List of Figures

List of Tables

A Computer-Aided Product Redesign System for Robotic Assembly

Wynne Hsu, Lecturer

Department of Information Systems and Computer Science

National University of Singapore

Lower Kent Ridge Road

Singapore 119260

C. S. G. Lee, Professor

Department of Electrical Engineering

Purdue University

West Lafayette, IN 47907

U. S. A.

Andrew Lim, Engineer

Information Technology Institute

National Computer Board

Singapore 117685

[0]This work was supported in part by the National University of Singapore Research Project Grant RP940644

ABSTRACT

It is an established fact that majority of a product's cost is determined by its design. Hence, effort should be directed to achieve a lower cost design without sacrificing the original functionality. A computer-aided product redesign system is proposed to provide help in generating assembly-oriented redesign by taking the robotic assembly constraints into consideration. The system operates in two phases. In the first phase, an objective evaluation of the input design is performed to determine whether there is a need for redesign. During the second phase, the focus is to aid the designer in searching for feasible design alternatives. Three quantitative measures have been proposed to evaluate the conformity of an input design to a list of DFA design guidelines. If the input design received a bad evaluation results on any of the three measures, a search for feasible redesign suggestions is initiated to derive suitable redesign suggestions.

List of symbols:

$\partial \mathbf{FR}_i$

$\partial \mathbf{DP}_j$

A_{ij}

μ_A

μ_R

$\mu_{\widetilde{\mathbf{contribute}}}$

$\mu_{\widetilde{\mathbf{strength}}}$

G_D

ϕ

θ

$\in$

$\cup$

$\sum$

$X \times Y$

$\widetilde{\mathbf{contribute}}^{\mathbf{T}}$

$|V|$

$O(|V| + |E|)$

$\neg$

1 Introduction

The aim of manufacturing industry is to create products which sell profitably. To achieve this aim, a product must constantly evolve to meet the changing needs. For example, with the introduction of robotic assembly, many products must be redesigned so that they are suitable for robotic assembly. Traditionally, redesign activity has always been left entirely up to the human designer's skill and knowledge. With the increasing complexity in today's environment, new tools and techniques are needed to support the redesign activity. In this paper, a systematic approach to redesign is proposed. Our viewpoints behind the proposed approach are the following:

Improvement Is Always Possible

With the changing market and the rapid advances in technology, a design must constantly evolve in order to stay competitive.

Quantify Whenever Possible

Vague and subjective judgements are not a good basis for redesign. For the redesign process to be effective and efficient, an objective and quantifiable basis is vital.

Integrate Design and Assembly

A small difference in the design can have a major effect downstream. By integrating design and assembly, the effect of design on downstream activities can be considered early.

Boothyrod, recognizing the significant impact a design activity has on the downstream activities(assembly), proposed a quantitative design-for-assembly approach in the 1980s [1]. His approach has been adopted in industry to this day and provides a good source of feedback to the designer concerning the assemblability of his/her design. At about the same time, another group of proponents proposed a more qualitative approach to help the designer. In this approach, high-level design principles are established which serve as guidelines to the designer on how to produce a more assembly-/manufacturing-oriented design [2]. Recently, the concurrent engineering advocates [3] also favor using a multi-disciplinary team during the design phase so as to bring later life-cycle issues into consideration in the design phase. These approaches all focus on the derivation of an 'optimal' initial design solution. However, in real-life situation, no design is perfect in the first attempt. Redesign is, and always will be, an essential part of a product's life-cycle. In this paper, we will like to discuss a systematic redesign approach which integrates the assembly phase and the design phase with the goal of providing dynamic, product-specific redesign suggestions to the designer. A brief comparison of the various approaches is shown in Table 1.

The fundamental issues addressed in our redesign approach are twofold: 1) to provide objective evaluation functions that will measure the robotic assemblability of a design, and 2) to provide suggestions on how the product should be redesigned so that it is better for assembly. The goal of this research is to provide redesign support to the human designers in the form of two important tools: an evaluation tool and a redesign suggestion tool. The evaluation tool allows the designer to evaluate his/her design based on some objective, mathematically-based evaluation criteria. By providing such objective evaluation criteria, an unbiased comparison between different design alternatives is made possible. The redesign suggestion tool aids the designer by performing analysis on large amount of data (typically generated by the computer-aided design tools and computer-aided assembly planners) and summarizing the results of analysis to the designer. In addition, a collective database of previous design knowledge is kept to aid in the search for feasible redesign alternatives.

Outline of this paper follows. Section 2 gives some background information and related work. The formalism of the proposed evaluation functions are given in Section 3. In Section 4, the process of generating suitable redesign suggestions is described. An illustration is given in Section 5 using the tilt-mechanism example. Finally, Section 6 summarizes the conclusions.

2 Related Work

Much work has been done in the area of design, ranging from the axiomatic approach proposed by Suh [4] to the application-oriented research such as case study of successful designs [5]. Many researchers have also proposed different design approaches to give better understanding of the design process itself. Design synthetic reasoning was proposed by Kannapan and Marshek [6] to derive the structure of a product from functional specifications. Their focus is in deriving a conceptual design from specifications. On the other hand, Jack Mostow [7] focused on the issues of design by derivational analogy. Various issues involved in design such as adapting the old design to fit a new design and partial reuse of an old design were discussed. Functional design [8] emphasizes the role of functional specification to aid in the selection of engineering features. It is a process of function-to-feature matching which forms the basis of a part taxonomy. In Kota and Ward's work [9], an interesting view was presented on the relationship among functions, structures and constraints in a conceptual design. The study by Watton and Rinderle [10] provided an interesting approach to enhance a given design. Their approach is to find alternative parameters to give a more direct correspondence to functional behavior.

Beside looking at design/redesign from a functional perspective, researchers also began to look at redesign from the assembly or manufacturing perspective. A wealth of research has been developed on design for assembly [1, 2, 11, 12, 13], design for manufacturing [14, 15] and concurrent engineering [3, 16, 17, 18]. Recently, design for disassembly approach [19] has also been proposed which aims to derive designs that are easily decomposable. This helps to increase the feasibility of reusing old components and enable easy recycling. In addition, with the introduction of robotic assembly, substantial changes are required in the assembly methods, assembly tools, assembly system designs and product design. Rathmill [20], Rampersad [21] investigated various issues specific to robotic assembly.

3 Design Considerations for Robotic Assembly

Robotic assembly, also termed automatic assembly, demands special design considerations. One consequence is that many existing product designs end up requiring some form of redesign to make them suitable for robotic assembly. For example, the insertion of an slightly asymmetrical object into a hole poses little problem to a human operator, however it can cause a great deal of problem in a robotic assembly environment. To derive a redesign solution such that the product is good for robotic assembly, it is necessary to first understand why the exiting design is not good for robotic assembly, and having understood the reasons, suggests modifications to overcome the weaknesses.

Due to the lack of support for redesign activity, a designer normally only has accessed to some high-level design guidelines to guide him in the redesign process. Examples of design guidelines for robotic assembly are summarized below:

- Minimize Assembly Surfaces.

  Multiple assembly surfaces mean wasted assembly motions and time. From an automation standpoint, multiple assembly surfaces mean expensive fixture and equipment costs.

- Design for Z-Axis Assembly.

  Avoid multi-motion insertion by designing for "top-down" assembly. Z-axis assembly permits simple robots and insertion tooling with gravity serving to assist in the assembly process.

- Improve Assembly Access.

  Provide a "clear view" for assembly operations. Avoid parts or assembly sequences that require tactile sensing for installation.

- Maximize Part Compliance.

  Part mating is a major challenge to automatic assembly. Compliance, the ability of one part to move so that it can mate with another, must be designed into both the product and the production process.

- Maximize Part Symmetry.

  The more symmetrical a part, the easier it is to handle and orient.

- Avoid Separate Fasteners Wherever Possible.

  Fasteners are a major barrier to efficient assembly. The best design approach is to incorporate the fastening function into a major component.

- Provide Parts with Integral "Self-Locking" Features.

  Provide projections, indentations or other surface features that maintain the orientation and position of the parts already in place.

- Drive Toward Modular Design.

  Modular design simplifies final assembly.

These design guidelines, however, lack the ability to propose remedies to awkward and difficult assembly operations. Our goal is to implement a redesign aid that is capable of giving specific redesign solutions. We propose to do this in two phases. In the first phase, the input design is evaluated to determine whether there is a need for redesign, and if so, identify the parts/subassemblies that need to be redesigned. In the second phase, a systematic search is conducted to look for potential product-specific redesign suggestions.

