Coolsys: A Cooling Systems Design Assistant

Patricia G. Friel
Richard J. Mayer
Jeffery C. Lockledge

Knowledge Based Systems Laboratory
Department of Industrial Engineering
Texas A&M University
College Station TX 77840

Abstract

The Cooling System Design Assistant (Coolsys), developed by the Knowledge Based Systems Laboratory at Texas A&M University for Chrysler Motors Corporation, is an integrated set of tools for engineering design support in the automobile cooling systems domain. In its primary mode of operation, the system models the reasoning process of an engineer as he develops design specifications for engine box cooling systems. The reasoning model has been effectively implemented using a situation specification technique that operates in the context of a history of design experimentation. Coolsys incorporates an existing engineering analysis model used to predict the performance of a proposed cooling system design. The incorporation of this program, called the Thermal and Airflow Model, poses problems of symbolic / numeric computing that have been addressed in Coolsys.

1 Introduction

Engineering design in the automotive industry is subject to two prominent characterizations:

1. Engineering design is a highly distributed activity with complex, time consuming mechanisms for managing the distribution of requirements and assimilation of design components.
2. Engineering design is highly prototype driven. In most cases, there are no deterministic methods for determining the adequacy of a design. Hence iterative prototype construction and evaluation is the standard method for producing a final design.

Both of these characteristics, while unavoidable given the current state of the art, contribute to long design cycle times - typically a three year span. Not surprisingly, the reduction of design cycle time is a major management objective in automotive engineering. The production of better initial designs (before the construction of physical prototypes begins) is one way of reducing time in design. Another point of possible speedup is in rapid reevaluation of designs in response to engineering change notices.

A second significant management objective typically addresses uniformity in engineering methods since it is expected that a higher product quality and a more consistent quality will result.

The application described herein addresses both of these objectives.

2 A Cooling System Design Assistant

The Cooling System Design Assistant (Coolsys), developed by the Knowledge Based Systems Laboratory at Texas A&M University for Chrysler Motors Corporation, is an integrated set of tools for engineering design support in the cooling systems domain. The system functions within the prototype and evaluate paradigm for engineering design, the prototypes constructed being computer models of vehicle component functionality. Coolsys incorporates three modes of operation, known as "expert mode", "sensitivity analysis mode" and "manual mode" which are implemented, not as separate programs, but as an integrated set of tools that the engineer may pick up and put down almost at will while using the system. The expert mode of operation was the primary focus of this work; however, the inclusion of the other two modes reflects a recognition that effective automated design support must not restrict an engineer to those tools deemed "intelligent," but should provide broad support for the design process.

In expert mode, the system generates design specifications for engine box cooling systems given a description of the vehicle from the cooling systems point of view, i.e. a description of the related subsystems such as engine, air conditioning system, transmission, etc.. Test conditions (e.g. speed and ambient temperature), and certain technical or administrative constraints on components are also input to the reasoner. The reasoner then generates as many adequate design solutions as it can given the heuristic capabilities that it has. Design proposals are generated using a combination of general domain knowledge and of knowledge specific to the problem case at hand. As each design proposal is generated, it is evaluated using a Fortran engineering analysis program known as the Thermal and Airflow Model[1] that models the performance of a cooling system given a cooling system description and a description of the other subsystems that affect the performance demands on the cooling system. The Thermal and Airflow Model returns a data set of performance indicators that the reasoner uses in deciding what modifications to make to a proposed design. The iterative process of redesign and test is complicated by the necessity of finding a design solution that is satisfactory under multiple test conditions that tend to work against each other; for example, added shroud will increase airflow at idle (a positive effect), but may increase coolant temperatures at high speeds (a negative effect). Thus the reasoner must keep track of the design configurations and accompanying test results that it has previously tried in order to make tradeoff decisions between these conflicting goals. The system frequently finds multiple solutions, and occasionally finds none.

In sensitivity analysis mode, Coolsys gives the engineer a tool for experimentation. In this mode, the engineer may specify repeated runs of the Thermal and Airflow model varying some parameter (either of the cooling system or of some other vehicle system) over a range. Thus he might, for instance, study the effects of different body styles on cooling system demands. Or he may verify that a design found satisfactory by the expert mode is in fact a stable design, i.e. small changes in a design feature do not produce large changes in performance.

In manual mode, the Thermal and Airflow Model may be run as a stand alone program.

---

[1] The Thermal and Airflow Model was written some years ago at Chrysler by Dr. Roger C. Shulze of Chrysler Cooling Systems.

