Towards an Optimal Degree of Design Coupling Through Fuzzy Analysis

Wynne Hsu
DISCS
National University of Singapore
Singapore 119260
Republic of Singapore

C. S. G. Lee
Department of EE
Purdue University
West Lafayette, IN 47907
USA

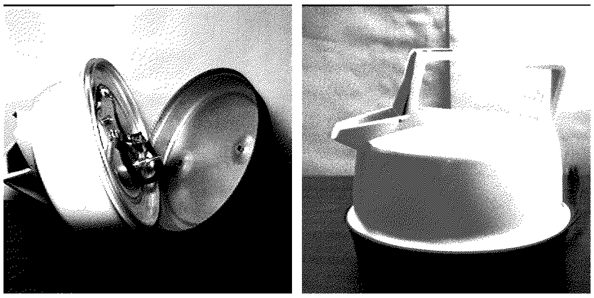
ABSTRACT

It has long been recognized that we need some fundamental, correct principles and methodologies to guide decision making in design. One important decision to be made in design is the determination of the optimal degree of design coupling among the components. Two components are physically coupled together if the design features of both components have been integrated to form one single physical component. This ability to decide on the optimal degree of design coupling has a direct impact on the number of components in the final design which, in turns, affects the maintainability, assemblability, testability, and and adaptability of the resultant product. Design for assembly guidelines tell us to reduce the number of components in the final design. This implies that we should, as far as possible, strive for the highest degree of physical integration of parts. This may prove to be too simplistic a guideline. In this paper, we attempt to show that there exists circumstances where design coupling is undesirable. We then propose a systematic approach for determining the optimal degree of design coupling among the components.

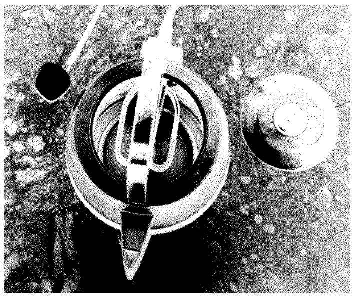
[0]This work was supported in part by the National University of Singapore Research Project Grant RP940644

1 Introduction

The challenge of industry today is to increase its competitiveness through lowering of production cost while maintaining the quality standard of its product. Studies show that 75% of the product cost is determined at the design phase [16]. It is, therefore, evident that improvement must be made at the design phase in order to effectively lower the cost of production. Much work has been done in the area of design, ranging from the axiomatic approach proposed by Suh [24] to the application-oriented research such as case study of successful designs [9]. In addition, researchers have also studied the effects of design on downstream activities such as assembly and manufacturing tasks [6, 1, 19]. All these studies came to a general consensus that a modular design is good. However, there is a lack of agreement on what constitutes a modular design. Suh attempted to give a guideline concerning design coupling in the following corollary: Two parts should be integrated provided functional independence is achieved. Unfortunately, this corollary does not take into account practical considerations such as maintainability, assemblability, and testability.



Design A



Design B

Figure 1: Design A: heating coil is totally integrated with kettle body; Design B: heating coil is a separate component from kettle body

Consider the two kettle designs as shown in Figure 1. Design A integrates the heating coil with the kettle body while Design B chooses to separate the heating coil from the kettle body. Design A is preferred according to Suh's corollary. However, an analysis of the two designs indicate that Design B is preferred over Design A in real-life situations because: (1) the welding of the heating coil to the kettle body (Design A) is more expensive to perform than a simple insertion of the heating coil into the kettle body (Design B). (2) it is easier to replace a faulty heating coil in Design B than in Design A. (3) if a new type of heating coil, that consumes less electricity, becomes available, Design B can easily replace its old heating coil with the new coil, but not so for Design A. The reason is because Design B is more modular than Design A. Briefly, a modular design is one where there is a well-defined physical boundaries between a group of components realizing one function from another group of components realizing another function.

The advantage of a modular design can be argued from three perspectives: the design for assembly perspective, the design for testing and maintenance perspective, and the design for adaptability perspective.

Design for Assembly
One primary objective in design for assembly is to develop a design which consists of the maximum number of independent subassemblies. An independent subassembly is one whose components can be assembled independently of another subassembly until the final stage of assembly. Such a design allows more parallel assembly operations thereby shortening the time for assembly. A modular design tends to maximize the number of independent subassemblies by reducing the number of mating relationships across different groups of components and, therefore, is good for assembly.

Design for Testing and Maintenance
The ability to isolate problem areas and the ease of access to these areas are important criterion in design for testing and maintenance. Tight physical mating relationships can cause functional coupling and this leads to headache in testing and maintenance. In a modular design, such physical and functional couplings are minimized. Hence, the replacement of faulty components will not cause any undesirable side effects. This results in ease of testing and maintenance.

Design for Flexibility
Design for Adaptability is closely linked to the fact that a adaptive design is one which can take advantage of new technology easily. Hence, with a modular design where each group of components has a well-defined boundary, replacing one group of components (with old material and/or old technology) with another group of components (with new material and/or new technology) presents no problem at all.

To take advantage of the many good points offered by a modular design, a two-phase approach has been proposed. During the first phase, a fuzzy analysis process is carried out to determine the modularity of the initial design. If the evaluation results indicate that the design needs to be improved, a redesign module is invoked in the second phase. The objective of the redesign module is to generate alternative redesign solutions that will result in an improvement in modularity. Outline of this paper is as follow: Section 2 gives an overview of the proposed method. In Section 3, the computational analysis of the degree of modularity is discussed. Section 4 describes the process of generating suitable redesign suggestions. The implementation and results are discussed in Section 5. Section 6 summarizes the related work and finally we conclude in Section 7.

