A Computer-Based Design History Tool

Aihua Chen
Thomas G. Dietterich
David G. Ullman

Departments of Mechanical Engineering
and Computer Science
Oregon State University
Corvallis, OR 97331

Abstract

This paper presents an introduction to the Design History Tool. This tool is a computerized design history that captures, represents, and plays back important information generated during the process of designing a mechanical system. This tool not only represents the final design results, but also the constraints developed and decisions made in the evolution of a product from initial specifications to final detailed design. This tool allows design, management, and manufacturing personnel to query design artifacts, design evolution, design rationale and alternatives behind each decision, and constraint dependencies among design objects.*

I. Introduction

When the design of a product is completed, its final form is usually recorded as a collection of drawings with attached notes. This design information is then used to support two major activities: (a) communication (with other designers, manufacturing, management, marketing, etc.) and (b) as a data source in redesign or design of similar products. In current practice, these activities often fail, because there is not enough information in the final drawings and notes to answer all of the questions that arise. In particular, manufacturing engineers often have difficulty understanding the importance of a specific feature or parameter from the drawings. Additionally, design engineers working on similar products at a later date frequently must start over, because they can't understand the old design documents.

It has been speculated that redesign and design understanding would be significantly improved if the final design included much more information about the "history" of the design process [Brown, 89, Ullman, 87]. This history could include information about why each component was included in the design, what alternatives were considered and why they were rejected, and how each of the product requirements was satisfied by the design.

Today's CAD systems provide an excellent means of representing and communicating the final design specification, but they lack the ability to recount the process that leads to the final product and information on decisions and constraint developments. Design notebooks can be used to record some fraction of this missing information, but they are often incomplete, difficult to interpret, and private. To improve these forms of design records, Oregon State University's Design Process Research Group has been concerned with the development of a computer-based Design History Tool. This Design History Tool records, represents, and plays back the evolution of the product's design along with the constraints and decisions that led to the final configuration.

In Section II of this paper, the concept of a design history and the purposes for developing a Design History Tool are presented. This is followed in Section III by a discussion of the methodology used. In Section IV, an overview of the Design History Tool is presented. This is followed in Section V by a discussion of design history representation. The four browsing facilities provided by the design history tool for querying the design information are illustrated in Section VI. Finally in Section VII, conclusions and plans for future work are presented.

II. The Need for Design History in Mechanical Design

A design history is a means of recording, storing, and reviewing the important information generated during the process of designing a mechanical component or system. The design is recorded and represented so that, not only can the initial and final states of the design be reviewed, but also all the important intermediate states as well. The evolution of designing a mechanical system, an individual component in a large system, or a feature of a component, can be traced from the initial design specifications to its final, manufacturable form. Maintained in the knowledge base are all the major decisions and constraints developed throughout the design process. This includes information on features of the design that were rejected by the designer as well as the reasons behind their rejection. Within the knowledge base, it is possible to determine such things as decision making processes, constraint dependencies and design evolution. Thus, with the aid of a design history all important aspects of the design and its development process can be inspected either during the

* This project is funded by National Science Foundation under grant number DMC-87-12091 and gifts from the Schlumberger Laboratory for Computer Science.

Proceedings of the 1991 NSF Design and Manufacturing Conference
Jan 9-11, Austin Texas, pp 985-994

design process or after its completion.

The primary purpose of developing the Design History Tool is to aid both design understanding and communication. Most design understanding is acquired through questioning and conjecturing about information on design drawings, hardware, or the knowledge of other engineers. Since the design history provides not only the "what" of a final design, but also the "how" and "why" that were involved in reaching that design, by providing a tool that supports the direct querying of a design, comprehension can be made more expedient and more complete. Communication is also facilitated by quickly and efficiently browsing a design from any desired viewpoint. A design manager can evaluate a design by examining not only the final detailed design, but also the steps that were taken to reach it. Engineers working on related problems can examine each other's work and focus directly on only those specific aspects of the design that affected them.

Another reason for the developing Design History Tool is to support the redesign process. A design is often modified many times during its life span, and frequently the modifications are not made by the original designer. By incorporating the Design History Tool, the process of redesign can be greatly enhanced. Not only can an engineer understand how a design came into being, he can also inspect the constraint dependencies and relations. These in turn give insight on how the design will be affected by changing or modifying an existing constraint. The overall effect will be to decrease redesign time and improve the final design, since the designer will know both the source and reasoning behind all previously made design decisions.

The design history involves both symbolic and numeric information manipulation. Today's CAD systems do not support symbolic manipulation. Some intelligent expert design systems, although they provide both symbolic and numeric environments, are domain oriented and more importantly, they don't help designers to create a design. By developing the Design History Tool, the techniques for supporting a combination of symbolic and numeric environments for mechanical design are being established. In addition, significant insight into the way designs are developed and the formal representation for this development are being obtained. These techniques provide the foundation for the development of future intelligent mechanical CAD systems.

