Design Fusion:
A Product Life-Cycle View for Engineering Designs

by

S. Finger, M. Fox, D. Navinchandra, F. Prinz, J. R. Rinderle

EDRC 24-28-90 *C.* 3

Extended Abstract for

Design Fusion:

A Product Life-Cycle View for Engineering Designs

Susan Finger, Mark S. Fox, Dundee Navlndiandra
Intelligent Systems Laboratory
Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Friedrich B. Prinz, James R. Rinderle
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Submitted to IFEP WG 5.2
Workshop on Intelligent CAD

19-22 September 1988
University of Cambridge, UK

Design Fusion:

A Product Life-Cycle View for Engineering Designs

Extended Abstract

Susan Finger, Mark S. Fox, Dundet Navinchandra Friedrieh B. Prinz, James R. Rinderle

Carnegie Mellon University
Pittsburgh. PA 15213

Introduction

This paper discusses the underlying philosophy and approach for the Design Fusion system th^g '9 currently being developed at Carnegie Mellon University. The goal of the Design Fusion project & to create the underlying theories and methodologies for a computer-based system that will assist in creating mechanical designs which meet their function, cost, and quality requirements while simultaneously meeting the constraints imposed by life-cycle activities such as manufacturing and maintenance.

The Design Fusion system is based on three underlying concepts:

- Integrating life-cycle concerns through the use of views from multiple *perspectives** where each perspective represents a different life-cycle concern such as manufacture, distribution, maintenance, *etc*
- Representing the design space at different levels of abstraction and granularity through the use *offeatures,* where features are the attributes that characterize a design from the viewpoint of any perspective
- Using *constraints* at different levels of abstraction to guide the design process and using constraints to maintain consistency and propagate design decisions.

Using the concepts of perspectives, features, and constraints, the Design Fusion system generates, prunes, and tests design alternatives. A key element of the Design Fusion architecture is the concept *of degree of fusion** that is, the degree to which ^ g n_rirritinnt--iir simultaneously-generated and evaluatetfcbjuhe interacting perspectives. In Design Rision, aD perspectives may generate and test design alternatives at all levels of abstraction and ac every stage in the evolution of the design. Thus, Design Fusion is quite distinct from systems that use after-the-fact design critics to evaluate completed designs.

The design space can be viewed as a multi-dimensional space in which each dimension is a different life-cycle activity such as fabrication, testing, serviceability, reliability, etc. These dimensions are called *perspectives* because each dimension can be thought of as a different way of looking at the design. As a design evolves, the designer moves from one viewpoint in the design space to another and moves from one level of abstraction to another both within a perspective and across different perspectives.

By continuously viewing, commenting on, and intervening in, the evolution of a design from each of the perspectives, the constraints of the product's life-cycle are accounted for in the completed design. The design system must allow implicit functional requirements such as manufacture, assembly, or testing to be integn^irto the derijn at the appropriate time and at the ^^priate level of detail.

Design RtsiOD occtits not only in the ^pace of life-cycle concerns^ but also in the space of design methodologies. During the design of a piodua nuny differtr* strategies may be empk^ Based on preliminary studies of designers [1,2J. it is reasonable to believe that there are common, basic problem-solving mechanisms that underlie most design modes. Therefore, a design system should support different problem-solving modes and allow the designer to mate maloe smooth transitions am^ them.

Perspectives

A design that is created based only on Amctknal considerations often requires major design changes when life-cycle concerns such as assembly or serviceability are considered. Each life-cycle concern can be viewed as a *penptattt.* A perspective defines both a representation of design knowledge and methods for generating or criticizing design decisions. As a design evolves, ft can be viewed ftom qpny different perspectives, eg* Amction, fabrication, assembly, fawprrrion and testing, distributidL-ilett service, reclamation, or training."

Each perspective has two roles; to synthesize some portion of the artifact or to evaluate what has been synthesized. How designs are generated depends upon which perspectives are important. Any perspective may become more or less important as the design evolves. The more wdgbt given to a perspective* the greater role it plays in synthesizing the 4fV.^IK Tbe functional perspective lends to dominate at the outset but may later recede depending upon the state of evolution of the design and the component under consideration. Figure 1 shows a possible interaction among perspectives during the design of a turbine blade.