2 Overview of the Proposed Approach

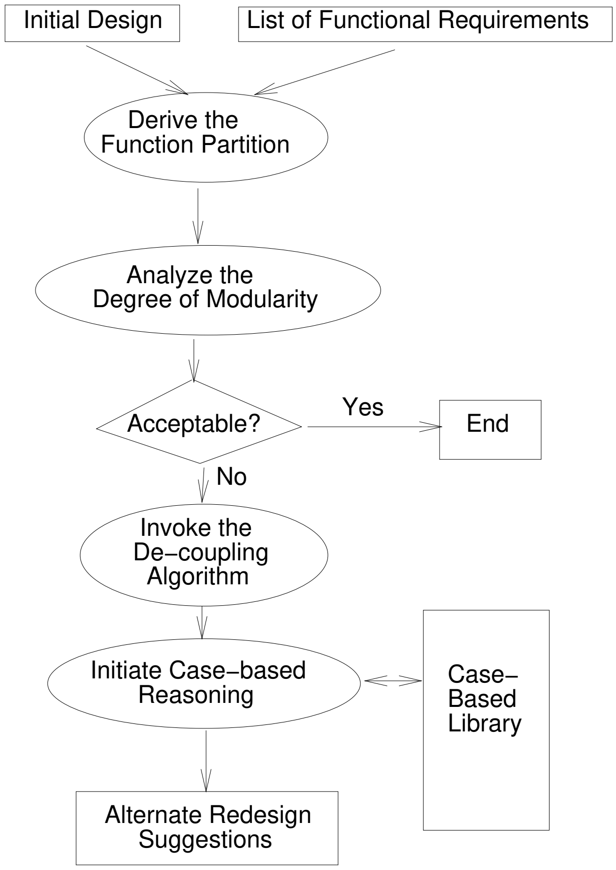


Figure 2: An Overview of the Proposed Approach

Figure 2 gives an overview of our proposed approach. The inputs are the initial design and a list of functional requirements to be achieved by the design. Recall that a modular design is one where the physical boundaries of the components/subassemblies are well-defined with respect to some function grouping. Hence, our first step is to determine an appropriate function grouping, called a function partition. A function partition is a collection of subsets of functional requirements, FR = $\{fr_1, fr_2, \cdots, fr_n\}$, that satisfies the following two conditions:

1. The fr_i's within each subset are all functional relevant to each other. Such a subset is called a function module, fm.

2. All the function modules are mutually exclusive and the union of all the function modules gives the set FR.

For example, suppose an electric kettle has the following functional requirements: {"to boil","to contain","to keep-warm", "to pour","to fill"}. A feasible function partition is: {fm_1 = ("boil","keep-warm"), fm_2 =("contain"), fm_3 =("pour","fill")}.

With the derivation of the function partition, next we analyze the modularity of the design by its well-definedness of the physical boundaries with respect to this function partition. A boundary is well-defined if the parts/subassemblies that realized a fm do not also contribute to the realization of another fm and the number of mating relationships between parts/subassemblies of one fm to the parts/subassemblies of another fm is minimal. Consider the functional structure graph of two designs (see Figure 3). We see that Design C has much more cross-couplings between one

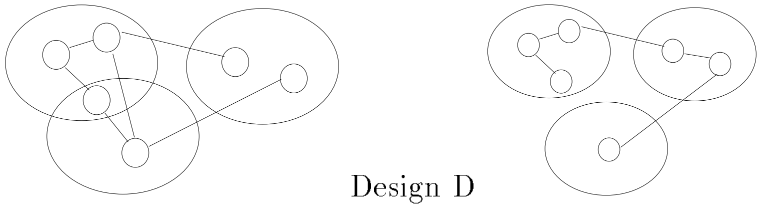


Figure 3: The Functional Structure Graph of Two Designs

fm and another fm whereas Design D's boundaries are clearer defined. We say Design D is more modular than Design C.

Fuzzy set theory has been utilized to quantify these degree of cross-couplings to give an objective evaluation of the modularity of a design. If a given design does not satisfy the modularity evaluation criterion, a redesign process is initiated. A proposed de-coupling algorithm is invoked to improve the modularity of the existing structure. The suggested structure is used as an index to retrieve past design cases from a case-based library. These are then adapted and offered to the designer as possible redesign alternatives.

3 Derivation of the Functional Partition

The first step in the approach is to derive the function partition of the given design. A list of evaluation parameters (ep), that will be used to verify the correctness of the design, is set up. In the kettle example, possible evaluation parameters are {final temperature of fluid, fluid volume, rate of heating, rate of pouring}. A two dimensional array is constructed, with the rows denoting the fr's and the columns denoting the ep's. Depending on the relevancy between fr_i and ep_j, the designer assigns a value ranging from 0 to 1 to the (i, j)th entry of the array. An entry of 1 signifies complete relevancy while an entry of 0 signifies no relevancy at all. Below shows an example of the 2D array for an automatic electric kettle.

	temperature	volume	rate of heating	rate of pouring/filling
to boil	1.0	0	0.7	0.2
to keep warm	1.0	0.3	0.7	0.2
to contain	0	1.0	0.2	0.2
to pour	0.2	0.2	0.5	1.0
to fill	0.2	0.6	0.5	1.0

Given this array, a clustering operation is then performed to partition the frs according to their relevancy similarities to the ep's. Many clustering methods have been proposed such as the fuzzy c-means methods [3] and the fuzzy equivalence relations methods [26, 15]. The fuzzy c-means method requires the number of clusters to be predetermined. This requirement makes it unsuitable for our application since we do not know beforehand the number of partitions that will result. Hence, the fuzzy equivalence relations method is adopted. Given a set of frs. We define a fuzzy compatibility relation, R, as follows:

$$R(x_i, x_j) = 1 - \frac{1}{n}\sum_{k=1}^{n}(|x(i,k) - x(j,k)|)$$

where $\cap$ denotes the interaction operation.

