ENHANCEMENT OF A DECISION SUPPORT SYSTEM FOR ECONOMIC JUSTIFICATION OF COMPLEX SYSTEMS

By

John J. Frank, PE
B.S.I.E., University of Louisville, 1999

A thesis
Submitted to the Faculty of the
University of Louisville
Speed Scientific School
As Partial Fulfillment of the Requirements
For the Professional Degree

MASTER OF ENGINEERING

Department of Industrial Engineering
University of Louisville
Louisville, Kentucky

December 2003

ENHANCEMENT OF A DECISION SUPPORT SYSTEM FOR ECONOMIC JUSTIFICATION OF COMPLEX SYSTEMS

Submitted by: John James Frank

A thesis approved on

12/15/03

(Date)

by the Following Reading and Examination Committee:

Dr. John S. Usher, Thesis Director

Dr. William E. Biles

Dr. Rammohan K. Ragade

ACKNOWLEDGEMENTS

I wish to thank my loving wife Karen for her extraordinary patience and assistance while completing this thesis and two master's degrees at the University of Louisville. Her efforts in maintaining our household can never fully be appreciated. For my children, Courteney and Nicholas, I thank them for their understanding while dad was completing his schoolwork.

I want to thank Dr. Usher for his patience and understanding while I completed my work. The numerous deadlines and false starts must have been extremely frustrating. No other single person at the University of Louisville has had as great an impact on the future course of my education than he by allowing my work to continue. Dr. Usher's patience, caring and understanding of personal issues bring great credit upon himself, the department of Industrial Engineering, the Speed Scientific School and the University of Louisville. Furthermore, I wish to thank Drs. William Biles and Rammohan Ragade for there assistance.

ABSTRACT

Economic justification for complex systems, such as material handling equipment or computer network infrastructure is difficult to perform when many intangibles exist. Traditional methods of economic justification typically exclude intangibles. JustMAT is a decision support system that incorporates both the tangible and intangible in an analysis. JustMAT uses the Dunn-Rankin methodology to transform the Net Present Value and the intangible attributes into normalized indexes plotted on a two-axis graph. This allows the analyst to determine if the intangibles are significant or not to the analysis.

Presented in this thesis is the enhancement of a decision support system called JustMAT. The DSS software enhancement enables the net present worth analysis to be performed using either before or after tax cash flows. The original JustMAT used only before tax cash flows and therefore ignored the consequences of taxes in the analysis. The addition of the after tax cash flows enabled a more precise analysis of the investment decision, which is particularly advantageous for investments that produce only cost savings.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Page

I. INTRODUCTION

Economic justification is an important part of many business decisions and follows a complex process. Decision Support Systems (DSS) are computer programs developed to improve the decision making process by combining information, with proven analytical techniques to produce data to assist decision makers in making accurate decisions. DSS programs can vary greatly in their form and purpose. In this section an overview of why economic justification is important to business will be reviewed. The problem statement will define the approach and the proposed solution followed by the final section detailing the scope of this analysis and the limits in which the analysis was performed.

A. Overview

Businesses in general are out to make a profit for their owners—maximizing shareholder return. Because some decisions are based upon relatively incomplete information, decision makers need proven analytical techniques that utilize the best information available to assist them in making decision. Economic Justification is the classification of many different methodologies which analyze investment decision for cost effectiveness, meaning the cost of the project is less than the expected return. Since many of the techniques and analyses used are lengthy and cumbersome DSS were created to simplify the application of Economic Justification. These systems semi-automate the economic justification analysis and enable the user to quickly analyze different scenario. Economic Justification and DSS are important to decision makers because they enable them to use more complex analysis techniques to assist them with investment decisions.

Simple economic justification usually considers only the easily quantifiable costs and benefits. Businesses, though, are inherently complex and many of the decisions made by them are very complex using imperfect information. This results in the necessity to use DSS that can handle many of the complex decision. The goal of economic justification is to increase the accuracy of decisions made in an investment.

For example, consider a proposed conveyor system that has two control options. The first method is a simple timer. When the timer trips on the controller, an event occurs on the line, such as the processing of the product at a station or releasing it for movement to the next station. This type of control is very cheap, but requires intense calculations to ensure that the line is properly balanced. The second controller requires an Ethernet network and a central computer to be installed controlling the line. Basic economic justification most likely would determine that the timed controller is the best system considering only the financial aspects of the investment. This investment decision ignores the future direction of the company, the philosophy of the company, the effects the investment may have on other investments, the amount of control line operators may have, and other situationally dependent factors. Decision makers need a DSS to empower them with the ability to improve their decision making abilities.

B. Problem Statement

As technology advances, the ability to analyze an increasing amount of data has resulted in the need to improve the decision making capabilities of the decision makers by enabling them to apply more complex economic justification methodologies. Computer systems is an area, much like material handling equipment, that many of the investment decisions are very expensive and decisions affect the future cost of other projects. Sullivan and Liggett (1988) tackled on this complex decision problem with

JustLAN. Consider the decision of whether a computer network with a centralized printer should be used or if printers should be bought for each computer system. Initially with two computers, it is probably quite obvious that buying two printers make sense and not to invest in a computer network and a network printer. The question then is at what point does the decision makers stop buying printers and install a network with a central printer. Moreover, a network has greater capabilities than sharing of printers. Computer networks enable users to accomplish tasks not possible without a network, such as sharing an Internet connection, transferring large files between system computers without the use of a disk, or the sharing of software, thereby reducing the number of license copies required. Sullivan and Liggett did just that and designed a DSS system to assist decision makers in including hard to quantify costs and intangibles in their investment decisions.

The Material Handling Institute of America (MHIA), in Charlotte, North Carolina, sponsored Dr. John S. Usher at the University of Louisville in cooperation with Ahmed Kamal to convert the original JustLAN software to Visual Basic and apply it to material handling problems. The software created was JustMAT 1.0. This software was a barebones conversion, with many problems; it was not intuitive; it had data entry problems, it lacked the ability to easily perform an after-tax evaluation of the investment decision; and, it needed cleanup of general appearance.

C. Scope of the thesis

This thesis focuses on the enhancement of JustMAT 1.0 and documentation of the software. Chapter II reviews the JustMAT 1.0 software explaining the different techniques used and provides an overview of problems with the software, requirements, programming methodologies. Chapter III details JustMAT 1.2 and the modifications

made to JustMAT 1.0 to create this version. Chapter IV presents a case study utilizing JustMAT 1.2 to solve a problem that requires incorporation of the tax effects into the decision. The final chapter summarizes the findings of this thesis with an overview and provides recommendations and areas of further study.

II. JustMAT 1.0

The following sections present an overview of JustMAT 1.0, a literary review of other economic justification methodologies, a critique, problems with JustMAT 1.0, as well as the screen shots of JustMAT. In the Appendix a portion of the JustMAT 1.2 programming code created as part of this thesis. The unmodified portions and slightly modified portions of the program for JustMAT 1.0 are not included as part of this thesis.

A. Overview

JustMAT 1.0 is a DSS designed to enhance the ability of decision makers to make choices when intangibles may have a significant effect on the decision. The software is based upon a group decision-making process, where a chairperson creates the project and the categories to be ranked and scored. JustMAT is then sent to each of the committee members for their input and prioritization of the attributes. Once the committee has completed their evaluation and classification, the results complied by the committee chairman determine the effects that soft benefits may have on an investment decision; determining whether to accept or reject the lowest cost alternative.

JustMAT is modeled after the work of William Sullivan and Hampton Liggett (1988). The application for the original DSS was a computer LAN network investment decision. The Material Handling Institute of America, in Charlotte, NC, commissioned the adaptation of the DSS for use with material handling equipment. JustMAT was created at the University of Louisville by Ahmed H. Kamal under the direction of Dr. John Usher.

The conversion was relatively straightforward considering the similarities between BasicA and Visual Basic. Kamal wrote the original JustMAT 1.0 in Visual Basic essentially duplicating Sullivan and Liggett's JustLAN.

JustLAN considered the use of a multi-attribute evaluation model for selecting between two competing networking solutions. Sullivan and Liggett's methodology uses a group decision making process where costs and intangibles are integrated into the decision making process, a ratio scaled importance weight for intangibles is used, and a graphical presentation of the tangible and intangible scoring is presented.

B. Existing Methodologies for Dealing with Intangibles

Various methodologies exist for dealing with investment decisions that involve intangibles. The following literature review examines existing methodologies for investment decisions.

Justification of large expenditures needs to take into consideration two parts according to Krinskey and Miltenburg (1991). The first part consists of the strategic analysis where a company sees how the investment fits with their long term objectives. The second and most obvious part is the economic alternative decision where standard financial measures are applied to the alternatives to determine the best outcome.

Sambasivarao and Deshmukh (1997) show that the high cost of material handling equipment and complexity of the decision can be simplified through the use of a DSS. They developed one that combines an economic, multi-attribute, and risk evaluation approaches for selection and justification of automation technology. The models used in the DSS are Pay/Back, Return on Investment, Net Present Value, Internal Rate of Return, Linear Additive Model, and Analytic Hierarchy Process.

The DSS is divided up into four components, which consists of Database, Model Base, Dialogue Management, and Solution Base. The DSS is demonstrated through a case study with the case situation taken from Datta et al (1997).

Chiadamrong and O'Brien (1999) propose a decision support system for choosing the best alternative manufacturing or production system. Their approach uses economic, analytical and strategic justification making use of each method to reduce the drawbacks of any one particular methodology.

Another method by Demmel and Askin (1992) creates a model that incorporates three objectives that are broken into the following attributes: the pecuniary, tactical, and strategic. The pecuniary is based upon the traditional discount cash flow technique, with the results normalized to a [-1, +1] scale. The tactical and strategic objectives are based upon the concept of qualitative flows, and qualitative discounting to achieve a present value. The three objectives trade off to produce a ranking.

A simpler system, developed by Lavell, Sorenson and Aye (1998), uses Microsoft Excel to incorporate costs and benefits as well as tax effects.

Computer technology is advancing quickly. As businesses expand their computer systems, normal decision making techniques they may not consider include intangible concerns such as expandability, control or networking. Simple payback and other financial analysis tools can be short sighted resulting in a higher upgrade costs. Sullivan and Liggett (1992) address these issues by developing a decision support tool that utilizes a multi-attribute evaluation model. The computerized decision support tool incorporates five features. The first feature is the incorporation of a group of decision makers who rate different aspects of the project according to categories created by the project manager. The decision support software then processes the data to develop a ratio-scaled importance weight of attributes. The monetary cost of each alternative is incorporated through discounted cash flows and considers the life cycle cost of each

alternative. To better understand the decision process, the discounted cash flows and ratio-scaled importance attributes are plotted on a graph for a visual assessment of the best alternative. Finally, the decision support software can perform a sensitivity analysis to determine the robustness of the recommended alternative to estimation errors.

C. Critique

The techniques and DSS reviewed in the literature seek through different methods to maximize the benefits and minimize the deficits when analyzing an alternative. The first methods, which do not consider the intangibles, are the typical methods used to evaluate an investment decision. For more complex analysis, a DSS can be used. Each methodology examined provides a slightly different result and each has slightly different limitations. JustMAT 1.0 enhances the Sullivan and Liggett's computerized decision support tool by reprogramming it in Visual BASIC and storing it in a simple package on a floppy drive. The problem with JustMAT 1.0 as with earlier versions of the methodologies is that the tax effects are ignored in the analysis.

D. Problems with JustMAT 1.0

The review of JustMAT 1.0 focused on the usability and methods for increasing accuracy of the analysis. Initial review of the software reveled serious usability issues, such as problems with data entry.

Data entry methods are inconsistent between modules, in some cases data entry cells would have to be deleted before a new number was accepted. In other cases pressing the enter key would not enter the data. Furthermore, entering data in a cell and clicking to a new cell would not always accept the data and in some cases revert back to the original entry. Problems with the data entry resulted from the belief that some

information had been entered but actually was never recorded. This added to confusion in the analysis.

JustMAT was designed around a group of windows that open and close based upon the actions of the user. As with the data entry issues, windows actions were not always intuitive. Some windows when opened for the first time would appear in front and then would not close even though another window had been opened in front of it. In some cases, the window would close and in others it would not. Also selecting a window sometime had undesirable consequences when choosing different functions such as deleting previous information.

D. Coding and Screen Shots

JustMAT consists of seventeen forms, three modules, one class module, and a MDImenu. JustMAT was derived from the BasicA programming language created by Sullivan and Liggett. This section will identify major forms in JustMAT 1.0

JustMAT 1.0 is a Visual Basic Program design around the concepts of forms and menus. These forms are accessed through buttons or through the menus on the tool bar. The menu structure can be seen in Figure 1. This section will detail JustMAT by describing each of the major forms.

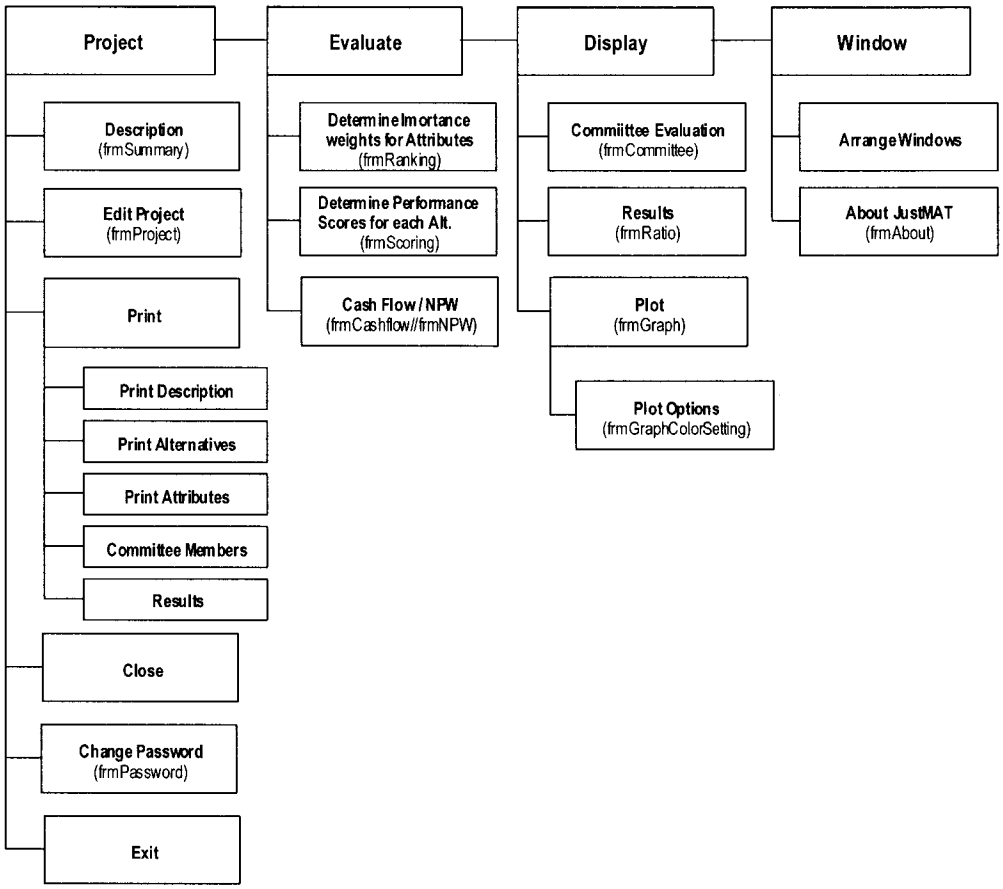

Figure 1 – Menu Structure

FrmComittee

The form Committee (frmComittee) provides summary information on each member's rankings, raw scores assigned, average scores for each alternative, and any biased scores, if any. Committee Rankings form shows the order with which each committee member ranked the alternative (Figure 2). The ranking of each alternative is done individually by each committee member when they logon through the form Rank. There are no provisions for changing the rating by the committee chairman. Furthermore, the only committee member with access to this form is the chairman.

