Rule-Based Flight Software Cost Estimation

Sherry A. Stukes[1] and Dr. John N. Spagnuolo, Jr.[2]
Jet Propulsion Laboratory, California Institute of Technology

This paper discusses the fundamental process for the computation of Flight Software (FSW) cost estimates. This process has been incorporated in a rule-based expert system [1] that can be used for Independent Cost Estimates (ICEs), Proposals, and for the validation of Cost Analysis Data Requirements (CADRe) submissions. A high-level directed graph (referred to here as a decision graph) illustrates the steps taken in the production of these estimated costs and serves as a basis of design for the expert system described in this paper. Detailed discussions are subsequently given elaborating upon the methodology, tools, charts, and caveats related to the various nodes of the graph.

We present general principles for the estimation of FSW using SEER-SEM as an illustration of these principles when appropriate. Since Source Lines of Code (SLOC) is a major cost driver, a discussion of various SLOC data sources for the preparation of the estimates is given together with an explanation of how contractor SLOC estimates compare with the SLOC estimates used by JPL. Obtaining consistency in code counting will be presented as well as factors used in reconciling SLOC estimates from different code counters. When sufficient data is obtained, a mapping into the JPL Work Breakdown Structure (WBS) from the SEER-SEM output is illustrated. For across the board FSW estimates, as was done for the NASA Discovery Mission proposal estimates performed at JPL, a comparative high-level summary sheet for all missions with the SLOC, data description, brief mission description and the most relevant SEER-SEM parameter values is given to illustrate an encapsulation of the used and calculated data involved in the estimates.

The rule-based expert system described provides the user with inputs useful or sufficient to run generic cost estimation programs. This system's incarnation is achieved via the C Language Integrated Production System (CLIPS) and will be addressed at the end of this paper.

[1] Software Systems Engineer, Cross Mission System Development And Implementation, 4800 Oak Grove Drive/301-270, Pasadena, CA 91109, Senior Member.
[2] Cost Engineer, Engineering Cost Estimation, 4800 Oak Grove Drive/301-465, Pasadena, CA 91109, and Non-member.

Nomenclature

ATLO	=	Assembly, Test, Launch Operations
BOE	=	Basis of Estimate
CADRe	=	Cost Analysis Data Requirement
C&DH	=	Command and Data Handling
CEH	=	Cost Estimating Handbook
CER	=	Cost Estimating Relationship
CLIPS	=	C Language Integrated Production System
EM	=	Engineering Model
EOM	=	End of Mission
ESLOC	=	Equivalent (new) Source Lines of Code
FFRDC	=	Federally Funded Research and Development Center
FSW	=	Flight Software
FY	=	Fiscal Year
GN&C	=	Guidance, Navigation and Control
GSW	=	Ground Software
ICE	=	Independent Cost Estimate
I&T	=	Integration and Test
ITAR	=	International Traffic in Arms Regulations
JPL	=	Jet Propulsion Laboratory
KB	=	Knowledge Base
LCC	=	Life Cycle Cost
Mgmt	=	Management
NPR	=	NASA Procedural Requirement
ONCE	=	One NASA Repository
S/C	=	Spacecraft
SCHERRI	=	Software Cost Heuristics Embedded in a Rule-Based Reasoning Infrastructure
SDC	=	Software Development Contactor
SE	=	Systems Engineering
SEER-SEM	=	System Evaluation and Estimation Review – Software Estimation Model
SLiC	=	Software Line Counter (code counter)
SLOC	=	Source Lines of Code
SMART	=	Software Measurement Analysis Repository Tool
SQI	=	Software Quality Improvement
SW	=	Software
WBS	=	Work Breakdown Structure

I. Introduction

COST estimation at CalTech's Jet Propulsion Laboratory (JPL) has in recent years become a crucial part of the mission formulation process [2][3][4]. Further, the rigor and exactitude constituting the basis of these estimates is attaining an importance that is becoming more and more pronounced with the passage of time [5][6]. Implicit in these analyses is a reliable and accurate estimation of the software costs involved in spacecraft, instruments (payload), simulation and testbeds, ground systems for commanding the spacecraft and instruments, and science data processing. Such software analysis is also required to support two other kinds of activities: Independent Cost Estimates (ICE's) and Cost Analysis Data Requirement (CADRe) documents.

ICE's are an integral part of the cost verification process to ensure that costs are reasonable. They may be variously requested by the project, NASA's Independent Program Assessment Office (IPAO)[3] and JPL's Cost Analysis & Pricing section. ICE's are required at milestone reviews and are performed separately from the project. Depending on how many ICE's are performed, a reconciliation exercise may be conducted in order to understand the differences in content and scope of the estimates. This allows for a single best estimate. During this process, interactions with project personnel are usually discouraged. One or more project independent data sources must be used to derive a software cost estimate.

[3] The main role of the IPAO is to enable the independent review of the NASA's Programs and to ensure mission success.

A CADRe is a report that provides present and future researchers with an encapsulated presentation of the technical and cost data of a project. The project could have already reached 'End of Mission' (EOM), or could be ongoing. A 'Software Metrics Section' in the report is used to categorize the modules of the software as defined by the project. Correspondingly, various parameters relating to the module are listed such as Source Lines of Code (SLOC), programmer and analyst experience, security requirements, multi-site development, work hours etc. These project given parameters (except for work hours) are used in conjunction with a computer program of choice to produce a software cost estimate for the project. The validation of the project given parameters is attained if the work hours produced by the program come close to those given by the project. If not, further project interaction and analysis is needed.

This paper focuses on the work done in computing the FSW costs for N_0 proposals done in the Engineering Cost Analysis Group. The techniques embedded in this work overlap considerably with those used for ICE's and CADRe's but differ in the sense that the work had to be done quickly and for many missions at once. It was therefore imperative that certain techniques and procedures had to be developed which not only streamlined the flight software analysis process but which also provided instantaneous confirmation that the data and processes used for these estimates was consistent across the board.

The execution of software cost analyses for so many projects as described above suggested the existence of general patterns that could be followed which were, in effect, a part of all software cost analysis. Therefore, aside from presenting the results of the analysis and describing what was done to get them, a high level generalized decomposition and illustration of the above mentioned techniques and procedures in a clear form is presented. Typically, a decision tree is used for such purposes. However, to give the reader insight as to what direction he or she should take for the creation of a cost analysis for a given project, it was decided that a decision tree with all its inherent detail would blur the high level concepts and direction for developing such an analysis. Hence, the embodiment of the implemented considerations took the form of high level directive 'boxes' followed by tree like alternatives given rise to as a result of these 'boxed' directives. The resulting structure will be referred to as a decision graph. In essence, this decision graph represents the structuring of the thought processes and data acquisition necessities of FSW cost estimates as they were done here. This will be referred to as Knowledge Engineering the estimate.

Formally, Knowledge Engineering (as defined by Edward Feigenbaum and Pamela McCorduck in 1983 [1]) "… is that discipline that involves integrating knowledge into computer systems in order to solve complex problems normally requiring a high level of human expertise". Embedded in this definition is the acquisition and structuring of the related information characterizing the knowledge domain of interest [7]. The decision graph described above relates to such acquisition, structuring, and representation of knowledge as it is applied to the computation of FSW estimates. Although the process at this point is not automated, various aspects of the work are embedded in and related to computer activity. Further, the work is done in such a way as to facilitate further automation of its procedures.

This paper is not only a description per se of the efforts by two software cost analysts. It is also an outline of the methodology used for FSW cost analysis presented in a form that serves as a foundation upon which others may gain insight into how to do FSW cost analyses for their own problems at hand. Further, at the end of the paper, we describe a rule-based expert system, Software Cost Heuristics Embedded in a Rule-Based Reasoning Infrastructure (SCHERRI), which incorporates all of the ideas discussed. The program is written in the C Language Integrated Production System (CLIPS) and incorporates approximately 145 rules.

II. Flight Software Cost Estimation for N_0 Type X Class Proposals

As mentioned in the introduction, this paper focuses on the development of FSW cost estimates for N_0 Type X class missions at the Jet Propulsion Laboratory of the California Institute of Technology. A Type X class mission is defined by the Type X Announcement of Opportunity (AO) issued by the National Aeronautics and Space Administration (NASA). In accord with this specification, the missions under discussion are of 3 types: Inner Heliosphere, Earth Orbiter, and Primitive Body Encounter.

Aside from the rigor and detail inherent in the execution of such analyses, the work was further complicated due to the relatively short deadlines and the varying schedules and availability of the cost leads for each proposal. This made it difficult if not impossible to get the job done in a timely fashion unless several FSW cost estimates were done simultaneously. Work was done on one or more proposals to the greatest extent possible, and then as 'information/personnel for discussion' became available on others, work proceeded to them. Further, when more personnel or data became available for analyses previously initiated, work resumed on them and so on. To maintain consistency in the analyses as well as to facilitate an immediate view of data obtained and data needed at any point of the estimation process, the results obtained at all stages of the work were tabulated in a large Excel™ spreadsheet . In the end, all data used in the computation of all flight software estimates were on this spreadsheet. The sheet thus stemmed as a necessity due to the parallel nature of the work being done.

It became clear to the authors, however, that the above spreadsheet not only served as an encapsulation of the data used in the N_0 estimates. It also helped illustrate the process involved in obtaining such estimates, suggesting a 'decision tree' structure [8] which in essence characterized the flight software cost estimation process used. A drawback is that such a tree structure is somewhat tedious and intricate to the point of hiding concepts and procedures important to the decision making process. To illustrate these fundamental building blocks of the FSW analysis work done here, a variant of the decision tree is used. Boxes giving directives followed by node structures listing the possibilities resulting from these directives are used. Such a structure is, in this paper, referred to as a 'decision graph'. This decision graph structure is more compact and intuitively palatable than a decision tree and expresses high level relationships and concepts to the point of suggesting to the reader how to construct his or her own decision graph for their own estimates. A decision tree would easily follow (see Section VII, "Summary and Future Work").

A notional view of the full decision graph is given in Figure 1. Details are discussed in subsequent sections. The variation in text coloring in the decision graph corresponds to the breakout of the sections which will discuss it. The corresponding color bars that appear above/below the portions of the graph designate the section numbers that the graph portions are discussed in. Note further that the decision boxes and column nodes are numbered $\boldsymbol{C_i}$ and $\boldsymbol{D_i}$ for ease of discussion [9].