Using this formula, the compatibility relation is:

$$R = \begin{pmatrix} 1 & 37/40 & 15/40 & 20/40 & 16/40 \\ 37/40 & 1 & 18/40 & 21/40 & 19/40 \\ 15/40 & 18/40 & 1 & 19/40 & 23/40 \\ 20/40 & 21/40 & 19/40 & 1 & 36/40 \\ 16/40 & 19/40 & 23/40 & 36/40 & 1 \end{pmatrix}$$

Partition is performed based on the concept of compatibility. The frs which are more similar in characteristics are grouped together. To perform this partition, we need to define the concept of transitive closure.

Definition The max-min transitive closure of a fuzzy binary relation R is

$$\widetilde{R} = R \cup R^2 \cup R^3 \cup \cdots$$

where $R^2 = R \circ R$ is the max-min composition of R.

Definition Let R be any fuzzy binary relation. If for some k, the max-min composition $R^{k+1} = R^k$, then the max-min transitive closure is

$$\widetilde{R} = R \cup R^2 \cup R^3 \cup \cdots \cup R^k.$$

In our example, the series of max-min compositions of the compatibility matrix are:

$$R^2 = \begin{pmatrix} 1 & 37/40 & 19/40 & 21/40 & 20/40 \\ 37/40 & 1 & 19/40 & 21/40 & 21/40 \\ 19/40 & 19/40 & 1 & 23/40 & 23/40 \\ 21/40 & 21/40 & 23/40 & 1 & 36/40 \\ 20/40 & 21/40 & 23/40 & 36/40 & 1 \end{pmatrix}$$

$$R^3 = \begin{pmatrix} 1 & 37/40 & 20/40 & 21/40 & 21/40 \\ 37/40 & 1 & 21/40 & 21/40 & 21/40 \\ 20/40 & 21/40 & 1 & 23/40 & 23/40 \\ 21/40 & 21/40 & 23/40 & 1 & 36/40 \\ 21/40 & 21/40 & 23/40 & 36/40 & 1 \end{pmatrix}$$

$$R^4 = \begin{pmatrix} 1 & 37/40 & 21/40 & 21/40 & 21/40 \\ 37/40 & 1 & 21/40 & 21/40 & 21/40 \\ 21/40 & 21/40 & 1 & 23/40 & 23/40 \\ 21/40 & 21/40 & 23/40 & 1 & 36/40 \\ 21/40 & 21/40 & 23/40 & 36/40 & 1 \end{pmatrix}$$

$$R^5 = \begin{pmatrix} 1 & 37/40 & 21/40 & 21/40 & 21/40 \\ 37/40 & 1 & 21/40 & 21/40 & 21/40 \\ 21/40 & 21/40 & 1 & 23/40 & 23/40 \\ 21/40 & 21/40 & 23/40 & 1 & 36/40 \\ 21/40 & 21/40 & 23/40 & 36/40 & 1 \end{pmatrix}$$

R^4 is the transitive closure and it induces the following function partition at its α-cut ($\alpha = 36/40$): {(boil, keep-warm),(contain),(pour,fill)}. Thus, the function partition consists of three function modules.

3.1 Modularity Analysis

In this section, we look at how to quantitatively evaluate the degree of modularity of a given design once the function partition has been found. Recall that a modular design is one where the physical boundaries of the realized parts/subassemblies are well-defined with respect to its function module. To quantify this degree of well-definedness, two fuzzy relations are defined. The first fuzzy relation, denoted by $\widetilde{contribute}$, captures the degree of membership of a part/subassembly belonging to a fm.

$$\widetilde{contribute}(x_i, y_j) = \{(\mu_{contribute}, x_i, y_j)\}$$

where $\mu_{contribute}$ represents the amount of contribution by part/subassembly y_j to the function module x_i.

The second relation, denoted by $\widetilde{strength}$, captures the degree of mating strength between the parts/subassemblies.

$$\widetilde{strength}(x_i, x_j) = \{(\mu_{strength}, x_i, x_j)\}$$

where $\mu_{strength}$ denotes the degree of difficulty in disassembly part/subassembly x_i from x_j.

These two relations allow us to evaluate: (1) the degree of cohesiveness of the components within each fm (the cohesion measure), and (2) the cumulative strength of the mating relationships that cross from one fm to the other fm, (the coupling measure). The combination of the cohesion and coupling measures give a good indication of the well-definedness of the physical boundaries of the parts/subassemblies and hence, the modularity of a design.

