PROMPT: An Innovative Design Tool

Seshashayee S. Murthy and Sanjaya Addanki

IBM T.J.Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

ABSTRACT

We describe a system, **Prompt**, used to design physical systems. Prompt employs a multi-level approach to design. When simple constraint propagation over prototypes [Adda85] fails, Prompt can significantly modify prototypes by reasoning about their structure and physics. Prompt derives the behavior of a prototype from its structure using knowledge of physics stored in a **Graph of Models**. It then uses heuristics called **Modification Operators** to control the process of modifying the prototypes. In this paper we describe how our system works in the domain of structural design. We describe the kinds of analysis Prompt performs on beams and how it makes innovative changes to prototypes.

1. Introduction

Existing design systems [Srir85, Mitc85, Mitt86b] configure their designs from a fixed finite library of atomic components. The design process in such systems is best thought of as hierarchical refinement. These systems start with a rough schematic that is successively refined at each step. Each stage of the refinement process generates constraints which are used to guide the partial design; successive refinements are performed and in the final stage, the atomic components and their parameter values are chosen so as to satisfy these constraints. Such design systems use functional models [Joha85] of prototypes and therefore cannot reason about, or modify, the structure or behavior of the atomic components.

It is often the case that the constraints generated by such a process cannot be satisfied by any combination of parameters of the components in the library. For example, it is evident that weight and torsional stiffness requirements conflict in the design of beams. If the system's entire library of beams consists of a solid, circular beam parameterized by the radius and the length of the beam, it is easy to conceive of a design specification that cannot be met by varying parameters over the library. The solution to such design problems lies in modifying one or more prototypes in the library. In order to modify a prototype, the system must first be able to derive the behavior of the prototype from its structure and the underlying physics. The system must then use the results of this analysis to appropriately modify the prototype.

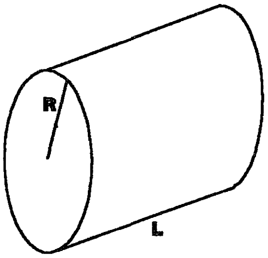
There are two major difficulties with developing a system that works in this manner. First, analysis is expensive. Analysis requires representing large engineering domains and prototypes can be analyzed along many different dimensions. Second, mapping the results of analysis to structural changes requires sophisticated reasoning about complex equations. In its full generality the Analyze-Modify loop is very powerful but inefficient.

Prompt is a design system that is capable of analyzing and modifying the prototypes in its library. Prompt introduces two mechanisms to help alleviate the difficulties of the Analyze-Modify loop. **Graphs of Models** , are helpful in reasoning about large engineering domains. [Adda87, Penb87] describes the Graph of Models paradigm in detail. **Modification Operators** consist of three parts; a set of preconditions that determine the application of the operator, a set of heuristics that help focus the analysis of the prototype, and a set of heuristics that control the mapping from the results of the analysis to the structural changes. In our example above, the second part of a Modification operator directs Prompt to analyze the prototype of the solid circular beam with respect to stress distribution. The analysis is carried out using the Graph of Models paradigm. The third part of the operator directs Prompt to redistribute the mass of the rod, thus inventing the concept of a tube. More powerful than pure parameterization and less powerful than general reasoning from first principles, Modification Operators offer an intermediate level of reasoning about structural changes.

The rest of this paper describes the main ideas in Prompt by working through our example. Beam design is an interesting domain for various reasons. First, the theory is well understood, thus providing a firm foundation for analysis from the first principles. Second, the domain is of some importance in robot design. Finally, the domain includes interesting aspects of reasoning about shape and mathematical equations.

2 Constraint Satisfaction.

The first stage in Prompt is similar to existing design systems. Hierarchical decompositions and the atomic components are stored in structures called Prototypes [Adda85]. Prototypes are similar to the frame-like representation structures used in other systems [Srir85, Mitc85]. A component prototype contains the constraints on its interactions with other components. The prototype also contains the behavior of the component stored as a function of predetermined parameters. The difference between our prototypes and those in existing design systems is that our prototypes contain the structural description of the component; this knowledge helps Prompt analyze the behavior of components from the underlying principles. Figure 1 shows a sample prototype of a solid cylindrical beam.



