key for PCFN parameters in (a).

standing the required activities within the manufacturing stage were we able to define (as given in Schmitz [12]) a set of manufacturing features (MF) and feature attributes (FA), such as holes, webs, notches, legs, and corners (see Schmitz [12] for a complete discussion). One of the advantages of defining the set of manufacturing features is that they serve as the basis for a domain-mapping scheme (formally defined in Schmitz [12]) that relates the design drawings to features on the basic parts of the process plan. Thus, Criterion 2b, which (in this case) requires that the product design domain be mapped to the manufacturing domain, is satisfied.

The next basic manufacturing step is process planning, which needs to be captured in order accurately to compute costs associated with manufacturing the product. For stamping, process planning takes many years of practical experience. After extensive study (from practice and from the literature), trial and error, and verifications, we were able to develop a process-planning algorithm [12] accurately to generate process plans for planar stampings of any complexity. The resulting process plans, including punch design, die-section design (including die-splitting lines), and strip layout design for progressive dies, have been verified in actual manufacturing environments and are extremely accurate.

The final three manufacturing steps (see Figure 3), namely tool fabrication, product manufacture, and costing, were all studied, parameterized, and structured in great detail to yield the producibility cost function network (PCFN), which is discussed in the following section.

A general and detailed procedure for applying DFP to any manufacturing domain is given in Schmitz [12].

4.2. The producibility cost function network

The systematic structuring of manufacturing process knowledge in order to meet the requirements of Criterion 3a results in the generation of a producibility cost function network (PCFN). The producibility cost function network serves several purposes. First, the cost functions that constitute a PCFN are used to evaluate a proposed process plan (as designed by the computer environment, discussed below) to determine if that process plan is the most appropriate one (i.e. the most reliable, cost-effective plan) to manufacture the given product design. If a process plan fails the cost function evaluation, then the cost function is used to guide the generation of a more feasible process plan. The PCFN is also used to compute values of the manufacturing cost factors for the given product embodiment once the process plan has been designed, and to generate redesign suggestions for improving the design. The first step in the creation of a PCFN is to generate entities called cost functions, defined as follows.

A cost function, consisting of complex, inter-dependent relations, heuristics, combinatorial logic, inferences, and computations, is a function whose inputs are parameters [manufacturing features (MF) and feature attributes (FA)] on a proposed process plan along with secondary specifications (e.g. tolerances, number of parts), and whose outputs are values of the manufacturing cost factors (MCF). More formally, if there are n manufacturing cost factors, then a cost function f_i, (i = 1,2, ... n) can be expressed as follows:

$$(\mathrm{MCF})_i = f_i[(\mathrm{MF})_1, (\mathrm{MF})_2, \ldots, (\mathrm{FA})_1, (\mathrm{FA})_2, \ldots (\mathrm{SS})_1, (\mathrm{SS})_2, \ldots)], \; (i = 1,2, \ldots n), \qquad (1)$$

where $(\mathrm{MCF})_i$ is the ith manufacturing cost factor; $(\mathrm{MF})_1$, $(\mathrm{MF})_2$, ... are the manufacturing features; $(\mathrm{FA})_1$, $(\mathrm{FA})_2$, ... are the feature attributes; $(\mathrm{SS})_1$, $(\mathrm{SS})_2$, ... are the secondary specifications.

The resulting set of n cost functions, described by Equation (1), has the structure of an interconnected network and is called the *producibility cost function network* (PCFN).

The relationships which constitute the PCFN model tool design steps, simulate tool fabrication and product manufacturing behavior and governing principles, and formally perform costing steps. It is because of this structured modeling of the manufacturing stage in terms of cause–effect relationships that it becomes possible for not only manufacturing costs to be computable, but also for these costs to be controllable by the product designer (a key requirement for proper concurrent engineering).

The structure of the producibility cost function network is best explained by reference to the PCFN shown in Figure 4, which was developed for planar stamping. The key for the symbolic parameters in Figure 4(a) is given in Figure 4(b). Essentially, the PCFN has four levels. At the input or lowest level of the PCFN

Table 1. Producibility evaluation for stamping the lead frame

Manufacturing cost factors	MCF VALUES		Suggestions
	Actual	Ideal	
(1) Spring stripper	Required	Not required	• Widen webs • Widen notches • Loosen tolerances
(2) Number of stations	7	1	• Reduce hole complexities • Remove notch concavities • Increase corner radii
(3) Number of punches	38	18	• See (2) above
(4) Inspection time	Often	Rarely	• Loosen tolerances • Widen holes
(5) Tool replacement	Possible (for 16 punches)	Small chance	• Increase hole widths
(6) Scrap	43% of stock	26% of stock	• Loosen tolerances