The Cohesion Measure
Let C_i denote the crisp set of parts/subassemblies that contribute to fm_i. The degree of cohesiveness of the set C_i is defined to be

$$coh(C_i)) = \frac{1}{|C_i||C_i - 1|}\sum_{c\in C_i}([\mu_{contribute}(c, fm_i)[\sum_{d\in C_i, d\neq c}\mu_{strength}(c,d)]) \qquad (1)$$

The rational for the formula is this: a function module is cohesive if (1) its realized parts/subassemblies all contribute significantly to its functions (that is $\mu_{contribute}$ is close to 1) and, (2) all the realized parts/subassemblies are tightly mated to each other to form a stable subassembly. In other words, the second summation should be as close to $|C_i - 1|$ as possible. Thus, if $coh(C_i)$ is close to 1, it implies the function module fm_i is very cohesive. The cohesion measure of the design is then defined, conservatively, as

$$\text{cohesion measure} = min_i\ \{coh(C_i)\} \qquad (2)$$

The Coupling Measure
Let C_i denote crisp set of parts/subassemblies that contribute to fm_i and, C_j denote the crisp set of parts/subassemblies that contribute to fm_j, $i \neq j$. The degree of coupling between fm_i and fm_j is defined to be

$$cou(C_i, C_j) = w_1\frac{|C_i \cap C_j|}{|C_i \cup C_j|} + w_2\sum_{\substack{a\in C_i, b\in C_j\\ a,b\notin(C_i\cap C_j)}}\frac{\mu_{strength}(a,b)}{|C_i \cup C_j|} \qquad (3)$$

where $w_1 + w_2 = 1$. The formula consists of two parts. The first part determines the coupling due to the overlapping of parts/subassemblies in both fm_i and fm_j. The second part determines the coupling in the form of mating relationships that cross from fm_i to fm_j or vice versa. Thus, if $cou(C_i, C_j)$ is close to 1, it implies the coupling between fm_i and fm_j is very high. The coupling measure of the design is then defined as

$$\text{coupling measure} = max_{i,j}\ \{cou(C_i, C_j)\} \qquad (4)$$

Figure 4 shows the $\widetilde{contribute}$ and $\widetilde{strength}$ relations for an automatic electric kettle.

Using formulae (1-4), we compute the cohesion and coupling measures as follows:

$$\begin{aligned}
coh(C_1) &= \frac{1}{12}\{1.0(0.8+0.8) + 1.0(0.8+1.0) + 0.2(1.0+0.8+1.0) + 0.2(1.0)\}\\
&= 0.35\\
coh(C_2) &= \frac{1}{6}\{1.0(1.0+1.0) + 0.1(1.0) + 0.3(1.0)\}\\
&= 0.4\\
coh(C_3) &= 0\\
cohesion &= min(4.16/12, 2.4/6, 0) = 0
\end{aligned}$$

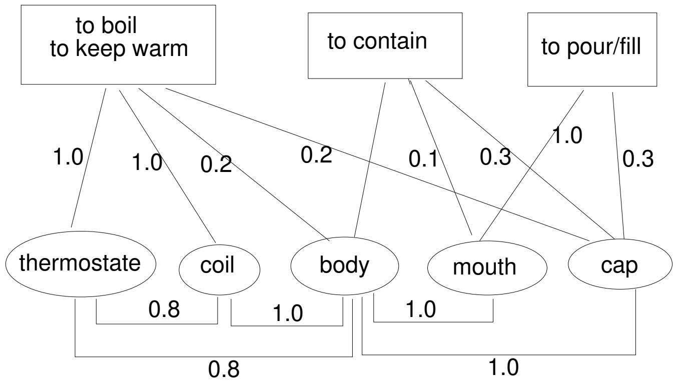


Figure 4: The $\widetilde{contribute}$ and $\widetilde{strength}$ Relations for the Kettle Example

$$
\begin{aligned}
cou(C_1, C_2) &= 0.5(2/5) + 0.5(0) = 0.2 \\
cou(C_1, C_3) &= 0.5(1/5) + 0.5(1/5) = 0.2 \\
cou(C_2, C_3) &= 0.5(1/3) + 0.5(1/3) = 0.33 \\
coupling &= max(0.2, 0.2, 0.33) = 0.33
\end{aligned}
$$

The computation results of the cohesion and coupling measures indicate that the automatic electric kettle design as shown in Figure 4 is not very modular and needs to be redesigned.

4 The Redesign Process

In this section, we examine a feasible approach to generate suitable redesign suggestions that will improve the modularity of a given design. In general, a badly-coupled design can be de-coupled in many ways. Figure 5 shows a few ways to de-couple the design with different impact on the cohesion and coupling measures.

Our objective is to de-couple the design such that the cohesion measure is maximized while the coupling measure is minimized. We propose a de-coupling algorithm that is based on the greedy approach. The algorithm is given below.

