Design for Serviceability Expert System

Christopher Bryan, Graduate Research Associate
Charles Eubanks, Graduate Research Associate
Kosuke Ishii, Assistant Professor
Department of Mechanical Engineering
The Ohio State University
206 W 18th Ave.
Columbus, Ohio

ABSTRACT

This paper describes a methodology and tool which assist deployment of serviceability in the early stages of life-cycle design. Unlike design for assembly, producability, etc., design for serviceability (DFS) commonly occurs in the later stages of the design process. By this time, any design changes required to enhance serviceability are either costly or infeasible. We have developed a graphics-based computer tool to be used early in the design phase that employs the concept of service mode analysis (SMA), coupled with a service-based design description, to assess the impact of component relationships on life-cycle service costs. We also employ design compatibility analysis (DCA) to assess qualitative aspects of the design for serviceability concerns and provide the user with comments and suggestions for design improvements. Significant reductions in life-cycle costs and significant improvements in customer satisfaction can be achieved by including DFS in the design trade-off analysis process.

1. INTRODUCTION

Life-cycle design is a methodology of considering various values and cost concerns that affect the product throughout its life (Barkan, 1988). These values include not only functional requirements but also those related to producability, assembly, testability, serviceability, transportability and disposability. Of the various concerns, perhaps the most mature methodology is Design for Assembly (DFA). Boothroyd and Dewhurst (1983) and many others have proven that DFA can bring significant cost savings in production.

Another area of life-cycle engineering that has not been rigorously addressed to date is serviceability. To enhance overall competitiveness and customer satisfaction, designers must also address the reliability and serviceability of the candidate designs at the layout stage. A design without thorough consideration for serviceability could lead to an unexpected increase in servicing and warranty cost. A significant downtime of the product could also hurt the customers significantly. In short, serviceability is a key attribute that links product functions and customers.

The DFA methodology covers the serviceability question to a certain extent. In fact, disassembly is a major part of most service operations. However, serviceability concerns are broader than those of DFA. A product goes through production assembly one in its life-cycle under a controlled environment, namely the assembly plan. Here, assembly procedure may adopt specialized tools and require trained personnel. Service operations, on the other hand, may take various forms. Different service needs may arise at different frequencies. Each service operation may require different tools and assume various levels of expertise. Hence, there is a need to extend the life-cycle engineering activities to include a broader understanding of serviceability and reliability in designs and to balance these concerns with producability and assembly.

Most companies have documented guidelines for serviceability design in one form or another. The guidelines address, for various service modes, 1) provisions to detect servicing needs, 2) design features to enhance the ease of servicing, and 3) estimated life-cycle service cost (service cost * service frequency). However, the lack of a systematic means to apply these valuable guidelines often leaves them unutilized. What we need is a methodology that effectively identifies the life-cycle cost drivers and deploys the serviceability guidelines at the early stages of design. This may require a computer program that assists designers to check the serviceability of their proposed design. Serviceability design, together with DFA, should enhance the life-cycle quality of the product by increasing availability, reducing service cost, and ensuring performance. The proposed method will lead to Rapid Product Realization by facilitating advanced process planning and accelerated testing as well as avoiding costly redesigns.

Many researchers have studied the effect of maintainability on systems design (Goldman and Slattery, 1964; Blanchard and Lowery, 1969). In many cases, people try to equate maintainability with serviceability. This tends to work in military applications and similar industries. In these industries, the major mode of service is periodic inspection. The inspection involves a series of maintenance checks. If a system does not pass one of these checks, the necessary service is performed. The goal in these industries is to use maintenance to create systems that never experience catastrophic malfunctions. Therefore, emphasis is on the prediction of 1) the mean time between failures (MTBF) and 2) the necessary mean time between maintenance of the product’s various sub-systems. This approach does lead to reduced service time and cost, but this work does not readily carry over to all product design. Most products do not have the advantage of having maintenance people nearby and owners who are willing to repair the product at almost any cost.

Many industries such as power plant construction and software design, two industries where much of the life-cycle cost occurs after production, currently consider maintainability at the detailed design stage (Houser, 1989; Keys, 1990). The consideration of diagnostics in electronic circuit design has been thoroughly researched (Murakami and Nakajima, 1988) but it has not been applied extensively to mechanical design.

Makino, et al, (1989) reports on some good initial work in the creation of serviceability design expert systems. Their research seeks to “... integrate the cumulative experience of field service into quality designs in a timely and cumulative manner.” Their work defines which elements should be included in a design for serviceability (DFS) expert system and what the economic benefits of such a system may be. However, their work does not predict the degree with which various elements of the design impact the life-cycle serviceability cost. Rather, they have created a program that checks a design for serviceability flaws.