[see Figure 4(a)] are the *manufacturing features*, *manufacturing feature attributes*, and the *secondary specifications*, a list of these being given in Figure 4(b) for the case of stamping. The next level of parameters are the *process variables*, defined as those specific physical parameters which govern the actual manufacturing process as well as the fabrication of the tools. The next level of parameters are the *intermediate cost factors*, which represent those adverse physical consequences that occur during the actual manufacturing process if the values (or magnitudes) of the process variables are not in the appropriate ranges. The final and highest level of the PCFN is the set of the *manufacturing cost factors*, which are the constituents of producibility for the specific manufacturing domain under consideration. A list of process variables, intermediate cost factors, and manufacturing cost factors for the case of planar stampings is given in Figure 4(b). The arrows in Figure 4(a) joining the key parameters in the cost functions represent the computations, estimations, heuristics, process simulations, and other cause–effect relations that stimulate actual events and decisions made during the manufacturing stage.

Notice that the inputs to the PCFN [in Figure 4(a)] are indeed the manufacturing features and attributes on parts of the process plan, along with the secondary specifications (e.g. tolerances). Thus, these inputs to the cost functions are all either directly (the secondary specifications) or indirectly (the manufacturing features and attributes) controllable by the product designer. Using the domain-mapping scheme, the manufacturing features on the process plan can be mapped back to features on the product design drawings, which in turn can be interpreted by the product designer, and mapped into his set of functional features.

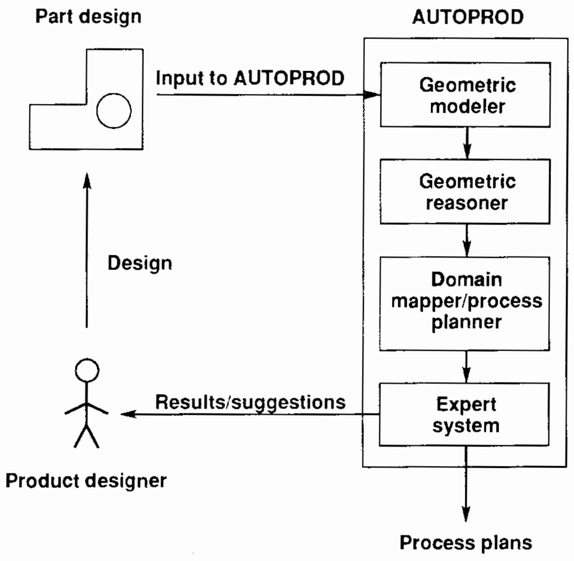
Also, notice that the outputs of the PCFN are the manufacturing cost factor values. In accordance with Criterion 3b, only information that can be controlled by the designer and that contributes to the manufacturing costs is integrated in the cost functions. This requirement is incorporated in the cost functions in the following manner. Every input to the cost functions does influence a parameter value at a higher level. Also, each process variable and intermediate cost factor has at least one input and one output. The input arrows mean that the process variables and intermediate cost factors are all controllable by the MFs, FAs, or secondary specifications. The output arrows reflect the fact that the process variables and intermediate cost factors in turn play a role in computing the values for the manufacturing cost factors. Finally, each MCF has at least one input, denoting that the MCFs are ultimately controllable by the MFs, FAs, or secondary specifications.

In addition to computing the values of the manufacturing cost factors, the cost functions also yield a set of re-design suggestions [not shown in Figure 4(a), but shown in Table 1] to improve product producibility. These re-design suggestions are all determined during the evaluation of the process plan by the cost functions. As high parameter values are computed (e.g. high tool stresses), notes are kept during the cost evaluations, and then form the basis for the re-design suggestions. Because of the cause–effect structure of the cost functions, it is possible to trace the cause of any high costs back to either manufacturing features and attributes or to secondary specifications.

4.3. Implementation of design for producibility

In the end, we have successfully studied, parameterized, and structured the basic activities that are relevant to DFP, as guided by the three criteria, in an elaborate cause–effect network, the PCFN, which, along with the domain-mapping scheme, relates design specifications to manufacturing costs. The next step is to embed the PCFN and the domain-mapping scheme into a computer environment that is capable of implementing DFP. We, therefore, next discuss the computer-based environment, called AUTOPROD, that automates design for producibility. AUTOPROD consists of four basic modules (see Figure 5), that we developed in accord with the three basic criteria of Section 3.1. (AUTOPROD is discussed at length in Schmitz [12] under its earlier, less descriptive name, PEP).

