Desa, S., Nagurka, M.L., and Ghosal, A., "Product Re-Design for Performance, Manufacture, and Assembly: A Rational Methodology towards Total System Design," Proceedings of the 1987 International Conference on Engineering Design, Vol.1, *ed.* Eder, W.E., Boston, MA, August 17-20, 1987, pp. 463-472.

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN

ICED 87

Boston, MA 17-20 August 1987

PRODUCT REDESIGN FOR PERFORMANCE, MANUFACTURE, AND ASSEMBLY: A RATIONAL METHODOLOGY TOWARDS TOTAL SYSTEM DESIGN

S. Desa, M.L. Nagurka, and A. Ghosal

ABSTRACT

This paper describes the objectives, the methodology, and the progress made in the project entitled "Product Redesign for Performance, Manufacture and Assembly". Redesign is the process of modifying or changing an existing design with the objective of improving one or more of its aspects. The aspects of an engineering design which concern us are performance, manufacture and assembly.

The objectives of the research are to develop methods for the total system redesign of an engineering product, and to create a computer environment to implement these methods.

In this paper we describe our overall approach towards achieving these objectives. One of the first steps towards achieving the aforementioned objectives was the development of a simple expert system which will be used to implement redesign procedures. The structure and current capabilities of this expert system are described.

Zusammenfassung

Diese Arbeit beschreibt sowohl Ziel und Methode des Projektes *"Product Redesign for Performance, Manufacture and Assembly"* als auch den Fortschritt, der darin erzielt wurde. Unter Redesign versteht man den Vorgang des Modifizierens oder Änderns eines bestehenden Designs mit dem Ziel des Verbessems von einem oder mehreren Designaspekten. Die für uns relevanten Aspekte eines technischen Designs sind Erfüllung der gestellten Anforderungen und Einfachheit der Herstellung und Montage des Endproduktes.

Das Ziel unserer Forschung ist, Methoden für das vollstandige Redesign eines technischen Produktes zu entwickeln und ein rechnerunterstütztes Umfeld zu schaffen, in dem diese Methoden zum Einsatz gebracht werden.

In dieser Arbeit beschreiben wir unsere Vorgangsweise bei der Erlangung dieser Ziele. Einer der ersten Schritte auf dem Weg der Verwirklichung des oben genannten Vorhabens war die Entwicklung eines einfachen Expertensystems, das beim Redesignverfahren eingesetzt werden wird. Die Struktur und die gegenwärtigen Fähigkeiten dieses Expertensystems werden beschrieben.

1 INTRODUCTION

This paper describes the objectives, the methodology and the progress made in the PRODUCT REDESIGN (FOR PERFORMANCE, MANUFACTURE AND ASSEMBLY) PROJECT carried out in the Mechanical Engineering Department and the Engineering Design Research Center at Carnegie Mellon University.

The paper is organized as follows. In Section 2 we describe the redesign problem and our motivations for studying redesign. The Research Objectives and Approach are then described in Section 3. In Section 4 we describe the current status of the research. Finally in Section 5 we draw some conclusions based on our work so far and also indicate future directions.

2 REDESIGN

In this section, after reviewing the product design process, we briefly describe what we mean by redesign, where it fits in the design cycle and our reasons for studying redesign.

The design of an engineering system is a highly iterative process which typically involves the following stages [1].

1. Problem Specification
2. Conceptual phase (Generation of design alternatives)
3. Evaluation (Selection of one or more of the alternatives)
4. Detailed Design (of the chosen alternative)
5. Testing

There are also several types of design; two which concern us are Original design [1], and Redesign.

Original design, typically arises when we have to come up with a design (solution) for an as yet unsolved problem; for example Alexander Graham Bell's invention of the telephone was an original design. In original design, the first three stages of the design process (viz. problem specification, conceptual phase and evaluation) are the crucial and most difficult stages.

For our purpose, Redesign is the process of modifying or changing an existing design with the objective of improving one or more of its aspects. The aspects of an engineering design which concern us are performance, manufacture and assembly. Typical examples are redesign of an electro-mechanical print-head assembly for better performance (i.e. higher speed and precision) and redesign of an existing printer for automatic assembly.