Structure:
(solid-cylinder rod)
(radius rod R)
(length rod L)

$$T_s = M_t/\phi = (\pi/2)GR^4/L \tag{1.1}$$

$$W = \pi R^2 LD \tag{1.2}$$

Here

T_s is the torsional stiffness of the rod.
M_t is the external torque applied to the rod.
ϕ is the angle by which the rod twists.
L is the length of the rod.
R is the radius of the rod.
W is the weight of the rod.
G is the shear modulus of the material of the rod.
D is the density of the material of the rod.

Figure 1. The prototype for a solid cylindrical rod. The prototype has the following parameters: the radius R, the length L, and the material of which the rod is constructed. The last parameter determines the density D and the Shear Modulus G. The behavior of the prototype as a function of these parameters is described by Eqns. 1.1 and 1.2.

In our example, Prompt is told to find a beam that satisfies the following constraints:

$$\begin{aligned}&\text{Weight } W \le w_1\\ &\text{Torsional Stiffness } T_s \ge T_1\\ &\text{Length} = L_1\end{aligned} \tag{2}$$

Prompt starts with the beam from Figure 1. Prompt finds that there are two parameters that can be altered to bring the prototype into conformity with the constraints from Eqn 2.

1. Varying the material of which the beam is constructed. This alters G and D to help satisfy the constraints on both T_s and W. Prompt can try to choose a system with a high G/D.

2. Varying R. Increasing R increases both T_s and W

It is easy to conceive of a situation where the material with the highest G/D does not satisfy the constraints on T_s and W for any R. Hence the system is unable to satisfy the constraints for any combination of the parameters and the system begins the analysis stage.

3 Analysis in a Graph of Models

Prompt has a powerful analysis component that is capable of analyzing prototypes from first principles. The large amount of physics knowledge required for reasoning from first principles is stored in a Graph of models [Adda87]. A model describes the behavior of the system under certain explicit assumptions. The models form the nodes in a graph. An edge in graph represents a set of assumptions that must be added or relaxed to go between adjacent models. Problem solving in a given domain is now reduced to iterating between search for the appropriate model and search for the appropriate knowledge within the model. Searching for the appropriate model is simplified by explicitly representing assumptions and reasoning about conflicts with respect to assumptions. Searching within a model is simplified by the relatively small size of the different models. The Graph of Models structure, and its advantages are described in detail in [Adda87, Penb87] Each domain has a unique Graph of Models structure.

Prompt will include the following types of analysis of structures:

1. bending loads.
2. torsional loads.
3. vibrational loads.
4. buckling.
5. distributed loads.

Each of these forms of analysis is a domain and has its own Graph of Models. Figure 2 is a Graph of Models for beams under torsional loads [Cran72]. It is to be noted that additional models can be very easily added to this structure. Given a simple structure that lends itself to analytical solution, Prompt obtains sets of equations that describe the behavior of the system. More commonly, the structure or its loading is complex, and it may not be possible to obtain closed form solutions. In such cases Prompt will have to use numerical techniques like finite-element analysis.[Desa72] to obtain the numerical distributions of the relevant variables, e.g. stress, in the various parts of the structure. In our simple case Prompt generates the following equations to describe the behaviour of the rod in Figure 1 under the assumptions of Model 7.

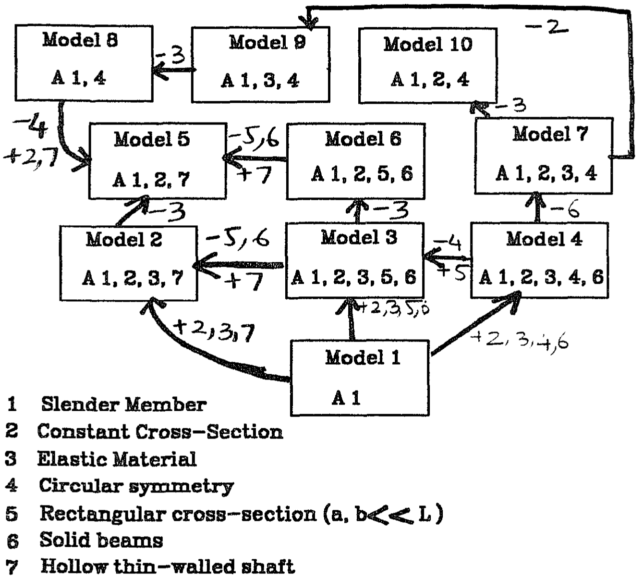


Figure 2. The Graph of models for beams under torsional loads. The assumptions are listed under the model name. The equations describing the behaviour of a beam described by Model 7 are listed below (3.1 - 3.6). The equations describing the rest of the models may be found in [Cran72].