This is sometimes desirable if an engineer wants to make a quick check on some proposed vehicle design change. Manual mode is also useful in situations in which an engineer wants to experiment with a very unusual design feature: something outside the purview of the expert system's knowledge, but which may be simulated using the Thermal and Airflow Model. In a future version, the manual mode may provide the base structure for a knowledge acquisition tool which would be used to capture design rationale as new product components or technologies are incorporated into cooling systems engineering practice.

From a knowledge based system view, Coolsys addresses two basic problems. The first is the problem of modeling the reasoning process used by an expert design engineer as he iteratively proposes a design and evaluates it against acceptance criteria. The reasoning process has been effectively modeled using a situation specification technique that operates in the context of a history of design experimentation. The second is the problem of integrating symbolic and numeric computing components. Major issues involved the understanding and handling by the expert system of errors arising in the analysis program and techniques for allowing the expert system to understand the assumptions underlying the analysis model.

A generic functional architecture for designer systems of this type was developed and Coolsys was implemented according to this architecture. The basic functional components are illustrated in Figure 1; they are explained in detail in [2].

Coolsys is written in a combination of three languages: ART for the reasoning model, Lisp for various procedural components, and Fortran for the engineering analysis model. The rationale behind the use of the ART and Lisp languages was simply that each was used for what it is good for; Fortran was used for the cooling system performance model because the program will be maintained by mechanical engineers versed in Fortran.

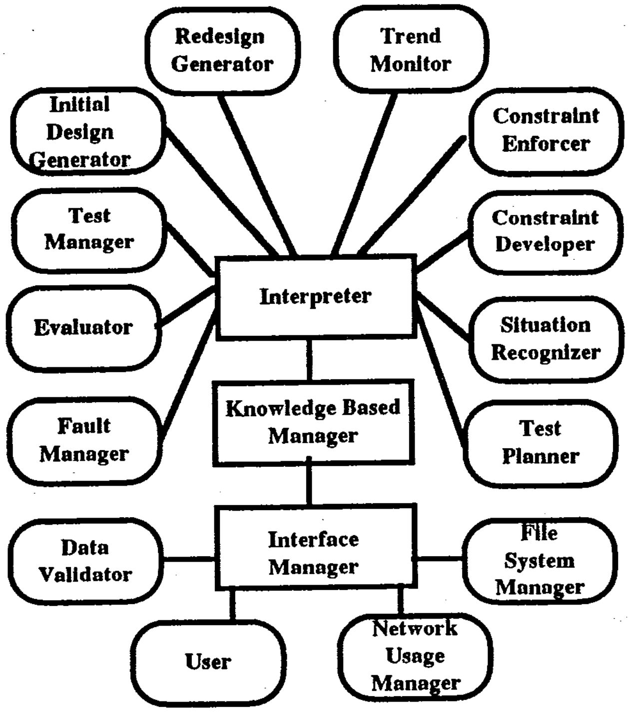

Figure 1: Generic Functional Components

3 The Design Reasoning Model

The primary innovation in the Coolsys work is the situation based model of engineering design reasoning implemented in Coolsys's reasoning component. Situation based reasoning is described briefly as a process of using a current event viewed in the context of an ancestral chain of past situations and against certain background conditions to determine the current situation. In the design context, the current event is comprised of a paired design decision and design evaluation. In an implementation of the model, each current event must be represented and must be accessible by the reasoner separately, but the situation representation includes the current event, a chain of past events, and some number of underlying background descriptions. Background descriptions are layered so that they may be viewed as a coherent unit but may still be "remembered" separately.

The recognition of situations is basic to the way the Coolsys reasoner is structured. That this was the way the system should be built was not immediately obvious when first interviewing the expert[2]. The comments one first heard were statements to the effect that "this should be done before that," implying some prioritizing scheme. But an engineer will immediately abandon his own stated prioritizing scheme if a situation arises that *he knows doesn't fit the general case.* Apparently the priorities do represent a "compiled" or abstracted knowledge that certain actions are generally more effective (either technically effective or cost effective) than others. An experienced design engineer will readily produce a flow chart representation of how he makes decisions. However, our experience was that specific design problem cases *never* fit the abstraction. It is the completely specified situations that over time have given rise to the abstraction that must be captured in order to effectively model the reasoning process of the engineer.

