Autonomous Critics

by

S. Talukdar, M. Sapossnek, L Hou,
R. Woodbury, S. Sedas, S. Saigal

EDRC 18-13-90*

AUTONOMOUS CRITICS

S. N. Talukdar
M. Sapossnek
L. Hou
R. Woodbury
S. Sedas
S. Saigal

Engineering Design Research Center
Carnegie Mellon University
Pittsburgh, PA 15213
412 268-8778

ABSTRACT

Decisions made in the early stages of design processes can have profound effects on later stages. Often, information on these effects can be obtained only after great delays, by which time it is too late to use in the design effort. The goal of the work reported here is to develop and demonstrate technologies for shortening information feedback loops to the point where crucial analysis and evaluation information is made available to designers continuously and automatically, as their pieces of the overall design evolve. Our approach is based on the use of expandable libraries of autonomous programs called *critics*. Each critic keeps track of a developing design, or a piece of a design, evaluates the design from the viewpoint of a downstream stage, and signals the designer when it detects a flaw. The research issues are four-fold. First, each critic must be able to understand the representation schemes in which the designer is working. Second, each critic must be able to decide when to act. Third, each critic must be able to report its critique in terms the designer will understand. Finally, a distributed framework must exist to support the expanding library of critics.

A demonstration system with an initial set of three critics has been completed and is described in the paper.

INTRODUCTION

Japanese automakers are able to design new cars in about two-thirds the time taken by U.S. automakers. To find where the Japanese gain their advantage, Clark and Fujimoto [1] have broken the automobile design process into five stages. The average lengths of these stages in the U.S and Japan ace shown in Figure 1. Notice that the later stages take about th$ same amounts of ti/ne in both countries. But in the earlier stages the Japanese are much quicker. Many hypotheses[1] have been put forward to explain why this is so, but none to our knowledge, have been proven. Our hypothesis is that a large part of the Japanese advantage stems from their use of agents we will call *critics*.

The need for critics arises because neither human nor automatic designers can know Everything, and what a designer doesn't know can cause him, her or it to make very unfortunate decisions. For instance, stylists, who know little about manufacturing, often select shapes that are inordinately difficult to make. And two designers working simultaneously often produce incompatible designs, because neither knew what the other was doing. The purpose of critics is to prevent such happenings by widening the designer's view and expanding on his store of

knowledge. Critics use their knowledge to provide early warnings of conflicts that are developing with concurrent or downstream stages, and to point out behaviors that would compromise the performance of the object being designed. For instance, a critic with manufacturing expertise would steer stylists away from shapes that are difficult to manufacture.