Historically, the heavy machinery and farm equipment industries have poineered incorporating serviceability into the conceptual design stages (Parks, 1986; Barquist and Malcolm, 1989). These industries focus heavily on decreasing service labor time, utilizing only standard tools, as well as simplifying the many preventive maintenance procedures incurred during the machine's lifetime. Since reducing down-time is critical in both industries, improving the maintenance and serviceability features of these products merely fulfills customer requirements.

This paper describes our effort to develop an interactive computer program that assists engineers to incorporate life-cycle serviceability into product designs at the early stages of product development. Our initial work has focused on automotive hardware (Gershenson and

Ishii, 1991), although recent projects also address other type of systems such as appliances and industrial plants. This paper concentrates on the computer software: its audience, purpose, detailed functions, and usage.

2. SYSTEM REQUIREMENTS

2.1 Audience of the system

Developing a tool to aid the design process requires the same attentions as developing a product for the general public. The first phases of development focus on fulfilling customer requirement and needs, which demands knowledge of the customer. For a system that assesses the serviceability of candidate designs, engineers constitute the bulk of the customer base. In particular, we target this methodology at three specific groups of engineers.

1) *Program Engineer*, who has knowledge of all systems involved in the complete design and has the power to manipulate physical locations of the system components.

2) *Service Engineer*, who is proficient in serviceability issues and currently assesses the serviceability of designs.

3) *Novice Engineer*, who can gain knowledge of systems and serviceability by using the serviceability system.

2.2 Benefits to the users

The program engineer will have a method of keeping track of serviceability improvements in the design, which has been difficult in the past. They will obtain some "leveraging" power with management in order to redesign systems with high warranty costs or systems not fulfilling customer satisfaction requirements. These leverage items are included in the output of our design for serviceability software.

The service engineer can benefit from this tool by reducing the time required to analyze serviceability of a candidate design . With more complex designs and greater design costs, the service engineer is in great demand throughout many industries. Any assistance to the service engineer will ultimately improve the final product.

The novice engineer will have a learning tool, which can teach the basics of serviceability as well as trade-off issues for designing a system.

2.3 When to use this tool

Typically, assessments such as DFA and QFD occur very early in the design process. Serviceability assessment should also fall somewhere in this time frame. In order to have a positive effect on the product, there must still be some latitude with the configuration of systems and components. The conceptual stage lends itself well to the use of a DFS tool. At this stage, functions are known (or surmised) yet design freedom still exists.

3. THEORY AND METHODOLOGY

3.1 Fundamentals of designing for serviceability

Definitions of serviceability differ from person to person, yet several fundamental concepts run common. The foremost of which includes the speed at which a problem can be repaired, which usually arises from ergonomic considerations; another may include the cost of repairing a problem. Our concept of serviceability seems much more broad. It not only addresses speed and cost, but incorporates the issues of diagnosability and maintainability.

In short, the three subcomponents of serviceability we currently address include the following.

1) *Corrective maintenance* primarily involves time and cost, which are related. Costs can come from labor or parts, of which labor costs are a function of the repair time. Costs can be directed to the customer or the manufacturer.

2) *Preventive maintenance* does not involve a malfunction, but addresses routine maintenance. The primary issue with preventive maintenance is accessibility. Good accessibility results in lower labor times and lower costs.

3) *Diagnosability* involves locating a malfunction. The heightened awareness of diagnosability arose from the electronics industries, where the extreme complexity and cost of products required quick and reliable fixes. These traits also emanate from mechanical systems.

Our methodology revolves around Service Mode Analysis (SMA). Service modes are the ways in which a system may be serviced. Service mode analysis can be viewed as the method of describing which service modes will impact a particular design and in what manner.

Our computer implementations of SMA employ a phenomena-based approach. Service mode phenomena are the malfunctions of a system through the customer's eyes. This presents a challenge, because the manufacturer must now acquire data concerning the customers' experience. Many industries, such as automotive, currently obtain some of this information, yet many gaps still exist. But, this method can reflect nearly 80% of the life-cycle service cost.

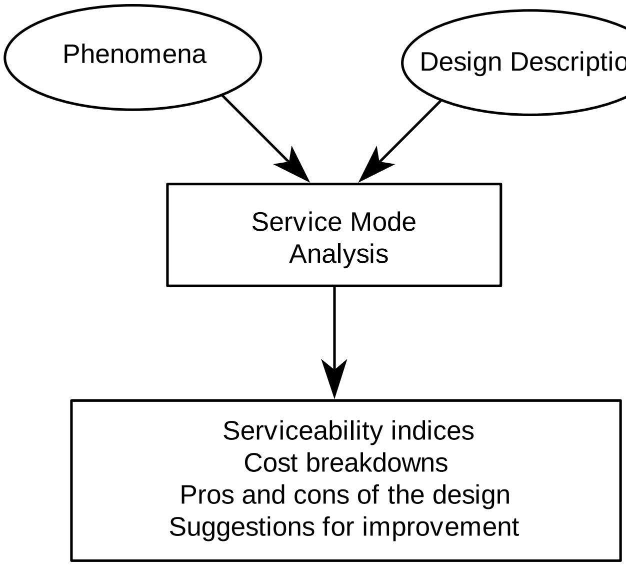

Figure 1: The Flow of Service Mode Analysis

Labor step cost is calculated based on labor time, part cost, and the tools and level of technician training required to perform the step. The labor step cost (LSC) equation takes the form: (See Section 4.3 for cost calculation details)