3.1 Quantitative Measures for Design Evaluation

Based on the design guidelines for robotic assembly, the authors have defined three quantitative measures to evaluate a design's conformity to these design guidelines. The three measures are: (1) degree of modularity, (2) degree of standardization, and (3) degree of uniformity.

3.1.1 Degree of Modularity

One of the design guideline is the "drive toward modular design." A modular design is one that achieves an optimal degree of functional-physical independence. Suh [4], in his axiomatic approach to design, defines functional independence as follows: A design, in the form of a design matrix (**DM**), relates the functional requirement (FRs) vector of the functional space to the design parameters (DPs) vector of the physical space. We write:

$$\mathbf{FR}s = [\mathbf{DM}]\mathbf{DP}s$$

where

$$[\mathbf{DM}] = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ \vdots & & \vdots & \\ A_{n1} & A_{n2} & \cdots & A_{nm} \end{bmatrix}$$

$$A_{ij} = \partial \mathbf{FR}_i / \partial \mathbf{DP}_j$$

If the matrix [**DM**] is a diagonal matrix, we achieve functional independence. To extend this concept of functional independence to physical independence, we introduce mappings from the functional domain to the physical domain as shown in Figure 1. For example, an electric kettle may have the functional requirements (FRs) of "to boil", "to hold water", "to keep warm", "to pour", and "to input water". Based on user's specification, these functional requirements are partitioned into non-overlapping subsets: {"to boil", "to keep warm"}, {"to hold water"}, and {"to pour", "to input water"}. Each subset is denoted as one modular functional requirement (MF). The physical realization of a MF is achieved by a set of design forms (physical components and their mating relationships), denoted by DFs. For example, the MF of {"to boil water", "to keep warm"} can be realized by a heating coil with thermostat. A modular design is one that minimizes the interface relationships between pairs of MFs while maximizes the physical coherence within a MF. By this definition of a modular design, we note that a design which is modular, is also one that minimizes the number of assembly surfaces and the number of separate fasteners. This is because a design with a large number of assembly surfaces or large number of separate fasteners usually has in a larger number of interface relationships and this tends to reduce the modularity of a design. A formal definition of the modularity of a design uses fuzzy set theory. Fuzzy set theory is first introduced [22, 23] to describe the classes of objects that do not have precisely defined criterion of membership. In other words, a fuzzy set is a class of objects with a continuum of grades of membership. A fuzzy set A in U is characterized by a membership function $\mu_A(u)$ which associates with each element in U a real number in the unit interval [0, 1], with the value $\mu_A(u)$ at u representing the "grade of membership" of u in A. A fuzzy relation R is a fuzzy set in a Cartesian product $X \times Y$ in universes U and V, respectively [24]. The membership value of an element (x, y) in R, denoted by $\mu_R(x, y)$, indicates the strength of the link (or relation) between x and y.

To enable reasoning of the degree of modularity of a design, we must capture the MFs-DFs relationships and the DFs-DFs relationships. The MFs-DFs relationship is modeled by a fuzzy relation $\widetilde{\mathbf{contribute}}(\mathrm{MF}_i, \mathrm{DF}_j)$ where $\mu_{\widetilde{\mathbf{contribute}}}(\mathrm{MF}_i, \mathrm{DF}_j)$ denotes the degree to which "DF_j contributes to the physical realization of the function MF_i". The DFs-DFs relationship is modeled by a second fuzzy relation $\widetilde{\mathbf{strength}}(\mathrm{DF}_a, \mathrm{DF}_b)$ where $\mu_{\widetilde{\mathbf{strength}}}(\mathrm{DF}_a, \mathrm{DF}_b)$ represents the strength of the mating relationship between the design forms DF_a and DF_b. The strength of the mating relationship between two design forms (DFs) is defined by the difficulty in which the two DFs can be separated. For example, a 'stack-on' mating relationship has a lesser strength than a 'welded' mating relationship because it is more difficult to separate the two DFs that are welded together than to separate the two DFs that are simply stack on top of one another. In the case of $\widetilde{\mathbf{strength}}(\mathrm{DF}_a, \mathrm{DF}_a)$, we assign $\mu_{\widetilde{\mathbf{strength}}}(\mathrm{DF}_a, \mathrm{DF}_a) = 1.0$.

Suppose there are m MFs and n DFs. The $\widetilde{\mathbf{contribute}}$ relation and the $\widetilde{\mathbf{strength}}$ relation are represented in matrix notation as follows:

$$\widetilde{\mathbf{contribute}} = \begin{pmatrix} \mu_{\widetilde{contribute}}(\mathrm{MF}_1, \mathrm{DF}_1) & \cdots & \mu_{\widetilde{contribute}}(\mathrm{MF}_1, \mathrm{DF}_n) \\ \vdots & & \\ \mu_{\widetilde{contribute}}(\mathrm{MF}_m, \mathrm{DF}_1) & \cdots & \mu_{\widetilde{contribute}}(\mathrm{MF}_m, \mathrm{DF}_n) \end{pmatrix}$$

$$
\widetilde{\textbf{strength}} = \begin{pmatrix} 1.0 & \mu_{\widetilde{strength}}(\mathrm{DF}_1, \mathrm{DF}_2) & \cdots & \mu_{\widetilde{strength}}(\mathrm{DF}_1, \mathrm{DF}_n) \\ \mu_{\widetilde{strength}}(\mathrm{DF}_2, \mathrm{DF}_n) & & 1.0 & \mu_{\widetilde{strength}}(\mathrm{DF}_2, \mathrm{DF}_n) \\ \vdots & & & \vdots \\ \mu_{\widetilde{strength}}(\mathrm{DF}_n, \mathrm{DF}_1) & \mu_{\widetilde{strength}}(\mathrm{DF}_n, \mathrm{DF}_2) & \cdots & 1.0 \end{pmatrix}
$$

Graphical Representations

Alternatively, the $\widetilde{\textbf{contribute}}$ and $\widetilde{\textbf{strength}}$ relations of a design D can be represented as a graph, $G_D = (V_1 \cup V_2, E_1 \cup E_2)$. V_1 is the set of vertices denoting the MFs. V_2 is the set of vertices denoting the DFs. Let v_i represent MF_i and v_j represent DF_j. If $\mu_{\widetilde{\textbf{contribute}}}(\mathrm{MF}_i, \mathrm{DF}_j) > 0$, then $(v_i, v_j) \in E_1$ and $\mathrm{cost}(v_i, v_j) = \mu_{\widetilde{\textbf{contribute}}}(\mathrm{MF}_i, \mathrm{DF}_j)$. Similarly, let v_p be the vertex denoting DF_p, v_q be the vertex denoting DF_q. Suppose $\mu_{\widetilde{\textbf{strength}}}(\mathrm{DF}_p, \mathrm{DF}_q) > 0$, then $(v_p, v_q) \in E_2$ and $\mathrm{cost}(v_p, v_q) = \mu_{\widetilde{\textbf{strength}}}(\mathrm{DF}_p, \mathrm{DF}_q)$. We call G_D the functional structure graph of design D. Figure 2 shows the graphical representation of a given $\widetilde{\textbf{contribute}}$ and $\widetilde{\textbf{strength}}$ relations. For the remainder of this paper, the matrix representation and the graphical representation will be used interchangeably.

$$
\widetilde{\textbf{contribute}} = \begin{pmatrix} 1.0 & 0.2 & 0 \\ 0 & 0.9 & 1.0 \end{pmatrix}
$$

$$
\widetilde{\textbf{strength}} = \begin{pmatrix} 1.0 & 0.9 & 0 \\ 0.9 & 1.0 & 0.4 \\ 0 & 0.4 & 1.0 \end{pmatrix}
$$

With these definitions of the "contribute" and "strength" relationships, we can now compute the degree of modularity for a given design.