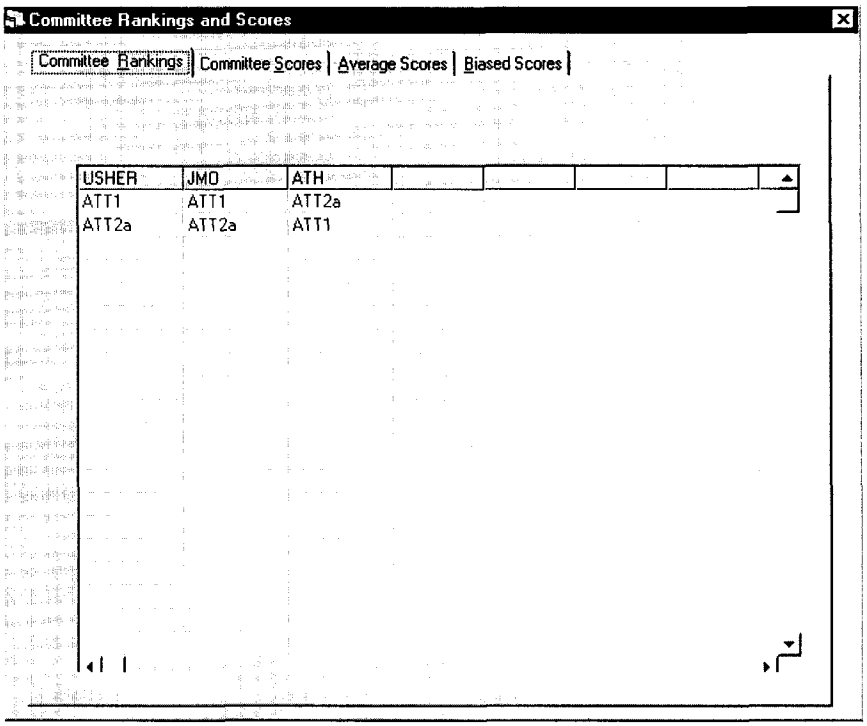

Figure 2 – Committee Form

The committee chairman is able to review the score of each committee member and can be seen in Figure 3. This form allows the chairmen to select each committee member's id and then display their scores.

The committee chairman is able to see the average score of the committee members by opening the Average Score window (See Figure 4). The average is calculated by adding each committee members score and then dividing by the number of

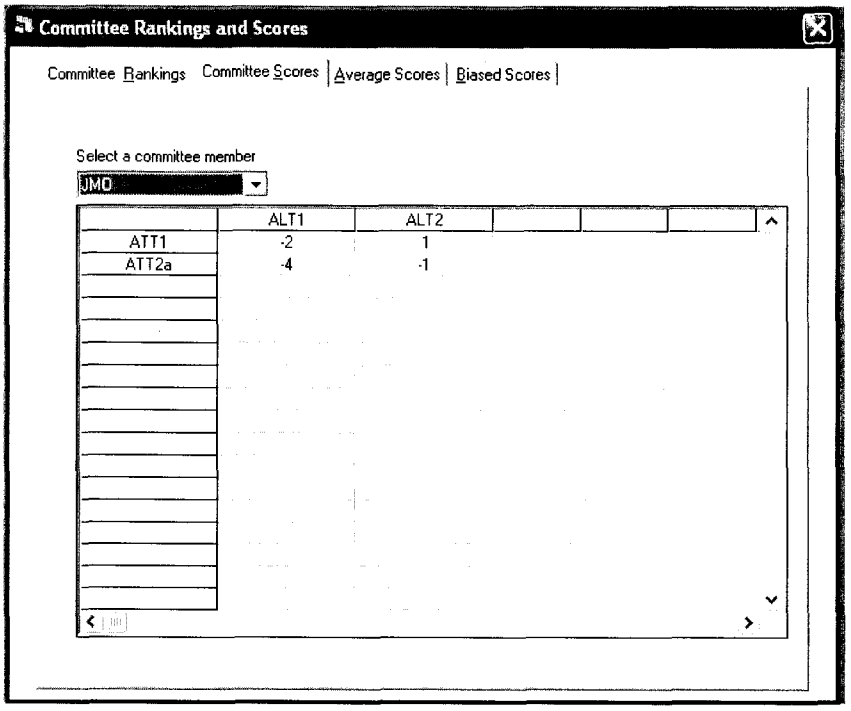

Figure 3 – Individual Committee Members Score

committee members. To assist with the decision process JustMAT looks for committee member's scores that are considerably different than the other committee members and displays the biased score on the committee ranking and scoring biased form.

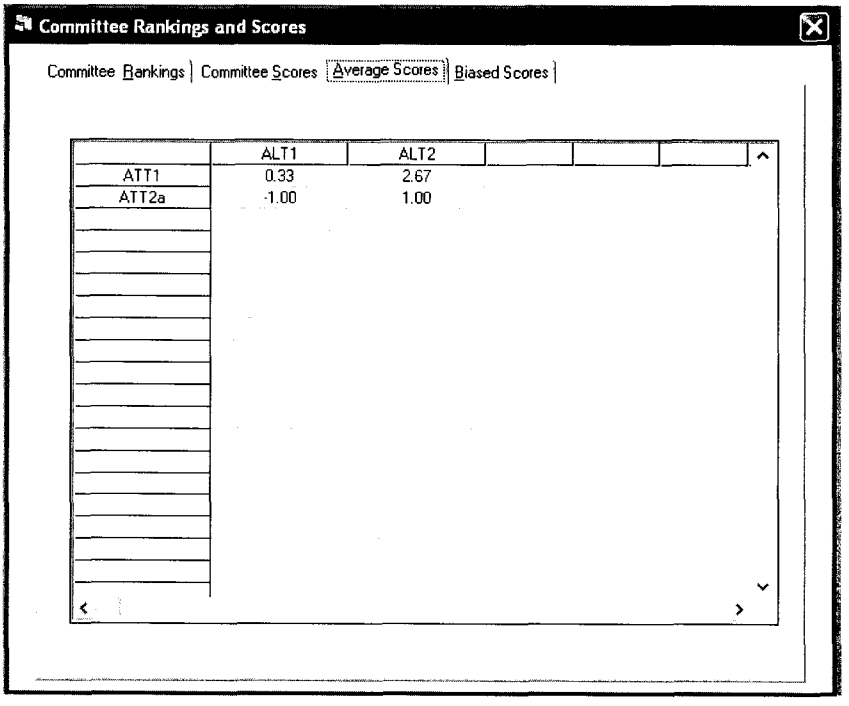

Figure 4 – Committees Average Score

frmcashflow

The form cash flow (frmcashflow) is where the quantifiable costs are entered along with the Minimal Acceptable Rate of Return and the number of periods for the analysis. Cash flows are entered directly into this form. Figure 5 is a screen shot from JustMAT 1.2. The only difference between JustMAT 1.0 and 1.2 is the Cash Flow Calculator button. Cash flows entered into this screen are used to calculate the Net Present Worth of the Alternative. Aggregate cash flows are entered in the period they occur for each alternative.

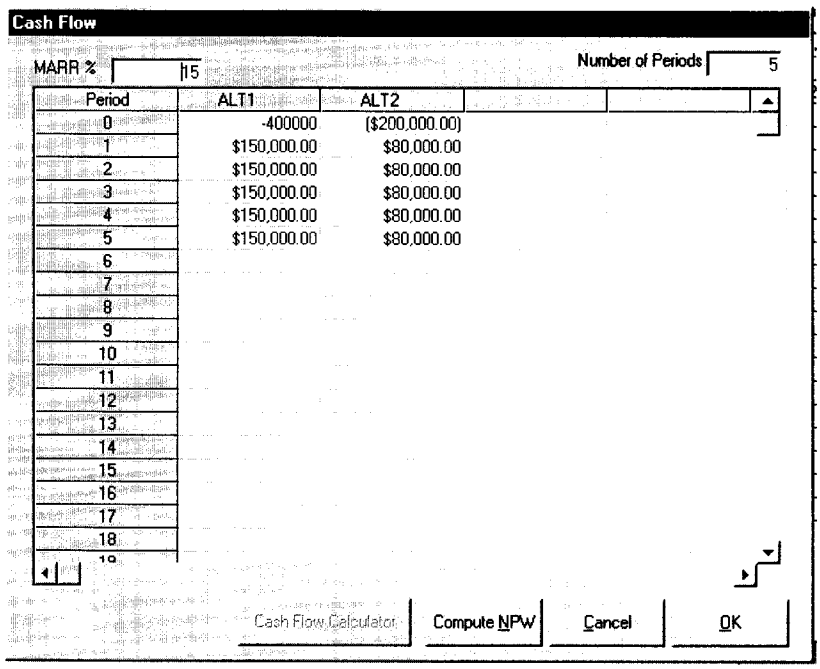

Figure 5 – Cash Flow Form

frmDictionary

The form Dictionary (frmDictionary) provides a library of predefined and user defined attribute information available for use in projects (See Figure 6).

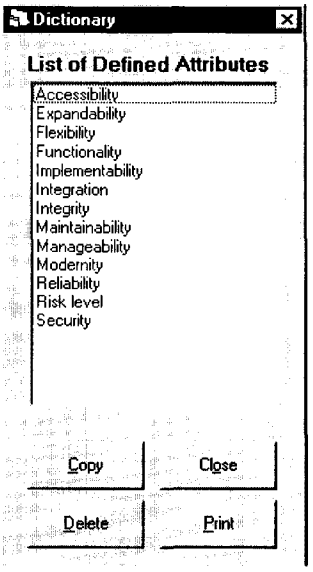

Figure 6 – Dictionary Form

frmgraph

The form graph (frmgraph) presents the overall results of the decision. The graph was updated by Athisan Wayrparb to provide a more accurate display of the results in JustMAT 1.2. From Figure 7 the decision maker can see that ALT 1 has he highest NPW, but the lowest value score. The scaling of the graph is such that the percent difference between value scores is larger than the percentage difference between the NPW values. This difference would have to be viewed as insignificant (the zone where the lines intersect) to select ALT 1 as the best.

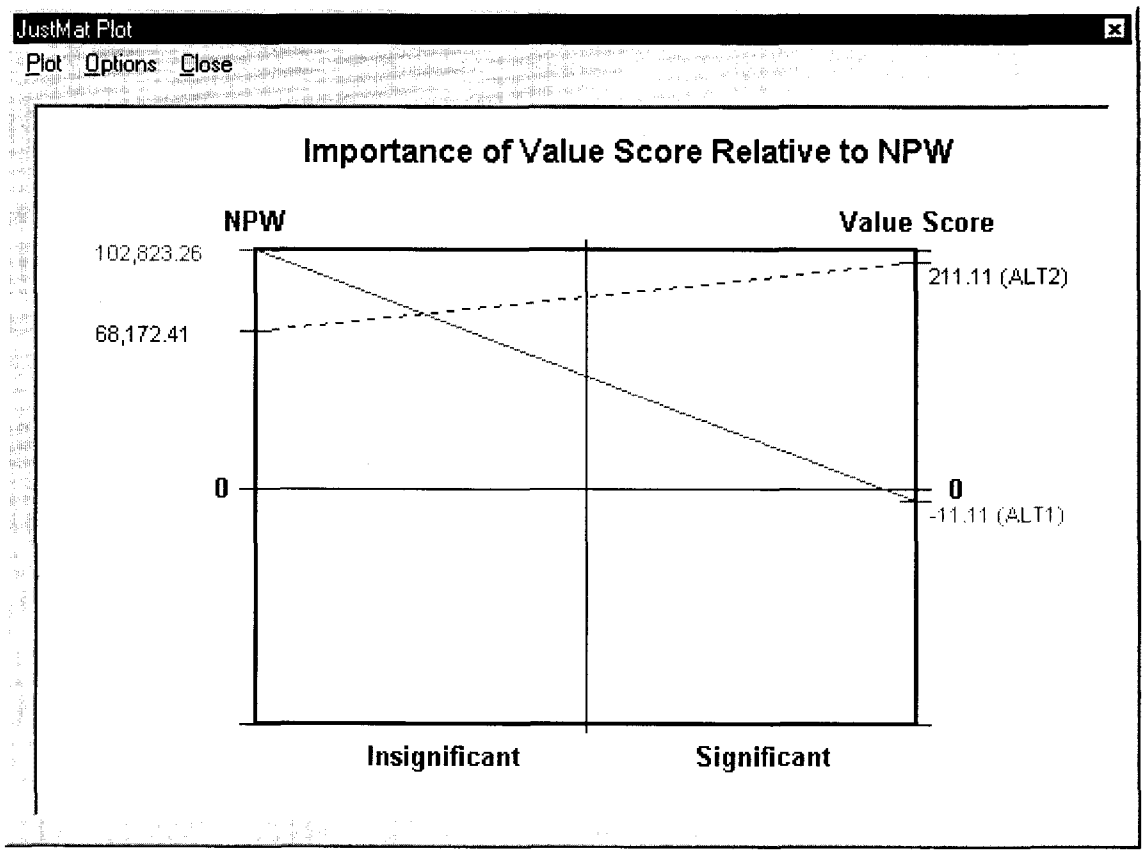

Figure 7 – Graph Form

frmgraphcolorsetting

The form graph color setting (frmgraphcolorsetting) is a new form added as a result of the modifications to frmgraph enhancing the results presentation. The form (Figure 8) allows the user to modify the colors and attributes of the graph.

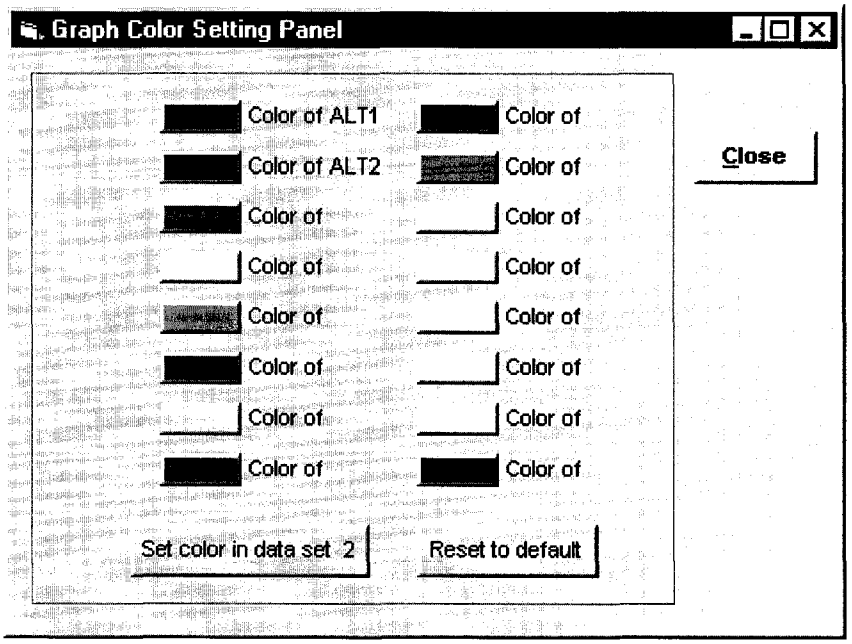

Figure 8 – Graph Color Setting

frmNPW

The form Net Present Worth (NPW)(frmNPW) calculates a net present worth value for each alternative (See Figure 9) from the cash flows on the form frmcashflow (See Figure 5).