Another benefit obtained in developing the Design History Tool is that the representation developed will lead to standards for design documentation that go beyond current PDES and CALS proposals.

III. Methodology

To construct the design history tool, study began with data from two videotape protocols of professional engineers "thinking aloud" as they solved design problems. These protocols were gathered as part of an earlier study [Ullman, 88a, Stauffer, 87] that developed a cognitive model of the mechanical design process. Each protocol is approximately 6 hours long and involves hundreds of design decisions. One protocol involves the design of a battery holder for a portable personal computer. This protocol is the basis of the examples shown in this paper. The other protocol involves the design of a piece of manufacturing equipment for casting thin organic membranes onto aluminum sheets. These protocols have been extensively analyzed (e.g., [McGinnis, 89]), and one benchmark for the knowledge representation work is to be able to represent all of the design decisions that appear in these protocols. At the present time, this goal is quite close to be being achieved.

In addition to providing an adequate representation, a design history system must provide an interface that makes it easy to answer the questions that designers typically have. To find out what kinds of questions are typically asked, an experiments was conducted in which professional engineers were given the initial specifications and final drawings for one of the two protocol problems mentioned above and asked to solve a series of redesign problems while thinking aloud [Kuffner, 89, 90]. They were encouraged to ask questions of a second engineer who was present and who was thoroughly familiar with the previous design. Each question or conjecture articulated by the designer was recorded on videotape, analyzed, and catalogued. An important benchmark for the playback interface work is to be able to easily and intuitively answer each of these questions and support or refute conjectures.

A final benchmark will be provided by having a third group of engineers use the design history system to carry out design comprehension and redesign tasks. By comparing the data from these subjects with a matched group that is performing the same tasks without the computer-based design history system, the strengths and weaknesses of the current system will be determined.

IV. Overview of Design History Tool

The Design History Tool is a computerized system that captures, represents, and plays back design history information. It is implemented in HyperClass** [Smith, 88, 89, Schoen, 89], an object-oriented programming environment that was developed by Schlumberger Technologies. In order to display design artifacts graphically, a solid modelling package, Vantage*** [Balakumar, 88], is integrated into HyperClass [Charon, 89]. The Design History Tool can interact with any design history data base. A total of 50,000 lines of lisp code about 3000 lisp procedures have been developed to support the tool. The tool consists of three sub-systems: design capture, design history knowledge representation, and design playback (Figure 1).

Figure 1
Design History Tool Sub-Systems

The design capture system is intended for entering design information into the design history during a design process. Currently, a basic design entry system has been developed. This system allows a user to efficiently enter design information taken from protocol data into the Design History

** HyperClass is a trademark of Schlumberger Technologies Incorporated.

*** Vantage is a trademark of Carnegie-Mellon University.

Tool. However, it is not suitable for a designer to use during the design process. An important goal of future research is to develop more natural and less obtrusive design capture techniques that can be used by the designer.

The design history knowledge representation provides a semi-formal language for describing the design process and the designed artifact. It also provides a fairly complete set of facilities for capturing design decisions and design constraints. Finally, the design playback system allows designers, manufacturing engineers, and management personnel to review, examine, and modify an existing design. It provides the facilities for the user to browse and retrieve the design information represented in the design history. In this paper, the design history representation and playback sub-systems will be discussed.

V. The Design History Representation

The design history representation was developed based on previous research efforts by [McGinnis, 89], [Ullman, 88a, 88b], [Kuffner, 89, 90], and [Tikkerpuu, 88, 89]. Generally, the representation is based on the following. At any time in the design process, the state of the design is given by the current configuration of the design objects and the constraints affecting them. Initially the state consists of the original, given constraints, which are usually abstract functional requirements along with specific spatial or geometric limitations. By the end of the design process, the state of the design is partially represented as refined objects with fully-defined topology and geometry. The state of the design refers to the physical characteristics of the design at any given point in time throughout the design process.

During the evolution of the design, many decisions are made (about one per minute, see [Stauffer, 87]). Each decision involves reasoning about one or more existing input constraints to produce new derived constraints. These derived constraints subsequently affect some features of a design object or some relationship between design objects. In this way, the design objects, along with their respective constraints, define the state of the design. The changes from one design state to another occur via the design decisions. By chronicling these design decisions and constraints, the process of the design can be recorded, and both the states of the design and the design process can be accounted for.

Thus, the design history representation is comprised of three main parts: design objects, constraints, and design decisions. These three elements interact together as shown in Figure 2. Each of these three is detailed below.

V.1. Design Objects

The design objects are the physical artifacts of the design. They represent the components of the design as well as the assemblies created with those components. Included in design objects are the features of the design. Features, in this research, are defined as any particular or specific characteristic of a design object that contains or relates information about that object. The features usually consist of physical parameters such as length, width, mass, location, etc., functional descriptions such as purpose or behavior, and composite features that construct the components of the design.

