Equation and Knowledge Based Generative Gear Design System

Karon Barber
Matthew Orel
General Motors Corporation
NAO Metal Fabrication Division and Manufacturing Center
Building 1-9 A/MD-16
P.O. Box 9040
30300 Mound Road
Warren, MI 48090

Abstract

GMGear is an equation and knowledge based system that uses an iterative process to generate multiple possible solutions for a gear set design. It evaluates and recommends the best gear set to meet performance requirements, geometric constraints and manufacturing considerations. Performance requirements include input torque and speed, gear ratio, weight, noise and duty cycle life. Geometry constraints are the envelope into which the gear set must fit, as defined by axial length and vertical size. Manufacturing considerations include tolerances, types of materials and the use of existing cutters, carriers and mating gears.

GMGear takes advantage of Object-Oriented technology for implementation of domain information, and uses a sophisticated back chaining mechanism to apply equations towards reaching design solutions. We represented all gear variables and equations, as well as all physical components, methodology components and design results as objects. We designed and implemented this system so that the domain experts are responsible for maintaining the domain knowledge.

Introduction

When General Motors established the Gear Center to improve GM's gear design processes, the Gear Center initiated an experimental effort with the Artificial Intelligence Knowledge Engineering (KE) group, which had successfully implemented several large, complex, Expert System based, manufacturing systems (references 1-7). This joint effort led to development of GMGear, a generative gear set design system. This project represents a major investment with significant person-years of effort, and addresses some of the following business needs:

* To ensure high quality gear designs. Gear sets are integral to power trains, a core business of General Motors. GM has several gear design groups, as shown in Table 1 (Anderson, Barber, & Kienzle 1991). Having high quality designs is central to GM's competitiveness. This must continue although gear experts with decades of experience are retiring.

* To speed up the gear design process. The reduction in design time for a gear set provided by GMGear (up to 95% for the design of an entire gear train) is a significant advantage.

* To allow domain experts to focus more on future concepts and perfection of gear designs. The design time reduction resulting from use of GMGear will contribute to this.

* To capture knowledge from across several GM divisions. The production release successfully designs gear sets for manual and automatic transmissions for automobiles, light trucks, off-road vehicles, tanks and other military vehicle transmissions. Table 2 lists the gear types and arrangements considered.

* To create a better understanding of the gear design process. Novices can use GMGear as a training tool and take advantage of its extensive explanation facilities.

* To promote commonalty of terminology use and design practices throughout General Motors. This facilitates exchange of information. Use of a common system helps ensure that different divisions uniformly apply validated design techniques.

* To catch potential issues with new designs without having to go through a long pre-production process. GMGear proved its value during system test by finding an error in a gear set about to enter production.

The Gear Designer's Function

The earliest planning for a new transmission includes considering what type of gear set arrangement to use (e.g., planetary or countershaft) and establishing the size requirements of gear sets. The primary constraints are the performance requirements for the transmission and the space available. Occasionally, an additional constraint requires that the new gear sets must use an existing gear, existing cutters, some other existing physical components, and /or existing manufacturing tooling.

Designers then refine the sizing of the gear sets, determining the physical characteristics that can meet the performance requirements. Conflicting constraints must be addressed. Geometric details are determined and performance demands evaluated.

The Objectives of GMGear

GMGear is an equation and knowledge based system that designs, evaluates and recommends gear set designs. Gear set designers with skill levels ranging from novice to expert, and whose job function ranges from future concepts designer to detailed gear designer, use GMGear. The objectives driving GMGear are based on the business needs enumerated in the introduction, and include:

* Commonization of gear design practices within General Motors. Many of our gear design groups used versions of an analysis program developed by GM Research in the late 1950s. Each site had modified the programs, so that all were different and results were difficult to compare.

* Provision of an easy-to-use tool for beginning gear designers and for automation of the design process. The analysis programs our designers used in the past required considerable experience to understand the technology and interpret the results. GMGear addresses these issues.

