

B. D. McGinnis
Schlumberger CAD/CAM
Ann Arbor, MI

D. G. Ullman
Associate Professor
Department of Mechanical Engineering
Oregon State University
Corvallis, OR

The Evolution of Commitments in the Design of a Component

This paper traces the evolution of a component of a mechanical design from its initial inception to completion. This research focuses on the interplay of design objects, features, constraints, and design decisions. Based on empirical data extracted from verbal design protocols, a design terminology is developed. Constraints were identified in the protocol and classified according to their source, level of abstraction, and form or function orientation. A structure for constraints was also developed which decomposes the constraints and classifies them into ten basic feature relationships.

1 Introduction

This paper traces the design of a mechanical component from its initial inception through to the final detailed design. The data for this paper is based on a protocol study in which five designers were video and audio recorded as they designed machines to meet abstract needs. This data has yielded a model of the design process in enough detail to answer many questions fundamental to the development of computer aided design tools, the evaluation of design quality, and the development of effective design methodologies and teaching techniques. Previous papers detailing this model are referenced in this paper.

To explore component development in design, the design history of a single component was extracted from one of the protocols. One goal of collecting this data is to show the development and propagation of constraints, the relation of design features to constraints, and to identify the needs for representation in mechanical design. Additionally, this effort continues research on the concept of a design history in that every decision made that helped the component progress from its inception to the final, detail drawings was noted.

In the design of a component some constraints on its form and function are given at the beginning of the problem. Other constraints arise from the designer's knowledge of the domain. Yet others are derived from each design decision made. It is the interplay of these types of design constraints that form the evolutionary refinement of the component. This evolution is traced in detail and the constraint-driven nature of the design clearly demonstrated.

A second goal is to clarify the definitions of and show the relationships between the terms "constraint" and "design feature." These terms are currently used in many inconsistent ways. In order to trace the history of the design of the component considered, the definitions of constraint and feature are forced to be clarified and the relations between them evolved. In developing this history, requirements for design data representation are identified. This was the subject of an

earlier paper but is still being refined in studies such as this.

The organization of this paper will be to first describe the protocol data that is the basis for this study (Section 2). In Section 3 the terms feature, design object, constraint, and design decision are defined as well as the interactions that exist between them. This is followed by Sections 4 and 5, where protocol data concerning features and constraints are presented. A constraint mapping technique used to reduce the data is presented in Section 6. Conclusions drawn from this data are in Section 7.

2 Data for the Study

To understand the performance of a design engineer it was necessary to observe engineers engaged in the design process. This was accomplished in this research through a technique called protocol analysis. Protocol analysis involved the video and audio taping of engineers as they solved specific design problems. During their solutions they were asked to "remain verbal" [1, 2]. The engineers' verbalizations were then transcribed. By reading the transcripts and viewing the designers' gestures and drawing action on video tape, many types of protocol data can be extracted from the design [1, 3]. Here, the protocol data extracted were the features and constraints of the design effort.

Protocol analysis is the best way known to these researchers to gather information on how humans solve complex problems. Since the major use of the data developed here is for the development of interactive computer aided design tools, it is important to know how human designers behave in order to develop acceptable human/machine interfaces. However, the protocol data is limited to what the designer verbalizes, gestures, or draws. It is conceded that not all the design activity is verbalized in the protocols as humans think faster than they can talk. But, past research has shown that what is recorded gives enough information on the overall activity to aid in understanding the design process.

In previous studies [1, 2], a total of five separate design

Contributed by the Design and Design Theory Committee for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received January 1990.