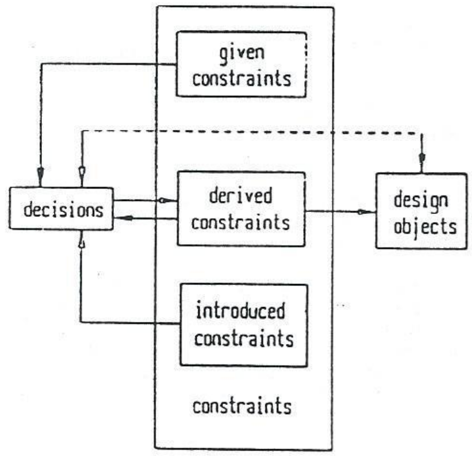

Figure 2
Design History Representation

V.2. Constraints

The most fundamental pieces of the design state are the design constraints. Each constraint specifies the value for some feature of some design object or describes some relationship between features of design objects. Hence, while the design objects provide the "vocabulary" of the design state, it is the design constraints that provide the "sentences" that actually describe the state. For example, to record the fact that the diameter of a battery is 1.3 inches, it is necessary to have a design object called a "battery" with a feature called "diameter". Then, one can write a constraint indicating that "battery.diameter = 1.3 inches".

In considering constraints, it is important to identify the source of the constraint. There are three sources for constraints: given, derived, and introduced. Given constraints are those dictated to the designer from external sources such as design specifications, adjacent or connecting designs, or clients. Given constraints define the beginning state of the design, the initial specifications. Derived constraints are generated inside the design space during the design process. They are intrinsic to the design being worked on. Introduced constraints are those that are brought in from outside the design space. They are not derived from any other constraints. Introduced constraints typically include designer's domain knowledge, handbooks, and other "domain knowledge" sources.

V.3. Decisions

Design decisions are the processes by which new derived constraints are created to change the design state. A decision is made by considering some previously existing constraints. From these, the decision produces one or more new derived constraints. The constraints that are input into the decision can be given, derived, or introduced. The decisions are based on the results of calculations, selection, simulation, analysis, and tests. The resulting constraints are always some new, derived constraints that change the state of the design in some way. Thus the design decisions constitute the smallest unit of the temporal design history. The design history is an ordered sequence of the design decisions.

VI. Design Playback

The design playback system provides the facilities for the user to browse the design information represented in the design history. In this paper, four main browsing capabilities will be illustrated, they are:

1. browsing design artifacts,
2. browsing design evolution,
3. browsing design rationale and alternatives, and
4. browsing constraint dependencies.

To allow users to efficiently retrieve and easily understand the design information, the playback system supports different schemes for displaying and retrieving design information. The following will briefly describe these browsing facilities.

VI.1. Design Artifact Browsing

The design artifact browsing begins with two top-level displays. The first allows users to browse the hierarchical structure, as shown in Figure 3. In this graph, it can be seen that the battery-case consists of two components: bottom-case and cover. The bottom-case is composed of several features, such as the left-side-wall, isolating-wall, and back-wall. The second top-level display is an isometric drawing of the entire design. The two top-level displays are synchronized, so that if an object is selected in the hierarchy via the mouse, the corresponding object will appear as an isometric drawing. Figure 3 and Figure 4 present an example of this facility. From the isometric drawing in Figure 4, the design object parameters or features can be inspected by selecting the command "Design object features" in the menu attached to the isometric drawing. This selection will open a design object feature interface. Figure 5 presents an example of the bottom-case design object feature interface. In the bottom box on the interface, there are two columns. One shows

