The alternate use of abstraction and refinement in conceptual mechanical design

J Paz-Soldan
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/meche
NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making
of photocopies or other reproductions of copyrighted material. Any copying of this
document without permission of its author may be prohibited by law.
The Alternate Use of Abstraction and Refinement in Conceptual Mechanical Design

by

J.P. Paz-Soldan, J. R. Rinderle

EDRC 24-22-90  C.3
The Alternate Use of Abstraction and Refinement in Conceptual Mechanical Design

JUAN PEDRO PAZ-SOLDAN
Advanced Systems Group
UNICAD Inc.
Norfolk, VA 23510

JAMES R. RINDERLE
Mechanical Engineering Department
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract
In this paper we identify the characteristics of conceptual mechanical design problems which make them hard to solve and hard to study. We discuss the relationships between these problems and other cognitive tasks and explain why conceptual mechanical design problems are difficult to study and explain using the Information Processing paradigm the theoretical framework for Verbal Protocol Analysis and Expert Systems.

The nature of conceptual design goals, constraints and constraint discovery increase the difficulty of conceptual design problem solving perse and analysis of problem solving methodologies. We postulate the use of alternate abstraction and refinement as a key to successful conceptual design problem solving and problem analysis and we identify three types of abstractions: Functional Perspectives, Localization, and Worst Case Evaluation. Protocol episodes demonstrate how alternate use of abstraction and refinement can help designers deal with circular constraints insufficiency of constraints; and bi-directional function to structure constraints.

Introduction
The process of conceptual mechanical design is still poorly understood as a cognitive task. Descriptive models of the design process do not correlate with observed behavior of subjects solving conceptual mechanical design problems. Expert System implementations of computational design models have only been successful in well defined problem domains and for problems where a predetermined hierarchical decomposition of the original design problem into relatively independent subproblems is possible and useful [Hoover 89]. Descriptive cognitive theories cannot yet explain problem solving behavior of general configuration design problems where three dimensional, and causal physical reasoning occurs.

We seek a better understanding of the conceptual design process, particularly mechanical system design. We hope that this understanding will provide a theoretical framework for better conceptual mechanical design software and will help to identify those problem solving skills that should be emphasized to improve the effectiveness of engineering design education.

Conceptual Mechanical Design as Problem Solving
Conceptual design of mechanical engineering systems is a special kind of problem with characteristics that set it apart from simpler problems used in cognitive psychology studies. This is not to say that conceptual mechanical design is a unique problem not comparable to any other type of problem, but only that it contains elements which increase the difficulty in applying any one set of conclusions obtained through simpler problems. Even when cognitive psychologists talk about "hard" problems, e.g. [Simon 85], they are talking about hard "toy" problems and easy "toy" problems.

Conceptual design problems have similarities with textbook physics problems, which have been studied by cognitive psychologists [Larkin 87,Laririn 80] and modeled using Artificial Intelligence techniques [Novak 77]. Both conceptual design and physics problems deal with mathematical abstractions, causal physical constraints, and three-dimensional objects with constraints. However, textbook physics problems have a clear goal statement which can be used as a test of success. In contrast, design problems usually require the problem solver to define (or redefine) the goal.

Conceptual design problems are not well defined according to the notion of "well defined-ness" suggested by [McCarthy 80] and formalized by Newell and Simon [Newell 72]:

A problem proposed to an information processing system is well defined if a test exists, performable by the system, that will determine whether an object proposed as a solution is in fact a solution.

Newell and Simon implicitly rely on this definition of "well defined-ness" to cast problems into either the Set or the Search Space representations that form the basis of their work [Newell 72].

In Closing, the conclusion of this article by Simon at is in a very familiar sense: "...the concept of a problem is not well understood..."
Although Newell and Simon cooside the "describing a machine" (p. 73) to be as example of a well defined problem, they have wisely cautioned to deal only with parameterized design tasks, where all of the possible design solutions can be generated by varying the parameter values and an explicit cost minimization function is used as a test criterion. During the initial conceptual design stage of realistic machine design, there are no parametric relations nor, in fact, parameters to speak of. The minimization of a cost function is therefore not a usable test of success, since no such function is available.