The first module of AUTOPROD is the geometric modeler, NOODLES, that was developed at the Engineering Design Research Center at Carnegie Mellon University [14]. It is a superset of boundary representation, which can model both manifold and non-manifold objects; this modeling scheme uses a common, general representational language and therefore satisfies Criterion 2a. The second module is the geometric reasoner, a software system consisting of procedures that we developed to define, recognize, and locate features on the generic model of the design, and to extract from these features all the attributes necessary for future reasoning (some details are provided in Schmitz [1], [12]). The third module is the domain mapper/process planner, which maps from the embodiment design domain to the manufacturing domains (as required by Criterion C2b), and which then designs the best process plan. The last module is the expert system, which evaluates the producibility of the design. The producibility cost function network, which is embedded in the expert system, is used to evaluate the process plans, estimate manufacturing costs, and, because of the precise cause–effect structuring of the cost



Outputs (from AUTOPROD):

1) M C F values that would be required to stamp current design
2) the "design source" of each high M C F value
3) re-design suggestions to improve producibility
4) Process plans

Figure 5. Design for producibility using AUTOPROD.

functions (as required by Criteria C3a and C3b), generate re-design suggestions.

The four components of AUTOPROD are shown in Figure 5, along with the closed-loop process for using the DFP method. First, the engineer designs an initial version of the product to satisfy a given set of functional specifications. The design is then entered into the AUTOPROD geometric modeler (see Figure 5). Next, the geometric reasoner identifies all the features (e.g. cavities, webs) and extracts feature attribute values. The domain mapper/process planner maps these features into the manufacturing domains (e.g. punching tools, die sections, stripper), and, using the mapping scheme as a basis, designs the process plan. The expert system then evaluates this proposed process plan and improves upon it until the best (i.e. most reliable, cost-effective) solution is determined.

The outputs of AUTOPROD are a detailed process plan, the values of the manufacturing cost factors and a set of redesign suggestions to improve the producibility of the product. The designer then has to determine what the implications of the redesign suggestions are to the functionality of his design. He must properly weigh product functional concerns against manufacturing costs and reliability, make the appropriate design modifications and enter the new design into AUTOPROD for evaluation. This closed-loop process allows the designer to qualitatively optimize the cost of manufacturing the product relative to product functionality.

4.4. Example: DFP of an IC lead-frame

In this section we will briefly discuss the application of our method and tool to the development of an actual integrated circuit (IC) lead-frame, shown in Figure 6.

A lead-frame is a planar, copper alloy product that is attached to an IC chip, with a specified number of leads to be inserted on a circuit board. The dimensions of the lead-frame in Figure 6 are (0.008" × 0.175" × 0.312"). Some of its functional specifications include exact fitting and proper conductance. (It is worth noting that a product designer may have used a mapping scheme to translate these functional specifications into the common, product-design specifications, such as sharp corners, tight tolerances everywhere, and "no burrs".)

As discussed previously, the product designer takes the functional specifications and the conceptual design, and synthesizes an initial design embodiment which is entered into the computer modeler of the PEP (in our system, as a list of coordinates, edges, and faces). Producibility analysis is performed by the PEP, which includes cost estimating, process planning, and generating re-design suggestions to improve the product's producibility. The process plans for stamping the lead-frame are completely derived by AUTOPROD and are shown in Figure 7. The producibility evaluation for manufacturing (stamping) the lead-frame design is shown in Table 1. Included in Table 1 are the actual MCF values, the ideal MCF values, and a few of the re-design suggestions to improve potentially the producibility of the lead-frame.