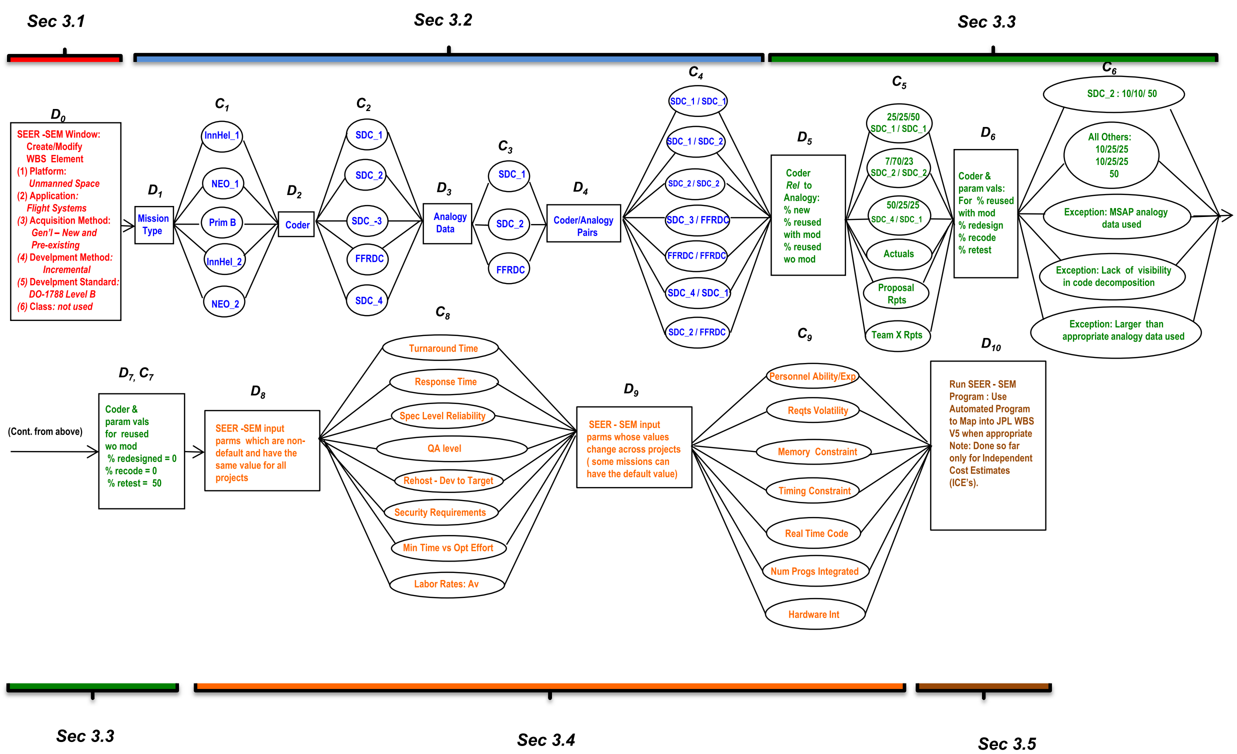


Figure 1: Decision Graph for Flight Software Cost Analysis of N_0 Type X Class Proposals

Even though the necessities of the FSW estimation process 'mothered' the need for a spreadsheet which gave rise to a decision graph, the ensuing discussions do not follow that ordering. The authors feel that the ideas implicit in this paper are best conveyed by discussing the details of the decision graph first and then illustrating the compiled spreadsheet data which served as its genesis.

It should be finally noted that the development of the spreadsheet played a pivotal role in formulating a unique sequence of events characterizing the effort discussed in this paper. Typically, decision tree formulation or rule-based development efforts begin with a domain expert/knowledge engineer workforce decomposition. The work presented here, however, resulted from the *evolution* of a FSW cost analysis team into the classic expert/engineer breakout via the development of the worksheet.

III. The Decision Graph

As shown in Figure 1, there are five 'color sub-graphs'. Each will have its own subsection in this chapter, be reproduced in a larger form for ease of reading, and discussed in detail.

As often occurs when discussing research with respect to real life data and organizations, much of the inherent information is of a proprietary nature. To allow useful discussion of the issues, the following variable representation of the real life entities are given.

Table 1 gives the variable representation of the mission types for which the FSW analyses were done.

In the actual cost exercise, there could exist several missions for each mission type. Variable names for each mission are not necessary for the purposes of this paper.

Table 1: Variable Representation of Real Life Mission Types

Mission Type	Description
Inn_Hel_1	Inner Heliosphere Mission (examples: Venus, Mars)
Inn_Hel_2	Inner Heliosphere Mission (examples: Venus, Mars)
NEO_1	Near Earth Orbiter (examples: Earth , Moon)
NEO_2	Near Earth Orbiter (examples: Earth , moon)
Prim B	Primitive Body Encounter (examples: comets, asteroids)

The organizations responsible for developing the code are roughly of two types: SW Development Contractors (SDC's) and Federally Funded Research Development Centers (FFRDC's). The SDC's are represented by SDC_1, SDC_2, SDC_3, SDC_4 and any FFRDC is simply represented by the acronym FFRDC.

A. Establishing Initial SEER-SEM Inputs

Due to the large number of costs that had to be estimated in such a short time, a parametric software cost model was used.[4] It is described in the following paragraphs.

1) Parametric Cost Model Overview

The System Evaluation and Estimation of Resources - Software Estimating Model (SEER-SEM)[5] was selected for use in the proposal effort because it is widely accepted within NASA and industry for software cost estimation and analysis.

The model is based on approximately 6,700 historical data points that are used to create the internal model equations. Approximately 30% (2,000) of the historical programs are based on Commercial environments and the remaining 70% are defense related programs. The model's internal database is significant because it is the basis for the default Knowledge Bases (KB's)[10] that represent cost drivers for the FSW estimate.

SEER-SEM requires four basic categories of information that represent the input data to the model. These categories include:

- **Software Systems Work Breakdown Structure (WBS)** - Identification of the software modules being developed (to the configuration item where possible, but often times to the software subsystem level).
- **Software Size** - The number of logical source lines of code (SLOC). This includes code that is anticipated to be reused from similar software developments. SLOC may be entered into the model using least likely, most likely, and highest likely values to reflect the uncertainty of the software size.
- **Knowledge Bases** - SEER-SEM contains industry data that is supplemented with related historical data that was used to calibrate, or adjust, the model parameters to reflect historical experience.
- **Parameter Settings** – Parameter settings are initially established by the selected SEER-SEM knowledge bases and have been adjusted to reflect proposal-specific knowledge. Parameter settings may also be entered as least likely, most likely, and highest likely values to reflect uncertainty.

[4] A Parametric Cost Model refers to a mathematical representation of cost estimating relationships (CERs) that provides a logical and predictable correlation between the physical or functional characteristics of a system, and the resultant cost of the system. A parametric cost model is an estimating system comprised of CERs and other parametric estimating functions, e.g., software size, amount of reused software, staff skills, and development environment. Parametric cost models yield product or service costs at designated levels and may provide departmentalized breakdown of generic cost elements. A parametric cost model provides a logical and repeatable relationship between input variables and resultant costs.

[5] SEER-SEM Version 8.0, Galorath Incorporated, Los Angeles, CA.

2) Initial Input Data

Decision box D_0, shown in Figure 2, identifies the knowledge bases required by SEER-SEM (e.g., Platform, Application, etc.). The selection of the appropriate subsets of these knowledge bases (e.g., unmanned space, flight systems, etc.) is the basis for the creation of default input parameters for SEER-SEM. Certain of these default parameters will be adjusted based upon the procedures and techniques discussed throughout the paper.

D_0

SEER -SEM Window:
Create/Modify
WBS Element
(1) Platform:
Unmanned Space
(2) Application:
Flight Systems
(3) Acquisition Method:
Gen'l – New and
Pre-existing
(4) Develpment Method:
Incremental
(5) Develpment
Standard:
DO-1788 Level B
(6) Class: *not used*

Figure 2: Initial SEER-SEM inputs

Table 2 provides definitions for the knowledge bases and identifies the selection made for use in establishing the initial model input parameters.

Aside from the initial inputs established by the SEER-SEM KB, additional data reflecting other facets of FSW cost analysis is required to run the SEER-SEM model. The approach used for obtaining this data included collecting historical data and descriptive information required as input to the model. This process is described in detail later in the paper. During this initial phase, the software architecture, related new and reused code estimates, knowledge base selections, and parameter setting adjustments, were closely coordinated and reviewed with the technical points of contact.

Table 2: Knowledge Base Definitions and Selections

Knowledge Base	Definition	Selection
(1) Platform	Establishes a collection of input parameter settings that characterize a particular host environment.	Unmanned Space
(2) Application	Establishes a collection of input parameter settings that characterize an application or application technology type.	Flight Systems
(3) Acquisition Method	Establishes a collection of input parameter settings that characterize from where the software will come.	New and Reuse
(4) Development Method	Establishes a collection of input parameter settings that characterize the particular Software Development Life Cycle method that will be used.	Incremental Development
(5) Development Standard	Establishes a collection of input parameter settings that characterize the software development process standard that will be used.	DO-178B Level B
(6) Class	A knowledge base calibrated to a specific set of data or domain.	Not used

B. Mission Category and Coder/Analogy Data Pair

After the initial inputs are fed to the program, further numerical and quantitative characteristics deemed important with respect to the FSW cost evaluation process have to be determined for each Computer Software Configuration Item (CSCI)[11]. The authors, in every case, found the initial reasoning as given in Figure 3 to be crucial in this respect. In what follows, we discuss the nature of each node structure under its ‘decision box’ heading.

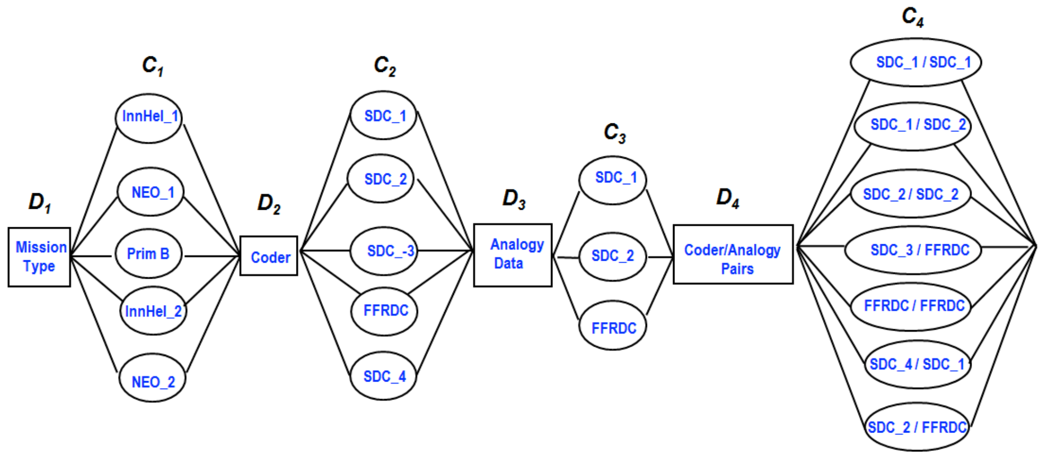


Figure 3: Mission Types, Developers, and Analogy Data Decision Dynamics

1) ***Mission Type***— The first concern is the nature of the mission. If the project does not give SLOC values for FSW, the estimator will have to locate data to determine approximate SLOC values for the FSW. The SLOC data used depends upon how similar the mission that it was developed for is to the one of current interest. It is therefore important to classify the missions of interest to the level that that similarity can be established. The classifications for the Type X proposals as given in C_1 in Figure 3 were inner heliosphere, near earth orbiters and primitive bodies and are listed in accordance to the variable names presented in Table 1.