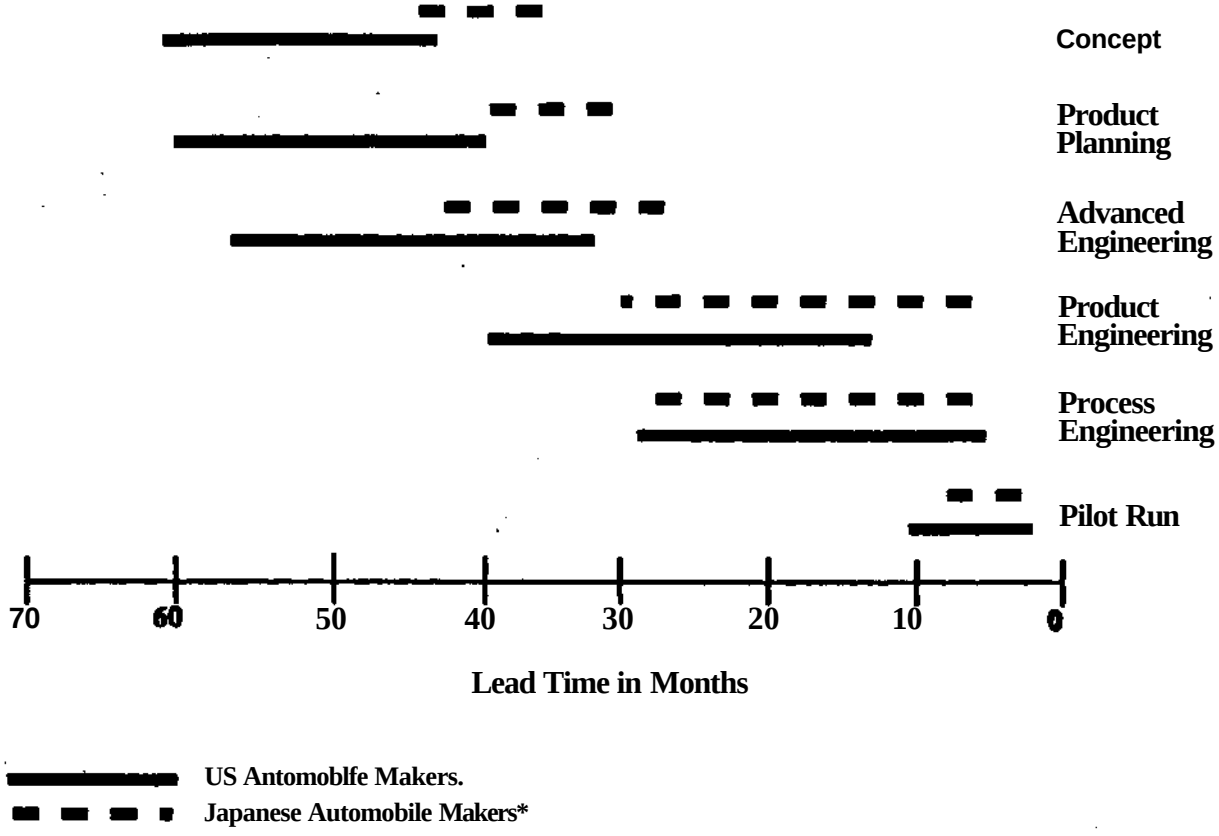

Figure 1. Stages in the Design of an Automobile (Adapted from Clark and Fujimoto [1])

Tb be useful, a critic must react to design decisions rapidly, while the decisions are still malleable. Also, the critic must be unobtrusive and not draw heavily on the time and energy of the designer. Tb expand on these points, think of a mechanical part, some of whose behaviors can be determined by finite element analysis. Two possibilities are for the designer to learn how to use a finite element program or to send the design to an outside specialist in fiinite elements. Neither possibility is attractive; the first because few designers could spare the effort required, the second, because an outside specialist could take weeks or months to complete an analysis, by which time the designer would probably have moved on to a new project To be useful, a critic would have to provide the finite element analysis without demanding any effort from the designer, and within minutes, so the designer could employ its results to improve her design.

We have been told that Japanese automakers assign the role of critics to sottie of their most experienced and competent engineers. Each engineer represents a different department or stage and follows the design process from start to finish. We do not know if such a system could be successfully implemented in the U.S, nor are we in a position to find out. However, the advantages of critics are obvious and it would seem that many, if not all, of their functions can be

automated* The remainder of this paper describes our work on automatic critics.

REQUIREMENTS FOR AUTOMATIC CRITICS

The following is a fist of attributes that we feel are necessary for an automatic critic to be useful to a designer or set of designers:

1. the critic must contain knowledge that goes beyond that of the designer,
2. the critic must be able to apply this knowledge quickly, so it can interact with the designer,
3. the critic must go about generating its reactions to the designer's work in a quiet and unobtrusive manner, without pestering him with questions or demanding great amounts of his time. This requirement implies that computationally intensive critics should reside in their own computers, critics should be self-activating or autonomous, and finally, critics should be able to understand the representations used by the designer, rather than requiring the designer to translate his work into their input formats.
4. the critic must be able to explain its results in terms the designer can understand;
5. the critic must be general enough to apply to a useful class of designs (i.e. it would not do if a new critic had to be created for each new part to be designed);
6. the infrastructure should be able to accommodate an arbitrarily expandable set of critics so that new critics can be added whenever necessary.