Having understood that the objective is the capture of design situations, the problem becomes one of designing a knowledge base structure that will enable that capture. Two particular characteristics of the design process under study affected the eventual structuring:

1. The utilization of experimentation history. Design situations and resultant decisions are frequently only determinable in the context of a history of design experimentation. As an example, suppose that the analysis routines of the Thermal and Airflow Model have been run on a cooling system design with a specified fin density of 20 for the radiator. The analysis predicts a coolant temperature of 260 degrees entering the radiator. The goal temperature is 250 degrees. Given this situation alone with no background information, the apparently correct decision would be to increase the fin density in order to provide more copper surface to dissipate heat from the radiator. However, if this situation is played against the background of a previous situation where a fin density of 18 yielded a coolant temperature of 255 degrees, and against the background of the known parabolic shape of the general fin density vs. temperature curve, the conclusion may be reached that the right thing to do is to decrease fin density in order to allow more air to pass through the radiator. Thus the maintenance of the history of design situations that form the background for the current one is necessary in order to intelligently assess the current situation.

2. The "one situation / many design options" phenomenon. The usual case is that in any design situation, multiple design changes could make sense. The engineer may want to try more than one option, possibly comparing the results. Thus it is desirable that multiple design situation histories be maintained in parallel.

In order to accommodate these characteristics, the knowledge base was structured into a hierarchically ordered set of situation representations. This allowed the "one design / many design options" characteristic to be modeled. The structuring also permitted rules to be written such that pattern matches on a current situation could be evaluated in the context of previously existing situations. ART

---

[2] Gary M. Smith of Chrysler Cooling Systems was the expert for this project.

Viewpoints[3] used with the ART production system paradigm provided the underlying language structure to implement the situation based reasoning model.

Effectively, Coolsys is able to learn the special behavioral characteristics of the vehicle being designed by viewing the design development and testing history. It learns what design options advance (or do not advance) the design goals and uses this knowledge to dynamically refine constraints on design parameters. That is, the technical constraints imposed by the user (or by default) at the Coolsys program initiation can be modified by the program itself as it learns the behavior of a specific vehicle. Tightened constraints act to narrow the space of probable design solutions and hasten convergence to an acceptable solution.

This model is a distinct departure from previously proposed models of design which have either viewed the design process as a quasi-logical process [3][5] or as constraint propagation, search, hierarchical decomposition, etc. (see [1] [6] [7] for some other views). Much has been written about how to model "design", or even "mechanical design." Our experience with working in the engineering design arena, however, leads us to believe that most of the models that have been proposed suffer from over-generality. The design process is probably not amenable to a single definition because many different reasoning processes take place in generating a design. The situation based model that we have implemented is only one of these. Certain forms of qualitative reasoning and curve based reasoning (which we are investigating in the Knowledge Based Systems Laboratory) evidently are also prominent in engineering design.

4 The Integration of Symbolic / Numeric Components

The work also explored the problems inherent in a tight integration among a design reasoner, engineering analysis models, and test databases, essentially the symbolic / numeric computing problem (see [4] for a background discussion). Two faces of the problem were evident in the Coolsys work. The first is the practical problem of the integration of languages primarily designed for either symbolic or numeric computing, but not both. The difficulties are illustrated in Coolsys, where the integration of ART and Lisp is natural and unobtrusive; but the incorporation of the Fortran program is workable, but far from elegant. The basic problem is that the integration of languages with the two orientations has not been recognized by the designers of "expert system languages" as a major design issue. In the arena of engineering design, however, the incorporation of existing analysis programs, which will most likely be written in Fortran, can be expected to be the norm.

The more serious integration difficulty arises when the reasoner in the symbolic world needs to understand what is going on in the numeric routines. It is frequently not sufficient for the reasoner to treat an engineering analysis program as simply a black box that returns results, since results may for a variety of reasons be incorrect or reflect incorrect assumptions. Additionally, error conditions may arise within the analytic code, and these are not normally directly available to the reasoner. Certain error conditions in the Thermal and Airflow Model, for instance, will clue the knowledgeable engineer to certain input data problems, e.g. too much trailer weight or an incorrect tire rolling resistance, but these associations are not obvious to the uninitiated. The expert system, like the knowleageble engineer, should be able to make these associations, re-

---

[3] ART (Automated Reasoning Tool) and Viewpoints are registered trademarks of Inference Corporation.

port them to the user, and recover gracefully. An innovation in Coolsys was the employment of a context sensitive approach to the identification of and recovery from such conditions. The Fault Monitor component in Coolsys watches for problems reported from the Thermal and Airflow Model, and makes suggestions to the user of possible causes of error conditions. Because of the layering of background conditions and the maintenance of the design history, the system is able to return the knowledge base to a state from which recovery options are possible. The structure of the knowledge base accommodates recovery well; however, the discovery of problem conditions in the analytic code still depends on reporting by the analytic code itself. The problem of *recognizing* what is going on in the analytic code is inherent to the symbolic / numeric computing dichotomy.