* Provision of a "glass box" as opposed to a "black box" view of the system. The user can inspect all GMGear knowledge, equations (figure 1), variables (figure 2), calculations, tolerances, definitions, and results. We provide a variety of techniques, from graphical displays (figure 3) to on-line help (figure 4). The user can mouse on screen items for explanations.

* Promoting use and understanding of American Gear Manufacturers Association (AGMA) standards for terminology and symbols (figure 5). Thus, when the user examines GMGear knowledge, he/she sees the same equations, with the same parameters, he/she finds in non-GM technical manuals.

* Allowing the user access to system evaluation and the ability to override system comparative design choices. This permits an experienced gear designer to push the system in a preferred direction, focus on future concept designs or improve gear design techniques.

Application Development

GMGear is a large system, using more than 1400 equations representing the accumulated knowledge of gear set designers from several divisions that produce different products. The project began in early 1990.

Knowledge acquisition sessions involved one or two knowledge engineers and an expert in gear set design or a potential end user of the system. We applied an iterative process to develop this system: knowledge acquisition, concurrent implementation, followed by refinement. We found this to be a good way to help the users define their requirements. This helped identify the needed equations, algorithms and test cases (see lessons learned). We generally tried to adhere to the following order: high level requirements, detailed requirements, high level specifications, test case development, detailed specifications, test case refinement, testing and refinement. All of these processes were concurrent with concept prototype or functional prototype development.

There were two interesting factors in our knowledge acquisition process. One, the primary expert was a mid-level designer who needed to draw on the resources of more experienced designers distributed throughout the corporation (Kienzle & Barber 1991). Two, the system needed to be able to design a broad range of gear sets manufactured by several divisions. We had to satisfy multiple customers, with different products and objectives (Marney & Barber 1992).

We delivered the alpha version of the prototype for testing to Allison Gas Turbine, Allison Transmission Division, Powertrain and the Gear Center in July 1992 and began formal release, bug reporting and enhancement request methods. At that time, the system approach to gear set design duplicated the generate and test methods in use at the corporation.

In September 1993 we released the beta version of the prototype. A more rigorous algorithmic design process to generate gear set designs had replaced the generate across ranges and test approach of the earlier release. A novice designer could use the system to design gear sets of comparable or slightly better quality than current production gear sets, and an expert designer could push the system to better designs. Using GMGear, a designer discovered interference at tight mesh in a gear set before it entered production. At that time, we began to focus on increasing domain coverage to provide full functionality for the gear types and arrangements (Table 2).

We initiated rigorous testing at the user sites and by an outside gear design consultant. The consultant is one of those retired designers previously mentioned who had 30+ years of design experience. We chose him because all the divisions involved considered him the most expert among our gear designers. After completion of testing and production hardening, GMGear entered production release in June 1994.

A joint General Motors, Electronic Data Systems and IntelliCorp team developed the system. The Gear Center provided the primary expertise to the development process. Engineers from Isuzu-Japan who had investigated tip to root interference due to shaving steps (Miao & Koga 1994) developed the initial program to graphically display this interference. Allison Transmission Division, Allison Gas Turbine, Delco Chassis, Saturn, the Electric Vehicle Program and Adam Opel provided test cases and/or testing.

Operational Functionality

GMGear Design Conditions

GMGear designs gear sets under the following conditions: under constrained, where minimal constraints exist, such as in a space claim condition; to fit into an existing housing or planet carrier; to mate with an existing gear; and for manufacture with existing cutting tools. Besides generating designs for gear sets under the preceding constraints, GMGear also determines the performance characteristics of existing gear sets. Table 2 lists the types and arrangements considered. Table 3 lists the analyses provided.