$$\mathrm{LSC} = \{[(t_{\mathbf{L}} + p_{\mathbf{L}}) \times c_{\mathbf{LR}}] + [c_{\mathbf{P}} + p_{\mathbf{P}}]\} \qquad (1)$$

where:

t_L = labor time (hours)
p_L = labor time penalty (hours)
c_{LR} = labor rate (\$/hour)
c_P = part or material cost (\$)
p_P = part or material cost penalty (\$)

Repair operation cost is the sum of the labor step costs times the frequency of occurrence. For repair operation 'j' associated with service mode phenomenon 'k', the life-cycle repair operation cost (LCROC) equation takes the form:

$$\mathrm{LCROC}_{j,k} = f_{\mathbf{R}_{j,k}} \sum_{i=1}^{n} (\mathrm{LSC}_i) \qquad (2)$$

where:

f_R = repair operation frequency

Thus, the life-cycle service cost (LCSC) equation for phenomena-based SMA takes the form:

$$LCSC = \sum_{k=1}^{n} \left[\sum_{j=1}^{m} \left(LCROC_{j,k} \right) \right] \qquad (3)$$

The phenomema-based approach is particularly suitable for evolutionary designs, because the set of required functions remain relatively similar. For brand new designs, there will be a need for combining component reliability into SMA.

3.2 Costs

The cost of correcting a phenomenon is simply the sum of the service costs of individual components involved in fixing that particular phenomenon. The definition of cost, however varies greatly between people.

Costs can be several measures.

1) Dollar amounts
    - Incurred by the customer.
    - Incurred by the manufacturer through warranty work.
    - Incurred by the service technician not passed to the customer.
2) Non-monetary amounts reflecting items such as
    - Potential danger to the service technician
    - Potential to damage other parts or systems
    - Potential to incorrectly or incompletely repair the malfunction
    - The excess use of special tools or shop equipment
    - Low customer satisfaction

3.3 Frequencies

Reliability can be the deciding factor of whether a product succeeds or fails. Historical failure frequencies reflect the reliability of previous designs. Unfortunately, the failure frequency of a component or system can be somewhat more difficult to obtain than a cost. With more attention being given to quality assurance, frequency data has become more readily available, but the quality of the data may still be questionable. This methodology reduces these problems by taking a systems approach to serviceability, which requires less frequency data and the use of frequency ranges instead of real numbers.

Because customers experience the failure of a *system*, and not a particular component, this phenomena-based method inherently focuses on systems. The systems-approach saves considerable time for the engineer and is more likely to make the engineer rethink the system function, which usually deepens the understanding of the root causes of malfunctions, and results in a more robust design.

3.4 Reliability vs serviceability

When designing a system, the reliability of the system usually takes precedence over system serviceability. A perfectly reliable system will not need service. Unfortunately, all systems experience failures sometime throughout their life-cycle. This exemplifies a trade-off to be made during design phases. Should the company invest in designing a perfectly reliable system, which may never be feasible because of cost, or should the company invest in reducing the annoyance to the customer by designing serviceability into the product, and, in turn, save the company untold amounts of warranty dollars. All products require a certain level of reliability, but how and where does one draw the line.

Our methodology takes advantage of the relationship between reliability and serviceability and can offer ratings which help designers realize when designing for reliability is costing too much.

3.5 The serviceability index

In determining what a serviceability index should signify, one question arises. Who will use the index? Sometimes more than one group will use the index. For example, the designer may use the index through several conceptual iterations to improve the serviceability of the product, but, when finished, the index provides a bottom-line cost to "sell" the concept to upper management (who allocate design funds). Here, we can see the possible need for several indices, one may simply reflect the dollar savings from reduced labor time and part costs. Another may reflect issues which assist the designer in keeping function foremost, such as the non-monetary costs previously mentioned. Yet another may combine dollars with non-monetary costs and assist the designer in optimizing both. The choice of index scheme depends upon who uses the index.

3.6 Technological challenges of a DFS tool

Several problems arise when investigating serviceability. As previously discussed, the lack of quality frequency data inhibits the attempt to use exact numbers for repair frequencies. By the use of frequency bands or ranges, SMA desensitizes the inaccuracies of incorrect data.

Another challenge includes the incorporation of completely new parts into a serviceability analysis. With no history, only estimates can provide the costs, failure frequencies, and other data required to include the new component in the serviceability analysis. Financial planning departments usually require cost estimates of components early in the design stages, but frequencies can be more difficult to acquire.