1) Let the contribute matrix be $\widetilde{M}$
2) Let $\alpha = min_i(max_j \widetilde{M}(i,j)$
3) Form $\widetilde{M}_\alpha$.
4) For $i = 1$ to row do
5)   if ith row of $\widetilde{M}_\alpha$ has with more than one 1's
6)   if there exists a column that is covered solely by this row,
7)   set that column to be 1 and set all the others to 0.
8)   otherwise
9)   for each column of 1's do
10)   compute the cohesion and coupling measure as if this
11   is the only 1 in row i

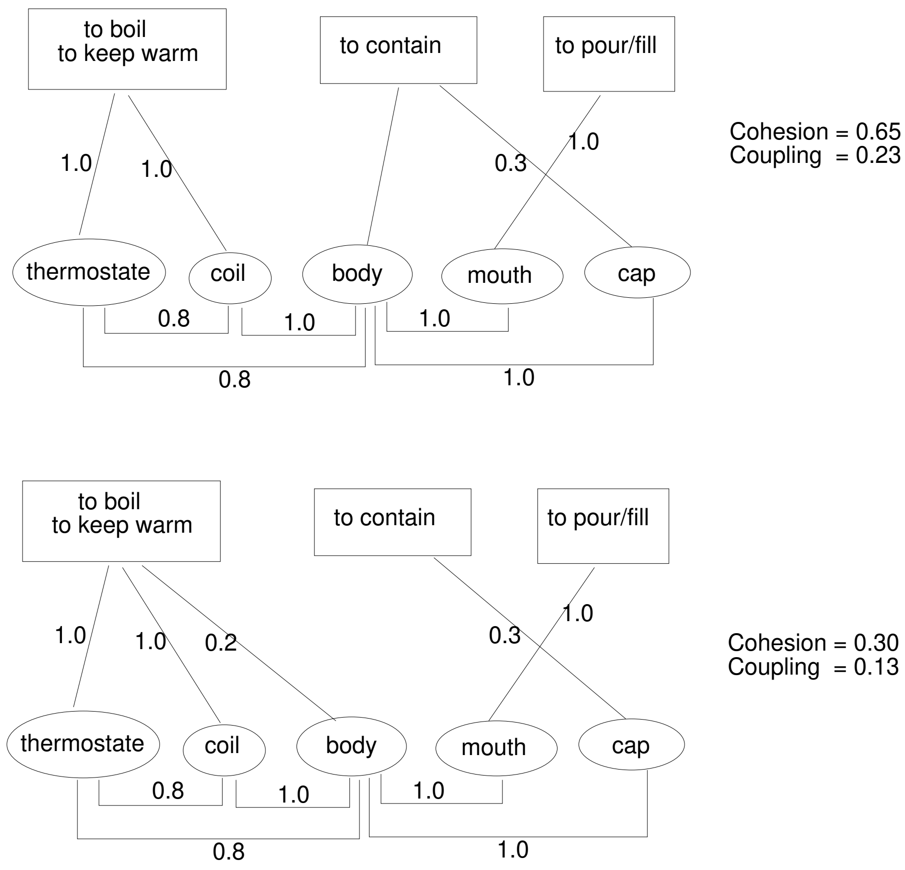


Figure 5: Some Alternate Ways to De-coupled the Design

12) set the column with the highest cohesion and lowest
13) coupling measures to 1 and set the rest to 0.
14) end

The basic idea behind this algorithm is as follows: each part/subassembly is temporary assigned to one fm based on the amount of contribution it made to the fm. This is performed in steps 2-4 in the algorithm. At the end of step 4, all the parts/subassemblies would have been assigned to at least one fm, but some parts/subassemblies may be assigned to more than one fm's. An interactive process is then initiated to de-couple those parts/subassemblies that belong to more than one fm's. This is done in step 5 through 9. The output of the algorithm is the suggested structure.
Refer to the kettle example, the $\widetilde{contribute}$ relation, $\widetilde{M}$, is

$$\widetilde{M} = \begin{pmatrix} 1.0 & 1.0 & 0.2 & 0.2 & 0 \\ 0 & 0 & 1.0 & 0.3 & 0.1 \\ 0 & 0 & 0 & 0.3 & 1.0 \end{pmatrix}$$

The α value is computed to be $min(1.0, 1.0, 1.0, 0.3, 1.0) = 0.3$.

$$\widetilde{M}_{0.3} = \begin{pmatrix} 1.0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 1.0 & 0 \\ 0 & 0 & 0 & 1.0 & 1.0 \end{pmatrix}$$

Row 2 and Row 3 both has two 1's. Hence, the cohesion and coupling measures are computed for each possible de-coupling alternatives. The final suggested structure is shown in Figure 6.

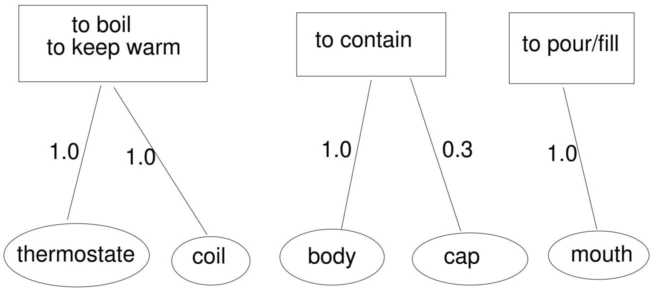


Figure 6: The Suggested Structure for the Kettle Example

To map this suggested structure to a realizable product design, case-based reasoning is employed. Case-based reasoning is an approach to solving problems in complex domains by reusing and/or adapting solutions from past experience. The main bottleneck in using the case-based approach for the design domain is the lack of a fairly comprehensive library of design knowledge. However, with the recent emphasis on collaborative design efforts [27], steps are taken to establish a common taxonomy and thereby build up resuable design databases. Such design databases are a good starting point for our case-based library. Case-based reasoning involves several basic operations: (1) the retrieval of relevant cases from a case-based library, (2) the selection of the most promising case or cases to reason with from among the collection of cases retrieved in the first step, and (3) the adaptation of a solution for the new case.

4.1 The Retrieval Phase

Retrieving cases from a case-based library is essentially a massive search problem. Associated with each case in the case-based library is a graph indicating the $\widetilde{contribute}$ relation between its intended functions and the realized parts.