2) ***Software Developer (Coder): SDC or FFRDC***— Knowledge of the organization assigned to develop the FSW for the proposal is important because that information, in conjunction with the analogy data available, will determine subsequent numerical SEER-SEM inputs (as discussed in Section 3.3 below).

   This knowledge is not always known at the beginning of a proposal. Sometimes it changes during the course of a proposal. This can, and does, cause a significant cost change during the estimation process. In the absence of any knowledge of the coder, the analyst and cost lead agree on a best guess as to whom the coder might be and the estimate is made with that assumption. The options are listed in C_2.

3) ***Analogy Data***— Once the nature of the mission has been studied, the appropriate analogy data must be determined. The analogy data used consisted of code developed by the organizations as listed in C_3. This data can be obtained by stored samples of code, reports (previous step 2 proposals, for example) or Technical Data Packages. In one case, there was a step 2 report giving actual SLOC values from a previous version of the mission of interest. In another case, there was a Technical Data Package (TDP) for a mission which was deemed very analogous to the proposal of interest. This TDP had SLOC values in it, and these were used. The vast majority of cases, however, required a search for data when the proposal gave inheritance directives without SLOC values or, in fact, when no inheritance directives were given at all. It was then up to the FSW analyst to determining appropriate data analogy sources for SLOC values approximating those that would apply to the Type X proposal at hand.

4) ***Software Development Contractor/Analogy Data Pair***— If it is known that Company X[12] is writing the SW, and we have analogy data that Company X developed, then that affords a cost advantage as compared to the case where Company X is doing the FSW and analogy data from Company Y is being used. In the first case, code already exists that Company X can use to do the present mission with. Further, having done the code, Company X has experience and infrastructure for that code. In the latter case, even though the Company Y analogy data can approximate the amount of code needed, there may be a lot of code and corresponding resources that Company X has to develop that it may not have developed enough to be consummate with Company Y’s work. The coder/analogy pairs shown in the final tree structure indicates those combinations experienced in the Type X proposal experience[13]. The determination of these pairs is important to the FSW cost computation process in ways which will be discussed in the remaining Decision Graph subsections.

C. Quantitative Parameter Determination

This portion of the decision graph uses the coder/analogy data pairs to determine several sets of numerical inputs to SEER-SEM which, in addition to SLOC values, are major cost drivers. Once the SLOC values are obtained, it is crucial to the cost estimating process to determine how much of the SLOC is new, reused without (wo) modifications (mods) and reused with mods. It is also important to determine, for the code that is reused with modifications, the percentages corresponding to redesign, recode and retest. These 3 percentage categories also apply to code that is reused as is, but in these analyses they are given fixed values of 0%, 0% and 50% for all proposals. Details and justifications regarding the elements described above are as follows.

For purposes of explanation, the triplet:

(% new, % reused wo mods, % reused with mods)

is referred to as vector 1 and the triplet:

(% redesign, % recode, % retest)

is called vector 2.

These percentages are applied to the delivered code size of the analogy data to produce Equivalent Source Lines of Code (ESLOC). ESLOC is defined as the equivalent 'new' size of delivered code after taking into account the percent new, percent reused 'as is', and percent reused modified. Also factored in are the percent redesign, percent recode, and percent retest as applied to both the percent reuse 'as is' code and the percent modified code.

1) ***Coder Relationship to Analogy Data as it Determines Vector 1***— As can be seen in Figure 4, the decision box, D_5, indicating the need for determination of vector 1 is followed by C_5 which shows several alternative sources for determining the value of this vector as encountered during the proposal cost estimating process. When actual % values for the vector were obtained with analogy data, proposal reports or Team X reports, they were used. In the absence of this data, default values based on cost estimating experience were used. For example, assume SDC_1 was developing the code and the analogy data (SLOC values) used was developed by SDC_1 as well. If the delivered logical SLOC size of the analogy data was X, Then if SDC_1 were to write code for the project, it typically would be approximately the same delivered size, X, as that of the analogy data but would be such that:

New code = 25% X
Reused Code wo mods = 25% X
Reused Code with mods = 50% X

which gives:

Vector 1 = (25, 25, 50).

The same reasoning is applied if SDC_2 were the contractor for the S/C and SDC_2 analogy data was used. In this case, based upon FSW cost estimating experience, vector 1 would have the entries:

New code = 7% X
Reused Code wo mods = 70% X
Reused Code with mods = 23% X

which gives:

Vector 1 = (7, 70, 23).

Entry of this vector as opposed to the one corresponding to SDC_1 generally results in a lower FSW cost due to the coupling of a lower new code % and a higher reused wo mod code %. This is consistent with the fact that SDC_2 code development is more of a 'production line' as compared to SDC_1's.

In the case where the analogy data used was *not* developed by the assigned contractor, experience dictates that the % of new code developed would be somewhat larger than the 2 previous cases mentioned. The degree to which this is true depends on the assigned contractor. In the case of SDC_4 being the assigned contractor where SDC_1 analogy data of delivered size X is used.

New code = 50% X
Reused Code wo mods =25% X
Reused Code with mods = 25% X

yielding:

Vector 1 = (50, 25, 25).

As stated above, when actuals are obtained with the vector 1 values, or if available reports give these percentages (sometimes with SLOC values), then the above discussed default reasoning is overridden and those values are used.

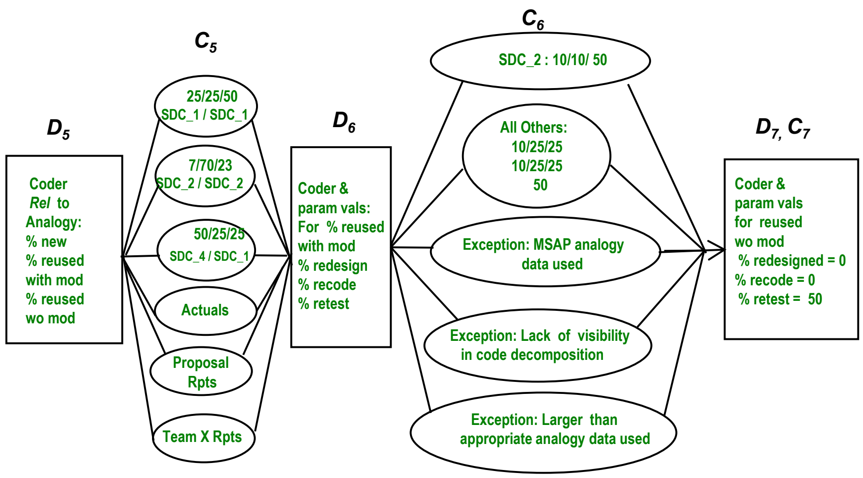


Figure 4: Reasoning for Quantitative Input Determination

2) ***Software Development Contractor and Parameter Values as they Relate to Vector 2***— The reasoning involved with vector 2 is computationally and conceptually similar to that of vector 1. In the case of SDC_2, experience with its product line code indicates that, in general, if X represents the amount of reused modified code, then:

Redesigned code = 10%X
Rewritten code = 10%X
Retested code = 50%X

yielding:

Vector 2 = (10, 10, 50).

In general, for those cases where the SW contractor was not SDC_2 and the analogy data was not SDC_2, each of the slots characterizing vector 2 entries was not given a single value but a range of values[14]. More precisely for the '% redesign' SEER-SEM input parameter, there are 3 values for input into SEER-SEM: A 'Least Likely' value for % redesign, a 'Highly Likely' value for % redesign and a 'Most Likely' value for % redesign (similarly for the value of %recode)[15]. As there is more uncertainty regarding the values of these parameters, a distribution of values as opposed to a single value was given with respect to the SEER-SEM input parameter rows.

The values used for the non SDC_2 cases thusly are:

Least likely value for % redesign = 10
Highly likely value for % redesign = 25
Most likely value for % redesign = 25

Least likely value for % recode = 10
Highly likely value for % recode = 25
Most likely value for % recode = 25

Least likely value for % retest = 50
Highly likely value for % retest = 50
Most likely value for % retest = 50.

These values were used because they tended to represent the cases where a non-product line FSW contractor (not at the level of experience of SDC_2) was used. Exceptions to this rule included cases where reusable FSW data was used. In that case, lower single valued % values, equal to those used for SDC_2, provided sufficient accuracy. Other exceptions existed as shown in Figure 4.

3) ***Treatment of Reused Code without Modification***— Finally, SEER-SEM requires inputs for % redesign , % recode and % retest for the code designated as reused without modification as a means to measure how 'New' the code is.

For all missions across the board, it is assumed that the % of code designated as new does not require any redesign or recoding. It however is assigned a value of 50% retest. This is due to the fact that in SEER-SEM 100% retest means that 52% of the code includes the work relating to test plans, test procedures, test drivers, and test scripts. The 48% requires the actual retesting and integration of the code. Again, it is the assumption that this new code does not require the activities which comprise the 52%. It only requires pure testing and integration. Hence 50% was chosen for convenience as it was close enough to 48%.

D. Non - Default Parameter Identification

This section deals with the assignment of Type X mission values to the SEER-SEM parameters not yet discussed in this paper. Figure 5 gives the decision graph component dealing with this issue.

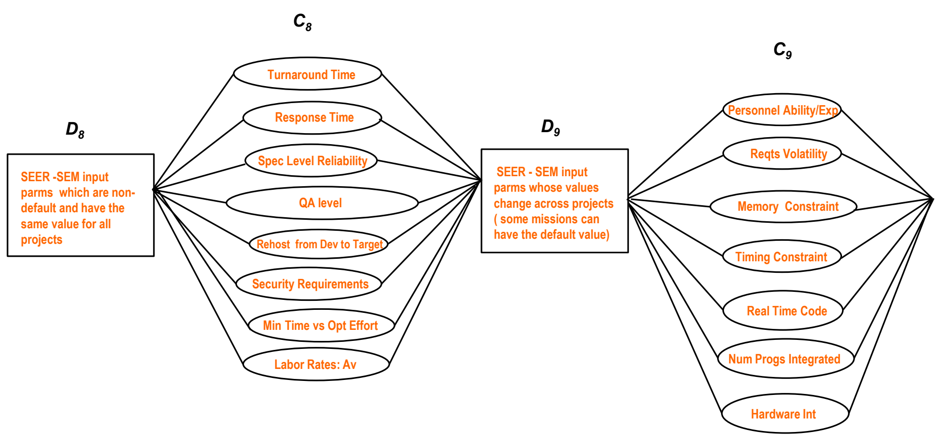


Figure 5: Non Default Knowledge Base Breakout

Decision box D_8, Figure 5, indicates those input parameters in SEER-SEM for which:

(1) the default values assigned by the program are not appropriate for the Type X missions

and

(2) which can be assigned a value which is the same for all of the missions.

