A SOFTWARE PACKAGE FOR CAPITAL COST ESTIMATION

P.T. VASUDEVAN, DEEPAK AGRAWAL
University of New Hampshire • Durham, NH 03824

EconExpert is a software package for capital cost estimation, primarily intended for use by chemical engineering students in plant design, and can be run on any Unix platform. The system prompts the user for various input, such as equipment category, equipment type, equipment sub-type, material of construction, and operating pressure, and then calculates the bare module cost of the equipment. EconExpert is also programmed to calculate the grassroots capital if the user enters information on all relevant equipment in a plant. It also provides the user with help.

IMPLEMENTATION OF ECONEXPERT

EconExpert is an interactive software package for capital cost estimation. The system was developed in CParaOPS5 and uses ‘C’ external functions. It uses cost data from A Guide to Chemical Engineering Process Design and Economics and is thus a useful supplement to the text. The cost data in the text are expressed graphically in the form of charts, and according to the author, “the charts are accurate enough for preliminary design estimates and are certainly adequate for classroom work.” In this software package, the plots are represented as polynomial equations, and these equations are stored as ‘C’ functions. If the purchase cost is a function of more than one variable, a multiple regression technique is used to fit the data. The output from the routine results in a polynomial equation that gives the closest fit to the data.

OPS5, developed at Carnegie Mellon University in 1978 by Charles Forgy, is a language especially developed for rule-based expert systems. CParaOPS5 is a parallel version of OPS5, written in ‘C’ programming language. One advantage of CParaOPS5 is that it has parallel processing capabilities in addition to the ability to run on a uniprocessor machine. It also allows the user to write external functions in ‘C’ language, which is an added advantage for calculation-intensive applications since OPS5 is not good in mathematical computing. Another advantage is portability of the program. CParaOPS5 converts the OPS5 program to ‘C’ language, and this ‘C’ language program can be compiled on any ‘C’ compiler without needing CParaOPS5.

In addition to the above capabilities, another important feature of CParaOPS5 is that it allows the database to be included in a separate file. The advantage is that the database can be accessed and modified by the user at any time. CParaOPS5 also allows the input to be read from a file and the output to be written to files, which makes it easier for the user to supply data and to obtain results.

THE SYSTEM STRUCTURE

The system consists of a user interface and an extensive knowledge base. This knowledge is represented in OPS5 rules and ‘C’ external functions. The OPS5 rules mainly ascertain the type of equipment and desired material of construction, and also provide help. In addition, the OPS5 rules contain information on the ‘control strategy.’ The ‘C’ func-

---

P.T. Vasudevan is Associate Professor of Chemical Engineering at the University of New Hampshire. He has taught a number of courses that include process control, biochemical engineering, reaction kinetics, mass transfer and polymer engineering. His research interests are in the area of catalysis and biocatalysis.

Deepak Agrawal graduated with a MS degree in Chemical Engineering from the University of New Hampshire in the area of Expert Systems. He is a management consultant helping leading global companies with strategic and tactical issues in the areas of supply chain, operations, and product development.
tions contain information on cost estimation, information on the minimum and maximum sizes available, and information on other key factors.

The system consists of several levels. At the upper level, basic information is sought from the user. After it is supplied, the system moves to the next level to gather more detail information. The levels are:

**Start**  
Starts execution of the program and asks about cost index.

**Level I**  
Asks about equipment category.

**Level II**  
Asks about equipment type.

**Level III**  
Asks about equipment sub-type and calculates purchase costs.

**Level IV**  
Asks about material of construction and other information, calculates various factors, and uses these factors to calculate bare module factor and bare module cost.

**Final Level**  
Adds bare module cost of the equipment to the total bare module cost, asks if the user wants to cost one more equipment. If “yes,” then goes to Level 1—otherwise calculates total capital cost and ends the session.

The structure of the software package is described more fully in the following sections.

**Start**

Once the program is started, it prints a welcome message, initializes the variables, and asks for the cost index. After the cost index is supplied, the information is stored for the duration of the session, avoiding the need to seek this information while costing several pieces of equipment.

The cost index is asked by the rule “ask-cost-index,” which in turn calls an external ‘C’ function. This function asks for the cost index, accepts the input, and returns the value to an OPS5 rule.

**Level I**

In this level, the user is prompted to choose the equipment category of interest to the user. There are 18 broad categories of equipment such as heat exchangers, pumps, mixers, etc. When the user supplies the desired information, the system checks the answer and transfers control to the next level. The validity of the input is checked by another rule, and if the answer is valid, the system moves to the next level in the hierarchy. If the answer is invalid, the system prompts the user again for a different input.

**Level II**

After the equipment has been selected, the system moves to this level to obtain additional information. There is one rule for every category, and each rule contains information on the type of equipment. For example, under “mixers” there are several types available. The system needs to ascertain the user’s interests and hence it asks about the desired “type.” When the system receives the input, it validates it by using rules written for checking invalid inputs. These rules are written in such a way that they can be shared by various categories. After the system gets the correct information, it goes to the next level to seek additional input from the user.