Given the suggested structure, G, we wish to find a set of cases such that, together, they satisfy the $\widetilde{contribute}$ relation in G. This problem can be reduced to the set-covering problem.

Set-Covering Problem:

In the set-covering problem, the data consists of finite sets $P_1, P_2, \cdots, P_n$ and positive numbers $c_1, c_2, \cdots, c_n$. We denote $\cup(P_j : 1 \leq j \leq n)$ by I and write $I = \{1, 2, \cdots, m\}$, $J = \{1, 2, \cdots, n\}$. A subset J^* of J is called a cover if $\cup(P_j : j \in J^*) = I$; the cost of this cover is $\sum(c_j : j \in J^*)$. The problem is to find a cover of minimum cost. Mathematically, the set covering problem may be written as

$$
\begin{aligned}
& & minimize \sum_{j=1}^{n} c_j x_j \\
s.t. & & \sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad i = 1, 2, \cdots, m \\
where & & \\
x_j & = & 1 \text{ if } P_j \text{ is part of the cover} \\
& = & 0 \text{ otherwise} \\
a_{ij} & = & 1 \text{ if element } i \text{ from I is an element of } P_j \\
& = & 0 \text{ otherwise.}
\end{aligned}
$$

The reduction process is as follows: first we construct a bipartite graph. A bipartite graph is one whose vertex set can be partitioned into two subsets X and Y so that each edge has one end in X and one end in Y. For each edge in graph G, a vertex v is created and placed in the partition X. For each case in the library, a vertex w is created and placed in partition Y. An edge connects the node x in partition X to a node y in partition Y if and only if the edge associated with node x is isomorphic to an edge in the $\widetilde{contribute}$ relation of the case associate dwith node y. Let the nodes in partition Y to be the finite sets $P_1, P_2, \cdots, P_n$ in the set-covering problem. $c_1, c_2, \cdots, c_n$ in set-covering problem corresponds to the difference in the degree of contributions in the two graphs. Finally, each node in partition X corresponds to an element in the set I. With this mapping, we see that a solution to the set-covering problem implies a solution to the retrieval problem and vice versa.

Now, the set-covering problem is known to be NP-complete. A heuristics for solving the set-covering problem is used to find a near optimal solution to this problem [2]. By using this heuristic algorithm to arrive at the solution for the set-covering problem, we can solve the retrieval problem. The algorithm begins with an empty set as the initial solution and successively add to the solution the subset P_j which covers previously uncovered elements of I at a minimum average cost per new cover. The process continues until all the elements of I are covered by the solution set J^*. Figure 7 presents the algorithm.

Step 1. Initialize: The solution set is empty.
No rows are covered.
Step 2. Select an augmenting Column:
Choose the column that has the minimum
average cost per uncovered row.
Call this column j^*.
Step 3. Update the solution:
Add column j^* to the solution.
Update row coverage.
If all rows are covered, STOP.
Otherwise Go To Step 2.

Figure 7: Heuristic Algorithm for Solving Set Covering Problem

4.2 Selection of Similar Functional Requirements Cases

After the retrieval phase, the retrieved cases are checked to make sure their associated design constraints are compatible to the set of functional requirements. The checking procedure is done through a fuzzy resolution refutation system. The set of functional requirements forms the goal, $\widetilde{W}$, that we wish to prove. The set of design constraints forms the set of well-formed formulas, $\widetilde{S}$, from which we wish to prove the goal $\widetilde{W}$. First, we negate the goal, $\neg\widetilde{W}$, and add it to the set of design constraints, $\widetilde{S}$. We then use fuzzy resolution in an attempt to derive a contradiction. If a contradiction is derived, then $\widetilde{W}$ logically follows from $\widetilde{S}$ and hence the design constraints of the retrieved case is functionally compatible to the set of functional requirements.

[13] covers extensively the fundamentals of reasoning based on fuzzy production rules. He concludes that among the various interpretations of fuzzy implications, Gaines-Rescher's t-norms is suitable for the generalized modus ponens, generalized modus tollens, and hypothetical syllogisms. This is the interpretation adopted in this paper. Suppose the set of design constraints for a retrieved case is as follows:

(1) coil-diameter(0.5)
(2) coil-turns(3)
(3) coil-attachment(top)

(4) if (coil-diameter is LARGE and coil-turns is AVERAGE) then
coil-heat-capacity is GOOD
(5) if (coil-diameter is LARGE) then
rate-heating is VERY FAST.

Let domain(coil-diameter) = {0.1, 0.3, 0.5, 0.7},
domain(coil-turns) = {1,2,3,4,5}, and
domain(rate-heating) = {gradual, straight-line, exponential}.