$$\tau_{\theta z} = G\gamma_{\theta z} \tag{3.1}$$

$$\gamma_{\theta z} = (d\phi/dz)r = (\phi/L)r \tag{3.2}$$

$$df = \tau_{\theta z} dA \tag{3.3}$$

$$dM = rdf \tag{3.4}$$

$$M_t/\phi = (1/L)\int^{A} rGrdA \tag{3.5}$$

$$W = \int\int D\,dA\,dz \tag{3.6}$$

Here

$\tau_{\theta z}$ is the shear stress.
$\gamma_{\theta z}$ is the shear strain.
$d\phi$ is the angle between two cross-sections a distance dz apart.
r is the distance from the center of an area dA
df is the force exerted by the area dA
dM is the torque exerted by the area dA

Although in this case the results of the analysis are valid, Prompt may find that its analysis invalidates the initial selection of assumptions. For example, the analysis may show that the stresses caused by an external torque are greater than the Yield Stress [Cran72] of the material. Also, the modifications Prompt makes to prototypes may invalidate the current model. For example, the assumption of a uniform cross-section will be violated if Prompt tapers the cylindrical rod. In such situations, Prompt can reason about the prototype with respect to the assumptions in order to find the appropriate model.[Adda87, Penb87]
Note that, in general, analysis is very expensive because it requires reasoning through very large domains and prototypes can typically be analyzed along many dimensions. In Prompt, as we shall shortly see, analysis is controlled by Modification Operators.

4 Modification Operators.

Modifying the prototype requires reasoning about the results of the analysis in order to determine the appropriate modification. In our case, modifying the rod requires the system to realize that the important equations are 3.5 and 3.6 because they describe the stiffness and the mass of the beam in terms of its underlying structure. Then, examining these equations shows that the contribution of a given area dA to stiffness varies proportionally to r^2 and the contribution of the same area to mass is invariant with respect to r. This leads to the final conclusion that moving mass from areas of low r to areas of high r increases the stiffness of the beam but does not affect the mass of the beam.

The difficulty of directly implementing such a scheme is that it requires very complex and sophisticated reasoning that can explode combinatorially. Fortunately, of the many different types of changes that may be derived from first principles analysis, a small number find frequent use in meeting design constraints. These changes are notable in that they can be precompiled in the form of heuristics. We call these packaged heuristics Modification Operators. Our example above illustrates the Mass Redistribution Operator (MRO). The MRO captures the heuristic "Move mass from areas bearing a low load to areas bearing a high load". Modification operators consist of three parts. The first part is a set of conditions under which the operator is likely to be useful, the second part consists of heuristics that direct the analysis of the prototypes, and the third part consists of heuristics that direct the changes that are to be made to the prototype. The three parts of the MRO can be summarized as:

- Use in case of conflict between strength and mass.
- Derive stress distribution and mass distribution.
- Move mass from areas bearing a low load to areas bearing a high load.

Precompiling these changes results in a level of reasoning that forms a bridge between the purely parameterized level of component descriptions, and the very general, but inefficient, level of first principles analysis. Although Modification operators may be regarded as parameterized operators, they are really more complex. Application of these Modification operators requires the ability to derive the behavior of the component from its structure; quite different from the simple constraint propagation normally used for choosing parameters.

Apart from efficiency, storing modification operators also allows Prompt to use operators that are non-trivial to derive from the underlying analysis. An example of such an operator is the Shape Modification operator below.

Each design space has its own set of Modification Operators. We describe below a set of operators that we have identified for the beam and structural design problem. These are well known to Civil and Mechanical engineers.