Describing the Requirements to GMGear

Describing the design case requires minimum input for a minimally constrained design case. A new design case under these conditions typically requires the following data: identification of the input and output members of the gear set; the gear ratio range; the torque and speed into the gear set; the allowable housing or carrier length and radius ranges; and for a planetary gear set, the number of planets. Expert gear designers may supply more information for minimally constrained designs, although GMGear supplies default site and application-dependent data through the use of application profiles (figure 6).

Describing the requirements for one of the highly constrained design cases may require the user to enter many parameter values (figures 6 - 9). One method of doing this is to select an existing gear set from the database and edit its input information as necessary to create a new design scenario. Alternately, a user may create an entirely new set of information. The figures in this example show how one might edit one of these dozens of parameters.

Generating Potential Designs and Selecting the "Best" Design

In the design process, geometric parameters are traded-off to achieve a good balance of performance characteristics, while meeting constraints. GMGear moves through a series of design steps, concentrating first on those parameters that have the most impact on gear set performance and geometry. At each step GMGear may iterate over multiple key parameters simultaneously, resulting in many (exceeding 1000 in some cases for minimally constrained designs) combinations of parameters. Each combination represents a potential design set. At appropriate points, GMGear applies limits and rejects design sets with critical parameters that do not meet requirements. GMGear evaluates and sorts the remaining design sets based on bending and contact stresses, noise, size and life. At the end of the process, GMGear recommends the best design set to the user. All design sets are available for review, and the user may override the system selection. This permits the user to examine and compare multiple possibilities.

Examining the Output

GMGear provides several methods to examine the results of the gear set design process, including summary reports, comparison reports and graphs. We highlight any parameter that is out of limits. This makes it more difficult for the user to overlook issues. The users also want to know the source of parameter values. We use different fonts to differentiate between calculated, default and user-specified values. The screen displays permit the user to examine parameter and/or equation descriptions. He/she can display value limits and the equation derivation sequences with values for any calculated parameter.

Output summaries include a description of the design case and important parameter values organized by category. The form of the output report matches the data sheets that gear set designers are accustomed to using. While on the screen, the user may access information about any displayed value, such as how GMGear derived a value. The user may also get printouts of all or part of the report.

GMGear can display two different design sets in a comparison report (Figure 10). Color-coding indicates the degree of difference between the respective values for a parameter. There are four colors for different degrees of difference and the threshold for each color is editable by the user. The comparison report enables the user to engage in detailed comparisons between new designs and existing gear sets, and to analyze specific differences between design set possibilities.

The user may also select and graphically display parameters across multiple generated design sets. He/she can sort and order them in various ways. This allows users to pinpoint areas of specific concern within the design process, and to examine results for those areas across many design set possibilities simultaneously. For example, if the user wants to know which gear sets gave the optimal values for bending stress, he/she could use the sort utility of the grapher to find the appropriate design set possibilities, and, if appropriate to continue processing with that design set possibility.

The Role of AI in Generative Gear Design

Backchainer

GMGear uses a sophisticated back chaining mechanism to sequence equations in order to find design solutions (Kienzle et al. 1994). The system currently uses over 1400 equations to solve approximately 1000 design variables. The domain experts at the GM Gear Center maintain the equations and variables.

Steps in *design approaches* dictate the items on which the Backchainer searches. These steps represent the methodology by which expert designers design the gear sets. Several contextual factors help govern the work of the Backchainer. Design steps may specify preferences for certain groups of equations. Some equations may apply only in certain situations, for example, when solving for a ring gear. Also, while some equations are always known to be true, there are also equations that represent the "best guesses" of domain experts, and the Backchainer must be able to distinguish between and choose amongst several potential certainty levels of equations.

The search employed by the Backchainer to find an equation to solve a given design parameter is available for inspection by GMGear's users. This capability has both helped domain experts refine the design approaches and helped end users understand how the system arrives at its conclusions.