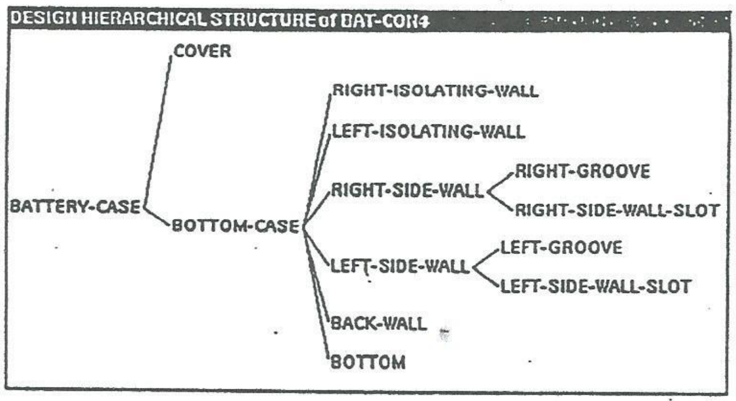

Figure 3
Hierarchical Decomposition Structure

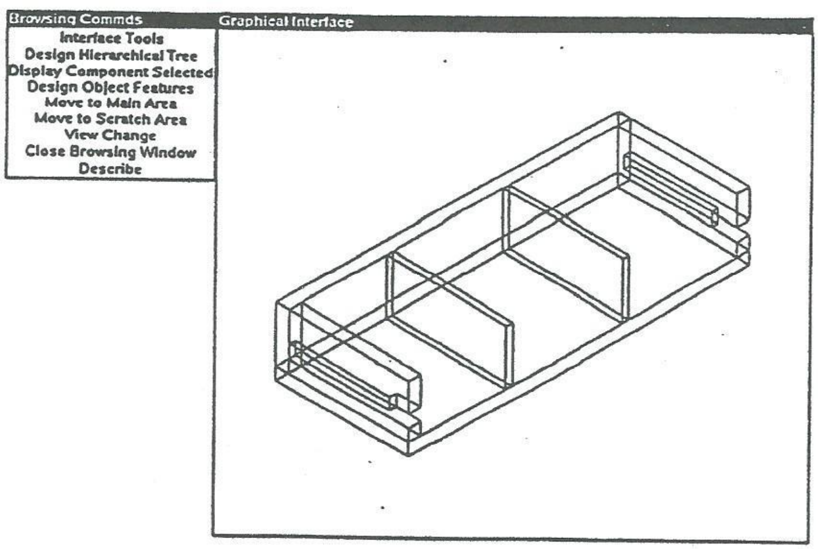

Figure 4
Design Object Isometric Drawing

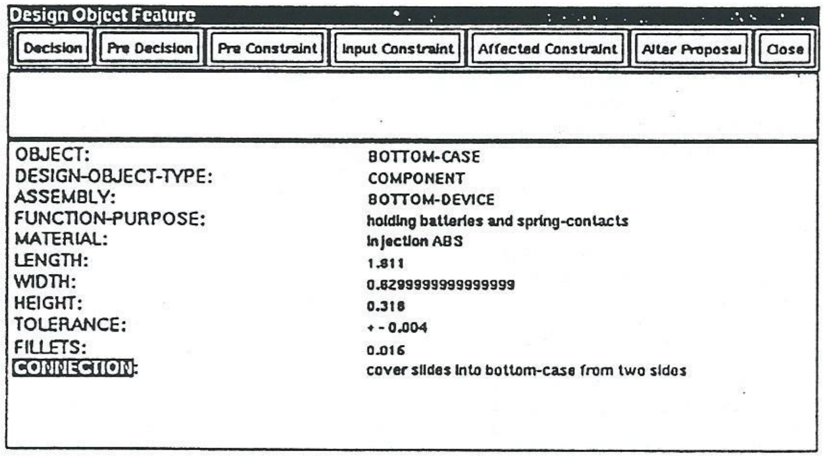

Figure 5
Design Object Feature

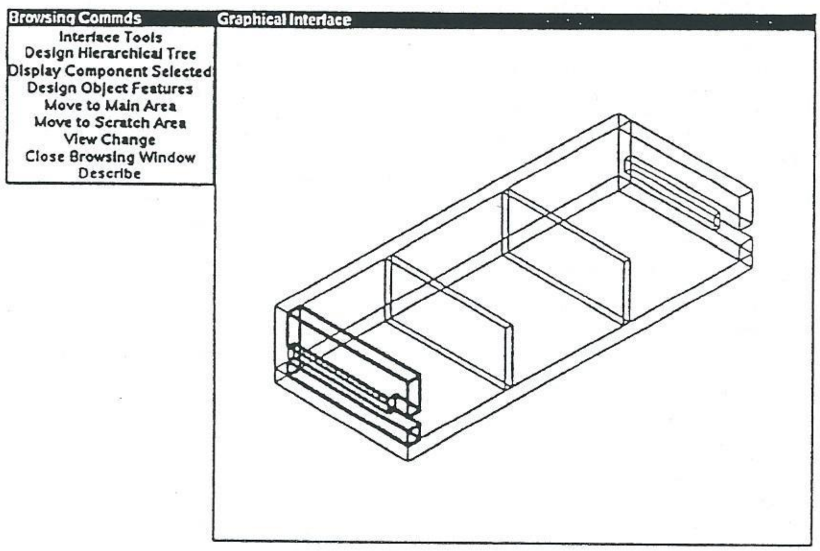

Figure 6
Two-way Associativity Example: Part 1

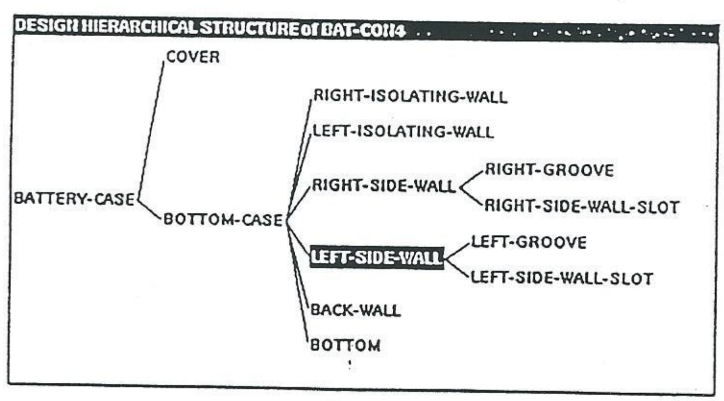

Figure 7
Two-way Associativity Example: Part 2

the names of the design object parameters; the other shows their current values. From these, users can inspect standard information of the kind stored by most current CAD systems, such as length, width, overall tolerance, and fillets. One of unique characteristics of the Design History Tool is that it allows users to inspect the function-purpose feature, which describes the purpose of the design object, and the connection feature, which describes how the design object connects to other objects. In this example, the bottom-case function-purpose is to "hold batteries and spring-contacts". The cover is connected to the bottom-case by sliding onto the bottom-case from the two sides.

Another unique feature of the Design History Tool is that the isometric drawing is mouse sensitive. This feature allows the mouse to select any individual design object in the isometric drawing, and correspondingly, the selected design object will be highlighted in the hierarchical structure graph. It has been observed that this two-way associativity is very useful for users to locate design objects and inspect the relationship among the design objects. This two-way associativity can be also achieved through selecting any design object node in the hierarchical structure graph (Figure 3) and the selected design object will be lighted in the corresponding isometric drawing. Figure 6 and Figure 7 present an example of the two-way associativity. In addition to the two-way associativity, the isometric drawing interface also allows the inspecting of an individual design object. This can be done by selecting a desired design object in the isometric drawing and then selecting command, "Display Component selected" in the menu attached to the drawing. Figure 8 presents an example of this operation from Figure 6.

VI.2. Browsing Design Evolution