1. Redistribution of mass. There are two aspects to this operator.
    a. Removal of mass. Mass can be removed from regions bearing a low load. This decreases the weight of the structure without affecting its load bearing capacity adversely.

    b. Addition of mass. Mass can be added to a beam in regions where it would bear a high load. This increases the load-bearing capacity of the structure without increasing its weight adversely. These two operators acting in conjunction can be used to decrease the weight of a structure without affecting its load bearing capability.

2. Changing the material distribution. If an element is under high stress a high strength material can be used to bolster the structure. If certain members are under low stress a low strength material can be used to build them. Changing the material alters the density, the shear modulus and the elastic modulus. The load bearing characteristics can be altered without affecting the volume of the beam. Different load mixes can be handled with composite materials.

3. Changes in shape. Changes in shape are used to remove areas of high stress in a beam, to change the connectivity of regions and to prevent buckling. An example of the operators use is given in Figure 3. Here the corners are rounded to decrease the high stress that occurs at corners under a torsional load. This is an example of an operator that is not easily derivable from the equations that describe the behaviors of structures.

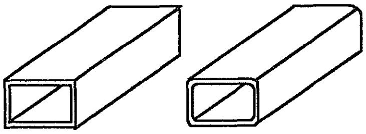


Figure 3. Rounding a square to reduce stress in the corners.

4. Addition or removal of elements. Elements can be added to a structure at points of high deflection. They decrease the deflection and the stresses in the rest of the structure. Elements can also be added to make a structure fail-safe. Elements that are under low stress can be removed totally. Figure 4 gives an example of the use of this operator.

Figure 4. Addition of a beam at points of high deflection in a structure.

Modification Operators thus constitute the intermediate level of our design system. They are derivable from the underlying principles but are precompiled and stored for of efficiency. The changes that they can effect are more general than those that can be effected by parameters. At the same time they are more expensive to apply.

5. Our Example Revisited.

Let us now see how Modification Operators can be applied to alter the prototype of Figure 1 to bring it into conformity with the constraints in Eqn. 2.

Analyzing the beam under a twisting moment M, Prompt obtains the stress distribution across the cross section of the beam. A graphical version of the distribution is shown in Figure 5. Matching its goals to the intended effects of the Modification Operators, Prompt chooses to apply Modification Operator #1 because one of the effects of this operator is reducing mass without significantly affecting the beam's load bearing capacity. The heuristic associated with the operator guides Prompt to remove regions bearing a low load and add mass in regions where it would bear a high load.

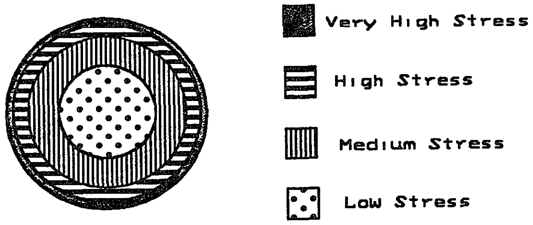


Figure 5. The loading diagram for a cylindrical beam under a torsional load.

Eqn 3.4 shows that the stress distribution is radially symmetric around the axis of the beam, and, further, stress increases with radius. Prompt therefore chooses to remove a cylindrical shape of radius R_i from the center of the beam. By going through a couple of iterations of analysis and choosing R_i and R_o, it selects values that satisfy the weight constraint and the stiffness constraint. It thus comes up with the design shown in Figure 6. This can be stored as a prototype to be used in future design.

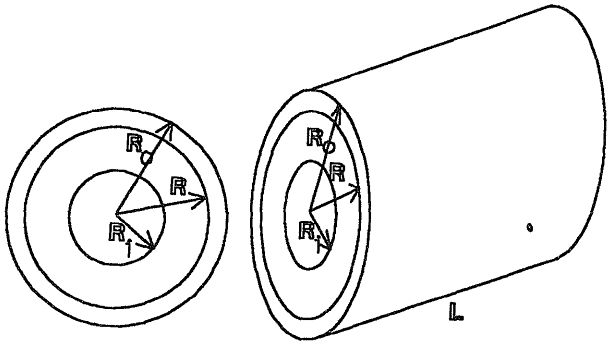


Figure 6. Modification of the prototype to meet weight and torsional stiffness constraints.