Conceptual mechanical design problems are in some ways comparable to creative writing tasks in the sense that neither are well defined problems. Many of the techniques we have observed in conceptual design seem also to be employed by writers. In fact, the advantages, limitations, and current development efforts in software aids for design and writing tasks have striking similarities: In both, software can facilitate the editing and storage at the detailed rework stage but do not yet provide time savings sufficient to justify their use during the conceptual stages.

Although there are similarities, in most cases, conceptual design differs from writing in a (cognitively) important respect. Conceptual mechanical design involves three dimensional geometric reasoning and causal physical reasoning. Geometric reasoning is a poorly understood human ability [Kosslyn 80] and causal physical reasoning has only recently been modeled to any extent [Bobrow 84, Hoover 89, Kuipers 84]. Only when we start to understand how humans are able to reason in these two domains will the differences become irrelevant for our purpose of understanding conceptual design as a cognitive task.

Characteristics of Conceptual Design Problems and Protocol Analysis

The nature of conceptual design problems makes them hard to solve and hard to study with the cognitive science technique of Protocol Analysis [Ericsson 84]. In this section we discuss the characteristics of conceptual design problems and relate them to problem difficulty and problem analysis difficulty.

To illustrate the type of problem we have in mind when we talk about conceptual mechanical design, consider the following: 1

Develop two rough configurations of a printer head drive mechanism. You should specify as many standard components as you can, for example, a motor, pulleys, cables, belts, gears, shafts, etc. Avoid the use of exotic or imaginary components. Ignore manufacturing and materials specification for now.

We are not aware of the me of a writing tasks as a sample problem of a Cognitive Psychology study, presumably because the characteristics they share with conceptual design tasks are those that inate bedi types of tasks difficult to analyze.

3 A Protocol Analysis is a technique used to study problem solving behavior through the analysis of statements made by the subject during problem solving. Briefly, the basic assumptions of Protocol Analysis are that the subject's behavior can be viewed as a search through a problem space, accumulating knowledge... as he goes', and that 'each step in the search involves the application of an operator', moving the subject to a new point in the problem space' (p.63) [Ericsson 84]. The fundamental objective of a protocol analysis is to generate a Problem Space and a set of Operators.

The problem space is simply an approximation to the subject's internal representation of the problem, which can be represented by a graph [Newell 72]. An operator is an action which 'produces new states of knowledge from existing states of knowledge' [Newell 72]. These two concepts are the theoretical equivalent of the Expert Systems terms working memory and production, respectively.

Appended is a rough sketch of the printer case into which your design has to be fitted.

The print head drive design task typifies the difficulties present in most conceptual design problems that as a whole, differentiate these problems from 'toy' problems used in cognitive psychology studies. The conceptual design problem can be characterized as follows:

1. Nature of goal
   - Problem has many "good" solutions
   - Solution domain is not explicit
   - Goal statement does not directly contain goal test

2. Nature of constraints
   - Problem constraints form webs (dense interdependencies)
   - Bi-directional Function to Structure constraints exist
   - Problem is under-constrained
   - Circular constraints exist
   - 3D geometric constraints exist

3. Nature of constraint discovery during problem solving
   - There is insufficient information
   - Problem has potential branching into difficult subproblems

These characteristics are discussed in the following sections.

Nature of goal

The absence of an explicit goal test, the multiplicity of solutions and the impracticality of finding solutions by elimination ame from the typical vagueness of conceptual design problem statements. Various psychological studies have dealt with these characteristics individually, but it is the collective existence of all of these characteristics that needs to be considered before we can confidently apply the Information Processing paradigm [Newell 72] and its data collection technique. Protocol Analysis to conceptual mechanical design problems.

