A Knowledge Base for Designing Casing Strings

Lloyd R. Heinze

Texas Tech University *

ABSTRACT:

1. PROJECT STATUS:

Casing is a prototype software program for designing casing strings. Given the smallest casing size that will be run in a proposed well it will select the required bit size and the next larger casing string. The program bases it's decisions on drift internal diameters, casing to hole clearances, and connection types. Once the correct casing size (OD) is found, this program will design the casing string according to API recommended practices. The program considers corrosive wellbore environments, collapse reduction due to tension, casing price, casing inventory and lists the recommended casing design of each casing size in the planned well.

2. EMPHASIS OF THE PAPER:

The emphasis of this paper is on application of an expert system to casing design and some special interfacing techniques. It uses a FORTRAN program to calculate a casing design after the correct casing data has been selected. The expert system decides which casing(s) should be used and also determines the correct sequence. The expert system selects the proper casing elements from a huge database. It then builds a file which the FORTRAN program will read in as data to do the calculations. The techniques used to interface with a database (DB3) will be discussed in this paper. The methods necessary to call or run an external program (FORTRAN) and return answers back to the expert system will also be covered in this paper. Problems with the limits of FORTRAN handling variables (real or integer number or character) contrasted with an expert system's variables (real or integer number fact, string fact, simple fact, attribute - value fact) will also be discussed. By keeping the database separated from the program, a person in the materials department, can update the database (using a familiar software) without effecting the program.

3. PLATFORM AND/OR SHELL USED IN THE PROJECT:

This knowledge base is written using the expert system shell Level-5. Level 5 was chosen because of its ability to interface with a large variety of programs. In this application the knowledge base handles the query of the users, makes decisions, retrieves the required information about casing from a large database (DB3) casing file and writes (FILEs) only the necessary Information in a data file. Then a FORTRAN program is called (ACTIVATE). This FORTRAN program reads pertinent data from the data file and calculates the correct casing tapers from bottom to top.

---

* Formerly with University of Missouri - Rolla

INTRODUCTION:

Casing strings must be designed based on sound engineering principles and economy. Casing must be designed so that it will not:

1. fail by collapse from external pressure (at the bottom of a taper)
2. fail by bursting from internal pressure (due to drilling operation at deeper depths requiring higher mud weights or to production operation at the middle of a taper)
3. fail under tension (at the top of a taper)

Properly designed combination casing strings result in savings in casing cost and also improved tension factors of safety due to reduction in total weight of the string. Although combination strings consisting of many different weights and grades of casing can be designed, it is general practice to set some reasonable limit on the number of sections in the string. Combination casing strings are designed to take advantage of the fact that the principal forces acting on a casing string are lowered in magnitude in the middle section than at the ends. Consequently, in this middle section, it is possible to use casing of lighter weight or lower strength (grade) than necessary in the top and bottom sections without reduction in the factors of safety for the string as a whole.

The program is divided into several modules. One of the big advantages of the LEVEL - 5 shell is that numerous knowledge bases and external programs can be chained together. This is especially useful because once a module is working it can be isolated so further changes won't effect that piece of the program. By breaking up the program into smaller modules it is easier to debug them and makes following the logic path of each module easier.

This shell uses the standard rule base technique of IF THEN ELSE rules. It allows some CYCLEing, RECYCLEing and INITALIZING. These are terms for starting over a rule, a series of rules, or starting over with new answers.

Reserved words in the AI shell will be in all capitals throughout the rest of the paper. Many of these reserved words (while demanding a different format) cause simular actions to familiar fortran commands as listed below:

Level-5	Fortran
CHAIN	CALL subroutine
ACTIVATE	CALL external program
FILE	WRITE to external file
OPEN	OPEN

Another expert system shell that was tried did not allow for this modulation. As a result when the program became too large, it would not run on smaller PC due to memory limitations. It was also becoming very troublesome because a change in one section of the program tended to affect other sections. Often it was too difficult to follow the programs logic sequence.

Being able to access and run external programs allows for quicker development. It is very convenient to use other fortran, basic, C, or Pascal programs or subroutines that one has already written. This way one can concentrate on using the expert system to make decisions access other routines to do the rest of the work. The access to external programs was easy, simple and trouble free.