5 Criteria for Success

Our criteria for a successfully deployed application include the following:

1. The application should be judged cost effective by the organization. Benefits may be assessed in a number of ways such as direct dollar savings, reduced training time, increased effectivity of personnel using the system, and better first time designs.
2. The application should be smoothly integrated with other systems that interface with the application, e.g. information systems, data base systems, or predictive analysis programs.
3. The application's users should find the interface natural to work with, that is, the interface should make the system fit unobtrusively into the user's work style.
4. The application should be maintainable and extensible by the using organization.
5. The application should be extensible as new computer technology emerges or as new problems in the domain arise.
6. Users in the domain regularly use the system as an integral part of their work process. User acceptance is often indicated by a steady stream of requests for modifications / enhancements to the system.

6 Payoff to Organization

The benefits to the organization in the case of Coolsys are several:

1. Reduction in time to generate an initial design. Coolsys is able to generate in a few minutes design specifications that would have taken as long as several days in the past.
2. The rapid generation of multiple acceptable designs. Coolsys frequently is able to generate a number of satisfactory designs. This enables an engineer to choose a best design from several where, in the past, time constraints prevented him from developing more than one acceptable design.
3. Solidification of the engineering method. One of the objectives of the Cooling Systems Department was to better understand how, as a group, they performed the design task. The knowledge engineering excercise served to help formalize engineering methods used in the department, thus addressing the management objective of uniform engineering methods.
4. Technology capture. The organization has gained experience in how to select future expert systems applications and in how to manage their development and deployment.

5. Identification of reasoning patterns commonly used in engineering design. These concepts will expedite the identification and development of future systems in engineering.

7 Project History

The development of Coolsys up to its initial deployment was carried out as a two phase project, essentially a problem selection and prototyping phase and a development phase. The first phase was a six month (8 man month) project, roughly half of which was devoted to the problem selection process. The second phase was a one year (18 man month) project that focussed on working with an expert designer to develop the prototype reasoning module and on working with potential system users to develop the user interface. The system has been in use in Chrysler Cooling Systems since 1987, and all cooling systems for new vehicles (except for trucks) are being designed with the aid of the system. Coolsys is now being maintained and expanded by Chrysler, with the engineer who served as the expert now in charge of maintenance of the system. In a sense, the system is still being deployed. At the time of its initial deployment, only one Symbolics 3645 was available for use. However, four MacIvory systems are currently in the purchasing processing, and these will be distributed at more convenient locations for access by the engineers. Also, the user interface is being revised in response to user requests. It is expected that this evolutionary process will continue as new capabilities are added to the system and as new technology is absorbed.

Acknowledgements

The authors would like to acknowledge the work of the entire project team, which included, in addition to the authors, Ms. M. Sue Wells and Mr. Paul Squitterri of Texas A&M University and Mr. Gary Smith and Dr. Roger Shulze of Chrysler Motors Corp.

References

[1] Brown, D.C. and Chandrasekaran, B., Knowledge and control for a mechanical design expert system, *Computer* (July 1986) 92-100.

[2] Friel, Patricia G., *Modeling Design Reasoning in Automotive Engineering*, Dissertation, Texas A&M University, 1988.

[3] Hillier, B., Musgrove, J. and Sullivan, P., Knowledge and design, in: N. Cross (Ed.), *Developments in Design Methodology*, (John Wiley & Sons, Ltd., 1984) 250-258.

[4] Kitzmiller, C.T. and Kowalik, J.S., Coupling symbolic and numeric computing in knowledge-based systems, *AI Magazine* **8**(2) (Summer 1987) 85-90.

[5] March, L., The logic of design, In: N. Cross (Ed.), *Developments in Design Methodology*, (John Wiley & sons, Ltd., 1984) 265-272.

[6] Mittal, S. and Araya, A., A knowledge-based framework for design, *Proceedings AAAI-86* **2**(1986) 856-865.

[7] Steinberg, L.I., Design as refinement plus constraint propagation: the VEXED experience, *Proceedings AAAI-87* **2**(1987) 830-835.