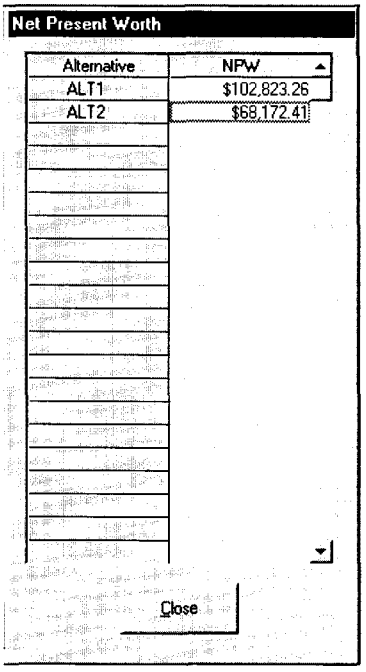

Figure 9 – Net Present Worth Form

For example, a company is deciding between two alternatives. One is an alternative to purchase a new automobile and the other is to continue to repair the vehicle already owned. The NPW for the first alternative calculates to be -$22,245 and the second alternative is -$12,456. Since, the second alternative has a higher NPW; we should accept it and reject the first alternative.

frmProject

This form (frmProject) creates the project and can only be done by the chairman. It defines the alternatives, attributes, and committee members. Figure 10 is where the project description is defined.

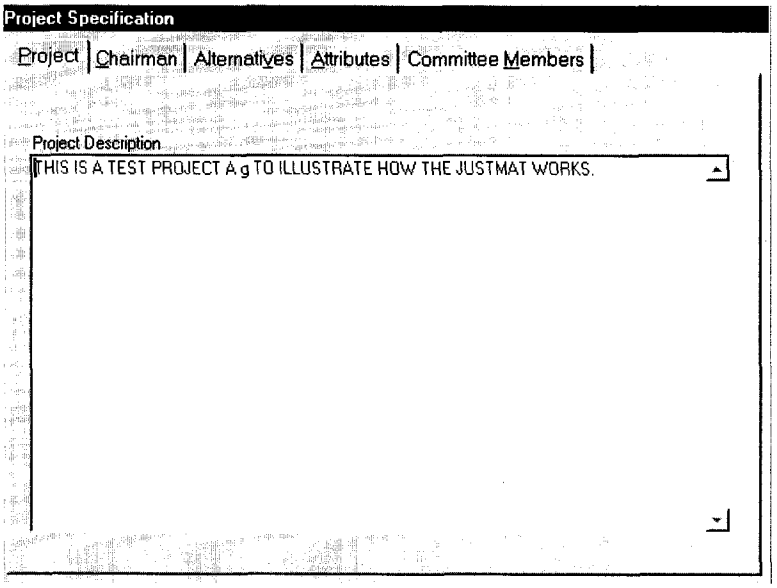

Figure 10 – Project Specification Forms

frmScore

The form Score (frmScore) is where members of the evaluation team score each attribute for each alternative (See Figure 11).

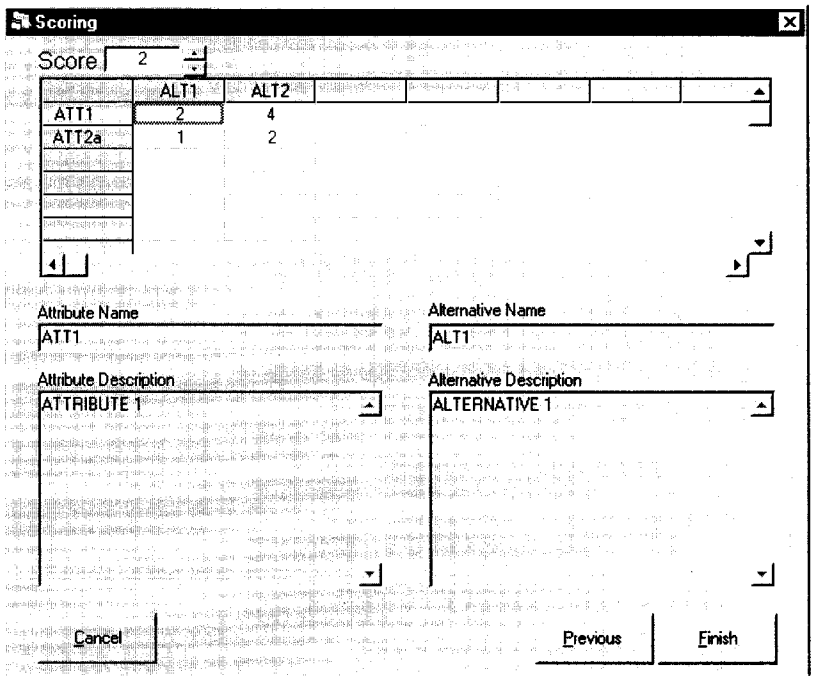

Figure 11 – Scoring Form

frmRank

The form Rank (frmRank) enables each committee member to rank the attributes in order of preference (See Figure 12).

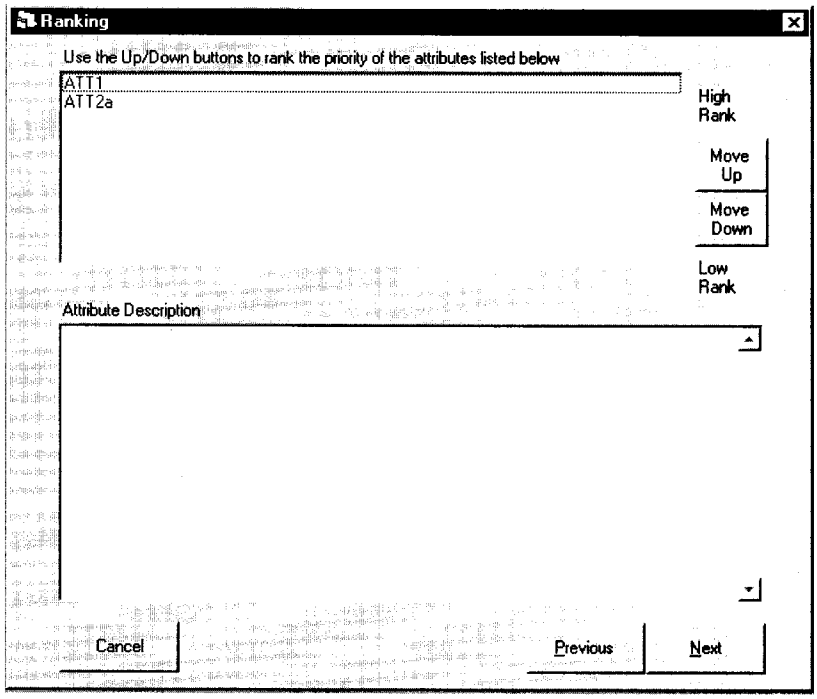

Figure 12 – Ranking Form

frmDunnRankin

The form DunnRankin (frmDunnRankin) calculates the rank, weight and normalized weight of the attributes.

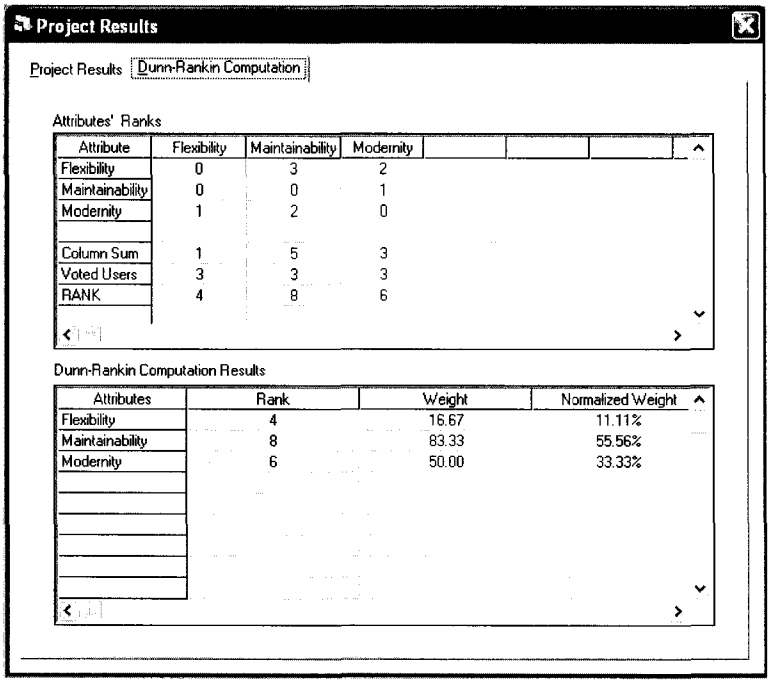

Figure 13 – Dunn-Rankin Computation

The Dunn-Rankin Technique uses an $m \times m$ matrix of paired comparisons, where the m represents the number of attributes. Columnar attributes are compared against horizontal attributes with the criteria that the more important attribute in the pair receives a value of one and the less important is assigned a zero. The following is a brief summary of the Dunn-Rankin procedure (Sullivan and Liggett, 1983).

- The total "votes" in each cell for all committee members are tabulated.
- The attributes are arranged in ascending order along both the horizontal and the vertical axes of the matrix.
- The votes are added up for each column.

- The number of committee members, *n*, is added to each column's total of votes to produce a rank-sum (R_i) for each attribute. Table 1 shows the numerical results of the committee's ranking of the importance of each attribute.
- The following parameters are computed as an intermediate step in obtaining the Dunn-Rankin weights (W_i). Table 2 shows the calculation and results of the Dunn-Rankin technique for the hypothetical project in table 1.

$R_{min} = n$; $R_{max} = mn$; $R_{avg} = \frac{n(m+1)}{2}$

Expected value of standard deviation = $s = \left(\frac{mn(m+1)}{12}\right)^{1/2}$

Normalized ratio of each attribute = $Z_i = \frac{(R_i - R_{avg})}{s}$

Weight of each attribute = $W_i = \frac{(Z_i + |Z_{max}|) \times 100}{(Z_{max} + |Z_{min}|)}$

Normalized weight of each attribute = $Nor(W_i) = \frac{W_i}{\sum_{i=1}^{m} W_i}$

Table 1 - Dunn-Rankin Matrix Developed by the Evaluation Committee

Attribute	*Maintain-ability*	*Flexibility*	*Modernity*
Maintainability	-	2	2
Flexibility	1	-	2
Modernity	1	1	-
Column totals	2	3	4
n	3	3	3
R_i	5	6	7

Table 2 - Calculations and Results of the Dunn-Rankin Technique

	Attribute				
	Min	*Maintain -ability*	*Flexibility*	*Modernity*	Max
$R_i - R_{avg}$	-3	-1	0	1	3
Z_i	-1.732	-.5774	0	.5774	1.732
Z_i+Z_{min}	0	1.1547	1.732	2.3094	3.464
W_i	0	33	50	67	100
$Nor(W_i)$	-	.22	.33	.44	-

- The overall value score of each alternative is computed by combining the normalized importance weights and average scores previously assigned by committee members and the chairman. An additive weighted model is used to obtain the overall value score and depicted as follows:

$$SCORE_j = 100\sum_{i=1}^{m} Nor(W_i)P_{ij}$$

where $Nor(W_i)$ = Normalized weight of attribute i
P_{ij} = Performance score for alternative j against attribute i
m = Number of attributes

The Dunn-Rankin Computation and the final result of evaluation of the alternatives are provided in *Project Results* window. It can be opened by clicking *Results* command under *Display* menu. Figure 13 and 14 show the *Dunn-Rankin Computation* and *Project Result*, respectively, for the hypothetical project in *Project Results* window.

frmRatio

The form Ratio (frmRatio) summarizes the Dunn-Rankin computations and Shows the score of the NPW.

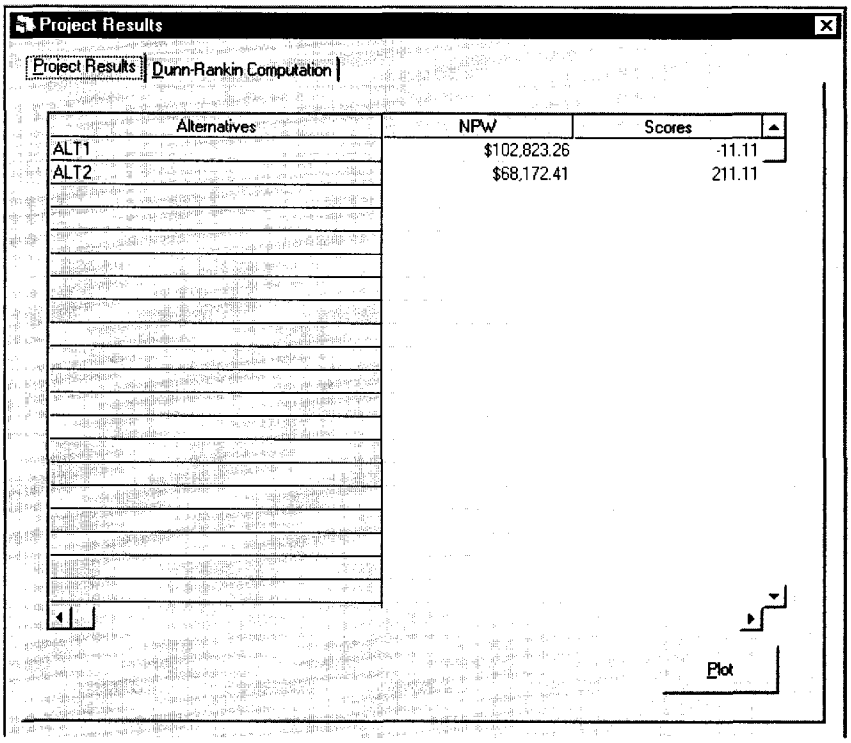

Figure 14 – Project Results Form

III. JustMAT 1.2

The creation of JustMAT 1.2 is in response to the MHIA desire to enhance JustMAT 1.0 by adding the ability to consider the tax effects on an investment. The Cash Flow Calculator is primarily responsible for consideration of the tax effects. Furthermore, the chairman no longer needs to aggregate the cash flows as this is done in the cash flow calculator (CFC) now. Considering the tax effect in the analysis will reduce the net present worth of an investment by adding the depreciation of an asset into the analysis, which is currently ignored and tends to overstate the worth of an investment. This section will discuss the CFC.

A. Cash Flow Calculator

The cash flow calculator (frmAfterTax) is reached by clicking on the alternative in any period. Once an alternative is selected the cash flow calculator button will become active with the alternative listed on the button (See Figure 15). Once the button is active select it and the CFC will activate (See Figure 16).