The semantic meanings of the following linguistic terms are:

"FAST" = {(0,gradual), (0.3, straight-line), (1.0, exponential)}
"LARGE" = {(0.2, 0.3) (0.8, 0.5) (1.0, 0.7)}
"AVERAGE" = {(0, 1) (0.8, 2) (1.0, 3) (0.8, 4) (0, 5)}
"GOOD" = {(0, 10) (0.8, 20) (1.0, 30)}

Suppose the functional requirement is $\widetilde{W}$= "rate-heating(fast)."
The resolution process begins with the negation of $\widetilde{W}$. We have $\neg\widetilde{W}$ = ¬rate-heating(FAST) = {(1.0, gradual), (0.7, straight-line)}. This can be resolved with the clause: "if coil-diameter is LARGE then rate-heating is VERY FAST. " The resolution results in a new clause, $\widetilde{Q}$, where

$$\begin{aligned}
\neg\widetilde{Q} &= \neg\text{rate-heating(FAST)} \circ (\text{coil-diameter(LARGE)} \rightarrow \text{rate-heating(VERY FAST)}) \\
&= \neg\text{speed(very fast)} \circ (\{(0, (\text{straight-line}, 0.3))(1, (\text{straight-line}, 0.5)) \\
&\quad (1, (\text{straight-line}, 0.7))(0, (\text{exponential}, 0.3)) \\
&\quad (0, (\text{exponential}, 0.5))(1, (\text{exponential}, 0.7))\} \\
&= \{(0.91, 0.5)\}
\end{aligned}$$

Next, $\neg\widetilde{Q}$ is resolved with the fact: coil-diameter(0.5) to arrive at a contradiction. At the conclusion of the resolution process, a set of functionally compatible cases are retrieved. These cases are then subjected to the adaptation phase.

4.2.1 The Adaptation Phase

During this phase, the functional compatible cases must be carefully adapted to satisfy the current design problem. Two adaptations are necessary: size adaptation and interface adaptation. Size adaptation refers to the scaling of the case solutions to fit the current design. Interface adaptation refers to resolving the mismatch between the new interface constraints and the old interface constraints.

Typically size adaptation is a simple instantiation procedure. The variables in the chosen case are instantiated with the actual values of the current design. On the other hand, interface adaptation is more complicated. We have identified three types of interface mismatch:

A straightforward mismatch is one where the interfaces have the same geometric shapes and mating types. The only difference may be in the values of some parameters as shown in Figure 8. To resolve the mismatch, we simply change the parameter

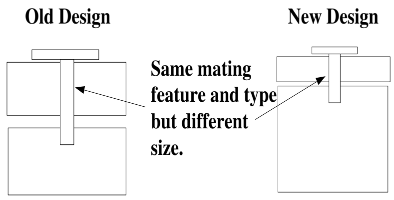


Figure 8: StraightForward Mismatch.

values of the new case to those of the old design and check to make sure that the change will not result in a violation of functional constraints.

A second type of mismatch is what we call over-constrained mismatch. Here, the mating shapes and mating types still remain the same with the exception that the new design has different number of mating constraints. An example of over-constrained mismatch is shown in Figure 9. Possible resolution methods include

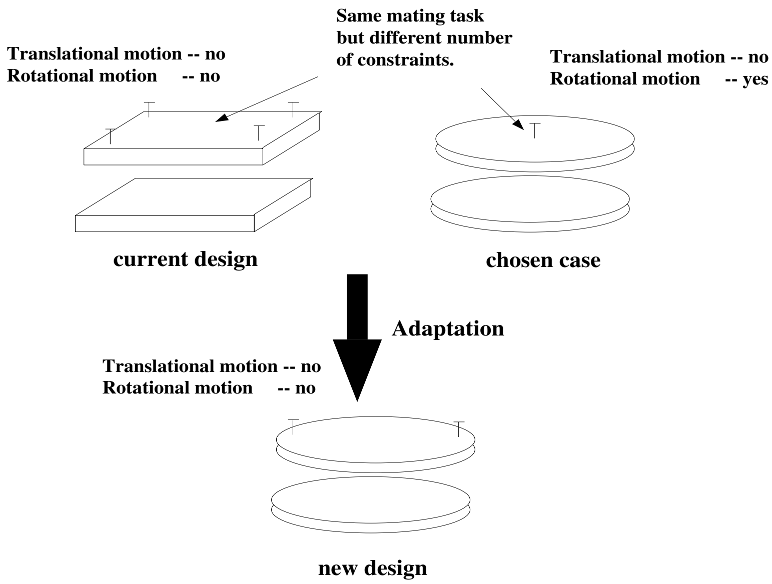


Figure 9: Over-Constrained Mismatch.

seeking the designer's help, modifying the mated component to accommodate for the different number of mating constraints using simple heuristic rules, or searching through the case-based library to modify the mated component.

Finally, we have the total mismatch category. In this category, differences are so great that it is difficult to use a set of rigid rules to reconcile the differences. The designer's assistance is often required in such case. Once the mismatch has been resolved, we can adopt the redesign suggestion. The resultant design is shown to the designer who can then make additional modifications or suggestions if necessary.

5 The Prototype System

A prototype system has been developed on a SunSparc station with graphic simulation on a IRIS Indy workstation. The program is written in C++.

Input to the system consists of the CAD description of a given design, the functional constraints, the *contribute* and *strength* relations, the list of evaluation parameters, and the list of functional requirements of the design. The output from the system is the generated redesign suggestions in the form of graphic display.

The prototype system works in two modes: an analysis mode and a redesign mode. During the analysis mode, the cohesion and coupling measures are calculated. Based on the calculation results, the designer may wish to invoke the redesign mode.

In the redesign mode, the suggested structure is displayed to the designer. If the designer agrees with the suggested structure, a case-based reasoning process is then invoked to find feasible redesign solutions that partially matches the suggested structure.

We tested the system on the kettle example. Figure 10 shows a sample of all cases retrieved from a case-based library that matches the suggested structure. The new redesign solutions exhibit better modularity while achieving the same functionalities.