Definition 1. (The ϕ operator)

Let R_1 be a binary fuzzy relation in the Cartesian product $Y \times Y$ of universe V and R_2 in the Cartesian product $Y \times X$ of universes V and U, respectively. The operator, ϕ, defines a new relation R,

$$R = R_1 \phi R_2$$

where

$$\mu_R(x,z) = \max_y \{w_1 \mu_{R_1}(x,y) + w_2 \mu_{R_2}(y,z)\}, \quad \sum_i w_i = 1 \tag{1}$$
$$\text{for each } (x,y) \in R_1 \text{and each } (y,z) \in R_2 \ .$$

Replacing R_1 with the $\widetilde{\textbf{strength}}$ relation and R_2 with the transpose of $\widetilde{\textbf{contribute}}$ relation, $\widetilde{\textbf{contribute}}^{\mathbf{T}}$, the new relation obtained can be interpreted physically as measuring the relatedness of the DFs to the MFs. The degree of relatedness of the DFs to a MF is directly proportional to the degree of contribution of the DFs to the MF and the strength of the mating relationship between the DFs. The weights, w_i, represents the emphasis a designer may wish to place on the strength of mating relationships. In future discussion, we set $w_1 = 0.9$ and $w_2 = 0.1$. Consider Figure 3, we compute $R = \widetilde{\textbf{strength}} \ \ \phi \ \ \widetilde{\textbf{contribute}}^{\mathbf{T}}$.

$$\begin{aligned}
\mu_R(\mathrm{MF}_1, \mathrm{DF}_1) &= 0.9(1.0) + 0.1(1.0) = 1.0 \\
\mu_R(\mathrm{MF}_1, \mathrm{DF}_2) &= 0.9(0.2) + 0.1(1.0) = 0.28 \\
\mu_R(\mathrm{MF}_1, \mathrm{DF}_3) &= undefined \\
\mu_R(\mathrm{MF}_2, \mathrm{DF}_1) &= 0.9(0.2) + 0.1(0.8) = 0.26 \\
\mu_R(\mathrm{MF}_2, \mathrm{DF}_2) &= max(0.9(1.0) + 0.1(0.8), 0.9(0.7) + 0.1(0.2)) = 0.98 \\
\mu_R(\mathrm{MF}_2, \mathrm{DF}_3) &= max(0.9(0.7) + 0.1(0.8), 0.9(1.0) + 0.1(0.2)) = 0.92
\end{aligned}$$

The graph shows that DF1 is loosely connected to MF2 whereas DF3 is more tightly linked to MF2. This is reflected in the computed result where $\mu_R(\mathrm{MF}_2, \mathrm{DF}_1) = 0.26$ and $\mu_R(\mathrm{MF}_2, \mathrm{DF}_3) = 0.92$. Hence we say that the ϕ operator faithfully models the degree of relatedness between a DF and an MF.

Definition 2. (The θ operator)

Let F_1 be a binary fuzzy relation in the Cartesian product $X \times Y$ of universes U and V, respectively, and F_2 in the Cartesian product $Y \times X$ in universes V and U, respectively. A relation composition operator θ, denoted by $F = F_1 \theta F_2$, forms a new relation F with membership function μ_F:

$$\mu_F(x,z) = \frac{\sum_y (\mu_{F_1}(x,y) + \mu_{F_2}(y,z))}{2\ n_y} \tag{2}$$

$$\text{for each } (x,y) \in F_1 \text{ and each } (y,z) \in F_2$$

where n_y is the number of occurrence of y such that $(x,y) \in F_1$ and $(y,z) \in F_2$.

Graphically, the θ operator measures the average cost of all paths of length 2 that originate from a node $X_i \in U$, pass through an intermediate node $Y \in V$, and terminate in a node $X_j \in U$ (see Figure 4).

Given the definitions of ϕ and θ, the degree of coupling between any pair of MFs, denoted by the modularity relation $\widetilde{\mathbf{MM}}$, is computed as follows.

Definition 3.

$$\widetilde{\mathbf{MM}} = \widetilde{\mathbf{contribute}} \ \ \theta \ \ (\widetilde{\mathbf{strength}} \ \ \phi \ \ \widetilde{\mathbf{contribute}}^{\mathbf{T}}) \tag{3}$$

Then

$$\begin{aligned} \text{interface complexity} &= \max_{i,j,i \neq j} \{\widetilde{\mathbf{MM}}_{i,j}\} \qquad i,j \in \{1,2,\cdots,m\} &(4)\\ \text{group coherence} &= \min_{i} \{\widetilde{\mathbf{MM}}_{i,i}\} \qquad i \in \{1,2,\cdots,m\} &(5) \end{aligned}$$

Using the same example as in Figure 3, we compute:

$$\begin{aligned} \widetilde{\mathbf{MM}} &= \begin{pmatrix} (1.0+1.0)/2 & (1.0+0.26)/2 \\ (0.8+0.28)/2 & (0.8+0.98+0.2+0.92)/4 \end{pmatrix} \\ &= \begin{pmatrix} 1.0 & 0.63 \\ 0.54 & 0.725 \end{pmatrix} \end{aligned}$$

The time complexity for the computation of $\widetilde{\mathbf{MM}}$ is $O(n^2 m + nm^2)$ where n is the number of DFs and m is the number of MFs. Hence, the evaluation of interface complexity and group coherence measures takes polynomial time. In addition, by definitions 4-5, we see that the interface complexity measure is higher if then MFs are more tightly bonded together; whereas the group coherence measure is lower if the components within each individual MFs are less related to one another. Thus, the combination of the interface complexity and group coherence measures gives a good indication of the modularity of a design.

3.2 Degree of Standardization

In robotic assembly, the degree of standardization of mating features plays an important role in determining the success or failure of an assembly task. A mating feature is considered standard if it conforms to the design principles of: maximize part symmetry; improve assembly access; maximize part compliance; and provide parts with integral self-locking features. A design that uses standard mating features has a higher chance of being assembled correctly. Consider two designs as shown in Figure 5. Design A uses a standard peg-in-hole mating feature whereas Design B uses a non-standard insert mating feature. Clearly, Design A is much easier to assemble and is less costly than Design B.

To measure the degree of standardization of the mating features in a given design, we have adopted a stored-and-compared approach. A group of human designers are asked to come up with a list of mating features that are considered standard. This list is translated into an internal representation and stored in a generic library of standard mating features. A comparison procedure is then invoked to find the best match between the mating features of a given design and the standard mating features in the library. A high degree of match indicates a high degree of standardization.

The first issue to tackle using this approach is: what is the internal representation for storing the list of standard mating features? This internal representation must be flexible enough to allow storing of generic mating feature class characteristics; yet it must be specific enough to make comparison among different mating feature classes possible. A parametric constructive solid geometry representation is proposed. In the parametric CSG representation, the orientation matrix and the parameters of the primitive solids are variables. Example of a parametric representation is shown in

Figure 6. The parametric CSG representation can then be instantiated to become a normal CSG tree representation. Figure 7 shows the results of different instantiation of the same parametric CSG representation.

With this parametric CSG representation, the computation of the similarity between two mating features can be carried out as follows. First, a graph matching algorithm is used to find all standard mating features whose graph structure is the same as the mating feature to be compared. A unification process is then carried out to instantiate the variables to values. The instantiated CSG tree represents a solid. Standardization measure is then computed based on the similarity between the two solids.

Two solids are considered similar if the ratio of intersection volume to the union volume is close to 1 (see Figure 8). Let $S1$ denote the instantiated CSG description of the mating feature f. Let $S2_i$ denotes the instantiated CSG description of the ith standard mating feature in the generic library L. The geometric similarity measure of mating feature f, gs(f), is

$$gs(f) = \max_i\left(\frac{volume(S1 * S2_i)}{volume(S1 + S2_i)}\right.$$

where $S1 * S2_i$ denotes the intersection between $S1$ and $S2_i$, and $S1 + S2_i$ denotes the union between $S1$ and $S2_i$. gs(f) takes values between 0 and 1. If gs(f) is close to 1, it indicates f is very similar to a standard mating feature. On the other hand, if gs(f) is close to 0, then perhaps a redesign is needed to transform f into a standard mating feature. The standard mating feature that is geometrically the closest to f is denoted by f^*.

For a given design with N mating features, the degree of standardization is

$$\text{Degree of Standardization} = \frac{1}{N}\sum_{i=1}^{N} gs(f_i)$$

The drawback of this approach is that it is limited by the size and quality of the library L. Future research is needed to explore other ways of performing this evaluation such as finding the extent of symmetry directly from the CSG representation.

3.3 Degree of Uniformity

Uniformity implies a same and unchanging set of conditions. This is particular important in the context of robotic assembly where the changes in the direction of assembly and the numbers of assembly tools used should be kept to a minimum. Such a design takes less time to assemble, has a higher chance of being assembled successfully, and hence better for assembly.