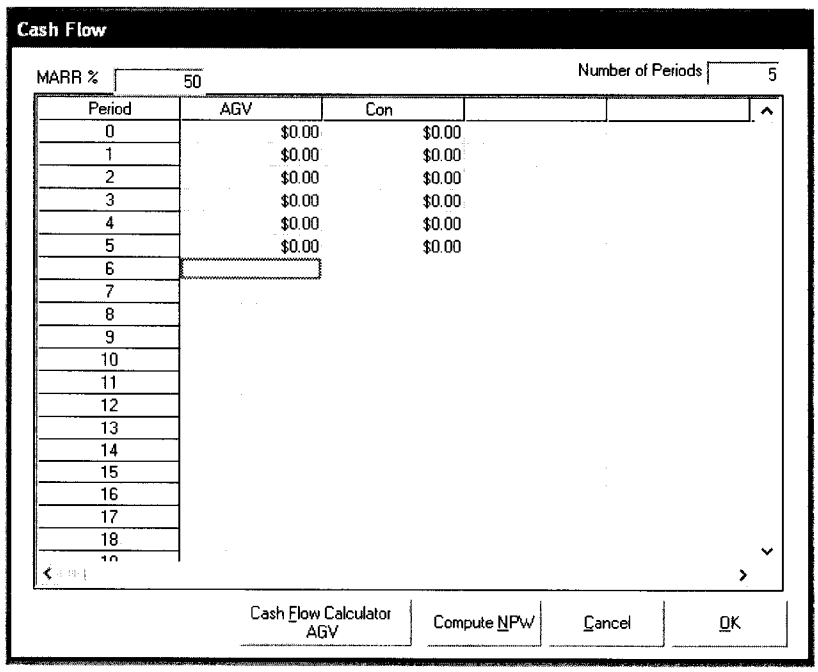

Figure 15 – Cash Flow

The CFC is divided into three quadrants. The top right quadrant is the data entry area that changes depending upon the selected investment, cost, or receipt. The lower right quadrant is the cash flow table showing the Before Tax Cash Flow (BTCF), Depreciation, Taxable Income, Tax, and After Tax Cash Flow (ATCF) by period. The values shown in the cash flow tables are aggregate amounts. The left side of the screen controls the data entry side.

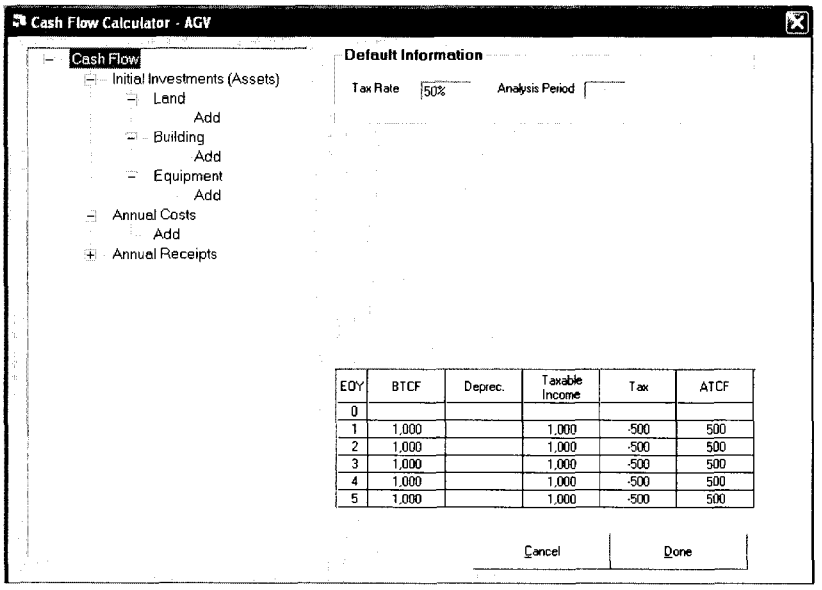

Figure 16 – Cash Flow Calculator Default Information

The CFC has three basic classifications of revenue and expenditures. They consist of the Initial Investment, Annual Costs, and Annual Receipts. The Initial Investment is further subdivided into Land, Building, and Equipment. Each type of investments has its own effects upon the cash flow. For example, Figure 17 shows the data entry screen for Land. For tax purposes land does not depreciate, only buildings do; so, the land uses only an investment amount and a salvage value.

Under United States tax code, buildings are able to be depreciated. The main concern is whether they are commercial or residential and what month they are put into service. The depreciation used is straight line using a mid month convention utilizing 27.5 years for residential and 39 years for commercial property (See Figure 18).

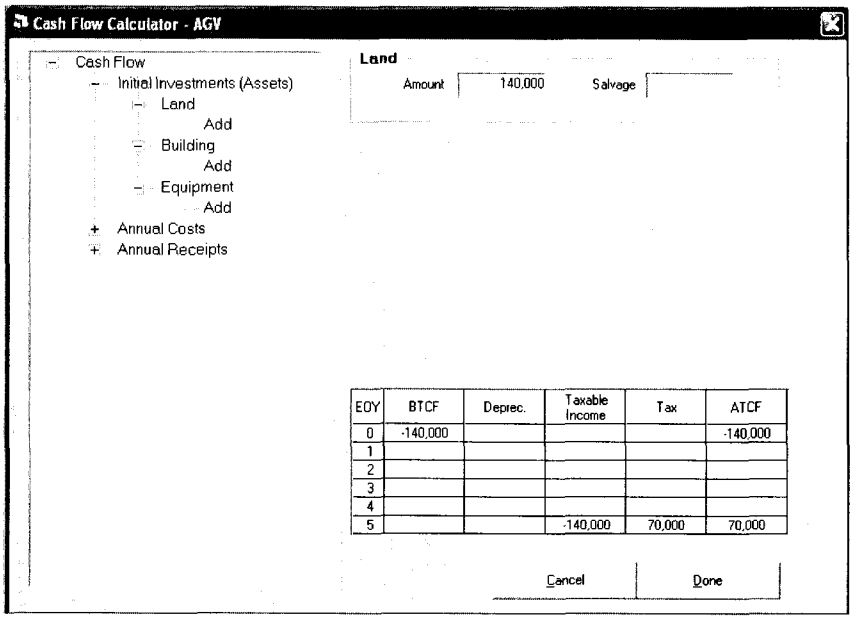

Figure 17 – Cash Flow Calculator – Land

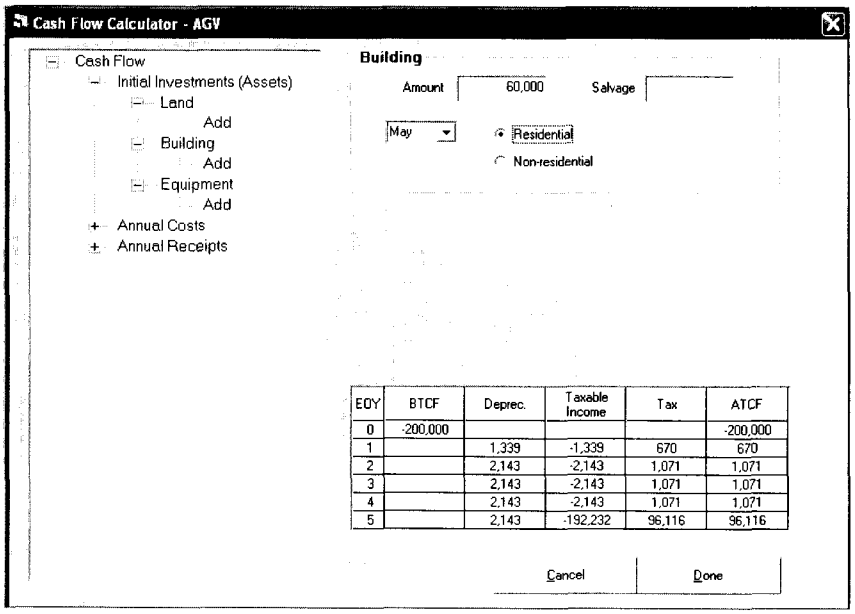

Figure 18 – Cash Flow Calculator – Building

Initial investment in equipment can be quite extensive and have a wide range of classifications in the eyes of the Internal Revenue Service (IRS) (See Figure 19). There are many different systems to depreciation equipment, but only one is allowed under the current tax laws. This system is called the Modified Accelerated Cost Recovery System

(MACRS). This system is separated into six broad categories that incorporate past depreciation system.

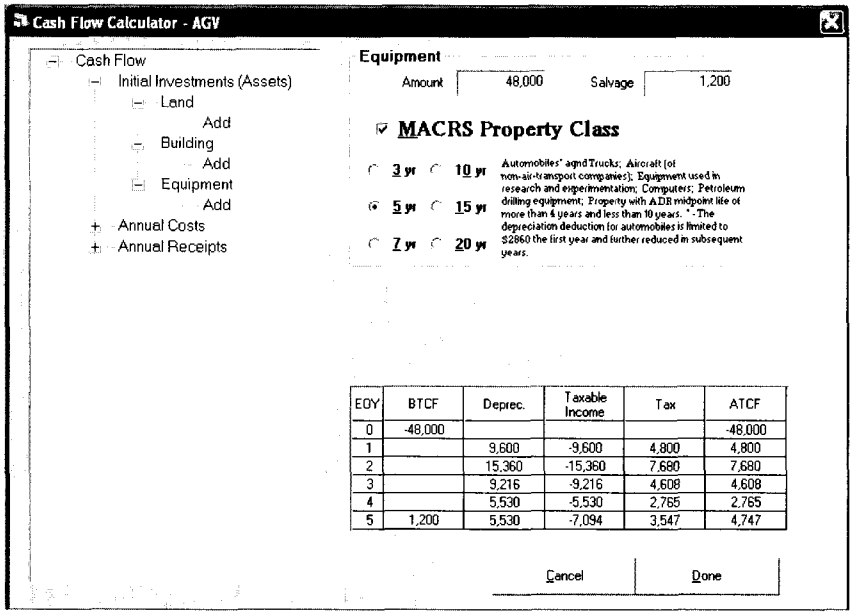

Figure 19 – Cash Flow Calculator – Equipment

MACRS six categories consist of a 3,5,7,10,15, and 20 year property class. All personal property except real estate is classified into one of these categories either by specific placement by name (such as computers in the five year category) or by classification of equipment (Vessel and water transportation equipment in the ten year category) or by default into the seven year classification (All other property not assigned to another class).

MARCS's depreciation schedule is based on two different methods. The first method is for the 3, 5, 7, and 10 year property categories. This category use double declining balance and then converts to straight line until the final year, which is one year longer than indicated. The 15 and 20 year property category use 150% declining balance and then transitions to straight line. So, a five year property classification actually takes six year to depreciate. This is done so that the salvage value can be assumed to be zero in

the depreciation schedule. To assist the chairman, in classifying equipment, a description is provided in JustMAT 1.2 when a property class is selected.

Annual costs and receipts are essentially the same because both can achieve the same results. The two are provided, however, to assist with the organization of an investment project. For example, in both sections dollars are input irrespective of the effect on the investment project. Costs look exactly like receipts; $5,000 is counted as an income in the receipts section and as an expense in the cost sections. Moreover, negative dollar values can be used in either section which means that receipts can be placed in the cost section by just making the dollar amount negative; although this should be avoided to reduce confusion, it is possible. Figures 20 and 21 are the annual costs and receipts screen shots.

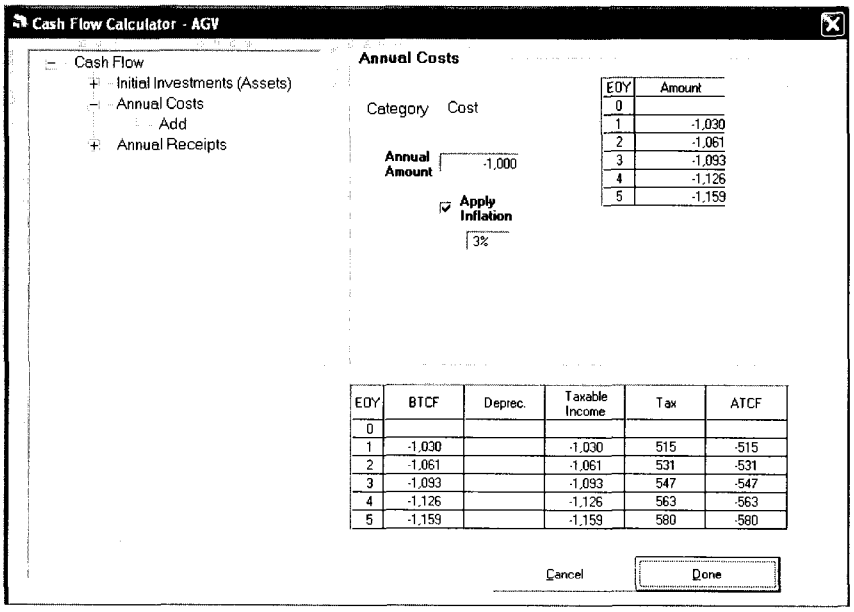

Figure 20 – Cash Flow Calculator – Annual Costs

Once the data entry for the project is complete, the CFC (frmAfterTax) passes the aggregate cash flows, ATCF, to the Cash Flow form (frmCashFlow).

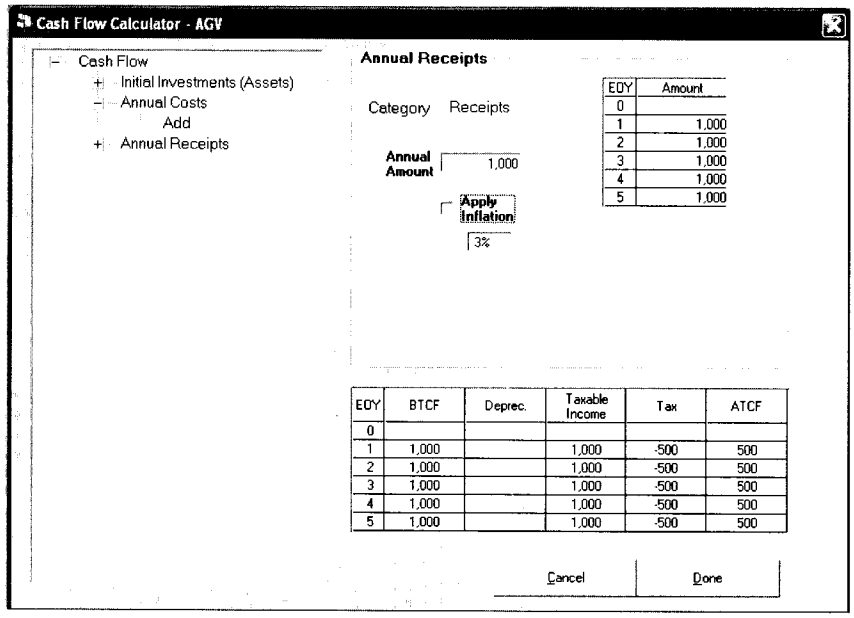

Figure 21 – Cash Flow Calculator – Annual Receipts

IV. CASE STUDY

The following section consists of an example demonstrating JustMAT. The example is a decision between two alternatives in a manufacturing facility where the project engineer is expanding the current production facility with a new assembly line utilizing either a conveyor system or an automated guided vehicle (AGV). The initial cost of the conveyor system is $15,000 and has no appreciable annual operating cost. The MARC for the conveyor is 7 years and the salvage value is $1,500. The AGV initial cost is $50,000 and has an annual operating cost of $1,000 inflated annually at 3%. The MARC for the AGV is 15 years and the salvage value is $5,000. Figures 22 through Figure 27 present the results of the case study. A committee is formed by the project engineer and two senior level management personnel who evaluate and rate this project. The project life is 15-years and the Minimum Acceptable Rate of Return (MARR) is 50%. Figure 18 shows the ranking of the three raters for the attributes.