If the user now decides to add an additional constraint, Bending Stiffness $\geq B_s$, Prompt needs to modify the prototype again. Prompt analyzes the beam under both torsional and bending loads. The results are graphically depicted in Figure 7. Here Prompt has the goal of improving the bending stiffness in one direction. The shape Modification Operator has this as one of its intended effects. The heuristic associated with the operator guides Prompt to elongate the shape in the direction of the bending load. It therefore decides to use an ellipse that has its major axis in the direction of the bending load. The result is shown in Figure 8.

6 *Current status of work*

The envisioning apparatus within Prompt has been implemented for the domains of Dynamics, Kinematics, Statics and Fluid Dynamics. We are at present extending it to do stress analysis. The design system built on top of this is not complete at this point.

7 *Comparison with other work.*

Current design systems do not attempt to analyze designs at more than one level. Expert human designers display this capability; they use handbooks to get standard parts, but they are capable of reanalyzing the parts to change them if the need arises. This ability is essential for a system to be able to do more than routine design.

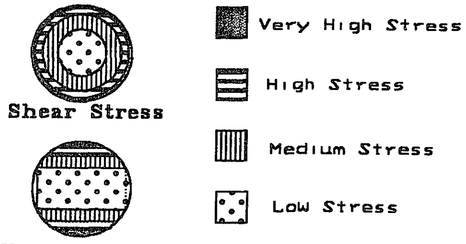


Figure 7. Loading diagram under torsional and bending loads.

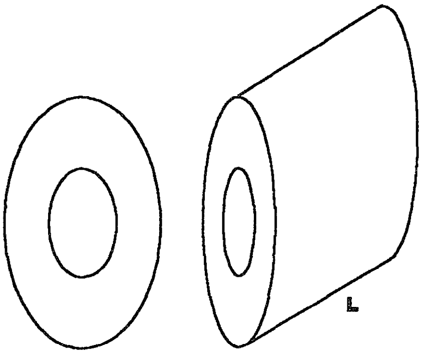


Figure 8. Modification of the beam to handle bending stress. Prompt uses the Shape Modification Operator to increase the bending stiffness while not compromising the torsional stiffness.

It has been recognized that systems that perform the task of diagnosis should have more than surface rules. Chandrasekaran [Chan83] argues that deep knowledge is indispensable to a system performing diagnostic problem solving. Randall Davis [Davi85] describes a system that reasons from first principles, i.e., using knowledge of structure and behavior. The system has been implemented in the domain of troubleshooting digital electronic circuits. More recently Iwasaki [Iwas86] describes Acord, a model-based diagnostic program that reasons about physical device behavior in the domain of a coal power plant. However this is the first attempt at using such deep knowledge that can be be derived from the structure of the device, to bring about changes that are not merely parameterized.

Dominic [Dixo86] embodies the idea that for physical systems, analysis is very important and that an iterative analysis-modify cycle is the correct way to approach design in these fields. However, the system is capable of only doing routine design; it can only vary the parameters of the prototype to bring it into conformity with the constraints.

All-Rise [Srir85] is a knowledge-based expert system for preliminary structural design. Problem solving in All-Rise involves generating a solution tree of alternatives by processing through a static knowledge hierarchy. It does not attempt to modify the prototypes that are available in the library.

Pride [Mitt86b, Mitt86a] is a system built to design paper transports in copiers. It uses data-dependency backtracking to do hierarchical refinement. We have not addressed this issue in our paper.

Edison [Dyer86]is a system designed to create a model for experimenting computationally with the processes of invention, analogy and naive mechanical device representation. The paper presents a theory of invention based on an episodic memory-based understanding of device functionality, memory generalization, analogical reasoning and symbolic representation. Edison has three general strategies for creating novel devices: generalization, analogy and mutation. However Edison pays little attention to the problem of finding the behavior of a device based on its structure. Also the mutation heuristics which the authors mention are not elaborated on. Modification operators are firmly based in the physics of the domain we are considering.