In redesign, while a large number of the functional needs of the overall engineering system are generally well defined and not variable, the functional needs (and hence form) of the subsystems and components usually change. Therefore redesign generally requires original design of subsystems and components.

There are at least three distinct situations in which redesign is important:

1. In the original design of an engineering system where we have come up with a first version of the design which needs to be improved (Here redesign is part of the original design cycle).

2. An existing design has to be changed to accommodate a new manufacturing or assembly environment. For example, the IBM Proprinter is a result of printer redesign for automatic assembly.

3. An existing design has to be redesigned for improved performance, lower cost, higher reliability etc. A familiar example of this kind of redesign, is any family of personal computers.

A schematic of the Redesign process is shown in Figure 1. The inputs to the Redesign Module are the initial design and a set of design specifications. At each cycle of the redesign iteration a check is performed in the Redesign Module to see if the design satisfies the design specifications. If the specifications are satisfied the iteration stops to yield the improved design, else the iteration continues. Note that if we cannot meet our design specifications after a reasonable number of redesign iterations, then we probably need to define a new set of design specifications.

Our motivations for studying the redesign process are the following.

1. Redesign is a much more tractable problem than original design. Studying redesign is therefore a natural first step to a clearer understanding of the original design of engineering systems.

2. At present, an adequate theory of redesign does not exist.

3. Even though redesign is a relatively simpler problem than original design, no systematic procedures exist for total system redesign. (We elaborate on this below.)

4. The development of systematic design procedures should aid in substantially reducing the large amounts of time and money currently spent on redesign in industry.

Some current approaches to redesign can be found in References [2] and [3].

3 OBJECTIVES AND APPROACH OF THE RESEARCH

In this section we first describe the general objectives of our research. We then describe our overall approach towards achieving these objectives.

While tools and procedures exist for (separately) dealing with the various analytical aspects of the design process, for example, tools for dynamic analysis, stress analysis, optimization and computer-aided graphics, we do not have unified tools or procedures to deal with the totality of the system redesign process. Furthermore good analytical tools for design for manufacture and assembly do not exist. While redesign for manufacture and assembly is addressed in [3], the resulting procedures are too complex.

One of our objectives, then is to develop tools and procedures for total system redesign of an engineering product. These tools would typically enable us to redesign the product for (improved) performance, manufacture and assembly, etc. Currently, most attempts at redesign that we are aware of address specific aspects of the redesign process; for example [2] is concerned primarily with performance whereas [3] is concerned primarily with assembly.

Both design and redesign are iterative and complex in nature. Furthermore they both have algorithmic and heuristic aspects. To deal with the iterative and complex nature of the redesign process we plan to make extensive use of the computer and create a suitable computer environment. We will create so-called EXPERT SYSTEMS to handle the heuristic and algorithmic aspects of the redesign process. We would like to emphasize that we see the computer as a tool to efficiently implement the redesign process. The crucial part of our research will be to develop the design rules and procedures which will "drive" the redesign process.

To achieve our objectives we plan to proceed as follows.

1. We will take specific engineering systems of increasing degrees of complexity and develop methods to redesign them for performance, manufacture and assembly (as appropriate). We plan to study the following systems.

   a. a hammer

   b. an electromechanical drive assembly of a print-head

   c. a printer

   d. robotic manipulators

2. As explained earlier, we plan to create a computer environment to implement the redesign methods developed for the above systems. A schematic view of the required computer environment being implemented is shown in Figure 2. As shown in Figure 2, the computer environment has domain-specific and domain-independent parts. We plan to expand the features and capabilities of the computer environment within the context of studying the (increasingly complex) systems listed above. Our first attempts at developing the computer environment are described in the next section.

3. >From the methods developed in 1 and implemented in 2 we will extract important redesign information, for example,

   a. measures of "good" design,

   b. identification of the critical design variables,

   c. useful design rules.