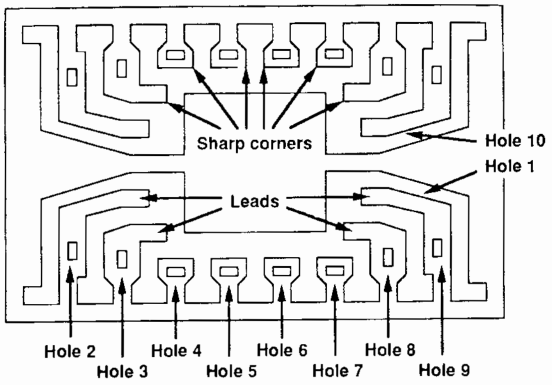
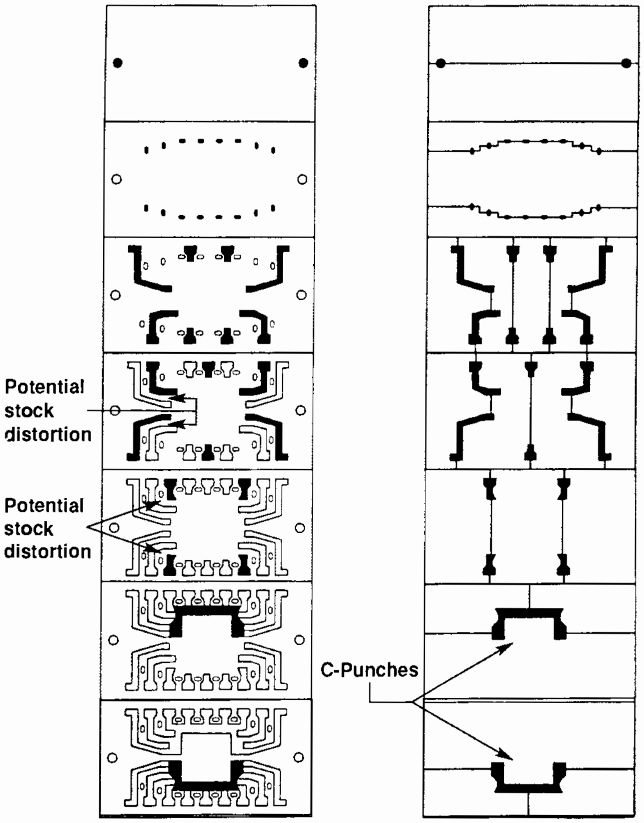


Figure 6. Cross-section of a lead-frame.



Strip layout

Die section designs

Figure 7. Computer-generated process plan for stamping the lead-frame.

The following issues, which need to be addressed during the implementation of design for producibility, are discussed in detail in Schmitz [12] within the context of the present example: how the "design sources" directly affect the manufacturing cost factors, how the MCFs must often be "juggled" to arrive at a collective optimal cost solution, how applying simple handbook rules often can result in an unnecessary decreased performance of the product, and how, in general, product functionality is integrated with manufacturing costs in arriving at an optimal design using DFP.

5. Summary and conclusions

Virtual concurrent engineering (VCE) is a computer-intensive method that, since it is structured and constructed in accord with the three basic criteria of Section 3.1, enables the engineer to optimize (qualitatively) the product design with respect to important product-development parameters such as performance, quality, development cost, and time. To demonstrate this property of virtual concurrent engineering, it was applied to refine earlier work [1], [3] in design for producibility (DFP), a product-development method that integrates design and manufacturing. The implementation of DFP for the concurrent product and process design of precision, complex, planar products [12] is briefly discussed in Section 4.4 and shown in Table 1 and Figure 7.

Specifically, the results of the comprehensive work in the domain of stamping clearly demonstrate the following practical uses of VCE (and DFP):

1. to help the designer develop a highly producible, yet functional, design;
2. to improve first-chance success in full-scale product development;
3. to help guide both experienced and novice manufacturing engineers;
4. to be part of a CAD/CAM network for automated tool manufacture (using N/C machines);
5. to help with scheduling, labor, and material estimates, machine requirements, etc.;
6. to help make more accurate and rapid bids.

DFP was also successfully applied, on a smaller scale, to the domains of (three-dimensional) forming and machining [12]. Two important contributions arising from DFP are the producibility cost function network (see Section 4.2), which restructures the manufacturing process from the viewpoint of the entire product-development process, and AUTOPROD, the automated producibility package.

Despite the extensive knowledge representation and automation of engineering procedures, we clearly recognize the need for the human element in implementing VCE (see Figure 5). Successful concurrent engineering of a product will ultimately depend on the effectiveness of the interaction of product development engineers with the virtual concurrent engineering environment envisioned in this paper.

References

[1] J. Schmitz, "Design For Producibility of Stamped Products: Methodology and Implementation", Master's Thesis, Carnegie Mellon University, Department of Mechanical Engineering, May 1989.

[2] R. Wheeler, "Small-Work Projects: Teamwork Counts More Than Computer-Based Tools", IEEE Spectrum, July 1991, pp. 32–33.

[3] J. Schmitz and S. Desa, "The Development and Implementation of a Design for Producibility Method for Precision Planar Stamped Products", ASME Design Automation Conference, Montreal, Canada, 17–21 September 1989, Advances in Design Automation-1989, Vol. 1, pp. 295–302, September 1989, to appear in the Trans. ASME, J. Mechanical Design.