This measure focuses on the impact of a design on the assembly process. With the advancement in automatic assembly plan generator, feasible assembly plans can now be generated automatically [25, 26, 27, 28]. An assembly plan is represented as a partial order graph $G = (V, E)$. G is a directed graph in which the vertices are partitioned into $k \geq 2$ disjoint sets $V_i, 1 \leq i \leq k$. In addition, if $< u, v >$ is an edge in E, then $u \in V_i$ and $v \in V_{i+1}$ for some $i, 1 \leq i < k$. We transform this graph into a multistage graph as follows: Two artificial nodes are introduced, $s \in V_0$ and $t \in V_{k+1}$, $|V_0| = |V_{k+1}| = 1$. s is the source and t is the sink. For each vertex $v \in V_1$, we add an edge from s to v. Similarly, for each vertex $w \in V_k$, we add an edge from w to t. The set of added edges is denoted by E'. A new graph results. This new graph, $G' = (V \cup \{V_0, V_{k+1}\}, E \cup E')$, is a multistage graph. Let $c(i, j)$ be the cost of edge $< i, j >$. We define

$$c(i,j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = \text{k} + 1; \\ -3 & \text{if the assembly direction and assembly tool of node } i \text{ is different} \\ & \text{from that of node } j; \\ -1 & \text{if either the assembly direction or assembly tool of node } i \text{ is} \\ & \text{different from that of node } j; \\ 0 & \text{otherwise} \end{cases}$$

The cost of a path from s to t is the sum of the costs of the edges on the path. Then, our problem of finding the degree of uniformity is reduced to the multistage graph problem of finding a minimum cost path from s to t.

An algorithm for solving the multistage graph problem is given in Figure 10. It uses dynamic programming technique [29]. For a graph with $|V|$ vertices and $|E|$ edges, the complexity of the algorithm is $O(|V| + |E|)$.

At the completion of the procedure, we obtain the degree of uniformity

$$\text{degree of uniformity} = -\text{Cost}(0)$$

4 The Redesign Suggestions Generator

Having performed an objective analysis on a design, we need to help the designer to search for feasible redesign alternatives. There are two types of redesign: local redesign (changes made to only one components) and global redesign (changes made to two or more components). Global redesign usually is needed to improve the degree of modularity while local redesign is usually effective in improving the degree of standardization and the degree of uniformity.

4.1 Global Redesign Suggestions Generation

An interactive approach is adopted where the computer interacts with the human designer to search for feasible redesign suggestions. The first step is to derive a new functional structure for the design. The human designer is presented with the graph of the the MFs-DFs and DFs-DFs relations. He/She is then asked to indicate a set of edges whose associated mating relationships should be relaxed as well as a set of nodes whose associated components can be decoupled. The new functional structure graph is called G'. G' is used as an index to retrieve similar cases in a case-based library [30]. The case-based approach has been utilized in a number of assembly planning research [31, 28, 32]. In these research, the primary concern is in the reuse and adaptation of old assembly plans. For our purpose, the focus is on the functional structure of the product.

Having retrieved all cases with similar functional structure graph, the next step is to check that the retrieved cases are functionally compatible with the current design. For each retrieved case, we have a set of design constraints specific to the given case. We say that the current design and the retrieved case are functionally compatible if the set of design constraints of the candidate case is fuzzy compatible to that of the current design. To determine the fuzzy compatibility of two sets, a fuzzy resolution refutation system is utilized. The set of functional constraints of current design forms the goal, $\widetilde{W}$, that we wish to prove. The set of design constraints of the retrieved cases and a set of inference rules form the set of well-formed formulas $\widetilde{F}$ from which we wish to prove the goal $\widetilde{W}$. First, we negate the goal, $\neg\widetilde{W}$, and add it to $\widetilde{F}$. We then use fuzzy resolution in an attempt to derive a contradiction. If a contradiction is derived, then $\widetilde{W}$ logically follows from $\widetilde{F}$ and we say the two sets are fuzzy compatible.

Definition 4

A fuzzy contradiction is derived if the resolvent,$\widetilde{R}$, is such that

$$height(\widetilde{R}) \leq 0.2$$

If during the resolution process, a fuzzy contradiction is derived, we say that the retrieved case is functionally compatible to the current design. After this step, all the functionally compatible cases are retrieved. The designer is asked to select one of the cases as the feasible redesign solution. The next redesign phase is then initiated.

4.2 Local Redesign Suggestions Generation

Local redesign suggestions aim to introduce changes that are confined to design features within one physical component. The objective is to improve the product design locally so that the degree of standardization and the degree of uniformity are improved.

The process of generating local redesign suggestions is shown in Figure 9.

The input to the suggestion generator includes: the CSG representation of the product design, the evaluation results as described in Section 3, the assembly plan, and the functional constraints to be satisfied by the product design. Generation of local redesign suggestions is performed by a rule-based inference engine. This inference engine interacts with a generic toolkit library to derive feasible redesign suggestions. First, the inference engine generates a list of possible redesign actions.

The corresponding redesign operators (in the form of routines in the generic toolkit library) are invoked to simulate the suggested redesign operations. The resultant redesign is checked to ensure none of the functional constraints are violated. If a functional constraint is violated, the redesign action is terminated and an alternate redesign action is initiated. At the end of the redesign process, a list of possible redesigns are displayed on screen.

Table 2 shows the list of local redesign operators available in the toolkit library. A sample of the rules used in the inference engine is given in Table 3

5 Result

A prototype system is being implemented on the Sunsparc workstation with graphic interface to an Iris workstation. The overall view of the system is shown in Figure 11.

A real-life example, the tilt-mechanism example, is used to illustrate how the evaluation process and the redesign process are performed. The objective of this design is to support a computer monitor such that it allows the monitor to tilt at a slight angle as determined by the user. Figure 12 shows the original design with eight components.

Based on the modularity analysis, it was found that the original design has a very high interface complexity measure. Hence, case-based reasoning is invoked to derive a global redesign suggestion. The suggested design is shown in Figure 13.

Local redesign analysis is then carried out. It was found that the degree of uniformity can be improved by changing the horizontal insertion of the rod into a top-down insertion. The final design is shown in Figure 14.

With this redesign, we reduced the number of assembly steps from 7 to 3 and the tilt mechanism exhibits a better modular structure. This is certainly a significant improvement in terms of cost for assembly.

6 Conclusion

In this paper, we have discussed an approach to product redesign for robotic assembly. The redesign can be performed at two levels: the global redesign level and the local redesign level. Three quantitative measures are defined to evaluate a design for robotic assemblability. The three measures are: degree of modularity, degree of standardization, and degree of uniformity. Based on the evaluation results, product-specific redesign suggestions are generated. The tilt-mechanism example is used to illustrate the promise of using the this approach for aiding in product redesign for robotic assembly.

References

[1] G. Boothroyd. *Design for Assembly Handbook.* University of Massachusetts, Amherst, 1980.

[2] M. M. Andreasen. *Design for Assembly.* Springer-Verlag, New York, 1988.

[3] J. L. Nevins, D. E. Whitney, and T. L. De Fazio. *Concurrent Design of Products and Processes: A Strategy for the Next Generation in Manufacturing.* McGraw-Hill Inc., New York, 1989.

[4] Nam Suh. *The Principles of Design.* Oxford University Press, New York, 1990.

[5] S. Frederic Guggenheim, editor. *Designers in Action 1985-1991.* ASME, New York, 1992.

[6] S. M. Kannapan and K. M. Marshek. Design synthetic reasoning. *Mech. Mach. Theory*, 26(7):711–739, 1991.

[7] Jack Mostow. Design by derivational analogy: Issues in the automated replay of design plans. *Artificial Intelligence*, 40:119–184, 1989.

[8] J. K. Blundell. Functional design. *Proc. of Manufacturing International '90*, 2:63–67, Mar. 1990.

[9] S. Kota and A. C. Ward. Functions, structures, and constraints in conceptual design. *The 1990 ASME Design Technical Conferences - 2nd International Conference on Design Theory and Methodology*, pages 239–250, Sept 1990. Chicago, Illinois.

[10] J. D. Watton and J. R. Rinderle. A method to identify reformulations of mechanical parametric constraints to enhance design. *Design Theory and Methodology DTM '90*, pages 77–84, Sept 1990. Chicago IL.

[11] B. L. Miles. Design for assembly - a key element within design for manufacture. *Proc. of the Institution of Mechanical Engineers Part D: Journal of Automobile Engineering*, 203:29–38, 1989.

[12] A.E.K. Holbrook and P. J. Sackett. Positive design advice for high precision, robotically assembled product. *Developments in Assembly Automation*, pages 181–190, Mar. 1988.

[13] J. Boorsma. Design for assembly. *Proc. of 8th Int'l Conf. Assembly Automation*, pages 141–148, Mar. 1987.

[14] J. R. Dixon. Designing with features: Building manufacturing knowledge into more intelligent cad systems. *Manufacturing*, pages 51–57, 1988.

[15] C. Poli, R. J. Graves, and J. E. Sunderland. Computer-aided product design for economical manufacture. *Manufacturing*, pages 23–25, 1988.

[16] J. R. Rinderle and V. Krishnan. Constraint reasoning in concurrent design. *The 1990 ASME Design Technical Conferences - 2nd International Conference on Design Theory and Methodology*, pages 53–62, Sept 1990. Chicago, IL.