Sequencer

GMGear's Equations Sequencer pieces equations together into sequences that may be used many times to solve for a series of design parameters (Kienzle et al. 1994). An equation sequence becomes a program that can solve a portion of the gear design case based on certain pre-existing conditions. So, when solving a design step within a gear design, GMGear will use its Backchainer to arrive at an equation sequence for the step. It will then re-use that sequence for all of the gear set design possibilities on which it is iterating.

GMGear uses some algorithms to solve large numbers of parameters. GMGear developers have used the Backchainer to derive the appropriate equation sequences for these algorithms, which we call *library sequences*. Use of these library sequences obviates the need for the Backchainer to solve the same derivation on successive gear set designs, thus greatly improving the overall efficiency of the system.

We often cache and reuse GMGear equation sequences, thus helping overall system efficiency. For example, GMGear derives only once an equation sequence for a design step that is considering many design set possibilities. Also, if the design approach is re-run with the same set of inputs (with possibly different values), the same sequences are re-used. Finally, for calculation under manufacturing tolerances, in which we adjust results to account for variations in the manufacturing process, we reuse the equation sequences used for the "nominal" results calculations without the need for re-sequencing.

System Architecture

We represent all key components of GMGear as objects. This includes not only the representation of the physical objects, but also of interface components, the equations and equation sequences, and even the design approaches. The specific methods used to activate GMGear actions are generally accessed as methods for object slots. These methods, which we wrote in Lucid Common Lisp, provide the underlying system mechanisms for using and manipulating object information.

The remainder of this section describes some of the key architectural features of GMGear and their use within the system.

Definers

We use *Definers* to define the object classes of GMGear. The definers lay out the slots, facets, methods, message passing routines and descriptions of object classes. By using definers, GMGear builds objects at system load time, rather than storing them in externally accessible object bases. This helps control access to and manipulation of the permanent (i.e., non-site-dependent) GMGear objects.

The following sections describe various object types within GMGear. We define all of these object classes through use of definers.

Variables

There are currently more than 1000 variables within GMGear. Each variable describes some item whose value we may need to compute in order to generate a gear set. Variables all have *defining levels* and *mapping restrictions,* which describe how the contexts in which they may be used. For example, the variable *D,* for diameter, has a defining level of *gear,* that is, each gear in the gear set has a diameter. The variable *d,* which is part of the variable class for *D,* has a mapping restriction of *(gear-pair smaller),* meaning it is assumed to represent the smaller gear of any particular gear pair, when used in an equation.

Depending on the current context, any given variable may have multiple mappings for which we need to solve. We call each mapping of a variable a *parameter*. In a simple planetary gear set, in which there are 3 gears, there will be 3 parameters for *D*. We strictly associate individual parameters with the design cases in which they exist, so there is no object representation for them. Rather, we store the parameters and associated values as part of the objects in their specific *parameter space*.

Equations

There are currently more than 1400 equations within GMGear. Each equation describes a particular potential solution to a particular variable. Equation objects have slots that describe the level of confidence in the equation, contextual information on the equation grouping, and mapping restrictions on variables used within the equation. Equation mapping restrictions take precedence over variable mapping restrictions -- for example, the equation $D = 2 * R$ has a mapping restriction on *D* of *(gear same),* meaning it can be used to solve for *d*, which has a mapping restriction of *(gear smaller).*

We use equation classes to describe all equations that may be used to solve for a given variable. Individual equations are members of their respective classes. We define equations as strings that GMGear converts to lisp code. We store the code within an equation slot, and the equation sequencer accesses it when appropriate.

Constraints

Constraints are special equations that we evaluate as part of pre-processing to determine whether a given set of inputs is viable. We evaluate constraints in similar manner to equations, but do not access them from the Backchainer. For a constraint to be used, all of its component variables must be present in the inputs.

Profiles

We use profiles to define sets of default values for given variables (or parameters) within GMGear. The use of profiles obviates users from the need to repeatedly specify the same set of parameter values for all design cases of a particular type, when default values may suffice for many of those parameters.