In practice, the redefinition of the goal condition (by the problem solver) during a verbal protocol may either force the intervention of the experimenter to clarify the intended goal or require the use of multiple Problem Spaces and Operator sets to explain the subject's behavior. Either of these weaken the underlying objective of the protocol analysis approach, which is to postulate a common Problem Space and set of Operators for the class of tasks being studied. The repeated reexamination of the goal statement was observed in the three protocols for the printer design task. The existence of many "good" solutions in such domains is a consequence of having a vague goal statement that needs to be redefined by the problem solver.

In many cases problems are posed in such a way that the problem itself dearly identifies the range of acceptable solutions. The solver may continuously refine the range by testing various subranges and ultimately puts forth a solution using the constructs of the original problem statement. In conceptual design the problem statement does not usually clearly delineate the range of acceptable solutions nor is it posed in the language which must ultimately be used to specify a solution.

With the possible exception of this last characteristic, the difficulties stemming from the vagueness of the problem statements in conceptual design translate more into difficulties in the analysis of problem solving approaches than in the problem solving itself due to the ability of humans to deal with abstract problems.
Function to refer to behavior

* function to refer to a device that reveals its purpose, and use behavior to refer to its operating characteristics. He offers the following clarifying example: The function of a steam-release valve on a boiler is to prevent explosion; the behavior of the system is simply that the pressure remains below a certain limit.

The existing literature frequently obscures this distinction by using the term function* to refer to behavior.

In addition, we are assigning a very specific meaning to structure which differs from its use in other related works (eg., see (Ulrich 87)). Since, the etymological latin root of "structure", means to heap together, to arrange. Listed definitions of common usage of "structure" include: "The interrelation of all the parts or the whole," and "structure of the interrelation of all the parts or the whole." Our definition is closer to the etymological origin than to common engineering usage. The characteristics of constraint discovery are similar to those discussed previously, however the relation is bidirectional in the sense that a required function may drive a component selection, but the geometric and behavioral component class constraints limit possible functionality. There are a limited number of these types of relationships among design variables that the designer can deal with effectively.

The amount of detail in these component relations often works against the designer during the conceptual stage. How can we determine the values needed to compute the required torque of the motor if we only have a vague idea of the print head drive configuration? Conceptual design problems are usually under-constrained, meaning that the designer may have to estimate certain values before he can use the component equations that provide behavioral or geometric parameter values.

As the configuration becomes more completely specified, a quite different situation often arises, that of circular dependencies among constraints. If, for example, we specify that the motor will be mounted on the platform it drives (as is the case in at least one commercially available printer we have used), then the required motor torque depends on load which depends on motor mass which depends on motor torque. It is easy to see in a verbal protocol when...
These findings can be related to the conceptual mechanical design problem characteristics discussed, and in particular, those dealing with the nature of constraints:

1. Since most mechanical components introduce bi-directional function/behavior constraints, a preliminary solution has to be "patched up" as new components are introduced.
2. The use of notes and drawings is necessary to free up attention to concentrate on constraint web disentanglement, circular constraints* and three dimensional geometric constraints†.
3. An opportunistic approach is necessary for the same reason that the "patching up" strategy is needed: Many of the bi-directional function to structure constraints only become apparent when the design resolution is increased.
4. Focusing attention on small aspects of the design is necessary to deal with these same characteristics of conceptual design problems, and also with those we have classified as pertaining to the nature of constraint discovery: The lack of information and subproblems of potential greater complexity than the original task.
5. The density of constraint interaction and the difficulty of reasoning about bi-directional function to structure constraints impose heavy demands on the memory of designers, causing them to forget previously identified constraints.

The results of Ullman et al’s study motivated one of us to use the print head drive problem in a verbal protocol study [Paz-Soldan 87]. The task was in many ways similar to Ullman’s, being an incomplete, vague specification, however, the instructions asked explicitly for several alternative solutions. Approximately 45 minutes were allocated for the design task. The three subjects were Mechanical Engineering graduate students at Carnegie Mellon University, and all were working on projects which required considerable conceptual mechanical design.