ASE: A PROTOTYPE CRITIC SYSTEM

We have implemented a prototype critic-based design system called *ASE* (Automated Simultaneous Engineering). ASE is a joint research project between Carnegie Mellon*s Engineering Design Research Center and General Motors[9] Inland Fisher Guide division. The initial domain of ASE is window regulator design. ASE consists of five components: a *synthesis system,* three critics: a *tolerance critic,* a *mechanical strength critic,* and a *kinematics critic,* and *FORS,* an integration framework.

These components, together with a design engineer, are intended to operate in the following manner: the designer interacts with the synthesis system to create a new design. Each critic observes the progress of the designer, and when appropriate, performs an analysis of the design. If the results of the analysis provide new and useful information, they are presented to the designer in an appropriate manner. The designer has control over the design, and serves to close the feedback loop from the critics. FORS provides a framework for controlling and interfacing the other components in the system. Additional details on ASE can be found in [2].

Each of these components is discussed in greater detail in the following sections.

FORS

FORS (Elexible QRganization£) [3] is our framework for integrating critics. It provides support for an extensible library of critics, is designed to work in a distributed computing environment, has an icon-based user interface, and has an open architecture making it suitable for use in different disciplines.

We visualize design activity as tracing paths through a directed graph called a *TAO graph.* Nodes in this graph represent *models,* arcs represent *operators.* A model is a partial description, view or aspect of the artifact being designed. Models can have multiple *representations;* these representations are informationally equivalent An operator, also called a *tool,* is an automatic or

manual procedure for transforming one set of models to another. Operators add and/or remove (i.e. abstract) information.

For example, consider a design process that transforms a set of specifications for a house into a sketch and then into a set of blueprints. This process is represented by three nodes (one each for the specifications, sketch and blueprints) and two arcs (one for the operator used in transforming specifications into sketches, another for the operator that maps sketches into blueprints).

The architecture used by FORS reflects the TAO graph view of design. Models and operators are treated as distinct objects and distributed over a network of computers. The computational paths they make possible are displayed via an icon-based interface. New models and operators can be easily added.

Some details on the features of FORS:

Models: FORS facilitates the creation of classes of models. A description of a model class includes a list of representations for the model, facilities to translate between representations, editors, browsers, and error detection and correction mechanisms. During the design process, model classes are instantiated to create models of specific parts.

Operators: FORS also allows for an expandable library of operators. A description of an operator includes a specification of the executable program that constitutes the operator (which may reside on any machine in the network), and lists of input and output model classes. Operators can, and have been, written in a variety of languages.

Control: FORS allows operators to be autonomous or non-autonomous. In addition, operators can be constructed in a hierarchical manner.

Distributed computing environment: FORS is built upon *DPSK* [4], a kernel for distributed problem solving, which provides facilities for programs residing on different machines to execute and communicate with each other.

User interface: FORS also provides a multi-window graphical user interface. Each model and operator is represented by an icon. An icon has an associated pop-up menu which provides commands that are appropriate for the type of object represented by the icon. Novice and expert menus are available.

SYNTHESIS

Synthesis is the process of transforming a set of specifications into (more or less) detailed component and assembly descriptions (i.e. transforming function to structure). Real-world design problems are very messy or ill-structured: when confronted with a neWj>roblem (or variant of an old problem) a designer must 'play* with the problem and various approaches, in ordfer to determine the trade-offs and the best way to attack the problem. Traditionally this process it done primarily in his/her head, often using the proverbial back of an envelope and occasionally a slide rule or calculator to perform calculations.