**Level III**

After the system receives input on the type of equipment, it queries the user about the “sub-type” desired. For example, sub-type could be items such as “stuffing box” or “mechanical seal.” When this information is obtained, it requests further information relating to that particular equipment, such as size, area, etc., in order to calculate the purchase cost. This is done by an external function written in ‘C.’ This function has information on the minimum and maximum sizes available for the equipment. If the size specified by the user is less than the minimum size available, it increases the size to the minimum size; if the specified size is larger than the available maximum size, EconExpert splits the unit into smaller equal-sized units. This is a simplistic approach, but this rule can be easily changed as more knowledge becomes available. The relationship between purchase cost and size is expressed in the form of equations, using ‘C’ functions. After the functions calculate the purchase cost, the system moves to the next level.

Certain types of equipment do not possess a “sub-type.” In these cases, the tasks of Levels II and III are combined; that is, after determining the type of equipment, the system directly proceeds to the cost-calculation step.

**Level IV**

The cost of a piece of equipment depends on the construction material. There are factors that take into account the effect of construction material on the equipment cost. In this level, the system asks the user for a preferred material of construction and then calculates the appropriate factor. This

*Summer 1999*
level also has information for validating the input.

In some cases, the cost of equipment varies depending on the operating pressure. This is taken into consideration by employing a "pressure factor." Additionally, the system has information on several other factors such as superheat, corrosion, etc. These factors are also calculated in this level. When the system has determined all these factors, the purchase cost is multiplied by an overall factor, obtained by multiplying all the relevant factors, to give the bare module cost. After the system has calculated the bare module cost of the equipment, it moves to the last level.

Final Level

In this level, the system keeps track of the total bare module cost of the plant by adding the total bare module cost of each piece of equipment. The system thus asks the user about costing new equipment, and if the answer is in the affirmative, the system goes to Level I again. If not, it calculates the total module cost of the plant and the grass roots capital. Once the system has calculated the grass roots capital of the plant, it prints the results and the session ends.

Help Facility

The system also provides help if requested. When the user prompts the system for help, it lists the different types of equipment available in a particular category. If the user asks for help in Level I, for instance, the system will print the list of available categories and will then prompt the user for input. The system provides appropriate help at each level and contains rules for transferring control between any level and the help facility, and vice versa. These rules are written in a way that can be shared by all categories.

SYSTEM TESTING AND USER REACTION

After development of EconExpert, the system’s performance, reasoning, and knowledge were tested and validated. It was used by all seniors in the spring of 1998, and the bugs reported by them have all been corrected. The program was tested to check for the validity of its reasoning and also for its ease of use and aesthetic appeal.

We have used the software in other courses, such as a biochemical engineering and thermodynamics. Thus, the software aids students not only in the senior capstone design class, but also in other courses where a quick economic evaluation of a chemical process plant or unit is required. The software has helped instructors in different courses to obtain relative-cost data. Students have found the program to be quite useful since it provides them with a quick means of determining the cost of equipment to obtain total capital cost, and in making economic evaluations of different processes or competing technologies.

The software is available on the University mainframe computers as well as Linux boxes, and students thus have ready access to it. Students have been uniform in their praise of the software since it is so simple to use and they can access it from anywhere in the University or from their homes. The software was used in the spring 1999 semester by seniors taking design and by juniors taking thermodynamics. Instead of spending an inordinate amount of time in obtaining cost data from the plots (which required careful interpolation, especially since the plots are rather small), the students are able to focus on the design problem itself and to quickly evaluate different flowsheets (technologies) or different options (equipment types) within a given process. Thus the pedagogical value of the package is quite high.

The system was tested by having users try it and noting their responses. EconExpert was tested for all the equipment it contains. The bare module cost supplied by the system is in agreement with values reported in the literature. The system is successfully carrying out all the tasks such as checking size, providing help, validating input, etc. The system also has the flexibility to be continuously updated as data become available in the future.

CONCLUSIONS

EconExpert is a software package for capital cost estimation. It is intended for use by students in chemical engineering and has a number of features, including

- The movement of the system is efficient since it moves from a node at one level to a node at the next level, until it tracks down the right equipment. The way this system is written, it finds information about the equipment using very little computer time.
- It checks for the minimum and maximum sizes available. If the desired size is out of range, it knows what to do.
- It has a help facility to assist the user.
- It uses ‘C’ functions extensively since the design process is highly computational. The calculation part is encoded in ‘C’ functions to increase the efficiency of the system.
- The system has the capability of checking user input at every level.
- The executable code runs on any Unix platform.

Readers may send e-mail to the author (ptv@cisunix.unh.edu) for more information about the software and its availability.

REFERENCES