After 45 minutes, each of the three subjects had a rough sketch of a configuration but two of the three subjects had difficulty outlining the configuration so as to require the use of alternatives to direct motor coupling. The problem solving approach observed during the protocols involved alternating increase and decrease of design detail.

The Alternate Use of Abstraction and Refinement
To avoid confusion, we provide some working definitions of what we mean by abstraction and refinement. Abstraction in this context is the cognitive process of considering only a simplified or limited set of attributes of an object. Refinement is the opposite of abstraction; the addition of detail or complexity to the object representation. The representation can be mental or external (e.g. a sketch) or a combination of both.

Both abstraction and refinement have been observed during conceptual design tasks but we suggest that the alternate use is an important aspect of their use by designers. This is an extension of Ullman et al’s observation of opportunistic problem solving during conceptual design tasks. It is also a refutation of many design loop diagrams of the design process which presume a progression of refinement until a design-impasse is reached and (he process is restarted. The problem solving approach observed during the protocols involved alternating increase and decrease of design detail.

Thus abstraction is not only used to "identify the existing problems" [Pahl 84], or to "hypothesize a... key idea or solution plan" [Kant 84]. It is also used to deal with dense constraint webs, circular constraints and unknown (and hard to determine) constraints by making simplifying or worst-case assumptions. This is in fact, the central idea underlying all successful engineering problem solving.

Similarly, refinement is not only used to "break down overall function into subfunctions" [Pahl 84], or to "decompose a problem into subproblems" [Kant 84]. It is also used to deal with the three dimensionality of mechanical systems and the varying amount of detail available about selected components. It is also used to generate new constraints from existing constraints.

To illustrate the use of alternate abstraction and refinement, we use an excerpt from S2’s protocol. Previous to this excerpt S2 generated some alternatives for the overall configuration. In this excerpt he starts to specify the motor capacity from geometric constraints:

137: you'd like to know about how big that oossr u going to be,
138: and that... you can kinda get an idea how much torque the motor can put out
139: by how much space you allocate for it
140: Best thing of all would be to have that motor directly coupled
141: to whatever is driving the platform.
142: But!, knowing that the platform goes...
143: probably all the way to the edge of the boi you can’t do it!
144: So you have to go to some kind of gearing srito... or some kind of cable, or whatever... timing belt

Several things are noteworthy in this excerpt. Noose’s flaw to the problem of motor behavioral sizing (torque) is simplified to be one of geometrical sizing. Then a new refinement on the configuration is proposed: Attaching the motor directly to the pinform. This proposal is immediately followed by discovering a geometric constraint on the specified printer casing and roller tmgangement. Finally, this is translated into a refinement of the overall configuration so as to require the use of alternatives to direct motor coupling.
It is obvious that the classification of individual thoughts as being either abstractions or refinements is not straightforward, however, it is clear that the designers' reasoning is not a simple progression of refinements by the method of addition of detail. Conceptual design involves quick alternation of reasoning in which detail is removed to focus on a single aspect of behavior or geometry, followed by the addition of a new detail of behavior or geometry which had not been mentioned previously in the protocol.

The following is a second example from a later stage in S2's protocol:

215: we'll... put a... timing belt.
216: And now we think why this is a bad idea.
217: One reason it is a bad idea is the timing belt is elastic by nature.
218: and it probably has some dynamics because it's so elastic.
219: You need quick starting and stopping,
220: and I probably can ignore those dynamics.

S2 is first considering a broad range of behavioral characteristics of the timing belt including the elastic properties but then adopts an abstraction (neglecting elastic behavior) which allows him to reason about the importance of dynamic effects.