We feel that one of the best ways to provide automated support for designers is to provide a means for representing *objects* (the artifacts being designed, or portions thereof) and *constraints* upon the objects. Constraints are nary relations representing performance specifications, physical laws (including the geometry of rigid bodies), design decisions and designer preferences. Most CAO systems (including solid modelers) are capable of representing objects but not constraints. (Parametric design systems have some support for representing constraints, but only in a very

limited manner) Current CAD systems are good at handling the output of the design process (i.e. blueprints), but are inadequate for actually performing design synthesis. As a result, current CAD systems are not used for design synthesis.

As part of work in constraint-based design systems we have created a constraint language called *DOC* (Design Objects and Constraints). We have used DOC, as part of the ASE project, to create a system for window regulator synthesis, called *WoRM* (Window Regulator Mechanism design). DOC and WoRM are described in detail in [5].

WoRM takes a specification model and produces a model describing the kinematics and stick figure geometry of the assembly. Currently, detailed part geometry is represented parametrically in Pro/Engineer, a commercial parametric solid modelling system.

Our research in design synthesis focuses upon determining what functionality a constraint-based design system should have, and determining ways to implement it. The two main issues in constraint systems are language (how to specify, edit and browse constraint-object networks) and solution (how to satisfy systems of constraints).

TOLERANCE CRITIC

The purpose of the tolerancing critic is to determine the worst-case behavior of an assembly of parts due to manufacturing errors. This critic is important for two reasons, 1) it is difficult for a designer to foresee the effects that manufacturing errors will have on the behavior of the device, and 2) the tolerances that a designer allocates have great impact on the cost, quality and reliability of the device.

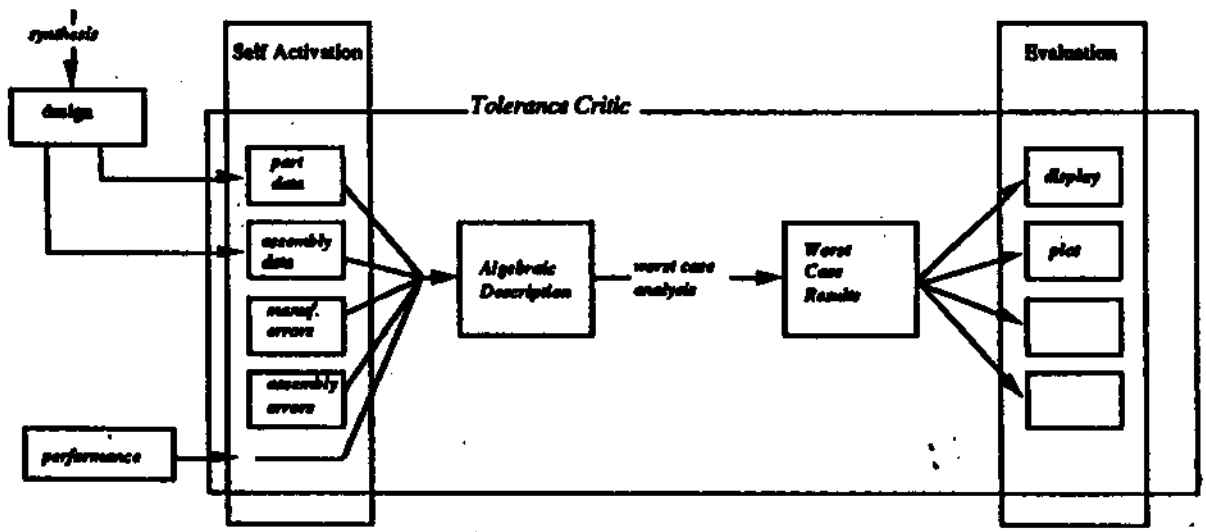

Figure 2. Simplified internal TAO graph of the tolerance critic

The critic consists of 5 input models, a set of analysis operators, a set of internal models, and a set of translation operators (Figure 2). The first model describes each part, the second one how the parts are connected. The third and fourth models describe the manufacturing and assembly errors, and the fifth model identifies the behaviors of interest

The input models are created as the output of the synthesis system discussed above. Alternately, the models can be created by other means, including manually, through an interactive user interface. The geometric description of the device is translated from the input models into an equivalent algebraic description, which is used to formulate a nonlinear programming problem that will find the worst-case behavior. Evaluation operators interpret the results and present them to the designer in the form of graphs and plots. The critic is activated when any of the input models changes.