The second browsing capability allows users to trace a design from initial specification to final detailed design. This capability will be illustrated through an example of browsing the evolution of the left-side-wall-slot function-purpose. As shown in Figure 8, the left-side-wall-slot is selected. In order to inspect the function-purpose evolution, the user opens its feature interface by selecting the command "Design Object Features" from the menu attached to the isometric drawing. Figure 9 presents this feature interface. It can be seen that the function-purpose is to "fasten cover". The design evolution information is retrieved by selecting the Pre Decision (preceding decisions) or Pre Constraint (preceding constraints) commands from the feature interface, as shown in Figure 9. The evolution is displayed as a graph, as shown in Figures 10a and 10b. Because of space limitations, the whole graph is divided into two graphs for this paper. The graph shows a sequence of decisions backward in time to the beginning of the design. These decisions ultimately determined the function-purpose of the left-side-wall-slot. From the graph, it can be seen that at the beginning of the design, the two functions of the battery-case were determined. Then the battery-case functions were refined into the bottom-case and cover functions. After that, the bottom-case connection was designed that determined how the bottom-case and cover were connected to each other. Based on the bottom-case and cover connection, the side-wall-u-groove function was determined. The side-wall-u-groove is shown in Figure 8. Then finally, the side-wall-slot function-purpose was determined.

Each node in the evolution graph can be further inspected. This is done by selecting the desired node in the graph via the mouse. Figure 11 presents an example of inspecting side-wall-u-groove function-purpose value. This example is retrieved by first selecting side-wall-u-groove node in the evolution graph, and then selecting the resulting-constraints entity in the side-wall-u-groove decision interface which is not given here.

VI.3. Browsing Design Rationale and Alternatives

The third browsing capability is to retrieve design rationale and alternatives. This browsing capability allows users to