[17] M. R. Cutkosky and D. R. Brown. Extending concurrent product and process design toward earlier design stages. *Concurrent Product and Process Design*, ASME, New York, 65–72, Dec 1989.

[18] S. N. Talukdar and S. J. Fenves. Towards a framework for concurrent design. *Concurrent Product and Process Design*, ASME, New York, 35–40, Dec 1989.

[19] R. M. Noller. Design for disassembly tactics. *Assembly*, pages 24–26, January 1992.

[20] Keith Rathmill, editor. *Robotic Assembly*. IFS Ltd, Kempston, Bedford, 1985.

[21] Hubert K. Rampersad, editor. *Integrated and simultaneous Design for Robotic Assembly*. Wiley and Sons Inc., New York, 1994.

[22] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *International Journal Fuzzy Sets Systems*, 1(1):3–28, 1978.

[23] George J. Klir and Tina A. Folger. *Fuzzy Sets, Uncertainty, and Information*. Prentice Hall, Englewood Cliffs, N. J., 1988.

[24] Arnold Kaufmann and Madan M. Gupta. *Introduction to Fuzzy Arithmetic: Theory and Applications*. Van Nostrand Reinhold, New York, 1991.

[25] L. S. Homem de Mello and A. C. Sanderson. And/or graph representation of assembly plans. *IEEE Trans. on Robotics and Automation*, 6(2):188–199, Apr. 1990.

[26] Y. F. Huang and C. S. G. Lee. A framework of knowledge-based assembly planning. *Proc. of 1991 IEEE Int'l Conf. on Robotics and Automation*, 2:599–604, Apr. 1991. Sacramento, CA.

[27] S. Lee and Y. G. Shin. Assembly planning based on subassembly extraction. *Proc. of 1990 IEEE Int'l Conf. on Robotics and Automation*, 3:1606–1611, 1990. Cincinnati, OH.

[28] P. Pu and L. Purvis. Assembly planning using case adaptation methods. *Proc. of 1995 IEEE Int'l Conf. on Robotics and Automation*, 1:982–983, 1995.

[29] Ellis Horowitz and Sartaj Sahni. *Fundamentals of Computer Algorithms*. Computer Science Press, Potomac, Md., 1984.

[30] Wynne Hsu and C. S. G. Lee. Paradigm shift and the integrated feedback approach. *Proc. of 1995 IEEE Int'l Conf. on Robotics and Automation*, 3:2853–2858, 1995.

[31] S. Lee, G. J. Kim, and G. A. Bekey. Combining assembly planning with redesign: An approach for more effective dfa. *Proc. of 1993 IEEE Int'l Conf. on Robotics and Automation*, 1:319–325, 1993.

[32] A. Swaminathan and K. S. Barber. Ape: An experience-based assembly sequence planner for mechanical assemblies. *Proc. of 1995 IEEE Int'l Conf. on Robotics and Automation*, 2:1278–1283, 1995.

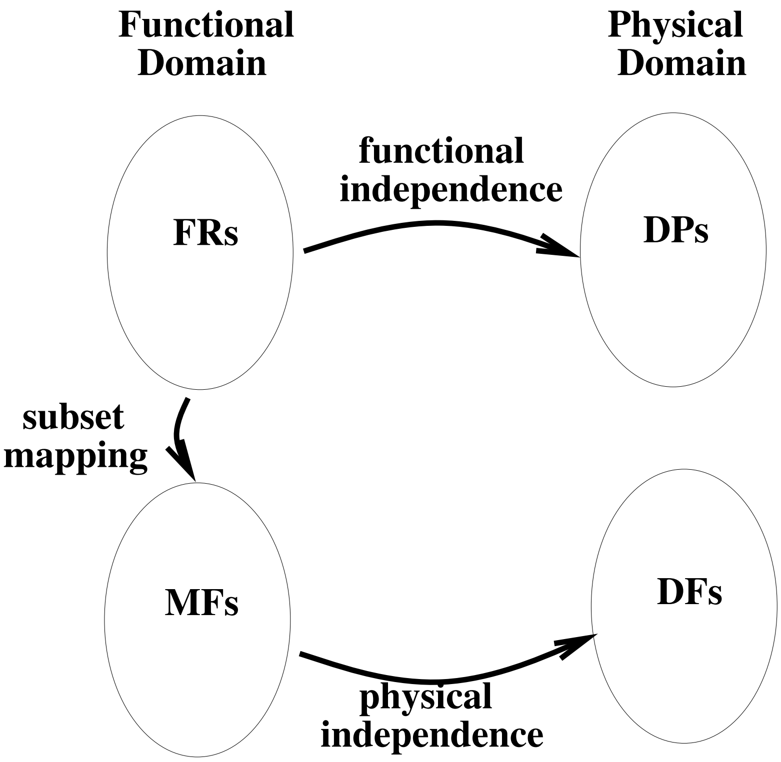


Figure 1: The Functional to Physical Domain Mappings.

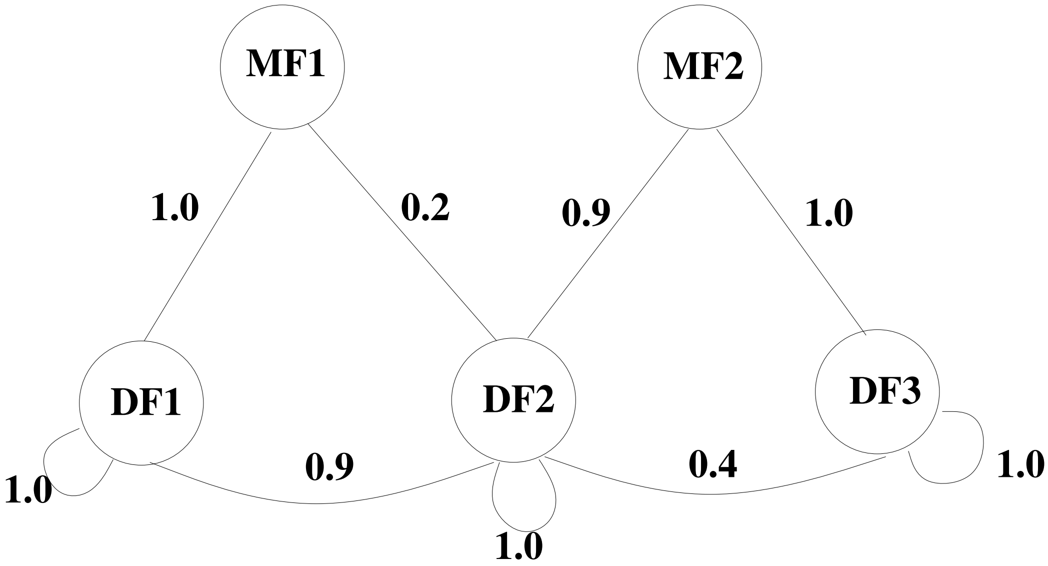


Figure 2: The Graphical Representation Example.