When defining a new design case, the user must select the profiles to use for that case. GMGear then installs the appropriate values from the profile as inputs to the case.

There are several different types of profiles. A *site* profile defines certain characteristics that are generally true at a given site. An *application* profile defines values for all cases defined within that application, and a *user* profile defines values that a particular user wants. User profile values take precedence over application and site profiles. There is typically no overlap between application and site profiles, though in such cases application profiles take precedence.

Physical Objects

We represent all physical objects in GMGear as objects. The physical objects include gears, cutters and planet carriers. We also represent particular aspects of the objects, such as gear-types, materials, and even parameter mappings, as objects. An object used as part of a physical gear set may provide parameter values to the gear set. For example, if a user specifies a particular gear for inclusion in a gear set, all of the values for that gear (number of teeth, face width, diameter, etc.) will become input values to the design. GMGear will not calculate them.

GMGear checks that all physical objects selected by a user for use in a design are consistent with one another. For example, if a user specifies usage of the spur gear type, he/she may not subsequently specify usage of a helical gear for that design.

Approaches

The GMGear approach objects describe the procedures by which expert designers produce world class gear sets. Each approach consists of one or more *strategies*, each of which is further divided into a series of *design steps*. Each design step defines a procedure to complete a certain portion of the gear design process. Basic types of steps, also represented as objects, include calculation, limiting and sorting steps.

Parameter Spaces

Parameter spaces are objects where all values for gear set designs are stored. Each design step has an input parameter space, which describes all values generated before the step, and produces an output space, which describes all values used by the subsequent step.

At many steps in the design process, there may be multiple gear set possibilities under consideration. In these cases, we represent each possible design at each step by its own object, which will use the appropriate input parameter space for that object, and create its own output parameter space.

GMGear Validation

We saw validation of this system as an enormous task. The equation knowledge base contains more than 1000 parameters and 1400 equations. GMGear currently creates and examines gear set designs for spur and helical gears, in either countershaft or simple planetary arrangements.

The test plan for the Beta release required that geometric and performance analyses from GMGear match those from existing General Motors analysis programs or experimental data. Our users required geometric analyses to match existing values to a minimum of four decimal places. The testers had to account for any deviations from existing General Motors analysis programs and demonstrate that differences in performance numbers were the result of improvements in methodology and accuracy.

We divided this part of the validation among five primary testers. Each had sign-off responsibility for parts of the system in his particular area of expertise. Three of these testers were not involved in the system development process. Intensive testing continued for eight months between the Beta release and prototype acceptance.

We hired an outside gear design consultant, a GM retiree, to stress test the system for four months with difficult design situations. He also assessed advice and recommendations for relaxing constraints when the system was unable to generate a design that met all the geometry and performance requirements.

Implementation

We built GMGear using IntelliCorp's Knowledge Engineering Environment (KEE) and LISP. We did initial prototyping mostly on LISP machines, because we had several available from earlier projects. During the pre-alpha phase, we shifted development to Sun Workstations. Before alpha release, GMGear developers and customers agreed to support GMGear as a program built specifically for color workstations running X-Windows. The bulk of implementation was done on Sun SparcStations. GMGear currently runs on Sun SparcStations and HP 9000 series Workstations, both of which are in widespread use at GMGear's delivery sites.

The GMGear development team designed and implemented GMGear's Backchainer and equation sequencer. The GMGear interface uses X-Common Windows. GMGear also interacts with a graphics program that the GM Gear Center developed independently.

Deployment Process

An expert from the Gear Center trains new users in the use of GMGear. Generally, a representative of each potential initial deployment site participated in the GMGear development process from the development of the initial requirements through final testing. That representative served as the system champion when we released GMGear to his/her site, and worked closely with new system users.