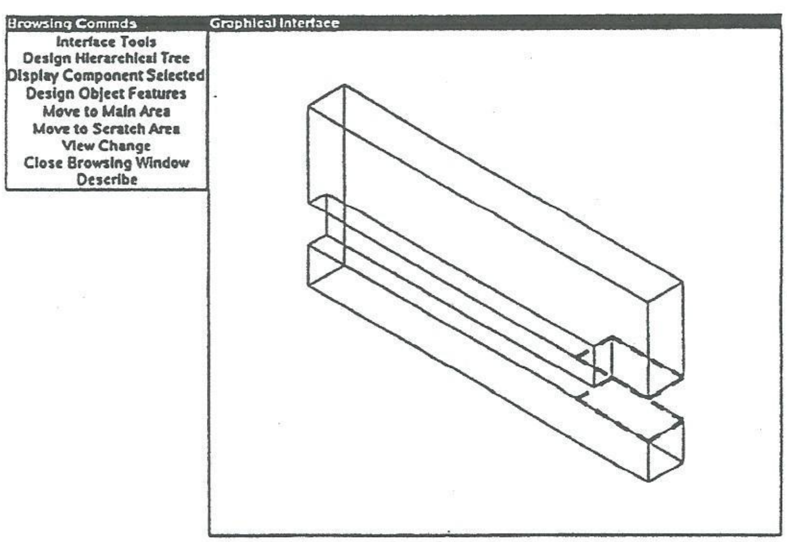

Figure 8
Display of Selected Feature

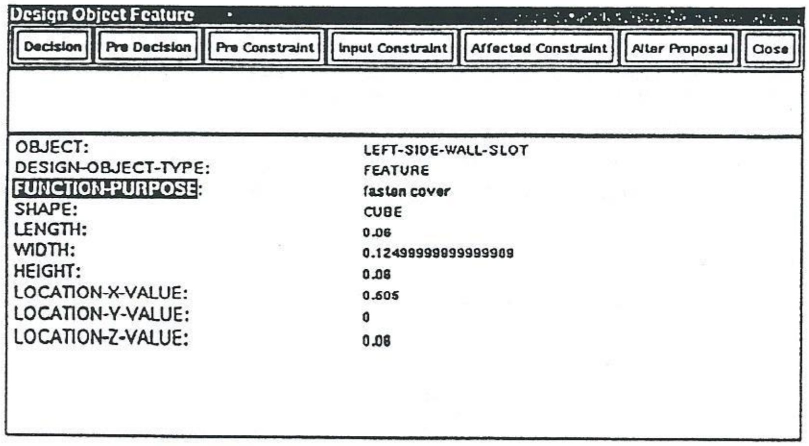

Figure 9
Design Object Feature Details

retrieve information on why and how each decision was made, what alternatives were considered, and why they were rejected. This retrieval is accomplished by first selecting the feature of interest and then selecting the **Alter Proposal** (alternative proposal) and **Decision** commands from the feature interface. To demonstrate the browsing capability, an example of browsing the alternatives of the **bottom-case** connection is shown. In Figure 5, the **bottom-case** connection feature is selected. Its value is "cover slides into bottom-case from two sides". By selecting the **Alter Proposal** command from the menu, an "alternative proposal" interface is opened, as shown in Figure 12. From the interface, it can be seen that, for the **bottom-case** connection, there are three alternative proposals. One is "Using glue or solvent to stick bottom case to cover", another is "Cover and bottom case snap down from top", and the last one is "Screw join the two cases". The **decision-rationale** entry gives the rationale for why these alternatives were not accepted. For example, the last proposal was rejected because "Use screw joint is more expensive and requires additional equipment." Currently, the representation of the alternatives employs unstructured text. A new version of the representation is being developed in which the alternatives are represented by a more formalized argument structure.

VI.4. Browsing Constraint Dependencies

The last browsing capability allows users to inspect the dependencies among constraints. This information explains how design object features depend on each other, and thus it is very useful for design understanding and redesign. Figure 13 presents an example of one of the constraint dependencies, called input constraint dependency. This dependency shows how a design object feature directly or indirectly depends on other design object features and other constraints. From Figure 13, for example, it can be seen the **bottom-case** function-purpose was derived from the **battery-case** function-purpose and the **lower-half-part** assembly. This means that these two constraints directly influenced the bottom-case function-purpose value setting because they were inputs to the decision that created the **bottom-case** function-purpose constraint.

Another constraint dependency that can be inspected is called the affected constraint dependency. This dependency shows how a design object feature or a constraint directly or indirectly influences other design object features. In other words, if the design object feature value were to change, all of the dependent design object features might need to change as well. This dependency is very useful for the redesign process. It gives users some idea of the potential impact of a proposed design change. Figure 14 presents a simple example of the affected constraint dependency graph. From Figure 14, it can be seen that the **bottom-case** length influence the **back-wall** length and the **back-wall-u-groove** location. If the **bottom-case** length is changed, the two design object features could be affected and may need to change as well.

The input and affected constraint dependency graphs are retrieved from the design object feature interface by selecting the **Input Constraint** and **Affected Constraint**