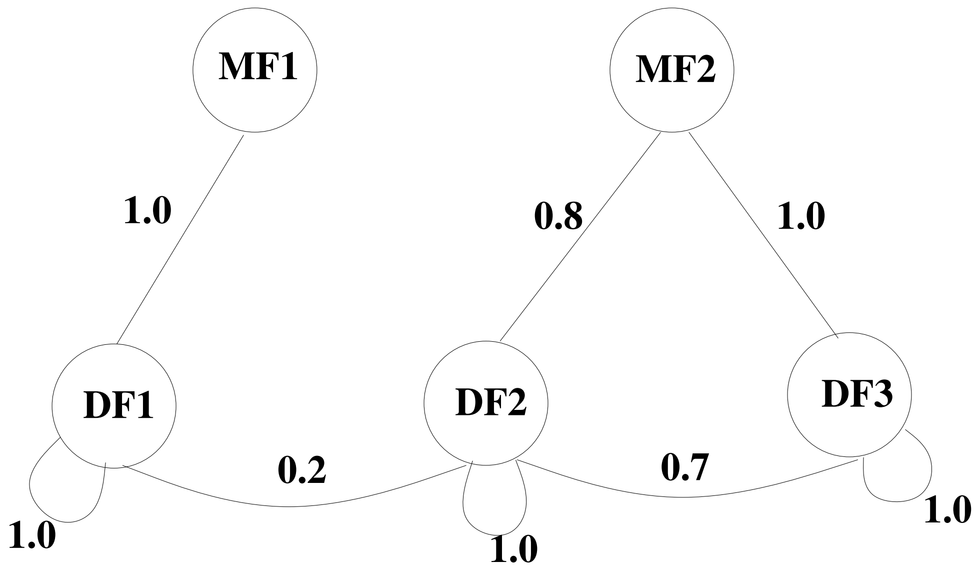
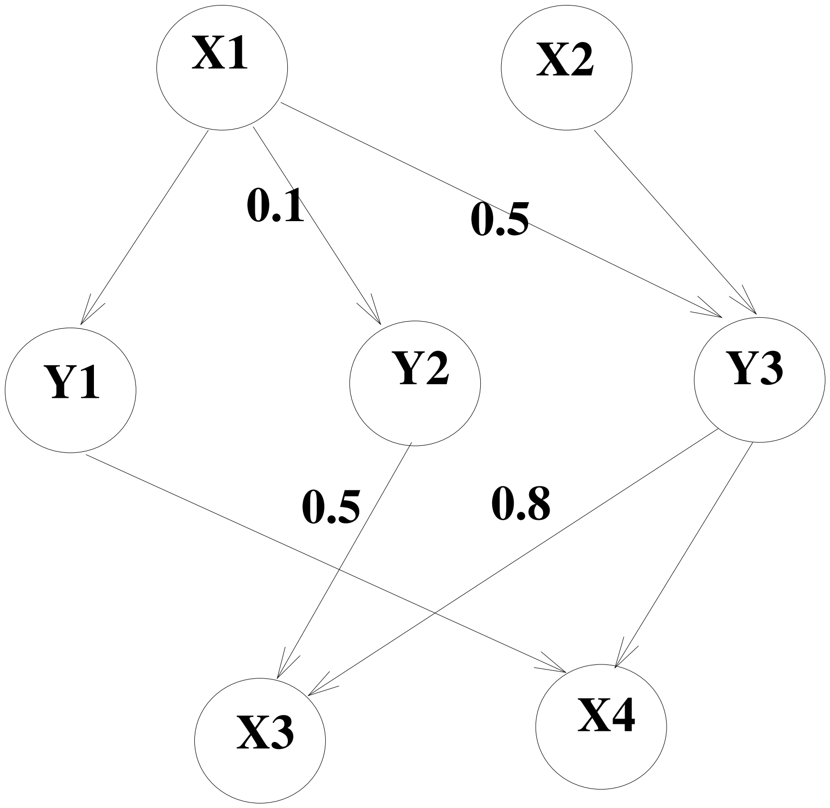


Figure 3: Graphical Interpretation of Operator ϕ.



$$\mu_F(X1,X3) = (0.1+0.5+0.5+0.8)/4 = 0.475$$

Figure 4: Graphical Interpretation of Operator θ.