The nodes in column C_8 give a listing of all parameters for which this is true.

Decision box D_9 in Figure 5 indicates the existence of parameters whose values varied from mission to mission followed by a listing of those parameters in C_9.

Any parameters appearing (other than the ones discussed in the sections above) that are not of the types described in this section have the SEER-SEM default values assigned to them. This is the case because, at that early stage in the cost estimation process, it was unrealistic to assign anything else. Table 3 gives the name and description of those parameters corresponding to the first decision box and the reasons as to why the default values are not appropriate in the missions studied. Further, the table justifies the values assigned in this costing effort.

Table 3: Reasoning for Non-Default Non-Varying Parameter Assignment

Parameter	Definition	SEER-SEM Default Value	Reason for Not Using Default	Value Given	Reason for Value Given
Turnaround Time	The time required to create a release version of the software solution.	LOW-	*Outdated Default Value*	VLO	*More reflective of Recent HW/SW Reality*
Response time	Rates the average transaction response time from the moment a developer presses a key or click a command, until that command is acknowledged and its action is completed.	NOM+	*Outdated Default Value*	LOW	*More reflective of Recent HW/SW Reality*
Spec Level reliability	Rates the level of documentation required. The level of documentation is often dictated by the development standard being used with government contracted software developments.	NOM	*Outdated Default Value*	HI-	*More reflective of Recent HW/SW Reality*
QA Level	Evaluates the completeness of the Quality Assurance (QA) activities. The Quality Assurance effort is usually directly related to the impact that a failure in the software would have during its operational phase.	VHI-	*Outdated Default Value*	HI	*More reflective of Recent HW/SW Reality*
Rehost from Development to Target	Rates the effort to convert the software from the development system to the target system on which the software will execute.	VHI	*Outdated Default Value*	HIGH-	*More reflective of Recent HW/SW Reality*
Security Requirements	Rates development impacts of security requirements for the delivered target system. (All classifications are identified in the Orange book.)	HI	*Security default value is too High for the work at hand*	NOM	*Security is Nominal for NASA Unmanned Space Work*
MinTime vs Opt Effort	Choose between optimizing the schedule or the effort estimate. Optimizing for schedule (minimum time) assumes the development will be finished as quickly as possible. Optimizing for effort assumes the software will be developed as cheaply as possible, but will take longer to complete.	Min Time	*Min Time has inappropriately high cost used only in special time constrained cases*	Optimal Effort	*Min Time would yield unallowable and unrealistic FSW costs*
Labor Rates Average	The average monthly labor rate for all personnel working on the project.	$28,400 per WM (FY10)	*Used an average of industry and JPL Values*	$xx per WM (FY10)	*Appropriate to use an average of industry and JPL Values*

The parameters from the "Reason for Not Using Default" column in Table 3 require additional elaboration:

Outdated Default Value – Due to the model not keeping pace with the state-of-the-art in software development.

Security default value is too High for the work at hand – Based on the National Security Agency (NSA) "Orange Book" [16].

Min Time has inappropriately high cost used only in special time constrained cases – Min Time is used only when there is a scheduling constraint.

Used an average of industry and JPL values – These values are based on an industry survey[17].

Table 4 gives the name and description of those parameters corresponding to the second decision box and the reasons as to why the values varied from mission to mission.

Table 4: Reasoning for Variable Assignments to Parameters

Parameter	Definition	Reason for Variation
Personel Ability/Experience	Characteristics of the software development personnel	Coder/Analyst Ability varies from company to company
Requirements Volatility	How frequently the customer changes the software development requirements	Industry varies from JPL
Memory Constraint	Is there sufficient memory to meet the systems requirements	Altered to adjust for lack of visibility in decomposition
Timing Constraint	Is the timing requirement met	Varies with GN&C complexity
Real Time Code	The amount of code that requires an instantaneous response	Altered to adjust for lack of visibility in decomposition
Number of Programs Being Integrated	How many CSCIs are concurrently being integrated	Have Different Numbers for Different Projects - Some not Broken out
Hardware Integration	The complexity of interfacing the hardware elements	Altered to adjust for lack of visibility in decomposition

The following parameters from the "Reason for Variation" column in Table 4 similarly require additional elaboration:

Coder/Analyst Abilities varies from company to company - Different organizations have different standards their programmers and analysts.

Industry varies from JPL - The Defense Industry are more stringent than JPL and unmanned NASA projects.

Altered to adjust for lack of visibility in decomposition - Lack of visibility into the breakout of CSCIs required additional adjustments.

Varies with GN&C with complexity – Dependent on the type mission (planetary, earth orbiter, lunar, etc.).

Have Different Numbers for Different Projects – Some not Broken out – Some project had more granularity than others based on their financial and engineering requirements.

E. Program Output Mapping into JPL FSW Work Breakdown Structure

The final decision box D_{10} (see Figure 6) alludes to the fact that when a FSW estimate is done, it is sometimes mandated that the costs be mapped as much as possible into the JPL Work Breakdown Structure (WBS). Although this requirement has thus far been in force only for the production of Independent Cost Estimates (ICE's) and Cost Analysis Data Requirements (CADRe's), it is possible that in the future this mapping could be required for proposals as well. Therefore, due to the coupling of the potential importance of realistic FSW WBS element costs with the fast turnaround time often required, the algorithm for and automation of the mapping from SEER-SEM FSW costs to the JPL WBS is an essential component to effective FSW costing activity.

The construction of a mapping from SEER-SEM to the JPL Standard WBS[18] first consists of choosing the format of the cost output in SEER-SEM. The format should be one which groups the output costs in such a way as to facilitate a clear and direct mapping to the JPL WBS. This is important for the abstract understanding of the JPL cost groupings a well as the practicalities of automating the mapping process. To this end, it was deemed that the "Cost by Labor Category" was the SEER-SEM output of choice. This format not only satisfied the above criteria but also served as a basis for cost analysis and cost comparisons by FSW analysts at JPL for many years. Having made this choice, the task is now to map this output into the JPL FSW WBS. The JPL FSW WBS essentially consists of *FSW management*, *FSW systems engineering*, *FSW testbed*, *FSW I&T* and *Coding Related Activities* (which correspond to the following S/C elements: Command & Data Handling (C&DH) , Guidance Navigation & Control (GN&C) , Engineering Models, Payload & Instrument Control SW, SW Systems Services).

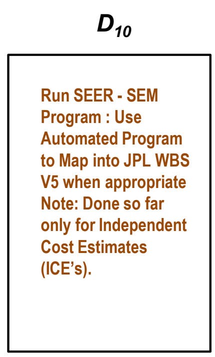
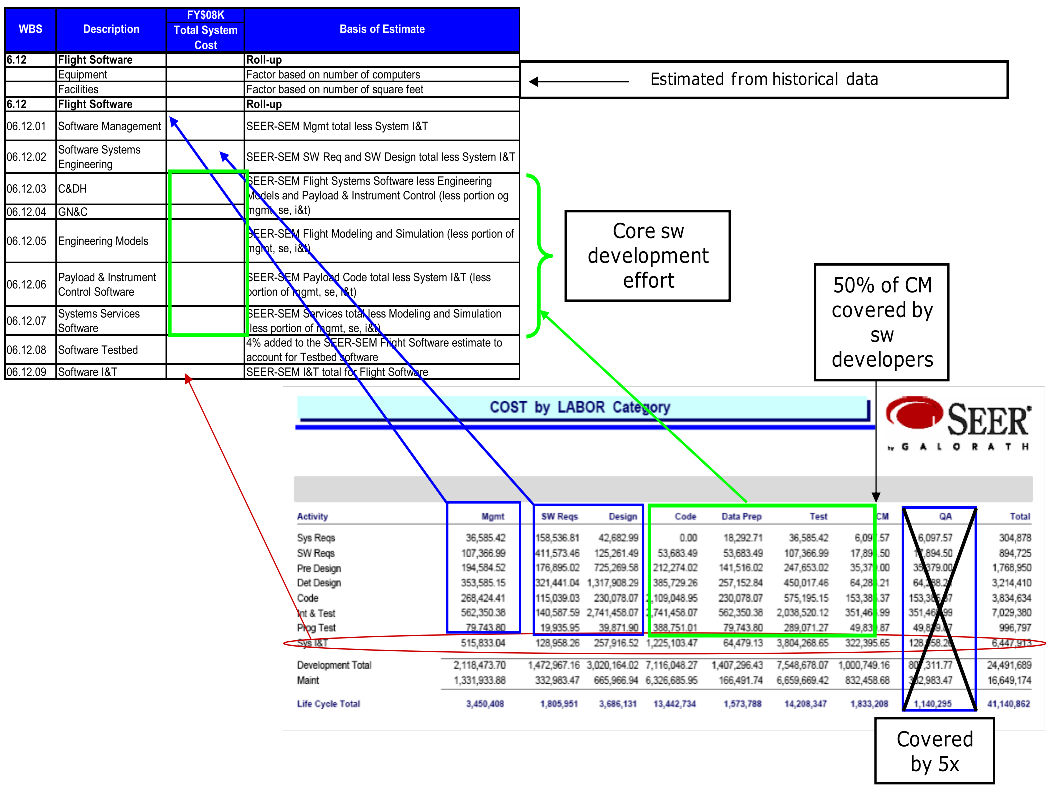


Figure 6: Terminal Decision Box WBS correlation to SEER-SEM Output

For each CSCI for which SLOC is available, the Cost by Labor Category of SEER-SEM produces column costs which can be grouped into all the above WBS elements except for FSW I&T for which it has a row cost and FSW testbed for which a calculation outside of SEER-SEM is done (see below). Note that the SW costs will have to be mapped into merged S/C elements of the JPL WBS if the SLOC values fed into SEER-SEM representing those S/C elements are correspondingly merged. For example, if a separate breakout of S/C GNC SLOC and S/C C&DH SLOC is not available to the FSW analyst, a breakout of costs into the GNC and C&DH JPL WBS elements is not feasible. Therefore, because these costs will be merged into the SEER-SEM input/output, they will be mapped into a merged WBS category consisting of both GNC and C&DH data. Figure 7 represents the SEER-SEM output and mapping to the JPL WBS for the more extreme (and most common case for the Type X proposals) where only one SLOC value is available for the total of all S/C elements.