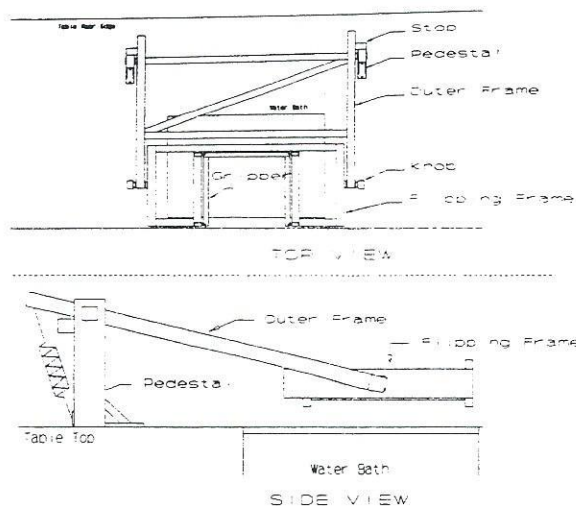


Fig. 1 Top and side view of flipper-dipper assembly

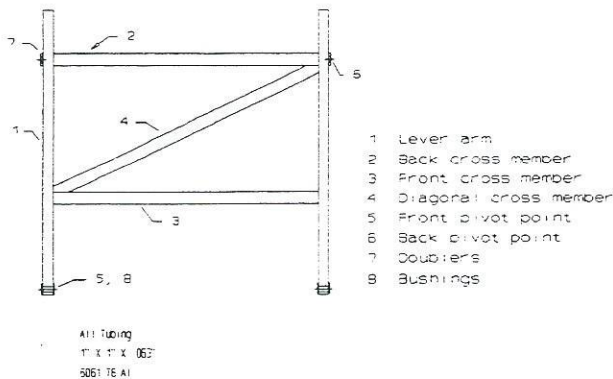


Fig. 2 Top view outer frame assembly

protocols were recorded. Two experienced design engineers solved a problem concerned with the packaging of small batteries in a computer and three design engineers solved the problem used for this study. The design problem presented to the engineer whose protocol is used in this study (the selection criteria follows this paragraph) was that of coating both sides of an aluminum plate with a thin film. The design specifications presented to the subject gave the required operation which included a worker loading a 10 in. \times 10 in. \times .063 in. aluminum plate into the machine to be designed. The machine then must lower the plate to the surface of a water bath. On the water surface is a sticky, thin chemical layer which adheres to the plate on contact. The plate must then be lifted off the surface, flipped over and the process repeated for the other side. Finally, the worker removes the plate with the film on it and loads a new one to be coated. Based on the action required by the machine, this problem has been dubbed the "flipper-dipper" problem. Along with the functional requirements above, specific form constraints on the handling of the plate and the allowable space for the machine were also given.

Since the goal of this research was to track one component through its evolution it was necessary to select a single subject's protocol for study. The selected protocol was chosen because of the clarity of the protocol data and the relative straightforward nature of the design process and of the finished design itself. The chosen protocol consists of a pedestal, outer frame, flipping frame, and gripper as is shown in Fig. 1. This figure is a cleaned up CAD drawing of the final design drawing generated by the designer. The operation of the mechanism is as follows: the operator places an aluminum plate into slots located on the gripper assembly, the gripper which is attached

"Let's see how much we've got in the back. We've got 32" minus 15, 17" here to play with, so we need, surface of the water is here, 1/2" below and if we make most of this take place near the front of the tank, say 1" from the front of the tank, here's a plate, that'd give us the longest moment arm there, or fulcrum, so we get the maximum elevation pivot point, pivot point is up above the, okay, if we have to we can make it up here. Okay. And the rods here, rod here, and it has to go equal distance past here, so that it slides equal distance from both sides of the pivot point. And the pivot point will also be, on the same plane as the pivot point we'd also have to have our guide so that it didn't extend below or equal distance on each side of the pivot point, like so, the framework. Okay, here we go. And, that would take care of that arm coming back. Okay."

19.17 we've got 17" here to play with [from back of sink to back edge of table] (drawing 4.2)
 19.23 most of this [machine process] take place near the front of the tank, 1" from front of tank
 19.27 longest moment arm there, or fulcrum
 19.27 get the maximum elevation in the [front] pivot point
 19.31 [front] pivot point is up above the..... make it up here (drawing 4.2 A)
 19.45 that arm coming back (drawing 4.2 B)

Fig. 3 Text and accompanying reduction from protocol

to the flipping frame is then lowered to the water surface for the chemical adhesion, the plate is then raised above the water bath and rotated about a pivot between the flipping frame and outer frame, and the plate is again lowered with the opposite side down; once the chemical adhesion has taken place the plate is again raised and removed from the gripper assembly. To account for the thickness of the flipping frame, the gripper assembly slides on rods so that the lower edge of the plate is always below the lower edge of the flipping frame.

Due to its simplicity the design object chosen for the study was that of the outer frame assembly of the machine. The outer frame along with its decomposition into its major components is shown in Fig. 2. The main function of the outer frame assembly is to raise and lower the flipping frame over the water bath. The components that comprise the outer frame are the lever arms, the front cross member, the back cross member, the diagonal cross member, the front and back pivot points, the doublers, and the bushings. Each of these components has certain features that are pertinent in the design. The constraints associated with these individual features are the ones focused on in this research.

The protocol studied encompassed a total design period of 4 hours and 38 minutes, recorded over three separate design sessions on subsequent days. The transcript for this protocol consists of 85 pages of typed, double-spaced verbalizations, and 28 separate design drawings made by the subject (all drawings were done by hand). These range from conceptual sketches to complete detailed drawings from which Figs. 1 and 2 were taken. The development of the outer frame assembly was contained in 40 percent of the total transcript and was represented on 34 percent of the drawings.