Preliminary Classification of Abstractions

In this section we attempt a preliminary classification of abstraction mechanisms based on examination of the protocols and introspection of our own reasoning during creative design tasks. The classification is based on the nature of constraints which are neglected or emphasized. We identify three major types of abstraction:
1. Functional Perspective
2. Localization
3. Worst Case

Functional Perspectives are used to ignore aspects of geometry or behavior to address a specific functional constraint. Given a known component such as a motor, Functional Perspectives involve the removal of detail in order to focus attention on a characteristic that permits checking that a particular constraint is satisfied. For example, consider the following excerpts from S2's protocol:

34: how do we get the printer head to move translationally.
60: We need a reversible motor
137: you'd like to know about how big that motor is going to be
178: a motor that has a very low starting torque

The motor is seen first as a source of power and none of its geometric characteristics are considered. Within this initial functional perspective, S2 considers only general behavioral aspects of the motor. Later, the geometric value motor height is the only aspect of the motor's structure that S2 uses to decide where to place it. Finally, the starting torque behavioral aspect of the motor is considered during the specification of the connection to the platform.

Localization is used to neglect the system wide effects of behavioral or geometric constraints to resolve circular constraints within a subsystem or across subsystems. Localization facilitates decision making on a limited scale by eliminating system wide considerations.

The following excerpt illustrates the use of localization:

309: Put a drive gear... Where can we put a drive gear?
310: How can we attach... well all we want to do is drive one end,
311: the other end is free,
312: if it goes right over a pulley,
313: we put the timing belt on,
314: attach the timing belt to the platform,
315: and zoom that anchor back and forth with the geared up motor.
316: ...
317: Looks like... means... just looking at the picture.
318: there's not much room between the print head and the side of the box.
319: So you've gonna have to have a really small gear there.
320: We put a little small... small gear here.

Notice in this excerpt how the overall system is considered, and then attention is focused on the localized geometric interference constraint discovered in statement 317.

Lastly, Best/Worst Case abstractions are used to establish boundaries for behavioral or geometric values in under constrained problems. These abstractions can be optimistic (best case) or pessimistic (worst case). They are used to establish bounds on values when there is not enough information to determine them more precisely.

Worst case abstraction for a geometric value is illustrated by the following:

292: We now are going to make our motor that... probably about an inch.
293: not more than an inch an a half in diameter.

In the following excerpt S2 deals quickly with a missing geometric dependency by assuming a "best case" scenario:

197: In one scheme now, we'll replace one of those poles by a ball screw.
198: We know the platform is...
199: We don't have any dimensions for the platform.
200: We can make it anything we want.
201: We'll make the platform big enough so we can pass the ball screw through it.

These three types of abstractions, Functional Perspectives, Localization, and Best/Worst Case, have been observed in the protocols and have been identified in our own design reasoning. Each of these abstractions are used during conceptual design to deal with the difficulties arising from the nature of the goal, constraints, and constraint discovery.

Hierarchical Problem Solving and Conceptual Design

The process of problem decomposition into subproblems is a central aspect of prescriptive, cognitive, and computational models of the design task. Each model also incorporates an iterative approach to design and implicitly incorporates abstraction and refinement. These models also assume a hierarchical decomposition of the problem into subproblems. Antecedent subproblems completely include posterior subproblems and there is limited or no interaction among subproblems. For example, Moursier and Dixon's model of mechanical design [Moursier 86] explicitly assumes that "the design problem has been decomposed into systems and sub-systems a priori." They observe that "usually, there is some natural decomposition based on function or the physical characteristics of the system." Other Systems for engineering design, such as PRIDE [Mittal 86], MICON [Balam 86], and HI-RISE [Maher 85] share these properties, i.e. an underlying assumption of problem decomposability into independent subproblems and the existence of a parametric model.7 Expert Systems cannot be easily developed for configuration design problems in which there is considerable interaction among subproblems.