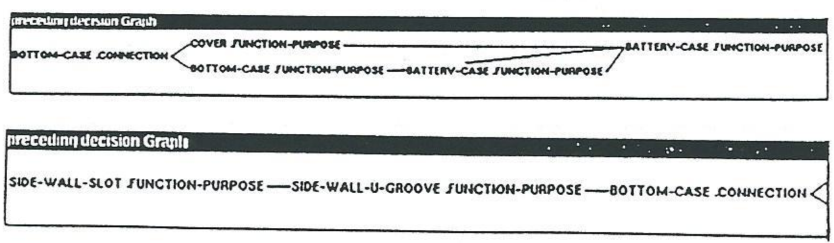

Figure 10
Design Evolution

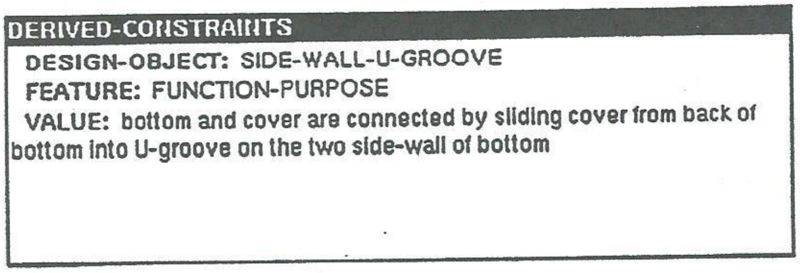

Figure 11
Details of Derived Constraints

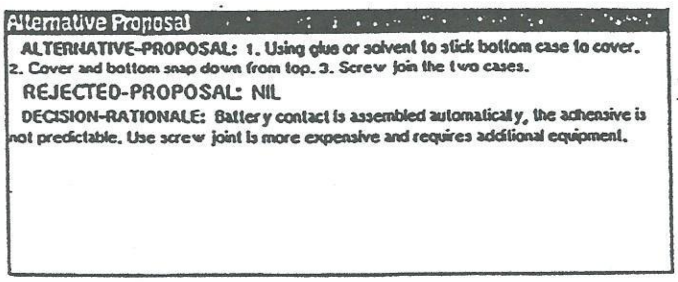

Figure 12
Alternative Proposals

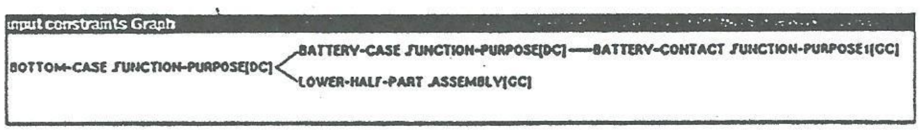

Figure 13
Input Constraints

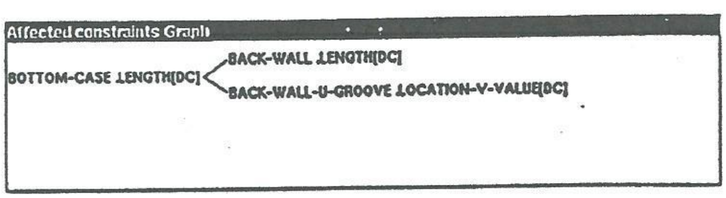

Figure 14
Affected Constraints

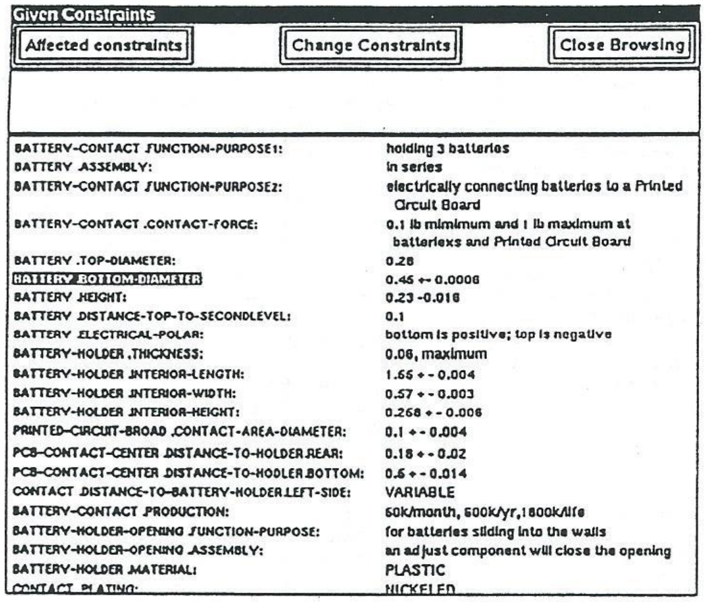

Figure 15
Given Constraints

commands. Before the retrieval, it is necessary to select a design object feature for which user wants to examine the constraint dependencies. The constraint dependency graphs, like the design evolution graph, also support further inspecting of each constraint in the graph by mouse selection.

In addition to browsing the relationships among the design object features, it is also possible to inspect the relationship between initial specifications (i.e., given constraints) and the design object features. Figure 15 presents a given constraint interface for the battery contact design. This interface displays all of the original design specifications given to the designer. It allows users to browse these initial design requirements. For example, the function-purpose of the **battery-contact** is to hold batteries and electrically connect them to the Printed Circuit Broad. More importantly, the interface supports retrieving the dependency relationship between each given constraint and the design object features. Suppose the **battery** bottom diameter increases, its original value is 0.45 inches, as shown in Figure 15. What constraints will be affected by the change? To retrieve this information, the user first selects the **battery** bottom diameter entity, as shown in Figure 15, and then selects the command **"Affected Constraint"**. Figure 16 presents the information graph. Because of space limitations, the graph is trimmed for the example, the whole graph is much bigger than this one. From the graph, it can be seen that if the bottom diameter of the **battery** changes, it will influence the **isolating-wall** length, and **bottom-case** material. If the bottom-case is redesigned for the variation of the battery diameter, both of these decisions will need to be reconsidered to avoid introducing new design flaws.