[4] J. Schmitz and S. Desa, "The Application of a Design For Producibility Methodology To Complex Stamped Products", Symposium on Concurrent Product and Process Design, ASME Winter Annual Meeting, San Francisco, 10–15 December 1989, pp. 169–174.

[5] Y. Wei, G. Fischer, and J. Santos, "A Concurrent Engineering Design Environment For Generative Process Planning Using Knowledge-Based Decisions", Proceedings of the 1990 ASME Design Automation Conference, Concurrent Engineering of Mechanical Systems, Chicago, Illinois, 16–19 September 1990, pp. 35–45.

[6] J. Altenhof *et al.*, "Concurrent Mechanical System Design: a Computer-Aided Demonstration", Proceedings of the 1990 ASME Design Automation Conference, Concurrent Engineering of Mechanical Systems, Chicago, Illinois, 16–19 September 1990, pp. 103–109.

[7] T.J. Cunningham and J.R. Dixon, "Designing with Features: 'The Origin of Features'", Proceedings of the ASME Computers in Engineering Conference, 31 July – 3 August 1988.

[8] D.W. Rosen *et al.*, "Features and Algorithms for Tooling Cost Evaluation in Injection Molding and Die Casting", Proceedings of the ASME Computers in Engineering Conference, 1992, pp. 45–52.

[9] C. Poli, P. Dastidar, and P. Mahajan, "Design for Stamping: Part 1 – Analysis of Part Attributes that Impact Die Construction Costs for Metal Stampings", Proceedings of the ASME Design Automation Conference, Miami, 1991.

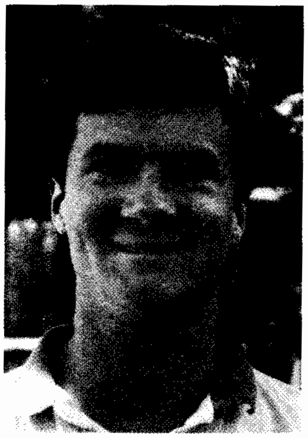
[10] S.K. Sitarama *et al.*, "A Computer-Aided Design System for Process Sequence in Axisymmetric Sheet Metal Forming", Technical Report, Engineering Research Center for Net Shape Manufacturing, Ohio State University, August 1988.

[11] K. Ishii, L. Hornberger, and M. Liou, "Compatibility-Based Design for Injection Molding", Symposium on Concurrent Product and Process Design, ASME Winter Annual Meeting, San Francisco, 10–15 December 1989, pp. 153–160.

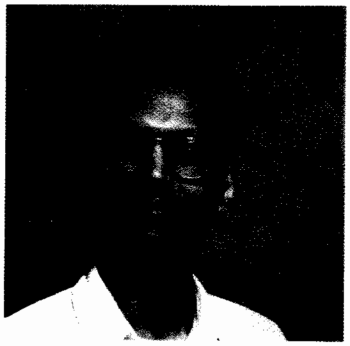
[12] J. Schmitz, "Virtual Concurrent Engineering: Theory, Implementation, and Application to Design for Producibility", Ph.D. dissertation, Department of Mechanical Engineering, Carnegie Mellon University, August 1992.

[13] J. Nevins and D. Whitney, Concurrent Design of Products and Processes, McGraw-Hill, 1989.

[14] L. Gursoz and F. Prinz, "Corner-Based Representation of Non-Manifold Surface Boundaries in Solid Modeling", Technical Report, Engineering Design Research Center, Carnegie Mellon University, 1988.



James M. Schmitz received his Bachelor's degree (1987) in Mechanical Engineering from Cornell University and his doctorate (1992) in Mechanical Engineering from Carnegie Mellon University. His areas of research include design, manufacturing, and concurrent engineering. His paper, coauthored with S. Desa, titled "Development and Implementation of a Design for Producibility Method for Precision Planar Stamped Products" received the 1989 ASME Design Automation Committee Award for Best Technical Paper.



Subhas Desa obtained his doctorate in Mechanical Engineering from Stanford University and has worked at Hewlett-Packard Company, Stanford University, and Carnegie Mellon University. His areas of research include kinematics, dynamics, control and design of electromechanical and robotic systems, design for producibility, and concurrent engineering. He is currently a visiting faculty member in the Sibley School of Mechanical and Aerospace Engineering at Cornell University.