Implementation of an AI based tool for serviceability assessment provides a proving ground for the methodology previously developed at The Ohio State University. (Gershenson and Ishii, 1991)

4. THE ROLE OF AI IN OUR SYSTEM

The program use AI techniques in two of its modules:

1) Labor operation inferencing
2) Design compatibility analysis

4.1. Labor operation inferencing

Labor operation inferencing attempts to deduce the required actions in order to affect a manner of repair in accordance with the design description. Recall that a manner of repair consists of two pieces of information: the repair action and the component requiring the repair. Also recall that the links used in the description are directional. So, if we are interested in a particular component, a link with the component in the first position can be thought of as "outgoing". In the same way, a link with the component in the second position can be thought of as "incoming".

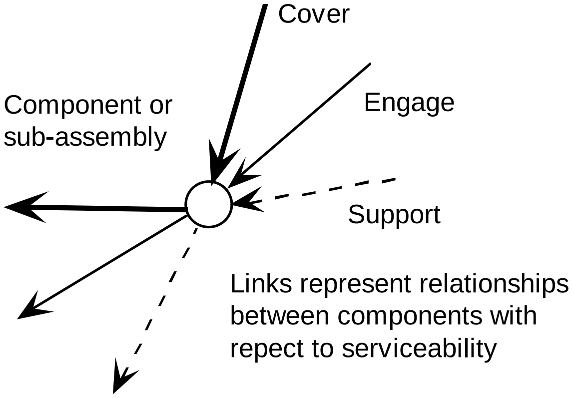

Figure 2: Serviceability Links

Starting with the manner of repair component, the program examines all links associated with it. Depending on whether the link in outgoing or incoming for a given relationship, the program will either 1) generate and store a disassembly labor operation, 2) place the other component in the link on a stack for later processing, 3) do both 1 and 2, or 4) do nothing. Once a link has been examined, its ID is placed on a stack. This stack is checked each time the program prepares to examine another link to ensure that links are not processed twice. When all links associated with the component have been examined, the program processes any components that may have been placed on the stack. If the stack is empty, the program checks the last component processed for the existence of a next higher assembly. Recall that the next higher assembly is a subassembly, so that if it exists, the subassembly will be processed as the next component. We can liken this to moving up one level in the component hierarchy. If no higher assembly exists, the inferencing process terminates: we have reached the highest level in the hierarchy.

The operations created by the inferencing process contain the disassembly action required, the component involved in the action, and the link ID that generated the action. From the disassembly action, the program generates the required assembly action and adds it to the labor operations list. Using the link ID, the program searches for any sublinks associated with it and computes any service costs associated with it; i.e. fastener removal and insertion costs. Similar cost data for the component involved is added to the service cost for the operation.

4.2 <u>Design compatibility analysis</u>

The design compatibility analysis attempts to match elements of the design description with the qualitative design guidelines contained in the knowledge base. The program binds appropriate values contained in the component, link and sublink data bases to test the rules in the knowledge base. When matches occur, the rule ID, descriptor, comment and suggestion along with the design characteristics which caused the rule to be true are stored. The descriptors are mapped to normalized values from 0 to 1 and combined to yield an overall compatibility rating. (see Ishii, K.,Life-cycle Engineering for Design, Quality Concepts 1990) The rule ID can be matched to a graphical representation of the comment and suggestion, providing the user with possible redesign alternatives, if the rule was the result of some incompatibility.

5. INITIAL SOFTWARE DESIGN

Our initial software design provides the user with a self-contained, graphics-based serviceability assessment tool. The program integrates a functional design description, historical service mode phenomena information, and serviceability knowledge to 1) infer life-cycle service requirements, 2)compute and rank their associated costs, 3) indicate serviceability concerns and 4) suggest possible design improvements.