As an example of the data, consider the section of transcript presented in Fig. 3, which is about one minute long. In this example the subject is about 1/3 of the way through the design. He is trying to position the front pivot point (the pivot between the outer frame and the flipping frame) and subsequently determine the lever arm length. This topic is reconsidered four other times during the protocol, each time with new design information having been generated in the interim. The numbers preceding the lines refer to page and line number of the protocol from which the data was extracted. In this section the designer is referencing a sketch he made that is shown in Fig. 4.

The text, the figure, and the associated video tape for this and other similar segments of the protocol that refer to the outer frame assembly are the raw data on which this study is based. In order to deal with the data it was first reduced to a more intelligible form. This reduction can be found in the lower section of Fig. 3.

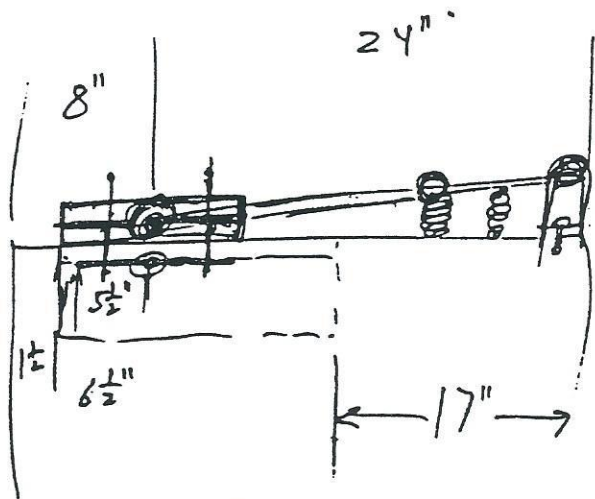


Fig. 4 Drawing from protocol

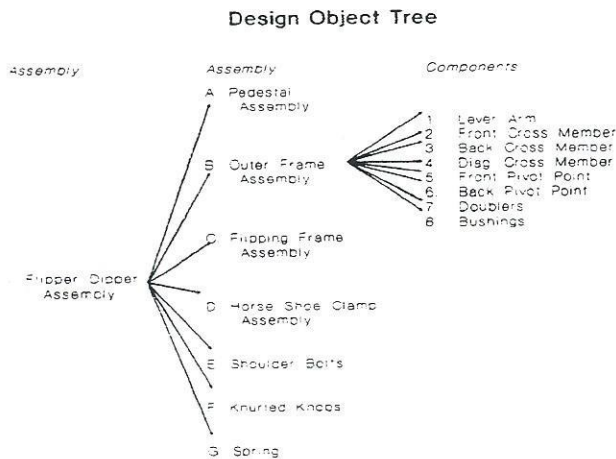


Fig. 5

It should be noted that not all of the information available in the transcript sample was extracted. Only information relevant to the outer frame was considered.

The important points to note about the above reduced text are that: (1) This reduction does capture the design progress of the protocol; (2) It is easier to follow than the original text; and (3) The six statements are all of a similar structure. This last point sets the tone for the remainder of this paper. Namely, after examining the constraint statements on the outer frame assembly, it was hypothesized that all the data could be represented in terms of features and constraints and that design decisions could be observed through the changes in them.

3 Design Objects, Features Constraints, and Design Decisions

One of the main objectives of this research is to establish an acceptable terminology which can be used in describing the design process. By following the progression of constraints and the evolution of features in the design protocol, the formulation of this terminology is achieved. Thus the terminology developed in the following sections was arrived at through considerable review of the design data.

3.1 Design Objects. During the design effort an engineer decomposes the design into assemblies, components, interfaces, and features of assemblies, components, and interfaces. All aggregations of components are termed an "assembly" [4]. Assemblies can be thought of as parent or child in nature, in that an assembly can have a larger parent assembly or have

Expanded Object Tree With Component Feature

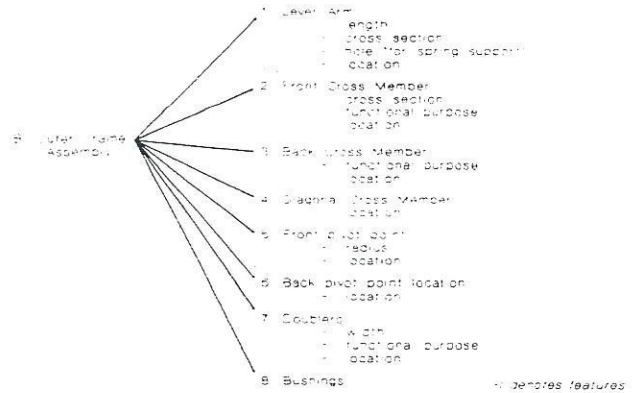


Fig. 6

children or (sub)assemblies. Since there is no limit to the number of assembly generations, the term "assembly" is used for all component aggregations. "Components" are individual parts of a design.