Application Use

GMGear is in use at four sites, with 10 additional sites planned for 1995 and beyond. They use GMGear for initial packaging by future concepts designers and for detailed gear design. Pilot use has shown that GMGear met the business needs for which it was designed. GM has lost many experienced designers to retirement. With GMGear, they can rapidly design new gear sets of equivalent or better quality than those currently in production. The system also significantly reduces the gear design process time. We have had engineers who earlier spent weeks working toward an optimum gear set design or to design all the gear sets in a great train. They found that with GMGear they could produce appropriate designs in a day or less, and often realized better designs.

Several GM divisions validated the knowledge and equations contained in GMGear. With GMGear the corporation is focusing on a common system of best design practices. The system has proven it's worth in resolving conflicting constraints in new designs and in exploring areas that our designers found difficult to pursue using the previous process. GM had earlier had been exploring a high-contact gear set for a new transmission. We used this as a test case for GMGear. It successfully designed a high-contact gear set that met geometrical and performance constraints.

GMGear also is an effective training tool for novice gear designers from several aspects. It teaches standard AGMA nomenclature, offers extensive explanation facilities and recovery advice when designs are over-constrained, and permits the use of alternative design exploration. Both novice and experienced gear designers favorably received GMGear. The General Motors gear design community uses it extensively.

Maintenance

GMGear entered the maintenance responsibility in January 1995, after we completed the pilot phase of production release. During production hardening, we devoted considerable attention to making the system more maintainable and extensible. We provided an extensive Users' Manual, Programmers' Guide and 5.5 weeks of hands-on training by one of the system developers.

Innovative Features of GMGear

GMGear automatically generates gear set designs from user provided geometric constraints and performance requirements. It explores the solution space for multiple key parameters (more than 1000 combinations for minimally constrained designs) simultaneously. It trades off geometric parameters to balance performance characteristics and to meet constraints. GMGear generates and ranks hundreds of possible designs.

GMGear uses graphics to increase the designers' understanding of the design issues. We show how variations affect dependent parameters and the overall design. Users view parameters across multiple design sets and see the effect of parameter variation on design goodness. The gear designer can use this enhanced understanding to override system intermediate design choices during the design process.

We account for cutting tool geometry and manufacturing tolerances in the design process. GMGear constrains the design to a "reasonable" cutting tool and adjusts it to account for manufacturing process variations. GMGear graphically shows the effect of the cutting tool on gear geometry.

Our customers rank highly the "glass box" aspect of GMGear. They can view the design methodology, design decisions and reasoning, value descriptions and sources, descriptions of terms and equations and equation sequences with input and intermediate values.

We offer recovery advice for over-constrained designs by identifying over-constrained parameters and suggesting which constraints should be relaxed and listing possible trade-offs.

GMGear provides a language for the gear designer that has been used extensively at the Gear Center to develop new gear design approaches and to make major modifications to earlier approaches. This language removes the requirement for a programmer to assist the designer in developing new methods. The user is also responsible for maintaining domain knowledge, which is explicitly represented, separate from the system code and is displayed using mathematical symbols and AGMA terminology.

We use a very compact data representation, with equations and variables that contain general knowledge and are used in many different situations. For example, the 1000+ variables in the system have approximately 4000 instantiations in the design process. In one typical test case using one approach, the 1400+ equations in GMGear used 4881 equation instantiations, and there are 13 design and analyze approaches.

GMGear uses a sophisticated Backchainer to generate context-specific equation sequences to solve for a series of parameters within a gear set design. Unlike a standard rule-based system, in which it may be sufficient just to find a path to the item to be solved, in GMGear it is often necessary to compute full solutions for several hundred design set possibilities simultaneously, using the same set of equations generated by the Backchainer. The back chaining/sequencing method used within GMGear is patented (Kienzle et al. 1994).