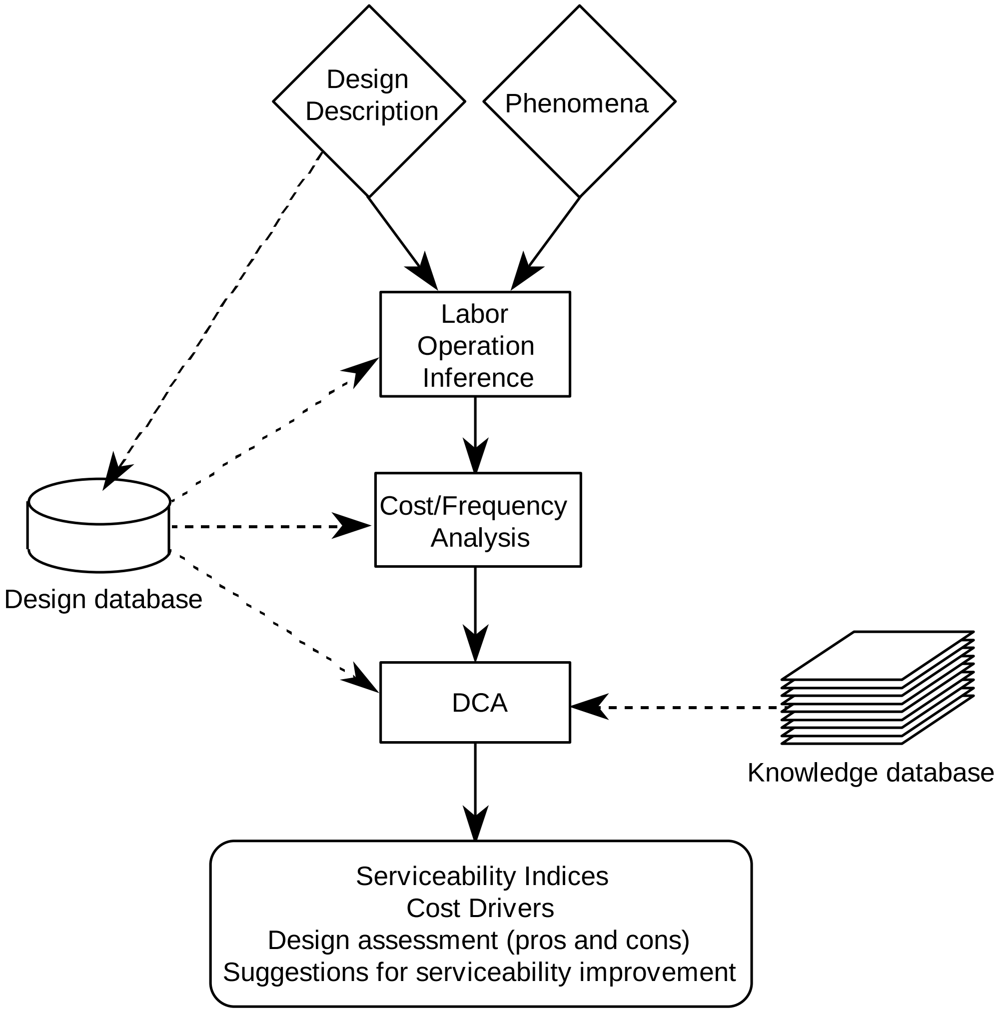

Figure 3: Serviceability Program Flow

The Alpha version of our DFS software is implemented in "ToolBook" from Asymmetrix Corp. ToolBook is a graphics based software construction set used to develop Microsoft "Windows" applications on IBM PC compatible computing platforms.

The user provides four primary inputs:

1. Design description
2. Icon/Link data
3. Malfunction/Repair data
4. Design guidelines knowledge base

5.1 **Design description**

When a system is serviced, it is quite common to remove parts which have nothing to do with the faulty component, but are simply "in the way". An access panel attached to a housing, for example, shares an assembly relationship with the housing. It also shares a service relationship with components inside of the housing, since it must be removed to access them. Our design description, called the linker, attempts to account for both types of relationships.

The user lays out the design in a top-down fashion, using icons representing the subassemblies, components and processes that comprise the system, as well as links and sublinks (lines) representing their relationships with each other. The icons and links each have an internal identification (ID), set by the program, and a name, set by the user.

<<<<< linker screen dump>>>>>

Figure 4: The Linker Screen

Icons represent the discrete items or processes that make up the system. Subassembly icons have an additional feature. They are used to create new layout screens on which the subassembly's components can be represented, creating a tree-like structure. When an icon is created,the program sets up a record in the data base with the item ID and the item's next higher assembly, if one exists. This record also stores any data for this item that is entered by the user via special data entry screens.

There are two types of component relationships, links and sublinks, which are used to form the semantic network for the design.

Links (arcs) establish a directional relationship between two icons (the nodes). The relationship names for the current implementation are "covers", "attaches to", "connects to", "engages" and "supports". Links between components and subassemblies are constructed in the same order as their semantic representation; i.e. "component 1 attaches to component 2". The directional nature of the links becomes particularly obvious when we consider the covers relation; where, "outer panel covers internal subassembly" makes service-wise sense, "internal subassembly covers outer panel" does not. When a link is created, a data record with the link ID and the two icons, in order of linking, is established. This record will also be used to store the relationship name.

If a fastener or fastening process is required to maintain the link, a sublink is required. Sublinks (also arcs) establish a relationship between an icon, in most cases a fastener or fastening process, and a link. This relationship takes the form of "using", "by", or "via". Although not explicit within the semantic network, it augments a link relation, such as "panel attaches to housing using screws". When a sublink is created, a data record with the sublink ID, the link ID and the icon ID is established. This record will also be used to store the number of fasteners or process points, clearance around the fastener of process point, tool orientation and removal and insertion direction.

5.2 Icon/link data

Certain icon and link data are entered by the user as a separate operation.