The design objects are the physical artifacts of the design. The conglomeration of the design objects are what constitute a particular design.

A design object tree, which graphically displays the hierarchical breakdown for the design studied, is shown in Fig. 5. This diagram shows the breakdown of the entire assembly into its children assemblies. Also depicted is the breakdown of the outer frame assembly into its eight components.

The next section focuses on features. However, before defining features it must be noted that the breakdown of design objects into assemblies, components, and interfaces is inadequate in that it is too coarse. Usually it is necessary to deal with the features of a design object or commonly, the features of a feature. For example, the lever arm is a component. However, much design activity was expended on the front pivot point, a feature of the lever arm. The front pivot point, in turn, has many features such as its location. Thus, features themselves must be treated as design objects.

It is important to define all the entities of a design as design objects, in order to discuss them as a group and because this breakdown is consistent with the way in which a designer refers to the design. At any given moment in the design, the designer focuses his attention on one of the design objects. The distinct properties of the design object that the engineer deals with when working on it are commonly called design features.

3.2 Design Features. There has been a substantial amount of work done in the area of design feature definition and use [5, 6, 7, 8]. These efforts represent a variety of different backgrounds and differing points of view. This has resulted in a variety of definitions for the term "feature." These definitions all have validity in their respective fields or areas of interest. However, here the concern is for the terminology used by the designer and so the definition must be very broad. Thus, this research's formal definition for a feature is:

A feature is any particular or specific characteristic of a design object that contains or relates information about that object. It is semantically represented in the form of a noun or noun clause.

From the above definition it can be surmised that anything and everything about a mechanical design object can be considered a feature, such as: location, length, material, operation, etc. Depending on the point of view taken, some features are obviously more important than others. In other words, the relevancy of a feature is dependent on the focus at the time. For example, the manufacturing feature of a surface finish may not be important to a designer during the conceptual phase

DESIGN OBJECTS

(assemblies, components, features, interfaces)

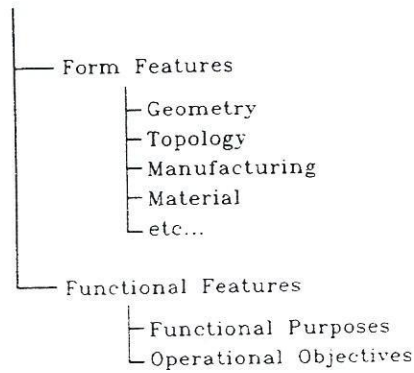


Fig. 7 Feature diagram

of the design process, but is important during manufacturing. Likewise some functional features of how a design object behaves or what its purpose is, which may be critical to a designer, may not be important from a manufacturing point of view. An expanded object tree displaying some of the features of the components used in the design of the outer frame assembly is shown in Fig. 6. The features shown in this tree are those that the engineer devoted a considerable amount of time to. From the sections of the protocol studied, some 277 features and 38 design objects used by the designer were identified. Out of the 277 features, 95 (34 percent) of them directly describe the outer frame. The remainder of the features were of components, interfaces, or assemblies that acted as constraints on the outer frame.

There are two distinct types of features used by the designer: form features and function features. Form features include geometrical, topological, manufacturing, and tolerance features along with any other features used to describe the physical structure of the design object. Functional features include both the *purpose* of the design object such as support, stability, or strength and the *behavior* that the design object performs like lifting, gripping, or rotating. The form features embody the physical characteristics of design objects in a design while the functional features explain what purpose the design objects achieve individually and what behaviors they exhibit in the overall design.

The feature diagram in Fig. 7 displays how the features are segregated into form or function. To better understand the way in which form and function features describe an object, consider the front pivot point in the design studied. Some important form features include the shape, depth, diameter, location, and manufacturing method. A functional feature of the pivot point is that of providing an axis on which the flipping frame can rotate.

3.3 Constraints. In the protocols, constraints are statements made about a design's features, such as: "the length is 19 inches" and "the knob diameter is smaller than the outer arm height." In the first example the feature "length" is constrained by the value 19 inches. The second example relates two features, "the knob diameter" and "the outer arm height" by a conditional relation of "smaller." In reviewing the constraint statements like those at the end of Section 2, for all the protocol sections focused on the design of the outer frame assembly, the 277 features were involved in 725 constraint relations. These have been classified into two types of semantic structures. By using these structures every constraint in the protocol has been modeled. Independent of the protocol studied, the constraint types given below are thought to represent all types of constraints possible.