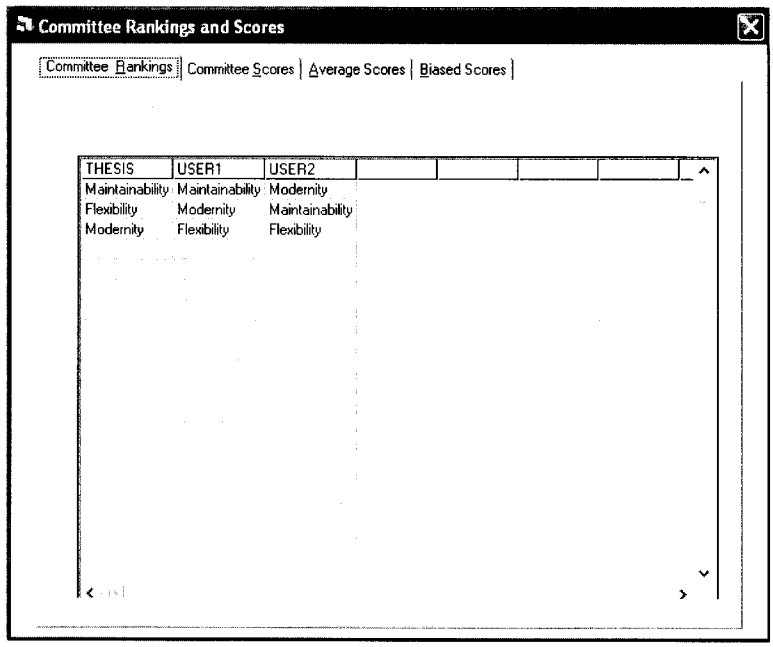

Figure 22 – Case Study Average Rankings

Figure 23 present the average scores for the three evaluators for each alternative and Figure 24 shows the equivalent annual cash flows for the net present worth analysis. These cash flows take into account the tax effects of each investment.

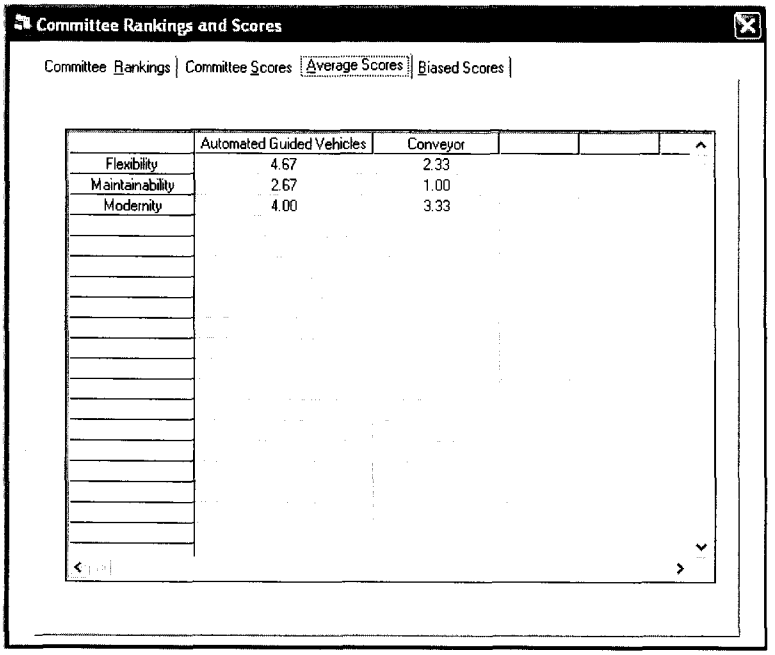

Figure 23 – Case Study Average Scores

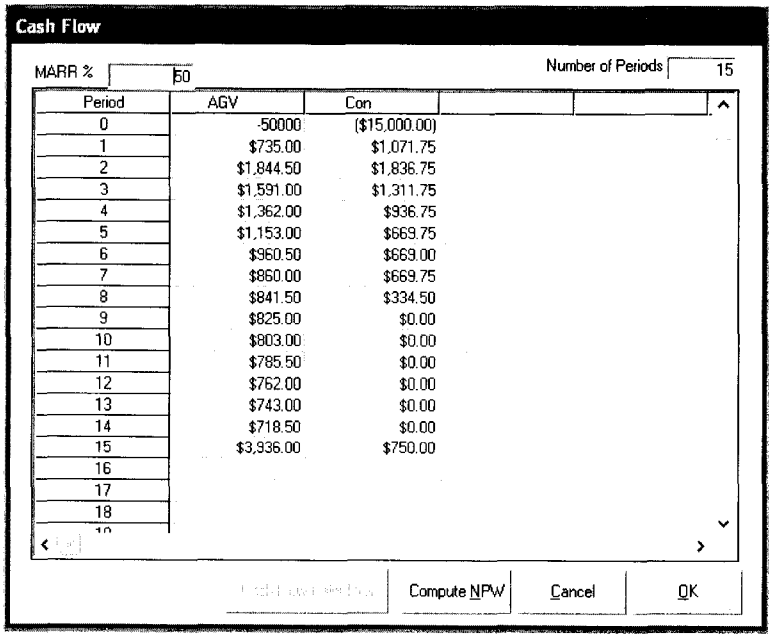

Figure 24 – Case Study Cash Flow

Figure 25 show the results of the net present worth analysis along with Figure 26 which shows the results of the Dunn-Rankin analysis.

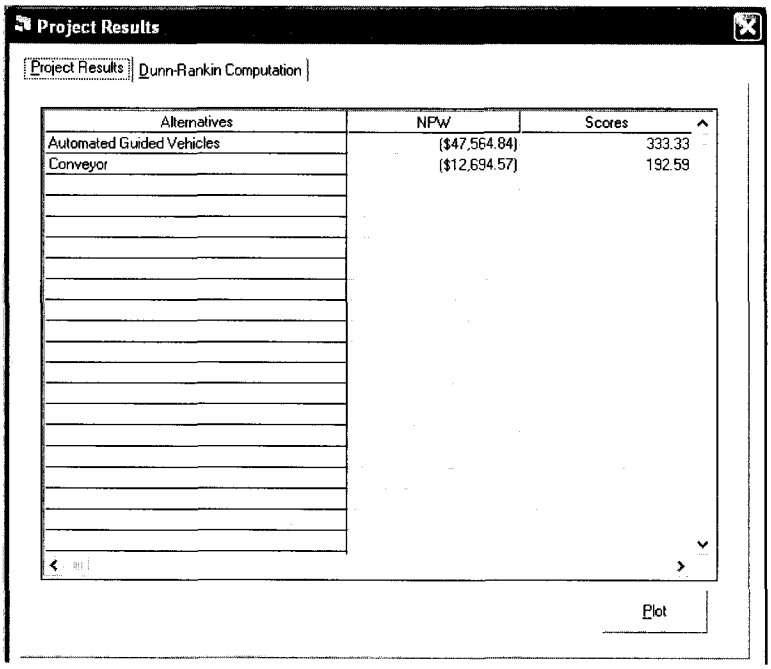

Figure 25 – Case Study Project Results

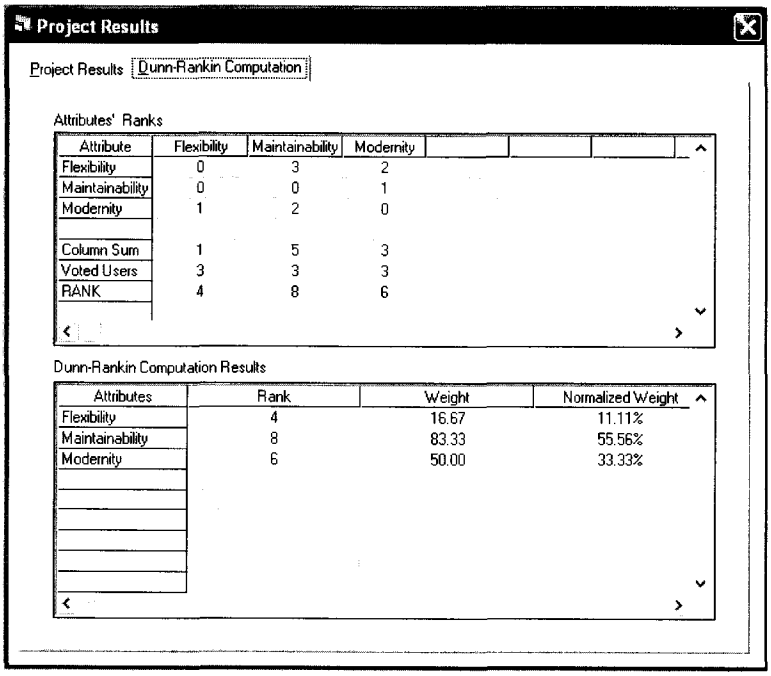

Figure 26 – Case Study Dunn-Rankin Results

From the results in Figure 25 and 26 a plot is made indicating the correct selection of the alternatives; Figure 27 shows the plot of the two alternatives, which indicate by the intersection of the two lines, that the intangible attributes are significant to the committee and therefore should not select the low cost alternative, but select the AGV system.

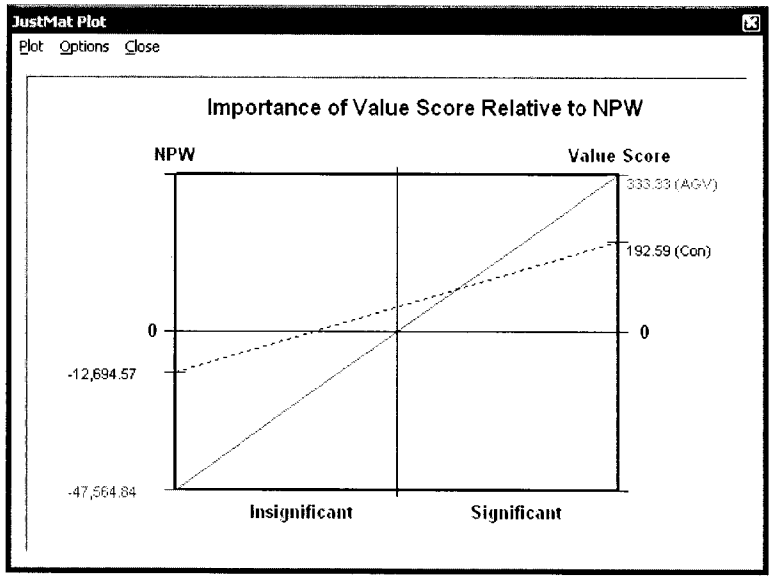

Figure 27 – Case Study Plot

V. CONCLUSIONS

A. Overview

Economic justification for complex systems, such as material handling equipment or computer network infrastructure is difficult to perform when many intangibles exist. JustMAT is a DSS that enables decision makers to use the group decision making process along with other proven analytical techniques to assist in the selection of the appropriate investment.

Presented in this thesis is the enhancement of the Decision Support System called JustMAT. The DSS software enhancement enables the net present worth analysis to be performed using either before or after tasks cash flows. The original JustMAT used only before tax cash flows and therefore ignored the consequences of taxes in the analysis. The addition of the after tax cash flows enabled a more precise analysis of the investment decision, which is particularly advantageous for investments that produce only cost savings.

B. Recommendations

The Internet has changed the way users and computers function. Through the Internet different computers easily communicate with each other and are located virtually anywhere. Adopting JustMAT to a web page format (web service) eliminates the platform and data transfer issues with the software. An inherent weakness of JustMAT is it's coding in Microsoft Visual Basic (VB). Although the Visual Basic program coding is very portable, it still requires programming for each operating system platform. The second weakness of the JustMAT is the necessity to move data files to each committee member's computer—on a LAN or WAN this is less of a hassle. A new requirement arises though from the web based design where offline access is needed for those without Internet access.

C. Areas for Further Study

Further research in four areas could enhance JustMAT. These four areas consist of the application to other fields besides computer infrastructure and material handling equipment, the application of Fuzzy Logic, consideration of the effects of a large group of raters, and the application of a statistical analysis on the results.

An application of this concept is possible for the transit industry where working groups are formed with the public and transit authority to make a group decision about public transportation. Fuzzy Logic may enhance JustMAT further by applying it to the ranking process. Applying JustMAT to the internet would enable a large number of members to rate projects. This large number may not have full information and the information provided to the rankings and scoring should be analyzed to ensure that the scoring was done consistently. Furthermore, additional data may need to be collected to see if there were racial, male/female, age related biases in the scoring. JustMAT is designed to operate in a group of relatively known individuals. The application of a statistical analysis to the results may provide additional information to the committee chairman. With the current design this may not be as important, but if this was used on a larger committee such as one in the thousands or tens of thousands of members. Just knowing if a score was biased may not be enough. The medium score, variances, and statistical test such as a Chi square test to see if the response from one group of voters were the same. The Chi square test would be important if independent public meetings were held where participants were enabled to rate the investment decision. It would be essential that the same information be provided and unbiased.

LIST OF REFERENCES

BIBILIOGRAPHY

Usher, J. S., Kamal, A. H., and Kim, S. W. 2001. A decision support system for economic justification of material handling investments. Computer & Industrial Engineering 39:35-47.

Sambasivarao, K. V. and Deshmukh, S. G. 1997. A decision support system for selection and justification of advance technologies. Production Planning & Control. 3:270-284.

Lavelle, J.P., Sorenson, G.E. and Aye, D.E. 1998. An integrated decision analysis tool for the economic evaluation and justification of computer integrated manufacturing technologies. International Journal of Industrial Engineering. 5(1):59-67.

Irani, Z., Ezingeard, J-N, and Grieve, R.J. 1999. Investment justification of information technology in manufacturing. International Journal of Computer Applications in Technology. 12(2):90-101.

Saleh, B., Randhawa, S. and Hacker, M. 1998. Decision support framework for evaluating manufacturing alternatives. International Journal of Flexibile Automation and Integrated Manufacturing. 6(3):311-327.

Sullivan, W.G., and Liggett, H.R. 1988. Decision support system for evaluating investments in manufacturing local area networks. Manufacturing Review. 1(3):151-157.

Sullivan. W.G. and Liu, M. Economic analysis of a proposed industrial robot. Industrial Engineering. 16(10):118-119.

Krinsky, I. And Miltenburg, J. 1991. Alternative method for the justification of advance manufacturing technologies. International Journal of Production Research. 29(5):997-1015.

Kakati, M. 1997. Strategic evaluation of advanced manufacturing technology. International Journal of Production Economics. 53(2):141-156.

Usher, J.S. and Kim, S.W. 1999. JustMAT 1.0: Economic Justification of Material Handling Investments User's Manual. Material Handling Institute.

Boucher, T. O. 1993. Review of Economic and Financial Justification of Advanced Manufacturing Technologies. The Engineering Economist. 38(4):348-351.

Liggett, H.R. and Sullivan, W.G. 1992. Multi-attribute evaluation of local area network topologies. The Engineering Economist. 37(2):91-114.