4. We will generalize the experience and conclusions gained from the above specific systems to the Redesign of mechanical engineering systems in general.

5. An important part of the above process will be the development of a reliable set of design rules for manufacture and assembly.

4 CURRENT STATUS OF THE RESEARCH

We have been working on the Redesign Project for four months. In this time we have taken the first of our examples, viz., the hammer, and created a suitable computer environment. In this section, we describe the components of the computer environment required for redesign (please see Figure 2).

The main parts of the computer environment are the User Interface and the Redesign Module.

4.1 The User Interface

The user interface consists of functions and procedures which aid the user in the redesign process. The aim of the user interface is to make the computer environment user-friendly and shield the user from the "details". Among other things, it lets the user input design specifications, load knowledge bases, modules for analysis and display changes in design in the redesign process. Implementation of this is in a preliminary stage.

4.2 Redesign Module

The Redesign Module consists of two main parts - (a) the domain independent inference engine "POGIE2" and (b) the domain specific parts such as the Knowledge Base, LISP Functions, etc. In this subsection we give a brief description of POGIE2 and other domain specific parts.

4.2.1 General Features of POGIE2

POGIE2 (POrtable Generic Inference Engine - version 2) is a standard inference engine that performs backward and forward inference as well as lookups on user supplied facts and IF-THEN rules. It is written in MACLISP and is based on POGIE [4]. Some of the general features are as follows:

1. Introduction

   POGIE2 performs inference on symbolic predicate formula represented by LISP lists e.g. (XY IS A HAMMER). It accepts variables which are denoted by a $ sign e.g. ($HEAD HAS SUFFICIENT MASS).

2. Inference Procedures

   The main inference procedure in POGIE2 is depth-first backward chaining. The function PROVE attempts to satisfy a goal by (a) looking in the list of facts in the Knowledge Base (b) backward chaining (c) asking the user. POGIE2 also lets the user do forward chaining. Many other variations of PROVE such as prove without asking the user and proving list of goals are also available.

3. Procedural Attachments

   To allow any LISP functions to be called during the inference process, the system defines two special forms. If the first character of any predicate is a * or +, then the predicate is satisfied by calling the associated function. For example in the case of a hammer, we may have a rule such as:

   ```
   (RULE 03
          (IF (*SAT-HEAD-REQ T))
          (THEN($HEAD HAS SUFFICIENT MASS)))
   ```

   In the above rule to prove ($HEAD HAS SUFFICIENT MASS) the LISP function (SAT-HEAD-REQ), which checks whether mass of head requirements are satisfied or not, is called. If the output of this function is T (true), the head requirements are satisfied and the THEN part is true.

4. Utility Functions and the History Mechanisms

POGIE2 provides many functions which help user keep track of the inference process, add and delete rules and facts, and offer explanation of "how" a goal was achieved and "why" a rule was used.

POGIE2 also allows options for 1) remembering intermediate results, 2) remembering answers to questions (asked to the user), 3) recording rules used to achieve a goal and 4) recording propositions answered negatively. These four options are used to maintain "history" i.e. we can find which rules were used, which rules failed, which propositions are true and which are not. With this history, we can take steps in the next iterative cycle to correct the deficiencies according to some redesign rules.

4.2.2 General Features of the KNOWLEDGE BASE

The Knowledge Base is domain specific - i.e. it depends on the object which is being redesigned. The Knowledge Base consists of two lists - <facts> and <rules>. <facts> is a list of propositions known to be true. <rules> is a list of IF - THEN rules. Each of the rules take the following form:

```
(RULE <RULE-IDENTIFIER> (IF (P1) (P2)...(PN))
                        (THEN (Q1)...(QM))
```