With the current system, we can perform a two-dimensional worst-case analysis of any mechanical object generated by the synthesis module. As part of our future work, we would like to expand our library of analysis operators to include statistical tolerance analysis and device sensitivity analysis. We would also like to expand the library of self-activating algorithms to include one that takes into account part sensitivity information.

MECHANICAL STRENGTH CRITIC

The *MSC* (Mechanical Strength Critic) monitors the structural integrity aspects of the part being considered. The tasks assigned to this critic include: a) detailed stress analysis of the part under the worst-case loading experienced by the part; b) synthesis of the results obtained from the stress analysis to locate overstressed regions or other design violations. The designer is cautioned about the existence of such regions; and c) development of recommendations to provide the designer with a set of alternate choices to remedy the design violations. The MSC is consistent with the attributes defined earlier in this paper for an automatic critic. These attributes, as they apply to this critic, are described below.

Knowledge: Mechanical parts come in widely varying shapes and sizes and may be subjected to widely varying loading environments. It is then essential that an automatic critic is equiped with a general analysis tool that is capable of analyzing such widely varying entities. *Finite Element Analysis* (FEA) is such a tool and forms the basis of knowledge of the MSC. For all but the simplest mechanical configurations, the designer is not able to perform a detailed stress analysis of the part and the incorporation of FEA does represent knowledge which goes beyond that of the designer.

Quickness: The MSC is being developed as a stand-alone capability and will be housed in a separate computer. It will then be able to provide the results to the designer quickly enough to allow him to make other decisions keeping this information in mind.

Unobtrusiveness: The input to the MSC consists of: a) detailed description of the configuration of the component. This is provided in the form of Pro/Engineer solid models; b) material properties of the constituent materials of the component and the loading environment; and c) the design criterion for allowable performance values for the component. The mesh generation required for the FEA is a critical procedure since besides dictating the accuracy of results obtained it is the main time-consuming procedure for the designer. An expert system, that uses the above information as input, is developed to automatically generate an 'intelligent' mesh. A set of rules have been developed which based on the geometry, the loading, and the geometry-loading interaction of the component, identify areas of low- and high-density for the mesh provided for the component. This information is then passed along to the Pro/Engineer mesh generation capability which generates the corresponding FEA mesh and which also generates an input data file for the commercial FEA code MSC/NASTRAN. At no stage of this operation is the help or the attention of the designer sought by the critic.

Ease of Understanding Results: The critic performs a synthesis of the FEA results obtained and then locates possible regions of design violations. These regions will be graphically displayed for

easy comprehension by the designer. In addition, a detailed report describing the design violations alongwith a list of recommendations to remedy these will be provided

Generality: As discussed in the section on knowledge for the MSM critic, the FEA is a tool that allows for the analysis of a general class of mechanical components. The rules developed for the FEA mesh generation are also general and do not depend on specific dimensions of a part or on a specific loading scenario.

Expandability: The set of rules written for the generation of the 'intelligent* FEA meshes can easily be expanded by simply adding on the additional rules. The commercial FEA code, though not available to the users, is modular in nature and can be expanded to include additional analysis capabilities, if desired. It is also easily possible to interface the critic with another FEA code that does have the analysis capability desired for a new class of components.

The MSC is capable of providing the designer, in a matter of hours, with information which would otherwise take weeks to generate, thus providing her with valuable structural integrity information at a very desirable stage in the design process.

KINEMATICS CRITIC

The role of the kinematics critic is to verify that assemblies with moving parts function correctly over their range of operation. To do this, the kinematics critic simulates the motion of the assembly and reports any interferences between parts.