VII Conclusion

The research conducted thus far has succeeded in developing a design history tool that is capable of documenting and playing back the initial, intermediate, and final states of a design as well as the design process that connects them. By modelling the design process as a series of the design decisions, the representation captures the original designer's intent. The design decisions result in new derived constraints that in turn instantiate or modify the values or relationships in the design. This representation was derived from two kinds of protocol studies, and it is still being refined and expanded as the research progresses and a wider range mechanical design problems are examined.

The design playback provides basic facilities for browsing, retrieving, and

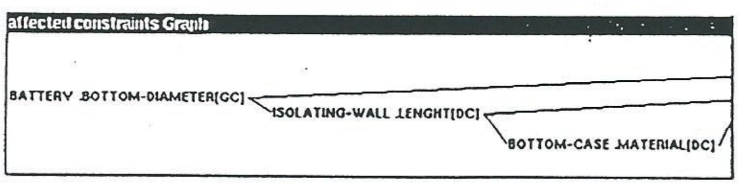

Figure 16
Effect of Given Constraints

displaying the design artifacts, the design evolution, the design rationale and alternatives, and the constraint dependencies. More browsing functionalities will be established based on experiments in which designers, manufacturing engineers, and management personnel use the tool to perform design reviews and modifications of existing designs.

The goal of this research is to develop methods for representing and displaying the design history information. Two important goals for future research are (a) to develop methods for nonintrusive capture of the design history information and (b) to evaluate the feasibility, generalizability, and effectiveness of the design history tool on a wider range of design problems.

REFERENCES

[Balakumar, 88]
Balakumar, P., Robert, J.C., Hoffman, R., Ikeuchi, K., and Kanade, T., "VANTAGE: A Frame-Based Geometric Modeling System", Carnegie-Mellon University, 1988.

[Brown, 89]
Brown, D.C., "Using Design History Systems for Technology Transfer", Proceedings of the MIT-JSME Workshop on Cooperative Product Development, Nov 1989.

[Charon, 89]
Charon, R., "Development of a Graphical Interface to the Design History Tool", Masters Thesis, Department of Computer Science, Oregon State University, Nov 1989.

[Kuffner, 89]
Kuffner, T.A., "Mechanical Design History Content: The Information Request of Design Engineers", Masters Thesis, Department Mechanical Engineering, Oregon State University, December, 1989.

[Kuffner, 90]
Kuffner T.A., and Ullman, D.G. "How Designers Understand Existing Designs" 1990 Conference on Mechanical Design Theory and Methodology, Chicago, Sept 1990.

[McGinnis, 89]
McGinnis, B. and Ullman, D.G., "The Evolution of Commitments in the Design of a Component", International Conference on Engineering Design, Harrogate UK, Aug 1989, pp. 467-495.

[Schoen, 89]
Schoen, E., R.G. Smith, R.G., and Carando, P.J., "Ruling with Class (Programming with Rules in Class)", Research Note, Schlumberger Laboratory for Computer Science, February 1989.

[Smith, 88]
Smith, R.G., Kleyn, M.F., and Schoen, E., "Impulse Reference Guide", Research Note SYS-47-41, Schlumberger Technologies Corporation, January 1988.

[Smith, 88]
Smith, R.G. and Carando, P.J., "Structured Object Programming In Strobe", Research Note SYS-86-26, Schlumberger-Doll Research, October 1987.

[Stauffer, 87]
Stauffer L., Ullman D.G., and Dietterich T.G., "An Empirical Study on the Process of Mechanical Design", Doctoral Dissertation, Department of Mechanical Engineering, Oregon State University, Sept. 1987.

[Tikerpuu, 88]
Tikerpuu, J. and Ullman, D.G. "Data Representations for Mechanical Design Based on Empirical Data", International Computers in Engineering Conference, San Francisco, Aug 1988, pp. 245-254.

[Tikerpuu, 89]
Tikerpuu, J., "General Feature-Based Frame Representation for Describing Mechanical Engineering Design, Developed from Empirical Data", Masters Thesis, Department of Mechanical Engineering, Oregon State University, March 1989.

[Ullman, 87]
Ullman, D.G. and Dietterich, T.G, Mechanical Design Methodology: Implications on Future Developments of Computer Aided Design and Knowledge Based System", Engineering with Computers #2 1987, pp 21-29.

[Ullman, 88a]
Ullman, D.G., Dietterich, T.G., and L. Stauffer, "A Model of the Mechanical Design Process Based on Empirical Data", Artificial Intelligence in Engineering, Design, and Manufacturing, Vol 2 (1) 33-52, 1988.

[Ullman, 88b]
Ullman, D.G., Dietterich, T.G., and L. Stauffer, "A Model of the Mechanical Design Process Based on Empirical Data: A Summary", AIEng88 conference, Palo Alto, August 1988. pp. 193-215.