where P's and Q's are propositions which may contain variables and call other LISP functions. A file containing the Knowledge Base can be easily loaded in the LISP environment. A subset of rules for the hammer is shown below. (The rules are explained later)

```
(SETQ RULES '(

(RULE O1
          (IF ($OBJECT CONTAINS $HEAD AND $HANDLE)
           ($HEAD HAS SUFFICIENT MASS)
           ($HEAD ATTACHED TO $HANDLE)
           ($HANDLE CAN BE GRIPPED)
           ($OBJECT CAN IN IMPART-FORCE))
      (THEN ($OBJECT IS A HAMMER))

(RULE O2
          (IF (*SAT-HEAD-REQ))
          (THEN ($HEAD HAS SUFFICIENT MASS))
          .
          .
          .
          .

(RULE C1
          (IF ($OBJECT IS HAMMER)
              ($OBJECT HAS HEAD OF MASS BETWEEN 1.0 AND 1.25 LBS)
              ($OBJECT HAS STRIKING SURFACE DIAMETER AROUND 1")
         (THEN ($OBJECT IS A CARPENTER-HAMMER))
              .
              .
              .
              .

(RULE G1  (IF (*CONNECTED $X $Y T))
         (THEN ($X ATTACHED TO $Y))
              .
              .
```

```
))
A typical list of facts for the hammer may be as follows:
(SETQ FACTS '( (H1 HAS HEAD)


                (XY CONTAINS $HEAD AND $HANDLE)
))
```

The user can also add and delete facts and rules by using POGIE2 supplied functions.

In the examples of the hammer, the Knowledge Base at present consists of 12 rules a) defining a hammer , for example RULE 01, RULE 02 above, b) classifying hammers according to mass of head, diameter of the striking surface etc., for example RULE C1 above, and c) general rules defining connectedness etc., for example RULE G1 above. The rules RULE C1 etc. were obtained by observing different types of hammers. We do not have rules for redesign as yet, and hence, for the hammer, we cannot do redesign. We can only identify (infer whether an object satisfies the definition of a hammer) and classify (determine whether the object is a carpenter-hammer, tack-hammer etc.). It is expected that once "good" performance measures and redesign rules are obtained, we could, with the aid of the history mechanism, redesign hammers.

4.2.3 LISP FUNCTIONS

In the redesign process, it is expected that we will use functions to compute performance and other measures. POGIE2 allows, through procedural attachments, any LISP functions to be called during the inference process. A file containing these required functions can be easily loaded in the LISP environment. For example, a simple LISP function, for evaluating whether a hammer satisfies a mass-requirement (for the head) or not, is given below:

```
(DEFINE  SAT-HEAD-REQ ()
  (PRINT " PLEASE ENTER MASS OF HEAD IN LBS")
  (TERPRI) (SETQ MASS (READ)
     (COND ((AND (> MASS 0.25)
                 (< MASS 6.5)) T)
           (T 'FAIL)))
```

4.2.4 Analysis Packages

In addition to the LISP functions called in the inference process, we would like to have the capability of calling procedures not written in LISP. Numerous packages for optimization, dynamics, control, stress analysis etc. are available in literature and commercially. Since it is very time consuming and difficult to write such packages in LISP, it is worthwhile developing methods for communication between POGIE2 and such packages. This part of the Redesign Module has yet to be completed.

4.2.5 Solid Modelling/Geometric Reasoning

The purpose of this block is the Redesign Module, in addition to display, is geometric reasoning. The intention is to reason about connectedness, interference, etc., from the solid model and communicate with POGIE2.

This part of the Redesign Module is currently being constructed.

4.2.6 Results and Accomplishments

At present, we have accomplished the following:

1. The inference engine POGIE2 has been fully tested and debugged. Through the mechanism of procedural attachment, we have shown how any LISP functions can be called during the inference process. We have shown that a "history" of the inference process can be maintained.

2. We have written a simple Knowledge Base for identifying and classifying hammers.

5 CONCLUSIONS AND FUTURE DIRECTIONS

The simple example of the hammer has served the purpose of initiating the development of the computer environment and expert system described in the previous section. The inference engine of this expert system has been fully tested and debugged. We have developed the history mechanism for use in redesign. While the resulting expert system does identify and classify hammers, it does not "redesign" them.

