KNOWLEDGE BASE SPECIFICATION FOR DESIGN COST ESTIMATION

by

Shauna Mae Gesner

TR94-091 November 1994

This is an unaltered version of the author's
M.Sc.(CS) Thesis

Faculty of Computer Science
University of New Brunswick
P.O. Box 4400
Fredericton, N.B.
Canada E3B 5A3

Phone: (506) 453-4566
Fax: (506) 453-3566

KNOWLEDGE BASE SPECIFICATION FOR DESIGN COST ESTIMATION

by

Shauna Mae Gesner
B.Sc.(C.S.), University of New Brunswick, 1990

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Masters of Computer Science
in the Faculty
of
Computer Science

This thesis is accepted.

....

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

October, 1994

ACKNOWLEGEMENTS

I would like to thank my thesis supervisor, Dr. Brad Nickerson for his assistance and guidance throughout the development and implementation of this thesis.

As well I wish to thank Laurie Tompkins of Neill and Gunter Ltd. for his input into the design of the prototype and for his help in providing the necessary technical information.

I am especially grateful to my friends in the Faculty of Computer Science for their support and encouragement. As well, a very special thank you is extended to my family who helped and supported me through the difficult times and who now share with me the sense of accomplishment from achieving my goal.

DEDICATION

To

my mother, Inez

and

in memory of

my dear father, Leo

ABSTRACT

This thesis presents an effective method of representing cost estimate knowledge in a readily accessible format for generating accurate cost estimates. The investigation also includes development of a prototype tool for modifying this knowledge and producing cost estimates.

Models of the important cost estimate inputs (equipment lists and device lists) and the resulting cost estimate have been developed. The cost estimate model is represented in the form of a context-free grammar. Key concepts and heuristics were abstracted following various discussions with an electrical engineer familiar with producing estimates. The individual was also consulted during the implementation of a user interface for the prototype tool.

An investigation into the integration of the prototype with other software tools (e.g., Lotus 1-2-3, WKS Library of C functions) was also conducted. The resulting prototype is called Assistant for Cost Estimating (ACE). It permits the user to record a client's preferences for a particular manufacturer and use this information to generate an estimate which reflects the economy and even geographic location of the project.

The prototype was developed using **ART-IM** version 2.1(Automatic Reasoning Tool - Information Manager) and the WKS Library of C functions using an IBM PS/2 Model 70 (Intel 80386 processor).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Cost Estimating

One of the most difficult, yet essential, design activities is producing a reasonably accurate cost estimate at the design stage. If an estimate is too far below the actual cost of the project, then funds may not be available to ensure its completion. Likewise, an estimate that is too high may raise doubts on behalf of the clients with respect to its feasibility [Gesner, 1991].

Currently, most cost estimators prepare and adjust estimates manually with the exception being those individuals who use a spreadsheet packages to calculate and store costs and then transfer the results to paper. Software for producing construction cost estimates (after the design is complete) is available, but, the reasoning capabilities of these packages are very limited [Timberline, 1990].

1.2 Cost Estimating Systems

The concept of expert systems is best described as follows [Stretton and Stevens, 1990]:

> Expert systems specifically involve the acquisition of knowledge from human experts. The knowledge is separate from the manipulation process, and therefore relatively easy to modify.

This characteristic of an expert system and the cost estimating process are very similar. In both situations the necessary knowledge is acquired from experts and must be easy to update. Cost estimating systems currently available do not integrate the costs with the design [Johnson, 1990; Clark et. al., 1989].

1.2.1 Cost Estimating Models

Cost estimating provides much information about a project, including project feasibility, life cycle, bidding cost and resource allocation. Internal (e.g., labor costs, scheduling decisions) and external (e.g., weather, government laws, inflation rate) parameters and their hierarchial importance add to the complexity of the estimating process and greatly affect economic decisions. A cost model is best described as an input-output relationship where historical data and project specifications are used to generate a cost estimate. The resulting estimate is not only used to determine the consequences of investing in a project, but also acts as a monitor as the project progresses.

The estimating process involves making rational cost versus benefit decisions with respect to what should be included in an estimate. Once the costing model has been established, the estimator must then determine what values are to become valid inputs and their source of origin. There exist at least three input sources which are described below [Ntuen and Mallik, July 1987]:

1. In-house estimates can be viewed as prior information coming from the estimator's personal experience or from other individuals in the firm.
2. Personal beliefs are based on the estimator's intuition and judgement.
3. Similar projects may also be used as a base or model upon which to build an estimate.

Various approaches may be used to generate a cost estimate. A method of classifying these approaches has been proposed by H. Wildemann and uses the following criteria [Wildemann, 1981]:

1. Application
2. Function range
3. Timing of the cost estimate in the project's life cycle
4. Essential information
5. Required accuracy

Although this helps in systematizing cost estimating models, it does not include the intuitive criteria used by management when choosing a cost model. The cost estimating process incorporates engineering, financial and management skills and is the prediction of time, resources and costs of future activities. Additional categorization using decision-making levels is therefore required. The three levels are [Ntuen and Mallik, July 1987]:

1. Strategic planning which is used to determine management's objectives and any necessary modifications to the objectives, as well as the required resources to reach the organization's goals. Policies with respect to how the resources are obtained, used and disposed of must also be considered.
2. Management Control is an extension of strategic planning and is used by managers to ensure that resources are efficiently and effectively employed to meet the organization's objectives.
3. Operational Control goes one step further than management control by ensuring that each task is completed on schedule while meeting all required standards.

Combining the original list of criteria for classifying cost estimating models with the three decision-making levels produces the following additional criteria [Ntuen and Mallik, July 1987]:

1. profitability
2. productivity
3. flexibility
4. "value" analysis (i.e., how useful is the cost breakdown)
5. bidding authorization (i.e., can the estimate be used for bidding purposes)

Using these characteristics enables one to select a feasible cost modeling approach for generating an accurate cost estimate. It should also be noted that the complexity of the project directly influences the complexity of the estimating process. A simple project may only require basic calculations such as unit cost by quantity required whereas a more complicated project necessitates more complex formulae. Accurate estimating involves vast amounts of intuitive knowledge and expertise. This, combined with the project's complexity and the many other facets of cost estimating, characterize the cost estimating process as an ideal expert system application [Ntuen and Mallik, July 1987].

1.2.2 Automating Cost Estimating

1.2.2.1 Artificial Intelligence and Its Benefits in Cost Engineering

Artificial intelligence is defined as [Firebaugh, 1989]

> Artificial intelligence is the science of making machines do things that would require intelligence if done by men.

Designers of artificial intelligence systems try to create programs capable of performing complex tasks using expert knowledge. Some situations where artificial intelligence has proven to be useful are [Ntuen and Mallik, May 1987]:

1. Large complexities where underlying entity hierarchies exist.
2. Unstructured problems that are not easy to define mathematically.
3. Problems where knowledge is scattered and needs to be condensed for future reference.
4. A meta model which investigates various solutions to a given problem.

Initially estimates were generated using simple desk calculators. The rapid development of new technology led to more complex projects and designs. As well, estimates had to be quickly and accurately generated enabling increased productivity. This, in turn necessitated more comprehensive estimating practices. Cost estimators then began to use spreadsheet systems to aid them in decision-making [Ntuen and Mallik, May 1987].

Problems arose as databases became large and cumbersome, and mathematical models proved insufficient for performing complex cost estimate calculations. Estimators needed a general estimating process that could be applied to any project while incorporating costing knowledge and expertise. Knowledge-based systems (KBS) and natural language processing (NLP) were the two subfields of artificial intelligence considered for this purpose. A knowledge-based system captures the knowledge of an expert in his/her domain of specialization and can make it available to less experienced users. It is often referred to as an expert system (ES) as it combines the knowledge of several experts and uses this knowledge to solve problems [Ntuen and Mallik, May 1987].

Conveying this information requires a language structure that differs from spreadsheet or database systems in that it incorporates both semantic and syntactic structures. This combination provides "both descriptive and prescriptive information feedback" for the user [Ntuen and Mallik, May 1987]. Cost estimators in particular, benefit from expert systems. Their jobs may involve defining the basis of an estimate, devising results or solving control problems. All these tasks require experience and good judgement which can be easily represented in an expert system [Ntuen and Mallik, May 1987].

1.2.2.2 **Rule Based Systems**

Expert system design is rapidly becoming a well-developed technology and is proving to be very viable commercially. The key to the success of this technology is the method used for knowledge representation. Rule-based systems have proven to be one of the most successful schemes for this purpose. They play a number of important roles, some of which are listed below [Firebaugh, 1989]:

1. Rule-based systems provide the user with a powerful knowledge representation scheme including both knowledge and action.
2. They act as a bridge between the field of artificial intelligence and expert systems by providing a language in which the representation of expert knowledge is very natural.
3. They provide a heuristic model for human behaviour.

The knowledge in a rule-based system is represented by a set of rules of the form shown below:

IF [condition] **THEN** [action]

combined with a control system and a global database (Figure 3.1). The global database is the main data structure of rule-based systems. It can range from a simple list or matrix to a complex, relational, indexed file structure. The database is the basic structure on which the production rules operate. It is dynamic in nature as it is constantly changing as a result of the operation of these rules. The control system is an interpreter program which controls the order in which the production rules are fired and resolves any conflicts if more than one rule applies. The control system continues to apply the rules to the database until a description of a goal state is reached. It detects the occurrence of the goal state and records the rules applied to reach it for future reference. The rules have a condition and an action part. If the condition is satisfied, the rule becomes applicable and may be fired (the action part is completed) by the control system. This separation of the program structure from the knowledge base is one of the advantages of rule-based systems [Firebaugh, 1989].

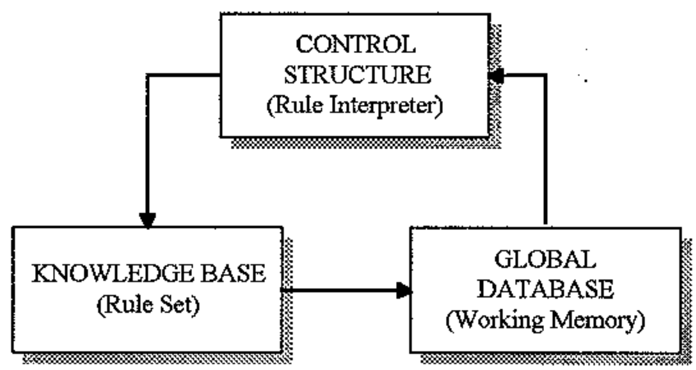


Figure 1.1 The three basic components of rule-based systems [from Firebaugh, 1989].

1.2.2.3 **Rule Models**

In order to successfully acquire and represent knowledge for an expert system, one must be able to conceptualize its domain. This involves abstracting the key concepts and heuristics represented in the domain. If this is not done properly it can have a negative impact on future knowledge acquisition, as well as on the resulting expert system. Rule models are very useful for this purpose as they represent knowledge at a high level and are easily understood by non-experts [M^cGraw and Harbison-Briggs, 1988].

A model can be viewed as a "compact, high-level description of structure, organization, or content that may be used both to provide a framework for lower-level processing and to express expectations about the world" [Buchanan and Shortliffe, 1984]. Establishing a model involves interpreting the natural language used by an expert to express their knowledge about a given field of study. The model must [Buchanan and Shortliffe, 1984]:

1. represent the structure, organization or content of this knowledge.
2. act as a guide in interpreting the knowledge.
3. help predict the contents of new rules.

The generation of such a model for a rule-based system is not as difficult as it appears. Rules about a specific topic tend to exhibit similar characteristics or methods of reasoning. Rule models can be constructed from this information and provide abstract descriptions of the various rules used in the system. They are often referred to as meta-level knowledge because they are used to establish standards for encoding knowledge in a specified rule format [Buchanan and Shortliffe, 1984].

1.2.3 Cost Estimating Approaches

An investigation of various cost estimating packages currently available and the existence of a general-purpose cost estimate model has been conducted. The following sections outline the results of this research and offer further insight into the cost estimating process.

1.2.3.1 PAINTER

The cost estimating process "involves finding and using information specific to a given job from a large amount of possible information" [Biegal et al, 1986]. The example outlined in this article is a rule-based, C-coded, cost estimating program for house painting entitled PAINTER. It uses decision tables and user input to generate an estimate. The user enters data about the painting job which is then used to access the appropriate decision table. Each table then accesses the next appropriate table for further evaluation. For example, when the user enters the surface type, the surface type table will only access tables pertaining to the specified surface.

The benefits of this methodology are as follows [Biegal et al, 1986]:

1. Unnecessary program evaluation is eliminated as only those tasks required for a project are performed.
2. The integrity of the decision tables can be evaluated by using ERGO Logic Kit (a tool used to develop the decision tables found in PAINTER's rule base and convert the table logic into C-code).
3. Programs using decision tables can accommodate both forward chaining (determines a solution given a set of conditions) and backward chaining (given a set of hypotheses, tries to find the correct one(s)).

PAINTER uses a form of blackboard architecture to manage its data. All key variables used by the system are declared global to the programs. That is, they can be accessed and modified by any programs and functions. The control architecture of the system is very simple. A main program calls various program modules and passes control to them. The advantage is that a module can be modified without affecting the other modules. To eliminate the need for a natural language interpreter, the user is provided with a set of answers for each question asked. This also prevents the user from going beyond the system's knowledge domain and preserves integrity [Biegal et al, 1986].

As previously mentioned, decision tables are used to direct the flow of control. A table is composed of four basic components. This design approach greatly simplies the rule base as irrelevant questions are not asked [Biegal et al, 1986]. The table components are:

1. A condition stub which contains the questions pertaining to that particular table.
2. A condition entry which contains possible answers to the questions
3. An action stub of actions that may be taken depending upon the answer chosen.
4. An action entry which is a numeric matrix relating the answers to their appropriate actions.

According to the designers of PAINTER, its intelligence "lies in the fact that PAINTER does not ask any irrelevant questions" [Biegal et al, 1986]. It uses prior information entered by the user to determine the what questions to ask. This is viewed as a way of building user confidence in the system regardless of the user's level of experience [Biegal et al, 1986].

1.2.3.2 **Costimator**

Mars Manufacturing in Mars, Pennsylvania produces machined parts and initially generated their estimates manually by performing many of the calculations mentally. Only 10 percent of the estimates were within plus or minus 15 percent of the actual cost, and approximately 50 percent were underestimated. This changed when the company purchased a cost estimating package for metalworking operations. Costimator, developed by Manufacturers Technologies, Inc., West Springfield, Massachusetts, has an extensive database containing information on materials, machine tools and labor standards that can be readily customized. Costimator uses a sonic digitizer to trace over drawings or CAD screen images to retrieve measurement data. This data is then combined with information regarding speeds, feeds, materials, standards and drawing scales to calculate total machining times and costs. Material costs may also be included in the database thus enabling cost-per-part estimates.

The main advantages of using a computerized estimating system are improved accuracy, consistency and speed. However, other benefits are highlighted once the system is fully implemented. An estimator can vary the quantities and the software will adjust the estimate given these changes quickly calculating a new bottom line. Estimates can be prepared, on average, in 15 to 50 minutes. The same detailed quote prepared manually could require up to four hours to complete. The estimator can investigate and compare various quotes to determine which is most viable. As well, the database can be readily modified or expanded. At the time of publication, Costimator's database contained information on more than 8500 machine tools, 1200 materials and 2500 standards. In addition to the final estimate, Costimator produces a detailed description of machine speeds, feeds and other pertinent information, as well as a listing of cutting and non-cutting times. With respect to accuracy, Mars Manufacturing has virtually eliminated its underestimating problem and "80 percent of the quotes are within five percent of the real manufacturing cost" [Nightingale, 1987].

Although computerized cost-estimating provides very comprehensive and accurate information, a great deal of skill is still required to produce valid results. Not all database values will apply to all situations and account for all variables. The most important thing to remember is that the company must draw on the user's estimating experience and expertise and apply this knowledge to the estimating process.

1.2.3.3 **PRECISION Estimating**

PRECISION Estimating is a computer estimating system produced by Timberline Software Corporation. It organizes, calculates, summarizes and reports on the various items found in an estimate. PRECISION contains the following features [Timberline, 1990]:

1. It combines a traditional cost spreadsheet with a powerful work package takeoff window (developed by Timberline) to produce cost estimates. The estimator can use the window to efficiently and accurately take off items needed for an estimate.
2. PRECISION provides bid analysis and bid summary capabilities.
3. The estimator can enter detailed information about labor crews thus improving the estimate's accuracy.
4. It produces dimensional audit trails (review of all items in the sequence they were taken off) which can be used to check items, quantities, locations and dimensions.
5. Prices are automatically updated from external pricing services and databases.
6. Numerous reports designed to maximize information and readability can be produced.
7. PRECISION is capable of importing and exporting from many popular spreadsheet, word processing and database programs.

When using PRECISION Estimating, the estimator is only required to specify the items and enter their quantities. The system calculates all amounts taking into account any conversions, waste and labor rates and material prices. As well, the estimator is provided with two methods of item takeoff. Items can be selected individually or taken off as a work package (an assembly of items assigned to a location). PRECISION uses user-supplied information about the dimensions of the location to calculate all formulas and selects the appropriate items. It displays the costs and unit costs, as well as a list of all takeoff dimensions for each work package. PRECISION then calculates various items (e.g., quantities, prices, extensions, conversions, addons) and finally generates the work package items out to the main estimate spreadsheet. The estimator can change any of these values and the system will automatically recalculate the estimate. The spreadsheet can also be scanned for incomplete items. When the information is available these items can be updated using the work package window and the changes readily propagated through the estimate. Prices for the items are then pulled into the spreadsheet from the database. It is possible to modify these values by [Timberline, 1990]:

1. clearing all prices in the spreadsheet and entering new ones.
2. raising a particular factor (i.e., labor) by a certain percentage.
3. adding in vendor quotes.
4. update prices in the database itself using external pricing services and databases.

PRECISION also provides the user with an extensive reporting facility. Listed are just a few of the numerous outputs available to the estimator [Timberline, 1990]:

1. takeoff audit - review of all items in the order they were taken off.
2. spreadsheet - copy of the complete estimate ordered by division and cost category.
3. standard estimate - shows quantities, unit costs and extensions in phase order
4. work package estimate - shows the estimate by work package with up to three levels of detail.
5. estimate details - fully detailed view of the complete estimate (all levels).
6. category costs - summary of estimate (total quantity and dollar amounts).
7. takeoff unit costs - unit costs and quantities.
8. unit price review - review of all items and their unit prices.
9. bill of materials.
10. crew analysis- complete listing of all labor and equipment class requirements and costs.
11. crew report - details about each crew and their productivity, extended hours and dollars.

12. field report - full listing of all items, crew makeup, quantity of materials and hours needed and subcontractors.

As well, a digitizer can be added to the system enabling the estimator to graphically view all areas, lengths and counted points and readily identify any errors or omissions [Timberline, 1990].

The PRECISION Estimating package is primarily used for producing construction estimates for tendering purposes following the design phase of a project. Although PRECISION does not posess reasoning capabilities, it does provide the user with quick and easy access to vast amounts of information. External pricing services and databases can be accessed for accurate costing information. Changes to items and quantities can be made and are immediately propagated through the estimate. The user interface is simple in design and easy to use.

Some of PRECISION's techniques can be readily used during the design phase of a project. The ability to make changes (e.g., incorporate various factors based on location or environment) and have these changes propagate through the estimate is useful for assessing "whatif" situations. The incorporation of cost databases helps improve accuracy during the design phase and lessens the amount of time needed to look up this information. As well, the various reporting facilities are consistent with an estimator's requirements to be able to provide up-to-date information to management.