Figure 5: The Comparison of Two Designs

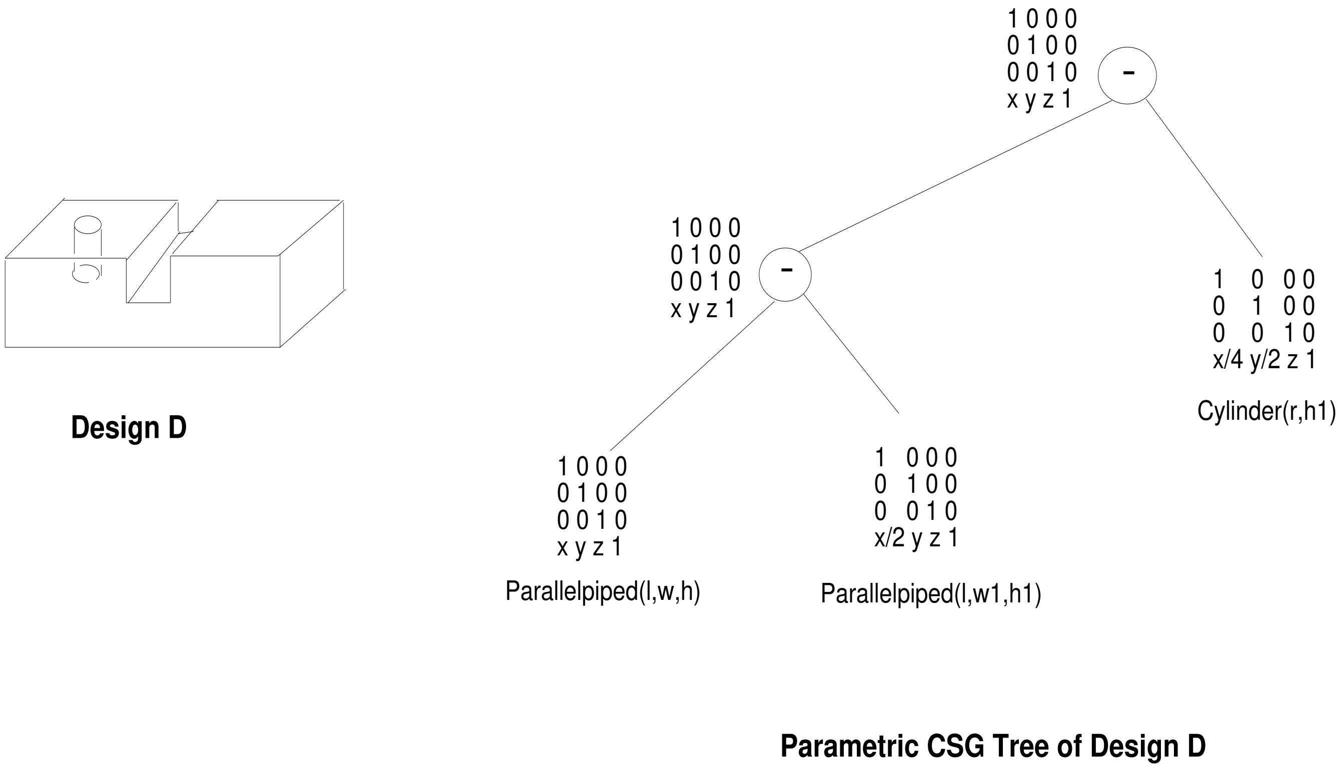


Figure 6: The CSG Tree Example

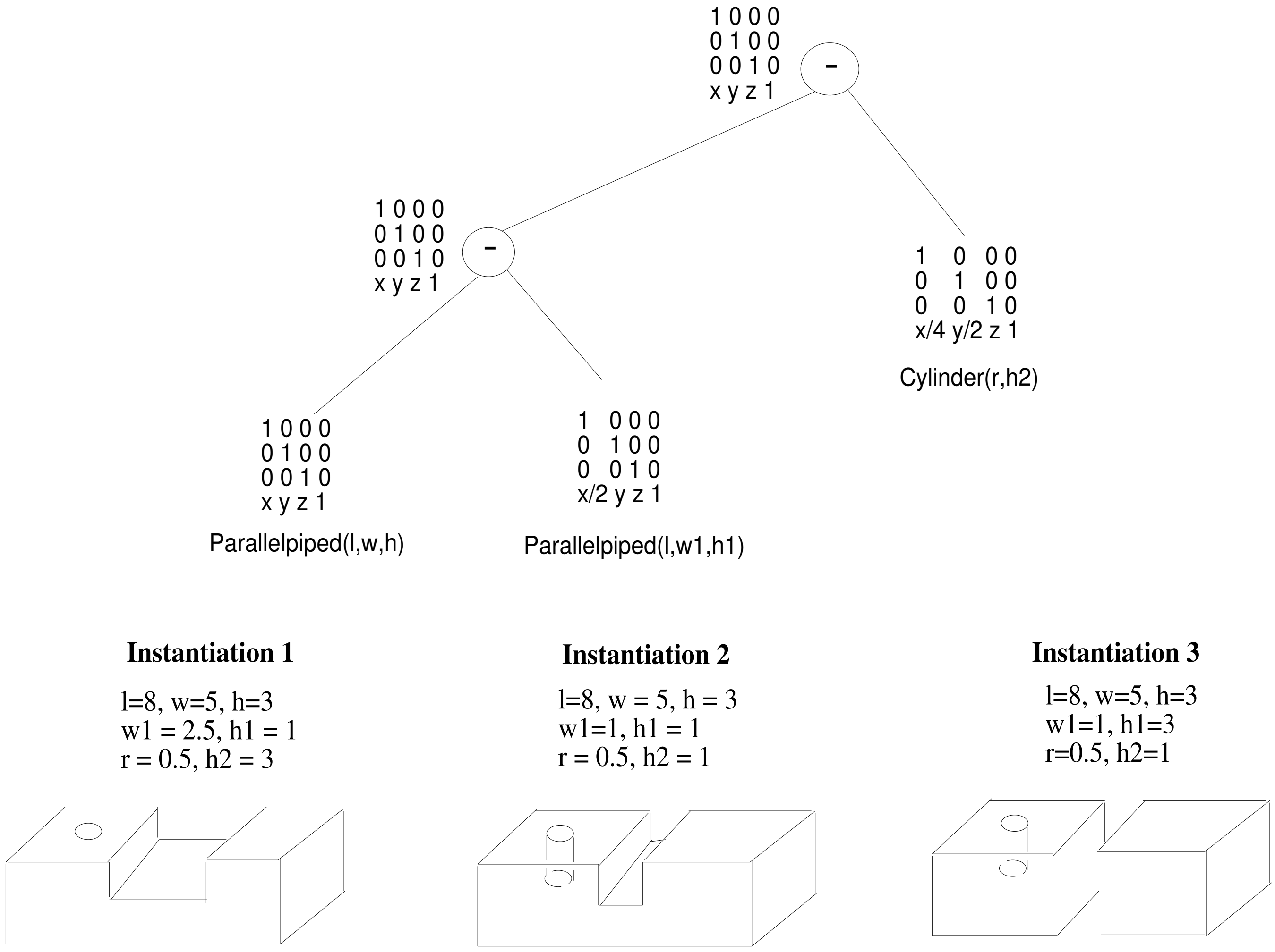


Figure 7: The Instantiations of a Parametric CSG Tree

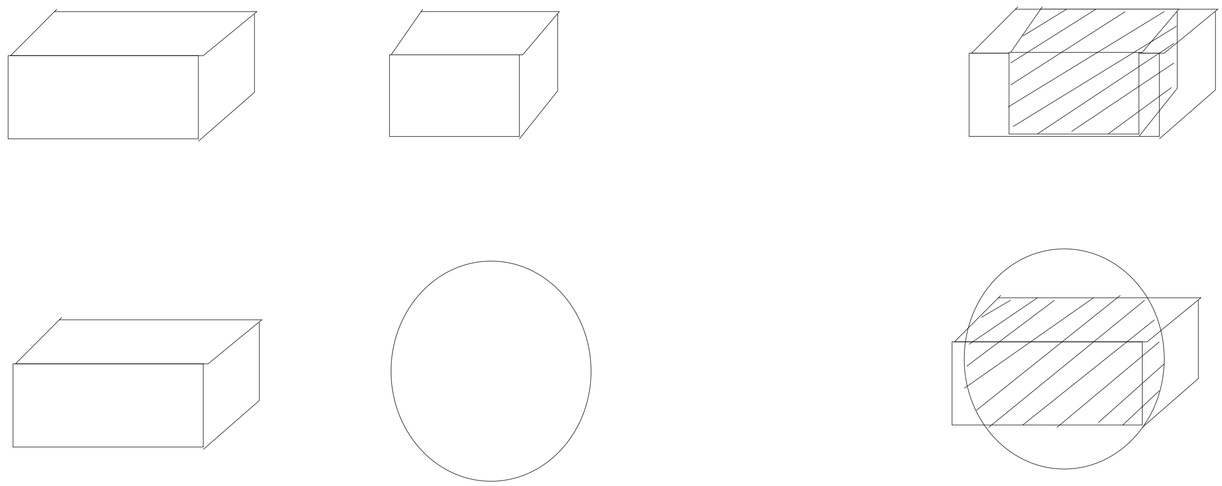


Figure 8: Intersection Volume and Union Volume of Solids

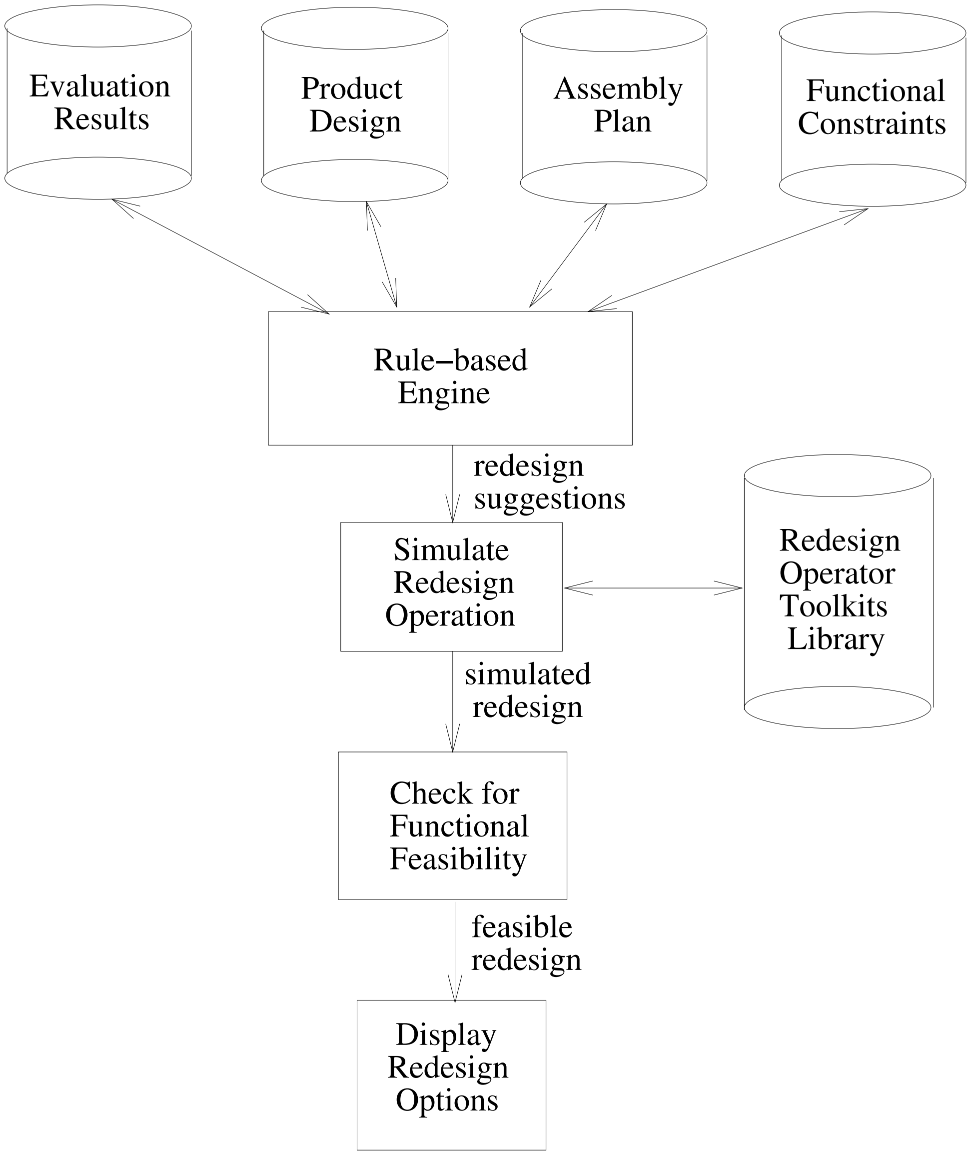


Figure 9: The Redesign Process

```
procedure Uniformity(E, k, n, P)
The input is a k stage graph with n vertices indexed in order of
stages. E is a set of edges and c(i, j) is the cost of edge < i, j >.
P is a minimum cost path.
Cost(n) ← 0
for j ← n - 1 to 0 step -1
    let r be a vertex such that < j, r > ∈ E and c(j, r) + COST(r) is minimum
    Cost(j) ← c(j,r) + COST(r)
    D(j) ← r
endfor
P(1) ← 1, P(k) ← n
for j ← 2 to k - 1
    P(j) ← D(P(j-1))
endfor
end Uniformity
```