In general, icon data entered by the user will consist of part or material cost, removal time, insertion time, required tools, part or material availability, training required to perform the action, the name of the item or process, a user-defined identifier (part number or code), and the next higher assembly (if applicable). In addition, components and subassemblies are identified as modules, fasteners as reusable, and processes as reversible.

<<<<< component data screen dump >>>>>

Figure 5: Component Data Entry Screen

For links, the only data entered by the user is the relationship name. Selecting the proper linking relationship, or verb, is crutial to the ability of the program to properly infer the required labor operations during the analysis portion of the program. In our implementation, the verbs take on the following characteristics:

COVERS: No physical connection exists between the two components, but the first component in the link must be removed to access the second component.

ATTACHES TO: This represents a solid connection with no relative motion between the two components during operation. This link is broken by physically removing the first component in the link.

CONNECTS TO: This represents a solid connection with no relative motion between the two components during operation. This link can be broken without physically removing either of the two components.

ENGAGES: This represents a meshing-type connection with relative motion between the two components during operation. This link can be broken by physically removing either of the two components in the link.

SUPPORTS: This represents a solid connection with no relative motion between the two components during operation. This link is broken by either physically removing the second (supported) item in the link, or by externally supporting the second (supported) item in the link and then physically removing the first (supporting) item.

Sublink data entered by the user includes number of fasteners or process points, clearance around the fastener of process point, tool orientation and removal and insertion direction. This information is used to assess the relative difficulty of performing the required operation.

5.3 Malfunction/repair data

The malfunction and repair data entered by the user consists of the service mode phenomena and their associated manners of repair. Generally, the service mode phenomena are obtained from a companies historical service data.

<<<<< phenomenon editor screen dump >>>>>

Figure 6: Service Mode Phenomena Editor Screen

The user enters the phenomenon description and the frequency of occurrence over the life-cycle of the product, followed by one or more manner of repair associated with the phenomena.

The manner of repair describes the repair operation necessary to remedy the malfunction. The user enters the description in two parts: the action, such as "remove and replace", "repair" or "align", and the affected component. The user also enters the the life-cycle frequency of occurrence of the manner of repair for the associated service mode phenomena.

5.4 Design guidelines knowledge base

The design guidelines knowledge base forms the basis of the design compatibility analysis. The rules in the knowledge base test for a qualitative design characteristic or set of characteristics which have an effect on design serviceability, either positive or negative. The knowledge used to construct the rule base is generally found in design guidelines references and lessons learned documents. Such knowledge is also found in the minds of experienced engineers, which can be both the most valuable and the hardest to obtain.

This approach views the design guidelines knowledge as compatibility comments. (see Ishii, K.,Life-cycle Engineering for Design, Quality Concepts 1990) The rules contain a rule ID number, a compatibility descriptor such as "very good" or "poor", reasons and suggestions, and the conditions for the rule to be true. An example of a "poor" condition might be:

```
if (operation is "remove"), and
        (operation frequency > 10), and
        (fastener is "adhesive")
then (ID = dfsr0001), and
        (compatibility is "poor"), and
        (comment is "components with high removal
                frequencies should not be bonded"), and
        (suggestion is "try using a mechanical fastener").
```

While some serviceability design guidelines, like the one above, may be somewhat universal, most will likely be product specific. Currently, the user is required to enter these rules directly into the knowledge base, thus requiring some knowledge of the knowledge base programming language.

5.5 Assessing the design

The user begins the analysis by selecting service mode phenomena to be processed from a palette of those phenomena previously input.

<<<<< phenomena selection screen dump >>>>>

Figure 7: Service Mode Phenomena Selection Screen

The program now infers the required labor operations using the selected phenomena and their associated manners of repair and computes the associated costs. The user can review the labor operations list and make additions or deletions as required.

<<<<< labor operations screen dump >>>>>

Figure 8: Labor Operations Definition Screen

The program provides three basic outputs:

1) service cost breakdown
2) serviceability indexes
3) design comments/suggestions

5.6 **Serviceability cost breakdown**

Service costs are calculated for service mode phenomena, manners of repair and labor operations. Costs are currently computed using a normalized penalty point system: the more serviceable the design, the less points are accumulated.

<<<<< SMA summary screen dump >>>>>

Figure 9: Service Mode Analysis Summary Screen

The costs and frequencies are displayed on a bar chart in rank order beginning with the item incurring the highest total cost (cost * frequency). As stated earlier, frequencies for service mode phenomena and manners of repair are entered by the user. Frequencies for labor operations, however, are computed based on how many times an operation occurs in the labor operations lists of all of the manners of repair being evaluated, multiplied by the manner of repair frequency.

<<<<< labor operations summary screen dump >>>>>

Figure 10: Labor Operations Summary Screen

5.7 Serviceability indexes

The serviceability indexes give relative ratings of service cost, service frequency and serviceability guideline compliance for the described design. In each case, the design is given a basic rating between 0 and 1; 0 representing a design incurring maximum cost or penalty, and 1 representing a design incurring no cost or penalty.