1.3 Electrical Design and Cost Estimating

Currently the process of producing a cost estimate at the design stage is a tedious and in some situations inaccurate process. Most cost estimates are prepared manually with the exception of those individuals who use a spreadsheet package to calculate and store costs and then tranfer the results to paper. The area of electrical engineering design relies extensively on past design experience, client preferences and various electrical standards and codes. Some of the basic inputs of an electrical cost estimate are listed below:

1. motor list - the motors to be used in the project.
2. device list - any switches or control mechanisms.

3. cable list - all necessary wiring for the motors. Cable type and quantity can be determined from the motor list.
4. any necessary services (e.g., telephones, fire alarms, heating, ventilation, air conditioning (HVAC), lighting or communications).
5. client preferences - some clients may use a specific motor type as this best integrates with their current environment.
6. standards/codes - various countries and industries have standards which must be strictly followed.

The above list is neither complete, nor constant. Motors may be upgraded, thus requiring different cables and switches. These changes must be propagated through the estimate and also through the design [Gesner, 1992].

1.4 Statement of Problem

The objectives of this thesis are summarized below [Gesner, 1991]:

1. What would be an effective and general-purpose method of representing various cost estimating rules in a knowledge base?
2. What is the best way to represent a client database so that rules or preferences from previous projects can be applied to current ones?
3. Is it possible to integrate other software tools (i.e., Lotus 1-2-3, electronic cost estimating databases) with an expert system and effectively propogate changes in a design among the packages?
4. Can a prototype expert system be designed and implemented to produce accurate cost estimates and permit modifications to its knowledge base?
5. What characterizes an appropriate user interface?

2 COST ESTIMATE MODELLING

A cost estimate model provides a basis for generating an estimate. It establishes general guidelines that with minor modifications can be applied to any project in various engineering fields and the construction industry. A model sets certain standards and constraints that must be followed and ensures overall consistency. It helps cost estimators produce more accurate estimates while improving efficiency and can also act as a training tool for inexperienced estimators.

2.1 Traditional Estimating Methods

During the early stages of a project's development, time and money are often restricted. For this reason the cost of preparing a fully detailed cost estimate for the preliminary design cannot be easily justified. As a result estimators will often apply "cost engineering principles". These principles are based on the costs associated with similar past projects, although the equipment layout and the processes involved may differ. The following sections briefly outline the complexities and complications associated with two traditional methods of cost estimating which incorporate some of these principles [Blecker and Smithson, 1985].

2.1.1 Factored Estimating

The factored estimating method involves developing costs for equipment items and using factors to determine installation costs and other indirect costs (i.e., prime contractor engineering costs, construction overhead costs) as shown in Figure 2.1. The equipment costs may be obtained by the estimator from vendors, cost versus capacity charts or from previous estimates kept on file. The prices are then increased to some future point in time when the equipment will be ordered. Cost engineers can only approximate price increases when applying factors for estimating equipment installation costs for new processes. The installation costs are usually calculated by selecting appropriate factors for the various installation accounts (e.g., civil, electrical, piping, equipment setting) and then applying them to escalated equipment costs to obtain labor and material costs. This can prove problematic as errors in judgment are magnified depending upon the quantity of equipment involved.

The direct field costs are then obtained by adding the material and labor amounts to the escalated equipment costs. Indirect costs are determined by selecting factors for the prime contractor engineering costs (associated with equipment and field material accounts) and construction overhead costs (associated with field labor activities) and applying these values to the direct costs. The summation of the equipment, installation and indirect costs is referred to as the bare plant cost. Contingency costs plus a contractor's fee are then added to the bare plant cost resulting in a factored estimated cost for the project [Blecker and Smithson, 1985].

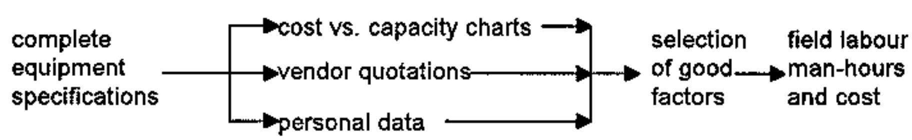


Figure 2.1 A factored manual estimate process [from Blecker and Smithson, 1985].

2.1.2 **Definitive Estimating**

The definitive method is quite different from the factored method in that drawings are prepared for the equipment items, foundations, buildings and structures and other items required for the project (Figure 2.2). Material take-offs are prepared from scale drawings and accurate counts are kept for any necessary piping, concrete, instruments or other bulk materials. The unit costs for bulk materials come from manufacturers' catalogues and current price lists. Labor costs are determined by first tabulating the bulk material quantities. Next the field labor required and any necessary construction rentals are determined for each material item and each project task. Finally the direct field material costs and the associated man-hours are calculated. Equipment manufacturers are then contacted to obtain actual prices as well as delivery times. The equipment costs and direct and indirect costs (prime contractor engineering and construction overhead costs) are then used to obtain the bare plant cost. The contingencies applied for a definitive estimate are usually much lower than factored estimate contingencies as the major equipment is already defined. The contractor's fee, based on the bare plant cost, is added to this running total resulting in a final project figure.

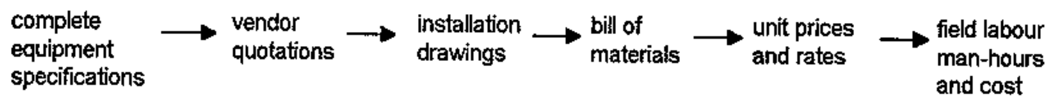


Figure 2.2 A definitive manual estimate process [from Blecker and Smithson, 1985].

Although the definitive estimating method is extremely accurate, there are limitations. It is time consuming and expensive, and variances with respect to site conditions and productivity may change during construction.

2.2 Computer-Aided Estimating

The preparation of accurate cost estimates using either the factored or definitive methods is a detailed process requiring a great deal of effort and attentiveness by the estimator. Many of the tasks involved are repetitious (e.g., transferring values from one section of the estimate to another or performing calculations for items where only the values differ) and can easily be performed by a computer (Figure 2.3). This gives the estimator more time to consider important decisions and make valued judgments.

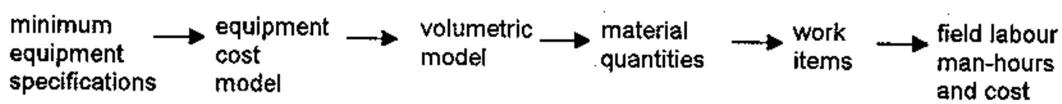


Figure 2.3 A computer-aided estimate process [from Blecker and Smithson, 1985].

The key concepts in designing and implementing a cost estimate model are flexibility and customization. Every project differs with respect to materials, machines and methods of organization. A cost estimate model must take these restrictions into consideration, as well as be readily adaptable to future changes in the design format. Historical data must be continually updated to ensure accuracy and new production technology costs must be included. Customization is essential as the methods used to generate a cost estimate vary depending upon the client and their specific needs [Lee and Ebeling, 1987].

The following sections outline various cost estimating models and the computerized systems that implement them.

2.2.1 C$\phi$$T System

The C$\phi$$T System, currently in its second decade of commercial use, produces detailed estimates for various process plants (e.g., chemical, oil, ore processing) using proven engineering costing and construction modelling processes. It does not rely on factors, exponents or lists of equipment costs. The system generates cost estimates similar to those produced by the definitive method. It performs this task more efficiently and with minimum information for the decision-making process during the project's preliminary stage. These advantages result from the inclusion of the equipment design and cost models, volumetric models (e.g., bulk material quantities) and various work items (e.g., labor) in the system.

As illustrated in Figure 2.4, the estimator inputs the equipment specifications for the project and the system generates the equipment designs and associated costs, as well as a take-off of the required materials using cost and volumetric models. The bulk quantities are then used to determine the amount of labor required. Each task to be performed in a project is broken into work items. The work items detail the amount of labor needed to install specific units of materials using the bulk quantities and work time measurements; the labor costs can then be estimated.

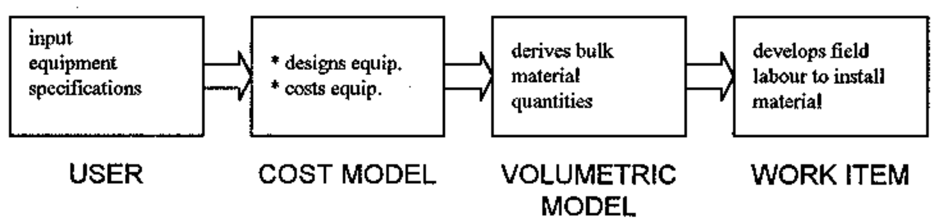


Figure 2.4 System information flow in C$\phi$$T [from Blecker and Smithson, 1985].

Data entered into the CϕST System is categorized as one of the following types [Blecker and Smithson, 1985]:

1. Project level information regarding the project's manner, style, currency, units of measure, escalation and materials cost indexing, design basis, contract relationship or field labor.
2. Unit level information about each operational unit (sections in the plant devoted to a specific function) and the dry equipment and necessary installation items included in these units.
3. Specific information concerning a particular piece of equipment or installation item.

Various design and cost reports are prepared outlining estimated equipment costs, as well as a take-off for bulk materials and labor. The data is available for each equipment item, each plant area, each contract work package and for the entire plant. The cost estimates generated are similar to those produced using the definitive method for the decision-making process during a project's preliminary stage. The difference is that the CϕST System perfoms this task more efficiently with minimum information. These advantages result from the inclusion of equipment design and cost models, volumetric models (e.g., bulk material quantities) and various work items (e.g., labor) in the system.

The above information is obtained from process specifications, plans and flow-sheets. Each equipment item entered into the system is described as being a minimum input, detailed input or supplemental input. Minimum inputs require only the equipment type and one or two size values. Detailed input allows the estimator to specify up to forty different size parameters. If an equipment item's purchased cost is known, it can be entered as a keyword in the item's description. The installation costs and associated materials are then determined using the computer models (equipment design and cost models, volumetric models, work item models) contained in the system. A variation of the key-word costing method is the supplemental cost method of data input. Equipment items of any type, size or shape can be assigned to codes of accounts and the system evaluates installation requirements using user-supplied piping, civil or steel criteria. If these requirements are not available then the bulk items are assumed to be included in the user-specified cost. It should be noted that as more details about the project become available, they can be entered into the system and a new estiimate can be generated incorporating those changes.

The cost of producing estimates with the CϕST System is roughly equivalent to about one and a half engineering man-hours per main equipment item. The estimator is able to generate detailed estimates outlining the quantities, man-hours, resources and costs of a project. In comparison with manual methods, it requires only one-fifth of the man-hours and can produce estimates in one-fifth of the time. The CϕST System has been used by contractors and owner companies in the preparation of "proposals for feasibility, quality, and cost" and "to optimize the design of facilities and processes" for thousands of projects [Blecker and Smithson, 1985]. Comparisons between manual method results and those generated by CϕST have "justified the confidence of System's users and validated its use" [Blecker and Smithson, 1985].

2.2.2 Cost Estimating System

This section discusses a cost estimating system (CES) implemented by Sungyoul Lee and Dr. K.A. Ebeling, North Dakota State University. It has a "standarized program structure and unified data base external to the program source code" [Lee and Ebeling, 1987]. Figure 2.5 outlines the flow process of the system.

Part and assembly drawings and operation process charts provide the information needed to produce a parts list (saved as a spreadsheet file) and the production sequence (saved as a data file). After these two files are generated a main program is run using the parts list file to determine which raw material costing programs to load and execute. These programs load the material cost estimating parameter file and prompt for inputs to determine the unit cost of the raw materials. The inputs and resulting calculations are stored in a material result file. Similarly the main program calls the work center programs specified in the production sequence file. The user is then prompted for inputs which are stored in a work center result file.

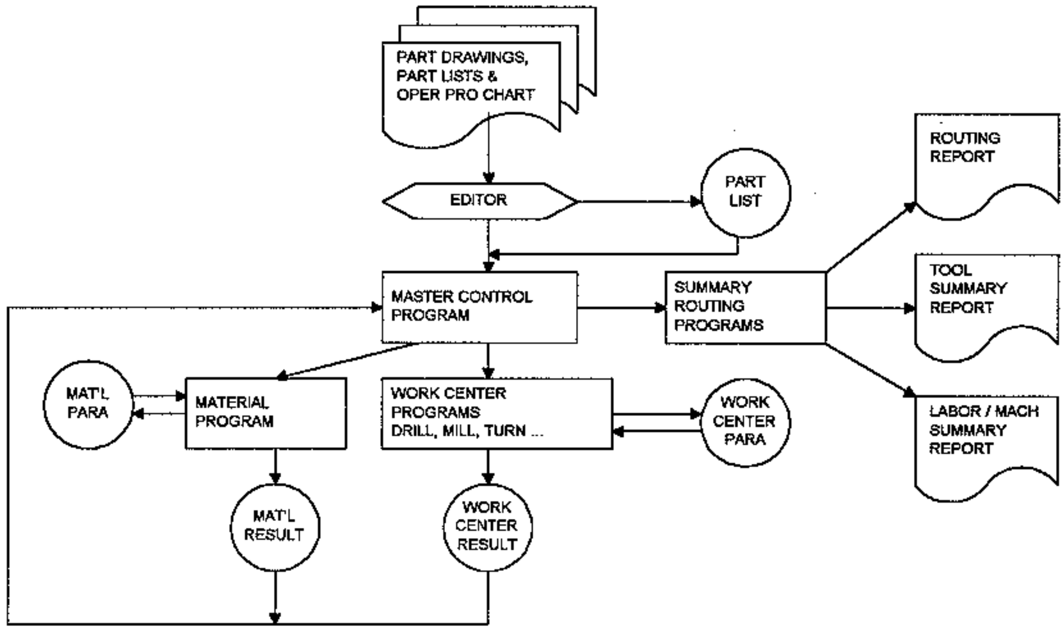


Figure 2.5 Cost estimating flow process for CES [from Lee and Ebeling, 1987].

The two result files are used as input to generate [from Lee and Ebeling, 1987]:

1. **work center detail report** - Details the set-up time and production run time for each part in the material list.
2. **detailed routing report** - list of the machines used, the operations performed and the required settings.
3. **tooling report** - outlines fixture design and manufacturing expenses.
4. **labor/machine summary report** - details the set-up time and production run time for each part in the material list.

Producing each of these reports individually allows the user to change original input values at any point during the estimating procedure without redoing the entire process. Cost estimating parameters and even menus may also be conveniently altered. This ability to customize greatly improves the functionality of the system from both the user's and maintenance points of view.

2.2.3 PREDICTE

PREDICTE (**PR**oject **E**arly **D**esign - Stage **I**ndicative **C**onstruction **T**ime **E**stimating) is an expert sytem developed by DEC (Digital Equipment Corporation) and Civil & Civic, a subsidiary of Lend Lease Corporation in Australia. It is used to estimate indicative construction times for concrete-framed, multi-storey buildings when little information is available. As well, PREDICTE aids in the evaluation and improvement of early building design concepts. The system is designed to simulate an expert assistant to the estimator. Its basic functions are as follows [Stretton and Stevens, 1990]:

1. Determine a viable construction time.
2. Analyse and evaluate the design and alternatives.
3. Explain the assessment.
4. Produce a report and a bar chart.

The most beneficial characteristic of the system is its ability to compare alternative designs. Thus, its main use is for evaluating early design concepts and suggesting improvements.

After the questions regarding the project's size, shape, appearance, ground conditions and surroundings are answered, PREDICTE generates an activity list which shows the start and end dates and duration in working days for each activity group (e.g. site establishment, footings, services). The user is then provided with other options to assist in developing more optimal solutions. The primary ones are [Stretton and Stevens, 1990]:

1. **Analysis and Opportunities** identifies time-intensive features, and suggests opportunities for potential time savings. If the user decides to use the facility to test the alternative, the system will recalculate and display the new results.
2. The option **Explain an Activity Time** permits the user to examine all paths that lead back to the initial input. Explanations take the form:

   When: (conditions)

   Where: (variable meanings)

   Sources of information are identified and may be a calculated value, a question number or a PREDICTE estimate. The user may obtain more information about variables or questions and can override any PREDICTE estimate by setting them to a user-defined value.

3. **Changing a Question response** allows the user to assess **what if** situations. Once the changes are entered into the system, a new time estimate is generated.

During the initial concept stages, little information is available and the project completion date may be four or more years away. The reliability of an indicative time estimate is usually no more reliable than plus or minus one month in twelve. When PREDICTE was tested on recently completed projects the indicative estimates were closer to the achieved results than would normally be expected. At the time of publication PREDICTE had only been in place less than two years and no examples were available for completed projects.

2.2.4 Summary and Analysis of Cost Estimating Systems

The following three tables summarize some of the more common attributes associated with cost estimating system.

Table 2.1 Comparison of Cost Estimating System.

System Name	Cϕ$T System
Area of Application	* The Cϕ$T System produces detailed quantitative estimates for process plants (oil, chemical, pharmaceutical).
Underlying Model	* It uses engineering costing and construction modelling techniques and incorporates the following models: 1) **equipment design and cost model** - generates a vendor material take-off for each component of each equipment item. 2) **volumetric model** - produces the engineeringdesign and material take-off for equipment handling and setting. 3) **work time model** - determines field manpower costs.
Base Organization	1) The equipment specifications are input by the user. 2) The equipment designs and costings, and a quantity take-off of bulk materials using Cϕ$T's equipment design, are generated using computer models of the project (factors are not used). 3) Field labor to install the bulk material is calculated using work items (work items relate field labor to bulk quantities at a specific production rate for each specific labor task.
Quantitative Results	* The results show quantities, man-hours, resources and costs. * The cost of using Cϕ$T System averages out to the cost of approximately 1 1/2 man-hours for each main equipment item. * Estimates can be prepared using 1/5 of man-hours normally required to do manual estimate.
Number of Installations	* The Cϕ$T System used for nearly 20 years in evaluating thousands of projects.
Required Input	* The user needs to select a base country location and provide process specifications, an outline plot plan and a flow-sheet. * A minimum number of parameters (equipment type and one or two size parameters) are required with system defaults values for unspecified ones. * The Cϕ$T System does not use factors, exponents or lists of equipment.

Table 2.1 Comparison of Cost Estimating Systems (continued).

System Name	CES
Area of Application	* CES is a system for producing manufactoring estimates.
Underlying Model	* It has a standardized program structure and unified data base external to the source code.
Base Organization	* The process begins with part and assembly drawings and operation process charts of the product. These are used to generate a parts list and a work center production sequence. * Appropriate programs are run to estimate the unit cost of raw materials given the above information. * Programs are run for each work center included in the process and these results are used to generate detailed reports. * Two databases are used: 1) **material database** - a separate file (table format) is used for each material type containing all information for each generic work center. 2) **work center database** - separate file for each generic work center containing user prompts and machine set up parameters and machine function parameters. * User can change inputs or intermediate results without having to redo the process.
Quantitative Results	* The output includes a work center detail report including the project data, user responses and the costing results. This report provides information required to generate the following: 1) **production labor/machine summary**. 2) **tooling summary**. 3) **routing summary**.
Number of Installations	* This was a graduate research project
Required Input	* The user enters a parts list and production sequence of work centers as a data file using a full screen editor. The parts list is then saved as a spreadsheet file.