Demmel, J.G. and Askin, R.G. 1992. A multiple-object decision model for the evaluation of advanced manufacturing system technologies. Journal of Manufacturing Systems. 11(3):179-194.

BIBILIOGRAPHY
(Continued)

Chiadamrong, N. and O'Brien, C. 1999. Decision support tool for justifying alternative manufacturing and production control systems. International Journal of Production Economics. 61:177-186.

Pulford, S.A. 1999. Future scout and calvary systems constructive experiment defining and refining requirements in a future combat platform. M.Eng. University of Louisville.

Kim, S.W. 2001. Performance index material handling system: a simulation-based performability index (PI) approach. PhD. University of Louisville.

APPENDIX

frmaftertax - 1

```
'Option Explicit
Dim cr As Integer
Dim CashFlowName As String
Dim FocusValue As String
Dim DisplayMatrix(0 To maxItem, 0 To maxPeriods) As String

Dim BTCF(0 To maxItem, 0 To maxPeriods) As Double
Dim Annual(0 To maxItem) As String
Dim Percent(0 To maxItem) As String
Dim InfCHK(0 To maxItem) As Integer

'Equipment Dep.
Dim Dep(0 To maxItem, 0 To maxPeriods) As Double
Dim MACRS(0 To maxItem) As Integer
Dim df(1 To 21) As Double ' Depreciation Factor

'Building Dep.
Dim RealFactor As Long
Dim Build(0 To maxItem) As String
Dim Res(0 To maxItem) As Integer

'General Dep
Dim Land(0 To maxItem) As Boolean
Dim Cap(0 To maxItem) As Double
Dim Salvage(0 To maxItem) As Double
Dim LSalvage(0 To maxItem) As Double
Dim book(0 To maxItem, 0 To maxPeriods) As Double

Dim LastAmount As String

Dim ClickGrid As Boolean
Dim Inflate As Integer
Dim TaxRate As Double
Dim nodeI As Integer

Private Sub chkInflate_Click()
    Inflate = chkInflate.value
    InflateAmount
    'txtInflation.SetFocus
    BTCFUpdate
End Sub

Private Sub chkMACRS_Click()
    If chkMACRS.value = 1 Then
                            picMACRS.Visible = True
                            optMACRS(0).value = True
                            txtMACRSSum.Text = "Special handling devices for f
ood and beverage manufacture; Special tools for the manufacture of finished pl
astic products, fabricated metal products, and motor vehicles;  Property with
ADR midpoint life of 4 years or less."
                                df(1) = 0.3333
```

```
                                df(2) = 0.4445
                                df(3) = 0.1485
                                df(4) = 0.07451
                                For x = 5 To 21
                                    df(x) = 0
                                Next

                              CalcDep
                              CalcATCF
                            Else
                              picMACRS.Visible = False
                              For x = 1 To 21
                                  df(x) = 0
                              Next
                              CalcDep
                              CalcATCF
                              MACRS(nodeI) = -1
                            End If


End Sub
```

```
Private Sub cmbBuildMonth_KeyPress(KeyAscii As Integer)
    Select Case KeyAscii
        Case Is = 13
            optnonRes(1).SetFocus
    End Select
End Sub

Private Sub cmbBuildMonth_LostFocus()
    BuildDep

End Sub

Private Sub cmdCancel_Click()
Close #1
frmCashFlow.cmdPaste.Visible = False
Unload frmaftertax
Unload FrmNPW
End Sub

Private Sub CmdPrint_Click()

Dim BeginPage, EndPage, NumCopies, i
With CommonDialog1
        .PrinterDefault = True
```