To generate acceptable designs fom the start each perspective must play an active role during design synthesis. So, the perspectives evaluate design decisions at every step and every level of abstraction. The *decree of fusion* is determined by the granularity of the decision that is evaluated TIie larger the decision step, the less Aision occurs. The boundary case occurs when evaluation is performed after the design ii complete. Consequently, design by perspective goes well beyond tbe simple notion of design critics Out evaluate designs only after they have been completely specified.

Features

The use of *features* is a key element of Design Fusion. Dixon f3J defines a feature as "any geometric fan or entity that is used in reasoning in one or more design or manufacturing activities" We use a similar, but broader definition. We define a feature to be a relationship among a set of elements of a design. Thus, features are not limited to being geometric entities nor are they limited only o the design tnd manufacturing perspectives. Features can be used in reasoning about a design from the viewpoint of •V perspective.

During the design process, the same product design looks quite different when viewed through die cxPenise of the different perspectives. Each perspective emphasizes particular aspects of the design and *PPresses certain details in order to evaluate and synthesize. In addition, as the design evolves, so does

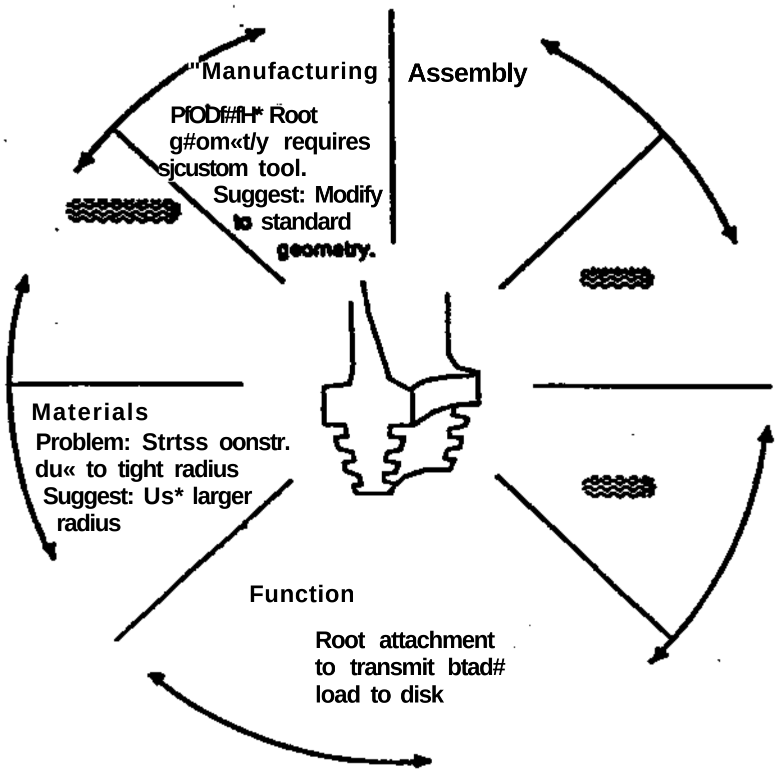

Figure 1: Each perspective generates, tests, and critiques design decisions as they are made.

the view from each of the perspectives: that is, what is emphasized and what is suppressed changes depending on the current state of the design. For example, selecting the diameter of a motor armature has quite different consequences depending on which perspective is viewing the design. The functional perspective may view the diameter as the primary determinant of the armature loading and the manufacturing (stamping) perspective may view the diameter as the determinant of the number of laminates that can be stamped from a stock sheet Hence, the *diameter* feature has different attributes and associated constraints is each perspective.

Features are hierarchical and may be composed from primitive features and constraints or from the underlying physical properties of the designed artifact Given a set of primitive features for each perspective and for each abstraction type (function, layout geometry, etc.). features may then be composed of other features or composed from underlying feature primitives. A major focus of our work on features is on the rules by which features are composed from primitive features and the relationships among features from different perspectives.

We view design as the successive refinement of features from high-level behavioral attributes, through schematics and layouts, down to topology and geometry. Specifying a feature may constrain the underlying description, but is not necessarily sufficient to specify all or even any of the associated physical properties. Features enable the designer to. focus on specific parts of the design, to perform detailed design of some parts, and to leave other parts of the design as more abstract features.

Constraints

to DesiptFWoiv we bring together differe* approaches to constraints to create a constraint-based design system, Son* critical issues that we address include the hierarchical repmcnurion of constraints, awtratesatisfMionaDdrelaxati^ constraint abstraction and constraint-directed design.