Challenges - Lessons Learned

The biggest challenge on this project was the amount of parallel development of specifications and system code. This also turned out to be one of the most rewarding aspects of this system, because it permitted the customers to refine their requirements during the iterative prototyping process we used. During this process the customers established a more rigorous and innovative method of solving for gear designs. Functional requirements, specifications detail and test cases underwent considerable refinement and increased complexity during this process. The scope expanded significantly. During the early system requirements development phase, it was estimated that there would be a maximum of 300 equations required in the system. The final system contains more than 1400 equations. Some areas of functionality are more complex than anticipated. For example, the development of the capability to design gears for both nominal conditions and conditions allowing for manufacturing tolerances is quite complex. The magnitude of development and testing was much larger than initially anticipated, exceeding our previous experiences.

Securing adequate resources for testing was a challenge. We drew on the user sites and an outside gear design consultant. We found that rigorous testing paid off. During the six month pilot production phase we received very, very few bug reports and enhancement requests.

Key Success Factors

* GMGear provides a significant design time reduction while allowing novices to design production quality gear sets.

* GMGear allows gear set designers to examine all facets of the gear designs in much greater detail than has ever been possible previously.

* GMGear permits expert gear set designers to override system choices and push the system beyond previously accepted production quality designs.

* GMGear is an easy to use tool that significantly automates the design process. Extensive on-line help is available, ranging in form from definitions of equations, parameters, limits, tolerances, etc., to graphical displays comparing alternative designs, to the complete sequence of equations and values used to calculate a geometric or performance parameter.

* The divisional end users already had accepted ownership of GMGear before production release. The many internal customers for GMGear reached consensus at each stage of requirements and specifications development. They agreed on functionality, and the means of achieving that functionality, as system development progressed. They were also responsible for testing and signing-off on all functionality.

Benefits to General Motors

These benefits reflect the business needs discussed in the introduction and are directed toward meeting the stated project objectives.

* The common system in use at GM divisions contains validated design procedures and data. We have consolidated GM gear design technology into one design system. The divisions contributed the knowledge and participated in the validation. They buy into the results. The use of a common system facilitates information exchange and increases confidence in others' results.

* GMGear automated the design process with an easy-to-use tool. It reduced the design time for the gear sets in a transmission by up to 95%. It allows time to focus more on future concepts and perfection of gear designs.

* In the past, designers often approached gear set design as a "generate and test" proposition. When a generated design tested out as good enough, it was accepted. Now GMGear, with one pass through the program, offers the designer more design options than he/she would likely have ever generated for any particular case. Novices' designs are as good as or better than previous production designs. GMGear high-lights potential issues in new design alternatives. The system makes suggestions for solving over-constrained designs.

* GMGear provides a "glass box" approach, where the knowledge in the system is available for inspection, the decision process is transparent, and the user can override system decisions at any step. We provide several methods for examination of output, including extensive graphical capability. GMGear compares design options for one gear set, or designs for two different gear sets, with differences high-lighted. Extensive explanation facilities for definition and explanation of domain concepts serve as a very effective training tool.

Conclusion

When we measured the current benefits resulting from implementation of this system against the business needs and the resulting objectives established five years previously, GMGear is a successful system.

This system is being accepted by the user community as an expert gear designer. The "glass box" approach, system ease of use, and generation of multiple possible gear set designs contributed to this acceptance. The "glass box" approach played a major role in user acceptance of a system that made decisions formerly made by the users themselves.

Parallel development of specifications and system code permitted the customers to refine their requirements, achieve a better understanding of the advantages of a new technology, and establish a more rigorous and innovative method for gear design. However, although this resulted in a better system, it extended the time required to deliver a production system.

References

1. Barber, K. 1989. Practical Application of Expert Systems to Automotive Manufacturing. Presented at the Society of Automotive Engineers International Engineering Congress, Detroit, Michigan.

2. Barber, K., Burridge, K. and Osterfeld, D. 1988. Expert System Based Finite Scheduler. Proceedings of the Third International Conference on CAD/CAM, Robotics and Factory of the Future, Vol. II, 1988.