The service cost index is simply based on the penalty points accumulated compared to the maximum number of points possible. The frequency rating is based on a user defined maximum acceptable frequency, and gives a relative measure of system reliability. The serviceability guidelines compliance index is computed in accordance with the DCA formulas developed by Ishii, Adler and Barkan. (see Application of Design Compatibility Analysis to Simultaneous Engineering, AI EDAM, 1988)

5.8 Design Comments/Suggestions

Design comments and suggestions are developed from serviceability design guidelines, sometimes referred to as serviceability pros and cons. They are triggered using a set of rules in a knowledge base which describe combinations, both good and bad, of characteristics related to serviceability design. The rules are compared with data from the design description. If a match occurs, the rule number and its rating are stored in a file.

<<<<< DCA summary screen dump >>>>>

Figure 11: Design Compatibility Analysis Summary Screen

The main program matches the rule with graphical displays offering advice to the user for design improvements. The graphical displays include figures and text related to the characteristic combination which caused the rule to fire, usually providing an example of a more serviceable design and an example of a not-so-serviceable design.

<<<<< con screen dump >>>>>

Figure 12: Design Comment Screen

<<<<< pro screen dump >>>>>

Figure 13: Design Suggestion Screen

6. FEEDBACK FROM DESIGNERS

We field tested the Alpha version of the software at General Motors in Warren, MI. We loaded it onto computer in four locating in engineering and service organizations. We provided the users with either group or individual training and a short set of instructions.

Feedback from the users was very positive. They felt that the analysis provided by the program would be very useful in conjunction with DFA analyses. They liked the linker as a tool for design description and found the cost breakdowns quite informative. Once they became familiar with the program's operation, they ranked the ease of use as good. Along with some user-friendliness suggestions, other recommendations were as follows:

1) *Actual dollar service cost estimates:*

Currently, the program approximates service costs by assessing penalty points for various design characteristics which adversely effect serviceability

2) *Ability to focus on manner of repair:*

The program builds the cost summaries in a hierarchical fashion, starting with service mode phenomena (customer complaint), then manners of repair, and finally individual labor operations. Users in the engineering functions stated that they were mostly interested in assessing costs for service of individual components, regardless of the phenomena that prompted the manner of repair.

3) *"Side-by-side" design comparison:*

The existing program evaluates one design at a time. Users who wish to do trade-off analysis stated that the ability to assess two designs at the same time would simplify this type of assessment.

7. CHALLENGES

7.1 Estimates for new components

Our methodology assumes that service modes, manners or repair and their associated frequencies are known, which will generally be the case for carry-over components in functionally equivalent systems. If new components are employed or new functionality is added to the system, this type of data will not be directly available.

For new components in a functionally equivalent design, reliability testing could provide the information necessary to estimate manner of repair frequencies for the phenomena associated with the function performed by the component. This data should be available at the time the DFS evaluation is performed.

New functionality added to a system presents a bigger challenge, since even the service modes associated with that function will probably not be clearly defined. In this case, service modes may have to be estimated using a functional deployment methodology. The manners of repair and their frequencies could then be estimated based on component testing data, thus allowing at least preliminary estimate of the service cost drivers and their magnitudes.

7.2 Tracking improvements

The true worth of any design aid is measured by the improvements that result from its use. For serviceability, the data normally appears in the form of pre-production service evaluations and field service reports. These data must be monitored and compared with the analysis provided by the program to ensure that improvements exist and that the equations and the knowledge used to assess serviceability design are accurate and up to data.

As with any expert system, it is the knowledge base that requires particular attention. Constant improvements in methods and technology dictate constant update and revision of the serviceability guidelines contained in the knowledge base. Some companies address this issue by assigning a knowledge base "owner", usually an experienced technical specialist, to oversee the maintenance of the knowledge base.

8. CONCLUSIONS AND FUTURE DIRECTIONS

The result of deploying serviceability early in the design phase is increased competitiveness. Service cost can account for as much as 10% of a product's purchace cost to account for anticipated warrenty costs. Identifying cost drivers and working to reduce service costs early reduces prices. Keeping service costs as low as possible over a products entire life-cycle increases customer satisfaction. This combination both attracts customers and maintains an established customer base, and that ensures a company's continued presence in the marketplace.

While an expert system provides an excellent platform for representing a computerized version of serviceability guidelines, these must somehow be input into the expert system as rules. This has historically been a difficult task in all AI application. The need arises for an interface, simple enough for a novice computer user, which does not require the user to be proficient in AI languages, such as Prolog or Lisp. We are currently developing such an interface, called the "Rule Editor", which will allow the user to input knowledge-base rules along with associated graphics describing the good and bad features of the rule as well as examples of improvements.