Accessing existing databases is very important to this program. Doing this was not easy to learn. The database file that was to be accessed had to be opened with a statement like:

OPEN filename AS DB3 FOR READ CALLED nickname

This translated to something like open filename (DOS eight character maximum with filetype DB3) of database for read only access which will be called nickname (a short name like "Db" which will be used every time instead of this long reference).

Other reserve words used in database handling are:

WRITE	gives read & write access
POSITION	goes to a record number
RECNO	record number
LOCATE	find something in a database field

Any field name in the database can be searched for using the LOCATE command. Once it has been found numerous other field values can be evaluated for a desired value.

After reading the manual (which is quite good in most respect) and examining some of the example programs, two calls to technical support were needed to figure out a method to access the database. It is suggested one spend less time trying to figure out how to do something before going to the particular AI shell's technical support. Level-5 support group has done an excellent job of clarifying any questions so far.

KNOWLEDGE BASE:

Knowledge Representation

The main part of the program is the knowledge it uses for problem solving. Knowledge on casing design methods and casing dimensions and performance properties are derived from the API Recommended Practices 5's. Casing prices are taken from Lone Star Steel Company's 1984 price catalog. After questioning additional pipe manufacturer as well as major and independent oil producers, it was determined that this price catalog is still current. It was noted that major discounts are being obtained from this published price list. However for the purpose of decision making these prices were assumed to be correct. For estimating the cost of a pipe string a discount was applied to this price list.

Safety factors used for collapse, tension, and burst are 1.125, 1.800, and 1.000 respectively. The user is allowed to change these to other values.

The effect of axial tension on collapse is considered. The Maximum Strain-Energy curve discussed by Messrs. Westcott, Dunlop, and Kemler is used in the program.

Short thread and coupled connections (ST&C) are only considered for use in surface casing strings. All other strings use only Long thread and coupled connections and check for the possibility of using Buttress threaded and coupled connections (BT&C) at the top of a string if this will result in a cheaper string design. The user can eliminate from consideration either ST&C or BT&C or both.

Casing.PRL - User Interface

This knowledge base makes the questions posed to the user. It handles user queries of why or explain. It also will give the user options or choices of acceptable answers. This section also checks for validity of the user's answers. If it can determine by the user's answer the correct units it will accept the answer and convert to desired units for additional calculations. If it can't determine correct units it will reformulate the question.

This module also keeps the user informed about what the program is doing. By observing students running programs, it appeared that if the computer doesn't respond within 10 seconds the user becomes impatient. Therefore messages such as "Searching the database for the bottom casing" or "Found the bottom casing, now selecting the remaining useful casing records" are displayed whenever it is anticipated the computer will appear unresponsive. One of this author's complaints about an expensive minicomputer software he has tested was that it took over 60 minutes to run some of the test programs. If this is the case a user should be told, rather than stare at a blank screen and wonder if he has done something wrong and therefore abort the program before it has finished. Additionally the user should be told somehow that the program is still operating correctly and hasn't gotten into an infinite loop.

The user interface still needs a lot of polishing. A method to allow the experienced user to use short cut should be devised. A simple way to change an answer and recalculate without reentering unchanged data needs to be constructed.

Casing1.PRL - Control

This module acts as a control. It is the main program in fortran terminology. It controls the chaining to other AI modules and activates outside programs. No calculating or decision making happens here other than deciding which is the next program to be run.

Casing2.PRL - Casing/Bit/Next Casing

Given a casing outside diameter this knowledge base will recommend the bit size and then the next larger casing size to be used. It takes into account clearances based on casing coupling external diameter, normal API bit sizes, casing drift diameters and normal API casing sizes. The user selects the deepest, smallest outside diameter casing and the program will recommend the remaining bit sizes and consecutively larger casing sizes. This information is displayed to the user and he is allowed to make adjustments or accept the recommendation. The results are stored for use in other modules.

Casing3.PRL - Tubing/Next Casing

This module performs identically to the previous Casing2.PRL module. The only difference is that the user would select the production tubing string, and the computer selects the progressive sizes of casing and bits.