Figure 8 represents the other extreme, which is more typical of ICE's and CADRe work, where SLOC values are given for each (or at least many) S/C elements. When several arrows with the same color coding from the SEER-SEM outputs merge into one and flow into a JPL WBS element entry (like orange for SW management), that means the cost for that WBS element is formed by taking the sum of the individual costs of the SEER-SEM outputs each of which is enclosed by a box of that same color. Similarly, for I&T, except those costs are enclosed by an ellipse rather than a box for purposes of illustration. Further, note that the SEER-SEM columns which are interpreted as coding related activity (in green) have a one to one mapping directly into the S/C element to which they correspond.

There are some calculations in the illustrated JPL WBS structure for both figures that do not have SEER-SEM as their Basis of Estimate (BOE). The reader will note that equipment and facilities costs appearing at the top of the JPL WBS listings were computed using formulae relating to assumptions on the number of computers used for the coding, average square footage of the coding facilities and composite labor rates. The software testbed cost appearing near the bottom of the WBS listings is computed by taking 4%[19] of the sum of all SEER-SEM costs for all other WBS elements from Project Management through FSW I&T.



WBS	Description	FY$08K Total System Cost	Basis of Estimate
6.12	**Flight Software**		**Roll-up**
	Equipment		Factor based on number of computers
	Facilities		Factor based on number of square feet
6.12	**Flight Software**		**Roll-up**
06.12.01	Software Management		SEER-SEM Mgmt total less System I&T
06.12.02	Software Systems Engineering		SEER-SEM SW Req and SW Design total less System I&T
06.12.03	C&DH		SEER-SEM Flight Systems Software less Engineering Models and Payload & Instrument Control (less portion og mgmt, se, i&t)
06.12.04	GN&C		
06.12.05	Engineering Models		SEER-SEM Flight Modeling and Simulation (less portion of mgmt, se, i&t)
06.12.06	Payload & Instrument Control Software		SEER-SEM Payload Code total less System I&T (less portion of mgmt, se, i&t)
06.12.07	Systems Services Software		SEER-SEM Services total less Modeling and Simulation less portion of mgmt, se, i&t)
06.12.08	Software Testbed		4% added to the SEER-SEM Flight Software estimate to account for Testbed software
06.12.09	Software I&T		SEER-SEM I&T total for Flight Software

Activity	Mgmt	SW Reqs	Design	Code	Data Prep	Test	CM	QA	Total
Sys Reqs	36,585.42	158,536.81	42,682.99	0.00	18,292.71	36,585.42	6,097.57	6,097.57	304,878
SW Reqs	107,366.99	411,573.46	125,261.49	53,683.49	53,683.49	107,366.99	17,894.50	17,894.50	894,725
Pre Design	194,584.52	176,895.02	725,269.58	212,274.02	141,516.02	247,653.02	35,379.00	35,379.00	1,768,950
Det Design	353,585.15	321,441.04	1,317,908.29	385,729.26	257,152.84	450,017.46	64,288.21	64,288.21	3,214,410
Code	268,424.41	115,039.03	230,078.07	2,109,048.95	230,078.07	575,195.15	153,385.37	153,385.37	3,834,634
Int & Test	562,350.38	140,587.59	2,741,458.07	2,741,458.07	562,350.38	2,038,520.12	351,46[illegible].99	351,46[illegible].99	7,029,380
Prog Test	79,743.80	19,935.95	39,871.90	388,751.01	79,743.80	289,071.27	49,839.87	49,839.87	996,797
Sys I&T	515,833.04	128,958.26	257,916.52	1,225,103.47	64,479.13	3,804,268.65	322,395.65	128,958.26	6,447,913
Development Total	2,118,473.70	1,472,967.16	3,020,164.02	7,116,048.27	1,407,296.43	7,548,678.07	1,000,749.16	80[illegible],311.77	24,491,689
Maint	1,331,933.88	332,983.47	665,966.94	6,326,685.95	166,491.74	6,659,669.42	832,458.68	332,983.47	16,649,174
Life Cycle Total	**3,450,408**	**1,805,951**	**3,686,131**	**13,442,734**	**1,573,788**	**14,208,347**	**1,833,208**	**1,140,295**	**41,140,862**

Figure 7: SEER-SEM /JPL WBS Mapping for all SW Elements Combined (Notional Sample Data)

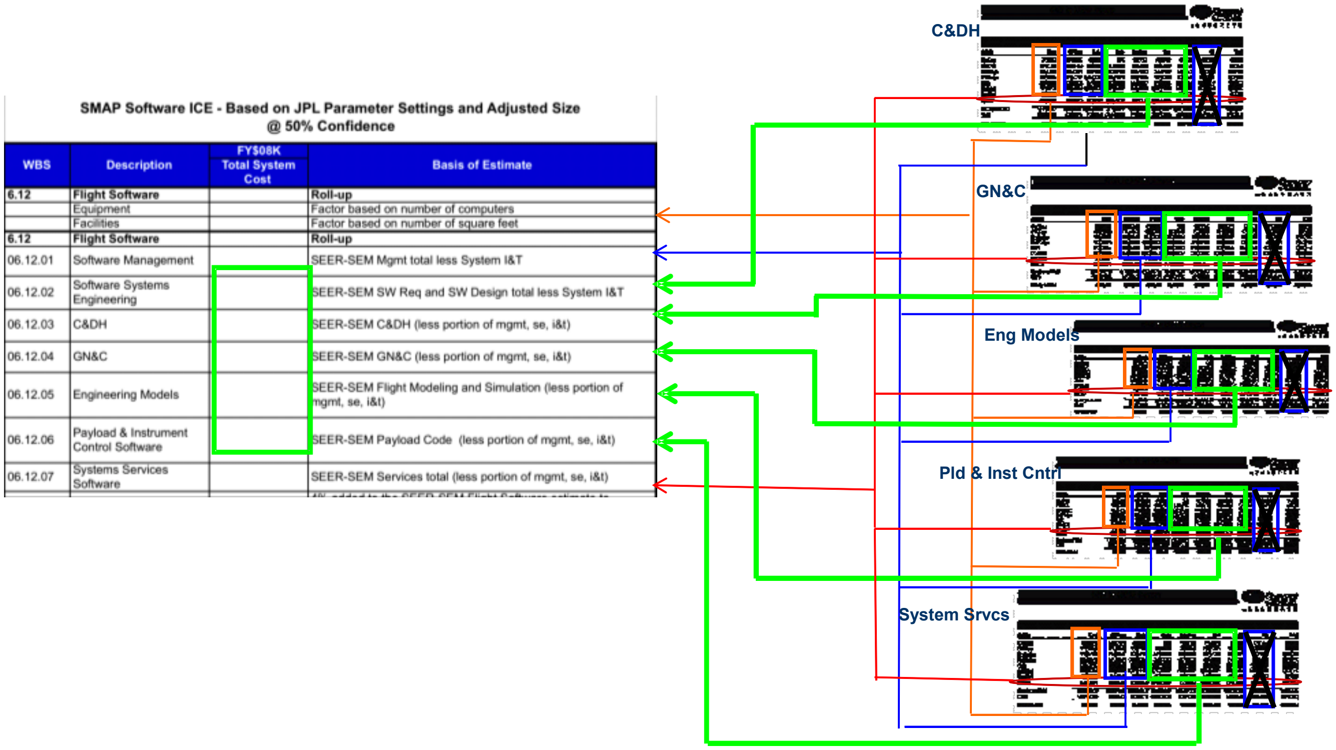


Figure 8: SEER-SEM / JPL WBS Mapping for Individual FSW Elements (Notional Sample Data)

IV. The Speadsheet

As mentioned in the introduction, the comprehensive nature of the spreadsheet yielded a deeper perception regarding the nature and processes of FSW cost estimation (resulting in the conception and creation of a decision graph). For each Type X proposal, the sheet included general mission information together with detailed and significant SEER-SEM input parameter data. Since there were N_0 missions that had to be costed, this yielded a spreadsheet whose size parameters made its complete inclusion in this paper prohibitive. A smaller portion of the sheet containing all of the parameters for only five of the missions sufficiently conveys the sense of expanse and detail implicit in the sheet and is displayed in Figure 9 and Figure 10[20]. Each mission has its own column. The rows display pertinent information for the corresponding column mission. The first 9 rows represent various 'mission facts such as the mission category, name, cost lead etc. Note that certain proprietary data have been blanked out such as the final cost (dollars and work hours) and the proposal name. Note the contractor/analogy data row refers to the flight software contractor and analogy data parameters discussed earlier in the paper. The following groupings (colored in aqua) show the knowledge base inputs, software sizing data (vectors 1 and 2, size Basis of Estimate (BoE)) and parameter settings (non-default constant and varying). All SEER-SEM parameters not shown in the rows are default across the board. Figure 11 is notional and displays a mapping from the main components of the decision graph to the corresponding row parameters that those components determine the values of.

Throughout the cost estimation process, hardcopy of the evolving spreadsheet was made (taped together) in its entirety to reflect the status of cost and cost estimation progress to higher level management. The use of a large paper sheet on a big table with pencils in hand added to the analysis and monitoring process in a way that might not have been achieved otherwise. Further, a better understanding of the nature and justification of the costs was achieved by the cost leads when columns representing only their proposals were distributed to them. Finally, the bird's eye view of the mission data and SEER-SEM inputs/outputs facilitated the cost estimation consistency analysis by the cost estimators. By checking the parameters mission by mission (i.e. column by column) and comparing costs resulting from the use of these parameters together with mission categories and contractor/analogy data, the analysts were allowed insights in a way consistent with the 'one picture is worth 1000 words' philosophy.