The input to the kinematics critic consists of:

- A description of the mechanism including allowable ranges of motion. This is represented and transferred as a DOC model.
- Solid model descriptions of each of the components in the mechanism.
- Solid models of other objects that might potentially interfere with the operation of the mechanism.

Once invoked, the kinematics critic performs its simulation by exercising the DOC model over its range of applicability. At every step in the simulation the mechanism components are transformed to their new location in space. The solid modeler is used to identify any interferences between components. Note that the use of DOC for this application serves two roles: 1) as a language for representing mechanisms, and 2) as a kinematics simulator. If an interference is detected, the critic will notify the designer, and can demonstrate exactly how the device fails, reporting the amount of interference.

Comparing the kinematics critic to the requirements for a critic presented earlier, we see that the kinematics critic:

1. is capable of performing detailed geometric calculations to determine the validity of the design;
2. performs this analysis much faster than the designer could;
3. does not require any additional information from the designer, and will only interrupt him with useful results;
4. can demonstrate exactly how the design fails;
5. is general enough to work with arbitrary mechanisms and parts.

RESEARCH ISSUES

The following are what we see as the main research issues that must be addressed before the benefits of automatic critics can be fully realized:

- Translation between representations. In order to add new critics and models to a design system it is necessary to build translators between the different models and representations. How can this process be automated?

- Modeling abstractions. Transfering data from one model to another requires that information be added and/or removed. Removing information can be viewed as an abstraction process. Making such abstractions often requires a great deal of knowledge. For critics to be useful even in limited domains, rather than for specific problems, this abstraction process must be automated.

- Automatic invocation. To be quick, critics must be invoked at appropriate times, e.g. when the design has been changed significantly. How can a critic judge what is and isn't a significant change?

- Useful feedback. The critic must present its conclusions to the user in a manner he can understand and use.

- Constraints. Our work on constraint systems demonstrates the utility of constraint systems for design synthesis. However, we feel that this work will also be applicable to the model/operator level of design. In particular, issues of constraint representation and satisfaction are important.

- Conflict resolution. The role of critics is to identify conflicts. Once identified they must be dealt with. This is an area we are just beginning to seriously investigate.

Our approach to these issues is to begin by working in several limited domains with specific parts. By actually implementing these limited-domain critics we are able to better understand the issues and complexities involved with creating more general critics. As we progress, we try to draw conclusions that are applicable to larger domains. These are then prototyped, and the process iterates.

CONCLUSIONS

Automatic critics, as we have defined them here, are a useful concept. Through the package of programs called ASE we have demonstrated that this concept can be translated into practice. We expect that ASE will prove, in field tests that are just beginning, that it can considerably shorten design cycles for certain automobile parts. We believe that the critic approach to design systems, as demonstrated in ASE, is generalizable to other domains and can also be scaled up to much larger problems, but this still remains to be proven. We believe that our bottom-up approach to these difficult problems will prove fruitful, but much work remains to be done.

REFERENCES

[1J K. B. Clark, T. Fujimoto, *Overlapping Product Solving in Product Development,* working paper, Harvard Business School, March 1987.

[2] M. Sapossnek, S. N. Talukdar, A. Elfes, S. Sedas, M. Eisenberger, L. Hou, *Design Critics in the Computer-Aided Simultaneous En* ss!as?SK ect), 1989 ASME Winter Annual Meeting, special symposium San Francisco, December 1989

[3] N- Papanikolopoulos, *FORS: Flexible Organizations,* Masters Project Report, Department of Electrical and Computer Engineering, Carnegie Mellon University, November 1988

[4] E. Cardozo, *A Kernel for Distributed Problem Solving,* PhD Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, January 1987

[5] M. Sapossnek, *Research On Constraint-based Design Systems,* in Proceedings of the Fourth International Conference on Applications of AI to Engineering, Cambridge, England, July 1989