The first semantic structure is of the form <feature-instan-

tiation>. Here the term "feature" is as defined in the previous section and term "instantiation" refers to the value specified for the feature. From the protocol sections reviewed, two types of feature relationships of this structure were found. These are:

- (1) <function feature> - <instantiation>
- (2) <form feature> - <instantiation>

Below are examples of these two constraint types found in the protocol. The dashes (-) signify the separation between the feature and instantiation.

(1) [machine operation time - is 40 seconds]

Here the feature being instantiated is the functional feature of "machine operation time." It is being instantiated with the descriptive value "is 40 seconds." This instantiation comes from the given design specifications and is therefore a reference made by the designer as opposed to his changing or giving some new value to the feature.

(2) [flipping frame depth - needs to be at least 10 1/2 in.]

The form feature here "flipping frame depth" is instantiated with the value "at least 10 1/2 inches." Here a feature is given conditional initial value that it must achieve.

Of the above two examples the first came from the given specifications and the second from the design itself. Many of the instantiations observed involved the initialization or modification of features in the design itself. It is these instantiations that give the current state of the feature. By observing the change in the representation and value of the instantiation, the history of the development of the feature can be tracked.

The other semantic structure observed for the constraints was of the form: <dependent feature—relation—dependent feature>. As there are both form and function features and there can be both form and functional relations, there are potentially eight different feature relationships of this structure. These are:

- (3) <form feature>—<form relation>—<form feature(s)>
- (4) <form feature>—<function relation>—<form feature(s)>
- (5) <function feature>—<form relation>—<function feature(s)>
- (6) <function feature>—<function relation>—<function feature(s)>
- (7) <form feature>—<form relation>—<function feature(s)>
- (8) <form feature>—<function relation>—<function feature(s)>
- (9) <function feature>—<form relation>—<form feature(s)>
- (10) <function feature>—<function relation>—<form feature(s)>

In each of these, one or more features constrain, through the relationship, a single feature. Examples of each of these constraint types are:

(3) [knob location—in middle of—flipping frame outer arm location]

Here the form feature, "knob location," is constrained by the form relation, "in middle of," to the form feature "flipping frame outer arm location." This was used as a form constraint on the positioning of the knob.

(4) [flipping frame—pivots on—shoulder bolts]

The first form feature, "flipping frame," is given the function of "pivots on" in reference to the second form feature "shoulder bolts." This relation acts as functional constraint on the flipping frame.

(5) [] (<function feature>—<form relation>—<function feature>)

This relationship was not found in the section of protocol that was codified. This relationship is theoretically possible but cannot be substantiated by the data reviewed.

(6) [front cross member purpose—to prevent—outer frame racking]

Here the functional feature of “front cross member purpose” is related to another functional feature of “outer frame racking,” by the functional relation of “prevent.” Here the functional purpose of preventing outer frame racking was satisfied by the front cross member.

(7) [front pivot point elevation—would be 8 inches to allow for—flipping frame rotation]

Here the form feature, “front pivot point elevation,” is subject to the conditional form relation of “would be 8 inches to allow for” in regards to the function feature, “flipping frame rotation.” This is used as form constraint on determining the location of the front pivot point.

(8) [back pivot point location—limits—framework side travel]

This relationship shows that the form feature of “back pivot point” has a functional relationship of “limits” on the function feature of “framework side travel.” This constraint performs a simulation on a function of the back pivot point.

(9) [flipping frame rotation—occurs 1 inch from—tank front edge]

In this relationship, the functional feature of “flipping frame rotation” is related to the form feature of “tank front edge” by the form relation of “occurs 1 inch from.” Thus the position of the water tank is imposing a form constraint on the flipping frame operation.

(10) [spring purpose—hold out of water—flipping frame]

Here the function feature, “spring purpose,” is related to the functional relation of “hold out of water” with respect to the form feature “flipping frame.” The only difference between this relationship and that shown above in [6] is that the independent feature is of a form type.

All the constraints extracted from the design protocol can be broken down into one of the ten relationships discussed above. By representing the constraints in these semantic structures, both the progress and interdependency of the features can be examined. The processes by which new constraints are generated or added to the design are the decisions made by the designer.

3.4 Design Decisions. In this constraint model of the design process, design decisions are defined in terms of constraint changes.

A design decision is defined as occurring each time the value of a feature of a design object or the relationship between features is changed or initialized.

Design decisions may be driven by several different constraints emanating from separate design objects. The result of a decision is usually a single constraint, but multiple constraints are sometimes generated. The result of the design decision can affect any level in the design object tree but is usually a modification of a specific feature value or relation, on which the decision was focused. The driving constraints of a design decision are referred to as the *forcing constraints* of that particular decision. The *resulting constraints*, on the other hand, are the direct consequences of design decisions and represent new information generated in the design. Once a design decision is made, this new information can be used as forcing constraints in future design decisions.