In the context of engineering design, a constraint can be thought of as a required relationship among design features and characteristics. Gxmraias may embody a design objective (e * weight), a physical law (eg. F» *ma)*, geometric compatibility (eg. matingofpamX production requirements (c.g. no blind boles), or any other design requirement Collectively, the constraints define what will be an acceptable design. The number, diversity, and variable context of constraints make finding an arrrptahte design a difficult task. furthermore, finding the design that satisfies aD the constraints is only possible when the constraint network represents aO design alternatives, is complete and nwtfftcnt, and results in a unique solution. These conditions are rarely. If ever, met If the constraint network it overoonstnined, no solution exists, and some constraints must be relaxed or the goals modified. If the network is underconstnûned, too many solutions exis^jnd oonstraints or goals must be added so that a design may be selected. It is not sufficient to satisfy a network of constraints; it is critical to kfcntity characteristics of the constraint network that influence the modiflcatktt of constrains and the |ieferences of the designer;

A large body of research exists on solving constraint propagation problems. These techniques provide a core of solution methods; however, a designer needs not just the solution, but also needs an understanding of the nature of the solution. In particular, a designer needs to understand how certain design decisions or variables were set, how those variables depend on other design variables, and the leverage that design variables and constraints have upon other design decisions. We address this need by providing not only a solution but also an explanation of the solution that tracks the dependencies in a constraint network and evaluates the impact of a decision on other design variables.

In design, a small set of constraints often is critical in determining many other design relations. The ability to identify and address these critical constraints early in the design process is important to the designer. As different perspectives impose new constraints on the design the importance of identifying bonle-neck constraints becomes even greater. We are currently exploring several different techniques for identifying these bottle-neck constraints.

Individually and collectively even the most detailed constraints affect preliminary design decisions. It is necessary to abstract from a complex network of constraints, those constraints that directly a/Tea Preliminary decisions. In most cases, however, a complete algebraic solution cannot be obtained. In (tec cases many techniques can be employed. One method is to identify differential rather than absolute ftbtionships among variables. In this way certain scaling relationships, for example, can be identified for ** designer. If it is not possible to provide an algebraic differential relationships, the constraint network **X be simplified using numerical dominance and domain dependent design practices. The final step is * introduce direct numerical methods to identify the numerical values of the differential relationships ****t design variables.

Each Perspective represents a life-cycle concern. During the design process, the perspectives introduce new constraints that guide the design by eliminating infeasible choices and by helping generate acceptable ones. The use of constraints to test design alternatives is well understood; however, we will use constraints to generate new alternatives. One approach to generating alternatives is the identification of a set of design decisions that satisfies the current set of constraints. By selecting design parameters that correspond closely with previously identified critical constraints, acceptable design can be identified. We will augment this strategy by transforming design constraints and features to reduce circularity in constraint network topology and to provide a less sensitive basis for constraint evaluation.

System Architecture

The architecture of the Design Fusion system is based on the blackboard model [4] and is shown in Figure 2. Each design version, composed of features connected by constraints, is represented on the *design blackboard*.