frmaftertax - 3

```
    ' Set Cancel to True
        .CancelError = True
        On Error GoTo ErrHandler
    ' Display the Print dialog box
        .ShowPrinter
    ' Get user-selected values from the dialog box
        BeginPage = .FromPage
        EndPage = .ToPage
        NumCopies = .Copies
    For i = 1 To NumCopies
      ' Put code here to send data to the printer
      MsgBox ("Not yet Implemented")
    Next i
    Exit Sub
ErrHandler:
    ' User pressed the Cancel button
        Exit Sub
End With
End Sub

Private Sub cmdDone_Click()
    Unload frmaftertax
    Unload FrmNPW
    frmCashFlow.Visible = True
    frmCashFlow.SetFocus
    For x = 1 To AnalysisPeriod + 1
        frmCashFlow.GridCash.TextMatrix(x, Namecol) = Format(ATCF(x), "Currenc
y")
        'frmCashFlow.GridCash.TextMatrix(x, Namecol) = Format(ATCF(x), Currenc
yFormat)
    Next
    SaveData
    temp = nodeI
    For nodeI = 0 To maxItem
        clearNode
    Next
    nodeI = temp
    frmCashFlow.cmdPaste.SetFocus
    frmCashFlow.cmdPaste.Visible = False
End Sub
```

```
Private Sub Form_Load()
 cr = 1 'cost/revenue constant either 1 or -1
 frmaftertax.Top = 0
 frmaftertax.Left = 0

 RealFactor = 39
```

```
frmaftertax - 4

'Assign default MACRS
Dim t As Integer
'Dim nodex As Node
                df(1) = 0.3333
                df(2) = 0.4445
                df(3) = 0.1485
                df(4) = 0.07451
                For t = 5 To 21
                    df(t) = 0
                Next
'ArraySetup
'GridDisplay
'Dimension
frmaftertax.Caption = "Cash Flow Calculator - " & strAlterAbbr(Namecol - 1)
GridDisplaySetup
'TestDisplay
UpdateDisplay

    Dim x As Integer
    Dim NodParent As String
    Dim NodeKey As String
    Dim NodeText As String
    Dim CashFlow As String


    Dim LastNode As Integer
    Dim LastNodeKey As String

txtAnalysisPeriod.Text = frmCashFlow.txtPeriods.Text
txtAnalysisPeriod.Enabled = False ' Set so the value Cannot Be Changed
On Error GoTo FileError

ErrorReturn:

'Determine Project Name

    CashFlowName = Left(ProjFileName, (Len(ProjFileName) - 4)) + "-" + strAlte
rAbbr(Namecol - 1) + ".JMD"

    Open CashFlowName For Input As #1

    While Not EOF(1)
        Input #1, NodParent
        Input #1, NodeKey
        Input #1, NodeText
        If NodParent = "" Then
                                Set nodex = treeCashFlow.Nodes.Add(, , NodeKey
, NodeText)
                            Else
                                Set nodex = treeCashFlow.Nodes.Add(NodParent,
tvwChild, NodeKey, NodeText)
        End If
```

```
        Input #1, CashFlow
        While CashFlow <> "%&%"
            Input #1, CashFlow
        Wend
    Wend
    Close #1


    framDefault.Visible = True
    treeCashFlow.LabelEdit = tvwManual
    treeCashFlow.LineStyle = tvwRootLines
    treeCashFlow.Nodes(1).Expanded = True
    optnonRes(1).value = True    'Setting default for building type
    optMACRS(2).value = True     'Default setting for MACRS

'Set up Cash Flow form
    CashFlowSetup
    Exit Sub

FileError:
    Select Case Err

    Case Is = 53
        'MsgBox ("Creating New File")
        Open CashFlowName For Output As #1
            Write #1, "", "CashFlow", "Cash Flow", "%&%"
            Write #1, "CashFlow", "Initial", "Initial Investments (Assets)", "
%&%"
            Write #1, "CashFlow", "Cost", "Annual Costs", "%&%"
            Write #1, "CashFlow", "Receipts", "Annual Receipts", "%&%"
            'Write #1, "Receipts", "Sales", "Sales", "%&%"
            Write #1, "Receipts", "AddAR", "Add", "%&%"
            'Write #1, "CashFlow", "Add", "Add", "%&%"
            Write #1, "Initial", "Land", "Land", "%&%"
            Write #1, "Land", "AddL", "Add", "%&%"
            Write #1, "Initial", "Building", "Building", "%&%"
            Write #1, "Building", "AddB", "Add", "%&%"
            Write #1, "Initial", "Equipment", "Equipment", "%&%"
            Write #1, "Equipment", "AddE", "Add", "%&%"
            'Write #1, "Initial", "AddI", "Add", "%&%"
            'Write #1, "Cost", "Maintenance", "Maintenance", "%&%"
            'Write #1, "Maintenance", "MaiPart", "Part/Supplies", "%&%"
            'Write #1, "Maintenance", "MaiContract", "Contract", "%&%"
            'Write #1, "Maintenance", "AddM", "Add", "%&%"
            'Write #1, "Cost", "Labor", "Labor", "%&%"
            Write #1, "Cost", "AddAC", "Add", "%&%"
            'Write #1, "Labor", "LabOperators", "Operators", "%&%"
            'Write #1, "Labor", "LabMaintenance", "Maintenance", "%&%"
            'Write #1, "Labor", "LabSupervisory", "Supervisory", "%&%"
            'Write #1, "Labor", "LabClerical", "Clerical", "%&%"
            'Write #1, "Labor", "AddLab", "Add", "%&%"
```

```
            Close #1
        Case Else
            MsgBox ("Error " + Str(Err) + " Call for help!")
        End Select
Resume ErrorReturn
End Sub

Private Sub grdCashFlow_Click()
    With grdCashFlow
        .CellForeColor = Gray
    End With
    'MsgBox ("Click")
    BTCFUpdate
End Sub

Private Sub grdCashFlow_DblClick()
    response = MsgBox("Clear", vbYesNo)
    If response = vbYes Then grdCashFlow.Text = ""
    BTCFUpdate
End Sub

Private Sub grdCashFlow_GotFocus()
    txtAnnualAmount.ForeColor = Gray
    BTCFUpdate
End Sub

Private Sub grdCashFlow_KeyPress(KeyAscii As Integer)
    Dim letter As String

    letter = Chr(KeyAscii)

    With grdCashFlow

        .CellAlignment = flexAlignRightCenter
        Select Case KeyAscii
            Case Is = 13
                .CellForeColor = vbBlack 'Current Cell
                If .row + 1 < .Rows Then
                                    .Text = Format(.Text, CurrencyFormat)
                                    .row = .row + 1
                                Else
                                    .row = 2
                End If
                .CellForeColor = Gray  'New Cell
            Case Else
                If KeyAscii = 27 Then
                            .CellForeColor = vbBlack
                            GoTo Escape
                        End If
                If .CellForeColor = Gray Then
                                        .Text = ""
                                        .CellForeColor = vbBlack
```

```
                    End If
                Select Case KeyAscii
                    Case Is = 8

                        If Len(.Text) > 0 Then .Text = Left(.Text, Len(.Text)
- 1)
                    Case Is = 45, 46
                        .Text = .Text & letter
                    Case Is < 48
                       KeyAscii = 0
                    Case Is < 58
                       .Text = .Text & letter
                End Select
        End Select
    End With
    BTCFUpdate

Exit Sub
Escape:
    cmdDone.SetFocus

End Sub

Private Sub grdCashFlow_LeaveCell()
    With grdCashFlow
        .CellForeColor = vbBlack
        .Text = Format(.Text, CurrencyFormat)
        .CellAlignment = flexAlignRightCenter
    End With
    'MsgBox ("Leave")
    BTCFUpdate
End Sub

Private Sub grdCashFlow_LostFocus()
    With grdCashFlow
        .ForeColor = vbBlack
        .Text = Format(.Text, CurrencyFormat)
    End With
    BTCFUpdate
    'MsgBox ("Lost:")
End Sub

Private Sub grdCashFlow_RowColChange()
    With grdCashFlow
        .CellForeColor = vbBlack
        .Text = Format(.Text, CurrencyFormat)
        .CellAlignment = flexAlignRightCenter
        'MsgBox ("rowcol")
    End With
    BTCFUpdate

End Sub
```

```
Private Sub grdCashFlow_SelChange()
     With grdCashFlow
        .CellForeColor = Gray
        .Text = Format(.Text, CurrencyFormat)
        .CellAlignment = flexAlignRightCenter
        'MsgBox ("sel")
    End With
    BTCFUpdate

End Sub

Private Sub optMACRS_Click(Index As Integer)

    Select Case Index

        Case 0
            txtMACRSSum.Text = "Special handling devices for food and beverage
 manufacture; Special tools for the manufacture of finished plastic products,
fabricated metal products, and motor vehicles;  Property with ADR midpoint lif
e of 4 years or less."
                df(1) = 0.3333
                df(2) = 0.4445
                df(3) = 0.1485
                df(4) = 0.07451
                For x = 5 To 21
                    df(x) = 0
                Next
        Case 1
            txtMACRSSum.Text = "Automobiles* aqnd Trucks;  Aircraft (of non-ai
r-transport companies);  Equipment used in research and experimentation;  Comp
uters;  Petroleum drilling equipment;  Property with ADR midpoint life of more
 than 4 years and less than 10 years.  * - The depreciation deduction for auto
mobiles is limited to $2860 the first year and further reduced in subsequent y
ears."
                df(1) = 0.2
                df(2) = 0.32
                df(3) = 0.192
                df(4) = 0.1152
                df(5) = 0.1152
                df(6) = 0.0576
                For x = 7 To 21
                    df(x) = 0
                Next
        Case 2
            txtMACRSSum.Text = "All other property not assigned to another cla
ss;  Office furniture, fixtures, and equipment;  Property with ADR midpoint of
 10 years or more and less than 16 years."
                df(1) = 0.1429
                df(2) = 0.2449
```

```
                df(3) = 0.1749
                df(4) = 0.1249
                df(5) = 0.0893
                df(6) = 0.0892
                df(7) = 0.0893
                df(8) = 0.0446
                For x = 9 To 21
                    df(x) = 0
                Next
        Case 3
            txtMACRSSum.Text = "Assets used in petroleum refining and certain
food products;  Vessels and water transportation equipment;  Property with ADR
 midpoint life of more than 16 years and less than 20 years"
                df(1) = 0.1
                df(2) = 0.18
                df(3) = 0.144
                df(4) = 0.1152
                df(5) = 0.0922
                df(6) = 0.0737
                df(7) = 0.0655
                df(8) = 0.0655
                df(9) = 0.0656
                df(10) = 0.0655
                df(11) = 0.0328
                For x = 12 To 21
                    df(x) = 0
                Next
        Case 4
            txtMACRSSum.Text = "Telephone distribution plants;  Municipal sewa
ge treatment plants;  Property with ADR midpoint life of 20 years or more and
less than 25 years."
                df(1) = 0.05
                df(2) = 0.095
                df(3) = 0.0855
                df(4) = 0.077
                df(5) = 0.0693
                df(6) = 0.0623
                df(7) = 0.059
                df(8) = 0.059
                df(9) = 0.0591
                df(10) = 0.059
                df(11) = 0.0591
                df(12) = 0.059
                df(13) = 0.0591
                df(14) = 0.059
                df(15) = 0.0591
                df(16) = 0.0295
                For x = 17 To 21
                    df(x) = 0
                Next
        Case 5
            txtMACRSSum.Text = "Municipal Sewers;  Property with ADR midpoint
```

```
frmaftertax - 10

life of 25 years and more."
                df(1) = 0.0375
                df(2) = 0.07219
                df(3) = 0.06677
                df(4) = 0.06177
                df(5) = 0.05713
                df(6) = 0.05285
                df(7) = 0.04888
                df(8) = 0.04522
                df(9) = 0.0462
                df(10) = 0.04461
                df(11) = 0.04462
                df(12) = 0.04461
                df(13) = 0.04462
                df(14) = 0.04461
                df(15) = 0.04462
                df(16) = 0.04461
                df(17) = 0.04462
                df(18) = 0.04461
                df(19) = 0.04462
                df(20) = 0.04461
                df(21) = 0.02231
    End Select
    CalcDep
    MACRS(nodeI) = Index
    CalcATCF
End Sub

Private Sub optNonres_Click(Index As Integer)
    Select Case Index
        Case Is = 0
            RealFactor = 27.5
        Case Else
            RealFactor = 39
    End Select
    BuildDep
End Sub

Private Sub optnonRes_LostFocus(Index As Integer)
 Res(nodeI) = optnonRes(Index).Index

End Sub

Private Sub treeCashFlow_KeyDown(KeyCode As Integer, Shift As Integer)

    Dim DeleteNode As Integer

    With treeCashFlow
        DeleteNode = .SelectedItem.Index
        If KeyCode = vbKeyDelete Then
                        If .SelectedItem.Index <> 1 And _
                            .SelectedItem.Children = 0 Then
```

```
                                    If Left(.SelectedItem.Key, 3) <> "Add" Then
                                        .Nodes.Remove DeleteNode
                                        Cap(DeleteNode) = 0
                                        Salvage(DeleteNode) = 0
                                        .SelectedItem.Selected = True
                                        ClearFrame
                                        framDefault.Visible = True
                                        CalcATCF
                                    Else
                                        MsgBox ("Can not Delete Add")
                                    End If
                                End If
            End If
        End With
        If KeyCode = vbKeyReturn Then MsgBox ("not yet working")

           nodeI = DeleteNode
        clearNode

        'ClearDep
        CalcDep
        CalcATCF
        LoadGrid
End Sub

Private Sub treeCashFlow_NodeClick(ByVal Node As MSComctlLib.Node)
Picture2.Cls
Picture2.Print Node.Index
Picture3.Cls
Picture3.Print Node.Key



On Error GoTo ErrorHandler
txtCashFlowDisc.Visible = False
Text1.Text = Str(book(Node.Index, AnalysisPeriod))
'Default
'MsgBox (Node.Key + Str(Node.Index))
ClearFrame
If Node.Key = "CashFlow" Then framDefault.Visible = True Else framDefault.Visi
ble = False
nodeI = Node.Index
Select Case Left(Node.Key, 3)
    Case Is = "Cas"
        framDefault.Visible = True

    Case Is = "Bui"
     If (Node.Children < 2) Or (Left(Node.Parent, 3) = "Equ") Then
                                    FramBuilding.Visible = True
                                    txtAmount(0).Visible = True
                                    txtSalvage(0).Visible = True
                                    optnonRes(0).Enabled = True
```

```
                                optnonRes(1).Enabled = True
                                cmbBuildMonth.Enabled = True
                            Else
                                FramBuilding.Visible = True
                                txtAmount(0).Visible = False
                                txtSalvage(0).Visible = False
                                optnonRes(0).Enabled = False
                                optnonRes(1).Enabled = False
                                cmbBuildMonth.Enabled = False
                            End If
        If Node.Key = "Building" Then
                                    FramBuilding.Caption = "Building"
                             Else
                                    FramBuilding.Caption = Right(Node.Key, Len
(Node.Key) - 3)
                             End If
        txtAmount(0).Text = Format(Str(-Cap(nodeI)), CurrencyFormat)
        txtSalvage(0).Text = Format(Salvage(nodeI), CurrencyFormat)
        With cmbBuildMonth
            If Build(nodeI) = "" Then .Text = "Jan." Else .Text = Build(nodeI)
        End With
        optnonRes(Res(nodeI)).value = True

    'Case Is = "Mai", "Lab", "Uti"
    '    If (Node.Key = "Maintenance") Or (Node.Key = "Labor") Then ' Or (node
.Key ="Utilities") Then
    '                        txtCashFlowDisc.Visible = True
     ''                      txtCashFlowDisc.Text = "This category has been i
temized"
    '                   Else
    '                        txtCashFlowDisc.Visible = False
    '                        txtCashFlowDisc.Text = ""
    '               End If
    '    cr = -1
    '    framAnnualCost.Visible = True
    '    framAnnualCost.Caption = "Annual Costs"
    '    If (Left(Node.Key, 5) = "Maint" Or Left(Node.Key, 5) = "Labor" Or Lef
t(Node.Key, 5) = "Utili") Then
    '                                txtCategory.Text = Node.Key
    '                            Else
    '                                txtCategory.Text = Right(Node.Key, Len(No
de.Key) - 3)
    '                            End If
    '
    '
    Case Is = "Equ"
        If (Node.Children < 2) Or (Left(Node.Parent, 3) = "Equ") Then
                            framEquip.Visible = True
                            txtAmount(2).Visible = True
                            txtSalvage(2).Visible = True
                            chkMACRS.Enabled = True
                            picMACRS.Visible = True
```

```
                                Else
                                    framEquip.Visible = True
                                    txtAmount(2).Visible = False
                                    txtSalvage(2).Visible = False
                                    chkMACRS.Enabled = False
                                    If chkMACRS.Enabled = False Then chkMACRS.valu
e = 0
                                    picMACRS.Visible = False

                                End If
        If Node.Key = "Equipment" Then
                                      framEquip.Caption = "Equipment"
                                    Else
                                      framEquip.Caption = Right(Node.Key, Len(No
de.Key) - 3)
                                    End If
        txtAmount(2).Text = Format(Str(-Cap(nodeI)), CurrencyFormat)
        txtSalvage(2).Text = Format(Salvage(nodeI), CurrencyFormat)
        If MACRS(nodeI) <> -1 Then
                                      optMACRS(MACRS(nodeI)).value = True
                                      chkMACRS.value = 1
                                  Else
                                      chkMACRS.value = 0
                                      picMACRS.Visible = False
                                  End If
'         txtAmount(2).Text = Str(-Cap(nodeI))
'         txtSalvage(2).Text = Salvage(nodeI)

    Case Is = "Lan"
        If (Node.Children < 2) Or (Left(Node.Parent, 3) = "Lan") Then
                                  framLand.Visible = True
                                  txtAmount(1).Visible = True
                                  txtSalvage(1).Visible = True
                                                        Else
                                  framLand.Visible = True
                                  txtAmount(1).Visible = False
                                  txtSalvage(1).Visible = False
                                End If
    If Node.Key = "Land" Then
                                      framLand.Caption = "Land"
                                Else
                                      framLand.Caption = Right(Node.Key, Len(Nod
e.Key) - 3)
                                End If
        txtAmount(1).Text = Format(Str(-Cap(nodeI)), CurrencyFormat)
        txtSalvage(1).Text = Format(LSalvage(nodeI), CurrencyFormat)

      Case Is = "Rec"
        If (Node.Children < 2) And (Node.Parent = "Cash Flow") Then
                                  framAnnualCost.Visible = True
                                  framAnnualCost.Caption = "Annual Receipts"
                                  txtCategory.Text = Node.Key
```

```
                                framLand.Visible = False
                            Else
                                If Node.Parent <> "Cash Flow" Then
                                        framAnnualCost.Visible = True
                                        framAnnualCost.Caption = "Annual Recei
pts"
                                        txtCategory.Text = Right(Node.Key, Len
(Node.Key) - 3)
                                        framLand.Visible = False
                                        txtCashFlowDisc.Visible = False
                                    Else
                                        txtCashFlowDisc.Visible = True
                                        txtCashFlowDisc.Text = "Annual Receipt
s (Savings)"
                                       End If

                End If
        cr = 1
     Case Is = "Cos"
        If (Node.Children < 2) And (Node.Parent = "Cash Flow") Then
                                framAnnualCost.Visible = True
                                framAnnualCost.Caption = "Annual Costs"
                                txtCategory.Text = Node.Key
                                framLand.Visible = False
                            Else
                                If Node.Parent <> "Cash Flow" Then
                                        framAnnualCost.Visible = True
                                        framAnnualCost.Caption = "Annual Costs
"
                                        txtCategory.Text = Right(Node.Key, Len
(Node.Key) - 3)
                                        framLand.Visible = False
                                        txtCashFlowDisc.Visible = False
                                    Else
                                        txtCashFlowDisc.Visible = True
                                        txtCashFlowDisc.Text = "Annual Costs (
Expenses)"
                                        For y = 0 To maxPeriods
                                            BTCF(Node.Index, y) = 0
                                        Next y
                                        Annual(Node.Index) = 0
                                        Percent(Node.Index) = 0
                                        InfCHK(Node.Index) = 0
                                        CalcATCF
                                       End If

                End If
        cr = -1

    Case Is = "Ini"
        ClearFrame
        txtCashFlowDisc.Visible = True
```

```
        txtCashFlowDisc.Text = "Initial Investment. This category is used to e
nter data on assets including Land, Equipment, and Buildings."

    Case Is = "Add"
ErrorNameReturn:
        framDefault.Visible = False
        framAnnualCost.Visible = False
        FramBuilding.Visible = False
        framEquip.Visible = False
        framLand.Visible = False
            For y = 0 To maxPeriods
                BTCF(Node.Parent.Index, y) = 0
            Next y
            Annual(Node.Parent.Index) = 0
            Percent(Node.Parent.Index) = 0
            InfCHK(Node.Parent.Index) = 0
            CalcATCF
    If Node.Index + 1 = 100 Then GoTo MaxNodes
        NodeName = InputBox("Enter Name of New Category", Node.Parent.Key)
        If NodeName <> "" Then  ' The add needs to be left at the end if possi
ble.
                                Set nodex = treeCashFlow.Nodes.Add(Node.Key, t
vwFirst, (Left(Node.Parent.Key, 3) + NodeName), NodeName)

                                Node.Parent.Sorted = True
                            End If
                            Cap(Node.Parent.Index) = 0
                            Salvage(Node.Parent.Index) = 0
                            LSalvage(Node.Parent.Index) = 0
                            For x = 1 To maxPeriods
                                Dep(Node.Parent.Index, x) = 0
                            Next




    Case Else
        framDefault.Visible = False
        framAnnualCost.Visible = False
        FramBuilding.Visible = False
        framEquip.Visible = False
        framLand.Visible = False

    End Select
    CalcATCF
    LoadGrid
Exit Sub

ErrorHandler:
     Select Case Err
        Case Is = 35602
```

```
            MsgBox ("Please enter a unique name")
        Case Else
            MsgBox ("Call for Help " + Str(Err))
    End Select
Resume ErrorNameReturn

MaxNodes:
    MsgBox ("No More Nodes can Be Created")
End Sub


Public Sub CashFlowSetup()
    'Grid Size
With grdCashFlow
    .ColWidth(0) = 450
    .Rows = AnalysisPeriod + 2
    If ((AnalysisPeriod + 2) * 248) < 2685 Then
                                                .Height = ((AnalysisPeriod + 2
) * 248)
                                                .Width = 450 + 1255
                                            Else
                                                .Height = 2685
                                                .Width = 450 + 1255 + 305
                                            End If
'Column width Setup

    .ColWidth(1) = 1245
'Labels Assigned
    .col = 1
    .row = 0
    .CellAlignment = 3
    .Text = "Amount"
    .col = 0
    .CellAlignment = 3
    .Text = "EOY"
    For x = 0 To (AnalysisPeriod)
        .row = x + 1
        .Text = x
        .CellAlignment = 3
    Next
'set start cell
    .row = 2
    .col = 1
End With

End Sub



Private Sub txtAmount_GotFocus(Index As Integer)
    With txtAmount(Index)
        FocusValue = .Text
```

```
        .SelStart = 0
        .SelLength = Len(.Text)
    End With
End Sub

Private Sub txtAmount_KeyPress(Index As Integer, KeyAscii As Integer)
    Select Case KeyAscii

            Case Is = 8
            Case Is = 45
            Case Is = 27
                txtAmount(Index).Text = FocusValue
                cmdDone.SetFocus

            Case Is = 13
                txtSalvage(Index).SetFocus
            Case Is = 47
                KeyAscii = 0
            Case Is < 46
                KeyAscii = 0
            Case Is > 57
                KeyAscii = 0
        End Select
End Sub



Private Sub txtAmount_LostFocus(Index As Integer)
    With txtAmount(Index)
            Cap(nodeI) = -Val(Format(.Text, "#.##"))
            .Text = Format(.Text, CurrencyFormat)
    End With
    Select Case Index
        Case Is = 2
                        MACRS(nodeI) = chkMACRS.value
                        CalcDep

    Case Is = 0
                        BuildDep
    Case Is = 1
                        Land(nodeI) = True
    End Select

    CalcATCF
End Sub

Private Sub txtAnnualAmount_GotFocus()

    With txtAnnualAmount
        .ForeColor = vbBlack
        .SelStart = 0
        .SelLength = Len(.Text)
```

```
    End With
End Sub

Private Sub txtAnnualAmount_KeyPress(KeyAscii As Integer)


      Select Case KeyAscii

        Case Is = 8

        Case Is = 13
            frmaftertax.chkInflate.SetFocus
        Case Is = 47
            KeyAscii = 0
        Case Is = 45
        Case Is = 27 ' vbKeyEscape
            GoTo Escape
        Case Is < 46
            Keeyascii = 0
        Case Is > 57
            KeyAscii = 0

    End Select
    grdCashFlow.row = 1
    grdCashFlow.Text = ""
Exit Sub
Escape:
    cmdDone.SetFocus
    grdCashFlow.CellForeColor = vbBlack


End Sub

Private Sub txtAnnualAmount_LostFocus()
    With txtAnnualAmount
        If .Text <> "" Then
                    .Text = Format(.Text, "#.#")
                    .Text = Str((Abs(Val(.Text))) * cr)
                    .Text = Format(.Text, CurrencyFormat)
              End If
    End With
    InflateAmount
    BTCFUpdate
End Sub

Private Sub txtInflation_GotFocus()
    'chkInflate.value = 1
    With txtInflation
        .SelStart = 0
        .SelLength = Len(.Text)
    End With
End Sub
```

```
Private Sub txtInflation_KeyPress(KeyAscii As Integer)
     Select Case KeyAscii

        Case Is = 8

        Case Is = 9, 13
            frmaftertax.grdCashFlow.SetFocus
            grdCashFlow.row = 2 'Assume there should be no year 0 cash flows.
            grdCashFlow.col = 1
        Case Is = 45
        Case Is = 47
            KeyAscii = 0
        Case Is < 46
            KeyAscii = 0
        Case Is > 57
            KeyAscii = 0

    End Select
End Sub

Private Sub txtInflation_LostFocus()
    With txtInflation
        .Text = Str(Val(.Text)) + "%"
    End With
    InflateAmount
    'grdCashFlow.SetFocus Thought it would be better setfocus on the grid but
didnot work
    cmdDone.SetFocus
    BTCFUpdate
End Sub



Private Sub txtSalvage_GotFocus(Index As Integer)
    With txtSalvage(Index)
        FocusValue = .Text
        .SelStart = 0
        .SelLength = Len(.Text)
    End With
End Sub

Private Sub txtSalvage_KeyPress(Index As Integer, KeyAscii As Integer)
    Select Case KeyAscii

            Case Is = 8
            Case Is = 45
            Case Is = 27
                txtSalvage(Index).Text = FocusValue
                cmdDone.SetFocus
            Case Is = 47
                KeyAscii = 0
```

```
            Case Is = 13
                treeCashFlow.SetFocus
            Case Is < 46
                KeyAscii = 0
            Case Is > 57
                KeyAscii = 0

        End Select
End Sub



Private Sub txtSalvage_LostFocus(Index As Integer)
    With txtSalvage(Index)
         Select Case Index
            Case Is = 1
                LSalvage(nodeI) = Val(Format(.Text, "#.##"))
                Land(nodeI) = True
            Case Else
                Salvage(nodeI) = Val(Format(.Text, "#.##"))
        End Select
        .Text = Format(.Text, CurrencyFormat)
    End With
'    CalcDep
    CalcATCF
End Sub

Private Sub txtTaxRate_GotFocus()
    With txtTaxRate
        FocusValue = .Text
        .SelStart = 0
        .SelLength = Len(.Text)
    End With
End Sub

Private Sub txtTaxRate_KeyPress(KeyAscii As Integer)
  Select Case KeyAscii

        Case Is = 8
        Case Is = 27
            txtTaxRate.Text = FocusValue
            cmdDone.SetFocus
        Case Is = 9
             treeCashFlow.TabIndex = 2
        Case Is = 47
            KeyAscii = 0
        Case Is = 13
            frmaftertax.treeCashFlow.SetFocus
        Case Is < 46
            KeyAscii = 0
        Case Is > 57
```

```
frmaftertax - 21

            KeyAscii = 0

    End Select
End Sub

Private Sub txtTaxRate_LostFocus()
    With txtTaxRate
        TaxRate = Val(.Text)
       .Text = Str(Val(.Text)) + "%"
    End With
    CalcATCF
End Sub

Public Sub ArraySetup()
    Cap(1) = -100000
    Cap(1) = 20000
    BTCF(1, 1) = -10000
    BTCF(1, 2) = -200
    BTCF(1, 3) = -3000
    BTCF(1, 4) = -10000
    Dep(1, 1) = 20000
    Dep(1, 2) = 32000
    Dep(1, 3) = 19200
    Dep(1, 4) = 11520
    book(1, 1) = 80000
    book(1, 2) = 48000
    book(1, 3) = 28800
    book(1, 4) = 17280
 '   Cap(2, 0) = -100000
 '   Cap(2, 4) = 20000
 '   BTCF(2, 1) = -10000
 '   BTCF(2, 2) = -200
 '   BTCF(2, 3) = -3000
 '   BTCF(2, 4) = -10000
 '   Dep(2, 1) = 20000
 '   Dep(2, 2) = 32000
 '   Dep(2, 3) = 19200
 '   Dep(2, 4) = 11520
 '   book(2, 1) = 80000
 '   book(2, 2) = 48000
 '   book(2, 3) = 28800
 '   book(2, 4) = 17280

End Sub

Public Sub GridDisplaySetup()
    Const NumberOfRows = 2530
    Const cwidth = 1075
    Dim x, y As Integer
    With DisplayGrid
        .col = 1
        .row = 1
```

```
        .CellAlignment = flexAlignRightCenter
        .ColWidth(0) = 430
        .Rows = AnalysisPeriod + 2

        If ((AnalysisPeriod + 1) * 300) + .ColWidth(0) < NumberOfRows Then
                                                .Height = ((AnalysisPeriod + 1
) * 248) + 470
                                                .Width = 5925
                                            Else
                                                .Height = NumberOfRows
                                                .Width = 6151
                                            End If

        'Set titles
        .row = 0
        .RowHeight(0) = 450
        .col = 0
        .Text = "EOY"

        For x = 0 To AnalysisPeriod
            .row = x + 1
            .CellAlignment = flexAlignCenterCenter
            .Text = Str(x)
        Next
        .row = 0
        '.col = 1
        '    .Text = "Capital"
        '    .ColWidth(1) = CWidth
        '    .CellAlignment = flexAlignCenterCenter

        .col = 1
            .Text = "BTCF"
            .ColWidth(.col) = cwidth
            .CellAlignment = flexAlignCenterCenter
        .col = 2
            .ColWidth(.col) = cwidth
            .Text = "Deprec."
            .CellAlignment = flexAlignCenterCenter
        ' .col = 3
        '    .ColWidth(.col) = CWidth
        '    .Text = "Capital Gains"
        '    .CellAlignment = flexAlignCenterCenter
        .col = 3
            .ColWidth(.col) = cwidth
            .Text = "Taxable Income"
            .CellAlignment = flexAlignCenterCenter
        .col = 4
            .ColWidth(.col) = cwidth
            .Text = "Tax"
            .CellAlignment = flexAlignCenterCenter
        .col = 5
            .ColWidth(.col) = cwidth
```

```
            .Text = "ATCF"
            .CellAlignment = flexAlignCenterCenter
        'Set cell Alignment
        For x = 1 To .Rows - 1
            For y = 1 To .Cols - 1
                .row = x
                .col = y
                .CellAlignment = flexAlignCenterCenter
            Next
        Next
    End With
End Sub

Public Sub InflateAmount()
    With grdCashFlow
        .col = 1
        For x = 1 To AnalysisPeriod
            .row = x + 1
            .CellAlignment = flexAlignRightCenter
            If Inflate = 1 Then
                            txtAnnualAmount = Format(txtAnnualAmount, "#")
                            .Text = Format(Str(Val(txtAnnualAmount) * ((1 + (V
al(Left(txtInflation, Len(txtInflation) - 1)) / 100)) ^ x)), CurrencyFormat)
                            txtAnnualAmount = Format(txtAnnualAmount, Currency
Format)
                        Else
                            .Text = Format(txtAnnualAmount, CurrencyFormat)
                End If
        Next
        .row = 2
    End With
    txtAnnualAmount.ForeColor = vbBlack

End Sub


Public Sub TestDisplay()
For x = 1 To maxItem
    For y = 1 To maxPeriods
        DisplayMatrix(x, y) = Str(x * y)
    Next
Next
End Sub

Public Sub UpdateDisplay()

    Dim x, y As Integer
    With DisplayGrid
        For x = 1 To .Rows - 1
            For y = 1 To .Cols - 1
                .TextMatrix(x, y) = Format(DisplayMatrix(x, y), CurrencyFormat
)
```

```
            Next
        Next
    End With
End Sub

Public Sub CalcATCF()
    Picture1.Cls
    Picture1.Print "Start of Calculations"


    Dim Capital As Double
    Dim TotalBTCF As Double
    Dim TotalDep As Double
    Dim itm, prd As Integer

    Dim TotalLSalvage As Double
    Dim TotalSalvage As Double
    Dim TotalBook As Double
    Dim TotalGain As Double
    Dim Gain As Double

    TotalBTCF = 0
    TotalDep = 0
    TotalGain = 0
    TotalBook = 0
    TotalSalvage = 0
    For prd = 0 To AnalysisPeriod
Picture1.Cls
Picture1.Print "Period "; prd
        For itm = 0 To maxItem - 1
            'Picture1.Cls
            'Picture1.Print itm
             If prd = 0 Then
                          Capital = Cap(itm)
                      Else
                          If prd = AnalysisPeriod Then
                                                      Capital = Salvage(itm)
 + LSalvage(itm)
                                                      If Land(itm) = True Th
en Gain = LSalvage(itm) + Cap(itm) Else Gain = 0
                                                      TotalSalvage = TotalSa
lvage + Salvage(itm)
                                                      TotalBook = TotalBook
+ book(itm, AnalysisPeriod)
                                                  Else
                                                      Capital = 0
                                                  End If
                End If

            TotalBTCF = TotalBTCF + Capital + BTCF(itm, prd)
                'Picture1.Print "Total BTCF "; TotalBTCF
            TotalDep = TotalDep + Dep(itm, prd)
```

```
                'Picture1.Print "Total Dep "; TotalDep
            TotalGain = Gain + TotalGain
            TotalLSalvage = TotalLSalvage + LSalvage(itm)
                'Picture1.Print "Total gain "; TotalGain
        Next

        DisplayMatrix(prd + 1, 1) = TotalBTCF
        TotalBTCF = 0
        DisplayMatrix(prd + 1, 2) = TotalDep
        TotalDep = 0 'Depreciation

    Picture1.Print TotalSalvage; " - "; TotalBook; "="; TotalSalvage - TotalBo
ok
        Select Case prd
            Case Is = 0
                        DisplayMatrix(prd + 1, 3) = 0

            Case Is = AnalysisPeriod
                        DisplayMatrix(prd + 1, 3) = DisplayMatrix(prd + 1, 1)
- DisplayMatrix(prd + 1, 2) + TotalGain - TotalLSalvage - (TotalBook) ' - Tota
lSalvage)

            Case Else
                        DisplayMatrix(prd + 1, 3) = DisplayMatrix(prd + 1, 1)
- DisplayMatrix(prd + 1, 2)
            End Select
            rt = (-Val(Format(txtTaxRate.Text, "#.###")))
            Text1.Text = Str(rt)
        DisplayMatrix(prd + 1, 4) = DisplayMatrix(prd + 1, 3) * rt '(-Val(Form
at(txtTaxRate.Text, "#.#")))
        DisplayMatrix(prd + 1, 5) = Str(Val(DisplayMatrix(prd + 1, 1)) + Val(D
isplayMatrix(prd + 1, 4)))
        ATCF(prd + 1) = DisplayMatrix(prd + 1, 5)
        TotalLSalvage = 0

    Next

    UpdateDisplay
    Picture1.Print "Exiting Calculation"

End Sub

Public Sub ClearFrame()
    framDefault.Visible = False
    framAnnualCost.Visible = False
    FramBuilding.Visible = False
    framEquip.Visible = False
    framLand.Visible = False
End Sub

Public Sub BTCFUpdate()
```

```
    For x = 0 To AnalysisPeriod
        BTCF(nodeI, x) = Val(Format(grdCashFlow.TextMatrix(x + 1, 1), "#"))
    Next
    Annual(nodeI) = txtAnnualAmount.Text
    Percent(nodeI) = txtInflation.Text
    InfCHK(nodeI) = chkInflate.value
    CalcATCF
End Sub

Public Sub LoadGrid()
    For x = 0 To AnalysisPeriod
        grdCashFlow.TextMatrix(x + 1, 1) = Format(BTCF(nodeI, x), CurrencyForm
at)
    Next
    txtAnnualAmount.Text = Annual(nodeI)
    chkInflate.value = InfCHK(nodeI)
    If Percent(nodeI) <> "" Then txtInflation.Text = Percent(nodeI) Else: txtI
nflation.Text = "3%"
End Sub

Public Sub CalcDep()
    'Clear book value
    For x = 0 To 21
        book(nodeI, x) = 0
    Next
    book(nodeI, 0) = -Cap(nodeI)
     Picture1.Cls
     Picture1.Print Cap(nodeI); book(nodeI, 0)
     For t = 1 To 21
        Dep(nodeI, t) = -Cap(nodeI) * df(t)
        book(nodeI, t) = book(nodeI, t - 1) - Dep(nodeI, t)
        If book(nodeI, t) < 0 Then book(nodeI, t) = 0
        Picture1.Print Dep(nodeI, t); book(nodeI, t)
     Next
End Sub



Public Sub SaveData()
    On Error GoTo ErrorHandler
    'Open CashFlowName For Output As #1
    'Close #1
Exit Sub

ErrorHandler:
    Select Case Err
        Case Is = 55  ' File Already Open
            Close #1
        Case Is = 58  'File already Exist
            Set fs = CreateObject("Scripting.FileSystemObject")
            'Set fs = CashFlowName
            fs.Delete CashFlowName
```

```
        Case Else
            MsgBox ("Error is " + Str(Err))
    End Select
Resume Next
End Sub

Public Sub BuildDep()
    Dim EntryMonth As Integer

    With cmbBuildMonth
        Select Case .Text
            Case Is = "Jan."
                EntryMonth = 11
            Case Is = "Feb."
                EntryMonth = 10
            Case Is = "Mar."
                EntryMonth = 9
            Case Is = "Apr."
                EntryMonth = 8
            Case Is = "May"
                EntryMonth = 7
            Case Is = "Jun."
                EntryMonth = 6
            Case Is = "Jul."
                EntryMonth = 5
            Case Is = "Aug."
                EntryMonth = 4
            Case Is = "Sept."
                EntryMonth = 3
            Case Is = "Oct."
                EntryMonth = 2

            Case Is = "Nov."
                EntryMonth = 1
            Case Is = "Dec."
                EntryMonth = 0
        End Select
    End With
    Picture1.Cls
    Picture1.Print Str(nodeI)
    Picture1.Print

        Dep(nodeI, 1) = ((EntryMonth + 0.5) / 12) / RealFactor * Val(Format(tx
tAmount(0).Text, "#"))
        book(nodeI, 1) = Val(Format(txtAmount(0).Text, "#")) - Dep(nodeI, 1)
    Picture1.Print Dep(nodeI, 1); book(nodeI, 1)
    'Depreciation for year 2 and beyond
    For x = 2 To AnalysisPeriod
        Dep(nodeI, x) = Val(Format(txtAmount(0).Text, "#")) / RealFactor
        book(nodeI, x) = book(nodeI, x - 1) - Dep(nodeI, x)
        Picture1.Print Dep(nodeI, x); book(nodeI, x)
        Picture1.Print Build(nodeI); Str(Res(nodeI))
```

```
        'MsgBox (Format(txtAmount(0).Text, "#") + Str(Dep(nodeI, x)))

    Next x
    Build(nodeI) = cmbBuildMonth.Text
    CalcATCF


End Sub


Public Sub clearNode()
For x = 0 To maxPeriods
    BTCF(nodeI, x) = 0
    book(nodeI, x) = 0
    Dep(nodeI, x) = 0
Next
Annual(nodeI) = 0
Percent(nodeI) = 0
InfCHK(nodeI) = 0

'Equipment Dep.
MACRS(nodeI) = 0
'Building Dep.
Build(nodeI) = 0
Res(nodeI) = 0

'General Dep
Cap(nodeI) = 0
Salvage(nodeI) = 0
LSalvage(nodeI) = 0
Land(nodeI) = False

End Sub
```