The example of the hammer has served the additional purpose of highlighting the crucial need for a) good performance measures, b) good understanding of the effect of design modifications on these performance measures and c) procedures which utilize a) and b) to systematically redesign a product.

We do not feel that it is worthwhile to develop good performance measures for a hammer. We therefore plan to go directly to the problem of redesign of the electro-mechanical drive assembly for (say) a print-head for improved performance. We feel a) that we have a good understanding of the design and performance issues involved in electro-mechanical drives and b) that the problem is of practical interest. The outcome of this particular investigation should provide us with reliable redesign procedures for electro-mechanical drive systems as well as a clear understanding of the critical design variables and performance trade-offs in the design of such systems. This example will also provide the motivation for increasing the capabilities of our computer environment: the solution of the electro-mechanical drive problem will require us to link the existing expert systems to various analytical software tools (dynamics, control system design, optimization) and graphical software tools (for display).

ACKNOWLEDGEMENTS

This research is being supported by the Engineering Design Research Center at Carnegie Mellon University. The figures were drawn by Yong-Yil Kim.

6 REFERENCES

1. Pahl, G. and Beitz, W., *Engineering Design*, Springer-Verlag, 1984 (English Translation edited by Ken Wallace).

2. Dixon, J.R., Howe, A., and Cohen, P.R., Dominic 1: Progress Towards Domain Independence in Design by Iterative Redesign, *Proceedings of the ASME Computers in Engineering Conference*, Chicago, July 1986.

3. Boothroyd, G. and Dewhurst, P., *Design for Assembly Handbook*, University of Massachusetts Press, 1983.

4. Ishii, K. (1987), POGIE Manual, *SIMA Technical Report*, 2-15-87, Stanford University, CA.

Department of Mechanical Engineering,
Engineering Design Research Center,
and the Robotics Institute