Category	Inn_Hel_1			NEO_1	
Proposal Name	1	2	3	4	5
Cost Lead	A	B	C	D	D
Spacecraft Provider	SDC_1	SDC_1	SDC_2	SDC_3	FFRDC
Analogy Program(s) Used	from	from	from	from	from
Contractor/Analogy Data	SDC_1/ SDC_1	SDC_1/ SDC_2	SDC_2/ SDC_2	SDC_3/ FFRDC	FFRDC/ FFRDC
Software Cost Estimates (SEER-SEM) (FY$10M) (excludes testbed, equip, facilities)	$XX	$XX	$XX	$XX	$XX
SEER-SEM (- ATLO, SQA, CM 50%)	$XX	$XX	$XX	$XX	$XX
Team X Estimate (for reconcilliation)	$XX	$XX	$XX	$XX	$XX
Software Duration (SEER-SEM) (mo)	27	30	23	30	26
Knowledge Bases					
SEER-SEM Window Name:					
Platform (Operating Environment)	Unmanned	Unmanned	Unmanned	Unmanned	Unmanned
Application	Flight Systems	Flight Systems	Flight Systems	Flight Systems	Flight Systems
Acquisition Method	New/Reuse	New/Reuse	New/Reuse	New/Reuse	New/Reuse
Development Method	Incremental	Incremental	Incremental	Incremental	Incremental
Development Standard	DO-178B Level B	DO-178B Level B	DO-178B Level B	DO-178B Level B	DO-178B Level B
Software Size (SLOC)					
Size BoE	Used actual SLOC counts from SDC_1. Assumed 25% new, 25% reused "as is", and 50% reused modified.	Used an average actuals from FFRDC projects with the inheritance percentages fromFFRDC.	Used SDC_2-derived SLOC values for new, reused, reused modified. Added correction factor to convert code counts.	Used FFRDC TDP information.	Used FFRDC size estimates. Duplicated reasoning used for FFRDC estimate.
ESLOC	69,888	92,238	61,848	85,533	61,450
Delivered Software (SLOC) - most likely	153,812	202,000	204,990	221,664	180,000
Software Size (SLOC)					
New SLOC - most likely	38,453	60,600	25,000	46,404	30,000
% of new SLOC	**25%**	**30%**	**12%**	**21%**	**17%**
Reuse SLOC (as is - no mod) - most likely	38,453	35,350	97,700	117,424	70,000
% of reused (as is) SLOC	**25%**	**17%**	**48%**	**53%**	**39%**
% re-design	0	0	0	0	0
% re-implementation (Re-coding)	0	0	0	0	0
% re-test	50%	50%	50%	50%	50%
Reuse SLOC (modified) - most likely	76,906	106,050	82,290	57,836	80,000
% of reused (modified) SLOC	**50%**	**53%**	**40%**	**26%**	**44%**
% re-design	10%, 25%, 25%	10%	10%	10%, 25%, 25%	10%
% re-implementation (Re-coding)	10%, 25%, 25%	10%	10%	10%, 25%, 25%	10%
% re-test	50%	50%	50%	50%	50%

Figure 9: Portion of Final Spreadsheet (a)

Parmeter Settings Notes					
Personnel Capabilities & Experience* (7 parameters)	In general, these values were defaults, but exceptions were made for certain projects with personnel having more mission-related experience.				
Analyst Capability					NOM-
Analyst's Application Experience					NOM
Programmer Capabilities					NOM-
Programmer's Language Experience					VHI
Developkent System Experience					HIGH
Target System Experience					VHI
Practices & Methods Experience					VHI
Development Support Environment*	Leave at KB settings with the exception of:				
turnaround time	VLO	VLO	VLO	VLO	VLO
response time	LOW	LOW	LOW	LOW	LOW
Product Development Requirements*	Leave at KB settings with the exception of:				
requirements volatility	HIGH	HIGH	HIGH	HIGH	HIGH
spec level - Reliability	HIGH-	HIGH-	HIGH-	HIGH-	HIGH-
quality assurance level	HIGH	HIGH	HIGH	HIGH	HIGH
rehost (development to target)	HIGH-	HIGH-	HIGH-	HIGH-	HIGH-
Product Reusability Requirements*	Should always be NOM (no reusability required by the contract). If the parameter is set to NOM the				
Development Environment Complexity*	Leave at KB settings				
Target Environment*	Leave at KB settings with the exception of:				
memory constraint	NOM	NOM	NOM	NOM	NOM
timing constraint	NOM+,NOM+,HIGH-	NOM+,NOM+,HIGH-	NOM+,NOM+,HIGH-	NOM+,NOM+,HIGH-	NOM+,NOM+,HIGH-
real time code	NOM, NOM, NOM+	NOM, NOM, NOM+	NOM, NOM, NOM+	NOM, NOM, NOM+	NOM, NOM, NOM+
security requirements	NOM	NOM	NOM	NOM	NOM
Schedule & Staffing Constraints*	Leave at KB settings with the exception of:				
start date	11/25/2012	11/25/2012	11/25/2012	11/25/2012	11/25/2012
Min Time vs Optimal Effort	Always start with **Optimal Effort**. Where possible, verify that the schedule duration is achievable. If not, evaluate schedule constraints to accommodate the estimated schedule. If the software development time is less than the **Minimal Time**, the SEER-SEM model contends that it is not possible to complete the software. Identify this as a significant risk issue!				
Confidence Levels*	Both effort and schedule should be run at 50% and 70% confidence. SQI recommends the 70%				
Requirements*	Leave at KB settings				
System Integration*	Leave at KB settings with the exception of:				
number of programs being integrated	5	5	7	5	5
hardware integration	N-, N, N+	N-, N, N+	N-, N, N+	N-, N, N+	N-, N, N+
Ecomonic Factors*	Labor rate based on NASA Center contractor developed software survery conducted in FY08. Escalated				
cost base year	2010	2010	2010	2010	2010
labor rate (FY$2010) work months	$xx	$xx	$xx	$xx	$xx

*Blue highlighted font corresponds to SEER-SEM parameter heading titles.

Figure 10: Portion of Final Spreadsheet (b)

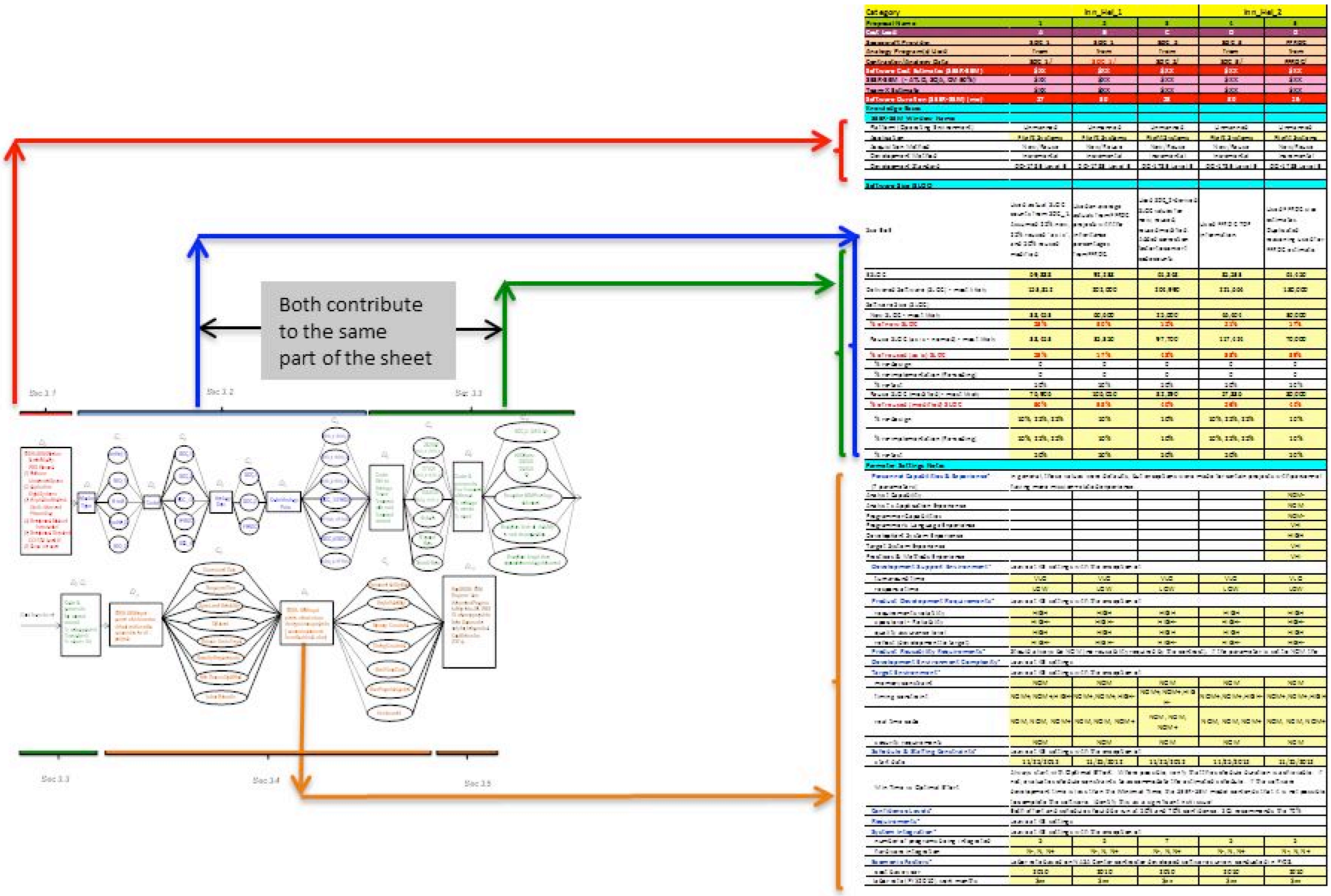


Figure 11: A Notional Retrospective of Decision Graph and Spreadsheet

V. Computer Generation of Graphics with Explanations

This section demonstrates not only how the rule based approach can produce useful tables and charts reflecting the results of FSW coat analysis but also how such a rule base can aid in the explanation of the output reflected in these charts. More precisely, it is demonstrated that the data used in the expert justification of analysis results for N_0 Type X proposals is contained in the rules which would comprise the system conceived of in this paper.

Upon completion of the FSW cost estimating activities for the N_0 Type X proposals, a presentation was addressed to section personnel that summarized the relevant conclusions of the analyses. Included in the discussion was a trend chart, a justification of its graph and relevant high level tabular data.

Figure 12 shows the trend chart giving the Costs c_i (FY$10) vs Mission Category. This graph gave an immediate high level view of the relative estimated costs which would be incurred with respect to the types of missions described earlier in Table 1.