Casing.DB3 - database three file

The information on pipe prices, inventory, dimensions, and performance properties was enter into a database file (DB3). This file was then resorted from the routine sort by casing OD, then by weight, and finally by grade to a more useful sort by casing OD, and price. This sort was thought to be more useful in performing a computer search where pipe choices would be made first by OD and second by price. An excerpt of the large database file that was generated is shown in figure one.

CasingDB.PRL

The module "CasingDB.PRL" takes the results of user responses and uses the following information:

Casing position   Production, Intermediate, Surface, or Liner

Casing setting depth

Mud weight at setting depth

Corrosive or non corrosive wellbore

and the results determined from previously asked questions to select the casing outside diameter. The required minimum collapse rating at the bottom of the hole is calculated based on casing setting depth, mud weight and collapse safety factor. The casing database is then searched. In order to speed up the search proses, the database selector is POSITIONed to the first record that has the correct casing outside diameter. A maximum record to be searched is flagged (last record with the correct outside diameter) in order to speed up the search. This limits it to less than a hundred records. Then a series of LOCATEs are performed within the database aimed at finding the lowest price pipe that will meet the requirements. These requirements in order of search are as follows:

meets minimum collapse pressure rating

meets casing grade rating based on Rockwell hardness and corrosiveness of wellbore

meets threads or connection rating

When the search is successful the candidate casing record is FILEd in file.DAT which will be used as input into a fortran program. It was found that only a limited amount of information could be passed in RAM (random access memory) to a fortran program (this appears to be a limitation of DOS or fortran on a DOS machine).

After this bottom casing is found, the program backs up through the database. As it backs up, it tests each record for the required grade of steel and threads. Records that pass are FILEd in file.DAT. Once all the records of the appropriate outside diameter have been checked in this backing up mode or 11 records have been selected, a "end of record" is FILEd in file.DAT. Control is CHAINed backed to Casing1.PRL, which then ACTIVATEs Casing.FOR.

Casing.FOR

Casing.FOR is a fortran program that does the calculations based on the information that has been selected and put into file.DAT by the expert system. **file.DAT**

These are the data files created by CasingDB.PRL and used as input into the fortran program mentioned above. They will have names (as will file.ANS and file.ERR) reflecting the casing position they are data for. For example:

PRODUCTI.DAT
INTERMED.DAT
SURFACEC.DAT

file.ANS

These are the answer files created by the fortran program and later read by Casing4.PRL for display of results.

file.ERR

These are the error files created by the fortran program. They allow for checking that the program ran routinely or not.

Casing4.PRL - Display results

This last module is still under construction. The exact method of displaying the results has not been decided.

PERFORMANCE EVALUATION

To evaluate the performance of the system, the program was given to a group of senior petroleum engineering students who had just completed a series of homework assignments involving casing design. Their calculated casing designs were checked against the program. Several minor flaws were found in the program and have been fixed. Numerous suggestion have been made on ways to improve the user interface. These fixes are still on the drawing board.

ACKNOWLEDGEMENTS

The author would like to thank two of his colleagues,* Dr. L. F. Koedertiz and Dr. D. T. Numbere, for their support and help in debugging the programs. He would also like to thank his students who acted as initial users and checked the programs results by hand calculations.

REFERENCES

1. Heinze, L. R.; Thornsberry K.; Witt L. D.; "AL: An Expert System for selecting the Optimal Pumping Method," SPE Production Operations Symposium, March 13-14, 1989., Society of Petroleum Engineers, SPE # 18872.

2. Waterman, D. A.; A Guide to Expert Systems, 1986, Addison-Wesley Publishing Company, Inc.

3. Lone Star Steel-A. O. Smith, Casing and Tubing Prices, 1984

4. Westcott, Dunlop, Kemler; "Setting Depths for Casing," API Division of Production, May 1940.

5. Heinze, L. R.; "A Knowledge Base for Designing Casing Strings", 1990 Conference on AI in Petroleum Exploration & Production, May 16-18, 1990, College Station, TX.

6. Heinze, L. R.; "A Knowledge Base for Designing Casing Strings", SPE Computer Applications Journal, Jan./Feb. 1991.

* University of Missouri - Rolla