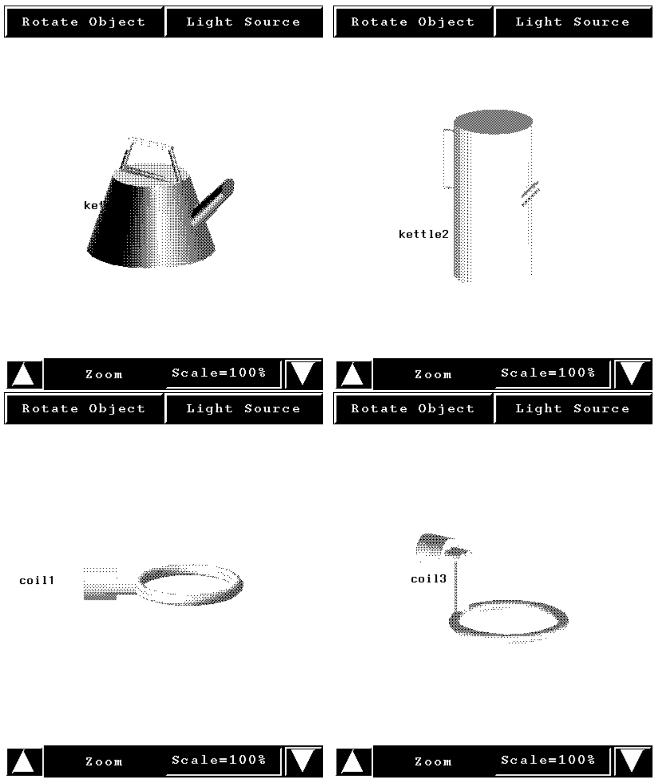


Figure 10: Sample of the Cases Retrieved

6 Related Work

Many researchers have worked on the problems relating to design. A number of design approaches have been proposed to give better understanding of the design process itself. Design synthetic reasoning [12] has been proposed by Kannapan and Marshek to derive the structure of a product from functional specifications. Their interest is in deriving a conceptual design from specifications. On the other hand, Jack Mostow [18] 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 [4] 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 [14], an interesting view was presented on the relationship among functions, structures and constraints in a conceptual design. Watton and Rinderle [28] suggested an interesting approach to enhance a given design. Their approach is to find alternative parameters to give a more direct correspondence to functional behavior.

In the 1980s, researchers also began to look into the effects of design on later life cycle such as assembly cycle or manufacturing cycle. As a result, a wealth of research has been developed on design for assembly [6, 1, 17, 10, 5], design for manufacturing [8, 22] and concurrent engineering [19, 23, 7, 25]. Recently, design for disassembly approach [20] 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. A feedback for assembly approach has also been proposed to integrate the design phase and the assembly phase into one functional unit [11]. Recently, Antonsson [29, 21] introduced the use of fuzzy logic to model uncertainty and imprecision in design.

Among all the related work, Suh's [24] axiomatic approach to design is the most relevant to our work. He advocated two design axioms to guide decision-making in design. His corollary on physical integration of parts provides a starting point for our analysis of a modular design.

7 Conclusion

In this paper, we have shown the advantages of having a modular design. A two-phase approach to derive a modular design has been proposed. During the first phase, the input design is analyzed to determine its modularity. If the evaluation results indicate a need to redesign the product, the second phase is invoked to generate suitable redesign suggestions.

References

[1] M. M. Andreasen, *Design for Assembly*, Springer-Verlag, 1988.

[2] E. K. Baker, "Efficient heuristic algorithms for the weighted set covering problem," *Comput. and Operations Research*, 8(4):303–310, 1981.

[3] J. C. Bezdek and S. K. Pal, *Fuzzy Models for Pattern Recognition: Methods that Search for Patterns in Data*, IEEE Press, 1992.

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

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

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

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

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

[9] S. F. Guggenheim, editor, *Designers in Action 1985-1991*, ASME, 1992.

[10] 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.

[11] W. Hsu, C. S. G. Lee, and S. F. Su, "Feedback approach to design for assembly by evaluating assembly plan," *Computer-Aided Design*, 25(7):395–410, July 1993.

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

[13] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice-Hall, 1995.

[14] 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.

[15] Y. Leung, *Spatial Analysis and Planning Under Imprecision*, North-Holland, 1988.

[16] B. Lotter, *Manufacturing Assembly Handbook*, Butterworths, 1986.

[17] 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.

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

[19] 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., 1989.

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

[21] K. N. Otto and E. K. Antonsson, "Tuning parameters in engineering design," *Transactions of the ASME*, 115:14–19, 1993.

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

[23] 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.

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

[25] S. N. Talukdar and S. J. Fenves, "Towards a framework for concurrent design," *Concurrent Product and Process Design*, pages 35–40, Dec 1989.

[26] S. Tamura, S. Higuchi, and K. Tanaka, "Pattern classification based on fuzzy relations," *IEEE Trans. on Systems, Man, and Cybernetics*, 1(1):61–66, 1971.

[27] G. Toye, M. R. Cutkosky, L. J. Leifer, J. M. Tenenbaum, and J. Glicksman, Share: A methodology and environment for collaborative product development, Technical Report CDR-TR 19930507, Center for Design Research, Stanford University, 1993.

[28] 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.

[29] K. L. Wood and E. K. Antonsson, "Modeling imprecision and uncertainty in preliminary engineering design," *Mech. Mach. Theory*, 25(3):305–324, 1990.