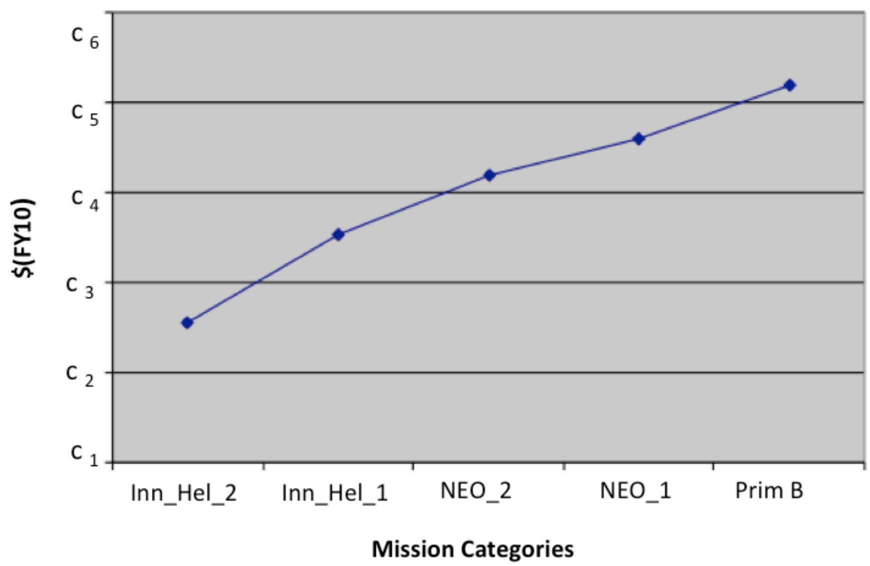


Figure 12: Cost vs Mission Category Trend Line

The mathematical evidence of the chart should properly be accompanied by a set of key factors influencing the relative nature of its ordinate values. To this end, a listing of distinguishing influences was given and is shown in Figure 13. As can be seen in the text of the Figure, a justification of lower to higher costs is given in terms of production line maturity, % of new code size, general code size, and general increased coding costs due to SW developer /analogy data pairings.

It is of interest to note that these relevant factors were proffered *before* the idea of a decision graph was developed. They were based solely upon the insights of and discussions between the cost estimators engaged in the FSW cost analysis for the proposals. The listing of the factors was not based upon any automated decision making system. Upon examining these earlier efforts, however, it is clear that all of these listed factors are included in the decision graph presented at the beginning of the paper. Hence these factors would be included in any rule base arising from the decision graph and could therefore be included in any explanation of the obtained costs arising from the execution of said rules[21] (also, see Section 6, Summary and Future Work). The modular nature of these rules facilitates the production of computer generation of these explanations as well.

It should be noted that the listing of factors influencing the nature of the trend line would serve as a first step in the justification of the trend line. With continued use of the system and/or development of its rule base, deeper explanations could be extracted.

- ***Inn_Hel_2*** Missions
  - FFFRDC/FFRDC or SDC_2/SDC_2
  - Both organizations experienced in *Inn_Hel_2* FSW development
  - SDC_2 product line
- ***Inn_Hel_1*** Missions
  - SDC_1/SDC_1 , SDC_1/SDC_2, SDC_2/SDC_2
  - Less SDC_2 product Line influence (mostly SDC_1)
- ***NEO_2*** Missions
  - SDC_1/SDC_1 , (TBD)SDC_1/SDC_1
  - Higher cost based on high new code % due to TBD for SDC_1
- ***NEO_1*** Missions
  - FFRDC/FFRDC, (TBD) FFRDC/FFRDC
  - Costs based on inherited FFRDC code
  - Higher costs due to large amount of FFRDC code
- ***Prim B*** Missions
  - TBD/SDC_1 , SDC_4/SDC_1 , SDC_1/SDC_1
  - Higher costs due to TBD/SDC_1 and SDC_4/SDC_1 pairings for 2 Prim B Missions.

Figure 13: High Level Factors influencing FSW Cost Estimates

It is also of interest to see the information as it is presented in Table 5. These data are embedded in the fabric of the decision graph nodes and thusly are incorporated in the execution of any rules derived from it. Hence an extraction of this data as needed from the rules is possible (in a more straightforward fashion than as above) and, as before, explanations/justifications for values are facilitated by the rule-based modular nature as well.

Table 5: SLOC Data Corresponding to Mission Categories*

	Inn_Hel_2	Inn_Hel_1	NEO_2	NEO_1	Prim B
Average ESLOC	#	#	#	#	#
ESLOC Range	# - #	# - #	# - #	# - #	# - #
# of Missions	#	#	#	#	#

***'#' represents a numerical value and '# - #' correspondingly designates a numerical range.**

The crucial point of this section is that the listing of factors relevant to trend determination produced by the experts (before the conception of the decision graph) is contained in, and can be easily extracted from, the decision graph and therefore from rules arising from its constitution.

VI. Brief Description of the SCHERRI Program

The program was built using CLIPS. CLIPS is a forward-chaining rule-based programming language written in C which provides a complete environment for the construction of rule and/or object based expert systems[22]. At present, there are approximately 145 rules.

S C H E R R I starts by querying the user as to the nature of the mission for which the cost analysis is to be performed. See Figure 14. The category of primitive bodies includes asteroids and comets. The Astrophysics category consists of earth orbiting spacecraft.

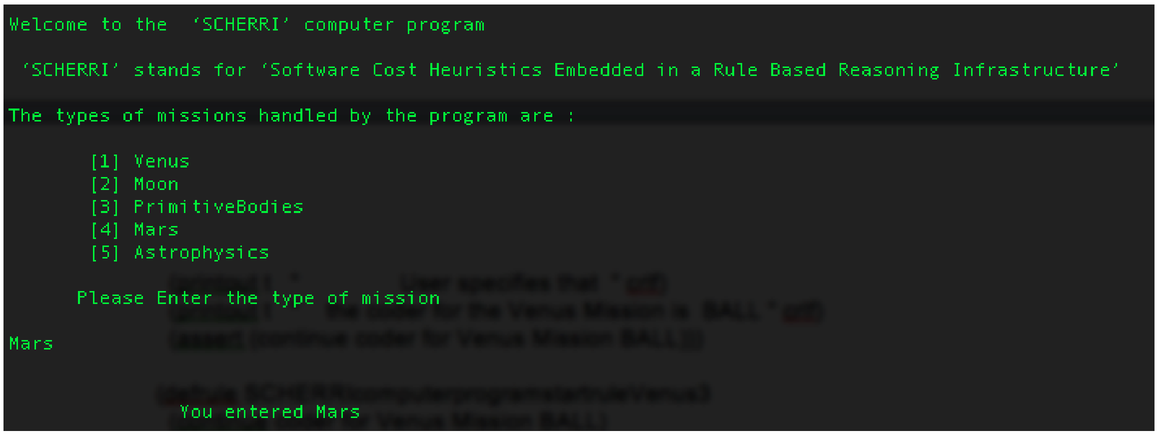


Figure 14: Initial Query from S C H E R R I

The program then uses various qualitative inputs from the user as described in the paper to determine the BOE for the analysis. An example of the BOE printouts is shown in Figure 15. Note that at he bottom the figure the program asks the user to enter the amount of SLOC being used in the analysis as described earlier in the paper.

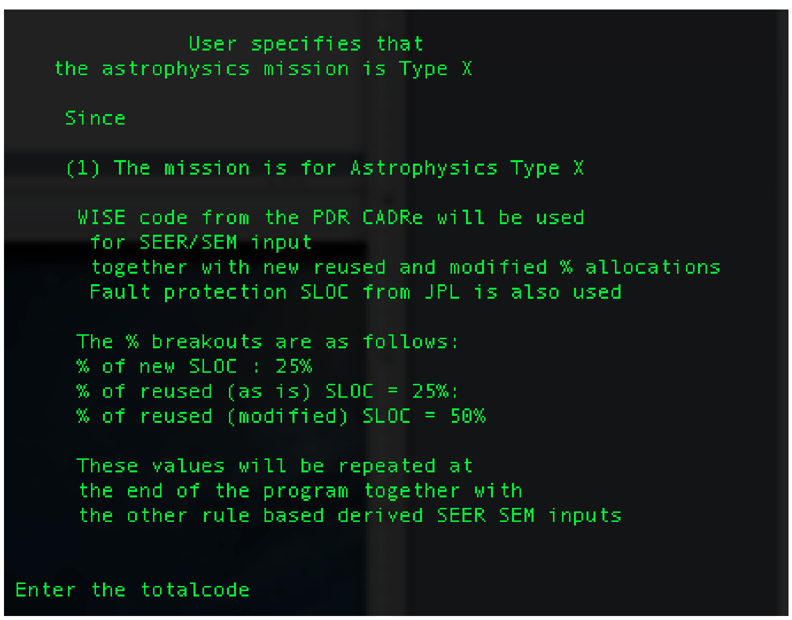


Figure 15: BOE for an Astrophysics Mission

Figure 16 shows the outputs computed by the program based upon the user inputs and the expert rules built into the system.

```
Enter the totalcode

1000

Total Delivered Code =  1000
     % of new SLOC =    250.0
     % of reused (as is) SLOC =    250.0
     % of reused (modified) SLOC =  500.0


The reused mod SLOC has the following input %s :
    the re-design SLOC %s are: 10 %, 25%, 25%
    the re-code SLOC %s are: 10 %, 25%, 25%
    the re-test SLOC % is : 50%


The reused as is SLOC has the following input %s :
    the fraction of the SLOC that is re-designed =  0 %
    the fraction of the SLOC that is re-coded =  0 %
    the fraction of the SLOC that is re-tested =  50 %


DEVELOPMENT SUPPORT ENVIRONMENT parameters :

    turnaround time is VLO
    response time is LOW

PRODUCT DEVELOPMENT REQUIREMENTS parameters :

    requirements volatility is HIGH
    spec level - Reliability is HIGH-
    test level is HIGH
    quality assurance level is HIGH
    rehost (development to target) is HIGH-

PRODUCT REUSABILITY REQUIREMENTS parameter :
reusability level required is  NOM
user note: the nom value for this parameter renders the
  percentage input in this section meaningless

DEVELOPMENT ENVIRONMENT COMPLEXITY parameter :
process improvement is NOM

TARGET ENVIRONMENT parameters :
    memory constraint is NOM
    timing constraint is NOM
    real time constraint is NOM  NOM   NOM+
    security requirements is NOM

SCHEDULE AND STAFFING CONSTRAINTS parameters :
start date is user's choice
min time vs optimal effort is optimal effort

CONFIDENCE LEVELS parameters :
The following 2 parameter values are recommended by JPL SQI
effort probability is 70%
schedule probability is 70%

REQUIREMENTS parameter :
requirements after baseline is YES

SYSTEM INTEGRATION  parameters :
    number of programs being integrated  is 5
    concurrency of I&T is  is HI
    hardware integration is is NOM-  NOM   NOM+

ECONOMIC FACTORS  parameters :
    cost base year is user's choice is present year
    labor rate is  is $26,281
```

Figure 16: Output Provided by the Rule-Based Structures

Figure 17 shows some of the code use in the construction of the program. The rules shown represent the request for SLOC data and the corresponding computations done on this data relating to determination of % of new code, % of reused "as is" code and % of reused "modified" code computations based upon the user input.

```
(defrule SCHERRIslocccomputationruleVenus8
   (comp rule seq Venus BALL BALL 1)
 =>
  (printout t   "     "        crlf)
  (printout t   " Enter the totalcode   " crlf )
  (printout t   "     "        crlf)
  (assert (TotalLogSloc (read)))
)

(defrule SCHERRIslocccomputationruleVenus9
  (comp rule seq Venus BALL BALL 1)
  (TotalLogSloc ?x)
 =>
  (printout t   "     "        crlf)
  (printout t   " Total Delivered Code =  " ?x crlf )
  (printout t   "     % of new SLOC =    "  (* .25 ?x) crlf)
  (printout t   "     % of reused (as is) SLOC =    "  (* .25 ?x) crlf)
  (printout t   "     % of reused (modified) SLOC =  "   (* .50 ?x) crlf)
  (printout t   "     "        crlf)
  (assert(trigger reused mod sloc)))
```

Figure 17: Two Rules of the Program

VII. Summary and Future work

The ideas presented in this report are best encapsulated by the tried and true sentiment "Necessity is the mother of invention". During the process of producing timely and accurate flight software cost analyses, it became clear that efficient and effective methodologies for consistency checking had to be established. Intuitively, the best way to accommodate this requirement was to develop a comprehensive spreadsheet that showed, for each mission, all parameters which served as the blueprint for that mission's FSW cost estimate. At first this only included certain parameters which were inputs to the SEER-SEM FSW costing model. It was subsequently determined that it was equally important to include mission information that gave rise to parameter value selection. The consideration by the expert of the interplay of data /reason for data forced upon both authors a progressively exacting analysis not only regarding the FSW costing exercise but also with respect to an examination of the thinking processes behind the execution of such an exercise. This resulted in the invention of the decision graph.