In many cases, much of the data required by the program, such as costs and frequencies, exist within company databases, the key word being the plural "databases". Two major advantages of having that information available to the program directly are decreases user input and the assurance that the data used by the program is up to date. Use of the Standard Query Language (SQL) may provide this capability in future versions.

Integration of this tool into a true DFX environment is always a source of speculation. We see this question hinging on the design representation: what type of description is needed to fully describe the design that is intuitive for the user and comprehensible by the various modules of the DFX system? Serviceability is closely tied to the spatial relationships between components. While these relationships can be modeled in a CAD environment, such environments must also provide the ability to establish the necessary relationships (links) for the inferencing process.

In addition, it can be seen that some of the aspects of DFA are present in the DFS description, but that the queries to determine the DFS and DFA coefficients are quite different. In some respects, the goals of the DFX modules are going to conflict, requiring either human intervention or intelligent algorithms to weigh out the various options

REFERENCES

Adler, R. and Ishii, K., (1988), DAISIE: Designers Aid for Simultaneous Engineering, *Proc. of the 1989 ASME Computers in Engineering 1989*.

Barkan, P., (1988), Simultaneous Engineering: AI Helps Balance Production and Bottom Lines, *Design News*, March 1988.

Barkan, P. , Ishii, K. et al,. (1990), Competitive Product Design for Manufacturability, To be published from McMillan Publishers.

Barquist, L. R. and Malcolm, R. L., (1989), *Serviceability: A Key Factor in Design*, Society of Automotive Engineers, Inc..

Boothroyd, G. and Dewhurst, P., (1983), Design for Assembly: a designer's handbook, Boothroyd Dewhurst Inc., Wakerfield, Rhode Island.

Dhillon, B., (1983), Reliability Engineering in Systems Design and Operation, Van Nostrand Reinhold Co., New York.

Dhillon, B., (1988), Mechanical Reliability: Theory, Models and Applications, American Institute of Aeronautics and Astronautics, Inc., Washington, D.C.

Gershensen, J. and Ishii, K., (1991), Life-Cycle Serviceability Design, *Proc. of the 1989 ASME Design Theory and Methodology 1991*, Vol. 31, pp. 127-134.

Goldman, A. and Slattery, T., (1964), Maintainability: A Major Element of System Effectiveness, John Wiley and Sons, Inc., New York.

Ishii, K. and Barkan, P., (1987), Design Compatibility Analysis--a framework for expert systems in mechanical system design, *Proc. of the 1989 ASME Computers in Engineering 1987*, Vol. one, pp. 95-102.

Ishii, K., Adler, R. and Barkan, P., (1988), Application of Design Compatibility Analysis to Simultaneous Engineering, *Journal of Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI EDAM)*, Academic Press, Vol. 2, No. 1.

Ishii, K. and Goel, A., (1989a), A model of simultaneous engineering, Artificial Intelligence in Engineering: Design (J.Gero, ed.), *Computational Mechanics Institute, Transaction of the 4th International Conference on AI Applications in Engineering*, July 1989, Cambridge UK, (AI ENG 89).

Ishii, K. and Nekkanti, R., (1989b), Compatibility representation of knowledge about design for net shape manufacturing, *Proc. of the 1989 ASME Design Theory and Methodology Conference, Sept. 1989*, Montreal, Canada.

Ishii, K., Hornberger, L., and Liou, M. (1989c) Compatibility-based Design for Injection Molding. *Proc. of the 1989 ASME Winter Annual Meeting 1989*, San Francisco, CA

Keys, L.K., (1990), System life cycle engineering and DF”X”, *IEEE Transaction on Components, Hybrids and Manufacturing Technology*, Vol. 13, No. 1, March 1990.

Levinson, C.H., (1964), “System reliability Analysis,” Mechanical Design and Systems Handbook, McGraw-Hill, New York.

Makino, A., Barkan, P, and Pfaff, R., (1989), Design for serviceability, *Proc. of the 1989 ASME Winter Annual Meeting 1989*, San Francisco, CA..

Maloney, L., Ishii, K., and Miller, R.A, (1989), Compatibility-based Selection of Forging Machines and Processes, *Proc. of the 1989 ASME Winter Annual Meeting, 1989*, San Francisco, CA.

Murakami, T. and Nakajima, N., (1988), Computer-aided design diagnosis using feature description, Artificial Intelligence in Engineering: Diagnosis and learning (J.Gero, ed.), Elseveir, New York.

Parks, Roby A., (1986), *Designing for Serviceability and Repairability-The FD30B*, Society of Automotive Engineers, Inc..

Priest, J., (1988), Engineering Design for Producibility and Reliability, Marcel Dekker, Inc., New York.

Rinderle, J.R., (1988), Implications of Function-Form-Fabrication Relations on Design Decomposition Strategies, *Proc. of the 1989 ASME Computers in Engineering* , Chicago, Ill., 1986.