4 Constraint Classification

Once the definitions were completed and the constraints extracted from the design protocol, the task of classifying the constraints was initiated. The classification of the constraints was performed not only for purposes of analysis, but also in hopes of developing a more formal constraint language, upon which a representation could be built. Even if the classification only provided an elaborate bookkeeping tool, it would still be beneficial in the development of a constraint management system. The classification also gives insight into how the designer uses constraints in a design and relates the significance of some constraints as opposed to others. Constraints are classified by three different measures: constraint source, constraint structure, and level of abstraction. Below is an explanation of each of the different measures and statistics from the protocol reduction of the outer frame assembly for the protocol studied.

4.1 Constraint Source. A constraint can originate from one of three different sources: it can be given, introduced or derived. Given constraints are those furnished to the designer which provide an idea of what is required by the design. A given constraint is usually received by the designer at the beginning of the design as part of the design specifications and requirements. These can be either form or function oriented and can originate from clients, designers of adjacent design objects, or initial specifications. Given constraints can be used at any time in the design process, whenever the designer requires information on the initial state of the design.

An introduced constraint is one which is brought into the design from the designer's domain knowledge, handbooks, or other “domain knowledge” sources. It is a constraint that is brought in from outside the current problem solution space and has not been derived from any other constraints. Introduced constraints often are vague and general by nature. Examples of introduced constraints are “keep everything as light as possible” or “make maximum use of the table.” Other introduced constraints include equations and hard data used in analysis such as material properties, but none of these were observed in the protocol examined.

Derived constraints are generated inside the design space and are intrinsic to the design being worked on. Derived constraints result from and can be changed by the outcome of design decisions. Although design decisions can be affected by given, introduced, or derived constraints, the result of the decision is always a change or initialization of a derived constraint.

The process of design for a mechanical part can be thought of as the creation and refinement of a set of constraints. Initially these constraints consist of a group of given constraints that define desired function and/or forms for a particular application. The given constraints comprise the initial specifications of the design. As the design process proceeds, this body of constraints grows and develops until a final concrete solution is achieved.

In the beginning of the design, the design engineer is presented with given constraints that are primarily functional, that describe the desired operation of the design. Along with the functional definition of the design, very specific form constraints are often presented in the given specifications (e.g., the dimensions of the aluminum plate).

As the designer proceeds, the given constraints become less abstract and develop into concrete form and function instantiations for specific design objects. In the protocol studied, the engineer took the given specification of “coating both sides of an aluminum plate” and developed it into “a person would load it [plate], dip it, rotate it, dip it, and then a person would have to flip it back to unload it.” This revision of the functional constraints allows the engineer to begin the conceptualization of the design objects needed for the design function. The next three paragraphs discuss the three stages of the design (con-

Constraint Source

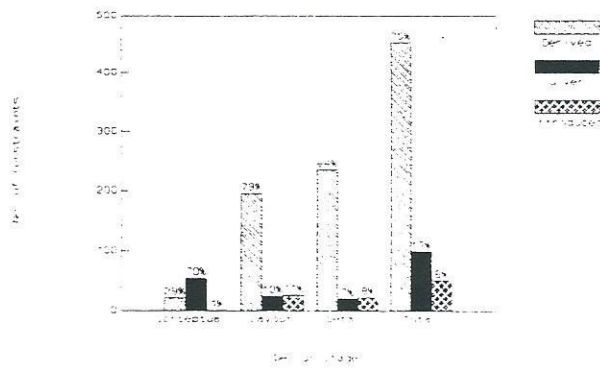


Fig. 8

Feature Relationships

(Constraint Structures)

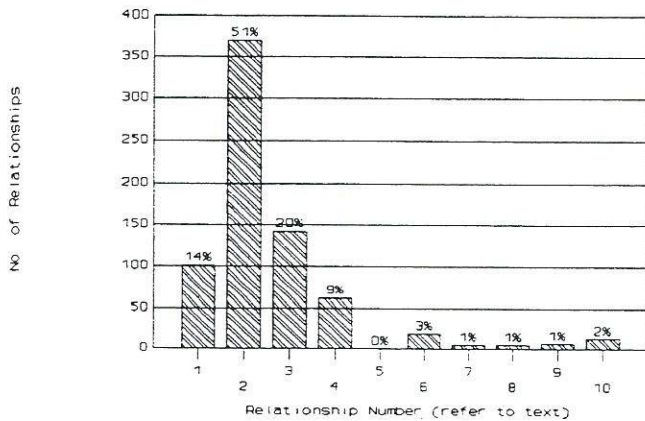


Fig. 9

ceptual, layout, and detail), in reference to the constraint source: given, introduced, or derived. In analyzing the data, the concept of three distinct design stages was adopted: conceptual, layout, and detail. These stages are really management labels given to the level of design detail and are hard to apply since the design process is so interleaved. While one feature may be in the layout stage, other adjacent components or features of the object in question may still be conceptual. Therefore the design stage divisions assumed in this research were based on the drawings produced by the designer during the protocol. Rough sketches were treated as conceptual design representations. After establishing a basic layout of the design with rough sketches, the designer used a straight edge to generate scale drawings with rough dimensions. These drawings were taken as representations of layout design. The designer then made drawings with detailed dimensions and manufacturing notes. These were considered as representations of detailed design. These stages are important when trying to understand how the constraints affect the design and how the constraints are propagated through the design. A graph of the constraint source percentage for the three design stages and for the total design process of the flipper dipper design protocol is given in Fig. 8.

As the design's initial constraints are refined, design objects are created to meet the new requirements. During the conceptual stage of the design, the majority of the constraints (70 percent, see Fig. 8), are given constraints. From the initial constraints are born the conceptual form of the design. In the

Level of Abstraction

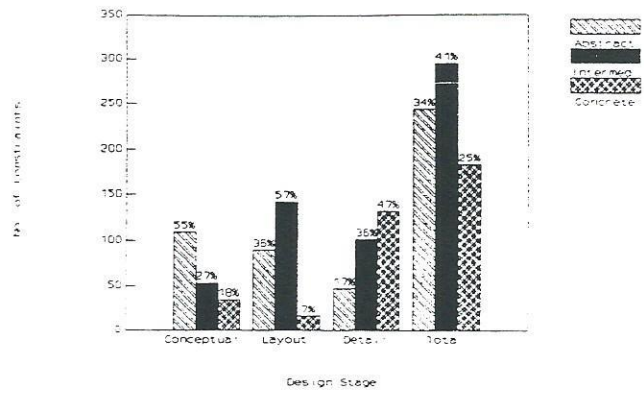


Fig. 10

development of the conceptual design, form features are created out of functional requirements and then verified by simulation of the design operation.

Once a completed conceptual design is created, the designer begins the layout stage of the design process. In the layout stage the designer patches and refines his design and checks to see that there are no conflicts with any of the form constraints given in the design specifications. The main driving force of the layout stage are the derived constraints (79 percent). It should be noted here that almost half of these derived constraints originated from design objects outside the outer frame assembly.

In the detail stage of the design, the engineer is primarily refining the design to ensure that all the design objects work together. The main drivers of this design stage are previously derived form constraints (84 percent). This refinement process consists of comparing different form constraints to ensure that there are no conflicts in the design, and changing or "patching" the constraints when conflicts arise. The resulting product of this constraint propagation and feature evolution is a formalized detailed design from which the mechanism can be constructed.

When examining the percentages for the overall design, the number of derived constraints is dominant, (75 percent of the total constraints). This value suggests that the designer deals primarily with constraints generated by the design itself rather than constraints brought in from outside the design space.

4.2 Constraint Structure. As discussed earlier there were ten different constraint structures possible in the protocol. The percentage of each type of structure was calculated and plotted in Fig. 9. By far the largest percentage of relationships (51 percent) fell into the second type, [form feature—instantiation]. The next highest percentage (20 percent) was for the third relation, [form feature—form relation—form feature]. This supports the earlier notion that the majority of the work performed was concentrated on the form features. It should be noted that most of the first relationship types, [functional feature—instantiation], were verbal simulations of the machine's operation. (i.e., [machine operation-operator slides plate in, pivots it and turns the frame that holds it].)

The 10 constraint structures have been divided into two separate categories: form constraints and function constraints. Structures 2, 3, 5, 7, and 9 are form constraints. Structure 2 gives the instantiation of a form feature whereas structures 3, 5, 7, and 9 give a form relation between either form or function features. Likewise, structures 1, 4, 6, 8, and 10 are function constraints. Structure 1 gives the instantiation of a function feature whereas structures 4, 5, 7, and 8 give functional relationships between either form or function features.

Nearly three-quarters (72 percent) of the constraints are of the form type and the other quarter (28 percent) are functional. Since the design problem studied was driven primarily by functional requirements, it might be expected that for a design with more form oriented specifications, the percentages observed would have an even wider spread.

4.3 Level of Abstraction. The level of abstraction for a constraint [3] can be classified into one of three sub-divisions: abstract, intermediate, or concrete. The concept of abstraction is actually a continuous function rather than the discrete form used here. However, codifying abstraction is difficult and thus the three divisions are very loosely bound and may overlap each other in certain instances. Also note that the abstraction level refers to the "instantiation" or "relation" the features attain rather than the representation of the feature. Therefore the current abstraction level of a feature is represented by its most recent instantiation.

Abstract constraints are very general and are usually concentrated in the conceptual stage of the design process. An example of an abstract constraint is "the small axle on a framework [the front pivot point] is here," which is the first conceptualization of the front pivot.

Intermediate constraints are more descriptive in nature but still not fully defined in a precise quantitative notation. These constraints are predominantly found in the layout stage of the design but can be dispersed throughout the design effort. An intermediate constraint is "our [front] pivot point we want right in the middle of the tank." This constraint tells approximately where the pivot should be (in the middle of the tank), but not exactly.

Concrete constraints are very specific, giving exact quantitative values or specific functional qualifications. As expected these constraints are found primarily in the detail stage of the design. A concrete constraint is "our [front] pivot point will be 6 in. from the center." Constraints tend to start as abstract and develop through intermediate into concrete specifications. However, this is not always the case as many design objects are brought into the design (introduced) at a concrete level.

As with the constraint sources, the levels of abstraction were also grouped by the three different design stages: conceptual, layout, and detail. The resulting percentages can be seen graphically in Fig. 10. In the conceptual stage the majority of the constraints were abstract (55 percent), which is expected. In this stage the concrete constraints appear to be higher than what might be anticipated (18 percent), but this can be accounted for when realizing that many given constraints were stated in a concrete form. In the layout stage, the intermediate constraints are predominant (57 percent), which is not surprising considering the above definitions. In the detail stage of the design, the concrete constraints have the largest magnitude (47 percent). They are followed somewhat closely by the intermediate constraints (36 percent). The design studied was not completed through to the level of full detailed drawings, but instead to the high level layouts with manufacturing notes. It is expected that with a more detailed level design, the number of concrete constraints would increase greatly in the detail stage of the design.

5 Conclusions

The focus of this paper is concerned with the development and propagation of constraints and features in the mechanical design process. This was documented by studying the constraints that affected the design of one part's features in an entire design protocol. The evolution of the part's features involved the generation of design decisions. These design decisions were forced by inputted given, introduced, and derived constraints and resulted in new derived constraints.

By identifying and codifying the constraints, a better understanding of the necessary requirements for representing them was established. The constraint classification which was proposed in references 3 and 4 was tested and validated. This classification consisted of the constraint source, level of abstraction, and structure. The structure developed for the constraints was based on feature relationships. These feature relationships were of two form:

<dependent feature>—<instantiation>
<dependent feature>—<relation>—<independent feature>

From these basic patterns, 10 different structures were generated. These structures were used to successfully model all the constraints identified in the design protocol studied.

The research performed resulted in a substantial insight into constraint and feature generation and propagation in the process of mechanical design. With the creation of a constraint classification based on feature instantiations and relations, the next step, that of building a formal feature/constraint representation, can proceed.

References

- 1 Stauffer, L. A., Ullman, D. G., and Dietrich, T. G., 1987, "Protocol Analysis of Mechanical Engineering Design," International Conference on Engineering Design, Boston, MA.
- 2 Ullman, D. G., Stauffer, L. A., and Dietrich, T. G., 1987, "Toward Expert CAD," *ASME Computers in Mechanical Engineering*, Vol. 6, No. 3, pp. 21-29.
- 3 Ullman, D. G., Dietrich, T. G., and Stauffer, L. A., 1988, "A Model of the Mechanical Design Process Based on Empirical Data: A Summary," *Proceedings of the AI in Engineering Conference*, Stanford University, Aug.
- 4 Tikerpuu, J., and Ullman, D. G., 1988, "General Feature Based Frame Representation for Describing Mechanical Engineering Design Developed From Empirical Data," International Computers in Engineering Conference, San Francisco, CA, August, p. 245.
- 5 Cunningham, J. J., and Dixon, J. R., 1988, "Designing With Features: the Origin of Features," International Computers in Engineering Conference, San Francisco, CA, August, p. 237.
- 6 Dixon, J. R., Duffey, M. R., Irani, K. L., Meunier, K. L., and Orelup, M. F., 1988, "A Proposed Taxonomy of Mechanical Design Problems," International Computers in Engineering Conference, San Francisco, CA, August, p. 41.
- 7 Unger, M. B., and Ray, S. R., 1988, "Feature-Based Process Planning in the AMFR," International Computers in Engineering Conference, San Francisco, CA, August, p. 563.
- 8 Shah, J. J., and Wilson, P. R., "Analysis of Knowledge Abstraction, Representation and International Requirements for Computer Aided Engineering," *Journal of Design Studies*, in press.