Carnegie Mellon University
Pittsburgh, PA 15213

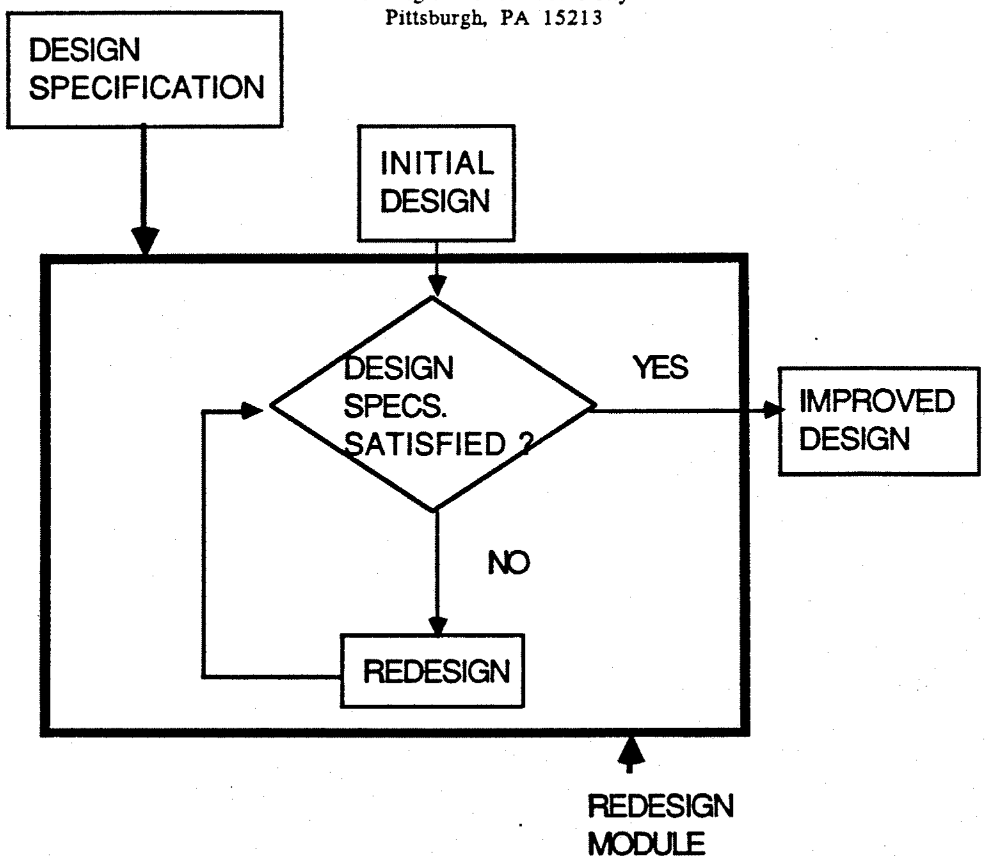

Figure 1 : Flow Chart of the Redesign Process

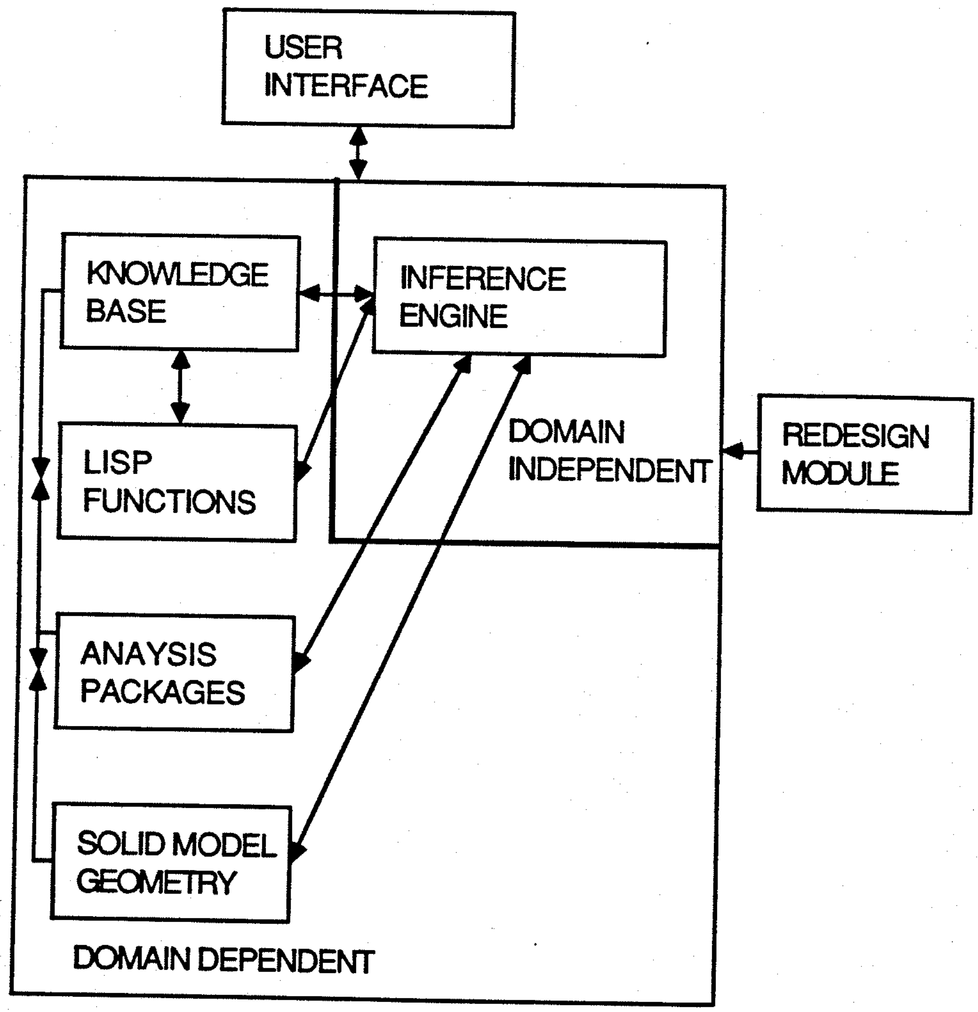

Figure 2 : Components of the Redesign Module