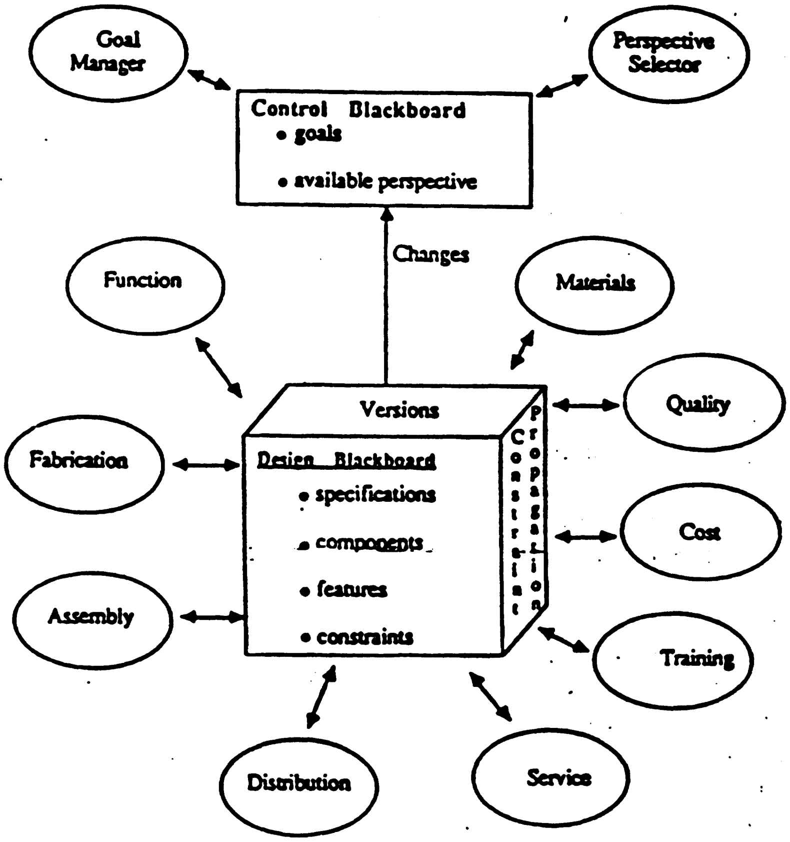

Figure 2: Design Fusion system architecture.

Perspectives are represented as kiwwW^ sources that can view the designs in parallel. Once a change occuixs anAb propagated, each perspective can criticize the change and can also request that control be given to it so that h can elaborate the design. Aawi0o/Wflctowniisusedtomanage which perspective is in control. Specifically. the knowledge sources associated with the control blackboard manage the design goals and determine which requests for control best match the current goal set

An integral put of the representation of a design is the design record. The design record trades the decisions that led to the creation of each part feature. A design decision is defined by the perspective that led to the decision, the type of processing that generated the decision, and the information upon which it was based. The design record also maintains a decision dependency netwotfc in order to support intelligent backtracking.

The problem solver *opportunistically [5]* moves from one perspective to another, from one level of abstraction to another, from one feature to another, and from one constraint to another. *The controlling perspective* is the one that leads the synthesis process. Non-controlling perspectives evaluate portions of the design at whatever level of opportunism is appropriate^

To generate acceptable designs, the non-controlling perspectives must play an active role as the design is created. For example, to guarantee ease of fabrication when the fttKtional perspective is in conSfoL the fabrication perspective must narrow the alternatives that can be synthesized. Thus, the controlling perspective is the generator of design states, while the non-controlling perspectives act as evaluators of these design states. The controlling perspective can and should shift depending on the salient characteristics at any point in the evolution of the design.

Conclusion

Work on the Design Fusion system is in its initial stages at Carnegie Mellon. The concepts that form the basis of Design Fusion are complex, and it will require a long-term research effort to make progress toward understanding the complex processes of design.

Design Fusion uses a feature-based representation to enable a design to be viewed from multiple perspectives and uses constraints to guide the generation of design alternatives to ensure that life-cycle concerns are explicitly incorporated in the design process. Design Fusion also entails the integration of multiple problem-solving methodologies so that a designer can move easily from one level of abstraction to another or from one perspective to another as the design evolves. It is our position that an intelligent CAD system must explicitly ftise life-cycle knowledge within the generation of the design.

References

1. Adelson* B. and Solowat E > g n i ^ Model of Software Design," Report 342, Department of Computer Sdence. Yak. 19S4.

X UHmin, D. a. Stauffer, L. A« and Dietteri ch, T. O, "PreHminaiy Results of m Experimental Study of Ibe Mechanical Design Process," *Proceedings from the NSF Workshop on the Design Process*. Waldron, M. *BjeL*. Ohio Stale Univerrity. Oakland, GA, Febraaiy S-10 1987. pp. 145-188.

3L Cunningham, J. J. and Dixon. J. IL, "Designing with Features: The Origin of Features," *198$ ASME Computers In Engineering Conference*, American Society of Mechanical Engineers. San Fnndsco, CA, July 31 - August 3 1988.

4. Ennan, L. D.9 Hayes-Roth, F., Lesser, V. IL, and Reddy, D. It, **The Heanay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty/* *Computing Surveys*. Vol. 12, No. % 1980, pp. 213-253.

5* Ow# P. S. and Smith S. F- "Viewing Scheduling as an Opportunistic Problem-Solving Process." in *Annals of Operations Research:-Approaches to Intelligent Decision Support*. Jeioslow, R. O.t ed^ Baltzer Scientific Publishing Co., 1987*