Using the Graph of models approach for representing knowledge, Prompt is able to analyze a prototype based on first principles and derive its behavior from its structure. It uses Modification Operators as a method of making changes to the prototype based on this analysis. By providing a means to alter prototypes that exist in the library, Prompt can handle situations where parameter variation alone will not suffice to handle constraints. No library, however complete can hold all possible changes to prototypes. By using Modification Operators, and when this fails, by discovering changes from first principles, Prompt is able to reach more points in the design space.

8 Conclusions.

This paper introduces Modification Operators as an intermediate level in reasoning about satisfying design constraints. Modification Operators lie between simple constraint propagation and first principles analysis in power and efficiency. We showed, through an example, that constraint propagation can fail to satisfy given design specifications. We also showed that satisfying these specifications requires reasoning from first principles about the behavior of existing prototypes. Modification Operators are compiled versions of some of the changes derivable from such reasoning. Applying these operators requires some amount of first principles analysis to determine their applicability.

The designs resulting from first principles reasoning, and from the application of Modification Operators, are novel in that they do not exist within the system library. Hence **Prompt** is an approach for moving towards systems that can produce "creative" designs for novel situations.

Bibliography

[Adda85] Addanki, Sanjaya and Davis, Ernest S.
A Representation for Complex Domains.
Proceedings of the Ninth International Joint Conference on Artificial Intelligence, 1, August 1985.

[Adda87] Addanki, Sanjaya, Penberthy, Scott, and Davis, Ernest.
Envisioning as Problem Solving: An Application of Prompt..
IBM Research Report, 1987.

[Chan83] Chandrasekaran, B. and Mittal, Sanjay.
Deep Versus Compiled Knowledge Approaches to Diagnostic Problem-Solving.
IEEE Journal of Man-Machine Studies, (19):425-436, 1983.

[Cran72] Crandall, S. H., Dahl, N. C., and Lardner, T. J.
An Introduction to the Mechanics of Solids.
McGraw Hill Book Company, 1972.

[Davi85] Davis, Randall.
Diagnostic Reasoning Based on Structure and Behavior, pages 347-411.
in Bobrow, Daniel G., *Qualitative Reasoning about Physical Systems.* The MIT Press, 1985.

[Desa72] Desai, Chandrakant S. and Abel, John. F.
Introduction to the finite element method.
Van Nostrand Reinhold, 1972.

[Dixo86] Dixon, John R.
Artificial Intelligence and Design: A Mechanical Engineering View.
Proceedings of the Fifth National Conference on Artificial Intelligence, 2, July 1986.

[Dyer86] Dyer, Michael G., Flowers, Margot, and Hodges, Jack.
Applications of Artificial Intelligence in Engineering Problems, chapter Edison: An Engineering Design Invention System Operating Naively, pages 327-341.
Springer-Verlag, 1986.

[Iwas86] Iwasaki, Yumi.
Model-Based Reasoning of Device Behaviour with Causal Ordering.
CMU CS Tech Report, 1986.
Thesis Proposal..

[Joha85] de Kleer, Johan and Brown, John Seely.
A Qualitative Physics based on Confluences, pages 7-84.
in Bobrow, Daniel G., *Qualitative Reasoning about Physical Systems.* The MIT Press, 1985.

[Mitc85] Mitchell, Tom M., Steinberg, Louis I., and Shulman, Jeffrey S.
A Knowledge-based Approach to Design.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 7(5), September 1985.

[Mitt86b] Mittal, Sanjay and Araya, Augustin.
A Knowledge-based Framework for Design.
Proceedings of the Fifth National Conference on Artificial Intelligence, 2, July 1986.

[Mitt86a] Mittal, S., Dym, C. M., and Morjaria, M.
PRIDE: An Expert System for the Design of Paper Handling Systems.
Computer, July 1986.

[Penb87] J. Scott Penberthy.
Incremental Analysis in the Graph of Models: A First Step Towards Analysis in the Plumbers World..
Massachusetts Institute of TechnologyCambridge, Massachusetts. 1987.
S.M. Thesis.

[Srir85] Sriram, Duvvuru.
Knowledge-Based Approaches for Structural Design.
PhD thesis, Carnegie Mellon University, Civil Engineering Department, Pittsburgh. 1985.