Figure 10: Algorithm for Multistage Graph Problem

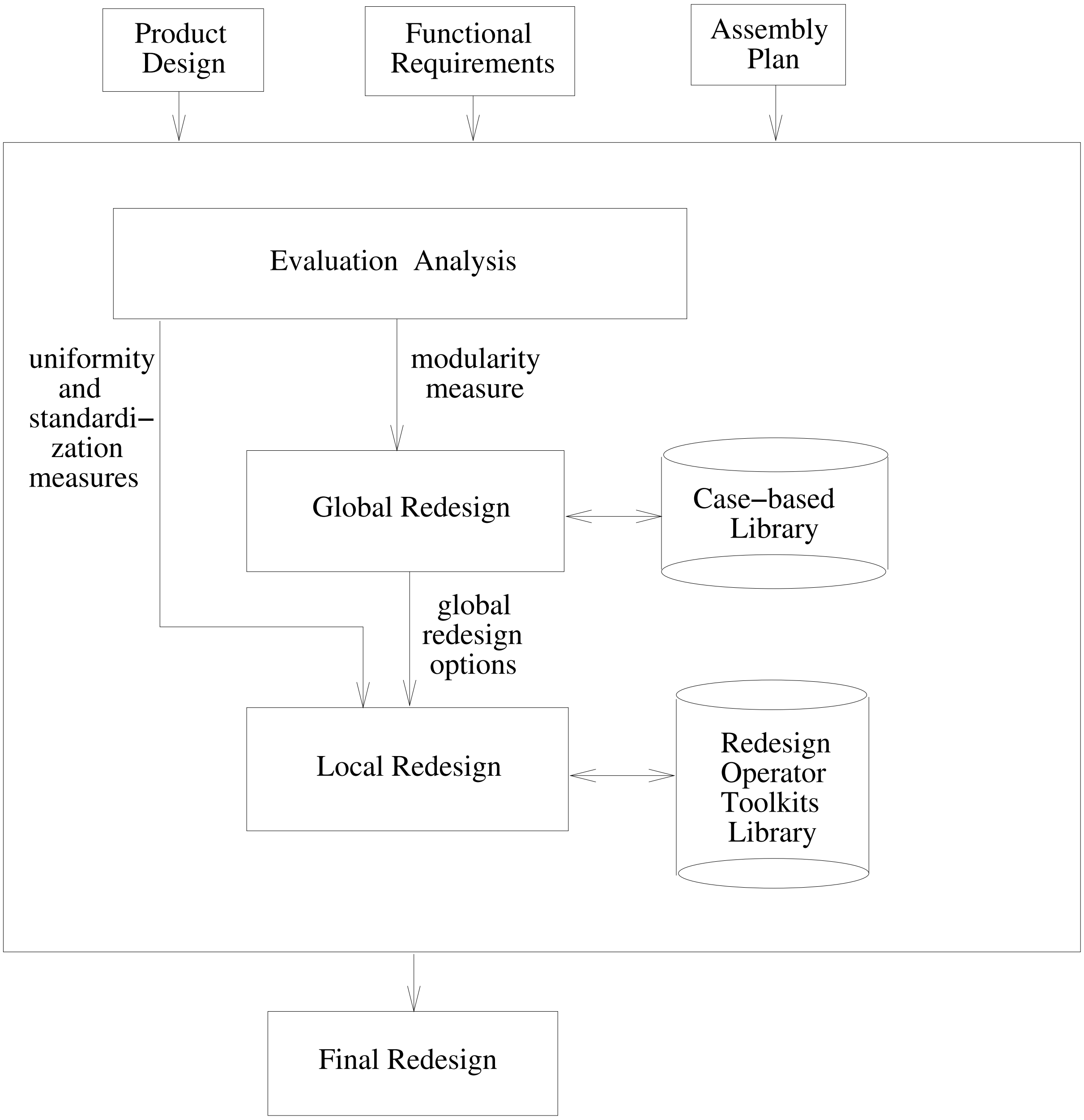


Figure 11: The Overall View of the Prototype System

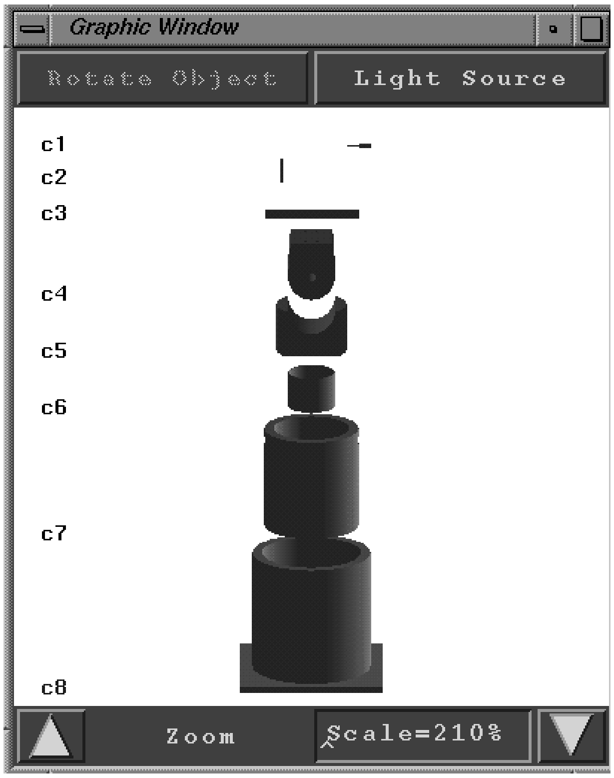


Figure 12: The Tilt Mechanism Example

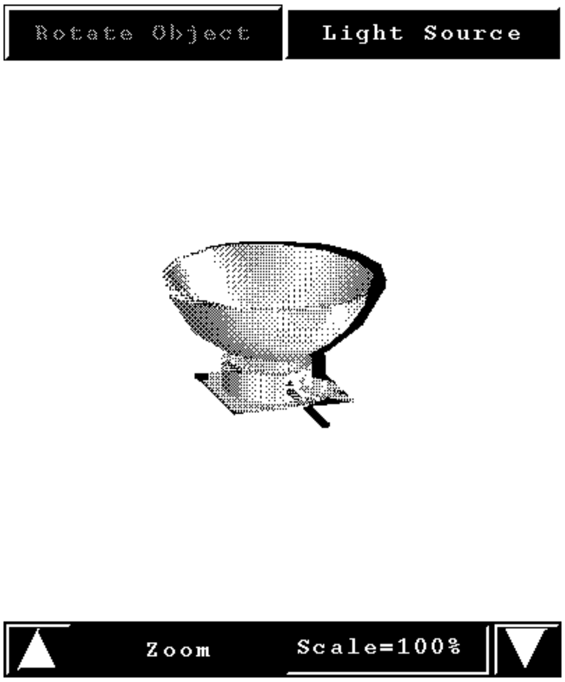


Figure 13: The Global Redesign for the Tilt Mechanism Example

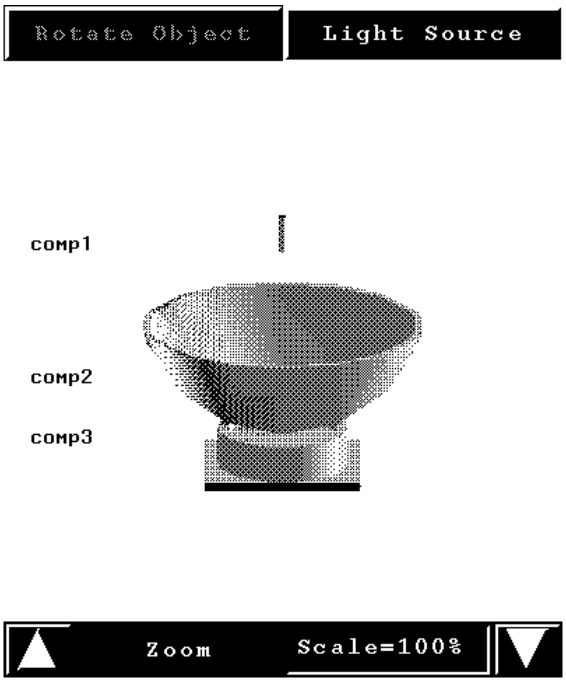


Figure 14: The Tilt Mechanism Redesign

Table 1: Comparison of the Various Approaches.

Approaches	Description	Characteristic
Concurrent Engineering	Take later life-cycle issues into consideration in design phase.	Forward analysis
Design for Assembly (Quantitative)	Assign time, costs, numerical code to parts/subassemblies.	Static analysis
Design for Assembly (Qualitative)	Provide high-level rules and guidelines	General, not tailored to specific product
Our Approach	Provide redesign aid through the the integration of design and assembly phases.	Backward analysis Gives product-specific redesign suggestions.

Redesign Operator	Description
change-primitive(C,P)	Change the CSG primitive of component C to P.
change-orientation(C, O)	Change the orientation matrix of component C to O. Care must be taken to ensure that the change does not result in an invalid solid.
change-parameter(C,P,p-name, p-value)	Change the parameter p-name of CSG primitive P of component C to p-value.
combine(C1, C2)	Combine two components to form one integrated component. Care must be taken to ensure the union is a valid solid.
split(C, d)	Split the component C along the axis denoted by d.

Table 2: Table of Available Local Redesign Operators

If geometric-similarity(F,P) $>$ threshold and DesignFeature(F,C), then change-primitive(C, P).
If orientation-similarity(F,O) $>$ threshold and DesignFeature(F,C), then change-orientation(C, O).
If change-direction(G,d) $>$ threshold and Component(C), then split(C, d).
$\vdots$

Table 3: Sample of Inference Rules used in the Redesign Process