Although initially we can decompose a conceptual design problem into a hierarchy of subsystems and corresponding subproblems, the problem solving itself cannot be considered hierarchical. Although

7The similarity among problem scopes in these engineering design Expert Systems and Newell and Simon's carefully worded example is no coincidence: The Information Processing paradigm [Newell 72] is the theoretical basis for all Expert Systems.
our subjects decomposed the primer system into three subsystems, their problem solving strategy was far from hieuchhial! The three subjects jumped back and forth between various subsystems in order to resolve dense constraint coupling among them. Dense constraint coupling is inherent in mechanical systems because designers seek to reduce weight and volume of collections of connected components [Sussman MtRinderle 86] and because stringent connectivity limitations reduce positioning alternatives [Hoover 89]. Uuman et al's observation [Uuman 87] that designers employ opportunistic refinement is an experimental confirmation of the limitation of hierarchical problem solving approaches in conceptual design.

Recent papers by Ulrich and Seeing [Ulrich 88], and Hoover and Rinderle [Hoover 89] on computational models of the design process start to address the problems presented by the non-hierarchical nature of mechanical configuration design problems. These papers highlight the need for function sharing in good mechanical designs as a result of the unintended structure (behavior and/or geometry) contributed by all real mechanical components. However, the difficulties associated with geometric reasoning are altogether ignored in [Ulrich 88], and only partially dealt with in [Hoover 89]. As pointed out in [Dixon 87] and [Libardi 88], supporting abstract geometric specification is an area in need of new research initiatives.

Conclusion
Conceptual mechanical design has characteristics which differentiate it from simpler problems used in cognitive psychology studies. These characteristics relate to the nature of the goal, to the nature of problem constraints, and to the discovery of constraints during problem solving. Certain problems studied by cognitive psychologists share some of these characteristics, but the presence of all these characteristics sets apart conceptual mechanical design problems from those used in most cognitive psychology studies.

Conceptual design problems are "ill defined" problems, and as such, are not easily cast into the Information Processing Paradigm, which provides the theoretical foundation for Verbal Protocol Analysis and Expert Systems worke These problems involve geometric and causal physical reasoning, two poorly understood cognitive activities. Due to these characteristics, conceptual design problems are both hard to solve and hard to study using current cognitive psychology methodology.

In spite of that, verbal protocols provide a useful experimental methodology for data collection on conceptual design problem solving. Using this technique, Uuman et al have observed the use of opportunistic refinement during conceptual mechanical design tasks. Using a similar study by Paz-Soldan, we have identified the use of alternate abstraction and refinement as a strategy used for successful conceptual design problem solving.

Abstractions used during conceptual design can be classified by constraint type. We have identified three types: Functional Perspectives, Localization, and Best/Worst Case abstractions. Each of these has been illustrated with excerpts from the protocols. The use of patterns in applying these types of abstraction and their associated refinement process is an area for further investigation.

In closing, we distinguish between hierarchical representations and hierarchical problem solving and we discuss the effects of functional integration and incidental behaviors, two characteristics of mechanical components which prevent a hierarchical problem solving strategy in conceptual design. The alternate use of abstraction and refinement facilitates successful conceptual design problem solving because it allows the problem solver to deal with dense constraint dependencies and bi-directional function to structure constraints. We believe the ability to use abstraction and refinement alternately during design will prove to be an important aspect of systems which can support or automate conceptual design tasks. A representation for conceptual designs that allows the alternate removal and addition of detail is discussed in [Rinderle 90].

Acknowledgments
The authors are pleased to acknowledge the support of the Design Theory and Methodology Program of the National Science Foundation (NSF Grants DMC-84-51619 and DMC-88-14760) and the Engineering Design Research Center at Carnegie Mellon University (NSF Grant CDR-85-22616).

References
[Bobrow84]

[Dixon 87]

[Kant 84]

[Libardi 88]

[Maher85]

[McCarthy 80]

[Meunier86]

[Mittal81]

[Polanyi 84]
[Newell 72]

[Novak 77]

[PiM 84]

[Pz-Soldan 87]

[Rinderle 86]

[Rinderle 90]

[Simon 85]

[Sussman 80]

[Ullman 87]

[Ulrich 87]

[Ulrich 88]

[Wallace 87]