Table 2.1 Comparison of Cost Estimating Systems (continued).

System Name	PREDICTE
Area of Application	* PREDICTE uses contruction time to evaluate and improve early design concepts for multi-storey buildings.
Underlying Model	* An activity list is generated showing the start and end dates and duration for each major activity group. Acitvities which elongate the overall time are the first areas to consider for optimization.
Base Organization	* Each new case in PREDICTE involves asking a series of general questions about the project. More questions are asked as information is needed. * PREDICTE performs intermediate calculations, assesses the quantity and complexity of the work available, available resources, methods used, sequence of operations and the optimal utilization of resources.
Quantitative Results	* The system decides on possible construction methods and their sequence while assessing available resources. It determines the duration of the project and explains how the results were achieved. * PREDICTE contains some 223 questions, but only 100 to 140 are usually asked. * It requires approximately 30 minutes to complete if **all** information is available. * Returned estimates are closer to achieved results than normally expected for recently completed projects.
Number of Installations	* The system is used solely by Civil & Civic, a subsidiary of Lend Lease Corporation. It was in place for less than 2 years (1985-7) at the time of article publication.
Required Input	* The user answers questions about the project regarding location, size, shape, appearance, ground conditions and surroundings.

2.3 **Defining a General-Purpose Cost Estimate Model**

Flexibility is of utmost importance when designing a cost estimate model. It must be general enough so it can be easily applied to various projects, yet it should permit the incorporation of data specific to a particular application. As previously stated, the research for this thesis centers on electrical engineering costing although the model may be employed in the civil and mechanical fields of engineering. Discussions with individuals familiar with cost estimating resulted in the identification of a general-purpose, multi-level costing process. The data required and the accuracy of the resulting estimate information is dependent upon the availability and detail of data, time restraints, project complexity and the importance and monetary value placed on the project.

The cost estimating process is divided into five levels. At each level, a valid cost estimate may be produced. However, the accuracy and detail of the estimate is dependent upon the amount of available information at that particular stage in the estimating process. The levels are defined as follows:

1. **Class 1** - At the preliminary class the cost estimators use a percentage of the previous similar project's cost for the new project. Depending upon time and budget constraints, this stage may or may not be used.
2. **Class 2** - Preliminary drawings are required at this class. Unit cost estimates are involved in the calculation of motor costs, input/output drawings, fire alarms and lighting per square metre of building.
3. **Class 3** - A plant layout design including location of cable trays is drawn up next. Details regarding power are finalized and manufacturers provide quotes for major equipment. The cost estimate produced at this class is required when going to tender.
4. **Class 4** - Following the tender a more detailed design is generated and actual costs are determined.
5. **Class 5** - Final cost figures are determined upon completion of the project.

Although all five will be discussed, the concentration of this thesis is on classes 1, 2 and 3.

2.3.1 Class 1 Estimate

A cost estimate produced at the class 1 stage, as stated at the beginning of this section, is based on actual figures from previous similar projects. Each project is identified by the type of plant being built, its location, the client and a brief description of the plant's function. A plant may be broken down into three main components:

1. building(s).
2. infrastructure.
3. land.

The type of plant being constructed determines the number of buildings required. For example, a food processing plant may be enclosed in one building with different areas designated for each of the plant's functions. A pulp mill is often comprised of several buildings. The building or buildings are assigned a monetary value based on their size and complexity of design. Factoring values may be applied to this amount to account for increases or decreases in manufacturer's prices, or a unit cost per square metre may be used.

The infrastructure includes site preparation (e.g., clearing, grading), roads to the plant site, utilities (e.g., hydro poles, plumbing, sewage) and other particulars not directly associated with the construction of the plant itself. Each of these items may be given a flat cost based on a sub-contractor's fee, calculated (similar to the building) using a unit cost or from an educated guess based on past experience. The land (if it has not already been purchased) may be assigned a cost or calculated using a unit cost value based on current land market prices.

2.3.2 Class 2 Estimate

Once the preliminary drawings have been prepared a class 2 estimate can be generated. At this stage the estimator is only concerned with the buildings and the infrastructure. A building's cost is broken down by the number of floors in the building. Each floor is them partitioned into various areas which are comprised of the major components in that area. In addition to a cost being associated with the major components, a time value is also assigned.

An example of this would be a food processing plant. The first floor may be entirely devoted to processing vegetables, the second floor for packaging them and the third for office space. The first floor may consist of an area for washing the vegetables, an area for preparing them for packaging (e.g., peeling, slicing) and a third area for cooking them. The area set aside for washing the vegetables may be further broken down into major components which perform the following tasks; initial rinse, leaf and stem removal and drying. The large equipment required for each of these components is determined (e.g., number of motors) and is then assigned a cost value. As well, a unit cost is given to any necessary cables or wiring (based on input/output drawings), fire alarms, lighting and heating, ventilation and air conditioning (HVAC). In addition to a cost being associated with the major components, a time value is also assigned. The infrastructure costs are carried over from class 1, but are assigned more accurate values as details concerning the plant layout and requirements are now available.

It should be noted that the above class 2 breakdown of the building or buildings, as well as the class 1 cost model, may be applied to the civil, electrical and mechanical fields of engineering. The infrastructure is usually associated with civil engineering.

2.3.3 Class 3 Estimate

The model developed in this thesis is principally aimed at a class 3 cost estimate where plant layout design drawings are available. It should be noted though, that the concepts and cost model outlined for this stage, similar to the class 2 estimate, may be applied to other fields of engineering. The cost breakdown and any details such as materials or units of cost will change, but the overall concepts remain the same. A class 3 estimate is generally used for tendering purposes and requires detailed plant layout drawings. It identifies the location of motor control centers (MCC), the motors in each MCC, cable trays, switches and devices in the plant. If this information is readily available then costs can be associated with these items. Sometimes only basic details about the plant are known at the time of tendering. In this situation, an overall cost may be assigned to a plant area or major components based on previous similar estimates or carried over from the class 2 estimate.

The cost estimate generated at this stage follows the same breakdown outlined in the previous section. Each building or buildings may have one or many floors. Each floor is made up of plant areas and each area may contain one or more motor control centers (MCC). The total cost assigned to each MCC is determined by the number of motors and devices required and the costs associated with them. This information is provided for the user from an equipment list and a device list. The equipment list contains MCC's and all motors, heaters, fuses, cables and any other large current drawing equipment and components connected to any MCC. The device list includes the various switches and control mechanisms not directly associated with motors or heaters and anything not listed as equipment.

2.3.4 **Class 4 And Class 5 Estimates**

Once the class 3 estimate has been submitted for tender and the bid has been accepted, a more detailed design is produced. At the same time, more detailed costs are assigned to each of the plant areas and their components. Manufacturers are consulted for price lists and material quantities are determined.

The final stage is class 5. The project is complete and final cost figures for the entire plant are used to generate a final cost document. This may then be compared to the class 3 cost estimate to determine how closely the actual project cost matched the estimated cost.

2.3.5 **Summary**

Figure 2.6 shows the processes required to generate class 1, 2 and 3 estimates. A class 1 estimate, as stated in section 2.3.1, is based on a previous estimate of a similar project. It requires only a preliminary design. Class 2 and 3 estimates require the preliminary and plant layout designs. Class 2 uses only the equipment list and costs are assigned to the large equipment items. Any necessary cable, wiring or other devices have a unit cost associated with them. The class 3 estimate uses both the equipment and device lists. It is more accurate than the class 2 estimate because it uses actual manufacturer's quotes. Both classes 2 and 3 access various knowledge bases containing client preferences, unit costs, general information and electrical code standards. This ensures accuracy while accommodating a particular client's preferences for a specific manufacturer. Further discussion of the inputs for cost estimation are contained in chapters 3 and 4.

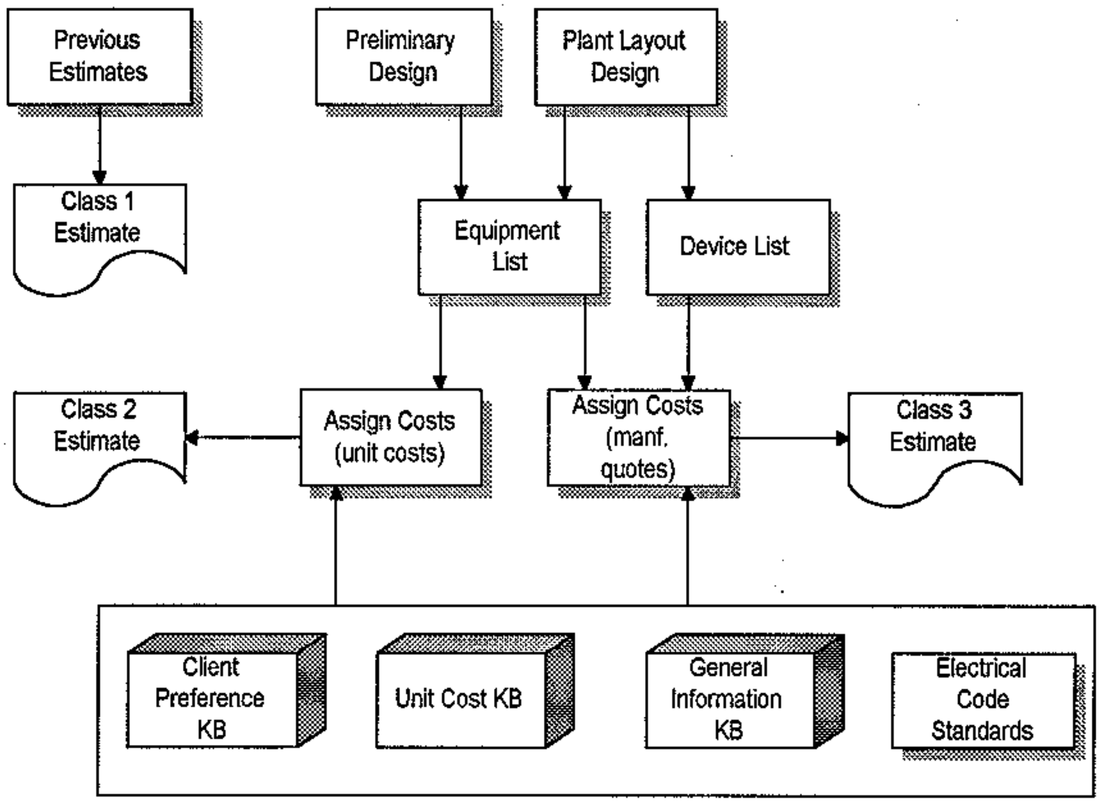


Figure 2.6 Process for producing class 1, 2 and 3 cost estimates.

3 COST ESTIMATING KNOWLEDGE REPRESENTATION

Neill and Gunter Ltd., Fredericton NB, was involved in the design and partial development of a prototype cost estimating expert system. The domain expert directly involved with the thesis was very experienced with preparing cost estimates and provided some of the basic design criteria. The majority of estimates are prepared manually. However, this individual produced equipment and device lists using Lotus 1-2-3 prior to generating a cost estimate. Therefore, one of his requirements was that these lists be used as inputs to the system and remain in the Lotus 1-2-3 spreadsheet format. This involved integrating the prototype with Lotus 1-2-3. The resulting cost estimate was also to be stored as a spreadsheet file to facilitate producing financial summaries. This chapter represents a design for a new cost estimating system which incorporates these criteria [Gesner, 1992].

3.1 Design for Representing Cost Estimate Information

An effective method of representing cost estimating knowledge in a readily accessible format is needed to produce reasonably accurate cost estimates for project design. Many different types of information from various sources must be compiled together to provide a basic template for generating estimates. The representation must be flexible enough to permit customization yet still provide the user with the necessary expert knowledge for making good design decisions. One of the objectives of this thesis is to determine if a prototype expert system can be designed to produce accurate cost estimates and permit modifications to its knowledge bases. This knowledge includes client preferences, general information and various external sources of information. The client preferences are those attributes over which the client has complete control such as choice of manufacturer or size of equipment, while general information encompasses knowledge determined by the estimator's expertise and the field of engineering or construction involved. External sources of information may include unit cost databases or design codes and standard tables.

Figure 3.1 is a conceptual design for a cost estimating system which meets the requirements mentioned above. The basic components are the knowledge base, cost estimating engine and the global database. The notion of a cost estimating engine is important. It is intended to contain the kernel functionality necessary to generate the first three classes of cost estimate mentioned in the previous chapter, given the necessary inputs. A user interface is an important part of any system designed for human interaction. Cost estimates are the direct output of the system, and can be used as input to other cost estimates.

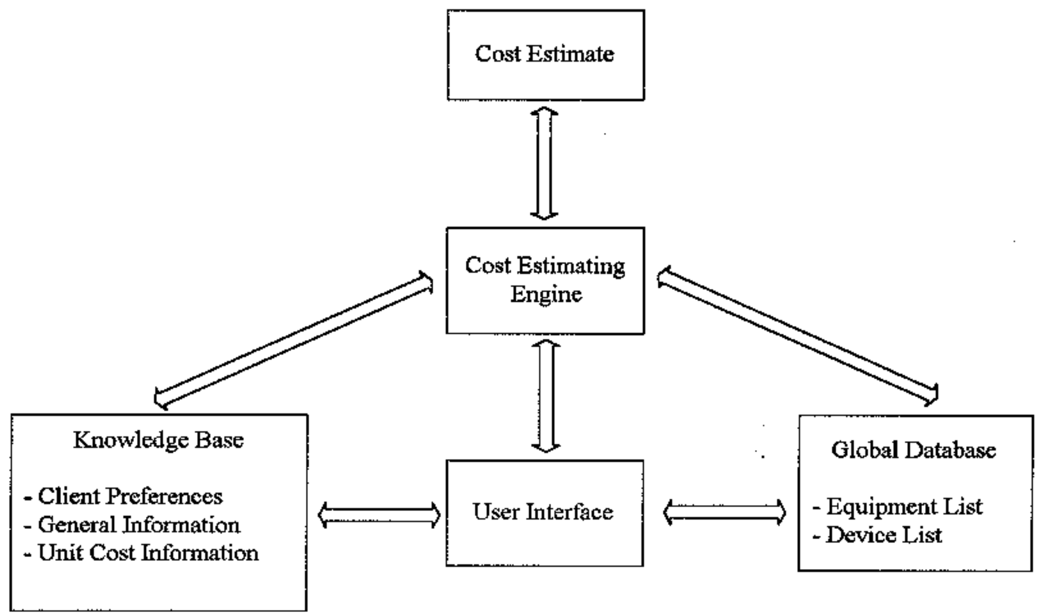


Figure 3.1 Conceptual design.

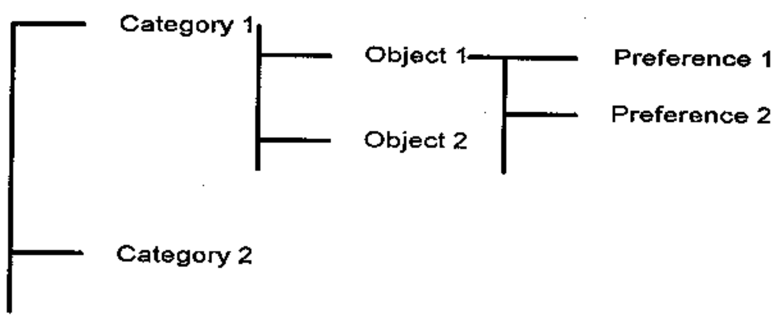
3.2 Incorporating Client Preferences

The cost estimating process is highly dependent upon the type of project and the amount of available information. Software designed to perform this task must be flexible [Lee and Ebeling, 1987].

> Central to the automation of the cost estimating function is the relative ease with which the user can customize the application software to a unique set of materials, machines and methods of organization.

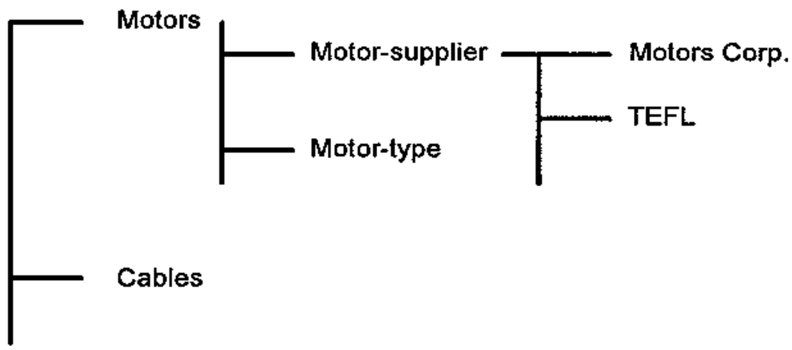
The key to a cost estimating package's flexibility is its ability to include a particular client's preferences in the decision-making processes involved in generating a cost estimate. This ensures that information concerning a client's choice of various pieces of equipment (e.g., motors, heaters, fuses or cables), devices (e.g., switches and control mechanisms), suppliers, item sizes, quantities and even labeling procedures is current and reflects that particular project's constraints.

Client preferences are represented in the knowledge base as a selection of possible choices available to the estimator, and as a collection of rules that perform specific actions provided certain criteria are met. The structure of the database is detailed in Figure 3.2. The preference file is stored in ASCII format and the naming convention is the client's name (or some abbreviation) with the extension **.dat**. Each category is subdivided into sub-categories referred to as objects with each object containing a preference. Normally only one preference is assigned to an object, although situations do arise in which the client will accept alternatives. In this case rules in the client preference rulebase are applied to determine when a second-choice preference is to be used.



a) Hierarchial organization of client preference objects.

Figure 3.2 Cost object representation.



b) Example of preferences for a specific client.

Figure 3.2 Cost object representation (continued).

All possible category, object and preference values currently existing in the sytem are stored in a separate global database. A sample of the global preference file is shown in Figure 3.3 (see Appendix I for the entire global preference file). Whenever a user wishes to update a client's preferences, selections are made from this database. When a new client is added to the system a default file of preferences is assigned to that client and saved under the client's name. This file contains all categories and objects found in the global preference database, but only one preference entity from the selection of possible preferences. This single value may be the preference most frequently chosen by clients, the first preference under each object or an estimator's personal choice based on past experience. The global and default preferences may be updated in the same manner as the client preferences files. The one difference is that new categories and objects, in addition to new preferences, may be added to the database. Figure 3.4 outlines the processes for modifying the global, client and default preference files.

C 2.0 MCC
O 2.1 MCC-supplier
P Allen-Bradley
O 2.13 MCC-voltage
P 360

Figure 3.3 Sample of global preference knowledge base file.

C 3.0 motors
O 3.1 motor-supplier
P Toshiba
O 3.2 motor-type
P TEFL
O 3.3 motor-HP
P 1
O 3.4 motor-voltage
P 575

Figure 3.3 Sample of global preference knowledge base file (continued).

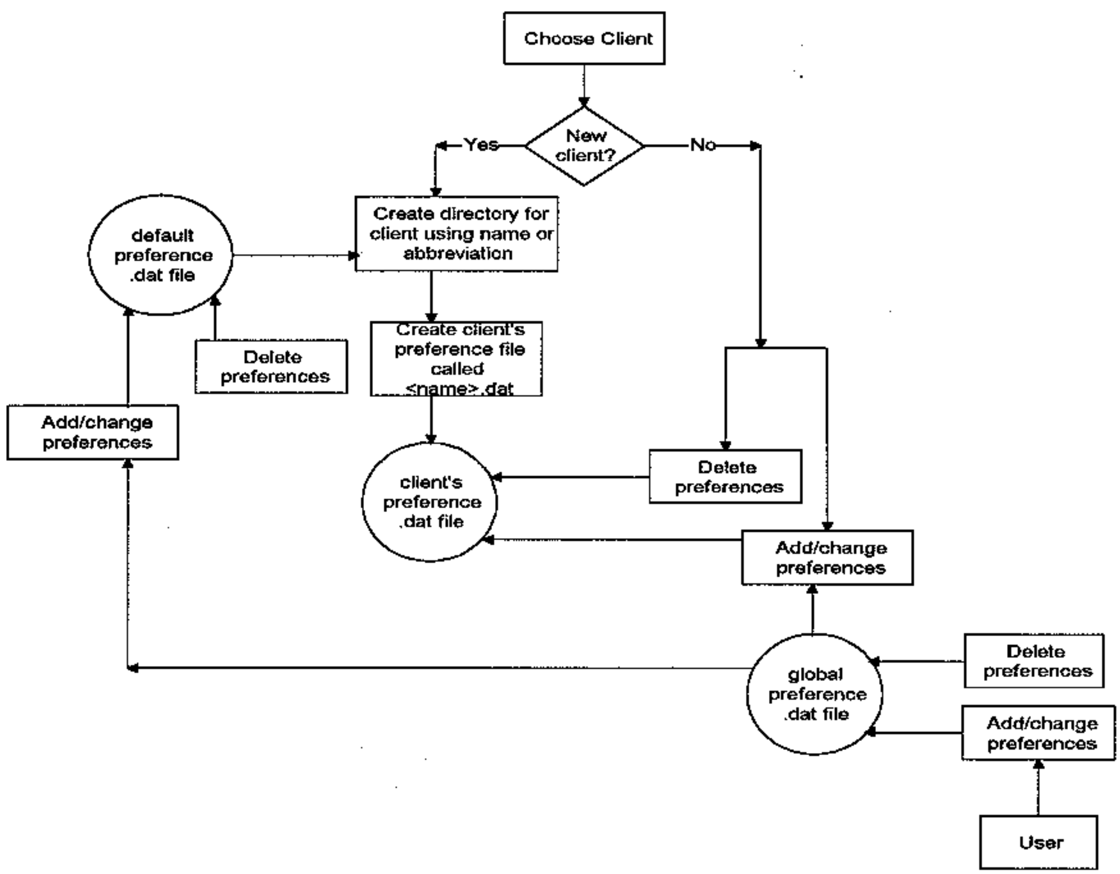


Figure 3.4 Processes for modifying default and client preference files.

As previously mentioned a rulebase of preference rules is used to determine when and where to apply specific preference values. For example, a client may choose to purchase equipment from a certain manufacturer for a project because the items are available in odd sizes or mass quantities. For another type of project the same client may wish to deal with a manufacturer who produces custom-designed equipment. Project location is another factor that is used to determine the particular preference values to be used. Examples of these types of rules are shown in Figure 3.5. Appendix II details the context-free grammar developed to represent the rules. It should also be noted that the rulebase, similar to the preferences database, can be modified to reflect a client's choices and available manufacturers (see Appendix III for a complete listing of client preference rules).

This arrangement provides junior engineers and consultants unfamiliar with available suppliers, equipment sizes and types and other equipment details with expert knowledge gained from experienced estimators. Providing this information permits the client to change their preferences or use alternative aids in producing more accurate and up-to-date cost estimates.

```
(client-rule-1:
        (IF      (= client "IPPL"))
        (THEN  (AND (= MCC-voltage 600)
                      (= MCC-supplier "Allen-Bradley"))))

(client-rule-2:
        (IF      (AND (= client "IPPL")
                      (= area "corrosive"))
        (THEN  (= cable-cover "corrosion boot")))
```

Figure 3.5 Sample of client preference rules.

```
(client-rule-5:
        (IF     (= client "IPPL"))
        (THEN  (AND (= LV-instrument-cable-supplier "Dekoran")
               (= T/C-instrument-cable-supplier "Dekoran")
               (= RFD-instrument-cable-supplier "Dekoran"))))
```

Figure 3.5 Sample of client preference rules (continued).

3.3 **General Information**

General information includes both essential and non-essential inputs. Those items considered to be essential tend to greatly influence an estimate's bottom-line values, while non-essentials are used to further clarify project details. Regardless of the class of importance assigned to these inputs, the main purpose is to improve an estimate's accuracy during the decision-making stage of a project.

Some sources of this information are as follows:

1. Individuals experienced in generating estimates can provide expertise regarding the best materials to use, the minimum requirements to be met and overall information for improving design efficiency and accuracy.
2. Previous cost estimates of similar projects (i.e., similar in size and type) can be used as a base for a class 1 estimate.
3. The initial description and functionality of a plant helps determine the different areas found in the plant, the equipment required for each area, as well as the amount of cable, number of switches and other devices needed to complete the project.
4. Basic estimating rules and calculations are used to determine necessary materials and their quantities.
5. Any in-house rules or factoring values applied to account for current economic fluctuations and price ranges based on a project's location.
6. Labor costs calculated on an hourly basis or by the total work required to perform a particular task must also be included.

7. Design tables from electrical code standards and from manufacturer provide equipment specifications, as well as necessary installation information.
8. Different regions and countries have location-specific equipment design and installation rules which must also be considered.

The previous list provides an overview of the information required to produce a cost estimate. Regardless of the type and source of input, it is represented in a similar fashion to a client's preferences. Rules (applied when certain conditions are met), various lists and tables, and even undocumented information (i.e., conversations among estimators) are used to provide further design details. Once again the accuracy of the resulting estimate is directly related to the amount and type of information available to the estimator. The following sections further explain the formats and sources of information identified in the cost model outlined in Chapter 2.

3.3.1 Using Rules to Represent General Information

The format of the rulebase representing general information is similar to the client preferences rules. The main difference is the type of information in the rules. General information applies to all projects for all clients, while client preferences are restricted to all projects for a particular client. As well, the rules can be modified to incorporate new information resulting from economic fluctuations, changes in technology or advances in an estimator's expertise. Samples of these types of rules are provided in Figure 3.6 (see Appendix IV for the entire general information rulebase).

```
(motor-voltage-rule-1:
        (IF     (= project-location "CANADA"))
        (THEN  (= motor-voltage 575)))

(motor-voltage-rule-2:
        (IF     (= project-location "EUROPE"))
        (THEN  (= motor-voltage 380)))
```

Figure 3.6 Example of general information rules.

```
(motor-voltage-rule-3:
        (IF       (= project-location "USA"))
        (THEN  (= motor-voltage 480)))
```

Figure 3.6 Example of general information rules (continued).

3.3.2 General Purpose Information

In addition to the rulebase, there are also many simple text files containing information required to produce a cost estimate. Some of the files contain required information such as a project's title, identification number, scope and the client's name. Others only provide sufficient data as determined by the project's stage of development and the class of cost estimate to be produced (see section 2.3).

Some examples of these files are described here:

1. **Work Order Sheet Information** - Each project has a work order sheet detailing its title, identification number and the client's name.
2. **Equipment List** - This list contains items necessary to complete the project. In the case of electrical design, this list includes motor control centres (MCC), motors, heaters, fuses and other large current drawing equipment and components connected to an MCC. This information is derived from site plans or from previous projects of the same type (see Figure 3.7).
3. **Device List** - This is independent of the equipment list. With respect to electrical design, it contains the various switches and control mechanisms necessary for the plant to function and comes from the same sources as the equipment list (see Figure 3.8).

LINE NO.	DESCRIPTION	REV NO	CABLE NO.	FROM
LINE #1	SPREAD VIB., AMB. PRECOOL NO.1		09M001	MCC 46
LINE #1	BELT DRYER, AMB. PRECOOL NO.1		09M002	MCC 46
LINE #1	FAN #1, ANB. PRECOOL NO.1		09M003	MCC 46
LINE #1	FAN #2, ANB. PRECOOL NO.1		09M004	MCC 46
LINE #1	FAN #3, AMB. PRECOOL NO.1		09M005	MCC 46
LINE #1	FAN #4, AMB. PRECOOL NO.1		09M006	MCC 46
LINE #1	FAN #5, ANB. PRECOOL NO.1		09M007	MCC 46

Figure 3.7 Part of a sample Lotus 1-2-3 equipment list spreadsheet.

REV NO	CONTROL PANEL	DEVICE NUMBER	I/O ASSIGN.	SERVICE	DEVICE TYPE	MANUF.
2	CP40-1	1Y01TAL001	I:000/01	BUILDING ROOF DRAIN THERMOSTAT	THERMOSTAT	
2	CP40-1	1U23JS001	I:000/02	I.Q. DATA PLUS KILOWATTHOUR PULSE	THERMOSTAT	
2	CP40-1	1U23XA001	I:000/03	I.Q. DATA PLUS ALARM	THERMOSTAT	
2	CP40-1	1U02TSH001	I:000/04	LCC #4 TRANSFORMER HIGH TEMPERATURE ALARM	TEMP SWITCH	
2	CP40-1	1U02TSHH002	I:000/05	LCC #4 TRANSFORMER HI/HI TEMPERATURE ALARM	TEMP SWITCH	
2	CP40-1	1Y24TSH001	I:000/06	LCC #4 ROOM HIGH TEMPERATURE ALARM	TEMP SWITCH	
2	CP40-1	1U07TAH001	I:001/00	FIRE ALARM ZONE 1, POLY SHED AISLE IN BASEMENT	FIRE AL. PANEL	EDWARDS

Figure 3.8 Part of a sample Lotus 1-2-3 device list spreadsheet.

3.4 External Information Sources

The integration of electronic cost-estimating databases with a cost estimating system provides the estimator with up-to-date unit cost values and ensures improved accuracy. Two common unit cost databases discovered in the course of this research include:

1. The Means Database, which is the electronic version of the unit price values tabled in the Means Construction Cost Data book. It is regularly updated through new releases [Mahoney, 1987].
2. The Catalogue is the electronic version of Yardsticks for Costing. It provides cost data for the Canadian construction industry [Gretton, 1988].

Although both unit cost databases provide accurate costing data, they do not contain local unit cost prices. Therefore, it is useful to maintain a database of material costs and labor rates based on the cost estimator's previous experience with manufacturers.

The information stored in the database is arranged in the same categories and sub-categories as the global preference database. Additional sub-categories are used to associate a unit cost, type of laborer and labor class (if applicable), date of entry, amount of time to install the item (in hours) and source of information (e.g., manufacturer, in-house estimate). Figure 3.9 outlines the format of the unit cost file using the grammar developed in chapter 4. It is stored in ASCII format using the client's name (or some abbreviation) and the file extension **.cst**. When new categories and sub-categories are added to the global preference database, the same structure is created in the unit cost database with the additional unit cost sub-categories.

```
(1) <category>        ::= <object>+
                          <labor-info>
                          <entry-date>
                          <quote-source>
```

Figure 3.9 Unit cost object representation.

(2) <object>	::=	<string>
(3) <labor-info>	::=	<labor-cost/unit> <laborer-type> <labor-class>
(4) <labor-cost/unit>	::=	<currency> <floating-pt>
(5) <laborer-type>	::=	<string>
(6) <labor-class>	::=	<string>
(7) <entry-date>	::=	<year> **"-"** <month> **"-"** <day>
(8) <year>	::=	<integer>
(9) <month>	::=	**"1" | "2" | "3" | "4" | "5" | "6" |** **"7" | "8" | "9" | "10" | "11" | "12"**
(10) <day>	::=	**"1" | "2" | "3" | "4" | "5" | "6" |** **"7" | "8" | "9" | "10" | "11" | "12" |** **"13" | "14" | "15" | "16" | "17" | "18" |** **"19" | "20" | "21" | "22" | "23" | "24" |** **"25" | "26" | "27" | "28" | "29" | "30" | "31"**
(11) <quote-source>	::=	**"actual" |** **"estimate" |** **"quote"**

Figure 3.9 Unit cost object representation. (continued)

Once all unit and labor costs and time values are assigned to the equipment and device list items, rules are used to calculate the cost per item, per plant area and overall cost of the electrical portion of the estimate. The output is an estimate in Lotus 1-2-3 spreadsheet format.

4 COST ESTIMATE MODEL

In order to successfully acquire and represent knowledge for an expert system, one must be able to conceptualize its domain. This involves abstracting the key concepts, attributes, values and heuristics represented in the domain. If it is not done properly it can have a negative impact on future knowledge acquisition, as well as on the resulting expert system. It is one of the more crucial tasks in the knowledge acquisition process. The benefits of generating an adequate domain conceptualization include [McGraw and Harbison-Briggs, 1988]:

1. A better understanding of the knowledge domain.
2. A way of estimating knowledge acquisition needs and plans.
3. Ideas for designing the knowledge base.

Initially the problem domain appears as a group of unrelated terms, objects and applications. The first task is to identify the basic categories of knowledge and their features. The result is a mental map of the domain which provides a basis for further knowledge acquisition. As the initial framework is decomposed, primary subsets of information and interrelationships become visible. Those areas requiring more in-depth analysis can be represented graphically [McGraw and Harbison-Briggs, 1988].

4.1 **Overview of the Cost Estimate Model**

The cost estimate knowledge discussed in chapter 3 can be further decomposed. Figure 4.1 shows the overall model of a class 1, 2 or 3 estimate. The plant, the infrastructure and the surrounding land are each costed separately. Although this thesis is concerned primarily with the plant itself, the format used is generic enough to be applied to other types of cost estimates.

The plant is broken down into its components (i.e., buildings, land) and each has an associated cost. The required inputs and type of estimate produced is dependent upon the field of engineering and the amount of detail required. Figure 4.1 shows the breakdown of class 1, 2 and 3 (section 2.3) electrical estimates for a plant. Class 1 requires a previous estimate as input, class 2 uses an equipment list and class 3 incorporates both an equipment list and a device list.

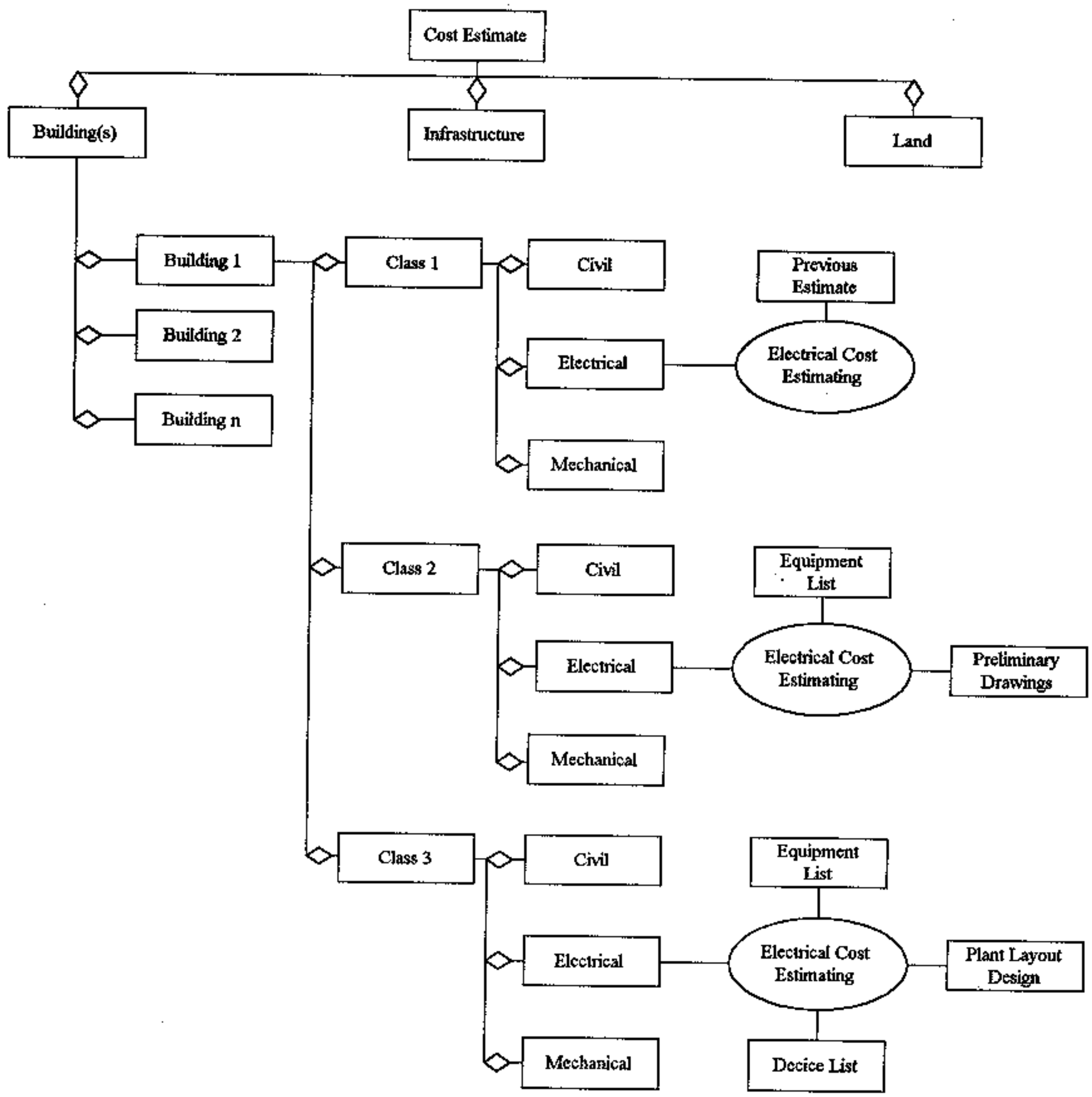


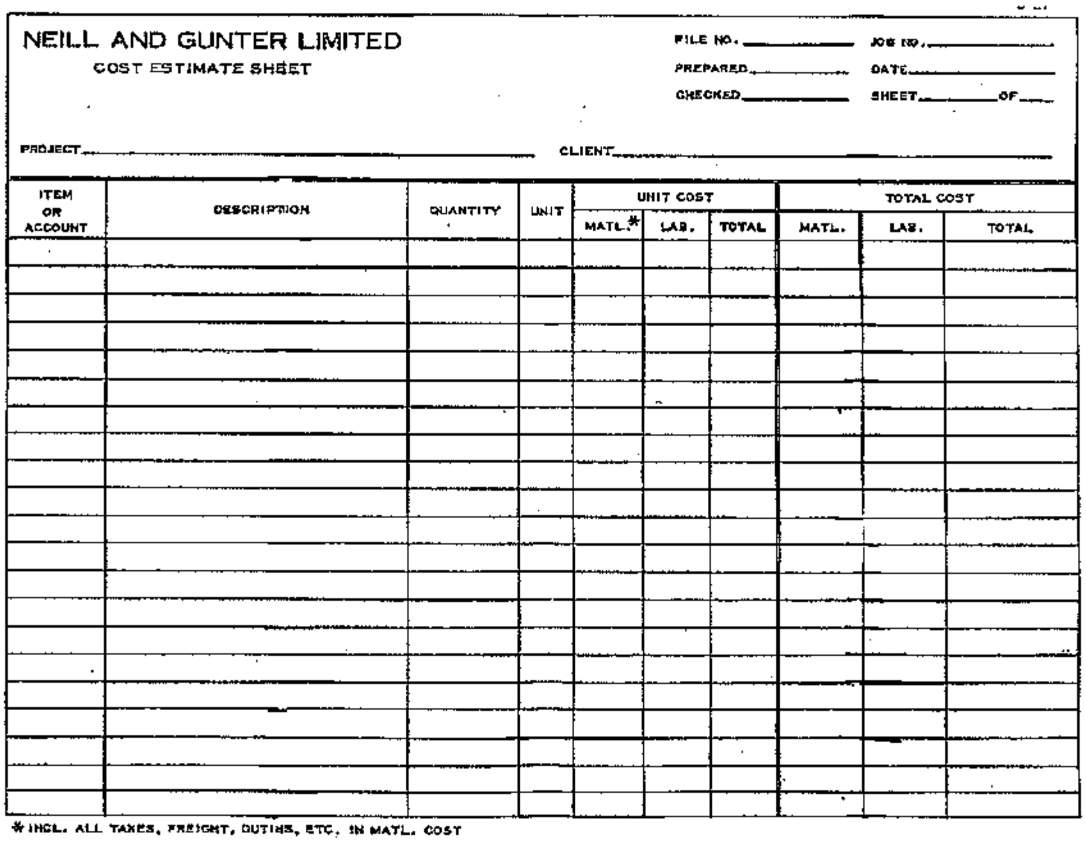
Figure 4.1 Cost estimate model for producing class 1, 2 and 3 estimates.

4.2 Cost Estimate Definition

The cost estimate grammar detailed in this section can be applied to the first three classes of estimates. It outlines the format of an electrical estimate for all buildings at a plant. The estimate is broken down into levels 1, 2 and 3:

1. Level 1 contains the summary information for the main components in each building (e.g., MCC, unit substation, power distribution panel).
2. Level 2 is a breakdown of level 1 into its sub-components (e.g., feed to MCC, cable tray).
3. Level 3 details the basic parts contained in each sub-component. For example, a starter requires a pilot light, a surge suppressor and a nameplate.

Each level has a description, quantity, unit cost information and total cost information associated with it. An estimate may incorporate all three levels or simply be a summary of a building's total costs. The complete grammar for a cost estimate definition is given in Figure 4.3. The grammar is written in Extended Backus Naur Form **(EBNF)** [Object Management Group, Inc., 1992]. It was developed from a cost estimate form as shown in Figure 4.2.


NEILL AND GUNTER LIMITED
COST ESTIMATE SHEET

FILE NO. ______ JOB NO. ______
PREPARED ______ DATE ______
CHECKED ______ SHEET ______ OF ______

PROJECT ______ CLIENT ______

ITEM OR ACCOUNT	DESCRIPTION	QUANTITY	UNIT	UNIT COST			TOTAL COST		
				MATL.*	LAB.	TOTAL	MATL.	LAB.	TOTAL

* INCL. ALL TAXES, FREIGHT, DUTIES, ETC. IN MATL. COST

Figure 4.2 Cost estimate form.

(1) <cost-estimate> ::= **"ce-total"** <project-id> <currency> <floating-pt > <ce-rest>

(2) <ce-rest> ::= <tot-build-cost> **";"** <infrastructure-cost> **";"** <land-cost>

(3) <tot-build-cost> ::= **"building-total"** <currency> <floating-pt > <build-cost>$^+$

Figure 4.3 Context-free grammar for a cost estimate.

```
(4) <infrastructure-cost>  ::= "infrastructure-cost"
                               <currency> <floating-pt>
(5) <land-cost>            ::= "land-cost"
                               <currency> <floating-pt>
(6) <build-cost>           ::= <class1-cost>
                               | <class2-cost>
                               | <class3-cost>
(7) <class1-cost>          ::= "building-cost" <build-id>
                               <currency> <floating-pt>
                               <building-size> <building-type>
(8) <class2-cost>          ::= "building-cost" <build-id>
                               <currency> <floating-pt>
                               <building-size> <building-type>
                               "("
                               <civil-build-cost> ";"
                               <elec-build-cost> ";"
                               <mech-build-cost> ")"
(9) <class3-cost>          ::= "building-cost" <build-id>
                               <currency> <floating-pt>
                               <building-size> <building-type>
                               "("
                               <civil-build-cost> ";"
                               <elec-build-cost> ";"
                               <mech-build-cost> ")"
(10) <build-id>            ::= <string>
(11) <building-size>       ::= "building-size" <floating-pt> <area-units>
                               <cost/area-unit>
                               "floors" <no-of-floors>
(12) <area-units>          ::= "sq. metres"
                               | "sq. feet"
(13) <cost/area-unit>      ::= <floating-pt>
```

Figure 4.3 Context-free grammar for a cost estimate (continued).

(14) <no-of-floors>	::=	<integer>
(15) <building-type>	::=	"food processing"
		| "power generating station"
		| "saw-mill"
(16) <elec-build-cost>	::=	**"elec-build-cost"** <currency> <floating-pt>
		<eb-cost-rest>*
(17) <eb-cost-rest>	::=	<level-1-cost>$^{+}$
(18) <level-1-cost>	::=	<item-no>
		<item-description>
		<unit-cost>
		<total-cost>
		[<level-2-cost>$^{+}$]
(19) <level-2-cost>	::=	<item-no> **"."** <sub-item-no>
		<item-desc>
		<unit-cost>
		<total-cost>
		[<level-3-cost>$^{+}$]
(20) <level-3-cost>	::=	<item-no> **"."**
		<sub-item-no> **"."**
		<sub-sub-item-no>
		<item-desc>
		<unit-cost>
		<total-cost>
(21) <item-no>	::=	<integer>
(22) <sub-item-no>	::=	<integer>
(23) <sub-sub-item-no>	::=	<integer>
(24) <item-desc>	::=	<description>
		<quantity>
		<unit>
		<unit-man-hours>

Figure 4.3 Context-free grammar for a cost estimate (continued).

(25) <unit-cost> ::= <material-cost/unit>
<labor-cost/unit>
<total-cost/unit>

(26) <material-cost/unit> ::= <currency> <floating-pt>
| <material-c/unit-var>

(27) <material-c/unit-var> ::= <alpha>$^+$ <alpha-numeric>*

(28) <labor-cost/unit> ::= <currency> <floating-pt>
| <labor-c/unit-var>

(29) <labor-c/unit-var> ::= <alpha>$^+$ <alpha-numeric>*

(30) <total-cost/unit> ::= <currency> <floating-pt>
| <total-c/unit-var>
| nil

(31) <total-c/unit-var> ::= <alpha>$^+$ <alpha-numeric>*

(32) <total-cost> ::= <tot-material-cost>
<tot-labor-cost>
<tot-cost>

(33) <tot-material-cost> ::= <currency> <floating-pt>
| nil

(34) <tot-labor-cost> ::= <currency> <floating-pt>
| nil

(35) <tot-cost> ::= <currency> <floating-pt>
| nil

(36) <unit> ::= **"each"** | **"lot"** | **"meter"**
| **"hour"** | **"feet"**

(37) <unit-man-hours> ::= <floating-pt>
| nil

(38) <quantity> ::= <integer>

(39) <currency> ::= **"C$"** | **"US$"** | **"A$"**
| **"£"** | **"¥"**

(40) <integer> ::= <digit>$^+$

(41) <floating-pt> ::= <digit>$^+$ **"."** <digit>*

Figure 4.3 Context-free grammar for a cost estimate (continued).

(42) <digit> ::= **"0" | "1" | "2" | "3" | "4"**
| "5" | "6" | "7" | "8" | "9"
(43) <string> ::= <character>$^{+}$
(44) <character> ::= <alpha-numeric>
| <any_character>
(45) <alpha> ::= **"a" | "b" | ... | "z"**
| "A" | "B" | ... | "Z"
(46) <alpha-numeric> ::= <alpha>
| <digit>

Figure 4.3 Context-free grammar for a cost estimate (continued)..

4.2.1 **Equipment List Definition**

The equipment list contains all large current drawing equipment and their components. It is broken down into two sub-lists:

1. **Motor list** - This contains all motors associated with a particular control panel (e.g., MCC, power distribution panel) that are used to run various pieces of equipment such as fans, conveyor belts or freezers.

2. **Cable list** - The cable list is a one-to-one matching of the motor list. Each motor has a direct electrical feed to the piece of equipment it runs.

The information associated with each item includes the production line or physical area where it is located, supplier or manufacturer, model, cost, source of quoted cost, type of equipment (e.g., horsepower for a motor, voltage for a cable) and pertinent remarks. One of the more important pieces of information mentioned above is the cost and the source of quoted cost. A cost may be one of three types:

1. **Estimate** - An estimated cost may be taken from a previous cost estimate or be an educated guess multiplied by an inflationary factor.

2. **Manufacturer's quoted cost** - This type of cost is still an estimate, but it may not include taxes and it may increase or decrease depending upon when the item is purchased.
3. **Actual cost** - The actual cost is the amount actually paid for the item.

The following sentences are a continuation of the grammar in section 4.3 and use some of the non-terminals previously defined:

(47) <equip-list> ::= <equip-item>$^{+}$
(48) <equip-item> ::= <equip-desc>
<equip-misc>
<equip-info>
<cable-info>
<motor-info>
(49) <equip-desc> ::= <line-no>
<description>
<rev-no>
<cost>
(50) <line-no> ::= <string>
(51) <description> ::= <string>
(52) <rev-no> ::= <integer>
(53) <equip-misc> ::= <supplier/manuf>
<model>
<quote-source>
<remarks>
(54) <supplier/manuf> ::= <string>
(55) <model> ::= <string>
(56) <quote-source> ::= **"actual"**
| **"estimate"**
| **"quote"**
(57) <remarks> ::= <string>

Figure 4.4 Context-free grammar for equipment list.

(58) <equip-info> ::= <control-panel>
<equip-no>
<motor-no>
(59) <control-panel> ::= <string>
(60) <equip-no> ::= <string>
(61) <motor-no> ::= <string>
(62) <cable-info> ::= <cable-no>
<volts>
<no-cond>
<cable-size>
<conduit-size>
< remarks>
(63) <cable-no> ::= <string>
(64) <volts> ::= <integer>
(65) <no-cond> ::= <integer>
(66) <cable-size> ::= <integer>
(67) <conduit-size> ::= <string>
(68) <remarks> ::= <string>
(69) <motor-info> ::= <horsepower>
<kilowatts>
<rpm> <volts>
<full-load-amp>
<safety-factor>
<space-factor>
<starter-size>
<fds-size>
<O/L-size>
<drawing-no>
<remarks>
(70) <horsepower> ::= <floating-pt>
(71) <kilowatts> ::= <floating-pt>
(72) <rpm> ::= <integer>

Figure 4.4 Context-free grammar for equipment list (continued).

(73) <volts> ::= <integer>
(74) <full-load-amp> ::= <floating-pt>
(75) <safety-factor> ::= <integer>
(76) <starter-size> ::= <integer>
(77) <fds-size> ::= <integer>
(78) <O/L-size> ::= <floating-pt>
(79) <drawing-no> ::= <string>
(80) <remarks> ::= <string>
(81) <cost> ::= <currency> <floating-pt>

Figure 4.4 Context-free grammar for equipment list (continued).

4.2.2 **Device List Definition**

The device list contains the switches and control mechanisms required to operate the various pieces of equipment found in the equipment list. Each item includes the following information: type of device, device number, equipment item controlled by the device, manufacturer, cost and pertinent remarks.

The following sentences are a continuation of the grammar in section 4.2.1 and use some of the non-terminals previously defined:

(82) <device-list> ::= <device-item>$^{+}$
(83) <device-item> ::= <rev-no>
<dev-id>
<device-desc>
<model>
<control-power>
<comments>
(84) <rev-no> ::= <integer>

Figure 4.5 Context-free grammar for device list.

(85) <dev-id> ::= <control-panel>
<dev-no>
<I/O-assign>
(86) <control-panel> ::= <string>
(87) <dev-no> ::= <string>
(88) <I/O-assign> ::= <string>
(89) <device-desc> ::= <service>
<dev-type>
<cost>
(90) <service> ::= <string>
(91) <dev-type> ::= <string>
(92) <model> ::= <manuf>
<model>
(93) <manuf> ::= <string>
(94) <model> ::= <string>
(95) <control-power> ::= <string>
(96) <comments> ::= <string>

Figure 4.5 Context-free grammar for device list (continued).

4.3 Cost Estimate Example

This section gives examples of estimates represented using the grammar from Figure 4.3. All non-terminals found in the grammar are replaced with reasonable values to simulate an actual cost estimate. Figure 4.6 shows a class 1 estimate with just the high-level costs, while Figure 4.7 provides a more detailed example of a class 3 estimate showing the various levels of costs.

In Figure 4.6 the first line provides the overall cost of the estimate including civil, electrical and mechanical costs for the building, infrastructure and surrounding land. The second line is the cost for all buildings considered to be part of the plant itself. This is further broken down into a cost for each building and includes the size and type of building. The infrastructure and land costs are listed as total costs.

```
ce-total 4752 C$ 1.570653E6
building-total C$ 1.271653E6
building-cost 4752-1 C$ 1.271653E6
building-size 3.02924E3 sq. metres C$ 4.1979E2 floors 1 food processing;
infrastructure-cost C$ 2.1E5;
land-cost C$ 8.9E4
```

Figure 4.6 Example class 1 cost estimate.

The class 3 estimate shown in Figure 4.7 follows the same format as the class 1 estimate. The main difference is the level of detail associated with the cost of each building. It is broken down into levels with the information provided by the equipment and device lists (see Figures 4.4 and 4.5). An item number is used to differentiate the various levels. A description field details the unit of measurement used, the quantity required and the man hours needed for installation. Following this are the costs per unit and total unit costs. Each of these are broken down into a material, labor and total cost (material plus labor costs). All costs are stored as floating point values and have a currency associated with them. In the provided example, the first level is **3** and the item is **1 LOT** of an **MCC #48 - RETAIL** costing **0** hours of labor per **LOT** to install. The material cost /unit, labor cost/unit and total cost/unit are **$50,804.00**, **$17,903.25** and **$68,707.25** respectively. The total material cost, total labor cost and overall total cost for the specified quantity of units are **$50,804.00**, **$17,903.25** and **$68,707.25**. The second level is **3.1** and the item is **1 LOT** of a **FEED TO MCC** costing **0** hours of labor per **LOT** to install. The material cost /unit, labor cost/unit and total cost/unit are **$7,830.30**, **$3,013.95** and **$10,844.25** with the total material cost, total labor cost and overall total cost listed as **$7,830.30**, **$3,013.95** and **$10,844.25**. The third level is **3.1.1** and the item is **45 meters** of **3x500 MCM** costing **0.28** hours of labor per meter to install. The material cost /unit, labor cost/unit and total cost/unit are **$103.00**, **$8.31** and **$111.31** and the total material cost, total labor cost and overall total cost, **$4,635.00**, **$373.95** and **$5,008.95**. All costs are in Canadian funds as indicated by the currency type **C$**. Figure 4.7 does not show the <civil-build-cost> or the <mech-build-cost> as they go beyond the scope of the thesis. They can be broken down similar to the electrical building costs. As well, an example of a class 2 estimate is not included either. Class 2 and 3 estimates are similar in format with the exception that class 2 does not include the device list.

```
ce-total 4752 C$ 1.570653E6
building-total C$ 1.271653E6
building-cost 4752-1 C$ 1.271653E6
building-size 3.02924E3 sq. metres C$ 4.1979E2 floors 1 food processing
(<civil-build-cost>;
 elec-build-cost C$ 1.271653E6
3 MCC #48 - RETAIL 1 LOT 0
C$ 5.0804E4 C$ 1.790325E4 C$ 6.870725E4
C$ 5.0804E4 C$ 1.790325E4 C$ 6.870725E4
3.1 FEED TO MCC 1 LOT 0
C$ 7.8303E3 C$ 3.01395E3 C$ 10.84425E3
C$ 7.8303E3 C$ 3.01395E3 C$ 10.84425E3
3.1.1 3x500 MCM 45 meter 2.8E-1
C$ 1.0300E2 C$ 8.31E0 C$ 1.1131E2
C$ 4.635E3 C$ 3.7395E2 C$ 5.00895E3
 <mech-build-cost>);
infrastructure-cost C$ 2.1E5;
land-cost C$ 8.9E4
```

Figure 4.7 Example of part of a class 3 cost estimate.

5 A PROTOTYPE TOOL

Although a conceptual model may appear feasible, a working model provides a better demonstration of the involved processes. As part of the requirements of this thesis, a prototype cost estimating expert system implementing the concepts outlined in earlier chapters is being developed. Although the system, **ACE** - Assistant for Cost Estimating, is to be used for electrical design, it is domain independent and can be readily adapted to other engineering fields. It operates at any of the five classes discussed in section 2.3, but is principally intended for classes 2 and 3. ACE uses preliminary design drawings and data lists (equipment and device lists) as inputs to generate a reasonably accurate cost estimate for electrical design. Client preferences, electrical design information, basic cost estimating knowledge and electrical codes and standards are encoded in knowledge bases and act as checks to ensure the validity of the design. It should be noted that the system operates under these contraints and is not responsible for ensuring the soundness of the estimate beyond these boundaries.

5.1 User Requirements

During the initial planning stages, basic software and hardware requirements for both the cost estimating system and the development environment were established. These requirements act as constraints ensuring that the system functions to its full potential.

5.1.1 Performance Requirements

Various restraints and requirements must be considered when developing the ACE system:

1. With regard to performance, 95% of all user requests should be met within 2 seconds.
2. The user should not be able to halt the system unintentionally from performing any input/output functions.
3. Only the super user and developer will be able to modify the knowledge bases of the deployed version of ACE.
4. To use the ACE system a user must have a valid userid and password.

5.1.2 Software and Hardware Environment

ACE was developed using ART-IM Version 2.1 and the WKS Library of C functions, and interfaces with Lotus 1-2-3 Release 2.01. ART-IM (Automatic Reasoning Tool - Information Manager) is an expert system shell produced by Inference Corporation, Los Angeles, California. Lotus 1-2-3 is an industry standard spreadsheet package produced by Lotus Development Corporation, Cambridge, Massachusetts. The physical development environment consists of an IBM PS/2 Model 70 (Intel 80386 processor) with 8 Mbytes of memory, or a computer of equal capabilities.

The deployed version of ACE requires an IBM PS/2, PC/AT or any other 100% compatible with an 80386 processor and 4 Mbytes of memory. ART-IM is not required to run ACE. A stand alone executable version of ACE comprised of ART-IM generated C code is created and captures the existing state of the system's environment (i.e., the current knowledge base).

5.2 Functionality of ACE

ACE was designed to perform the following functions:

1. It provides a mechanism for recording each client's preferences, storing pertinent information about projects and accounting for past cost estimate decisions.
2. The user is able to enter new equipment lists and device lists, as well as modify existing ones using ACE. The files are stored in a format readable by Lotus 1-2-3.
3. Provisions for adding new clients and projects (information taken from project work order sheets) and updating the client preference knowledge base by non-programmers are included in ACE. The security information files (i.e., user id's, passwords and user security levels) and the default and global preference files may also be modified using ACE. These files exist when ACE is first installed and contain only the basic information required for generating an estimate.
4. A job cost estimate and an engineering design cost estimate are produced in a format readable by Lotus 1-2-3 or other comparable spreadsheet package.

5. It assists experienced engineers by providing an efficient means of preparing cost estimates.
6. An electronic cost database based on previously generated cost estimates can be setup. This helps reduce the time required to determine unit cost prices.
7. Finally, ACE acts as a mentor for those unfamiliar with cost-estimating procedures by ensuring all necessary details are included.

5.3 Software Structure

The basic design of ACE is best described as a control element surrounded by various satellite knowledge bases and inputs/outputs (Figure 5.1). The shadowed boxes indicate that a file of this type exists for each project for a particular client. The filled boxes denote knowledge bases and the plain boxes represent a single file. Interaction among these entities occurs only through the prototype itself.

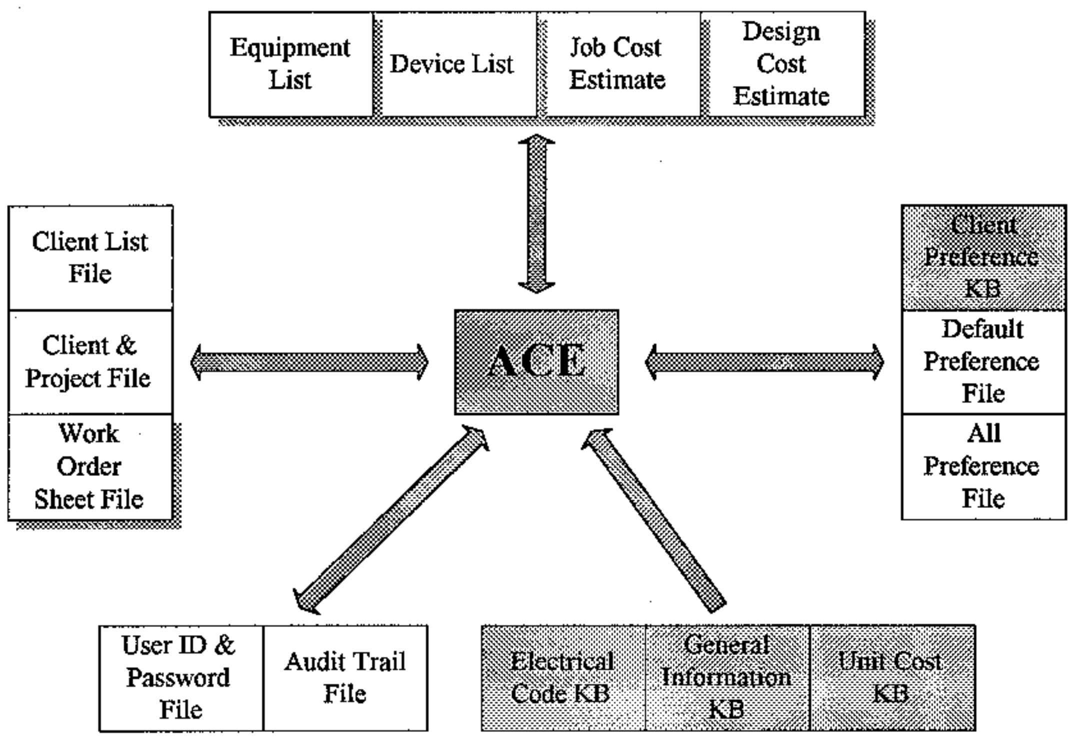


Figure 5.1 ACE Architecture.

1. **Equipment List** - In the case of electrical design, this list includes motor control centres (MCC), motors, heaters, fuses and other large current drawing equipment and components connected to an MCC, along with their power requirements (e.g., horsepower or KW, voltage, phases or current). This information is derived from site plans or from previous projects of the same type (section 3.2.2).
2. **Device List** - It is independent of the equipment list. With respect to electrical design, it contains the various switches and control mechanisms necessary for the plant to function and comes from the same sources as the equipment list (section 3.2.2).
3. **Job Cost Estimate** - This is the estimated cost of purchasing and building a plant. The equipment and device lists are the inputs and the client preferences, general information, unit costs and electrical code are used to generate an accurate estimate.
4. **Design Cost Estimate** - The design cost estimate is the cost associated with drawing up the plans for a project. The equipment list and device list detail what is found in each drawing. Generally a cost is associated with a drawing and the total amount is this value multiplied by the number of drawings.
5. **Client List File** - This file contains a list of all clients.
6. **Client and Project File** - Each client will have a file listing all of the projects and corresponding projects files (e.g. equipment list, job cost estimate and client preference file).
7. **Work Order Sheet File** - Before beginning a project, a work order sheet is prepared containing project information (e.g., client name, project number and project title). These details are used to identify each project (section 3.2.2).
8. **Userid and Password File** - All userids and passwords are stored in this file.
9. **Audit Trail File** - A record containing the user's id, file being updated, modification date and beginning and ending time of update will be written to this file each time a file is modified.
10. **EC Knowledge Base** - Various electrical code tables and rules will be encoded to ensure that all necessary standards are met.
11. **General Information Knowledge Base** - This knowledge base contains any general cost estimate rules and facts provided by cost estimating experts.

12. **Unit Cost Knowledge Base** - An electronic cost estimating database and rulebase will be created using current unit cost prices.
13. **Client Preference Knowledge Base** - Each client has a separate data file outlining their preferences. As well, various rules are used to apply this information based on current conditions. Individuals familiar with the client are consulted and the information is represented as a hierarchical structure of objects.
14. **Default Client Preferences File** - When a new client is added to the client list and client and project file, a default client preference file is created from this file.
15. **All Client Preferences File** - All possible choices for each client preference object are stored in this file.

5.4 ACE-Assistant for Cost Estimating

5.4.1 User Environment

The system is user-friendly and provides cost estimators with an efficient and accurate means of generating cost estimates. The user environment consists of three types of users:

1. **Normal ACE User** - Individuals at this level will only be able to run ACE as a cost estimating tool. They will not be permitted to modify knowledge bases.

2. **ACE Super User** - The super user will have all of the normal user's capabilities, but will also be permitted to make additions, changes or modifications to the knowledge bases.

3. **ACE Developer** - The developer will be involved in the initial development of ACE. Only individuals familiar with ART-IM and ACE will be permitted to make modifications to the system. Changes at this level will involve altering screen and file layouts, and updating ART-IM code.

5.4.2 **User Interface**

The user interface is one of the more important aspects of ACE, particularly if the system's users are not familiar with its screen layout or the procedures followed to generate a cost estimate. It is essential that the design of the interface permit the user to produce an estimate more efficiently while decreasing the chances of errors. The user should always be able to backtrack to a previously completed level if problems arise while generating or editing an estimate. Strict security measures must be maintained to ensure the validity of the knowledge bases and the system should not abend during the user's session unless so specified by the user [Gesner, 1992].

This section describes ACE as it currently exists. As with all prototypes, improvements both in design and functionality are essential to enhance usability and performance. Section 5.4.3 outlines how ACE could be better organized to make it more user-friendly and ensure that it is consistent with the cost estimate definition in chapters 3 and 4.

The ACE system is **DOS**-based (**D**isk **O**perating **S**ystem) invoked by typing the command **"ACE"** at the c: prompt. Once the system is completely loaded a dialog box containing information about ACE appears permitting the user to continue or exit. If the user selects **OK**, then he/she is required to enter a userid and password. Valid logins are stored in a file called **userpass.fil** and include a field indicating the user's level of accessibility within the system (section 5.4.1). The menubar that appears on the screen after the user has logged in the system contains the following items: **New Plant, Cost Estimate, Client Preferences, Rules** and **Options**. This is illustrated in Figure 5.2.

5.4.2.1 **Cost Estimate Menu Item**

The **Cost Estimate** menu item permits the user to choose **Civil, Electrical,** or **Mechanical** as the field of engineering, or **All Inclusive**. As ACE is initially concerned with electrical design, only the **Electrical** sub menu is operational. It produces a new menubar allowing the user to choose **Classes 1, 2** or **3**. These represent the classes of estimates that can be produced (classes 4 and 5 will be implemented as future work). An **Options** sub menu is also provided. The options available are listed below:

Figure 5.2 ACE main menu.

Class 1	The sub menus are **Building, Infrastructure** and **Land**. A total cost value is entered for each using a dialog box.
Class 2	The sub menus are **Building, Infrastructure** and **Land**. Total cost values are entered for **Infrastructure** and **Land** using dialog boxes. The **Building** displays a new menubar with sub menus enabling the user to enter equipment costs for a plant.
Class 3	The sub menus are **Building, Infrastructure** and **Land**. Level 3 includes both equipment and device costs. **Building** displays a new menubar with the same sub menus as **Class 2**.
Options	The **Options** sub menu provides miscellaneous items.
Exit	System exits to DOS.
Previous Menu	The user is returned to the previous menubar.
Information	A dialog listing information about ACE is displayed.

The **Building** sub menu under **Class 3** displays another menubar containing the following sub menus:

Equipment	It displays a dialog box to capture total equipment costs.
Structure	The plant's structural costs are divided into: **Design, Materials** and **Labor**. **Design** and **Labor** costs are entered using dialog boxes. **Materials** required are obtained from the equipment list (levels 2 and 3) and the device list (level 3 only). A new menubar is displayed permitting the user to view and modify a particular project's files (i.e., equipment list, work sheet).
Options	The **Options** item provides the same options as in the **Electrical** menubar.

The **Materials** menubar contains the menu items used to load, edit, save and view a client's project files.

Client	**Client** displays a list of all current clients.
File	It provides options for accessing, modifying and deleting files.
<u>Projects</u>	**Projects** loads project numbers associated with a client.
<u>Save</u>	It saves modifications to a project
<u>Delete</u>	**Delete** removes files from the appropriate directory.
<u>Print</u>	**Print** produces a hardcopy of the current file.
Edit	It contains the **Projects** sub menu.
<u>Projects</u>	**Projects** displays a list of project numbers and associated files. Files selected with the **.wrk** extension (work files) produce a dialog box containing the project title and number and client. Only the project title and client can be changed. The **.eq** extension indicates that the file is an equipment list. A new menubar appears with the menu items **Add, Modify, Delete** and **Options**. **Add** and **Modify** produce a form containing data fields for entry of new equipment items or modifications to existing ones. **Delete** removes equipment

list items. A class 2 equipment list contains motors used in the project, while class 3 lists both the motor and cables.

View — **View** permits the user to **browse** files (see Figure 5.3).

Options — This permits the user to **Exit** the system, return to the **Previous Menu** or display **Information** about ACE.

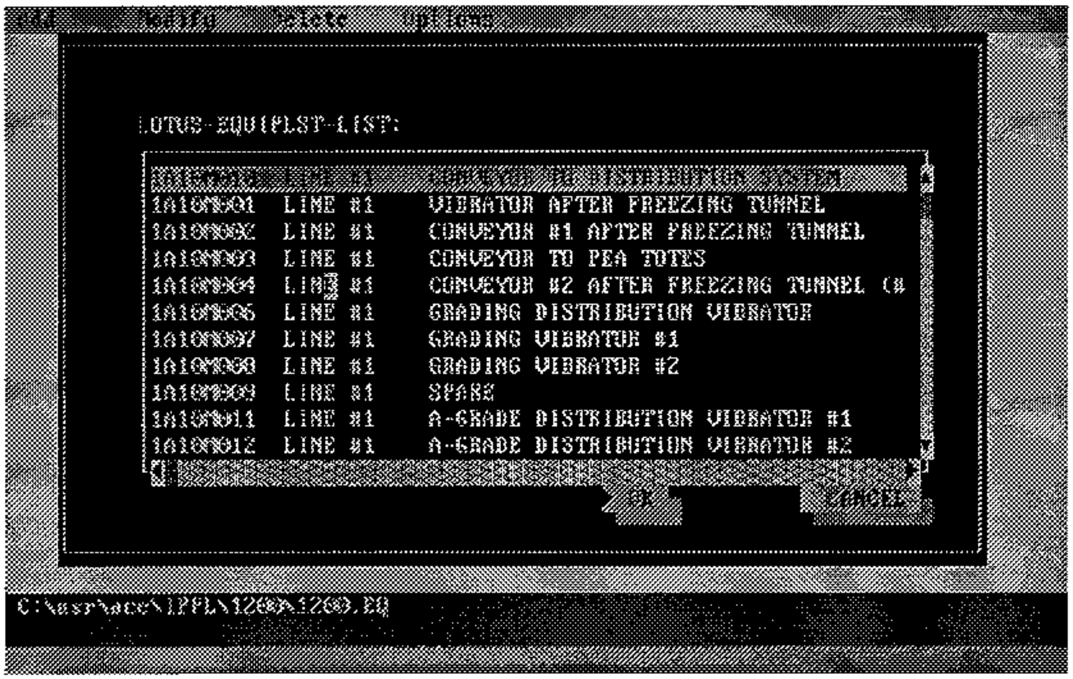


Figure 5.3 Editing equipment list.

5.4.2.2 **Client Preferences Menu Item**

The **Client Preferences** sub menu displays a new menubar containing the following selections:

Client — **Client** displays a list of current clients including the <u>New</u> option for adding new clients. If <u>New</u> is chosen, a dialog box prompts the user for the new client's name and a default set of preferences is saved as **<client-name>.dat** under a directory of the same name.

File	**File** contains the sub menus **Preferences** and **All Preferences** which enable the user to load a client's preferences or the global preferences.
Save	This sub menu permits the user to save changes.
Delete	**Delete** is used to delete a client, their preferences or both.
Print	**Print** produces a hardcopy of the current file.
Edit	This item displays a new menubar permitting the user to edit just one client's **Preferences** or **All Preferences**.
View	**View** permits the user to **browse** preferences.
Options	This permits the user to **Exit** the system, return to the **Previous Menu** or display **Information** about ACE.

The **Edit** menu bar contains the following items:

Add	**Add** contains the sub menus **Categories**, **Objects** or **Preferences**. By selecting the appropriate item the use can add categories, objects and preferences to a client's preferences database or the global preferences database (see Figure 5.4).
Change	This functions similar to **Add** except that it replaces a selected category, object or preference of a client's preferences with a corresponding entity from the global preferences database. Changes can also be made to the global preferences database and are not restricted by the system.
Delete	**Delete** removes entities from the database.
Options	This permits the user to **Exit** the system, return to the **Previous Menu** or display **Information** about ACE.

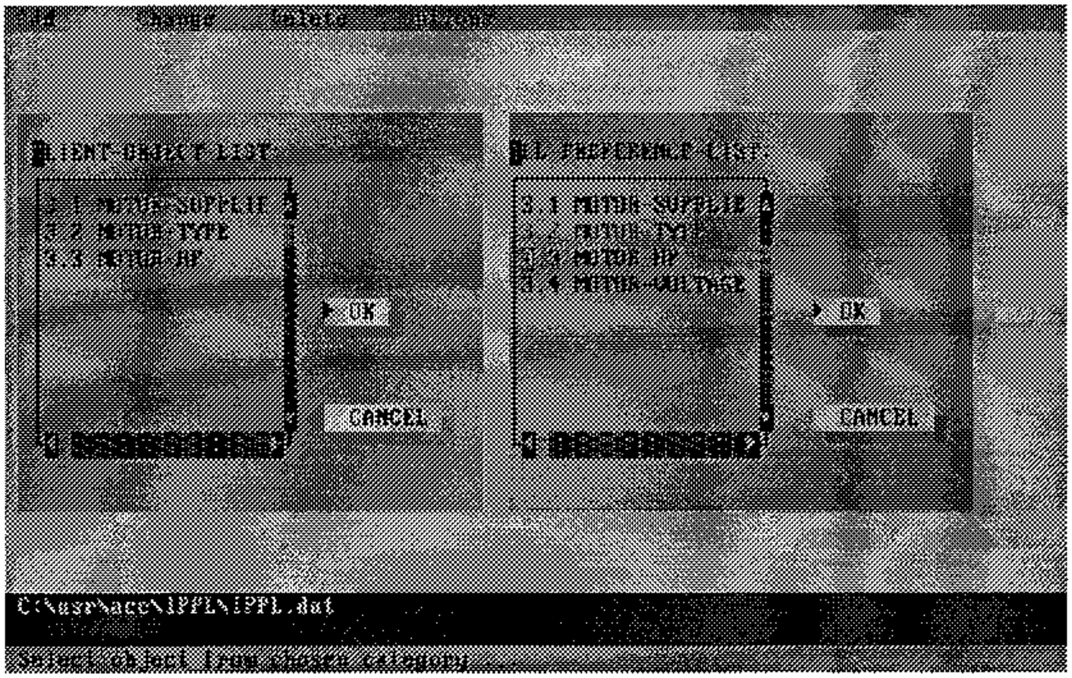


Figure 5.4 Adding an object.

5.4.2.3 **Rules Menu Item**

This section briefly details the design of the **Rules** menu item as it is not implemented in this version of ACE. It provides the user with the ability to add, modify and delete various rules in the four knowledge bases in Figure 5.1. A new menubar is displayed permitting the user to choose the desired knowledge base and the specific rule to be edited. The format of the rules follows the grammar defined in chapter 3. The user is allowed to add, modify and delete the objects in Appendix II, while the values of these objects can be user-defined. This is necessary to preserve the integrity of the knowledge bases and ensure consistency throughout ACE. The rules are then converted to the format used by ART-IM. The user does not have to exit ACE to invoke these rules.

5.4.2.4 **Options Menu Item**

The **Options** sub menu displays miscellaneous items.

Exit	User can return to DOS by selecting **Exit**.
Information	A dialog listing information about ACE is displayed.

5.4.3 Improved ACE User Interface Design

After completing the prototype outlined in section 5.4.2, the need for improvements became apparent. The new design is based on Figure 4.1 and incorporates a more efficient screen layout. Only three separate menubars are required as opposed to the seven outlined in the current version of ACE. The **Level** of an estimate is now referred to as its **Class**. **Level** is used to describe the different levels of detail found in a cost estimate (section 4.2).

The login procedure remains the same. The main menu contains the following items:

Estimate	**Estimate** is used to determine which part of the project is being included in the resulting estimate.
Building(s)	Selecting this sub menu produces a new menubar which enables the user to edit the various files associated with a project and generates a cost or design estimate.
Infrastructure	Beyond the scope of this thesis.
Land	Beyond the scope of this thesis.
Preferences	The **Preferences** menu item produces a new menubar.

Figure 5.5 New main menu design.

The **Building(s)** menubar, as previously stated, is used to edit a project's files. There is also a provision for loading just one project's files called **Open Project #**. Instead of selecting a client, then a project number, the user can go directly to a project file using just the filename.

File	This item contains the sub menus necessary for accessing project files.
<u>New Project</u>	A dialog box is produced permitting the user to enter work order sheet information (section 3.2.2).
<u>Open Project</u>	The user selects the client, project number and the desired project file to edit.
<u>Open Project #</u>	Quick access method.
<u>Save</u>	Any modifications are saved to the file currently being edited.
<u>Save As</u>	Any modifications to the file currently being edited are saved under a new filename.
<u>Main Menu</u>	The user is returned to the initial ACE menubar.
<u>Exit</u>	This sub menu exits ACE completely and also prompts the user to save any unsaved modifications.
Edit	The Edit menu item is only enable when an equipment list or estimate is being edited. The sub menus enable the user to **<u>Add</u>**, **<u>Modify</u>** or **<u>Delete</u>** rows in the spreadsheet file. They operate the same as the Add, Modify and Delete in the existing ACE prototype.
Generate Estimate	The user can generate either a cost or design estimate.
<u>Cost Estimate</u>	A dialog box is produced enabling the user to choose the class and the type of estimate (i.e., civil, electrical or mechanical). A **Class 1** estimate is generated using a percentage of a previous similar project's cost. the user chooses a particular client's project number and the percentage to be used to generate the estimate. The resulting estimate shows only the total estimate for each building in the project. The user can also use the Class 1 option to test "what if" situations for existing projects. The **Class 2** and **3** estimates use the equipment and device lists and are much more detailed.

Figure 5.6 New **<u>Building</u>** menubar design.

<u>Design Estimate</u>	The Design Estimate is only generated for **Class 2** and above estimates. It uses the equipment list to determine how many drawings were created and charges a flat rate per drawing. The rate is displayed and the user is given the opportunity to override this amount

Figure 5.6 New **<u>Building</u>** menubar design (continued).

The menubar produced by the **Preferences** menu item is used to edit a client's preferences, the global and default preference values.

File	File contains all the sub menus for accessing a client's preference files.
<u>New Client</u>	New Client prompts the user for the client's name. As with the current ACE version, a copy of the default preferences file is assigned to the new client.
<u>Open Preferences</u>	This sub menu permits the user to edit a particular client's preferences.
<u>Global Preferences</u>	This sub menu permits the user to edit the global preferences.
<u>Default Preferences</u>	This sub menu permits the user to edit the default preferences.
<u>Save</u>	Any modifications are saved to the file currently being edited.
<u>Save As</u>	Any modifications to the file currently being edited are saved under a new filename.
<u>Main Menu</u>	The user is returned to the initial ACE menubar.
<u>Exit</u>	This sub menu exits ACE completely and also prompts the user to save any unsaved modifications.

Figure 5.7 New **Preferences** menubar design.

Edit Edit is only enabled when a preference file is opened. The sub menus permit the user to **Add**, **Modify** and **Delete** categories, sub-categories and preferences. These sub menus perform the same functionality as they did in the current version of ACE.

Figure 5.7 New **Preferences** menubar design (continued).

5.4.4 Interfacing With LOTUS 1-2-3

Currently the equipment list and device list are stored as Lotus 1-2-3 worksheet files. To prevent duplication of the data it is advantageous to be able to access these files from ACE.

Two C toolkits that serve as interfaces between the C language and Lotus worksheet files are **Goldstein's C Programmer's Library** and **WKS Library**. The following sections outline the investigation and possible incorportation of each with ACE.

5.4.4.1 Goldstein's C Programmer's Library

Goldstein's toolkit provides the user with various libraries containing functions to save and retrieve worksheet information and also defines the data structures containing the contents of the file. The user's reference manual provides a hardcopy of the library source code. Additional source code for reading, writing and printing worksheet files is also included as user-acccessible files.

The Goldstein's toolkit [Goldstein, 1989] sets up a 2D array containing pointers to the worksheet data. This enables one to readily modify a cell's contents and attributes. That is, the actual values found in the file and any information about the file itself are stored in global memory locations. This location is assigned a value which is then stored in the array in the row and column corresponding to the worksheet row and column. As well, cells are added to the worksheet by allocating space for the cell contents and updating the array to point to the newly allocated structure.

5.4.4.2 **WKS Library**

WKS [Raima Corporation, 1990] differs from Goldstein's in that only the source code is provided and each C function must be compiled into an object file and then combined into library files. This is not done for the user as the files must be compiled differently according to the C compiler being used. WKS stores a worksheet in a format similar to Goldstein's. However, rather than use a pointer array, each worksheet is read into a globally defined record and the row can then be modified by the user. The user must know the layout of the record in order to access the cell value and associated attributes.

5.4.4.3 **Accessing Lotus Files Using Goldstein's C Functions**

This section outlines the various test harnesses set up to test the feasibility of using Goldstein's functions as an interface between ACE and Lotus 1-2-3. The test data is a sample of the first eight records in an actual equipment list file with all cells of type string.

State1

Goldstein's C functions use a 2D array to hold pointers to the memory locations allocated for the contents of the worksheet cells and the cell attributes. It uses two functions to store and retrieve data:

1. **StoreCell** stores the Lotus worksheet information in a global memory location and sets up the array of pointers.
2. **GetCell** is used to access a particular row and column in the array and retrieves the memory location of the cell contents.

An example of a Goldstein routine taken from the Goldstein reference, called Getfile was modified to skip over all record types except strings. At this point in time the code was running outside of ART-IM.

State 2

A new version of ART-IM Studio is generated incorporating the Goldstein's functions and some user-defined functions to open, load, read and close a worksheet file.

The user-encoded C functions were defined in ART-IM. The actual values from the pointer locations were passed back to ACE (running in the ART-IM Studio). These values were then placed in schemas which could be displayed by the user in a window in ACE.

State 3

An insert function was defined to insert a new record into the worksheet file. The algorithm is as follows:

1. Start at bottom of worksheet and copy the last **row** to the blank row following it (memory is allocated for the cell values and attributes and the pointer array is updated).
2. Go to **row - 1** and copy that record to **row**; decrement **row**.
3. Continue at step 2 until the row after which the new record is to be inserted is reached. The previous cell values are erased and the new cell values are saved in the memory locations.

State 4

The modifications to GetFile forcing it to skip over non-string cell types were removed. As well, hidden columns are accessed using **GetHIDCOL** (a 32 byte array where each bit represents a possible column in the worksheet; if a bit is set to 1, then the column it represents is hidden). An attempt was made to create a new version of ART-IM Studio incorporating these. Both Goldstein's and ART-IM have **Get_DOUBLE** routines and the compilation of the C code failed due to a redefinition error.

The solution was to not include the **wkstools.h** include file in the user's C code, but create a new include file called **acetools.h** which does not contain the library **IWKSGETF.CB** (this contains the source code for some of the Goldstein functions). A scanner was used to scan the contents of the IWKSGETF.CB source code from the Goldstein's reference (excluding the Get_DOUBLE routine) and place it in the user-defined C code file. Various errors with respect to missing functions could not be resolved.

5.4.4.4 **Alternative to Lotus Format**

The one unfortunate aspect of using the Goldstein or WKS C code is that a new version of ART-IM must be generated and ACE must be recompiled to incorporate any changes to the code. One solution would be to store the equipment and device lists and the resulting spreadsheet as a comma-separated file or in flat ASCII format. Once the estimate is generated and read into Lotus, it could use Lotus' totaling and propagation capabilities to perform calculations and incorporate any modifications.

5.5 **Alternative Tools**

Release 4.0 of Lotus 1-2-3 for Windows started shipping in 1993. This release provides direct program access to the spreadsheet structures through a dynamic link library. Using this mechanism for creating, modifying and maintaining some of the data files of ACE directly in Lotus 1-2-3 spreadsheet format seems promising. In addition, the Microsoft Visual Basic development environments should be investigated as an alternative for future development of the graphical user interface.

6 SUMMARY AND CONCLUSIONS

6.1 Overview of Thesis

One of the main objectives of this thesis was to determine an effective and general-purpose method of representing cost estimating knowledge. Research revealed the need for a system incorporating both this knowledge and client preference information that could be used for various engineering fields, as well as for the construction industry. Software for producing cost estimates exists, but it lacks reasoning capabilities.

A context-free grammar was designed to represent the resulting cost estimate, as well as the required inputs, the equipment and device lists. This grammar describes the five classes of estimates ranging from a preliminary estimate to a detailed final cost estimate. A method of representing and modifying client preferences for a particular manufacturer or type of equipment was also devised. This lead to the development of a prototype user interface capable of capturing and recording this information.

Knowledge gathering is another important aspect of expert system design. Various interviews with cost estimating experts revealed the need for a method of capturing cost estimating information and representing it for use by the system. The grammar detailed in Appendix II for client preference rules can be applied to the general information rulebase. The type of information is different, but the format remains the same.

A prototype expert system was designed and partially implemented to gather data about clients (both new and existing), capture client preference information and permit changes to equipment lists. The integration of other software tools with ART-IM for the purpose of modifying equipment list spreadsheet files was investigated. Although development environments capable of accessing spreadsheet files now exist, the exercise was useful in that it did not require the users to store their existing information in a new format. Various demonstrations revealed the importance of designing an appropriate user interface. Simplicity and ease of use are essential ingredients to a good design.

6.2 Lessons Learned

The development of a prototype expert system to generate cost estimates is a useful concept. The approach taken for this thesis, although feasible, can be improved. ART-IM Version 2.1 (the expert system shell used) functions well as a rule interpreter. Unfortunately, because it is DOS-based, the user interface functions do not produce the type of GUI (Graphical User Interface) that can be developed in a Windows environment. The prototype tends to be slow with respect to data access. It takes approximately 20 seconds to load the equipment list spreadsheet file containing only 27 records. The client preference file which is similar in size to the global preference file in Appendix I requires approximately 5 seconds to load. As well, ART-IM does not provide the developer with clean file access methods. The functions return only a boolean value. There are no error numbers to indicate, for example, that a file does not exist or that the hard drive or floppy diskette is full.

A screen produced using ART-IM requires more coding time than one produced using a GUI development tool. For example, the position and size of the dialog boxes shown in Figure 5.4 are entered as part of the code. In order to view the results the code must be run and the desired values determined by trial and error. Object oriented programming languages allow the programmer to actually size the dialog box and see the results without running the code. It can take approximately three times more coding time to generate a screen using ART-IM as opposed to using a GUI development tool.

Another area of improvement would be the knowledge collection process. For example, the information gathered for the general information database came from only one domain expert. It would be better to devise a grammar similar to the preference grammar in Appendix II for representing this knowledge and require a minimum number of participants. This would ensure consistent feedback and provide a broader information base and level of expert knowledge.

The domain expert responsible for providing the cost estimate knowledge also assisted in the design of the prototype. One major complaint was the degree of difficulty experienced navigating the system. The individual felt that the prototype was not very intuitive. The various levels of menus appeared confusing to the user. It was also felt that the system required excessive mouse movements when selecting and modifying data. The domain expert was very positive with respect to the system's ability to capture a client's preferences. As well, the incorporation of existing data files such as the equipment list was felt to be very useful because it meant that data did not have to be re-entered in a different format.

6.3 Future Development

One of the main areas requiring future development would be the implementation of the cost estimating rules for generating a cost estimate. A rule editor for the client preference, general information and unit cost rulebases would be a valid addition as well. This would allow the system to grow as the user's cost estimate knowledge increases. The rules would be converted to a format capable of being used by ACE. Currently only the equipment list is modified by ACE. The integration of the device list would be a definite asset for producing more detailed estimates.

The user interface itself is DOS-based and conversion of the system to a Windows environment would provide the user with a more consistent and user-friendly product. Capturing error messages is easier in this type of environment and the ability to access information in various formats is greatly simplified as well. As well, the incorporation of spreadsheet information would be greatly simplified using Windows development tools. Many of these packages permit other Windows application to be to launched from within themselves (Object Linking and Embedding). This would enable the user to use any spreadsheet package to edit the equipment and device lists as long as it is stored in a format readable by the cost estimating system.

An evaluation of possible incorporation of an electronic cost database would be useful for ensuring that current unit costs are always available to the cost estimator. One final area of future work would involve extending the grammar, rule definitions and the prototype to include lighting, power and other fields of engineering such as civil and mechanical. The design of ACE was done in such a manner so as to facilitate this. It would enable the system to be used by individuals involved in various facets of cost estimating regardless of their level of experience.

REFERENCES

Biegel, P.E., John E., Mark D. Bearden, P.E., David Dickerson and William M. O'Donnell, "Building an expert system for cost estimating", 1986 International Industrial Engineering Conference Proceedings, Dallas, Texas, USA, Institute of Industrial Engineers, May 11-14, 1986, pp. 504-10.

Blecker, H.G. and D. Smithson, "Detailed Cost EStimating During the Process Engineering Phase of a Project", Process Economics International, Vol. V, No. 2, 1985, pp. 17-28.

Buchanan, Bruce G. and Edward H. Shortliffe, Rule-Based Expert Systems, Addison-Wesley Publishing Company, 1984, pp. 188-191.

Clark, J.A., J.H. Rutherford and D.M. Randel, "An intelligent front-end for computer-aided building design", ESC 89 Proceedings of the 3rd European Simulation Congress, Edinburgh, UK, September 1989.

"Estimating", American Machinist and Automated Manufacturing, June 1987, pp. 61-2.

Firebaugh, Morris W., Artificial Intelligence, Boston, Massachusetts, PWS-Kent Publishing Company, 1989, pp. 307-310.

Gesner, Shauna M., Bradford G. Nickerson and Laurie Tompkins, "ACE - A knowledge-based assistant for cost estimating", 4th UNB Artificial Intelligence Symposium, Fredericton, NB, Canada, University of New Brunswick, September 20-21, 1991, pp. 133-141.

Gesner, Shauna M., Draft Software Requirement Specifications for ACE - A Knowledge-Based Assistant for Cost Estimating, Fredericton, NB, Canada, January 3, 1992, 10 pages.

Goldstein, Larry Joel, Goldstein's C Programmer's Library, New York, NY, Brady Software, 1989.

Gretton, Robert, ed., Hancomb's 1988 Yardsticks for Costing, Don Mills, ON, Southan Business Publications, 1988.

Johnson, D.G., "Knowledge Representation and Manupulation forEngineering Design", GEC Review, Vol. 6, No. 1, 1990, pp. 47-59.

Lee, Sungyoul and Dr. K. A. Ebeling, "Cost Control", Computers & Industrial Engineering, Vol. 13, Nos 1-4, 1987, pp.356-8.

Mahoney, William D., Building Construction Cost Data 1988, Kingston, MA, R.S. Means Company, Inc., 1987.

Massimino, Sal, pub., "Computerized Estimating Benefits Bidding", World Construction, July 1985, pp. 26-8.

M^c^Graw, Karen and Karan Harbison-Briggs, Knowledge Acquisition, Englewood Cliffs, New Jersey, Prentice Hall, Inc., 1989, p. 126.

Nightingale, Ed, "Computerized Cost Estimating Removes Wild Guesses", Modern Machine Shop, February 1987, pp. 62-71.

Ntuen, Celestine A. and Arup K. Mallik, P.E., "Applying Artificial Intelligence to Project Cost Estimating", Cost Engineering, Vol. 29, No. 5, May 1987, pp. 8-12.

Ntuen, Celestine A. and Arup K. Mallik, P.E., "Cost Estimation as a Complexity Modeling Problem", Proceedings of the 31st Annual Meeting of the American Association of Cost Engineers, Atlanta, Georgia, USA, June 28-July 1, 1987, pp. G.6.1-G.6.7.

Object Management Group, Inc., The Common Object Request Broker: Architecture and Specification, Draft 10 December 1991, pp. 45-76.

Raima Corporation, The WKS Library, Bellevue, WA, Raima Corporation, 1990.

Stretton, Alan and Geoff Stevens, "PREDICTE: An Expert System for Estimating Indicative Construction Times for Multi-Storey Buildings at Concept Stages", CIB 14-21 March 1990 Building Economics and Construction Management. Sydney: International Council for Building Research Studies and Documentation, 1990.

Timberline Software, Precision Estimating, Beaverton, OR, Timberline Software Corp.,1990.

Wildeman, H., "Cost Estimating Approaches of Screening and Budget Estimates for Large Scale Technical Projects", Production Managements Systems, Falster, P., and Ralstada, A., Eds. North Holland Pub. Co., 1981, pp.179-210.

APPENDIX I

GLOBAL PREFERENCE FILE

O 1.0 unit-substation

C 1.1 unit-substation-feeder

C 1.2 unit-substation-conduit

C 1.3 unit-substation-cable

C 1.4 unit-substation-terminations

C 1.5 unit-substation-spare-breakers

C 1.6 unit-substation-start-up-service

C 1.7 unit-substation-freight

O 2.0 MCC

C 2.1 MCC-supplier

P Allen-Bradley

P Square D

P Siemens

P Telemechanique

PKluckner Mueller

C 2.2 MCC-feeder

C 2.3 MCC-process-panels

C 2.4 MCC-cable-tray

C 2.5 MCC-equipment-feeds

C 2.6 MCC-pushbutton-controls

C 2.7 MCC-capacitors

C 2.8 MCC-starters

C 2.9 MCC-fuses

C 2.10 MCC-heaters

C 2.11 MCC-drives

C 2.12 MCC-amps

C 2.13 MCC-voltage

P 360

P 380

P 480

P 600

O 3.0 motors

C 3.1 motor-supplier

P Toshiba

P Eurodrive
P Brooks
P Lincoln
P Westinghouse
P General Electric
C 3.2 motor-type
P D.P.
P TEFL
P TEXP
C 3.3 motor-HP
P 1
P 5
P 10
O 4.0 motor-feeders
C 4.1 motor-feeder-supplier
C 4.2 motor-feeder-type
C 4.3 motor-feeder-connectors
C 4.4 motor-cable-tags
C 4.5 motor-connection
C 4.6 starter-connection
O 5.0 motor-controls
C 5.1 metal-enclosed-switch-supplier
P Federal Pioneer
P Westinghouse
C 5.2 metal-enclosed-switch-voltage
P 360
P 380
P 480
P 600
C 5.3 metal-clad-switch-supplier
P S & C
P Federal Pioneer
P Allen-Bradley
C 5.4 pushbutton-supplier

P Allen-Bradley

P Telemechanique

C 5.5 pushbutton-type

P EEMAC4X

C 5.6 pushbutton-location

P field

P panel room

C 5.7 pushbutton-start

P red

P green

C 5.8 pushbutton-stop

P red

P green

C 5.9 safety-switch-supplier

P Allen-Bradley

P Federal Pioneer |

P Siemens

C 5.10 safety-switch-type

P heavy duty

P standard duty

C 5.11 fuse-type

P HRC class J

P code fuses

O 6.0 process-panels

C 6.1 process-panel-supplier

P Westinghouse

P Federal Pioneer

C 6.2 process-panel-type

P EEMAC4X

C 6.3 process-panel-feeder

C 6.4 process-panel-breakers

C 6.5 process-panel-voltage

P 120/240

P 120/208

P 347/600 Y

P 600 Delta

C 6.6 process-panel-number-wires

P 4

C 6.7 process-panel-location

P 3

O 7.0 PLC

C 7.1 PLC-supplier

P Allen-Bradley

P Siemens

P Modicon

P Square D

C 7.2 PLC-inputs

C 7.3 PLC-outputs

C 7.4 PLC-cable

C 7.5 PLC-cable-connections

O 8.0 junction-boxes

C 8.1 junction-box-supplier

P Rob-Roy

P Stalin

P Hammond

C 8.2 junction-box-type

P EEMAC4X

C 8.3 junction-box-material

P fiberglass

P steel

O 9.0 lighting

C 9.1 lighting-supplier

P Holophane

P Keene

P Westinghouse

P Hubble

C 9.2 lighting-type Mercury-vapour

P HPS

P HID
C 9.3 lighting-wattage
P 70
P 100
P 150
P 250
P 400
C 9.4 lighting-circuit-voltage
P 120
P 208
P 240
P 347
P 600
O 10.0 cables
C 10.1 power-cable-type
P TECK
P conduit
C 10.2 power-cable-voltage
P 1000
P 5000
P 15000
C 10.3 TC-instrument-cable-supplier
P Dekoran
P other
C 10.4 LV-instrument-cable-supplier
P Dekoran
P other
C 10.5 RFD-instrument-cable-supplier
P Dekoran
P other
C 10.6 cable-supplier
P T & B
C 10.7 cable-material
P spin-on

P aluminum

C 10.8 cable-cover

P corrosion boot

C 10.9 TECK-cable-tag-number-size

P 3/4"

C 10.10 TECK-cable-tag-material

P stainless steel

C 10.11 TECK-cable-connector-type

P watertight

C 10.12 TECK-cable-connector-material

P spin-on aluminum

P stainless steel

C 10.13 TECK-cable-connector-supplier

P T & B

P Crouse Hinds

C 10.14 TECK-cable-connector-cover

P corrosion boot

P no boot

O 11.0 cable-trays

C 11.1 cable-tray-supplier

P Electrovert

P Pilgrim

P Rob-Roy

C 11.2 cable-tray-type

P heavy duty

C 11.3 cable-tray-material

P aluminum

P fiber glass

P stainless steel

C 11.4 cable-tray-standard

P CSA-classI

C 11.5 cable-tray-support-hardware

P 316 stainless steel

P mild stainless steel

C 11.6 cable-tie-material

P nylon

C 11.7 cable-tray-class

P C

P D

C 11.8 cable-tray-max-tray-support-spacing

P 7

P 14

C 11.9 cable-tray-loading

P 100

P loading

O 12.0 globals

C 12.1 PF-correction-capacitor-supplier

P ASEA

C 12.2 CCTV-systems-supplier

P Commercial Communications

P N.B. Tel

C 12.3 fire-alarm-system-supplier

P Edwards

C 12.4 mimimum-cct-panels

P 42

C 12.5 work-force unionized

C 12.6 charge-out-rate-markup

P 1.9

C 12.7 derating-factor

P. 7

C 12.8 pushbutton-sealed?

P Y

P N

C 12.9 electrical-nameplate?

P Y

P N

C 12.10 instrument-nameplate?

P Y

```
P N
C 12.11 cable-tray-100%?
P Y
P N
C 12.12 units-of-measure
P metric
P imperial
```

APPENDIX II

GRAMMAR FOR CLIENT PREFERENCE RULES.

```
(1) <rule>          ::= "(" <name> <if-clause> <then-clause> ")"
(2) <if-clause>     ::= "(" "IF" <if-rest> ")"
(2) <if-rest>       ::= "(" "AND" <lhs-term>+ ")"
                      | <lhs-term>
(3) <lhs-term>      ::= "(" <if-op> <lhs-object> <value> ")"
(4) <if-op>         ::= "<"
                      | ">"
                      | "<="
                      | ">="
(5) <lhs-object>    ::= client
                      | area
                      | project-location
                      | structure-age
(6) <then-clause>   ::= "(" "THEN"<then-rest> ")"
(7) <then-rest>     ::= "(" "AND" <rhs-term>+ ")"
                      | <rhs-term>
(8) <rhs-term>      ::= "(" <then-op> <rhs-object> <value> ")"
(9) <then-op>       ::= "="
(10) <rhs-object>   ::= unit-substation-feeder
                      | unit-substation-conduit
                      | unit-substation-cable
                      | unit-substation-terminations
                      | unit-substation-spare-breakers
                      | unit-substation-start-up-service
                      | unit-substation-freight | MCC-supplier
                      | MCC-feeder | MCC-process-panels| MCC-cable-tray
                      | MCC-equipment-feeds | MCC-pushbutton-controls
                      | MCC-capacitors | MCC-starters | MCC-fuses
                      | MCC-heaters | MCC-drives | MCC-amps
                      | MCC-voltage | motor-supplier | motor-type
                      | motor-HP | motor-feeder-supplier
                      | motor-feeder-type | motor-feeder-connectors
                      | motor-cable-tags | motor-connection
                      | starter-connection | metal-enclosed-switch-supplier
```

| metal-enclosed-switch-voltage
| metal-clad-switch-supplier | pushbutton-supplier
| pushbutton-type | pushbutton-location
| pushbutton-start | pushbutton-stop
| safety-switch-supplier | safety-switch-type
| fuse-type | process-panel-supplier
| process-panel-type | process-panel-feeder
| process-panel-breakers | process-panel-voltage
| process-panel-num-wires | process-panel-location
| PLC-supplier | PLC-inputs | PLC-outputs
| PLC-cable | PLC-cable-connections
| junction-box-supplier | junction-box-type
| junction-box-material | lighting-supplier
| lighting-type | lighting-wattage
| lighting-circuit-voltage | power-cable-type
| power-cable-voltage | T/C-instrument-cable-supplier
| LV-instrument-cable-supplier
| RFD-instrument-cable-supplier
| cable-supplier | cable-material | cable-cover
| TECK-cable-tag-number-size
| TECK-cable-tag-material
| TECK-cable-connector-type
| TECK-cable-connector-material
| TECK-cable-connector-supplier
| TECK-cable-connector-cover
| cable-tray-supplier | cable-tray-type
| cable-tray-material | cable-tray-standard
| cable-tray-support | cable-tie-material
| cable-tray-class | cable-tray-loading
| process-panel-max-size
| PF-correction-capacitor-supplier
| CCTV-systems-supplier | fire-alarm-system-supplier
| mimimum-cct-panels | work-force unionized
| charge-out-rate-markup | derating-factor

```
                   | pushbutton-sealed? | electrical-nameplate?
                   | instrument-nameplate? | cable-tray-100%?
                   | units-of-measure | manual-starter-location
                   | manual-starter-type | manual-starter-supplier
                   | motor-voltage | pushbutton-atmosphere
                   | metal-enclosed-switch-min-voltage
                   | metal-enclosed-switch-max-voltage
                   | PLC-min-size | PLC-max-size | Xmers-type
                   | Xmers-min%-taps | Xmers-max%-taps
                   | Xmer-supplier | lighting-VAC | lighting-max-VAC
                   | lighting-min-VAC | cable-type | cable-numbering
                   | nameplate-size..| nameplate-type
                   | nameplate-letter-height
                   | nameplate-background-color | wire-marking-type
                   | wire-marking-size | wire-marking-supplier
                   | wire-marking-letter-color
                   | wire-marking-background-color
                   | cable-tray-num-bolt-nut-washers | strut-material
                   | clamp-type | cable-tray-bolt-nut-washer-type
                   | strut-attached-cable?
                   | cable-tray-bolt-nut-wahser-material | clamp-material
                   | cable-trough | cable-numbering
                   | cable-tag-number-size | cable-tag-material
                   | cable-tray-support-spacing
                   | process-panel-phase | process-panel-min-size
(11) <value>     ::= <digit>+
                   | <string>
(12) <string>    ::= """" <character>+ """"
(13) <character> ::= <letter>
                   | <digit>
                   | " "
```

APPENDIX III

CLIENT PREFERENCE RULES

```
(client-rule-1:
        (IF       (= client "IPPL"))
        (THEN  (AND (= MCC-voltage 600)
                    (= MCC-supplier "Allen-Bradley"))))

(client-rule-2:
        (IF       (= client "IPPL"))
        (THEN  (AND (= metal-enclosed-switch-voltage 600)
                    (= metal-enclosed-switch-supplier "Federal Pioneer"))))

(client-rule-3a:
        (IF       (= client "IPPL"))
        (THEN  (AND (>= metal-clad-switch-voltage 2400)
                    (<= metal-clad-switch-voltage 13800)
                    (= metal-clad-switch-supplier "S & C"))))
(client-rule-3b:
        (IF       (= client "IPPL"))
        (THEN  (AND (>= metal-clad-switch-voltage 2400)
                    (<= metal-clad-switch-voltage 13800)
                    (= metal-clad-switch-supplier "Federal Pioneer"))))
(client-rule-3c:
        (IF       (= client "IPPL"))
        (THEN  (AND (>= metal-clad-switch-voltage 2400)
                    (<= metal-clad-switch-voltage 13800)
                    (= metal-clad-switch-supplier "Allen-Bradley"))))

(client-rule-4a:
        (IF       (= client "IPPL"))
        (THEN  (AND (= power-cable-type "TECK")
                    (= power-cable-voltage 1000))))
(client-rule-4b:
        (IF       (= client "IPPL"))
        (THEN  (AND (= power-cable-type "TECK")
                    (= power-cable-voltage 5000))))
```

```
(client-rule-4c:
        (IF     (= client "IPPL"))
        (THEN   (AND (= power-cable-type "TECK")
                (= power-cable-voltage 15000))))

(client-rule-5a:
        (IF     (= client "IPPL"))
        (THEN   (AND (= instrument-cable "low-voltage")
                (= instrument-cable-supplier "Dekoran"))))
(client-rule-5b:
        (IF     (= client "IPPL"))
        (THEN   (AND (= instrument-cable "T/C")
                (= instrument-cable-supplier "Dekoran"))))
(client-rule-5c:
        (IF     (= client "IPPL"))
        (THEN   (AND (= instrument-cable "RTD")
                (= instrument-cable-supplier "Dekoran"))))

(client-rule-6:
        (IF     (= client "IPPL"))
        (THEN   (= PF-correction-capacitor-supplier "ASEA")))

(client-rule-7a:
        (IF     (= client "IPPL"))
        (THEN   (= CCTV-systems-supplier "NB Tel")))
(client-rule-7b:
        (IF     (= client "IPPL"))
        (THEN   (= CCTV-systems-supplier"Commercial
        Communications")))

(client-rule-8:
        (IF     (= client "IPPL"))
        (THEN   (= fire-alarm-system-supplier "Edwards")))
```

```
(client-rule-9:
        (IF     (= client "IPPL"))
        (THEN   (AND (= cable-tray-material "fiberglass")
                (= cable-tray-type "heavy duty")
                (= cable-tray-supplier "Electrovert" | "Rob-Roy")
                (= cable-tray-support-hardware "316 stainless steel"))))

(client-rule-10:
        (IF     (= client "IPPL"))
        (THEN   (AND (= junction-box-material "fiberglass")
                (= junction-box-type "EE MAC4X-TR")
                (= junction-box-supplier "Rob-Roy"))))

(client-rule-11:
        (IF     (= cable-tray-material "aluminum"))
        (THEN   (AND (= cable-tray-type "heavy duty")
                (= cable-tray-standard "CSA Class I")
                (= cable-tray-supplier "Pilgrim"))))

(client-rule-12:
        (IF     (= client "IPPL"))
        (THEN   (AND (= cable-material "spin-on aluminum")
                (= cable-type "watertight")
                (= cable-supplier "T & B"))))

(client-rule-13:
        (IF     (AND (= client "IPPL")
                (= area "corrosive")))
        (THEN   (= cable-cover "corrosive boot")))
```

```
(client-rule-14:
    (IF     (AND (= client "IPPL")
            (= pushbutton-location "field")))
    (THEN   (AND (= pushbutton-sealed? "Y")
            (= pushbutton-type "Cema 4X")
            (= pushbutton-stop "green")
            (= pushbutton-start "red")
            (= pushbutton-supplier "Allen-Bradley"))))

(client-rule-15:
    (IF     (AND (= client "IPPL")
            (= pushbutton-location "panel room")))
    (THEN   (AND (= pushbutton-atmosphere "clean")
            (= pushbutton-supplier "Telemechanique"))))

(client-rule-16:
    (IF     (= client "IPPL"))
    (THEN   (AND (= manual-starter-location "field")
            (= manual-starter-type "Cema 4X")
            (= manual-start-supplier "Siemans"))))

(client-rule-17a:
    (IF     (= client "IPPL"))
    (THEN   (AND (= panel-board-type "Cema 4X")
            (= panel-board-location "field")
            (= panel-board-phase 3)
            (= panel-board-num-wires 4)
            (= panel-board-min-size 120)
            (= panel-board-max-size 208)
            (= panel-board-supplier "Westinghouse"))))
```

```
(client-rule-17b:
    (IF     (= client "IPPL"))
    (THEN   (AND (= panel-board-type "Cema 4X")
                 (= panel-board-location "field")
                 (= panel-board-phase 3)
                 (= panel-board-num-wires 4)
                 (= panel-board-min-size 120)
                 (= panel-board-max-size 208)
                 (= panel-board-supplier "Federal Pioneer"))))

(client-rule-18:
    (IF     (= client "IPPL"))
    (THEN   (AND (= Xmers-type "Dry EPt")
                 (= Xmers-min%-taps 2)
                 (= Xmers-max%-taps 2.5)
                 (= Xmers-supplier "Westinghouse"))))

(client-rule-19:
    (IF     (AND (= client "IPPL")
                 (= mill "old")
                 (= area "outside")))
    (THEN   (AND (= lighting-type "HPS")
                 (= lighting-VAC 120)
                 (= lighting-supplier "Holophane"))))

(client-rule-20:
    (IF     (AND (= client "IPPL")
                 (= mill "old")))
    (THEN   (AND (= lighting-type "mercury-vapour")
                 (= lighting-min-VAC 120)
                 (= lighting-max-VAC 220)
                 (= lighting-supplier "Holophane"))))
```

```
(client-rule-21a:
    (IF     (= client "IPPL"))
    (THEN   (AND (= fuse-type "form 1 class J HRC")
                 (= safety-switch-type "heavy duty")
                 (= safety-supplier "Federal Pioneer"))))
(client-rule-21b:
    (IF     (= client "IPPL"))
    (THEN   (AND (= fuse-type "form 1 class J HRC")
                 (= safety-switch-type "heavy duty")
                 (= safety-supplier "Allen-Bradley"))))
(client-rule-22:
    (IF     (= client "IPPL"))
    (THEN   (AND (= control-wiring "general PLC")
                 (= terminal-supplier "Wiedmuller"))))

(client-rule-23:
    (IF     (= client "IPPL"))
    (THEN   (AND (= PLC-min-size 2)
                 (= PLC-max-size 30)
                 (= PLC-supplier "Allen-Bradley"))))

(client-rule-24:
    (IF     (= client "IPPL"))
    (THEN   (AND (= nameplate-size "2 x 4")
                 (= nameplate-type "Lamacoid")
                 (= nameplate-letter-height "3/16"))))

(client-rule-25:
    (IF     (AND (= client "IPPL")
                 (= electrical-nameplate? "Y")))
    (THEN   (AND (= nameplate-letter-color "black")
                 (= nameplate-background-color "white"))))
```

```
(client-rule-26:
        (IF       (= client "IPPL"))
        (THEN  (AND (= wire-marking-type "heatshrink")
                     (= wire-marking-size "3/16")
                     (= wire-marking-letter-color "black")
                     (= wire-marking-background-color "white")
                     (= wire-marking-supplier "Raychem"))))

(client-rule-27:
        (IF       (= client "IPPL"))
        (THEN  (AND (= cable-tag-number-size "3/4")
                     (= cable-tag-material "stainless steel"))))

(client-rule-28:
        (IF       (AND (= client "IPPL")
                     (= cable-tray-100%? "Y"))
                     (= cable-tray-%-filled 100)))
        (THEN  (= derating-factor 0.7)))

(client-rule-29:
        (IF       (= tray-class "D"))
        (THEN  (= tray-spacing 14)))

(client-rule-30a:
        (IF       (AND (= client "IPPL")
                     (= lights "HID")))
        (THEN  (= light-wattage 150)))
(client-rule-30b:
        (IF       (AND (= client "IPPL")
                     (= lights "HID")))
        (THEN  (= light-wattage 250)))
```

```
(client-rule-30c:
        (IF       (AND (= client "IPPL")
                  (= lights "HID")))
        (THEN  (= light-wattage400)))

(client-rule-31:
        (IF       (AND (= client "IPPL")
                  (= strut-material "stainless steel")
                  (=  cables-attached-to-struts? "Y")))
        (THEN  (AND (= clamp-type "universal")
                  (= clamp-material "stainless steel"))))

(client-rule-32:
        (IF       (AND (= client "IPPL")
                  (= cable-trough "fiberglass-polyester")))
        (THEN  (AND (= tray-support-material "stainless steel")
                  (= tray-bolt-min-torque 45)
                  (= tray-bolt-max-torque 60)
                  (= tie-material "nylon")
                  (= num-bolt-nut-washers 2)
                  (= bolt-nut-washer-type "flat")
                  (= bolt-nut-washer-material "stainless steel"))))

(client-rule-33:
        (IF       (= tray-class "C"))
        (THEN  (= tray-spacing 7)))
```

```
(client-rule-34a:
        (IF      (AND (= client "IPPL")
                 (= cable-trough "fiberglass-polyester")))
        (THEN    (AND (= format "IPPL")
                 (= equipment-numbering "IPPL")
                 (= cable-numbering "IPPL")
                 (= motor "Toshiba")
                 (= motor-HP 1)
                 (= light-circuit-voltage 120)
                 (= cable-tray-100%? "Y")
                 (= minimum-cct-panels 42)
                 (= work-force "unionized")
                 (= charge-out-rate-markup 1.9)
                 (= contract "5/30-IPPL"))))
(client-rule-34b:
        (IF      (AND (= client "IPPL")
                 (= cable-trough "fiberglass-polyester")))
        (THEN    (AND (= format "IPPL")
                 (= equipment-numbering "IPPL")
                 (= cable-numbering "IPPL")
                 (= motor "Toshiba")
                 (= motor-HP 5)
                 (= light-circuit-voltage 120)
                 (= cable-tray-100%? "Y")
                 (= minimum-cct-panels 42)
                 (= work-force "unionized")
                 (= charge-out-rate-markup 1.9)
                 (= contract "5/30-IPPL"))))
```

```
(client-rule-34c:
        (IF       (AND (= client "IPPL")
                  (= cable-trough "fiberglass-polyester")))
        (THEN  (AND (= format "IPPL")
                  (= equipment-numbering "IPPL")
                  (= cable-numbering "IPPL")
                  (= motor "Toshiba")
                  (= motor-HP 10)
                  (= light-circuit-voltage 120)
                  (= cable-tray-100%? "Y")
                  (= minimum-cct-panels 42)
                  (= work-force "unionized")
                  (= charge-out-rate-markup 1.9)
                  (= contract "5/30-IPPL"))))

(client-rule-35:
        (IF       (AND (= client "IPPL")
                  (>= motor-HP 200)
                  (<= motor-HP 800)))
        (THEN  (AND (= motor-voltage 2300)
                  (= motor-type "D.P."))))

(client-rule-36:
        (IF       (AND (= client "IPPL")
                  (>= motor-HP 0.5)
                  (<= motor-HP 200)))
        (THEN  (AND (= motor-voltage 575)
                  (= motor-type "TEFL"))))
```

APPENDIX IV

GENERAL INFORMATION RULES

IF

the short-circuit capacity is greater than 22000 amps

THEN

use a fuse

IF

the short-circuit capacity is less than or equal to 22000 amps

THEN

use a circuit breaker

IF

the motor horsepower is greater than 40

THEN

use an ammeter

IF

new plant

THEN

unit substation required

IF

retrofit

THEN

unit substation required

IF

60 foot candles

THEN

place them 22 ft apart

IF

100 foot candles

THEN

place them 16 ft apart

IF

bulk plant

THEN

incorporate 10% spare starters of each size

AND

incorporate 25% spare space for starters

IF

bus-load

THEN

80% of maximum load

VITA

Candidate's full name: Shauna Mae Gesner

Place and date of birth: Woodstock, NB, Canada
September 29, 1967

Permanent Address: 817 Main Street
Woodstock, NB
E0J 2B0

Schools Attended: Woodstock High School
Woodstock, NB, Canada
1982-1985

Universities attended: University of New Brunswick
Bachelor of Science Computer Science
Fredericton, NB, Canada
1985-1990

Publications:

Gesner, Shauna M., Bradford G. Nickerson and Laurie Tompkins, "ACE - A knowledge-based assistant for cost estimating", 4th UNB Artificial Intelligence Symposium, Fredericton, NB, Canada, University of New Brunswick, September 20-21, 1991, pp. 133-141.