VITA

The author, John James Frank, is married to Karen Jill (Slusher) Frank. He is the first-born son of Kenneth John and Barbara Ann Frank. He was born on April 20, 1969 in Livonia, Michigan. He is the father of two children—Courteney Mädchen and Nicholas Eric Frank.

He attended Oakland Community College before transferring to Michigan Technological University in Houghton, Michigan. From there he joined the United States Army as a scout and completed two and a half years of active duty. In 1993, he joined the Kentucky Army National Guard and began his studies at the University of Louisville. He was commissioned an officer in the Kentucky Army National Guard in August 1996 and completed his bachelors of science in industrial engineering in 1999.

As a co-op student in college, he worked for Clark Memorial Hospital as a management engineer. After completing his bachelor's degree, he joined American Consulting Engineers where he has been working as a transit engineer specializing in economic and financial data. His most recent work involved the upgrade of a bus vehicle maintenance facility with a new paint booth and a fluid distribution system. This work was interrupted by his mobilization for National Defense in Operation "Noble Eagle II", where he spent 12 months on active duty supporting the US Army during operations "Enduring Freedom" and "Iraqi Freedom".

He entered the University of Louisville Speed Scientific School and the Business School to complete a joint Masters of engineering and a Masters of Business Administration. In December 2003, he was awarded both degrees.