3. Jain, S., Barber, K. and Osterfeld, D. 1990. Expert Simulation for On-line Scheduling. *Communications of the ACM*, Vol. 33, No. 10, October, 1990.

4. Bajpai, A., et. al. 1991. Method and Apparatus for Diagnosing Machines. United States Patent Number 4,985,857.

5. Barber, K., Osterfeld, D. and Burridge, K. 1991. Expert System Scheduler. United States Patent Number 5,040,123.

6. Lorenzen, T., Spangler, S., Truss, L. and Corpus, W. 1993. Expert System for Statistical Design of Experiments. United States Patent Number 5,253,331.

7. Bajpai, A.. 1994. An expert system approach to design of automotive air-conditioning systems. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, Vol. 1, No. 8, 1994.

8. Anderson, N., Barber, K. and Kienzle, K. 1991. Artificial Intelligence for Computerized Gear Design. Presented at Japan Society of Mechanical Engineers International Conference on Motion and Power Transmissions, Hiroshima, Japan, November 23-26, 1991.

9. Kienzle, K. and Barber, K. 1992. Applying Peer Review to Improve the Design Process. Presented at the ESD/ASQC Quality Concepts Conference, Detroit, Michigan, October 1992.

10. Marney, S. and Barber, K. 1992. Voice of the Customer in Product Design. Presented at the ESD/ASQC Quality Concepts Conference, Detroit, Michigan, October 1992.

11. Miao, H. and Koga, H. 1994. Analysis of Plunge Shaving for Finishing Gears with Tooth Modifications. Presented to The American Society of Mechanical Engineers Conference, Minneapolis, Minnesota, September 1994.

12. Kienzle, K., Jeffery, M., Jaeger, T. and Barber, K. 1994. Automated Generative Gear Design Process. United States Patent Number 5,297,054.

Table 1: General Motors Gear Producing Divisions

Site	Product
GM Powertrain Transmissions, Opel, Saturn, Isuzu	Manual and automatic transmissions for automobiles and light trucks
Allison Transmission	Off-road vehicle manual and automatic transmissions; tank and other military vehicle transmissions
Saginaw	Differentials for rear wheel drive autos
Delco Products (now Delco Chassis)	Plastic and powder metal gears for windshield wipers, Door locks and power window drives
Delco Remy (now AC Rochester)	Starter motor gear sets
Inland Fisher Guide (now AC Delco)	Plastic gears for power window and power seat drives

Table 2: Gear Types and Arrangements Considered in the First Production Release

Spur, helical and double helical gears
External and internal gears
Low and high profile contact ratio gears
Countershaft (external gear pairs)
Planetary gear sets (all modes, all inversions)
Planet/ring mesh
External spur and helical plastic gears

Table 3: Analyses Provided by the Current Production Release

Geometry:

- Contact ratio
- Face widths off-set
- Face widths mismatching
- Hobbed, shaped and shaved root geometry
- Interference
- Over pin/ball calculations
- Planet phasing
- Planet spacing
- Standard and non-standard proportions
- Standard and non-standard centers
- Tool shift
- Undercut

Performance:

- Bending stress: AGMA, Almen, modified Almen, General Motors Gear Center
- Contact stress along the path of contact
- Duty cycle life
- Gear forces
- Multiple power inputs to a planetary gear set
- Noise
- Reversed bending
- Scoring: flash temperature, PVT
- Tooth deflection
- Weight

Figure 1. Equation description

Figure 2. Variable description

Figure 3. Graphical display of results

Figure 4.   Example of on-line help

Figure 5. Equation trace with Greek symbols and subscripts

Figure 6. Design specifications window for an under constrained design

Figure 7. Auxiliary inputs for a highly constrained design

Figure 8.   Editable defaults for a highly constrained design

Figure 9. Design specification window for a highly constrained design

Figure 10. Comparison report