Logically, the next step would be to start expansion and refinement of the rule-based system[23] described in the previous section. Our approach there was to use the decision graph presented in this paper for the N_0 Type X proposals and build a decision tree using the descision graph as a high-level road map (see below). This same approach could be used to enhance the current system, further allowing future analysts to obtain the costs of interest and have a complete explanation for how the cost was obtained based upon the triggered rules[24],[25],[26],[27].

An example of how a decision tree corresponds to the decision graph can be seen in the following example. Figure 18 shows an initial part of the decision graph. As can be seen, the first column of nodes contains all the mission types that appear on the spreadsheet while the second column lists all the FSW contractors. No correlation between the two is given. Figure 19 shows the actual pairings of Mission Type/Coder that actually appeared in the spreadsheet. This serves as direction for the design of the expert system. One way in which this can be seen is that in a constructed system, after the user enters the Mission Type based upon the possibilities of column 1 nodes, the computer could then query the user to select from the limited range of FSW contractor possibilities based upon that initial input.

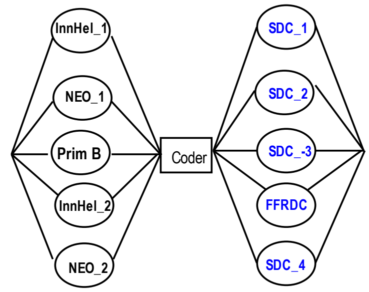


Figure 18: Portion of Decision Graph

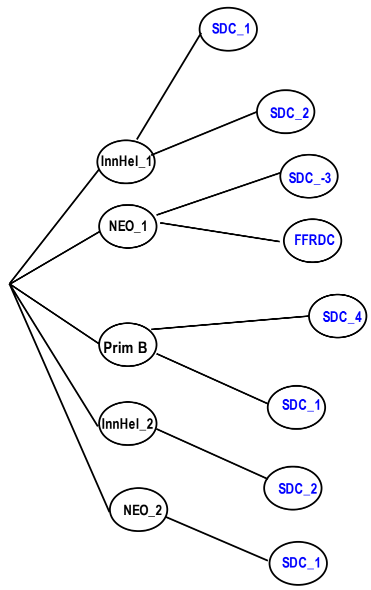


Figure 19: Corresponding Decision Tree

So, for example, if the user chose that he was interested in a primitive body mission, the computer could then say:

"In the Type X Mission Proposal Data Base for Year 2010, the two possible FSW contractors for Primitive Body Missions were SDC_4 and SDC_1. SDC_4 worked on a comet mission to examine its core and SDC_1 worked on a main belt comet and an asteroid mission. Please select which of these missions (as listed below) you would like to see a derivation of FSW cost for."

Expansion of the rest of the decision graph into a tree structure would allow a complete step by step explanation and justification of the how's and why's of the FSW cost estimate for the mission of interest. Details of precisely what the format, nature and expanse of the explanations at each step are one of the subjects for future research even within the strict confines of the Type X mission FSW costing effort.

Previous efforts have been made to integrate neural networks, expert knowledge, and rule-based systems for use with algorithms and data sets of conventional software cost estimating models.[28],[29] The continued refinement of the work presented in this paper would serve as a foundation for the efficacious use of these techniques to further enhance the accuracy and descriptive content of the FSW cost estimates of interest.

The graph shown in Figure 12 was a first attempt to show a high level quantitative perspective that could serve as a quick reference for spotting trends in FSW cost estimation. One area of further work could be to compile more and more estimated costs and create corresponding graphs. Corresponding could also be made for actuals. These graphs could serve as an aid in determining and graphing a 'difference metric' between actual and estimated FSW costs. Analyses could be done to improve the estimation based on the comparisons. Further estimate data with rule-based generated explanations and graphs could be compiled to see if the difference metric is improving. A cycle of estimation-cost metric determination-analysis-improvement could occur through time as the cost estimation data base increases.

Finally, as previously discussed, the mapping scheme which takes the SEER-SEM output into the JPL FSW WBS is semi-automated. Complete automation of the mapping for fast delivery of precision FSW cost estimates would be a natural follow up to the efforts thus far. Integrating this system into an expert system as discussed above would, with proper interaction and feedback from appropriate personnel, yield a powerful interactive costing tool within the JPL community and beyond.

References

[1] The Fifth Generation : Artificial Intelligence and Japan's Computer Challenge to the World, Edward A Feigenbaum, and Pamela McCorduck, Addison-Wesley, Reading, MA, 1983.

[2] LA Times Article, "Thirty Year Veteran of JPL Chosen as its Leader", Lee McFarling: Times Science writer, February 1, 2001, reference link:
http://articles.latimes.com/2011/Feb/01/local/mc-19734.

[3] CNN.com/technology, "Mar Science Lab Delayed Two Years", Alan Silverleib: CNN, December 4, 2008, reference link:
http://edition.cnn.com/2008/TECH/Space/12/04/NASA.Mars.Delay/availableindex.html.

[4] Nature: A International Journal of Science Published on Line, "Mars Rovers Debut Delayed", Eric Hand: Nature, reference link:
http://dx.doi.org/10.1038/news.2008.1283.

[5] "Software Cost Risk Estimation and Management at the Jet Propulsion Laboratory", Jairus Hihn and Karen Lum, 17th International Forum on COCOMO and Software Cost Modeling, October 22-25, 2002.

[6] "2008 NASA Cost Estimating Handbook" (2008), reference link:
http://www.nasa.gov/pdf/263676main_2008-NASA-Cost-Handbook-FINAL_v6.pdf.

[7] Thinking Allowed: DVD, Jeffrey Mishlove, host. Edward Feigenbaum & Penny Nii: Expert Systems. reference link:
http://www.youtube.com/watch?v=Uk9YA1kwZLw.

[8] Decision Analysis, Howard Raiffa, Addison-Wesley, Reading, MA, 1968.

[9] The Concise Oxford Dictionary of Mathematics, Christopher Clapham and James Nicholson, Oxford University Press, 2005: Pages 27, 189 and 290.

[10] A default knowledge base is a selected database in the SEER-SEM program which provides values for input parameters when such parameters are not provided by the user.

[11] CSCI is a standard term from DoD-STD-2167 representing a managed and configured block of code.

[12] The same letter "X" is used throughout the paper to indicate an unknown quantity. Unless implied by the text, there is no correlation between the different uses of this variable.

[13] Note that the number of coder/analogy data pairs shown (7) indicates a subset of all combinations of coder and analogy data possibilities (15).

[14] In SEER-SEM, each value for a '% type' parameter can be given a range of three values corresponding to least likely, most likely and highly likely. The fact that previous to this discussion, only one % value was used means that that one value was given to all three possibilities.

[15] %retest does not require a distribution as do the others (see subsection 3.3.3 for details).

[16] The NSA "Orange Book" is titled "A Guide to Understanding Discretionary Access Control In Trusted Systems," and is issued by the National Computer Security Center (NCSC) under the authority of and in accordance with Department of Defense (DoD) Directive 5215.1, "Computer Security Evaluation Center." The guidelines defined in this document are intended to be used by computer hardware and software designers who are building systems with the intent of meeting the requirements of the Department of Defense Trusted Computer System Evaluation Criteria, DoD 5200.28-STD.

[17] "CxL Software Composite Labor Rate Justification", Sherry Stukes and Bob Estes, February 2008.

[18] JPL Standard WBS v4.0, Flight Software elements 06.12.01 – 06.12.09.

[19] The 4% factor is based on discussions and analysis with Elihu McMahon, Deputy Section Manager, Flight System Avionics, with over 20 years of direct experience working on JPL software testbeds.

[20] The contents of Figure 10 are juxtaposed directly beneath the contents of Figure 9 in the actual spreadsheet.

[21]Software Cost Estimation Using a Decision Graph Process: A Knowledge Engineering Approach, S. Stukes, J. Spagnuolo Jr, 2011 ISPA/SCEA Joint Annual Conference, June 7-10, 2011.

[22]Rule-based systems, Frederick Hayes Roth, Communications of the ACM, Volume 28, Issue 9, Sept. 1985.

[23]Logical Foundations for Rule-Based Systems (Studies in Computational Intelleigence) 2nd Edition, Antoni Ligeza, Springer-Verlag Berlin Heidelberg, 2006, pages 91-97.

[24]Lindsay, Robert K., et al (1980). "Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project". ISBN: 0-07-037895-9.

[25]Shortliffe, E. H. (1976). "Computer-Based Medical Consultations, Mycin". ISBN: 0444001794.

[26]Hart, P., & Duda, R. (1977). "Prospector: A Computer-Based Consultation System for Mineral Exploration". Volume 155 of Technical Note. Artificial Intelligence Center. Stanford Research Institute: Artificial Intelligence Group.

[27]Ed Fegenbaum's Search for A.I. , "Feigenfest 70th" at Stanford University 2006, iAM Digital Archival Media, 2006. Internet reference:

http://www.youtube.com/watch?v=B9zVdU3N7DY.

[28]Wei, L. D., et al. (2010). "Improving Software Estimation Using A Neuro-Fuzzy Model with SEER-SEM".Global Journal of Computer Science and Technology. Volume 10. Issue 12 (V1.0). Page 52-64.

[29]Khan, I.R. et al, (2010), Efficient Software Cost Estimation using Neuro-Fuzzy Technique, Proceedings of the ISCET 2010, pages 59-62.

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA.