MCR Technology Inc.
55 Depot Road
Goleta, Ca. 93117

Final Report for
An Expert System Based Software Sizing Tool, Phase II
NASA Contract NAS7-1033

JPL
IN-61-CR
290747
340P.

Project Summary

The purpose of this research was to develop a software tool for predicting the size of a future computer program at an early stage in its development. The system is intended to enable a user who is not expert in Software Engineering to estimate software size in lines of source code with an accuracy similar to that of an expert, based on the program's functional specifications. The project was planned as a knowledge based system with a field prototype as the goal of Phase II and a commercial system planned for Phase III.

The researchers used techniques from Artificial Intelligence and knowledge from human experts and existing software from NASA's COSMIC database. They devised a classification scheme for the software specifications, and a small set of generic software components that represent complexity and apply to large classes of programs. The specifications are converted to generic components by a set of rules and the generic components are input to a nonlinear sizing function which makes the final prediction.

The system developed for this project predicted code sizes from the database with a bias factor of 1.06 and a fluctuation factor of 1.77, an accuracy similar to that of human experts but without their significant optimistic bias.

The code size of a computer program is the most important factor in determining its budget and schedule. Current methods suffer from a high optimistic bias and/or can only be applied late in the design stage of software development. The Code Sizing Tool will allow nonexpert users to make unbiased estimates at the proposal stage and will therefore reduce the costs and risks of developing software intensive systems. This will make the system commercially viable and of benefit to NASA, other government agencies, and private industry.

(NASA-CR-186706) AN EXPERT SYSTEM BASED SOFTWARE SIZING TOOL, PHASE 2 Final Report (MCR Technology) 340 p CSCL 09B

N90-26525

Unclas
G3/61 0290747

Table of Contents

List of Figures

List of Tables

1.0 Introduction

This document is the final report for NASA contract NAS7-1033 for *An Expert System Based Software Sizing Tool*. The project was funded through a Phase II SBIR (Small Business Innovation Research) grant and resulted in a system that can predict the size of computer programs, in lines of code, based on the functional specifications of the program. The Code Sizing Tool is an interactive computer program with the capability of storing and retrieving partial or complete specifications from files. The user does not have to be an expert in Software Engineering but does have to have enough expertise in the application to provide its functional specifications to the system. The environment used to develop the Code Sizing Tool is also documented in this report and included with the project delivery. The Code Sizing Tool is sometimes referred to as the Production System and the development environment as the Knowledge Acquisition System.

This section contains background material on the code sizing problem, results obtained from testing the system, and future recommendations. The rest of the report contains a description of the work carried out and the resulting system. Section 2 contains a description of the general approach including the early activities in the knowledge based system development: problem definition (requirements analysis), knowledge source and tool acquisition, and conceptual design. Section 3 is a project management history including the task definitions, schedule, and budgeted expenditures. Section 4 is a technical description of the system including a detailed analysis of the formulas and algorithms in each of its major components. Section 5 is an operational description that contains specific instructions for using the system and formal descriptions of its important data structures. This section is designed to serve as a user's manual. Section 6 is a systems description that contains an analysis of the way in which the various components of the system fit together and interact with each other. This section (along with Section 5 and the Appendices) is designed to serve as a programmer's manual and provide documentation for the maintenance of the system. Section 7 is a bibliography and is followed by a detailed set of Appendices that contain the source code, rule base, program database, detailed test results, and summaries of the interviews with outside experts. Finally, the last page of the report is a Report Documentation Page (NASA Form 1626).

1.1 Background

A long standing problem in Software Engineering is how to determine the resources necessary to develop a proposed software system. The lack of a satisfactory solution often results in software projects being notoriously difficult to budget and schedule. Various software cost estimation methodologies and tools were created in the 1970's (such as COCOMO, SOFTCOST, etc.) in an attempt to address these recurring problems. The models use several parameters or "cost drivers" to estimate the required number of man-hours and the time schedules. The most important input to these models is the number of delivered source lines of code (SLOC). Research has shown a strong correlation between the SLOC and the eventual cost. Code sizing has long been known as a difficult and important problem in Software Engineering. Some progress has been mode in estimating the code sizing based on various aspects of the detailed design of the program [1,2]. However, the accuracy of these methods depends on a significant amount of design work having already been done on the project. They must be applied by people with considerable expertise in Software Engineering and the application area of the software. The purpose of this research was to develop a software tool to enable non-expert personnel to estimate software sizes during the early stages of a project when only the functional specifications are known.

The software sizing problem has been compared to the automatic programming problem [6] in that both require detailed knowledge about the program being handled.

The sizing problem is less difficult and can be considered as a subset of automatic programming in the sense that automatic production of the code would determine its size. Since this in not currently feasible, other methods are employed. There are presently two types, both of which rely on expert estimators using various aspects of the detailed design of the software.

Program sizes are often estimated by breaking the system into as many subroutines as possible and using the experience and knowledge of the Software Engineer to directly estimate the size of each module. This has the advantage that over estimates of some modules can partially cancel under estimates of others. It also reminds the estimator to consider as many aspects of the project as possible. The PERT sizing method is a refinement of this technique in which the Software Engineer supplies a highest, lowest, and most likely size estimate for each routine. Expected values and standard deviations are derived from these estimates based on a beta function [6]:

$$M^* = \frac{M_{min} + 4\, M_O + M_{max}}{6} \tag{1.1}$$

$$S = \frac{M_{max} - M_{min}}{5.29} \tag{1.2}$$

where M_{min}, M_O, and M_{max} are the minimum, mean, and maximum estimates, M^* is the most likely estimate and S is the standard deviation of the code size estimate.

The other general method for sizing code is to estimate it in terms of a complexity metric. Early work in this area derived measures of a program's complexity from the source code. Halstead [8] used the number of tokens in a program as a measure of its complexity. He derived an equation for the number of tokens, N, in terms of the number of operators, n_1, and the number of operands, n_2:

$$N = n_1 \log_2 n_1 + n_2 \log_2 n_2 \tag{1.3}$$

This approach cannot be used directly to estimate code size since the measurements are based on the source code itself. Extensions of it, which derive complexity metrics based on the program's design, can be used for size estimation. Albrecht [1,2] developed the "function point" metric in terms of the top level input and output items:

$$F = P_c\, S_i\, W_i\, F_i \tag{1.4}$$

In the above equation, F_i represents the number of items in each of five categories, and W_i represents a weight given to each category depending on the complexity of the data item within the category. The weighting factors for each category are shown in Table 1.1. P_c is an overall complexity adjustment factor that can range from 0.35 to 1.35 depending on the presence or absence of fourteen influence factors such as whether the system is designed for for a high transaction rate, whether it is designed to be used at multiple sites, etc. The code size can then be determined in terms of the function point metric as follows [15]:

$$L = 0.20\, c_f\, F^{0.89} \tag{1.5}$$

where L is the size in thousands of lines of code, F is the function point metric, and c_f is an implementation factor (1.0 for COBOL and 0.72 for PL/I, for example). There are a number of other software metrics in use [5,9] which generally correlate well with both code size and cost. Studies have shown [4] that there is no single metric that works best in all cases.

Table 1.1
Weighting Factors for the Function Point Method

	Simple	Average	Complex
External Inputs	3	4	6
External Outputs	4	5	7
Logical Internal Files	7	10	15
External Interface Files	5	7	10
External Inquiries	3	4	6

The present methods have the following disadvantages:

1. Optimistic Bias
2. High variance
3. The need for expert estimators
4. The are based on details of the program design

Software size, cost, and schedule estimates are subject to the "fantasy Factor". Projections tend to be underestimated resulting in serious cost and time overruns. There are at least two reasons for this systematic bias:

1. People are basically optimistic and have a desire to please management. They are often under pressure to produce estimates that fit in with cost and time constraints.

2. People often fail to consider all aspects of the project. They often concentrate on the most visible parts of the software that deal directly with the application and ignore the parts that do housekeeping, I/O, error handling, etc. These secondary aspects of the project often create the majority of the code.

Estimates for both software size and cost tend to have high biases and variances. This is due in part to the large number of factors which influence these estimates and the subjective nature of the present procedures. Estimates vary considerably with the skill and experience of the estimators. Data comparing code size estimates with the final size of the software are scarce. One study [14] of code for the Deep Space Network of the Jet Propulsion Laboratory showed a bias factor of 25% and a fluctuation factor of 16%:

$$L_{act} = 1.25\, L_{est} \qquad (*/\ 1.16) \qquad (1.6)$$

where L_{est} is the estimated code size and L_{act} is the actual code size. Estimates from the Jet Propulsion Laboratory would tend to be better than average because of their extensive experience in software development and Software Engineering research. The tendency to be consistently optimistic and for the bias to be greater than the fluctuation is typical for software estimates. Since code size is the most important single component in determining the cost and schedule of software development projects [6], it is expected that the uncertainties and biases in cost and scheduling would be similar to that in code sizing. A study of 100 schedule estimates for the Department of Defense [3] also shows a consistent underestimation:

$$T_{act} = 1.6\, T_{est} \qquad (*/\ 1.25) \qquad (1.7)$$

where is T_{est} is the estimated time to completion, and T_{act} is the actual time to completion. A number of software tools have been developed for cost and schedule estimation. A recent study [11] was conducted in which experts in the use of four popular tools were given identical sets of detailed specifications of a software project. These specifications included the code size, language, development environment, and operation environment. The results for level of effort are shown below:

Table 1.2
Level of Effort Predictions from
Commercial Software Engineering Tools

Estimation Tool	Level of Effort (person months)
JS-2	940
SLIM	200
GECOMO	363
ESTIMACS	107

These estimates vary over almost an order of magnitude and have a standard deviation of 93%. The more that is known about a software project, the greater the accuracy in prediction costs. Boehm [6] has given the following error factors as a function of software development stage:

Table 1.3
Size Prediction Error vs Software Development Stage

Stage	Error Factor
Concept feasibility study	4
Software budget proposal	2
Software task budget negotiation	1.5
Software detailed cost estimate	1.2
Implementation cost-to-complete analysis	1.1
System Delivery	1

The goal of the system is to provide a tool for estimating software size, in lines of source code, over a significant domain of practical software systems with an accuracy comparable to that of human experts. The system also eliminates the optimistic bias that occurs in present expert estimates and allow users without expertise in Software Engineering to perform the estimation task. The system is implemented as an interactive computer program that sizes code based on its functional specifications. The user inputs the specifications and sees the estimated size divides among generalized program components. Partial or complete specifications can be stored and retrieved from files.

The functional specifications are the requirements about what the program does and include information about the application area, the techniques used in transforming the input, the volume and complexity of the input and output data, the important factors relating to the intended users, and the user interface. This information would be determined at the Software Budget Proposal stage in the table above and the estimated accuracy level for human experts would therefore be within a factor of 2 by this source.

1.2 Results

Testing resulted in predictions that were, on average, 6% lower than actual code sizes. Predicted sizes ranged between a 66% over prediction and an 47% under prediction at the one standard deviation level. The approach appears to yield results that are essentially free of bias and of accuracy comparable to that of human experts, including the results from the literature that were used in the original Phase II proposal. We estimate that the functional specifications, used in the Code Sizing Tool and the JPL and DoD results are determined by the software budget proposal stage (see Section 1.1), where Boehm predicts and error factor of 2. This is consistent with the with both the project results and the work of human experts in the two studies as shown in Table 1.4:

Table 1.4
Test Results

Source	Predicted Size (% of actual size)		
	$-\sigma$	Most Likely	$+\sigma$
Code Sizing System	53	94	166
JPL	69	80	93
DoD	50	62	78

The results are comparable to those of human experts and do not have the systematic optimistic bias resulting in bias factors significantly greater than 1 in the JPL and DoD studies. It was noted in the JPL study that the size estimates in the study were used to budget the software development and there was a tendency to cut off the development after reaching or slightly overrunning the budget giving these results more accuracy than may otherwise have been achieved.

1.3 Conclusions and Recommendations

The test results indicate that the Code Sizing Tool is able to predict the size of programs in the COSMIC database based on functional specifications, with an accuracy similar to that of human experts, and without the optimistic bias characteristic of human experts. The project also resulted in the development of a Knowledge Acquisition Environment that was used to create the Code Sizing Tool and can be used to expand its domain and increase its accuracy. The project plans, for the development of a field prototype expert system, were fulfilled in Phase II. Our recommendation is that a field testing program be developed under Phase III with the goal of establishing and marketing the Code Sizing Tool as a commercial system. It would be desirable to have the participation of a number of independent developers of large software systems with an agreement that the knowledge base of the Code Sizing Tool would be expanded to include their software in exchange for the use of the tool. This would result in a useful product that would cover software in application domains, such as real time programming, that were unavailable during the Phase II project.

2.0 Approach

Small Business Innovation Research (SBIR) projects are performed according to a three phase process. The Phase I effort is designed to prove the feasibility of the scientific and technical ideas of the project. Phase II is the principal research effort. Under Phase III, non-Federal capital is used to develop commercial applications of the research and development. The development of an expert system can be classified into five stages [16]. The beginning stage is a demonstration prototype, a small demonstration system that handles a portion of the total problem domain. The goals for this stage of development are to convince potential sources of funding that the expert system approach is fusible and to test the ideas about problem definition and knowledge representation. A typical demonstration system will contain 50 to 100 rules and perform adequately on a few test cases within the restricted area of the problem domain. A demonstration prototype was developed as Phase I of the project and resulted in the funding of Phase II.

The next stage of development is a research prototype which will perform well on large numbers of test cases over the whole domain. Because the system development and testing is not complete at this stage, it will tend to fail on problems near the boundary of its knowledge domain and may fail on some problems well within its domain. A typical research prototype is a medium sized system with 200 to 500 rules. A research prototype was completed during the first year of Phase II.

When the system has achieved the level of a research prototype, its structure and performance are analyzed for possible enhancements and corrections. After the changes have been made and additional testing and refinement is done, the system will reach the stage of a field prototype. These systems have reached a performance level close to that necessary for a commercial system. They contain smooth, friendly interfaces and address the needs of the user community. These systems tend to be medium to large with 500 to 1000 rules. A field prototype was completed during the second year of Phase II to fulfill the requirements of the project.

The next stage is a commercial prototype, reached after extensive testing of the field prototype. At this point, the system may have been reimplemented in a more efficient computer language to reduce execution time and conserve computer memory. It must reliably satisfy the user's needs in an operational environment. Commercial prototypes typically contain from 500 to a few thousand rules. If the commercial prototype proves to be a viable product, it will reach the final stage of development as a commercial system. The commercialization of the Code Sizing Tool is left for Phase III of the SBIR schedule.

Each of the development stages described above can be described by a set of five activities [16]. The development process is too complex to be done as a series of sequential tasks and should be planned as a set of interacting and overlapping activities:

- Requirements Analysis: Identification, in detail, of the problem, the available resources, and the goals of the system. It is important that the goals be broad enough to be of practical significance but narrow enough to be practical to develop.

- Acquisition of Knowledge Sources and Tools: This includes sources of knowledge such as human expertise, the literature and existing computer codes, and the tools necessary to develop the system including the hardware and software such as expert system shells or development languages.

- Conceptualization of the important knowledge, relationships, control structures, subtasks, strategies, and constraints. It includes a determination of the level of detail, or granularity, in which the knowledge is represented.

- Formalization of the concepts into the frame work of a symbolic computer language or expert system shell. This includes a determination of the structure of the frames, rules, or other formal representations fo the knowledge.

- Implementation of the concepts into a working computer program. The interaction between the knowledge base, control structures, and the user must be taken into account at this stage.

- Testing the performance and utility of the system should be done continually. Problems uncovered in testing will focus the developer's attention effectively and cause a recycling through the other activities.

This section summarizes the approach for the development of the Code Sizing Tool, including an analysis of the requirements, the available knowledge sources and necessary tools, and the high level design.

2.1 Requirements Analysis

An expert system to size software will require various types of domain knowledge, depending on the type of input and the level of detail provided by the user. Figure 2.1 shows an idealized diagram of the stages in software development. The diagram shows the general goals set by a general manager. On the basis of these goals, the applications manager determines the functional specifications for a desired software system. These specifications determine "what" the program will do, but not "how" it will be done in terms of the computer implementation. The functional specifications require and reflect a knowledge of the domain or particular application the software will be designed for. The systems analyst then creates a detailed design including important data, file, and control structures. The programmer writes the code based on the detailed design.

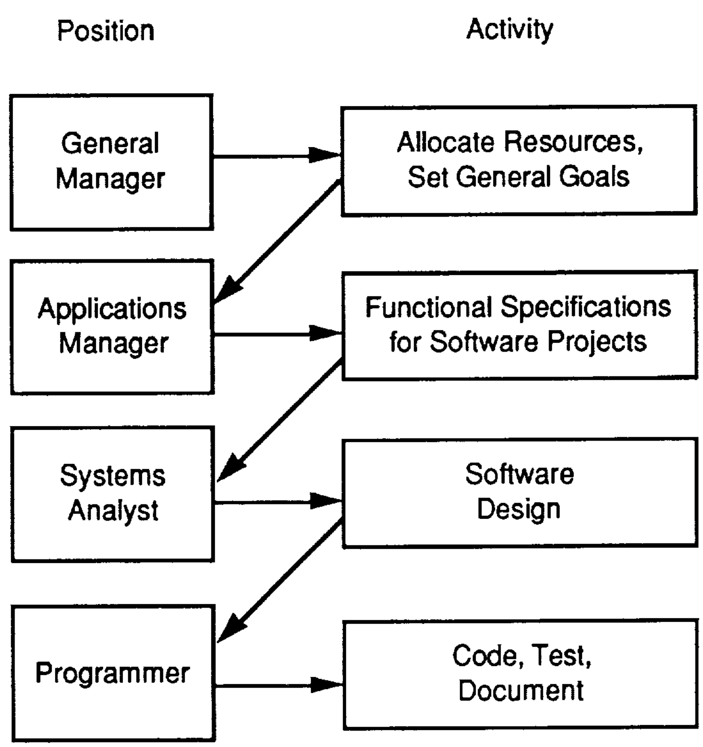
The procedures described above are an abstracted version of what actually happens during software development. Often people in more than one of the roles described above will work together on the same phase of development, and the same person may be involved in more than one stage. The diagram is not meant to be the best method, or even a typical method of software development. It is meant to illustrate the types of knowledge that would be useful in a code sizing tool. A tool that sizes code based on the detailed design would need some of the programmer's and system analyst's knowledge. It would have to have an understanding of the data, control, and file structures, and some knowledge of how they would impact the size of the codes. A tool that sizes code based on functional specifications would need knowledge from the application manager, the systems analyst, and the programmer. It would not be practical to go any higher than this and size code based on organizational goals or policies. It is therefore necessary to include both domain specific knowledge and Software Engineering Knowledge in the expert system.

The basic strategy of the system is to use application dependent knowledge, mostly from human experts, to classify the software and relate it to generic (application independent) software components. The generic components are then used to determine code size where the relationships between them are determined through regression techniques from a database of existing programs and documentation provided by NASA. This is similar to the Function Point [1,2] method for code sizing where generic components of the detailed design are supplied directly by the user and used to estimate size. In our case, a neural net facility was developed to provide flexibility in implementing both linear and nonlinear regression.

2.2 Knowledge Source and Tool Acquisition

The available knowledge sources were identified to be existing codes and documentation, human experts, and the literature. The necessary tools and facilities for performing the project can be classified into three groups: software tools for implementing the system, software tools for analyzing source code, and a hardware platform for system development.

Figure 2.1
Stages of Software Development



2.2.1 Existing Codes and Documentation

Code sizing is an unusual application for an expert system because human experts tend to perform relatively poorly in this area and produce results that are optimistically biased. The project strategy reflected this fact by acquiring knowledge from existing codes and documentation in addition to the knowledge acquired from experts. This strategy produced unbiased estimates that were as accurate than those of human experts. The project received the 38 programs shown in Tables 2.1 through 2.3 from NASA's COSMIC database. These programs were a valuable source of knowledge for analyzing scientific and engineering software.

Table 2.1

COSMIC PROGRAMS
1986 Edition, Part I

ENGINEERING

M86-10090	AFILE, ADRFT, INDEK- A NAVAL ARCHITECTURE ARRANGEMENTS DRAFTING SYSTEM, Pg 93.
M86-10253	ESTIMATING PRESSURE SURGES IN CRYOGENIC LIQUID-VAPOR SYSTEMS, Pg.93
M86-10206	IDSP- INTERACTIVE DIGITAL SIGNAL PROCESSOR, Pg. 94
M86-10093	BIASD- INTERACTIVE CIRCUIT SIMULATION, Pg. 99
M86-10207	THEORETICAL INVESTICATION OF DIELECTRIC HORN ANTENNAS, Pg. 100
M86-10175	SSPTA- SIMPLIFIED SHUTTLE PAYLOAD THERMAL ANALYZER (DVV), Pg 112
M86-10250	GTRAN- TRANSIENT ANALYSIS OF GAS PIPING SYSTEMS, Pg. 112
M86-10975	SINDA- SYSTEMS IMPROVED NUMERICAL DIFFERENCING ANALYZER (DVV), Pg. 137
M86-10985	TRASYS II- THERMAL RADIATION ANALYSIS SYSTEM (VAX VERSION), Pg. 137
M86-10215	HICAP- HISTOGRAM CLUSTER ANALYSIS PROCEDURE FOR MULTIDIMENSIONAL IMAGE DATA, Pg. 141
M86-10187	APT- NASA ENHANCED VERSION OF AUTOMATICALLY PROGRAMMED TOOL SOFTWARE, Pg. 147
M86-10201	NBOD2- PROGRAM TO DERIVE AND SOLVE EQUATIONS OF MOTION FOR COUPLED N-BODY SYSTEMS, Pg. 148
M86-10466	CARE III- COMPUTER AIDED RELIABILITY ESTIMATION, Pg. 155
M86-10195	DISCOS- DYNAMIC INTERACTION SIMULATION OF CONTROLS AND STRUCTURES (DEC VAX VERSION), Pg. 164

Table 2.2

COSMIC PROGRAMS
1986 Edition, Part II

M86-10198	MODEL- MULTI OPTIMAL DIFFERENTIAL EQUATION LANGUAGE, Pg. 164
M86-10199	NASTRAN NASTPLT PLOTTING POST PROCESSOR, Pg. 165
M86-10209	NEXUS/NASCAD- NASA ENGINEERING EXTENDIBLE UNIFIEL SOFTWARE SYSTEM WITH NASA COMPUTER AIDED DESIGN, Pg. 165
M86-10210	NASDS- NASTRAN / DISCOS / SAMSAN DMAP BRIDGING PROGRAM, Pg. 166
M86-10240	STAGSC-1- STRUCTURAL ANALYSIS OF GENERAL SHELLS (DEC VAX VER), Pg.168
M86-10332	SPAR- STRUCTURAL PERFORMANCE ANALYSIS AND REDESIGN (DVV), Pg. 173
M86-10442	PASCO- STURCTURAL PANEL ANALYSIS AND SIZING CODE (DVV), Pg. 182

MATHEMATICAL AND COMPUTER SCIENCES

M86-10039	SHADE- SHADED COLOR PICTURE GENERATION OF COMPUTER DEFINED ARBITRARY SHAPES, Pg. 221
M86-11098	LONGLIB- A GRAPHICS LIBRARY FOR THE DEC VAX COMPUTER, Pg. 229
M86-10046	FPT- FORTRAN PROGRAMMING TOOLS FOR THE DEC VAX, Pg. 231
M86-10180	NPP- NAMELIST PREPROCESSOR PROGRAM (DEC VAX VERSION), Pg. 235
M86-10181	SFORT- STRUCTURED FORTRAN PREPROCESSOR (DEC VAX VERSION), Pg. 235
M86-10184	SAP- FORTRAN STATIC SOURCE CODE ANALYZER PROGRAM (DVV), Pg. 235

Table 2.3

COSMIC PROGRAMS
1986 Edition, Part III

M86-10186	DIDS- DECISION INFORMATION DISPLAY SYSTEM, Pg. 236
M86-10200	NDDBS- N-DIMENSIONAL DATA BASE SYSTEM, Pg. 236
M86-10208	TAE- TRANSPORTABLE APPLICATIONS EXECUTIVE (DEC VAX VMS VERSION), Pg.236
M86-10220	IAC- INTEGRATED ANALYSIS CAPABILITY (IAC WITH CONTROLS), Pg. 238
M86-11066	SFTRAN3- STRUCTURED FORTRAN PREPROCESSOR, Pg. 250
M86-11080	ARCH- A FILE ARCHIVAL SYSTEM FOR THE DEC VAX, Pg. 252
M86-11086	LOOK- A TEXT FILE DISPLAY PROGRAM, Pg. 252
M86-10197	SAMSAN- MODERN NUMERICAL METHODS FOR CLASSICAL SAMPLED SYSTEM ANALYSIS, Pg. 255

PHYSICS

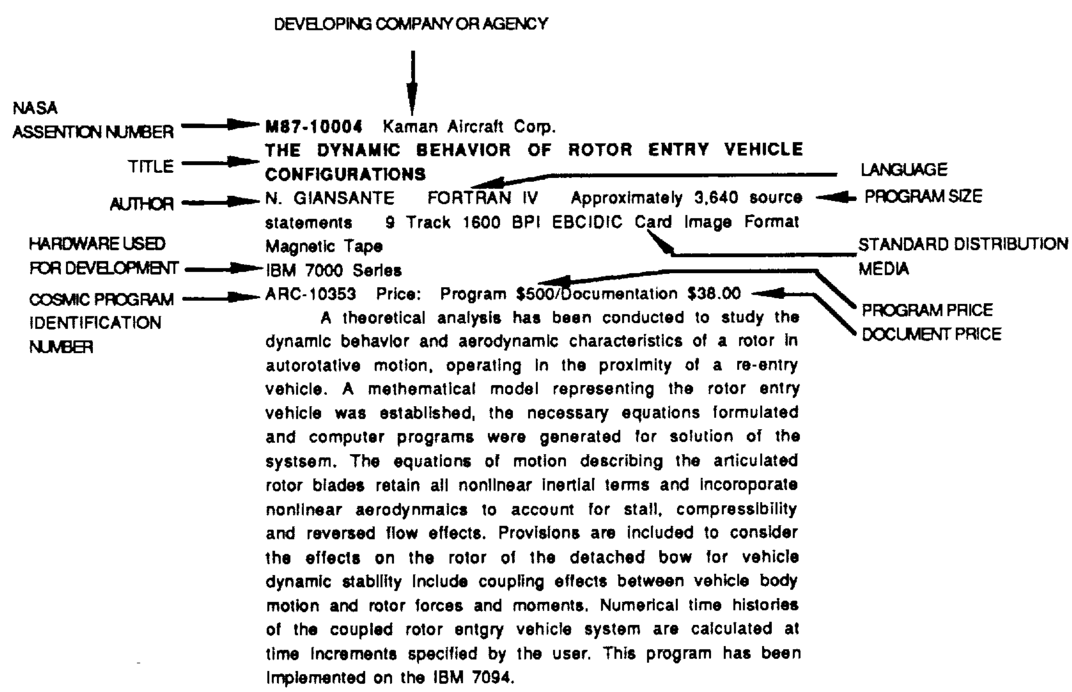
M86-10196	GENOPTICS- A GENERAL OPTICAL SYSTEMS EVALUATION PROGRAM (DEC VAX VERSION), Pg.282

DOCUMENTATION AND INFORMATION SCIENCE

M86-10418	RIM5- RELATIONAL INFORMATION MANAGEMENT DATA BASE SYSTEM (DEC VAX VERSION), Pg. 294

Another source of knowledge was the set of system descriptions from the Cosmic Catalogue. This resource provided broad, but shallow, information on approximately 1,100 systems in contrast to the codes and manuals which provided deep, but narrow, information on the 38 systems obtained from the COSMIC database. We have found that both types of knowledge were useful and that the catalogue information was useful in generalizing the knowledge obtained from detailed documentation of specific systems. Figure 2.2 contains a sample description of a program. The code size and a description of its functionality are included. The program description is rather short and many important aspects of its specifications are missing. It is a rather small program, and fortunately the codes that we are interested in, containing 10,000 line and over, tend to have explanations that are 3 or 4 times as long that provide much better coverage. Even for the larger programs, the descriptions are not complete, but this does not destroy their usefulness since defaults can be used to estimate the values of the incomplete specifications. This reflects the conditions under which the system will often be put to use in the field.

Figure 2.2
Cosmic Catalogue: Sample Program Abstract



2.2.2 Human Experts

During the course of the project, agreements were carried out with three consultants, two domain experts and one company expert in AI software development, ExperTelligence, Inc., a local (to Goleta) AI company, specializing in the development of intelligent software tools for the MicroExplorer. They have created the graphics interface for the MicroExplorer and developed an interactive, object-oriented user interface generator called Action™ which was purchased for the project and used in the code sizing system. They also provided support to the project by developing customized utilities for the expert system software environment and instruction in the use of the computer and user interface generator. This included most of the development of the source code analysis tools discussed later in this section.

The domain experts were taken from the University of California, Santa Barbara. Professor Yoshihiko Nakamura of the Robotics Laboratory is an expert in the design of software for mechanical control systems. Interviews were conducted with him from February through March of 1989. Mr. David Girardot of the Chemical & Nuclear Engineering department provided expertise in the area of software for process control. Interviews were conducted with him were conducted during April and May of 1989. Summaries of the interviews with the domain experts are given in Appendix xx.

2.2.3 Source Code Analysis Tools

We have developed tools that allow us to effectively analyze the structure of existing programs. They aided in extracting the essential structure and higher level subsystem components of large programs. There is a facility for displaying the subroutine calling structure of programs as a (tree-like) directed graph. With any large program, there is too much detail for much useful information to be determined from a display of the entire system. The facility also allowed us to focus in on and examine detailed pieces of the system. Software was developed to automatically determine the high level structure of the program by eliminating any subgraph whose "fullsize" parameter (a weighted sum of the sizes of the routines in the subgraph) is below a given cutoff. The results of applying this pruning algorithm to the SSPTA program with a cutoff of 500 lines, is shown in Figure 2.3. The routine name and fullsize parameter is shown for each node. It is possible to see the overall structure of the program and the contribution that each component makes to the total. Analysis based on Figure 2.3 is shown in Table 2.4. The table lists the root node of each major component of the system along with its function and size, in thousands of lines of executable Fortran code. It is apparent that the system contains five models, each of which simulates some important physical phenomenon related to the thermodynamics of an orbiting space shuttle. The results also show code requirements for generic software functions in the context of the application. Input processing, for example, uses 800 lines.

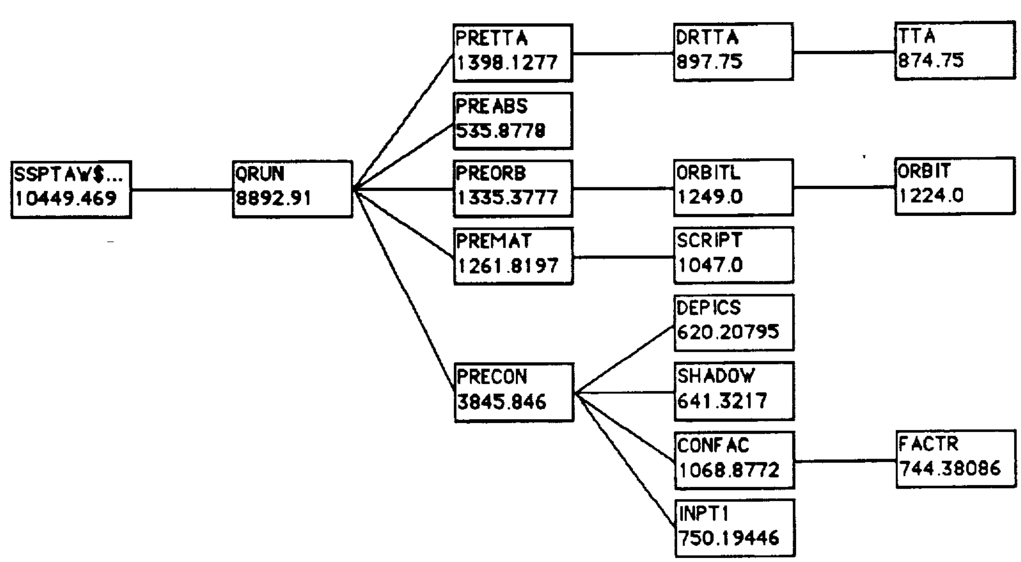


Figure 2.3
Top Level Tree Structure of Program SSPTA

Table 2.4 Higher Level Analysis of Program SSPTA		
Root	Function	Size
SSPTAW	Entire System	10.4
QRUN	Control Model Runs	8.9
PRETTA	Thermal Model	1.4
PREABS	Radiation Absorption Model	0.5
PREORB	Model Fluxes on Orbiting Shuttle	1.3
PREMAT	Model Radiation Couplings	1.3
PRECON	3D Geometrical Radiation Model	3.8
DEPICS	Wire-frame Plots	0.6
SHADOW	Shadow Factors	0.6
CONFAC	View Factors	1.1
INPT1	Input Processing	0.8

2.2.4 Neural Net Simulation Facility

We have created the software for simulating neural nets to use as a sizing function relating generic software components to code size. Its important features are summarized in Table 2.5. The system can simulate both linear and nonlinear, single and multilayer perceptrons [10,12]. There is no reason, however, to use multilayer linear nets since there is always an equivalent single layer linear net. Nonlinear nets use the sigmoid nonlinearity:

$$f_S(x) = \frac{1}{1+e^{-x}} \qquad (2.1)$$

The Perceptron Convergence Algorithm and the Back Propagation Algorithm are used for training linear and nonlinear nets, respectively. Nonlinear, multilayer perceptrons have the capability to classify regions of arbitrary complexity in feature space. Illustrations of single and multilayer nets are shown in Figures 2.4 and 2.5.

2.3 Conceptual Design

This section describes the high level design and knowledge representation scheme selected for the system. The design of the Code Sizing Tool and its development environment is specified in a number of ways in this report including algorithms in pseudo code, function and data definitions in LISP, and transformation diagrams, explained in Section 6. One element of the design philosophy is that attention was paid to the Knowledge Acquisition Environment as well as to the final production environment. Because code sizing itself cannot be divorced from software development, the process of software development is outlined and the types of knowledge involved in the different stages are described. Knowledge from the development stages, including the determination of the specifications through the coding of the program, are reviewed as a basis for selecting the knowledge representation scheme: frame structures in a LISP environment.

Table 2.5

Summary of Neural Net Simulation Capabilities

	Linear	Nonlinear
Layers	Single Layer	Arbitrary Number of Layers
Excitation	Linear	Sigmoid Nonlinearity
Learning	Perceptron Convergence Algorithm	Back Propagation Algorithm

	Capabilities
Single Layer	Gaussian Classifier on half planes in feature space
Two Layer	Classify Arbitrary Convex Regions in Feature Space
Three Layer	Classify Regions of Arbitrary Complexity

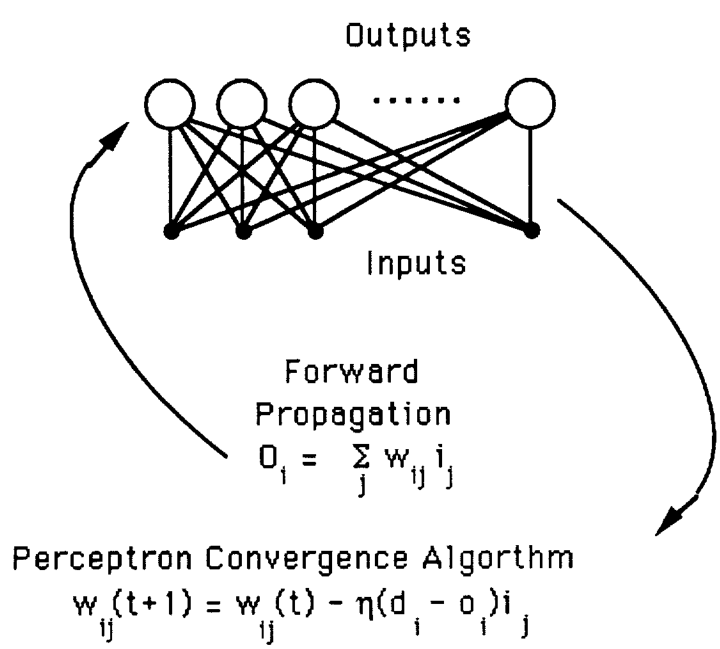


Figure 2.4
Linear, Single Layer Perceptron

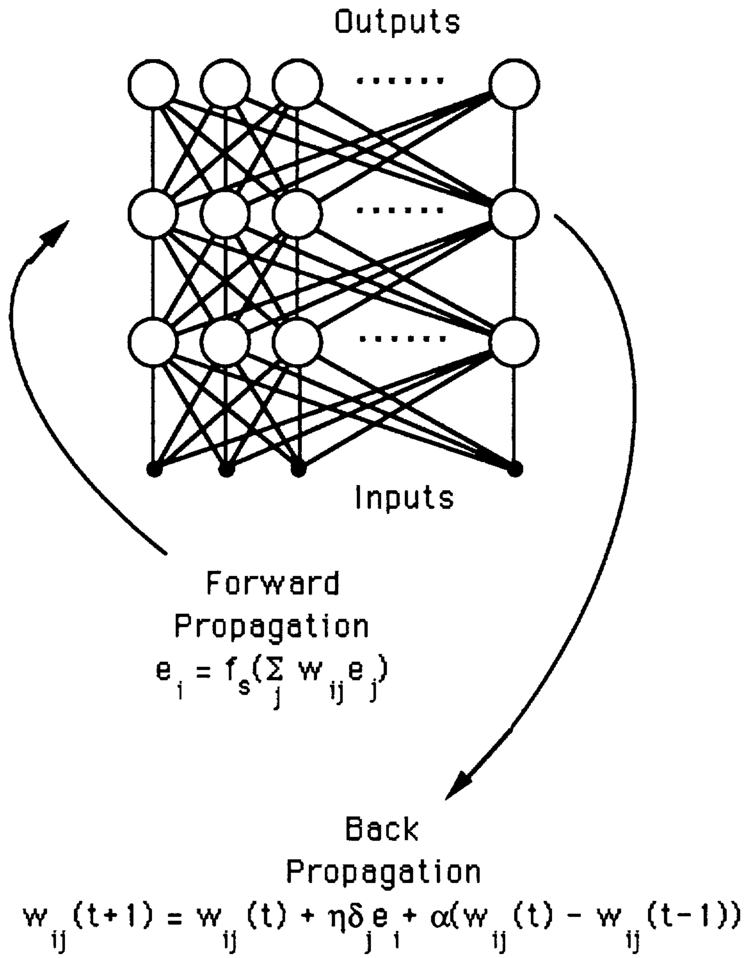


Figure 2.5
Nonlinear, Multilayer Perceptron

It was found that existing software metrics estimate size from parameters derived either from the system design or from the source code itself. The latter types of metrics are useless to the Expert System. The design based metrics require that the Expert System contain some knowledge of the design of the software at some level, based on the requirements. This approach is more promising that going directly from requirements to code size and was followed in the development of the system. The system operates in three stages:

1. Capture the User's Requirements
2. Determine the high level design components of the software
3. Use the design to estimate code sizes

The first two stages are the most difficult and require the most knowledge. The high level design is derived from the user's requirements and expressed by a set of generic software features, but it is not always possible to unambiguously determine these design components from the requirements.

The knowledge representation scheme was based on the principal that code sizing knowledge is software development knowledge and that some of the high level design of the software must be known by the system in order for it to size code. This is determined through the interaction between the user and the KBS. The KBS developer must therefore study the software development process to determine how to represent the knowledge. This process takes place in stages with different types of knowledge about the software under development created during each stage. It should be viewed in context, as part of the operations of organizations that develop computer technology:

- Set general goals for the organization
- Mandate a specific software development project and determine its specifications
- Design the software
- Code and debug the software
- Document the system
- Maintain the system

The stages typically go from more general, higher level activities to more specific, lower level activities. The results of the higher level activities are represented in natural (i.e., human) language. Natural language is also a part of all of the other results, even the source code will have comments. As the project progresses, however, more formal knowledge is created. The design is often expressed in some graphical representation such as a flow chart, and the source code is written in a computer language, which is a type of formal language specified by a grammar. The KBS is concerned only with the specification through coding stages. No input more general that the specifications is considered by the KBS. The source code for the each component of the system will, in the end, be represented by a size. As a program progresses from the specification to coding stages, the proportion information about the purpose of the program decreases in favor of information about the implementation of the program. Natural language is often represented by a frame based system, program designs are often represented graphically, and formal languages are often represented by a grammar. The frame representation is the most general of the three and can be used to represent graphs and grammars. Knowledge based systems are usually either rule based or frame based. The analysis of the knowledge in the system suggested that it be represented as frames. The knowledge going into the code sizing system is highly structured and the knowledge representation scheme should reflect this structure. Highly structured knowledge is easier to implement in a frame based system rather than in a rule based system where the knowledge base is a collection of independent heuristic rules. It was therefore chosen for the design of the KBS.

A more formal definition for the knowledge base is given below:

knowledge base = a set of frames

frame = a frame name and a set of slots

slot = a slot name and a set of facets

facet = a facet name and a set of values

This definition states that the knowledge base consists of an arbitrary number of frames. Each frame contains a name and an arbitrary number of slots. Each slot contains a name and an arbitrary number of facets, and finally, each facet contains a name and an arbitrary number of values. There are no restrictions on the choice of slot and facet names, although some will have special significance within the context of the KBS. Some of the important slots are defined below:

A-Kind-Of - The concept in this slot is a generalization of the concept represented by the frame.

Instance-of - The object represented by this frame is an example of the concept in this slot.

For example, an "even-number" can be generalized to (a-kind-of) "number", and "6" is an example of (instance-of) an "even number". Some of the important facets are defined below:

Value - One or more values can be assigned to the VALUE facet of a slot. This is an important facet because it is often the goal of the KBS to fill in one or more VALUE facets.

Range - A method for determining the validity of values for the VALUE facet.

If-needed - A routine to be executed if needed, to find a value for the VALUE facet.

If-added - A routine to be executed whenever a value is added to the VALUE facet.

If-removed - A routine to be executed whenever a value is removed from the VALUE facet.

The most basic routines for dealing with frames are defined below:

(FGET FRAME SLOT FACET)

Get a list of the value(s) given the FRAME, SLOT, FACET names. The empty list will be returned if any of the three elements are missing or if they are present but there are no values in the requested facet.

(FPUT FRAME SLOT FACET NEW-VALUE)

Add a single value, given the FRAME, SLOT, and FACET names. These elements are created if they don't already exist. The function returns NIL if the facet already has NEW-VALUE as one of its values, otherwise it returns NEW-VALUE.

(FREMOVE FRAME SLOT FACET VALUE)

Remove VALUE from the given FRAME, SLOT, FACET combination. The routine will remove VALUE if FRAME, SLOT, and FACET exit and have VALUE assigned to it. The routine will then return "t". The routine will do nothing and return "nil" if any of the previous conditions have not been met.

More sophisticated frame handling routines can be constructed from the primitive ones. An example is shown below that uses the "default" and "if-needed" facets when the "value" facet of slot is empty.

(FGET-V-D-P FRAME SLOT)

if there is a value in the VALUE facet, return it
otherwise if there is a value in the DEFAULT facet, return it
otherwise if there is a value in the IF-NEEDED facet, execute it to produce a value

The routine described below uses inheritance from more general frames to attempt to find a value. There is than one way to implement inheritance in a frame system. The example shown looks at the *value* facets of more "general" frames to find a value:

(FGET-I FRAME SLOT)

if there are any values in the VALUE facet, return them
otherwise form a list of all the frames connected to FRAME by the
GENERALIZATION or INSTANCE-OF facets, go through this list until a
non-empty VALUE facet is found for SLOT and return the contents, or *nil*
if none is found

The goal of the system is to fill in the VALUE and UNCERTAINTY facets of the SIZE slot of the PROGRAM frame. Since these values are initially unavailable, the IF-NEEDED function will be executed in order to get them. These functions can, in general, fill in other slots and therefore initiate other IF-NEEDED routines. They can also ask the user questions and access stored knowledge. In this way, the knowledge base and user's specifications are combined to determine the code size. The total size will depend on the various functional components of the program.

In the first stage of code sizing, the system captures the functional specifications for the program which are represented as a set of frames structures as a tree with the first layer of nodes containing frames for:

- The specific tasks performed by the software
- Solution methods employed by the software
- The time dependence of solutions modeled by the software
- Linearity of valid solutions
- Boundary conditions in the simulation
- Objects represented in the system

- Shape limitations on objects
- User Interface
- Graphics
- Commercialization
- Systems Programming

The specification structure contains a relatively large number of frames and is designed to represent application dependent requirements. It is designed to represent objective criteria that can be determined early in the software cycle. It can be expanded independently if the domain of the system is increased. The next stage of the code sizing procedure is to translate the specifications into a set of generic software components. The generic components apply to all software and are therefore application independent. The were designed to be able to represent the complexity of a program with a relatively small number of components which are not objective and therefore not input directly by the user. These components include various subjective aspects, such as complexity and generality, of the following criteria:

- Tasks
- Solution methods.
- Objects
- Representations
- Graphics
- User Interface
- Data Management
- Systems Program
- Hardware Architecture

The main goal of the user interface is to make the system efficient, easy to use, and accessible to the widest possible user community. The interface is window-based and includes graphics as well as text in order to accomplish this goal. A number of windows or "virtual displays" can be created and moved, pasted, scaled, and removed from different parts of the physical screen. This separated the task of determining the flow of information from that of formatting the screen and allowed for the development of a more sophisticated graphic interface. The use of the mouse improved the flexibility of the graphic inputs. The use of windows, a mouse, and menus resulted in an extremely flexible interface that can be used to handle many sources of information at the same time without confusing the user.

It was important to design an environment for building the knowledge base as well as using it. The knowledge base was constructed with a set of LISP tools in an interactive LISP environment. Much of the knowledge base was built using the primitive frame functions described above. These functions were also combined to create more complex, customized tools. Two important knowledge sources were the source codes and documentation from existing programs. Special utilities were designed and implemented for analyzing source codes and creating a database of routine frames that represent the calling structures of the codes. Processing the existing programs in this way was useful in developing the final knowledge base. The source codes were sent to a parser in order to determine the calling structure and code size statistics. This process is described below. The functionality of the important subsystems is determined by hand. These data and other knowledge obtained from experts and the literature is used to create the final knowledge base for code sizing. The actual process was iterative. The developer was continually switching between the knowledge acquisition environment to build or modify the knowledge base, and the production environment to test it.

The structure for the routine frames is given below:

```
(routine-name
      (size (executable-value) (total-line-value)
            (executable-subtree-size))
      (full-size (value) (if-needed get-fullsize))
      (routines-called (value) (number))
      (called-by (value) (number))
      )
```

There are some special types of routines. Routines with an empty "called-by" slot are "main programs", routines which exist as entries in one or more "routines-called" lists but which are not in the source code are "library routines", and routines with more than one entry in the "called-by" slot are "utility routines".

The "full-size" slot is filled in by taking a weighted sum of the subtree starting at the routines node. This parameter represents the weighted code size contribution of a routine and all of the other routine it can call either directly or indirectly. The determination of this parameter is complicated by the fact that a routine can call any other routine. Although program structures are generally tree-like, they are in fact more general directed graphs. It is important that the size of utility routines is not counted more than once. The contribution of a child to each parent is therefore divided equally among each parent. An example is shown in Figure 2.6. If routine B were identified with a specific functional specification, that specification's contribution to the code size would be the "full-size" of B, 20 lines of code, 10 from the routine B itself and 10 from its "share" of the utility routine D.

Initially, the parser will create a frame for each routine and insert the code size values in that frame:

```
create the first routine frame as the current frame
while there are tokens left in the source file
        get the next line
        if it's blank, do nothing
        else increment total-line-value
            if it's an executable line, increment executable-value
            if it's the last line in a routine, create a new frame as the current
                    frame
            if the routine name hasn't been found yet and the line has the routine
                    name, add it to the frame
```

The next stage is to determine the "routines-called". For this purpose, it might be valuable to temporarily save the starting and ending token number for each routine and take a second pass through the code. Then, as a first cut, the list could be built as follows:

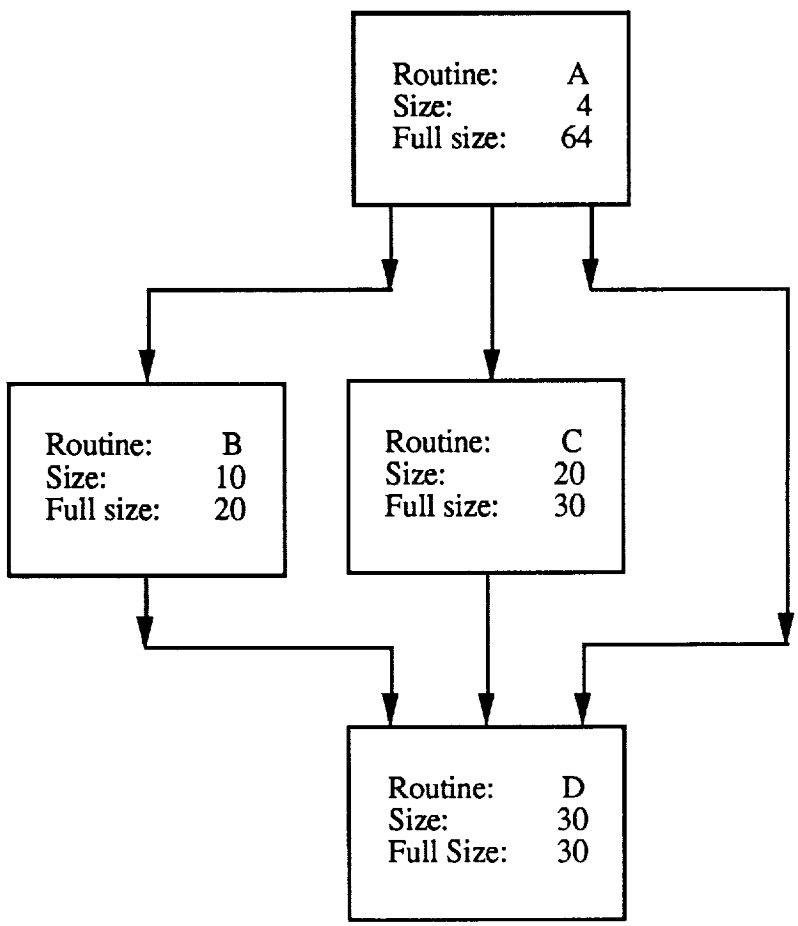
```
for each token in each routine:
        if the token is in a comment, do nothing
        else if the token matches a routine in the database,
            add that routine to the "routines-called" list
```

However this will not catch "library routines" as defined in the previous section. In a language like Fortran, array references cannot be distinguished from functions calls locally and an array list must be determined first. It is necessary to take "include" files into account.

for each token in each routine:
    if the token is in a comment, do nothing
    else if the token defines an array, add it to the array list
    else if the token qualifies as a call or array reference,
        if it's in the array list, do nothing
        else add it to the "routines-called" list

In Fortran, a token qualifies as a call or array reference if it starts with a legal character for a name, its not a keyword, and the next token is a left parenthesis. Other languages will have similar rules for determining a call. The "called-by" slot can be filled by inverting the calling tree. The knowledge is represented as directed graphs with a frame at each node. A utility was developed to display these structures with the option to display selected slot values at the nodes. The utility has the ability to display the trees only down to some specified level and to scroll the tree around in the window.

Figure 2.6
An Example of Subgraph Contributions



3.0 Project History

This section contains a short history of the project including project management information such as expenditures and the chronology of the task completions. Phase I occurred during the first six months of 1987 and resulted in the development of a demonstration prototype. Much of the knowledge for this prototype was derived from examining existing programs and relating source code modules to the software specifications they addressed. The quantitative relationship between them was derived in Phase I by using linear regression. The final report was submitted in June and the proposal for Phase II was submitted in July. The Phase II proposal was based on the typical development cycle for knowledge based systems with the major goals of producing a research prototype during the first year and a field prototype during the second. The task definitions from the proposal are shown in Tables 3.1 to 3.4.

NASA approved the Phase II proposal during the summer of 1988 and work started on July 7. MCR began the project with an analysis of project requirements and a request to change the hardware platform from a VAX 8600 to a MicroExplorer, a dedicated work station specifically designed for symbolic processing. The change was approved and resulted in a savings of approximately $90,000 (the difference between the budgeted time sharing costs on the VAX and the purchase price of the workstation system), which was applied toward increasing the amount and level of expertise available to the project. During the second quarter, the developers decided to implement the system in LISP and expert consultants were hired to assist in the development of software tools for the code sizing system and the analysis of source codes. Interviews were begun with domain experts and a search was initiated for an additional staff member at MCR to assist on the project. The search was successful and Dr. Yingti Xu was hired during the third quarter of the project. Knowledge acquisition and system development activity continued through the third and fourth quarter, resulting in the completion of the research prototype at the end of the first year.

An assessment of previous work was conducted during the fifth quarter of the project and a decision was made to increase the power and flexibility of the available regression techniques by implementing a neural network simulation facility that could perform both linear and nonlinear regression. It was also decided that, rather than going directly from software specifications to code size, an intermediate representation would be created. This intermediate representation was called *generic components* and represented traits, such as the complexity of the solution methods or data management, that are common to large classes of software systems. The idea was to classify programs by their generic components and eliminate the application specific knowledge after the initial stage of processing. Size is estimated from the generic components in the final stage of processing. The initial stage was done through the use of a rule base that related software specifications to generic components with the quantitative contributions determined by linear regression. The final stage was done through the use of a sizing function implemented as a nonlinear neural net, and calibrated through the use of nonlinear regression (the back propagation algorithm). The generic components were formalized and the neural net facility was developed during the fifth quarter. The specifications and rule base were developed during the sixth quarter and initial results for the field prototype were produced towards the end of the quarter. During the seventh and eight quarters the results were finalized, the user interface and other refinements were implemented, and the documentation was completed. The finished field prototype is able to produce results similar to that of human experts but without the systematic optimistic bias characteristic of them.

The project tracked well with the scheduled tasks and expenditures as shown in Table 3.5 and Figure 3.1.

Table 3.1
Task Definitions 1100 to 1230

1100 Project Management

This task includes the effort necessary to integrate and track the efforts of the project to achieve the objectives of the Phase II program within budget and on schedule. It includes responsibility for the scientific standards of the project work, status reports, documentation, and interface with NASA.

1210 Research Prototype Identification and Requirements Analysis

This task includes the effort to refine the objectives and requirements for the development of a research prototype. It includes the formulation of a more precise definition of the problem and the resources available to reach a solution. The available resources will be allocated to the tasks and potential difficulties will be determined and used in planning the efforts of the first year of the project.

1220 Research Prototype Knowledge and Tool Acquisition

This task contains an analysis of the available sources of knowledge for fulfilling the requirements of the research prototype. An analysis of the available languages and shells for system development will be preformed. The available computer codes for building the knowledge base will be identified, selected and procured. The task will also include a search of relevant literature. After decisions have been made on the knowledge sources and software tools, the implementation shell or language will be acquired and brought up on the computer. Expert consultation will be scheduled in the areas of Knowledge Based Systems and Software Engineering.

1230 Research Prototype Conceptualization

This task includes the high level design of the research prototype. It includes an analysis of how the knowledge will be structured and represented, how the various operations of the system will be controlled, and how the system will interface to the user. The interfaces and communications between the different components of the system will be designed. The conceptual design will be checked for consistency and verified against the objectives for the research prototype.

Table 3.2
Task Definitions 1240 to 1270

1240 Research Prototype Formalization

The acquired knowledge and high level design will be formalized in terms of the selected knowledge representation and expert system shell or symbolic language. The logical consistency and the consistency with the requirements and conceptual design will be maintained in the formal system.

1250 Research Prototype Implementation

At this stage, the formal structures will be integrated into a working system, debugged, and executed. An iterative process will take place between implementation and testing. As inconsistencies, errors, and other deficiencies are uncovered in the developing system, corrections will be implemented. Knowledge will be incrementally added to the system and the control structures and user interface will be refined. The objective is to bring the system"s performance up to the level of a research prototype.

1260 Research Prototype Documentation

Preliminary documentation of each task will take place during its performance. At this stage, the task based documentation will be integrated into a report on the first year's work. It will contain a description of the work done, and assessment of the results to date, recommendations for the second year, and an analysis of the feasibility of successfully completing the project in the second year. Preliminary documentation for the system will also be produced.

1270 Research Prototype Documentation

Preliminary documentation of each task will take place during its performance. At this stage, the task based documentation will be integrated into a report on the first year's work. It will contain a description of the work done, an assessment of the results to date, recommendations for the second year, and an analysis of the feasibility of successfully completing the project in the second year. Preliminary documentation for operating and maintaining the system will also be produced.

Table 3.3
Task Definitions 1310 to 1340

1310 Field Prototype Identification and Requirements Analysis

The objectives and requirements to bring the system's level of performance from the level of research prototype to the level of field prototype will be developed at this stage. Particular attention will be focused on those areas where the current system's performance is weak and on the problems near the boundary of the system's area of expertise. The remaining project resources will be allocated to the field prototype tasks.

1320 Field Prototype Knowledge Acquisition

This task contains the selection of additional knowledge sources that will fill in any gaps in the system's performance. It will also include another literature search for relevant work that has been published during the previous year.

1330 Field Prototype Conceptualization

At this stage, the design of the system will be examined to determine what changes are necessary to bring the performance up to the level of a field prototype. This will include an examination of the user interface, knowledge representation, and control structures in light of the previous year's experience.

1340 Field Prototype Formalization

The knowledge base and upgraded system design will be reformulated at this stage. The logical consistency and the consistency with the requirements and conceptual design will be reverified in the formal system.

Table 3.4
Task Definitions 1350 to 1370

1350 Field Prototype Implementation

The objective of this task is to implement a system that will satisfy the project goals. The system should perform competently over the entire knowledge domain. Attention will be given to overcoming the problems uncovered during the previous year. The system should have a smooth user interface and satisfy the needs of the potential users. The implementation process will therefore be focused on specific areas of the system's requirements.

1360 Field Prototype Testing

As in the research prototype, both incremental and formal testing will be included for the field prototype. The final formal testing procedures will document the extent to which the system has reached the goals of the Phase II research project.

1370 Field Prototype Documentation

Preliminary documentation of each task and the previous year's work on the research prototype will be integrated into the final report for the project. It will contain a description of the work done, and an assessment of the feasibility of developing a commercial product. Final user's and system's documentation will also be produced.

Table 3.5
Percent Completion by Task and Time

Task	10/88	1/89	4/89	7/89	10/89	1/90	4/90	7/90	Scheduled
1100	12	25	38	50	62	75	87	100	7/90
1210	90	100	100	100	100	100	100	100	9/88
1220	20	70	80	100	100	100	100	100	1/89
1230	15	90	100	100	100	100	100	100	3/89
1240	0	60	100	100	100	100	100	100	4/89
1250	0	0	50	100	100	100	100	100	5/89
1260	0	0	20	100	100	100	100	100	6/89
1270	0	0	10	100	100	100	100	100	7/89
1310	0	0	0	0	100	100	100	100	8/89
1320	0	0	0	0	100	100	100	100	10/89
1330	0	0	0	0	50	100	100	100	11/89
1340	0	0	0	0	0	100	100	100	1/90
1350	0	0	0	0	0	33	83	100	5/90
1360	0	0	0	0	0	16	67	100	6/90
1370	0	0	0	0	0	0	20	100	7/90

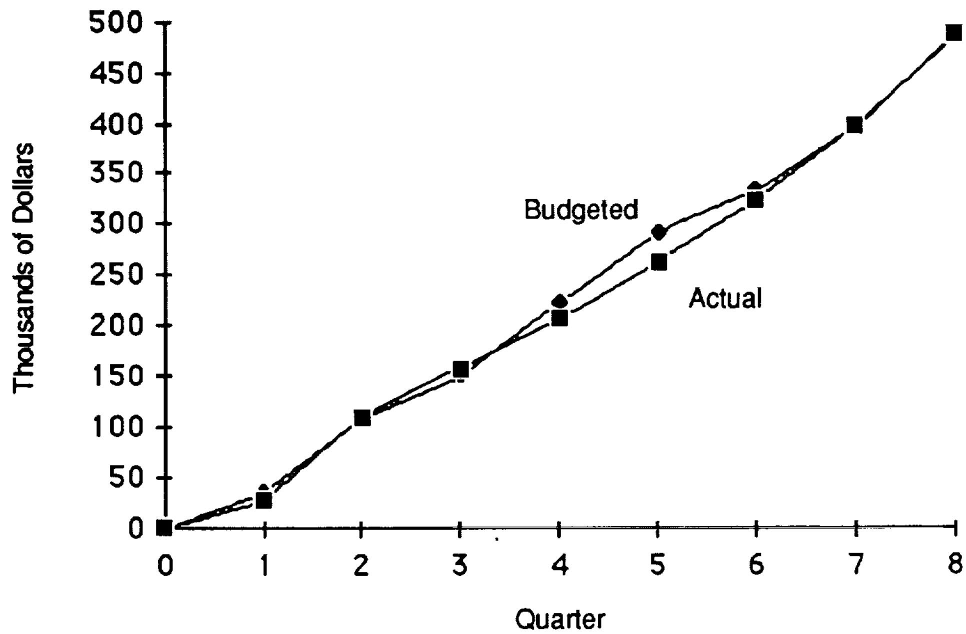


Figure 3.1
Cumulative Expenses

4.0 Technical Description

This section contains a technical description of the code sizing system including the procedures and techniques that were used in development, design and architecture of the implementation, and the mathematical formulas and algorithms that were implemented. Section 4.1 contains an overview of the development process and the major ideas that were incorporated into the system. Section 4.2 described the code sizing system itself including a subsection for each of the major components.

4.1 Overview

Figure 4.1 shows the elements involved in developing the major components of the code sizing system, the generic rule base and the sizing function. The initial step in the approach to the field prototype was the definition of a set of generic software components that could be used to classify any software program. These components take quantitative values and assign each program to a point (or distribution of points when certainty factors are taken into account) in an abstract pattern space. Pattern classification techniques can then be used to assign code sizes to the generic component vectors. Given this approach, there are a large number of techniques that can be applied, and we decided upon neural networks because of their power and flexibility. After a period of trial and error, we selected a nonlinear network with 16 inputs (one for each generic component), one 8 node hidden layer, and a single output node representing code size. Generic component values for existing programs were derived by examining their documentation and applying the generic component definitions. (This was a temporary measure since these components are derived from the specifications in the final system.) The results formed a generic database which also contained the code sizes of the programs and could therefore be used to calibrate a neural net designed to predict code size from generic components. During the initial period of the database development, the reasons for the particular value assignment were included in the database. They were used to help identify which software specifications were used in determining the generic component values. A testing procedure was devised that made effective use of the limited statistics available. Each program was removed from the database in turn and the net was calibrated with the remaining programs. The missing program's size was predicted by the net as a test of its accuracy.

The next stage of development was to determine a set of software specifications that would be supplied by the user and serve as the input to the code sizing system. The specifications were designed to be qualitative, as objective as possible and, of necessity, contained application specific information. This is in contrast to the generic components which are general and, to achieve the necessary level of generality, were somewhat subjective. The system was designed to relate specifications to generic components through a rule base that would combine the contribution from each specification in the software toward each generic component. The quantitative contribution of each specification to each generic component was determined through linear regression performed by using the neural net facility in a linear mode. A specification database was created and tests were performed. An examination of the initial tests showed that better results could be achieved by including two hand tuned examples: a large engineering program with most of the possible software features included, and a small scientific program with almost none of the possible software features included. A number of other experiments in the design of the networks and knowledge representation schemes were performed with no detectable improvements in the results.

The next stage was to use the linear network coefficients to determine the rule base for deriving generic components from specifications. This was done with LISP software which created the source code for the generic rule base. Additional knowledge was added to establish consistency within the specifications, and the final enhancements: the user interface, certainty handling, and help facility, were added to complete the implementation.

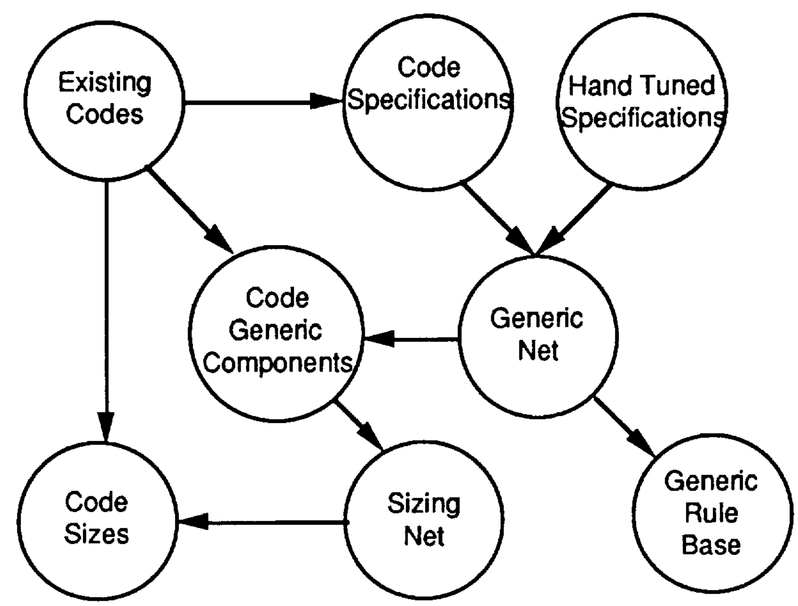
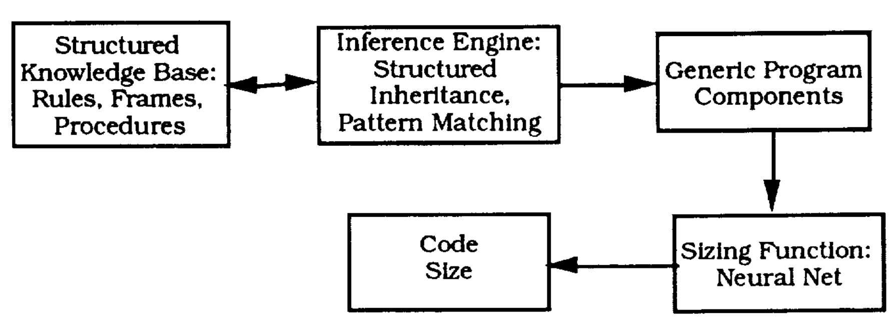


Figure 4.1
Major Components in the Development of the Code Sizing System

4.2 Production System

Figure 4.2 outlines the system architecture. The Knowledge Base is structured by a frame hierarchy and includes frames, procedures and rules. The Inference Engine uses structured inheritance, and the sizing function is implemented as a neural net. The basic approach to code sizing includes two procedures. The program is classified, based on its specifications, in terms of generic software components, and its size is estimated form the generic components by a neural net.

Figure 4.2
Expert System Architecture



This section describes the implementation of the code sizing tool. The user is asked for program specifications and their certainties. The specification values are converted to the elements of the "most likely" specification set using a portion of the knowledge base that maintains the consistency of the specification components. The results are fed to a code sizing facility that converts them to the predicted code size in two steps. First, a rule base is used to determine the generic software components from the specifications. The generic components, taken as a 16 component vector, are input to a sizing function, implemented as a neural net, to determine the predicted size. The specifications and certainties are also used by a Monte Carlo module that creates a population of additional specification sets which, when processed in the code sizing module, are used to determine the predicted certainty, expressed as a fluctuation factor, of the code size prediction.

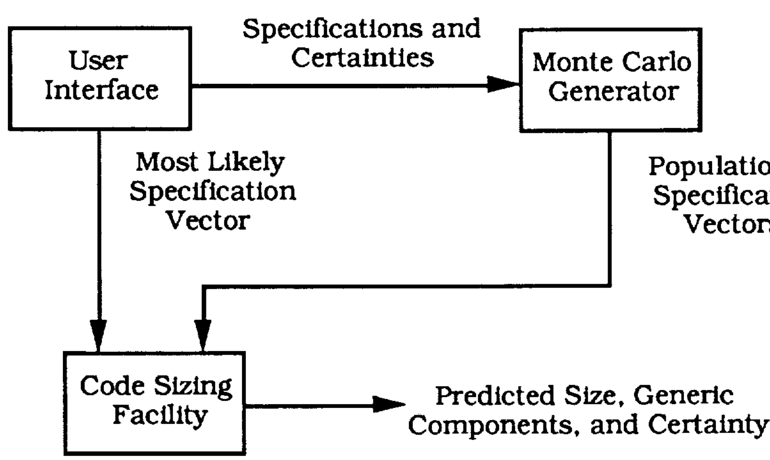


Figure 4.3
Overview of Code Sizing Tool Implementation

Figure 4.4 is a more detailed outline of the procedures used in determining the predicted code size. The user queries will be described in Section 5. The responses, which determine the program specifications and related certainties, will be described in this section, along with the related portions of the knowledge base and the procedures used to convert the responses to a consistent set of specifications which are used to determine the predicted (most likely) code size. The code sizing facility uses a rule base to determine the contribution of each component of the specification set to the code size.

4.2.1 Specifications

Figure 4.5 shows the structured set of specifications used in the Code Sizing Tool. Each specification is represented by a frame-slot pair, starting with the column containing "task", "methods", etc. The first generation of specifications contain the pairs "Task Engineering-program", "Task Database-program", "Methods Number-redundant-choices", etc. The second generation contains the pairs "Engineering-program

Structural-engineering", "Engineering-program Electronic-circuit", etc. These specifications cover general aspects of the software such as the user interface and graphics, and also task specific aspects such as the subject and solution methods. These specifications were based on the programs available to the project from NASA's Cosmic software database and cover Engineering programs as some database programs. It is expected that additional specifications will be included during the field testing and commercialization of the system.

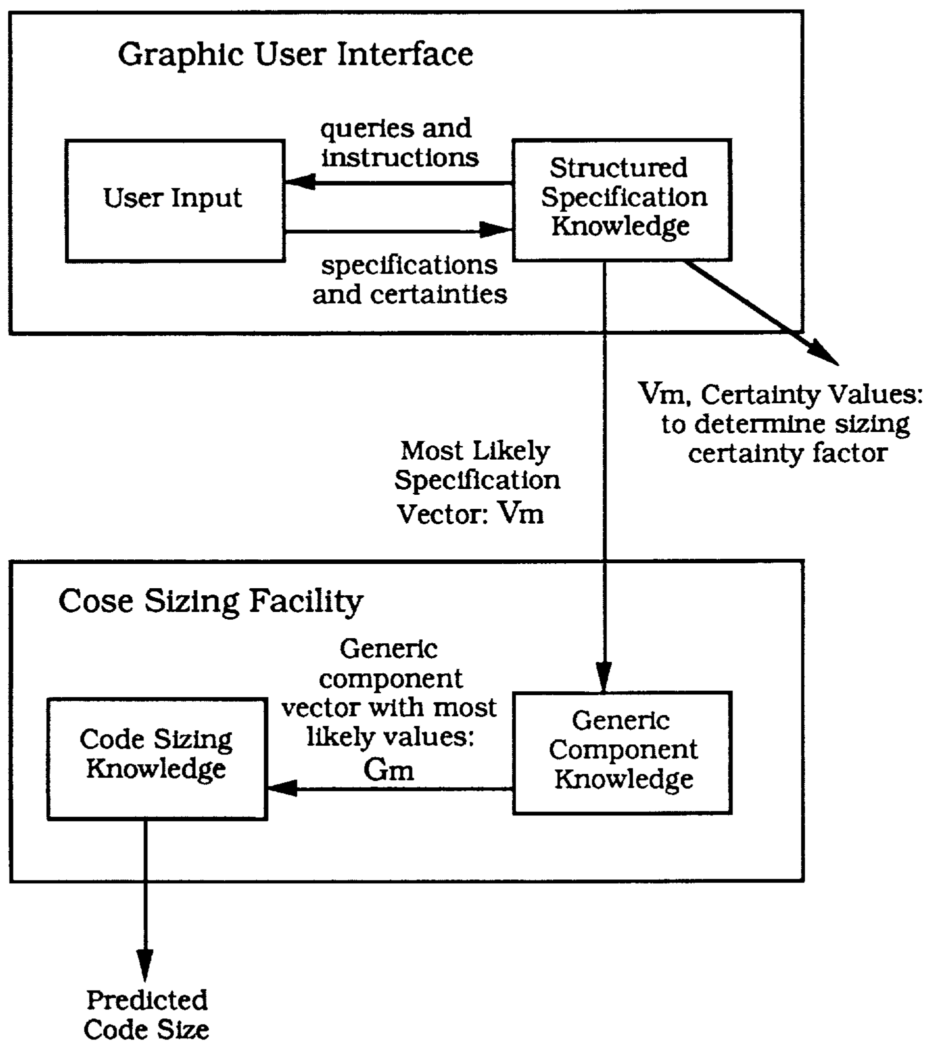


Figure 4.4
Determining Predicted Code Size

Each of the specifications has a corresponding user query, which is described in Section 5. The possible responses from the user and resulting possible specification values are shown in Table 4.1. Specifications are divided into two types *qualitative* and *quantitative*. Qualitative specifications have a value of *yes* or *no*, and the user is asked to provide a certainty from 0 (definitely not) to 1 (definitely). Quantitative specifications can take a numeric value which is represented by selecting one of a set of bins, each of which corresponds to a range of values. In the case of *Structure Spatial-dimension*, the specification can take a value of 0 (physical space is not represented) to

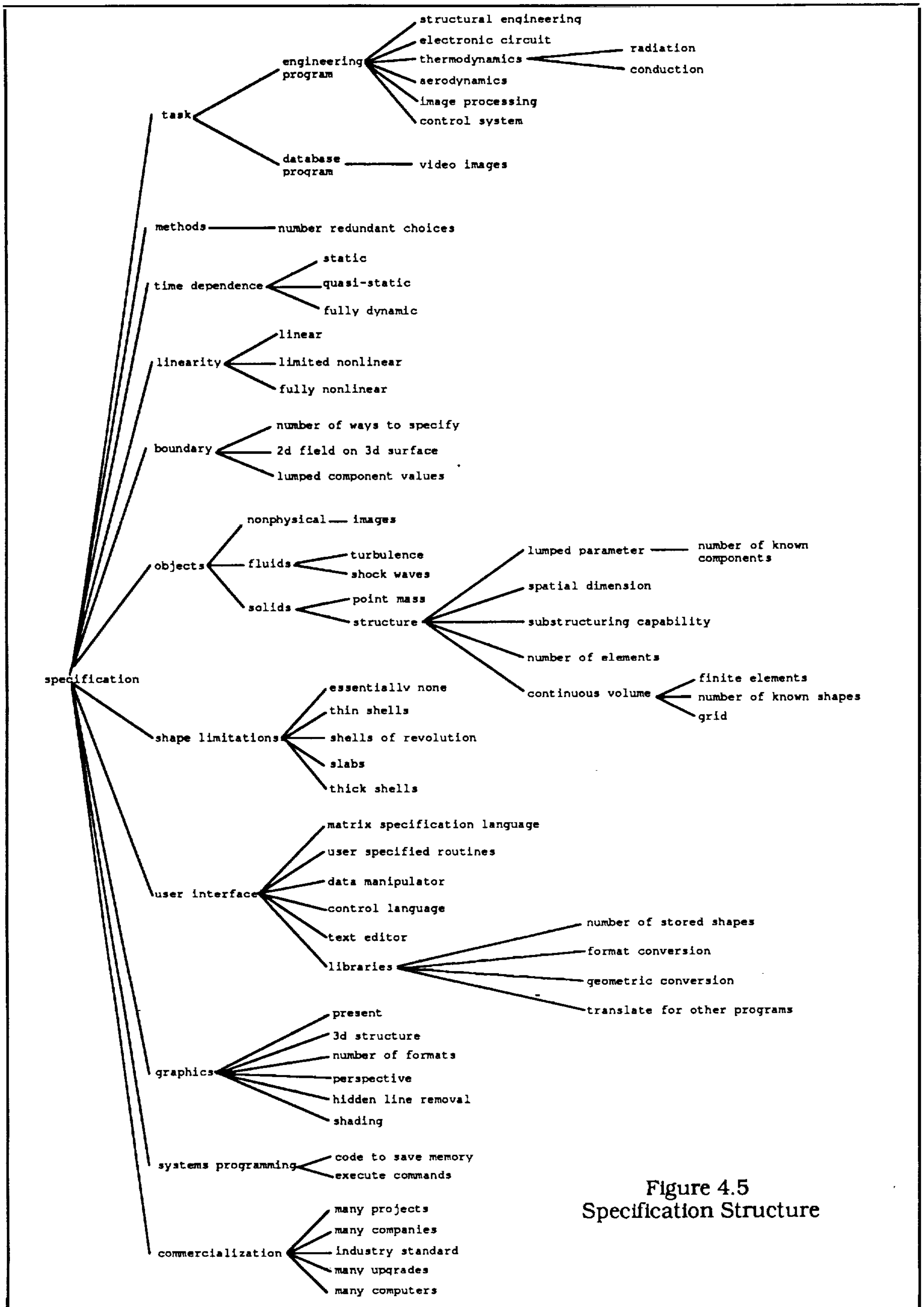


Figure 4.5
Specification Structure

ORIGINAL PAGE IS
OF POOR QUALITY

3. The other quantitative specifications can take on values of *very-high, high, medium, low*, or *very-low*. In addition to the selection of a bin, the user supplies a certainty factor from 0 (completely uncertain) to 1 (certain) for each quantitative specification.

Table 4.1
Representing Specifications for Programs

Specification Type	User Input	Resulting Specification Value
Qualitative	$0 \leq c \leq 1$	Yes, or No
Spatial Dimension	$0 \leq c \leq 1$, and $D = 0,1,2$, or 3	0, 1, 2, or 3
Other Quantitative Specifications	$0 \leq c \leq 1$ Select One of 5 bins	Very-High, High, Medium, Low, or Very-Low

Table 4.2 shows the ranges for quantitative specifications. The user selects the highest possible bin for the specifications. For example, the specification *Structure Number-of-Elements* refers to the program's capacity for representing large structures. It will have a value of *very-high* if the maximum is more than 40,000 elements, a value of *high* if it is between 40,000 and 10,001, a value of *medium* if it is between 10,000 and 2001, a value of *low* if it is between 2000 and 501, and a value of *very-low* if it is 500 or less.

As suggested by Figure 4.5, the specifications are interdependent. For example, *Engineering-program Structural-engineering* must have a value of *no* if *Task Engineering-Program* has a value of *no*. This is shown by their relationship in the figure. *Task Engineering-program* is the *parent* of *Engineering-program Structural-engineering*. In addition to its parent, the knowledge base contains other specification dependencies. For example, *Engineering-program Structural-Engineering* must have a value of *no* if *Solids Structure* has a value of *no*. The restrictions from parents and other dependencies are shown in Tables 4.3 to 4.5.

All of the dependencies are qualitative specifications as shown in the tables. If all of a specification's dependencies are not satisfied, ie, if any of them do not have a value of *yes*, then the dependent specification must have a value of *no, 0*, or *very-low* depending on whether it is qualitative, a spatial dimension, or another quantitative specification, respectively. This notion is formally expressed as a *meta rule* in Figure 4.6. The logic is referred to as a meta rule because it expands to 41 specific rules when combined with the structured specification dependencies in the preceding tables. An example of one of the specific rules governing specification dependencies in shown in Figure 4.6.

Figure 4.7 shows the implementation of the logic described in figure 4.6. The specification's dependencies are all evaluated before its value can be determined. This can result in a series of recursive calls to the evaluation routine since the dependencies may be dependent on other specifications. It necessary that there be no circular dependencies or the recursion will not terminate. This will not be a problem with the dependencies due to a specification's parent component because the specifications form a tree. The other dependencies must be checked to avoid this condition, however. At present this is done by inspection since the dependencies are few and not overly complex. If any of the dependencies have a value of *no*, the specification is given a negative value (no, 0, or very-low, depending on its type). Otherwise its value will depend on the user's input. Quantitative specifications take the value associated with the bin selected by the user, as described in Table 4.1. Qualitative specifications will take a value of *yes* if the certainty is greater than 0.5, and *no* if it is equal to or less than 0.5.

Table 4.2
Classification of Quantitative Specifications

Specifications Frame Slot	Classification Very High	 High	 Medium	 Low	 Very Low
Methods Number-redundant-Choices	>4	4	3	2	<2
Boundary Number-Ways-to-Specify	>4	4	3	2	<2
Structure Number-of-Elements	>40K	>10K	>2K	>500	≤500
Lumped-Parameter Number-Known-Components	>100	>30	>10	>3	≤3
Continuous-Volume Number-of-Known-Shapes	>20	>10	>5	>1	1
Libraries Number-of-Stored-Shapes	>100	>30	>10	>3	≤3
Graphics Number-of-Formats	>50	>30	>10	>5	≤5

Table 4.3

Specification Dependencies on Frame-Slot Pairs, Part I

Specification	Parent	Other Dependencies
engineering-program structural-engineering	task engineering-program	solids structure
engineering-program electronic-circuit	task engineering-program	
engineering-program thermodynamics	task engineering-program	
engineering-program aerodynamics	task engineering-program	
engineering-program image-processing	task engineering-program	
thermodynamics radiation	engineering-program thermodynamics	
thermodynamics conduction	engineering-program thermodynamics	
solids structure	objects solids	
solids point-mass	objects solids	
structure spatial-dimension	solids structure	
structure lumped-parameter	solids structure	
structure continuous-volume	solids structure	
structure number-of-elements	solids structure	
structure substructuring-capability	solids structure	
continuous-volume finite-elements	structure continuous-volume	

Table 4.4

Specification Dependencies on Frame-Slot Pairs, Part II

Specification	Parent	Other Dependencies
continuous-volume grid	structure continuous-volume	
continuous-volume number-of-known-shapes	structure continuous-volume	
shape-limitations essentially-none		objects solids
shape-limitations thin-shells		objects solids
shape-limitations shells-of-revol		objects solids
shape-limitations slabs		objects solids
shape-limitations thick-shells		objects solids
fluids turbulence	objects fluids	linearity fully-non-linear
fluids shock-waves	objects fluids	fluids turbulence
libraries number-of-stored-shapes	user-interface libraries	
libraries format-conversion	user-interface libraries	
libraries geometric-conversion	user-interface libraries	
libraries translate-for-other-progs	user-interface libraries	
graphics present		
graphics 3d-structure		graphics present

Table 4.5
Specification Dependencies on Frame-Slot Pairs, Part III

Specification	Parent	Other Dependencies
graphics number-of-formats		graphics present
graphics perspective		graphics present
graphics hidden-line-removal		graphics present
graphics shading		graphics present
lumped-parameter number-known-components	structure lumped-parameter	
database-program video-images	task database-program	non-physical images
non-physical images	objects non-physical	

Figure 4.6
Meta Rule for Structured Specifications

Given that:

S_i = The value of the i^{th} specification, and

$$N(i) \equiv \begin{cases} \text{no} & \text{if } i \text{ is a Qualitative Specification} \\ 0 & \text{if } i \text{ is Spatial Dimension} \\ \text{very-low} & \text{if } i \text{ is an Other Quantitative Specification} \end{cases}$$

the dependencies in Tables 4.3 to 4.5 form a set of rules of the following form:

If S_i is dependent on S_j
and S_j = no,
Then $S_i = N(i)$.

There are 41 such rules. For example:

If the value of (Task: Engineering-Program) is no
Then the value of (Engineering-Program: Structural-Engineering) is no.

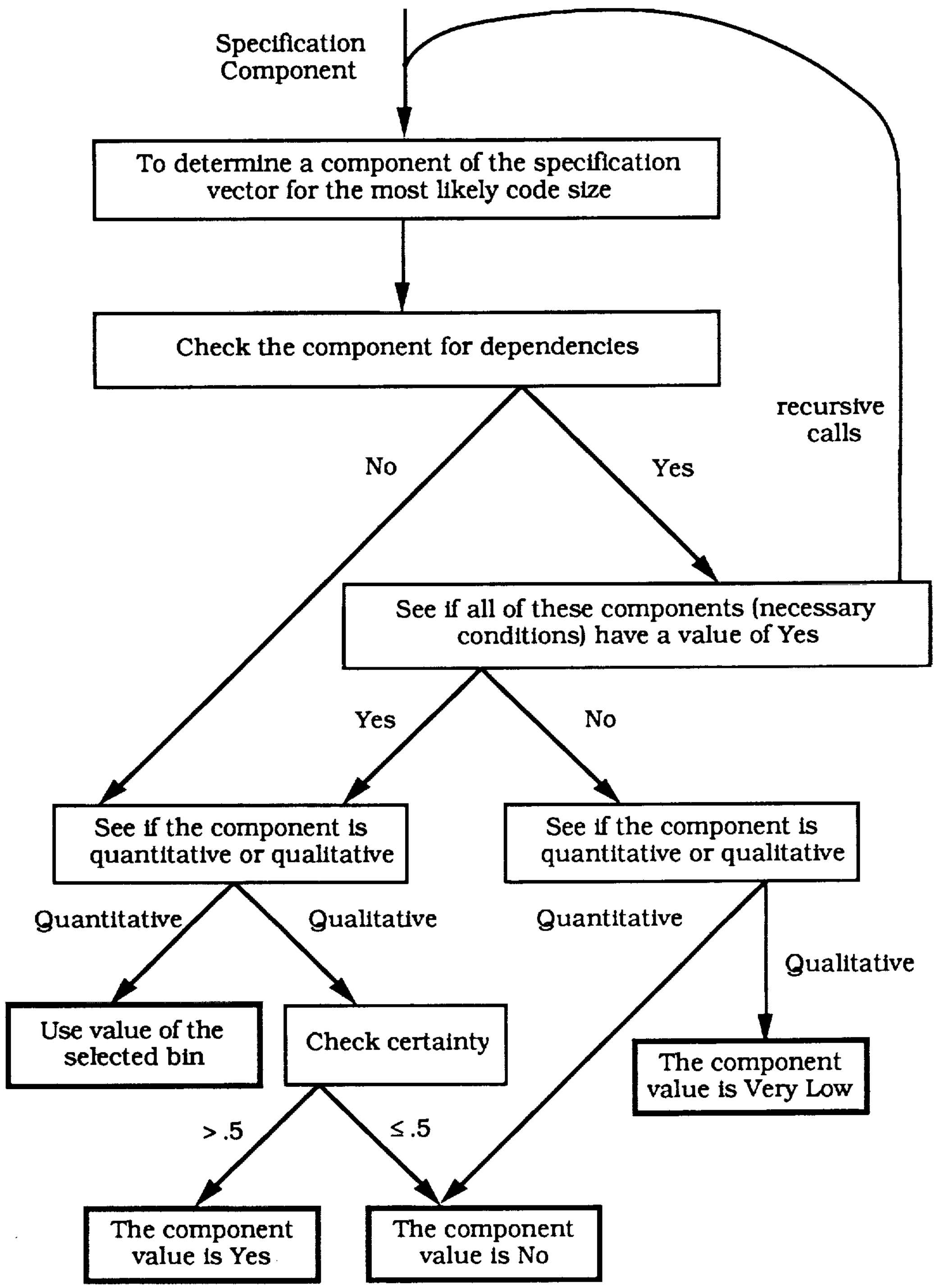


Figure 4.7
Determining the Specification Set of the Most Likely Code Size

4.2.2 Generic Components

Figure 4.8 shows an abstract representation of a computer application. The application exists in the "real world" and is implemented in both software and hardware. The "real world" of the application may, in fact, be another artificial world such as the world of abstract mathematics. It is sometimes referred to as the "problem space" or "perception/action space". This space is conceptualized as a set of objects and tasks or processes that act on them. For real time applications, the tasks are to measure and control some objects in the environment. For Engineering applications, the problem space is usually part of the physical world where objects interact through processes described by physical laws. The tasks or processes are implemented through some solution methods as software procedures that operate on data representations of the objects in the problem space. The application must reside on computer hardware and interact with the user. We have therefore defined the user interface as the link between the application and user, the system programming as the link between the application software and the computer, and the data management and the link between the data structures and the date storage devices.

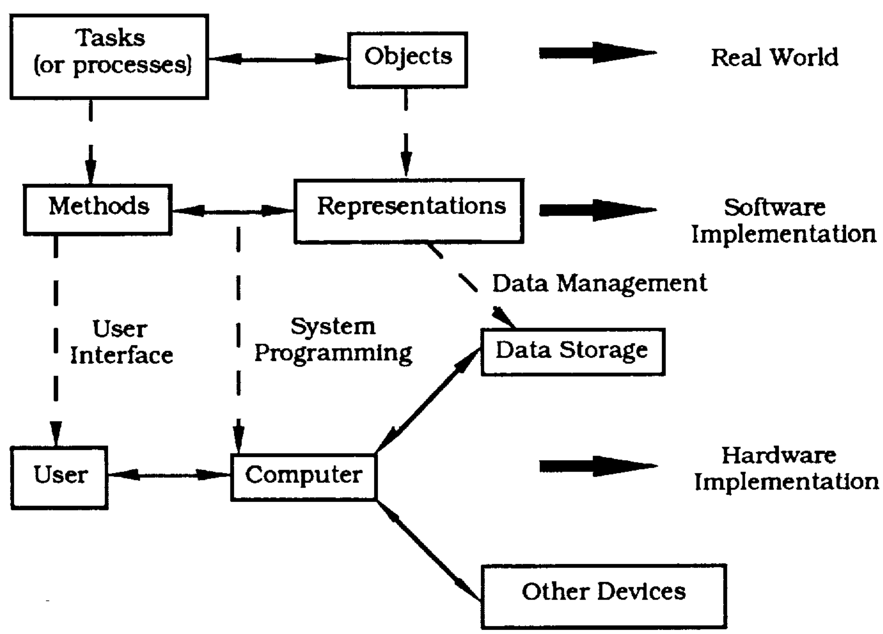


Figure 4.8
Abstract Representation of a Computer Application

We have used the abstract representation to determine the design criteria shown in Table 4.6. These criteria are being used as the generic software components from which code sizes are derived. These criteria are listed as entity, attribute, value triplets with the range of values given in the right-most column. This formalism is consistent with the frame, slot, value triplets used in frame based knowledge representation. The most important thing about this list is its generality. It can be applied to essentially any computer application.

These criteria are defined in Table 4.7. We have tried to make the definitions as objective as possible, but there seems to be a trade-off between generality and objectivity. Values for these criteria will not be supplied directly by the user, however, but will be inferred from more objective questions about the functional specifications. It is in the specifications where the application specific information will be contained. We believe the most important criteria are the tasks, objects, methods, and representations that define the problem and its basic software implementation. Table 4.8 contains definitions for the design attributes. They form a stereotypical set of characteristics which contain many of the, often conflicting, goals for computer based applications. We believe that the complexity is the most important attribute relating to code size and always include it. Other attributes are included for the most important criteria in order to better access their impact on the system. Finally, the values are defined in Table 4.9. We wanted a set that can be easily transformed into a quantity and provide a adequate level of detail.

As part of the Field Prototype design, we determined criteria for further formalizing the Knowledge Base and increasing its scope while remaining consistent with previous work. This included the introduction of certainty calculations (described later in this section) that were created in a way that was consistent with the most likely size calculations. In order to accomplish this, it was helpful to assign numeric values to all of the elements that are used to represent the software such as its features, design criteria, and functional specifications which are represented as frame-slot pairs in the code sizing system. The (quantitative) values of elements are either be taken from the user (when they correspond to functional specifications), or inferred from other elements. Finally, some of these elements (the generic software components) are used to determine code size through the use of a sizing function. This part of the system was calibrated through the use of existing examples and nonlinear regression techniques. Some components, such as spatial dimension, are naturally quantified. Other components were quantified in a straight forward manner, ie, 1 for true, 0 for false; or 1 for very high, .75 for high, .5 for medium, .25 for low and 0 for very low.

Table 4.6
Design Criteria

Criterion	Attribute	Range of Values
task	complexity	very low - very high
methods	complexity	very low - very high
	generality	very low - very high
	accuracy	very low - very high
	efficiency	very low - very high
	redundancy	very low - very high
objects	complexity	very low - very high
	generality	very low - very high
representation	complexity	very low - very high
	capacity	very low - very high
	knowledge	very low - very high
graphics	complexity	very low - very high
user-interface	complexity	very low - very high
data-management	complexity	very low - very high
sys-programming	complexity	very-low- very high
hard-architecture	complexity	very low - very high

Table 4.7
Design Criteria Definitions

Objects	Entities in the application that possess some relevant attributes. Analogous to *nouns* in language.
Tasks	Processes or transformations that change the condition of the world in which the applications takes place. Analogous to *verbs* in language.
Representations	Data structures in the software that correspond to OBJECTS in the application.
Methods	Algorithms or procedures that correspond to TASKS in the application.
Graphics	Visual output other than text.
User-interface	Algorithms or procedures that handle the interaction between the user and the software.
Data-management	The interaction between the software and information kept on mass storage devices.
Sys-programming	The interaction between the software and the operating system, such as multitasking and interrupts.
Hard-architecture	The configuration of the hardware used in implementing the system.

Table 4.8
Design Attribute Definitions

Complexity	The amount of information needed to specify a component.
Generality	The range of specific instances covered. The lack of restrictions or special cases not handled by the component.
Efficiency	The relative speed in which activities are accomplished. The effort that went into speeding up the activities.
Accuracy	The relative level of correctness. The effort that went into increasing the correctness of the results.
Redundancy	The amount of alternative choices for the user in accomplishing the same task. Where each choice may have some advantages and disadvantages over the others.
Capacity	The quantity of representations or representation components that can be used by the system simultaneously.
Knowledge	Application data to help the user represent objects. This includes libraries of shapes or physical properties, and preprocessors.

Table 4.9
Design Value Definitions

Very-high	Complete, As high as possible or practical, State of the art.
High	Through, One of a few major goals.
Medium	Typical, One of a set of goals.
Low	Minimal, Not a goal.
Very-low	As small as possible or practical, Ignored or avoided.

4.2.3 Rule Base

Although the system contains structured knowledge expressed as frames, rules, and procedures, the contents of the knowledge base will be expressed as rules in this section for easier documentation. A description of the various types of rules that go into the system follows. The rules are divided into three groups: control, program specifications, and generic software components. The rules that control the execution specifications, and generic software components. The rules that control the execution of the system determine the transition between its internal states, most of which correspond to windows in the user interface.

If the system is in the initial state
Then display the control window and set the specifications to their default values

If the system is displaying the control window and
the user issues a request to edit the specifications
Then display the appropriate specification window

If the system is displaying a specification window and
the user disposes of the specification window
Then return to the control window

If the system is displaying the control window or a specification window and
the user issues a request for help
Then display the appropriate help window

If the system is displaying a help window and
the user disposes of the help window
Then return to the previous window

If the system is displaying the control window and
the user requests a code size estimate
Then begin determining generic software components

If the system is determining generic software components and
all generic components have been determined
Then determine code size by applying the sizing function and produce output

If the code size has been determined
Then display the results window

If the system is displaying the results window and
the user disposes of the results window
Then return to the control window

If the system is displaying the control window and
the user selects *Quit* from the upper menu bar
Then exit the Code Sizing System and Action.

Specifications can be determined either by asking the user or by deduction from other specifications. An example of each type of rule is given below:

If the system is displaying the *Task* specification window and
the value for TASK ENGINEERING-PROGRAM is set in the window
Then store the value in the ENGINEERING-PROGRAM slot of the TASK frame

If a certainty of 0 is stored in the ENGINEERING-PROGRAM slot
of the TASK frame
Then set the value of the STRUCTURAL-ENGINEERING slot of the
ENGINEERING-PROGRAM frame to a certainty of 0

The system will determine a numeric value for each generic component. There are rules to determine the initial value of each component and rules to determine the incremental contribution from each specification:

If the system is determining generic software components and
  the current component is TASK COMPLEXITY
Then initialize the value of TASK COMPLEXITY to v_0.

If the system is determining generic software components and
  the current component is TASK COMPLEXITY and
  the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES
Then increment the value of TASK COMPLEXITY by v_k.

The rules shown above can easily be represented in the frame based system in a structured manner. Both specifications and generic components are represented as frame/slot pairs and all of the rules that determine the value of a given component are included in its *if-needed* routine. The example shown below (Table 4.10), for TASK COMPLEXITY, uses the *iff* and *inc* macros:

```
(iff FRAME SLOT VALUE S1 S2) =>
        (if (equal (fget-z FRAME SLOT) VALUE) S1 S2)

(inc SYM VAL)=> (setq SYM (+ SYM VAL))
```

We have developed and applied a set of procedures for determining this part of the knowledge base, including a calibration of the values for the parameters such as $v_0,\ldots\ldots v_n$, defined in Table 4.10. The databases containing program specifications and generic components were used to determine the contribution of each specification to each generic component with linear regression. An estimate of the accuracy was determined by testing programs not used in the calibration. The calibrated parameters and the frame structures for the specifications and generic components were used to determine the knowledge base, expressed as a set if *if-needed* routines that contain the rules for determining each of the values needed by the system. Once the rules have been written onto disk and compiled, the system can make predictions. At this stage the rules are accessible to the developer and can be edited by hand and recompiled if necessary. This process is shown in Figure 4.9.

Table 4.10

Sample IF-NEEDED Routine

```
(defun task-complexity (frame slot &aux val)
  (setq val v0.)
  (iff 'task 'engineering-program 'yes (inc val v1))
  (iff 'task 'database-program 'yes (inc val v2.))
  (iff 'engineering-program 'structural-engineering 'yes (inc val v3))
          :
  (iff 'systems-programming 'code-to-save-memory 'yes (inc val vn))
  (fput-list frame slot val))
```

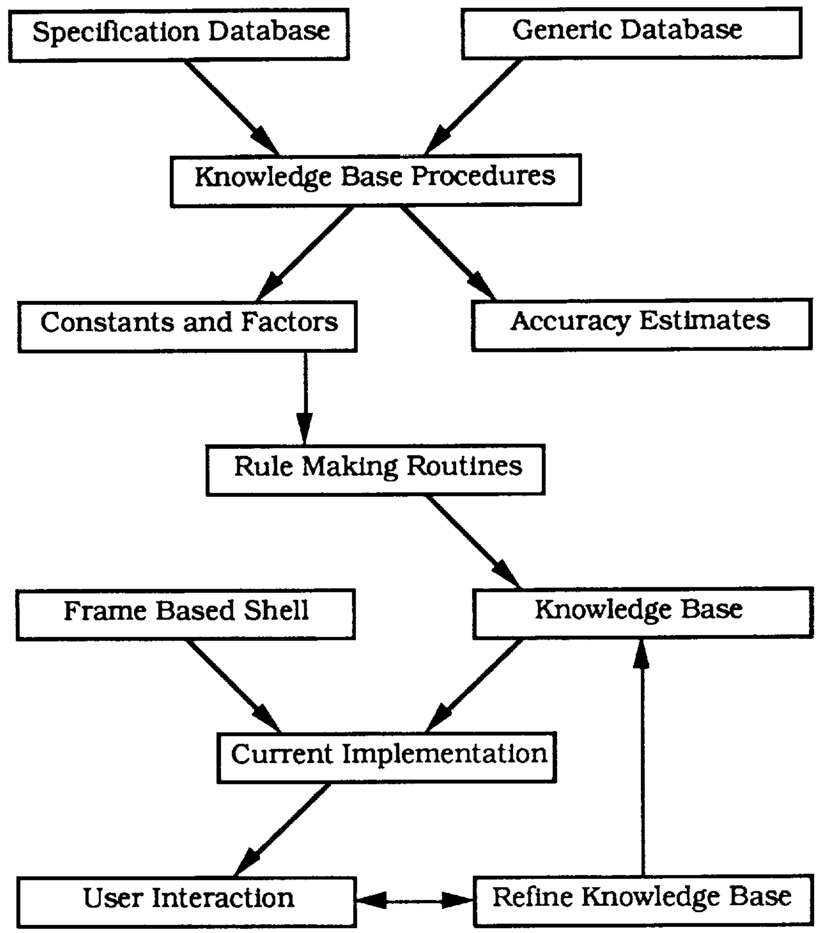


Figure 4.9
Implementation Procedures

4.2.4 Sizing Function

The generic software components are used as inputs to the neural net that determines code size. The code size prediction is a function of the output of the net. We experimented with two configurations for the output nodes. In the first configuration, there was a set of output nodes, each one covering a small range of code sizes. The output node with the highest excitation would then determine which of the discrete set of sizes is being predicted. For example, if we classify programs into 40 size groups starting at 1,000 lines and increasing by 20% for each successive group, the system would have the capability to size programs form 1,000 lines to 1,200,000 lines with an accuracy of 20% which was adequate for our purpose. In the other configuration, there is a single output node whose excitation determined the code size according to the following equation:

$$S = 1000^{(e+1)} \qquad (4\text{-}1)$$

where S is the code size and e is the excitation of the output node. This function has a continuous range from 1,000 to 1,000,000 lines. The second configuration, with a single output node, was the most accurate and was therefore selected for the Code Sizing Tool.

4.2.5 Certainty Handling

This section describes the certainty handling and its role in the estimation of the accuracy of the predicted code size. An overview of this process is shown in Figure 4.10. The specifications and certainties are used by a Monte Carlo procedure to create a population of specification sets, each of which is used by the code sizing facility to determine a code size. These code sizes are statistically analyzed to produce an uncertainty associated with the uncertainties in the specifications provided by the user. This is combined with the uncertainty estimated to be inherent in the system itself as determined by the verification procedure. The results are expressed as a fluctuation factor for the predicted code size.

The system creates a statistical population of specification sets and assigns values to the components of the software specifications of each member of the set. This assignment is based on the user's input and is similar to the procedure for determining the most likely code size, except that the assignment is based on a probability density function for each component of the specification. Each set of specifications is consistent in terms of the internal dependencies of the individual components within the set, although their values will be different among different members of the statistical population. The probabilistic version of Figure 4.6 is shown in Figure 4.11. If a specification has no dependencies, its value will be stocastically determined (by weighting the results of a random number generator) from a local probability density function. The value of dependent specifications will be determined in the same way but only if the dependencies have all been satisfied.

Figure 4.12 describes the implementation of the logic in figure 4.11 and includes a description of the local probability density functions. Determining this function was straight forward for qualitative specifications where the probability of a *yes* value is equal to the certainty supplied by the user. The determination was more complex for quantitative specifications. It was defined as a normal distribution where the mean value corresponds to the value selected by the user and the standard deviation depends on the certainty supplied by the user. The dependence on the specification's certainty was defined so that all choices have equal probability when the certainty is 0, and the selected value has a probability of 1 when the certainty is 1.

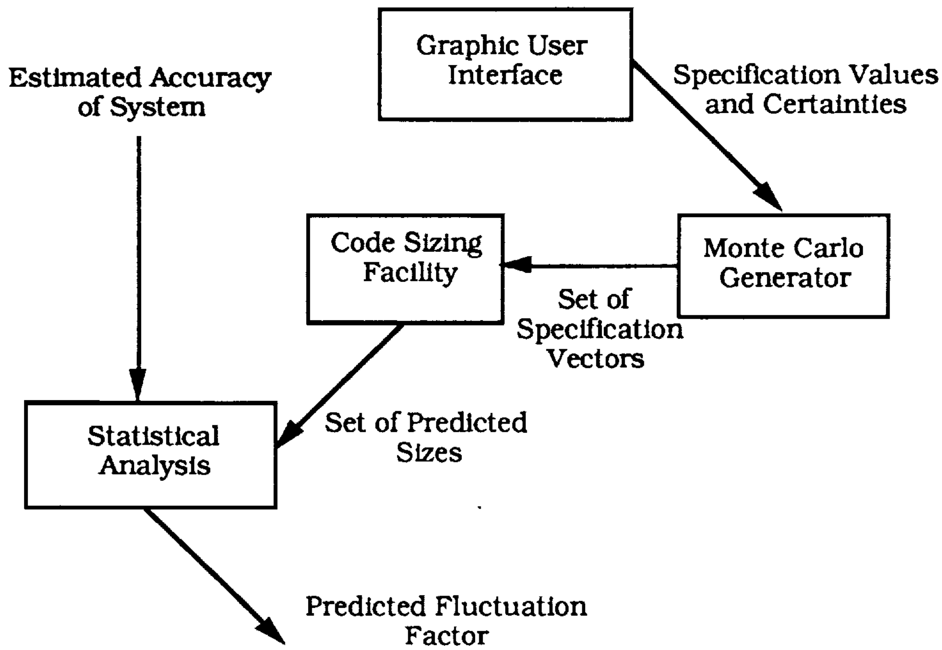


Figure 4.10
Determining The Predicted Fluctuation Factor

Figure 4.11
Probabilistic Interpretation of Meta Rule for Structured Specifications

$P^i_k \equiv$ the probability that the ith specification node will have a value of k

In the implementation, the probability for a given specification depends on a local distribution of probabilities for that specification and on the probability that all of its dependencies have been satisfied:

$$P^i_k = P^i(x = x_k \mid \bar{D}^i)$$

Where D^i is the probability that all dependencies have been satisfied. It is determined by the following product:

$$D^i = \prod_{j \in C_i} P^j_{yes}$$, where C_i are the dependencies for the i^{th} specification,

and the local distributions, $P^i(x)$, are as described in Figure 3.8.

Figure 4.12
Monte Carlo Generator

For Each Specification

If components dependencies aren't satisfied, give it a negative value (No, Very-Low, or 0)

Otherwise, use Monte Carlo Method with the following weights:

For qualitative specifications :

$$P_{yes} = c, \qquad \text{and} \qquad P_{no} = 1 - c,$$

where c is the certainty

For quantitative specifications, the probability for a specification taking the k^{th} value in its range is:

$$P_k = \int_{x_{k-1}}^{x_k} p(x)\ dx$$

where $x_i = i/N$, and N is the number of values in the specification's range. The probability density function p(x) is defined by:

$$p(x) = K\, e^{-f(c)\,(x-x_m)^2}$$

where K is a normalization factor:

$$K = \int_0^1 p(x)\ dx,$$

$x_m = m/N$, the value corresponding to the selected (most likely) value for the specification, and f(c) is a function of the selection certainty defined so that all values have equal probability when c = 0, and the most likely value has a probability of 1 when c = 1.

$$f(c) = 4c\,/\,(1 - c).$$

Code sizes have log-normal statistics, ie, their logarithms are distributed normally. The uncertainty in code size prediction is expressed as a fluctuation factor, which is based on the standard deviation of the logarithms of the code size estimates produced by the Monte Carlo procedure described above. This uncertainty in code size, due to the uncertainty in the inputs, is combined with the uncertainty in the code sizing system as shown in Figure 4.13, to determine the total estimated uncertainty.

Figure 4.13
Code Size Statistics

Code sizes have a log-normal distribution with statistics defined as follows:

$$p(\xi) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{\xi-\xi_0}{\sigma}\right)^2}$$

where $\xi = \log(s)$, s = code size, and ξ_0 and σ are, respectively, the mean and standard deviation of ξ. This results in code size accuracy predictions that take the form of bias and fluctuation factors. The code sizing tool will produce essentially unbiased predictions with errors due to uncertainty from both the system itself and the specifications provided by the user. The resulting predicted fluctuation factor will be:

$$\log^2(f_t) = \sqrt{\log^2(f_s) + \log^2(f_u)}$$

where f_t is the predicted total fluctuation factor, f_s is the fluctuation factor due to the uncertainty in the code size prediction process as estimated from tests of the system (as described in the previous report), and f_u is the uncertainty due to uncertainty in the user's specifications. The factor f_u is estimated from the population of size estimates, $\{s_i\}$ derived from the Monte Carlo process:

$$\log(f_u) = \sqrt{\frac{\Sigma_i (\log(s_i) - \xi_0)^2}{N}},$$

$$\text{where } \xi_0 = \frac{\Sigma_i \log(s_i)}{N},$$

and N is the population size, currently set to 20.

4.2.6 Code Sizing Tool Predictions

The system predicts the total number of delivered lines of source code including both executable and comments, in units of thousands of lines. It also provides error estimates in terms of a dimensionless fluctuation factor. The Code Sizing Tool was calibrated with Fortran programs from the COSMIC database and estimates are initially provided for the Fortran language. Estimates for some other languages are provided through the use of the conversion factors shown in Table 4.11. The sources for these factors were Boehm [6], which provides object/source instruction expansion ratios (p 478), Halstead [8], which provides measures of computer language complexity (p68), and the experience of the researchers.

The system also provides the normalized weights of the generic components that are derived in the first state of the prediction process. They are meant as a guide to how the predicted code will be used in the program.

Table 4.11
Computer Language Conversion Factors

Language	Factor
Fortran	1.0
Cobol	1.2
PL/I	0.8
Pascal	1.0
APL	0.5
Jovial	1.3
C	1.0
Algol	0.9
Assembler	3.0

5.0 Operational Description

This section describes the Code Sizing Tool from an operational point of view and functions as a User's Manual for the software. It includes some information on using the hardware, but only on a superficial level. Users not familiar with the Macintosh and MicroExplorer will have to consult their manuals, which are included with the project delivery. A short description of how to power up the hardware and start the Code Sizing Tool is included in these introductory remarks. There are two major subsections: one on the Production System, and one on the Knowledge Acquisition System. The first subsection begins with a description of the Action™ based interface to the Production System and of how to use the mouse to operate the screens and display objects which make it up. It contains a detailed description of each screen in the Code Sizing Tool and the options available to the user with its use. The second subsection describes the operation of the Knowledge Acquisition System. Since this system was used to develop the Production System, it's operations are more technical. It is a LISP based system and its operation is documented through the use of function descriptions and data structure definitions.

The instructions for bringing up the system are summarized in Table 5.1. The external disk must be turned on first so it will be recognized by the PC when the PC is brought up.

Table 5.1
Getting Started

1. Power up the hardware
2. Bring up the MicroExplorer
3. Log in under NASA
4. Set the default LISP package to TB
5. Run Action
6. Start the Production System interface application

It is powered up by pressing the key on the upper right hand corner of the keyboard. The MicroExplorer is started by clicking on the MicroExplorer Icon:

microExplorer,

which resides in the MicroExp folder of the internal hard disk, HD. Once the MicroExplorer is running, the user logs in under the NASA user name by typing:

```
> (LOGIN 'NASA)
```

and setting the default LISP package:

```
> (PKG-GOTO 'TB)
```

The machine is now set up for the use of the Knowledge Acquisition System. If the user wants to use the Production System, he starts it through the Action™ interface package. When the Production System has reached the level of a commercial prototype, it can be separated form the development environment of the interface package (see the manual), and will then exist as a separate icon which is clicked on with the mouse to initialize it. The user currently types:

(ACTION)

and activates the interface to the Production System according to the procedure for selecting an application under Action. It is documented in detail in the manual and is summarized below:

Table 5.2
Selection an Application in Action™

1. Click the mouse anywhere on the first screen
2. Select *Open* from the *File* menu on the menu bar at the top of the screen
3. Double click on the SCREEN-INTERFACE.LISP file in the NASA folder
4. The *Control* screen for the Production System will appear. Press the space bar while holding the key to go into *run* mode.

The first screen of the Production System will be displayed and ready to use. The instructions for operating it are given in the next subsection.

5.1 Production System

The interface to the production system will be used by a large and diverse group and was therefore carefully designed for simplicity and flexibility. In order to achieve this goal as efficiently as possible, the interface was implemented with the Action software tool. It is a utility for creating graphic user interfaces that are object oriented and which use the screen display, mouse, and keyboard to achieve maximum flexibility. The interface conforms to the MacIntosh's user friendly interface standards. It is described in this section including a short description of how to use MacIntosh interface elements such as *buttons*, *pop-up menus*, and *edit fields*. Users not familiar with the MacIntosh may need to consult its manuals.

The procedure for creating the interface is outlined in Figure 5.1 (more detail are available in the Action™ manual). The interface is composed of a set of windows (sometimes referred to as *screens* in this section), each of which contains a set of *display objects*. Both the windows and display objects are *objects* as defined in object oriented programming, and can therefore have internal variables and attached methods and can send and receive messages. The developer creates an interface with window and display objects and attaches characteristics to them including the names of methods that are activated when the display objects are accessed during runtime. The actual code for these methods is written by the developer as part of the application program. The developer can interactively switch from development to run mode with this system by pressing the space bar while holding down the key. This is a powerful tool for creating user interfaces that maintains independence from the details of the application software.

The interface makes extensive use of the mouse during both the development and operation of the software. A number of operations with the mouse will be defined here in order to utilize the graphic elements of the interface. Moving the mouse will cause the cursor on the screen to move and the user can *point* to an object by moving the cursor within its boundaries. The user can *click* on an object by pointing at it with the mouse and pressing the mouse button. The user can *drag* the cursor by pointing at an initial position and moving it to a final position with the button held down. Figure 5.2 shows the display objects used in the Code Sizing Tool within an application window. The top one is called a *Field Edit Object*. It can be used to enter text. The user clicks within the rectangle and uses the keyboard to enter the data. The delete and arrow keys, as well as the standard MacIntosh edit features, will work within the rectangle. A *Button Group Object* is used to select one of a number of alternatives by clicking the mouse on the selected one. If *two* were selected in the figure, the second circle would fill in and the first would clear. A *Button Object* is used to preform some action by clicking on it with the mouse, after which the attached method is executed. The *Scroll Bar Object* is used to input quantitative values between a developer specified minimum and maximum. The value will be proportional to the location of the *slide* within the body of the scroll bar. The slide can be moved in one of three ways. Pointing to one of the arrows and holding the mouse button down will cause the slide to move toward the cursor; pointing to a location in the body of the scroll bar and clicking the mouse will cause the slide to jump toward the cursor; and pointing to the slide and dragging the cursor to a new location along the bar will cause the slide to move there. The *Pop Up Menu Object* is used to select one of a number of alternatives. The user points at the rectangle and holds the mouse button down. This will cause the menu to pop up and the cursor is dragged to the desired selection.

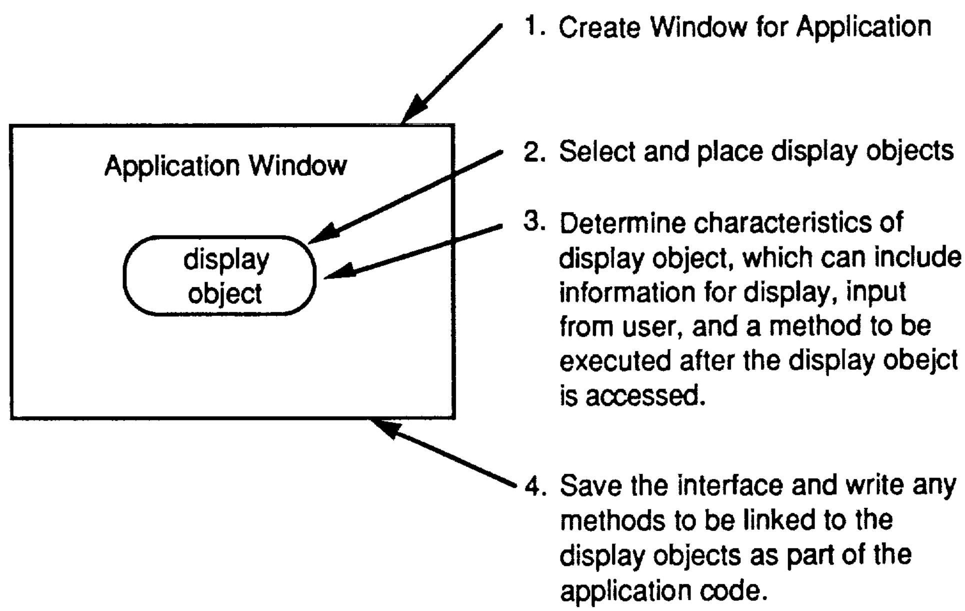


Figure 5.1
Steps in Creating the User Interface

The *Control* screen in Figure 5.3 is the initial screen of the application. The button marked *specify* will cause each of the seven specification screens, named *task*, *methods*, *objects*, *structure*, *user-interface*, *graphics*, and *other*, to appear in turn so that the user can enter the program's specifications. Below the *specify* button is the pop-up menu *Edit* for selecting one of the seven specification screens for editing. The next two buttons, marked *load* and *store*, work in conjunction with the edit field on the right labeled *file*. The load or store button will load or store program specifications using the file whose name appears in the edit field. The next button, marked *Size Estimate* should be pressed after all of the specifications have been entered to obtain the code size and fluctuation factor estimates which will appear in the *Results* screen shown in Figure 5.4. The screen shows the estimated size in thousands of lines of the default language (Fortran) and the estimated error expressed as a fluctuation factor. The estimate can be converted for other computer languages by using the pop-up menu below the size estimate. The relative weights of each generic component are also given on the right side of the screen. Pressing the *OK* button will release the *Results* screen and bring the user back to the main *Code Sizing* control screen.

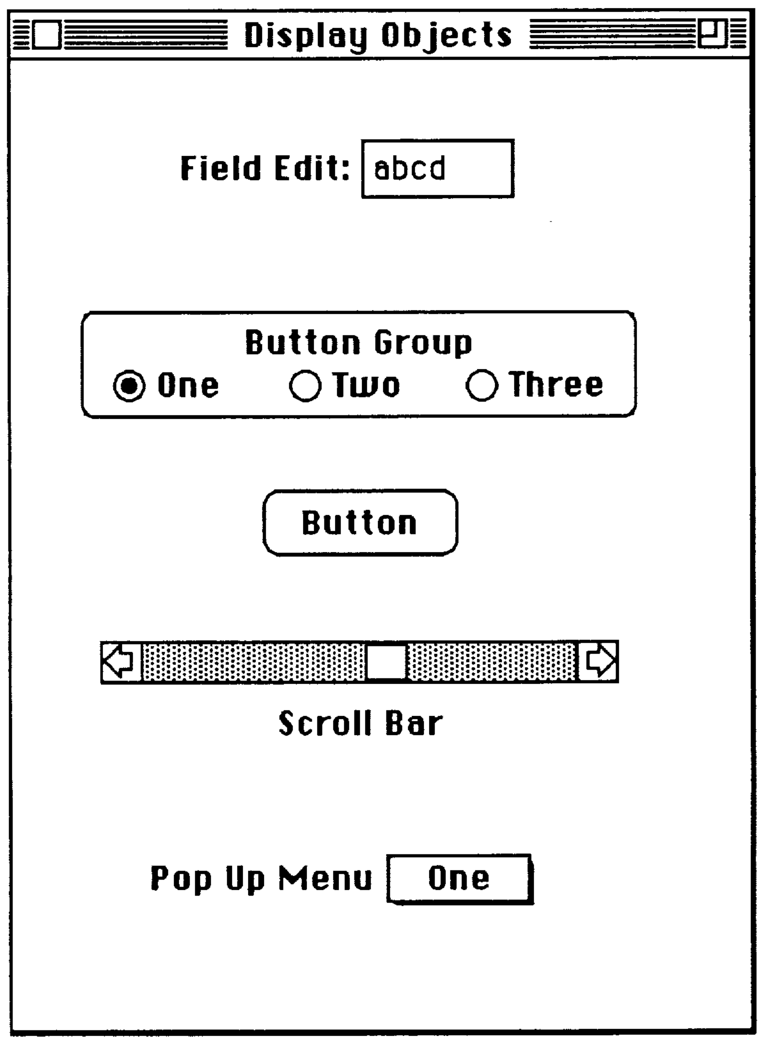


Figure 5.2
Window with Display Objects

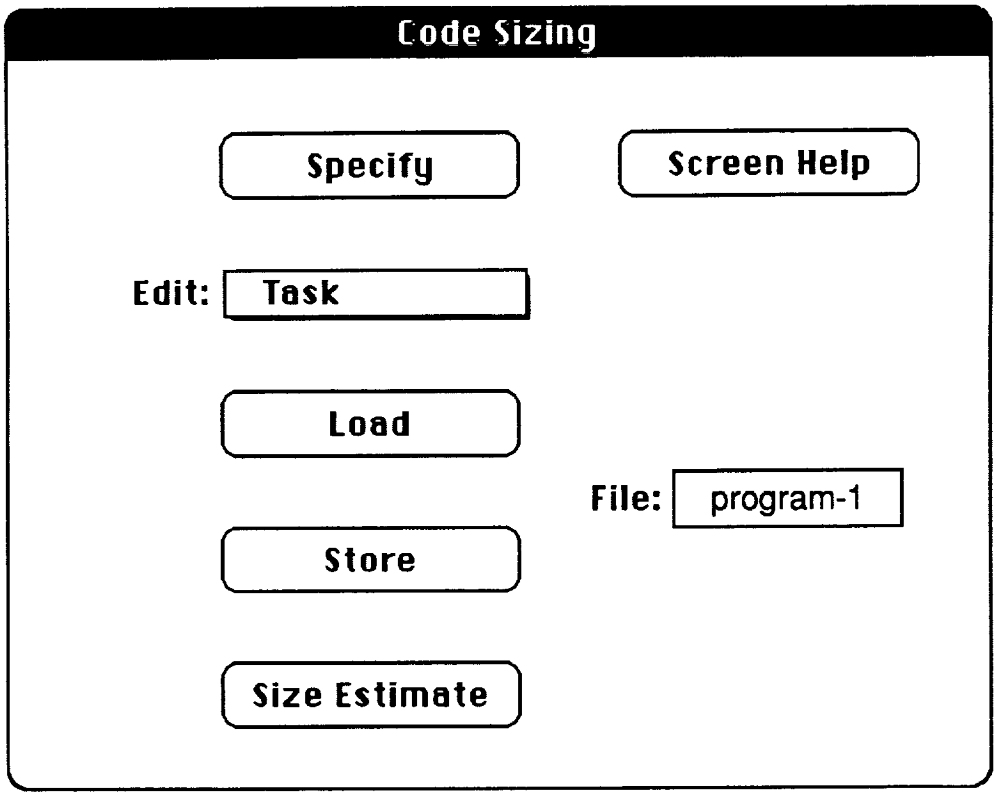


Figure 5.3
The Control Screen

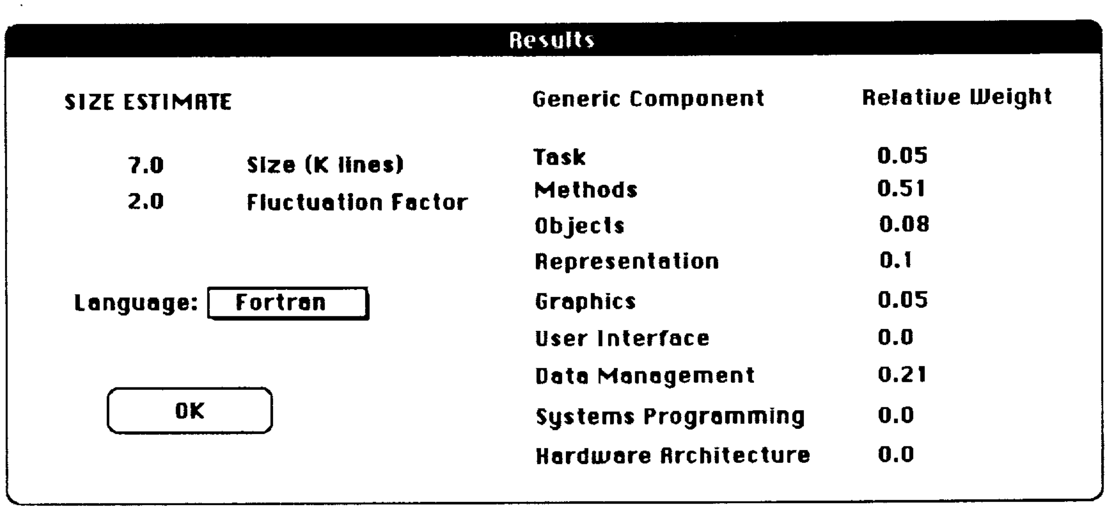


Figure 5.4
The Results Screen

The user interface contains the control screen, results screen, seven specification screens for entering the program's specifications, and help screens for each specification and for each of the other screens. The program's specifications are input in response to specific questions. These questions and the help information for each specification are meant to refine the definitions of the specifications. The specification and help screens are described in the next two subsections.

5.1.1 Specification Screens

Figure 5.5 shows the first of the specification screens which is used to enter information about the program's task. The screen contains a set of questions on the left, each of which is opposite a scroll bar. Each question corresponds to a single specification. The answer to "Is it an Engineering Program ?" determines the value of the specification represented by the *engineering-program* slot of the *task* frame, and the answer to "Does it do Structural Engineering ?" determines the value of the *structural-engineering* slot of the *engineering-program* frame. The latter question is indented by one column relative to the former because it is a direct descendent as shown in the specification tree structure in Section 4. The scroll bar is used to enter the certainty that the answer to the left is *yes*. The value is determined by the position of the slide. If the slide is all the way to the left the specification is definitely not true, and if the slide is all the way to the right the specification value is taken as definitely true. All of the specifications on this screen are qualitative and all qualitative specifications are entered in this way. There is a *help* button to the left of each specification and one for the screen as a whole. When all of the specifications on the screen have been set to the desired value, the user presses the *OK* button and the screen will be removed.

Figure 5.6 contains the screen for solution *methods*. The first two lines correspond to quantitative specifications and, in addition to the other display objects, contain a button group which is used to select the desired quantitative bin, as described in the previous section. The slide determines the certainty of the bin selected in the button group. If the slide is all the way to the left, the choice is completely uncertain and all bins are equally likely. If the slide is all the way to the right, the choice is completely certain. The initial values for the specifications default so that all qualitative specifications are not true and all quantitative specifications are set to the minimum bin with the maximum uncertainty. The values of the specifications change as soon as the button groups or slides are moved, not when the *OK* button is pressed and the window disappears.

Figures 5.7 to 5.11 contain the rest of the specification screens and Tables 5.3 to 5.8 show the specification frames corresponding to each of the user's questions. The user's response to a given specification is not necessarily independent of the values of the other responses, as implied by the specification dependencies described in Section 4. There is a mechanism in the user interface to aid in maintaining consistency. For example, if the user changes the value of the *Engineering Program* specification from a value greater than zero to zero (ie, moves the scroll bar all the way to the left), and the specification for *Structural Engineering* is greater than zero, the user interface will automatically set it to zero. The mechanism is applied recursively so that any specifications dependent on *Structural Engineering* would also be set to zero. The user can override this mechanism by setting the dependent specification first. For example, if the certainties for *Engineering Program* and *Structural Engineering* are both zero, the user can increase the certainty for *Structural Engineering* without effecting the certainty of *Engineering Program*.

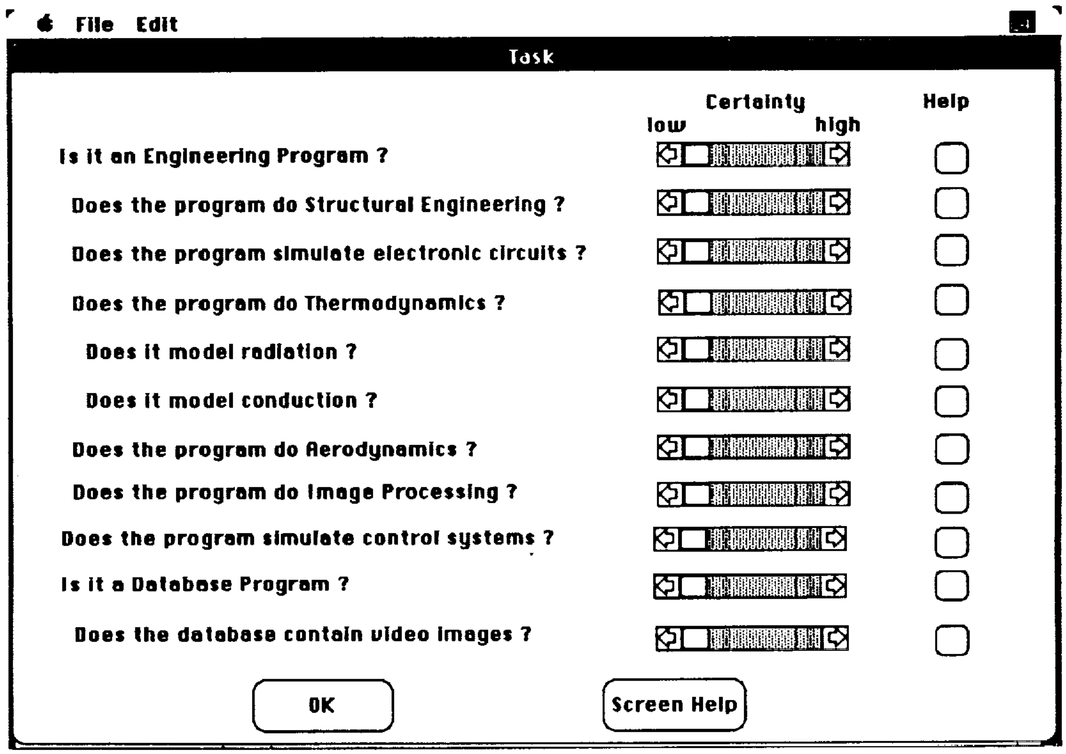


Figure 5.5
The Task Screen

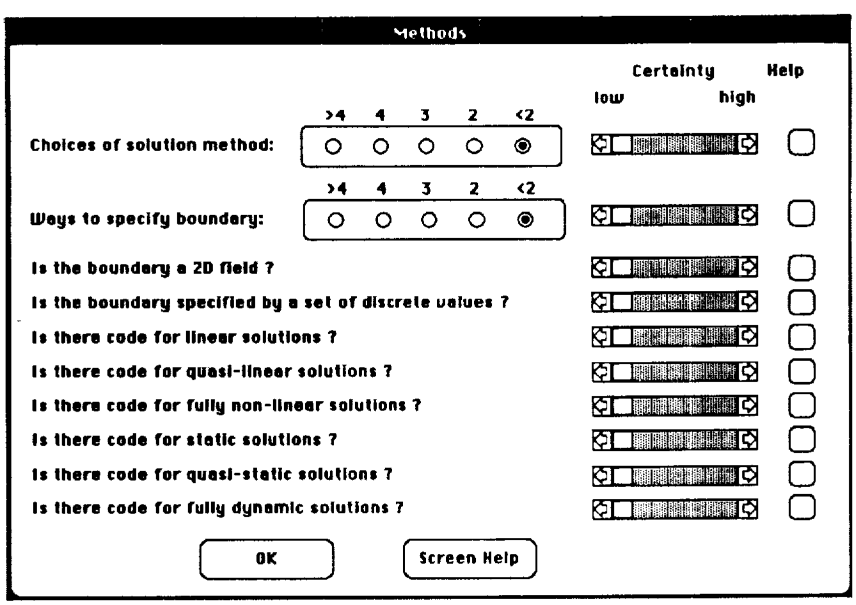


Figure 5.6
The Methods Screen

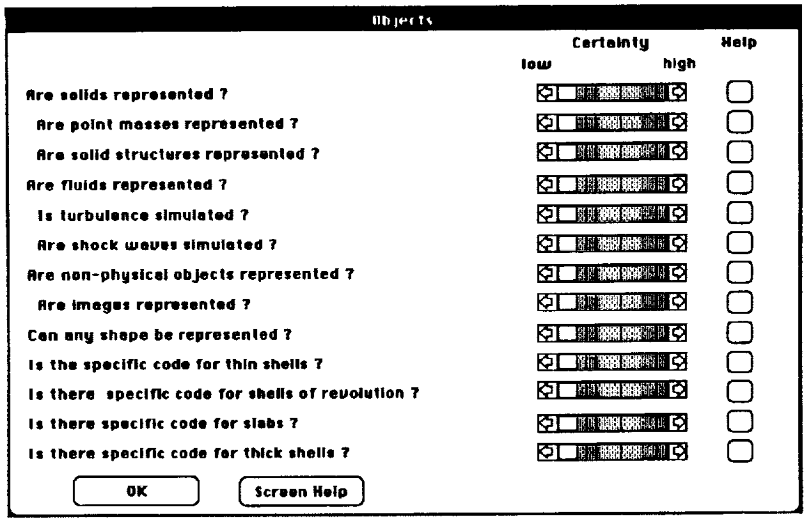


Figure 5.7
The Objects Screen

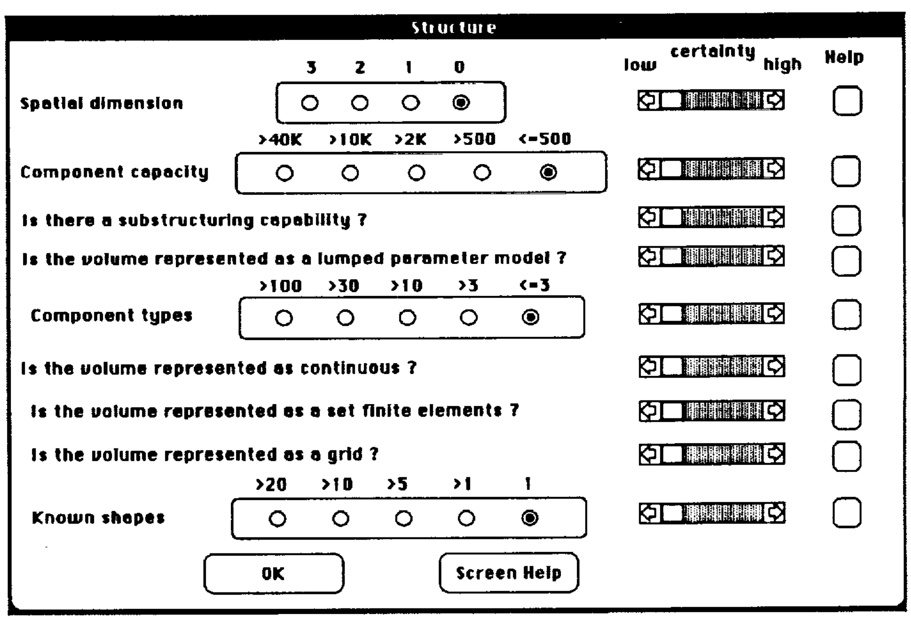


Figure 5.8
The Structure Screen

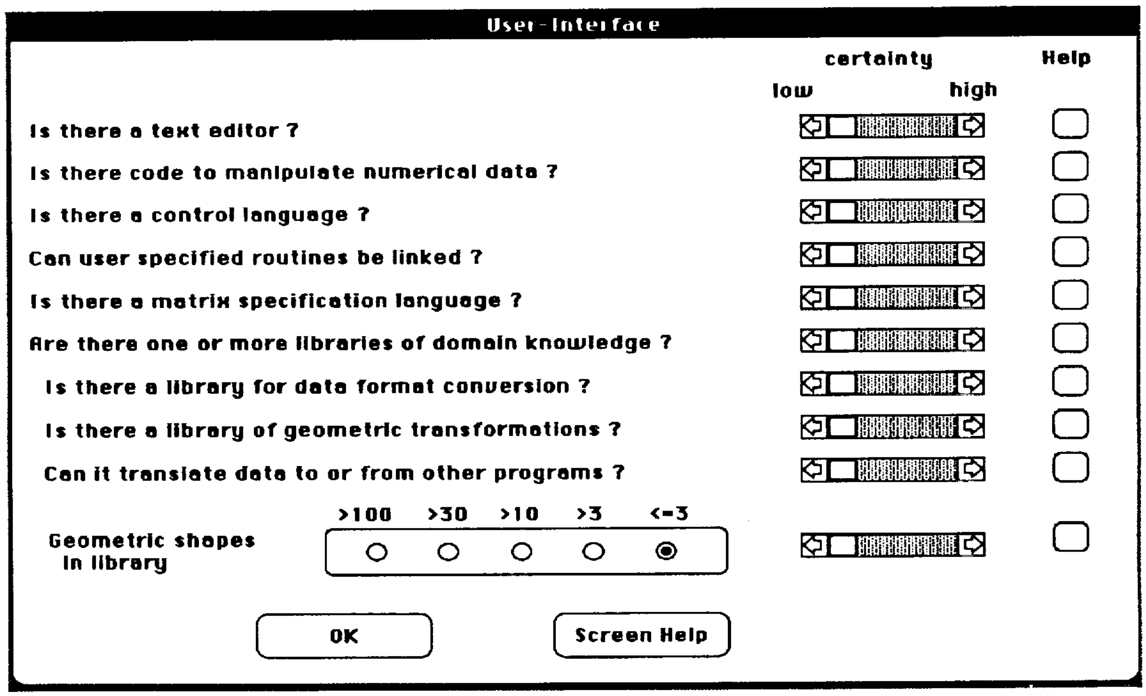


Figure 5.9
The User Interface Screen

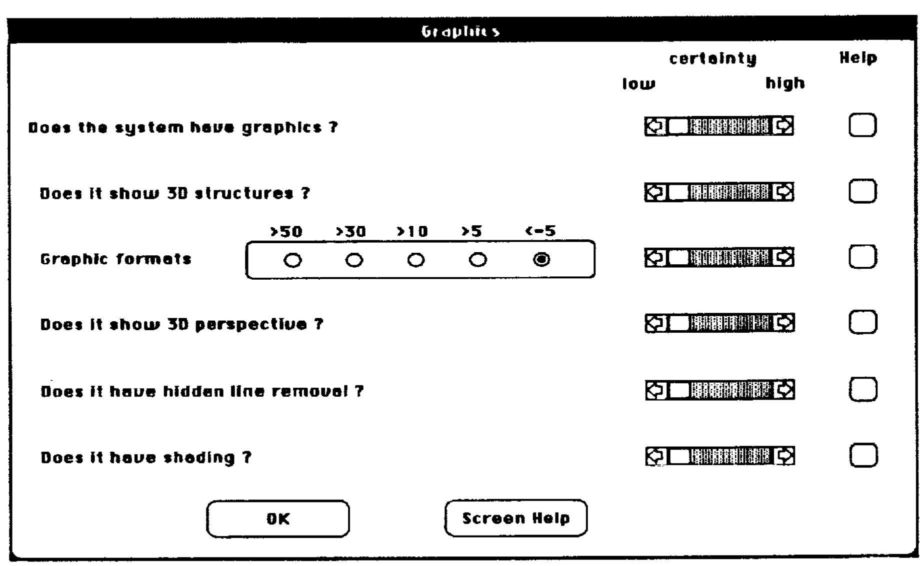


Figure 5.10
The Graphics Screen

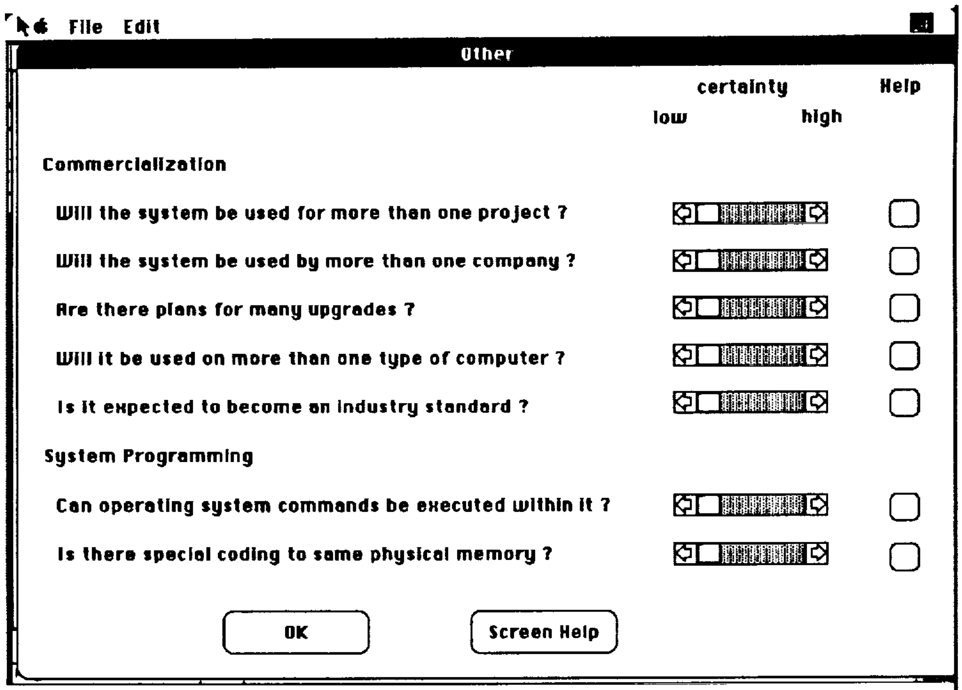


Figure 5.11
The Other Screen

Table 5.3
Specifications and Corresponding Queries for the Task Screen

Frame / Slot	User Question
task engineering-program	Is it an Engineering Program ?
engineering-program structural-engineering	Does the program do Structural Engineering ?
engineering-program thermodynamics	Does the program do Thermodynamics ?
thermodynamics radiation	Does it model radiation ?
thermodynamics conduction	Does it model conduction ?
engineering-program aerodynamics	Does the program do Aerodynamics ?
engineering-program image-processing	Does the program do Image Processing ?
task database-program	Is it a Database Program ?
database-program video-images	Does the database contain video images ?
engineering-program electronic-circuit	Does the program simulate electronic circuits ?

Table 5.4
Specifications and Corresponding Queries for the Methods Screen

Frame / Slot	User Question
methods number-redundant-choices	Choices of solution method:
boundary number-ways-to-specify	Ways to specify boundary:
boundary lumped-component-values	Is the boundary specified by a set of discrete values ?
boundary 2d-field-on-3d-surface	Is the boundary a 2D field ?
linearity fully-non-linear	Is there code for fully non-linear solutions ?
linearity limited-non-linear	Is there code for quasi-linear solutions ?
linearity linear	Is there code for linear solutions ?
time-dependence fully-dynamic	Is there code for fully dynamic solutions ?
time-dependence quasi-static	Is there code for quasi-static solutions ?
time-dependence static	Is there code for static solutions ?

Table 5.5
Specifications and Corresponding Queries for the Objects Screen

Frame / Slot	User Question
objects solids	Are solids represented ?
solids point-mass	Are point masses represented ?
solids structure	Are solid structures represented ?
objects fluids	Are fluids represented ?
fluids turbulence	Is turbulence simulated ?
fluids shock-waves	Are shock waves simulated ?
objects non-physical	Are non-physical objects represented ?
non-physical images	Are images represented ?
shape-limitations essentially-none	Can any shape be represented ?
shape-limitations thin-shells	Is the specific code for thin shells ?
shape-limitations shells-of-revol	Is there specific code for shells of revolution ?
shape-limitations slabs	Is there specific code for slabs ?
shape-limitations thick-shells	Is there specific code for thick shells ?

Table 5.6
Specifications and Corresponding Queries for the Structure Screen

Frame / Slot	User Question
structure spatial-dimension	Spatial Dimension
structure number-of-elements	Component capacity
structure substructuring-capability	Is there a substructuring capability ?
structure lumped-parameter	Is the volume represented as a lumped parameter model ?
lumped-parameter number-known-components	Component types
structure continuous-volume	Is the volume represented as continuous ?
continuous-volume finite-elements	Is the volume represented as a set of finite elements ?
continuous-volume grid	Is the volume represented as a grid ?
continuous-volume number-of-known-shapes	Known shapes

Table 5.7
Specifications and Corresponding Queries for the User Interface Screen

Frame / Slot	User Question
user-interface text-editor	Is there a text editor ?
user-interface data-manipulator	Is there code to manipulate numerical data ?
user-interface control-language	Is there a control language ?
user-interface user-specified-routines	Can user specified routines be linked ?
user-interface matrix-spec-language	Is there a matrix specification language ?
user-interface libraries	Are there one or more libraries of domain knowledge ?
libraries format-conversion	Is there a library for data format conversion ?
libraries geometric-conversion	Is there a library of geometric transformations ?
libraries translate-for-other-progs	Can it translate data to or from other programs ?
libraries number-of-stored-shapes	Geometric shapes in library

Table 5.8
Specifications and Corresponding Queries for the Graphics and Other Screens

Frame / Slot	User Question
graphics present	Does the system have graphics ?
graphics 3d-structure	Does it show 3D structures ?
graphics perspective	Does it show 3D perspective ?
graphics hidden-line-removal	Does it have hidden line removal ?
graphics shading	Does it have shading ?
graphics number-of-formats	Graphic formats
Commercialization many-projects	Will the system be used for more than one project ?
Commercialization many-companies	Will the system be used by more than one company ?
Commercialization many-upgrades	Are there plans for many upgrades ?
Commercialization many-computers	Will it be used on more than one type of computer ?
Commercialization industry-standard	Is it expected to become an industry standard ?
System Programming execute-commands	Can operating system commands be executed within it ?
System Programming code-to-save-memory	Is there special coding to same physical memory ?

5.1.2 Help Screens

A sample help screen is shown in Figure 5.12 and the text for each of the help messages in provided below.

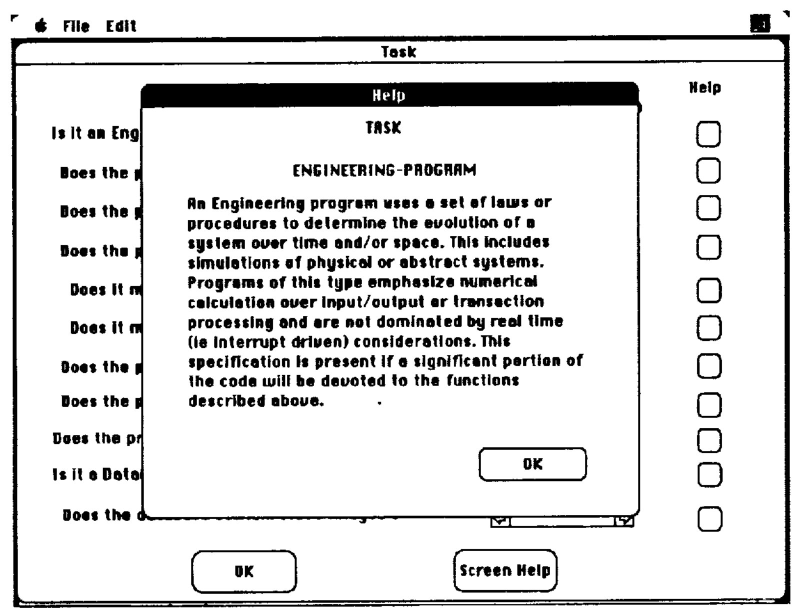


Figure 5.12
Sample Help Screen

screen code-sizing

This screen controls the Code Sizing Tool. To enter the program's specifications press the SPECIFY button. To edit the specifications, select one of the seven specification screens from the EDIT menu. To load a file of existing specifications type the file name in the text widow and press the LOAD button. To store the specifications in a file, enter the file name in the text window and press the STORE button. To obtain the code size prediction, press the SIZE ESTIMATE button. To exit the program, select QUIT from the FILE menu at the top of the screen.

screen task

The TASK screen contains specifications related to the type of applications performed by the program. The domain of the code sizing system is divided into Engineering Programs and Database Programs, and various disciplines and subdisciplines are included.

task engineering-program

An Engineering program uses a set of laws or procedures to determine the evolution of a system over time and/or space. This includes simulations of physical or abstract systems. Programs of this type emphasize numerical calculation over input/output or transaction processing and are not dominated by real time (ie interrupt driven) considerations. This specification is present if a significant portion of the code will be devoted to the functions described above.

task database-program

A database program is used to store and retrieve large amounts of data. It includes permanent data storage in a structured format and usually contains a query language and report writing facilities. Programs of this type tend to emphasize input/output over calculation. This specification is present if a significant portion of the code will be devoted to the functions described above.

engineering-program structural-engineering

An engineering program that simulated the stresses and strains in solid structures.

engineering-program electronic-circuit

An engineering program that simulated the currents and voltages in a network of electrical components.

engineering-program thermodynamics

An engineering program that represents the temperatures in matter.

engineering-program control-systems

An engineering program that simulates feedback systems, ie, systems that use the input from sensors and an internal model of some system to determine the signal to some actuator to influence the external environment towards some goal. This specification is for programs that simulate control systems not control systems themselves, which are real-time programs.

engineering-program aerodynamics

An engineering program that simulated the lift and drag of the atmosphere over a structure, generally an aircraft.

engineering-program image-processing

A program that manipulates images,ie, representations of a visual and/or spatial field. Such programs include but are not restricted to visual images. They can also include the data from other types of sensors or synthetic data from simulations. They are characterized by the facility to display and manipulate these images.

thermodynamics radiation

A program that simulates the propagation of thermal energy through the emission and absorption of electromagnetic radiation by matter.

thermodynamics conduction

A program that simulates the propagation of thermal energy through collisions between adjacent atoms or molecules.

screen methods

This screen contains specifications about the solution methods implemented by the program. It includes the level of redundancy, the level of approximation, and the specification of the boundaries for the program.

methods number-redundant-choices

This is a measure of the level of redundancy in a program. The number of ways in which the program's task can be accomplished. For example, the number of solution methods the user has to select from. If there is more than a single major task, use the number for the task with the highest level of redundancy.

time-dependence static

This specification is true is there is code devoted to determining the static solution (ie, constant over time) for a simulation.

time-dependence quasi-static

This specification is true is there is code devoted to determining the solution for a simulation which is based on the assumption that conditions change slowly over time.

time-dependence fully-dynamic

This specification is true if there is code devoted to determining the solution for a simulation in which conditions can change rapidly over time.

linearity linear

This specification is true if there is code devoted to determine linear solutions for the behavior of a system. Linear solutions have the property that a linear combination of two solutions is itself a solution.

linearity limited-non-linear

This specification is true if there is code devoted to determining quasi-linear solutions, ie, solution that deviate by a small amount from being linear. This would be true of perturbation methods, etc.

linearity fully-non-linear

This specification is true if there is code that can determine solutions without any restrictions on there linearity.

boundary number-ways-to-specify

This specification is for the number of ways to specify the input to a simulation. Inputs include values at the spatial and/or temporal extremes of the phase space for the simulation.

boundary 2d-field-on-3d-surface

This specification is true if there is input that can be represented as a two dimensional field on a three dimensional surface.

boundary lumped-component-values

The boundary conditions are specified as set of discrete values that are not directly connected to points within a continuous field. They would typically be input for a lumped parameter model.

screen objects

This screen contains specifications about the types of entities represented in the program. These entities are divided into physical objects such as solids and fluids (both gasses and liquids) and nonphysical objects such as images. Solids can be represented as point masses or as structures which contain component parts.

objects solids

This specification is true if solid objects are represented in the program.

objects fluids

This specification is true if fluids (liquids or gases) are represented in the program.

objects non-physical

This specification is true if there is a nonphysical system represented in the program, as would be the case in Operations Research, for example.

solids structure

This is true if solids are represented as structures with spatial extent, as opposed to point masses.

solids point-mass

This is true if solids are represented as point masses with no spatial extent or other continuous properties.

screen structure

This screen contains specifications about solid structures including their spatial dimension and the way they are represented. Structures can be represented as either a set of connected primitive components (as in a lumped parameter model) or as a continuous volume (as in grid or finite element models). The screen also contains specifications for the capacity and flexibility of these representations.

structure spatial-dimension

This specification takes the value of the dimension of space in the program. It has a value of 0 if space is not represented, or 1, 2, or 3 is space is represented in the system.

structure lumped-parameter

This is true if spatial structures are represented as a set of primitive components as opposed to a continuous volume.

structure continuous-volume

This is true if spatial structures are represented as a continuous volume as opposed to a set of primitive components.

structure number-of-elements

This component is used to represent the maximum possible number of elements that can be used in representing a structure in the system.

structure substructuring-capability

This specification is true if the user can specify a structure as a set of modular substructures.

continuous-volume finite-elements

This specifies finite element models, usually used in structural analysis programs. Values represent the field within the volumes of the elements.

continuous-volume grid

This specifies the use of a grid to represent a field over a continuous volume. Values represent the field at the grid points.

continuous-volume number-of-known-shapes

A measure of the number of basic or primitive shapes known to the program for representing continuous volumes. For finite element models, this corresponds to the number of element types. For grid models, this corresponds to the number of grid types.

shape-limitations essentially-none

This specification is true if there are no restrictions on the shape of the structures that are represented in the model, or at least, if the restrictions would only apply to pathological shapes not normally need in applying the program.

shape-limitations thin-shells

This specification is true if there is code that can only be applied to thin shells.

shape-limitations shells-of-revol

This specification is true if there is code that can only be applied to surfaces of revolution about an axis.

shape-limitations slabs

This specification is true if there is code that can only be applied to slabs.

shape-limitations thick-shells

This specification is true if there is code that can only be applied to thick shells.

fluids turbulence

This specification is true for simulations that represent properties of turbulent fluids.

fluids shock-waves

This specification is true for simulations that represent shock waves, as during the flight of supersonic aircraft.

screen user-interface

This screen contains the specifications for the user interface. It includes the control the user has over the operation of the program, the methods of data input, and the knowledge stored in user accessible libraries.

user-interface text-editor

This specification is true if the program provides the user with a text editor for creating input for the program.

user-interface data-manipulator

This specification is true for programs with the capability to have the user directly control various mathematical operations of numerical data as in time series analysis or image processing programs.

user-interface control-language

This specification is true for programs with enough flexibility and complexity so that the user needs a control language for operating the program. Such a language is similar to the command line interpreter in many computer operating systems.

user-interface user-specified-routines

This specification is true if the program allows the users to link their own subroutines to the program.

user-interface matrix-spec-language

This specification is true if the program provides a specialized language for matrix specification. This is a typical feature of large structural analysis programs.

user-interface libraries

This specification is true if the program contains libraries of stored knowledge about the application.

libraries number-of-stored-shapes

This specification represents the number of stored shapes in the user library, including geometric shapes, primitive structural components, and more complex, specialized substructures.

libraries format-conversion

This specification concerns the program's ability to translate files written in a number of formats, generally by other programs, for its own use.

libraries geometric-conversion

This specification concerns the storage of geometric knowledge such as coordinate transformations.

libraries translate-for-other-progs

This specification concerns the program's ability to create output in a number of formats for use by a variety of other programs or facilities.

screen graphics

This screen contains the specifications for the graphics produced by the program. Specifications determine the flexibility of the graphics and whether three dimensional objects are represented.

graphics present

This specification is true if there are graphics in the program including both screen and plotter/printer output.

graphics 3d-structure

This specification is true if the program creates views of three dimensional structures.

graphics number-of-formats

This specification represents the number of significantly different types of graphics produced by the system.

graphics perspective

This specification represents the use of perspective in views of the shapes in the system's graphics.

graphics hidden-line-removal

This specification represents the use of hidden line removal from the graphical display of three dimensional structures.

graphics shading

This specification represents the use of shading for the graphical displays of three dimensional structures.

screen other

This screen is used for specifications about the level of commercialization intended for the program and the access that the program has to the operation system.

commercialization many-projects

This specification concerns the level of commercialization of the software. It is true if the program is intended for many projects.

commercialization many-companies

This specification concerns the level of commercialization of the software. It is true if the program is intended for use in many different companies or other organizations.

commercialization industry-standard

This specification is true if the program is intended to become a widely used, industry standard for its application.

commercialization many-upgrades

This specification is true if the program is intended for continual upgrade and enhancement over a long period of time.

commercialization many-computers

This specification is true if the program is intended for use on many different computers.

lumped-parameter number-known-components

This is a measure of the number of primitive components available for use in a lumped parameter model.

database-program video-images

This is true if the database is designed to handle high resolution bit mapped images as would be produced by a video camera.

non-physical images

This is true if the contains data structures that are significant to its purpose that do not represent objects in the physical world, as would be the case in an Operations Research or Risk Analysis program.

systems-programming execute-commands

This specification is true if the program has a significant link to the computer's operating system for performing system commands.

systems-programming code-to-save-memory

This specification is true if the program contains code specifically optimized for saving physical memory.

5.2 Knowledge Acquisition System

This section contains a discussion of the procedures, files, and data structures that make up the Knowledge Acquisition System. An overview is shown in Figure 5.13. The knowledge sources were analyzed to determine a knowledge representation scheme and a database of existing software. The database, which was organized according to the knowledge representation scheme, was analyzed with a facility that can perform both linear and nonlinear regression to create a rule base of specification contributions implemented as a set of *if needed* routines, and a nonlinear sizing function, implemented as a neural net. The rule base and sizing function were combined with tools for controlling inference and for creating the user interface, to complete the implementation of the code sizing tool.

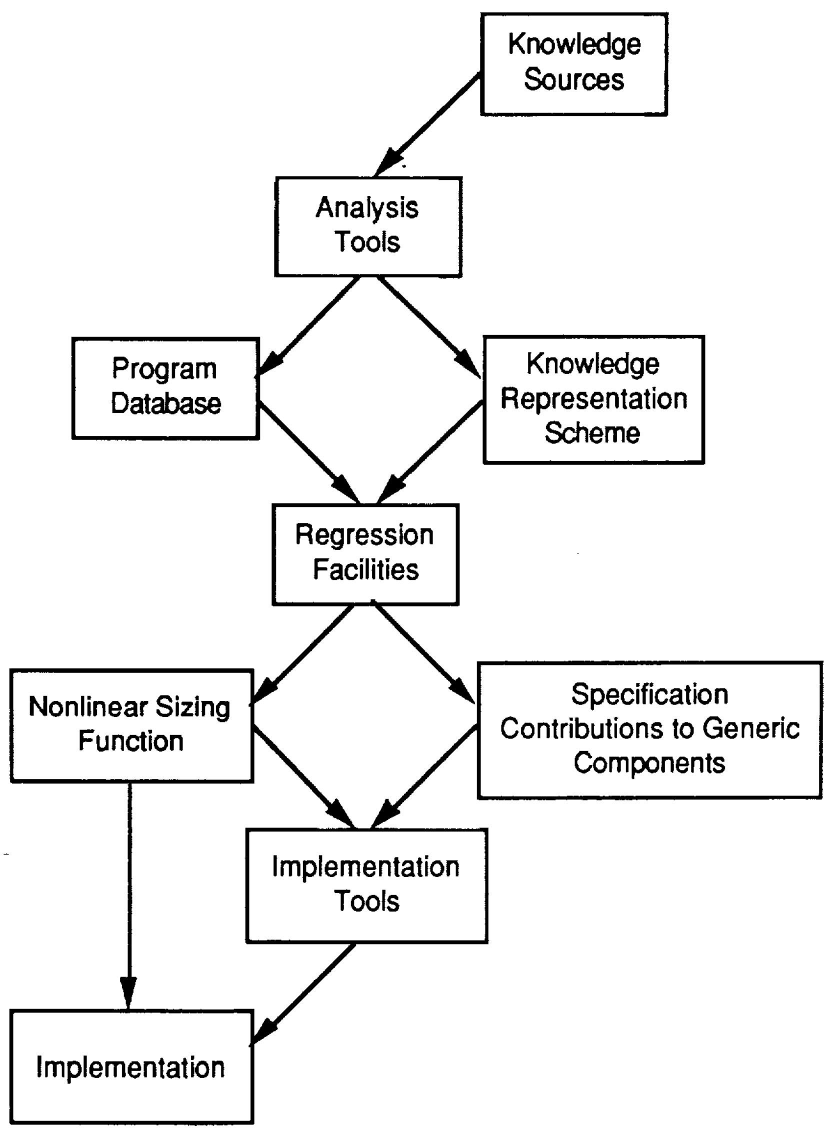


Figure 5.13
Operational Overview of the Knowledge Acquisition System

5.2.1 Code Analysis

Existing Fortran codes were analyzed to produce a structured set of subroutine frames as described in Section 2. This is done through the use of the MAKE-CALLING-TREE function in file FORTRAN-PARSER.LISP, described below:

(MAKE-CALLING-TREE file-name)

The parameter *file-name* is a string containing the name of the Fortran source code file. It will create one frame per routine (in memory) with the following structure:

```
(routine-name
        (SIZE (VALUE number-of-executable-lines))
        (COMMENT (VALUE number-of-comment-lines))
        (CHILDREN (VALUE child-1 ...... child-n)))
```

A list of the created frames is bound to the global TREELIST. The function will skip part of the source file if the user sets the global SKIP-LINES to a positive integer, the number of lines to skip. The function will print:

PARSING A routine-type STATEMENT

when it begins a new routine, where *routine-type* is PROGRAM, SUBROUTINE, FUNCTION, or BLOCK DATA. It will also print a line count every 100 lines and, if it can't parse a line, it will print an error message that includes the text of the problem line, and continue.

An analysis of the existing codes, documentation, and expert interviews lead to the knowledge representation scheme described in Section 4, including the representation of software in terms of structured specifications and generic components. This scheme was used to create two databases from the existing programs, one for specifications, and one for generic components and code sizes. It may be necessary to modify these files in the future in order to enhance the Code Sizing Tool by adding additional programs to the database or additional specifications or generic component categories. The files are therefore discussed in detail. There are two versions of each database, for convenience. One is formatted by MicroSoft Work™ for viewing, and the other is unformatted text for program I/O.

Table 5.9
Program Database File Names

	formatted	unformatted
Specifications	specification database	specification-base.lisp
Generic Components and Code Sizes	generic database	generic-base.lisp

The most convenient procedure for modifying the database was found to be:

1. Modify the formatted file using MicroSoft Word™
2. Save it with the *Save* command
3. Use the *Save As* command with a file format of *Text only with line breaks*, and the file name of the unformatted version.

A sample from SPECIFICATION DATABASE is shown in Table 5.10. The database is organized in sets, each of which contains the specifications for 2 to 4 programs. The first line of a set contains the program names and the rest of the set contains lines, each of which contains the frame and slot names for a specification and its value for each program in the set. The lines containing specification names and values are organized into groups under the headings of TASK, METHODS, OBJECTS & REPRESENTATIONS, USER INTERFACE, GRAPHICS, and OTHER, which are ignored when the specification database is read into memory. If a value of *n/a* is present, the default (lowest) value of the corresponding specification is assumed. For the future enhancements, a template for creating new sets is included at the end of the file. New types of specifications can be added by inserting them in the template. The older records would not necessarily have to be modified since the Knowledge Acquisition System software takes the union of all of the specifications in the database and assigns the default value to programs with absent specifications.

Table 5.10
Sample from SPECIFICATION DATABASE file

Program	nastran	trasys
TASK		
Task Engineering-Program	yes	yes
Engineering-Program Structural-engineering	yes	no
Engineering-Program electronic-circuit	no	no
Engineering-Program Thermodynamics	yes	yes
. . . .		
OTHER		
commercialization many-projects	yes	yes
commercialization many-companies	yes	yes
commercialization industry-standard	yes	yes
commercialization many-upgrades	yes	yes
commercialization many-computers	yes	yes

A sample from GENERIC DATABASE is shown in Table 5.11. The database is organized in sets, each of which contains the size (total lines of code) and generic components for a single program. The first (left most) column contains the program name, the second and third columns contain the generic component frame and slot names, and the fourth column contains the values. The fifth column is optional and contains the reasons for the assignment of the particular value. As in the previous case, a template is included at the end of the file for future maintenance.

Table 5.11
Sample from file GENERIC DATABASE

Program		Entity	Attribute	Value	Reason
panair	size	total	289	Cosmic catalogue	
	task	complexity	very-high	full sub and supersonic	
	methods	complexity	very-high	Higher Order Panel Method	
		generality	very-high	linear potential flow theory	
		accuracy	very-high	Higher order panel method	
		efficiency	high	****unknown****	
		redundancy	very-low	****unknown****	
	objects	complexity	very-high	arbitrary structures	
		generality	very-high	arbitrary structures	
	representation	complexity	high	set of surface grids	
		capacity	high	assumed from context	
		knowledge	high	good set of input options	
	graphics	complexity	very-low	not mentioned	
	user-interface	complexity	high	user has extensive control	
	data-management	complexity	high	internal database	

Unlike the case for specifications, if the types of generic components are modified, one of the routines, SET-UP-FEATURES in file DATABASE-PARSER.LISP, must be modified. These routines are explained in the next section. The SET-UP-FEATURES routine is shown in Table 5.12. It initializes a set of structures frames for the generic components, and the modifications are straight forward: add, delete, or modify the corresponding generic frame or slot names.

5.2.2 Database initialization procedures

These procedures are contained in file DATABASE-PARSER.LISP and are used to read the program database files and create corresponding frame structures in memory. The procedures and data structures are discussed in this section. The following routine is used to initialize the specification database:

(MAKE-SPECIFICATION-DATABASE specification-file initialization-flag)

The parameter "specification-file" is a string containing the name of the specification database file. It is optional and will default to SPECIFICATION-BASE.LISP. The flag "initialization-flag" will cause the previous specification criteria to be deleted if it is set to *t*. It is optional and defaults to *t*. If there is more than one file of specifications to be initialized, MAKE-SPECIFICATION-DATABASE should be run once for each file with the initialization flag set to *t* for the first file and *nil* for all subsequent files. The programs are represented in the specification database in groups, as previously described. For each group of programs, the following screen messages, containing the programs and specifications in the group, will appear:

PROGRAMS: $(prog_1 \ldots\ldots prog_n)$

$(\text{spec-frame}_1\ \text{slot}_1\ \text{val}_{111} \ldots\ldots \text{val}_{11n})$

$\vdots$

$(\text{spec-frame}_k\ \text{slot}_m\ \text{val}_{km1} \ldots\ldots \text{val}_{kmn})$

Table 5.12
Code for Function SET-UP-FEATURES

```
(defun set-up-features ()
  (set-frame-list
   '((program
              (a-kind-of (value thing)))
     (design-criteria
              (a-kind-of (value thing))
              (children (value task methods objects representation
                        graphics user-interface data-management
                        sys-programming hard-architecture)))
    (task
              (a-kind-of  (value design-criteria))
              (complexity))
    (methods
              (a-kind-of  (value design-criteria))
              (complexity)
              (generality)
              (accuracy   (default medium))
              (efficiency (default medium))
              (redundancy (default very-low)))
    (objects
              (a-kind-of  (value design-criteria))
              (complexity)
              (generality))
     (representation
              (a-kind-of  (value design-criteria))
              (complexity (default medium))
              (capacity   (default medium))
              (knowledge))
    (graphics
              (a-kind-of  (value design-criteria))
              (complexity (default very-low)))
     (user-interface
              (a-kind-of  (value design-criteria))
              (complexity))
     (data-management
              (a-kind-of  (value design-criteria))
              (complexity (default very-low)))
     (sys-programming
              (a-kind-of  (value design-criteria))
              (complexity (default very-low)))
     (hard-architecture
              (a-kind-of  (value design-criteria))
              (complexity (default very-low)))
    )))
```

The run will create frames that contain the specifications for each program in the database and some global frames that contain the specified programs and the specification criteria:

```
(PROGRAM (SPECIFIED (VALUE prog1.........progn)))

(SPECIFICATION
    CRITERIA (VALUE spec-frame1...........spec-framek))
    (spec-frame1 (VALUE spec-slot11.......spec-slot1n))
            .
            .
            .
    (spec-framek (VALUE spec-slotk1.......spec-slotkm)))
```

For each program, the following frame is created:

```
(progi      (spec-frame1 (spec-slot11 val11) ....... (spec-slot1n val1n))
                         .
                         .
                         .
            (spec-framek (spec-slotk1 valk1) ....... (spec-slotkm valkm)))
```

To initialize the generic component database:

(INIT-DB generic-file set-up-flag)

The optional parameter "generic-file" is a string containing the name of the generic component database file, which defaults to GENERIC-BASE.LISP if omitted. The optional parameter "set-up-flag" will cause the generic component types to be initialized if set to *t*, which is its default value. If there is more than one file of generic components to be initialized, INIT-DB should be run once for each file with the initialization flag set to *t* for the first file and *nil* for all subsequent files. The name of each program in the generic component database will be printed on the screen. The following frames will be created:

```
(PROGRAM (A-KIND-OF (VALUE THING))
    (CHILDREN (VALUE prog1 ....... progm)))

(DESIGN-CRITERIA (A-KIND-OF (VALUE THING))
    (CHILDREN (VALUE generic-frame1......... generic-framen)))
        .
        .
        .
        .
        .
        .
(generic-framei (A-KIND-OF (VALUE parent))
    (NUMBER-OF-INSTANCES (VALUE m))
    (generic-sloti1 (DEFAULT defaulti1))
        .
        .
        .
    (generic-slotim (DEFAULT defaultim)))
        .
        .
```

.
.
.
.

```
(prog_k (A-KIND-OF (VALUE PROGRAM))
    (TOTAL-SIZE (VALUE size_i))
    (design-criteria (VALUE generic-frame_1k ............ generic-frame_nk)))
            .
            .
            .
            .
            .
            .

(generic-frame_ik (A-KIND-OF (VALUE generic-frame_i))
    (generic-slot_i1 (VALUE value_i1k))
            .
            .
            .
    (generic-slot_im (VALUE value_imk)))
```

Finally, if both the generic and specification database are to be used, the generic database (INIT-DB) must be run first. This can be done automatically by running:

(INITIALIZE-SYSTEM generic-file specification-file)

This routine will run INIT-DB with file "generic-file" as the database and the initialization flag set to *t*, and then run MAKE-SPECIFICATION-DATABASE with "specification-file" as the database and the initialization flag set to *t*. Both the "generic-file" and "specification-file" parameters are optional and default to GENERIC-BASE.LISP and SPECIFICATION-BASE.LISP respectively.

5.2.3 Regression Techniques

Once the database has been initialized, it can be used to determine the contributions that the specifications make to the generic components and to calibrate the function that determines code size based on the generic components. Both activities are done with regression techniques, linear regression to determine the specification contributions, and nonlinear regression to determine the sizing function. Both types of regression are done using the neural net facility developed for this project. Linear regression for the specification contributions is performed on a single layer network, referred to as the *generic net*, using the perceptron convergence algorithm. Nonlinear regression for the sizing function is performed on a two layer nonlinear network, referred to as the *sizing net*, using the back propagation algorithm. The two layers in the sizing net consist of a hidden layer and a single node output layer whose excitation determines the predicted code size according to the following equation:

$$S = (1000)^{e}, \qquad (5\text{-}1)$$

where S is the code size in thousands of lines and e is the excitation of the output node. The system has the capability of predicting sizes between a thousand and a million lines, as shown by the equation.

A number of functions and data structures have been created to allow the developer to implement these procedures. In the descriptions that follow, the inputs to the neural nets are considered as an additional layer. The code is contained in three files: PERCEP2.LISP, SIZE-NET.LISP, and GENERIC-NET.LISP. The file PERCEP2 contains basic code for setting up and calibrating linear and nonlinear neural nets. The network and its parameters are contained in global variables:

Table 5.13
Global Variables for file PERCEP2.LISP

variable	initial value	definition
alpha	0.9	The momentum coefficient (for nonlinear nets)
eta	0.25	Learning rate
layers	nil	A list of the nodes in each layer starting at input
nlayer	0	The number of layers, including input and output
ngap	0	The number of gaps between layers (nlayer-1)

The operationally important routines in PERCEPT2 are described below:

(SET-UP-NET LAYER-SIZES)

Set up a neural net with one row for each element in LAYER-SIZES. Each element in LAYER-SIZES is the number of nodes in the corresponding layer. The first element is the number of inputs and the last element is the number of outputs. Any elements in between represent hidden layers. If there are no hidden layers, it creates a linear perceptron. If there are one or more hidden layers, it creates a nonlinear net using the sigmoid nonlinearity. It initializes all weights and biases to random values between -0.2 and 0.2. The properties of nodes and links such as excitation, weights, and biases are retained on the property lists of global atoms that represent nodes and weights as shown below:

layers -> ($layer_0$, ..., $layer_m$)

$layer_i$ -> $node_{i0}$, ..., $node_{ik}$

(get 'to-weights $node_{ij}$) -> ($weight_{ijkl}$, ..., $weight_{ijuv}$)

(get 'from-weights $node_{ij}$) -> ($weight_{deij}$, ..., $weight_{fgij}$)

(get 'val $node_{ij}$) -> NUMBER, the node's bias

(get 'excite $node_{ij}$) -> NUMBER, the node's excitation

(get 'val $weight_{ijkl}$) -> NUMBER, the value of the weight from $node_{ij}$ to $node_{kl}$

(get 'old $weight_{ijkl}$) -> NUMBER, the value for the previous learning cycle

(get 'del $weight_{ijkl}$) -> NUMBER, the value of the increment in the weight

(FORWARD INPUT-LIST)

Given an INPUT-LIST of values for the input nodes, FORWARD will return a list of the output node excitations resulting from a forward propagation through the network.

(BACKWARD DESIRED-OUTPUTS)

Once a forward propagation has been run on a given set of inputs. BACKWARD will calculate the increments of the weights and biases based on the back propagation training algorithm for the DESIRED-OUTPUTS and put them on the network property lists.

(LEARN INPUTS OUTPUTS NCYCLE &optional PNUM)

Given a training set comprised of INPUTS, a set of inputs, and OUTPUTS, a set of corresponding desired outputs, LEARN will cycle through the training set NCYCLE times using the back propagation training algorithm. It will print out an error measure every PNUM cycles if this number is supplied.

(SAVESTATE)

This routine is used to save the network and all its parameters. It creates and returns a list structure with all the necessary information. It is generally used to store the data in a file as follows:

```
(WITH-OPEN-FILE (CH filename :DIRECTION :OUTPUT)
        (PPRINT (SAVESTATE) CH))
```

(RESTORESTATE)

This routine uses the list structure returned by (SAVESTATE) to create the neural network structures used by the neural net facility. It is generally used to retrieve the data from a file as follows:

```
(WITH-OPEN-FILE (CH filename)
        (RESTORESTATE (READ CH)))
```

The file SIZE-NET.LISP contains code for the nonlinear sizing net and some additional global parameters:

Table 5.14
Global Variables for file SIZE-NET.LISP

variable	initial value	definition
hidden-layer	t	not nil if there is a hidden layer
output-nodes	1	the number of nodes in the output layer
out-linear	nil	t is its linear (with no hidden layers)

Some of the operationally important functions are described below:

(SET-UP-SIZING &optional PLIST)

SET-UP-SIZING is used to set up a network whose inputs are based on the collective generic components in of programs in PLIST (taken from the Program Description Database). If this list is not supplied, the routine looks in slot CHILDREN of frame PROGRAM. It uses the global variables to determine the characteristics of the network. If HIDDEN-LAYER is t, it creates a nonlinear network with one hidden layer, otherwise, it creates a linear network with no hidden layers. If OUTPUT-NODES is set to 1, it creates a single output node whose excitation level determines predicted code size. If OUTPUT-NODES is greater than one, it assumes that each output node signifies a range of code sizes and the output node with the greatest excitation determines the predicted code size. The routine returns a list containing the inputs (generic software component values) and desired outputs (actual code sizes) of the programs in PLIST.

(TEST-PROG-BASE &optional PLIST TEST-LIST)

TEST-PROG-BASE takes list of programs PLIST (which defaults to the value in slot CHILDREN of frame PROGRAM if not supplied) and TEST-LIST, a subset of PLIST (which defaults to PLIST if not supplied), and cycles through the following:

For each program in TEST-LIST
- temporarily remove the program from PLIST
- use SET-UP-SIZING to create a network based on the reduced list
- use LEARN to calibrate the network based on the reduced list
- use FORWARD to predict the size of the removed program
- record the predicted and actual sizes for the program

The file GENERIC-NET.LISP contains the code for the linear generic net and the following global variables:

Table 5.15
Global Variables for file GENERIC-NET.LISP

variable	initial value	definition
max-iter	200	Maximum backprop interations
max-err	.02	Maximum error for nets

These parameters limit the number of iterations of the calibration algorithm when either the maximum number of iterations have been reached or when the error (discrepancy between actual and desired outputs) is less that the max-err parameter, by causing the the algorithm to terminate. The SET-UP-GENERIC function is used to initialize the generic net:

(SET-UP-GENERIC &optional PLIST)

SET-UP-GENERIC is used to set up a network whose inputs are based on the collective specifications of the programs in PLIST (taken from the specification database). If this list is not supplied, the routine looks in slot SPECIFIED of frame PROGRAM. It uses the global variables to determine the characteristics of the network. If HIDDEN-LAYER is t, it creates a nonlinear network with one hidden layer, otherwise, it creates a linear network with no hidden layers. The routine returns a list containing the inputs (software specification values) and desired outputs (generic component values) of the programs in PLIST.

The TEST-SIZING-SYSTEM functions test the program database on both the generic and sizing nets:

(TEST-SIZING-SYSTEM)

TEST-SIZING-SYSTEM tests the generic net on programs in the specification database and the sizing net on programs in the generic database. It cycles through the following procedure:

For each program in the specification database
temporarily remove the program from the calibration list
use SET-UP-GENERIC to create a network based on the reduced list
use LEARN to calibrate the network based on the reduced list
use FORWARD to predict the generic components for the program
record the predicted and actual generic components the program

For each program in the generic database
temporarily remove the program from the calibration list
use SET-UP-SIZING to create a network based on the reduced list
use LEARN to calibrate the network based on the reduced list
use FORWARD to predict the size of the removed program
record the predicted and actual sizes for the program

The table above (5.15) refers to the convergence error of the regression. It is defined as the root mean square difference between the actual and desired output vectors:

$$e = \frac{1}{n}\sqrt{\sum_{i}^{n}\sum_{j}^{m}(d_{ij}-a_{ij})^2} \qquad (5\text{-}2)$$

where e is the convergence error, d_{ij} and a_{ij} are the j^{th} components of the i^{th} actual and desired output vectors, respectively, m is the number of components in each output vector and n is the number of the output vectors in the set. This is implemented by the function VECTOR-RMS:

(VECTOR-RMS VECTOR-LIST)

The parameter vector has the following structure:

$(((d_{11} \;.....\; d_{1m}) \;...........\; (d_{n1} \;.....\; d_{nm}))\;((a_{11} \;.....\; a_{1m}) \;...........\; (a_{n1} \;.....\; a_{nm})))$

with the individual components as defined above and the function will return the convergence error as defined above.

5.2.4 Testing and Calibration

The system was originally calibrated and tested with the two test programs described above (TEST-PROG-BASE and TEST-SIZING-SYSTEM). Eventually, however, we created more flexible and convenient tools for carrying out these tests, which are stored in file NEW-TEST.LISP. The basic approach didn't change, and the definitions of data structures and results in this section also apply to the previously described tests.

There are two procedures for testing and calibration. The generic net determines the contributions of program specifications to generic software components, and the sizing net determines the code size based on the generic components. The generic net can be calibrated or tested by creating a *test set* of examples from programs that are in both the specification and generic databases. The sizing net can be calibrated or tested by creating training sets from the generic components and code sizes, both of which are in the generic database. Calibration is done by using all available examples to create a training set. Testing is done by creating a different training set for each program by temporarily removing that program from the training set, calibrating the network with it, and using the network to make a prediction for the program that was removed:

For each program in the database
    Remove that program from the list
    Apply the calibration procedures to the rest of the programs
    Predict the size of the program that was removed from the list
    Compare it to the actual size
Use the comparisons to derive bias and fluctuation factors

The alternative to this procedure would be to retain separate test and calibration sets. The procedure we used has the advantage that each test has the best possible statistics. The disadvantage is that it is more time consuming than the alternative since the system is recalibrated for each test case. The small sample size was the reason for our choice. We define *calibration components* as the generic components predicted by the calibration of the generic net, the *test components* as the generic components predicted by the generic nets resulting from the test of each program, and the *actual components* as the components from the generic database. We also define *calibration sizing* as the results of the calibrated sizing net and *test sizing* as the sizes from the sizing nets resulting from the test of each program. There are therefore three types of generic components that can be input to two types of sizing nets resulting in six types of predicted sizes:

Table 5.16

Types of Code Size Estimation for Calibration and Testing

Size Estimate	Generic Components	Sizing Function
1	Actual	Calibration
2	Actual	Test
3	Calibration	Calibration
4	Calibration	Test
5	Test	Calibration
6	Test	Test

Each of these types of size predictions can be compared to the actual sizes to determine results in the form of bias and fluctuation factors. Only the last (test-sizing with test-components) is a valid prediction of how the system will perform, but the others can be used to help determine where possible inaccuracies come from.

The previous paragraphs described how the results are defined. This paragraph described how they are structured. The global results are stored in an instance of the *test-system* frame:

```
(TEST-SYSTEMn (A-KIND-OF (VALUE TEST-SYSTEM))
        (PROGRAMS (VALUE prog1 ....... progn))
        (RANDOM-SIZING (VALUE (BIAS-FACTOR b FLUCTUATION-FACTOR f)))
        (TEST-SET (VALUE p1 ....... pn))
        (TESTED (VALUE p1 ....... pk))
        (FINAL-GENERIC-CALIBRATION (VALUE (ITER iter ERR err)))
        (FINAL-SIZING-CALIBRATION (VALUE (ITER iter' ERR err'))))
```

The *programs* slot contains the programs in the database, the *test-set* slot contains the ones that were used in the test, which are added to the *tested* slot as they are done. The *final-generic-calibration* and *final-sizing-calibration* slots contain the number of iterations and convergence error for the calibration (as opposed to testing) of the generic and sizing nets respectively. The network parameters for the two nets are completely independent, as shown. Finally, the *random-sizing* slot shows the results, in terms of bias and fluctuation factors, that would be calculated by predicting a program's size by randomly selecting a size from the database. This is done as a check against the predicted results, to show that the knowledge base is significantly better at predicting sizes than a random selection.

There is a frame for each program in the test set, and these frames contain the information local to the individual programs. This includes the specifications and generic components that were added when the databases were initialized, and the test parameters and results for each program:

```
(prog_k (A-KIND-OF (VALUE PROGRAM))
       (TOTAL-SIZE (VALUE size_k))
       (design-criteria (VALUE generic-frame_1k ............ generic-frame_nk)))
       (spec-frame_1 (spec-slot_11 val_11) ....... (spec-slot_1n val_1n))
              .
              .
              .
       (spec-frame_k (spec-slot_k1 val_k1) ....... (spec-slot_km val_km))
       (GENERIC-COMPONENTS
              (TEST r_1 ............ r_p)
              (CALIBRATED r_1 ............ r_p)
              (ACTUAL r_1 ............ r_p))
       (GENERIC-CALIBRATION (VALUE (ITERATIONS t_k ERROR f_k)))
       (SIZING-CALIBRATION (VALUE (ITERATIONS t_k' ERROR f_k')))
       (SIZING-LEARNING (VALUE (ETA η_k' ALPHA α_k MAX-ITER i_k MAX-ERR e_k)))
       (GENERIC-LEARNING (VALUE (ETA η_k ALPHA α_k MAX-ITER i_k MAX-ERR e_k)))
       (TEST-SIZING (TEST-COMPONENTS s_k4) (ACTUAL-COMPONENTS s_k5))
       (CALIBRATION-SIZING (ACTUAL-COMPONENTS s_k1)
              (TEST-COMPONENTS s_k2)
              (CALIBRATION-COMPONENTS s_k3)))
```

There are slots for the three types of generic components and the resulting six types of predicted sizes. The network parameters for the test of $prog_k$ are included for both the generic and sizing nets. These include the MAX-ITER and MAX-ERR parameters for automatic termination of the learning procedure. Since the interation procedure can also be terminated manually, the actual number of iterations and convergence error are also recorded in the *sizing-learning* and *generic-learning* slots, which also contain the network learning parameters for the sizing and generic nets respectively. Only the nonlinear sizing net uses the α (momentum) parameter.

The test functions in NEW-TEST were enhanced in response to a number of needs that were encountered during the early testing and calibration. As described in the preceding paragraphs, the network data are stored in global data structures. This has a disadvantage in that information for both the generic and sizing nets cannot be stored in memory at the same time. Earlier test procedures went through the program database for the sizing network before storing the results and creating the generic network to determine specification contributions. It was desirable to have the capability to follow one program at a time through the whole procedure during the testing and calibration of the generic net. This allows the size prediction results to be displayed during the testing and calibration of the generic net, which is a more informative indication of performance than the convergence error. It was accomplished by creating a new global data structure for the sizing net:

slayers - A list of the nodes in each layer starting at input
snlayer - The number of layers, including input and output
sngap - The number of gaps between layers

These are identical to the corresponding structures in Table 5.13 with an *s* preceding the name. Functions were developed to load and operate on these structures. They are designed for use with an already existing sizing net and cannot create one from the database. This must be done as previously described. The function are described below:

(RESTORE-SIZING-NET file)

This loads the sizing network parameters stored in a file with the SAVESTATE function. The parameter *file* is the file name. It is optional and defaults to "calibrated-sizing-net".

(FORWARD-SIZE input-vector)

This function takes a set of inputs in list *input-vector*, propagates them through the sizing network stored in the data structures shown above, and returns a list of output node excitations.

It was also convenient to allow tests to be performed without parsing the specification database. In order to accomplish this, a global list of the software specifications were defined and bound to the symbol SPECS:

$$
\begin{array}{lcl}
((\text{spec-frame}_1\ \text{spec-slot}_{11}) & \ldots\ldots. & (\text{spec-frame}_1\ \text{spec-slot}_{1n}) \\
\quad\vdots & & \quad\vdots \\
(\text{spec-frame}_k\ \text{spec-slot}_{k1}) & \ldots\ldots. & (\text{spec-frame}_k\ \text{spec-slot}_{km}))
\end{array}
$$

This global structure was convenient because the specification list was relatively stable. The structure must be updated, however, when the types of software specifications change.

Experience with the system revealed that a single set of network parameters would not work in all cases. Some tests converged quickly, some diverged until the "learning" parameters (a and h) were reduced, and some converged very slowly until the rates were increased. The testing and calibration tools were therefore enhanced so that the predicted results were displayed and the data structures were fully updated on each iteration. This allowed the run to be terminated when convergence was achieved or restarted after automatic termination by the MAX-ITER or MAX-ERR parameters. The functions for performing the tests are described below:

(SAVE-CURRENT-RESULTS)

Saves the current instance of the *test-system* frame and the program frames that contain the test and calibration results, in file "test-system". The highest instance of the *test-system* frame will be saved along with the program frames listed in its *test-set* slot.

(GET-RESULTS test-file)

Retrieves the results stored with the SAVE-CURRENT-RESULTS function from the file named *test-file*, which is an optional argument that defaults to "test-system".

(INITIALIZE-GENERIC-TEST sizing-file)

Initialize the system for testing or calibrating the generic net. The argument *sizing-file* is optional and defaults to "calibrated-sizing-net". The sizing net is also initialized so that size predictions can be determined. If the default sizing network is used, the sizing prediction is flagged as *calibration-sizing*, otherwise it is flagged as *test-sizing*.

(GENERIC-TEST prog)

Perform a test of the generic net for program *prog* and add it to the *tested* slot of the current instance of the *test-system* frame. One line of text will be displayed per iteration showing the program name, iteration number, convergence error, actual size, and predicted size:

prog ITER iter ERR err SIZE: ACT actual-size PRED predicted-size

(FINAL-GENERIC-CALIBRATION)

Perform a calibration of the generic net using all of the programs in the *test-set* slot. The following text will be displayed after each interation showing the iteration number, convergence error, bias factor, and fluctuation factor:

ITERATION iter ERROR err BIAS bias FLUCTUATION fluct

(INITIALIZE-SIZING-NET)

Set up the system for testing and calibrating the sizing net.

(SIZING-TEST prog)

Perform a test of the sizing net for program *prog* and add it to the *tested* slot of the current instance of the *test-system* frame. One line of text will be displayed per iteration showing the program name, iteration number, convergence error, actual size, and predicted size:

prog ITER iter ERR err SIZE: ACT actual-size PRED predicted-size

(FINAL-SIZING-CALIBRATION)

Perform a calibration of the sizing net using the programs form the *test-set* slot of the current instance of the *test-system* frame. The following text will be displayed after each interation showing the iteration number, convergence error, bias factor, and fluctuation factor:

ITERATION iter ERROR err BIAS bias FLUCTUATION fluct

(DISPLAY-RESULTS program-list sizing-type generic-type no-print-flag)

This function displays the results of calibrations or tests on either network. All of the parameters are optional. The argument *prog-list* is a list of programs to display. It defaults to the *test-set* sorted in order of prediction error determined as the absolute value of the logarithm of the ratio of predicted to actual size. The *sizing-type* and *generic-type* arguments are used to select one of the six types of code size predictions for display. *Sizing-type* can take a value of *calibration-sizing*, or *test-sizing* and defaults to *test-sizing* and *generic-type* can take a value of *actual-components*, *calibration-components*, or *test-components*, and defaults to *test-components*. A table of the results will be printed out if the *no-print-flag* is not set to *t* (It defaults to *nil*). The table contains the name, predicted code size, actual code size, and their ratio for each program. At the bottom of the table, the bias are fluctuation factors are provided:

PROGRAM	PREDICTED	ACTUAL	RATIO
$prog_1$	p_1	a_1	r_1
.	.	.	.
.	.	.	.
.	.	.	.
$prog_n$	p_n	a_n	r_n

BIAS FACTOR = bias FLUCTUATION FACTOR = fluctuation

The function returns a list containing the bias and fluctuation factors in the following format:

(BIAS bias FLUCTUATION fluct)

5.2.5 Creating the Implementation

This section reviews the functions and data structures used in creating the production system. It includes descriptions of operations on some of the files that are actually part of the production system. The file CREATE-IMPLEMENTATION contains functions that transform the generic net into a set of rules for determining the contribution of each of the software specifications. The following production system files will also be discussed:

IMPLEMENTATION - contains the rule base produced by CREATE-IMPLEMENTATION.

IMPLEMENTATION-CONTROL - contains functions that control the execution of the production system.

SCREEN - contains functions that control the user interface for the production system.

SCREEN-INTERFACE, RESULTS-INTERFACE, HELP-INTERFACE, TASK-INTERFACE, METHODS-INTERFACE, OBJECTS-INTERFACE, STRUCTURE-INTERFACE, USER-INTERFACE-INTERFACE, GRAPHICS-INTERFACE, and OTHER-INTERFACE - are created with the Action™ software tool and directly control each of the screens in the user interface for the production system.

The file CREATE-IMPLEMENTATION contains functions that take the coefficients from the generic net, which determine the contributions of program specifications to generic software components, and use them to create a code sizing rule base in the form of IF-NEEDED routines and frame structures. These routines and data structures are implemented as LISP code and stored in file IMPLEMENTATION.LISP. This file can be compiled and is a major component of the production system. It is created by executing two functions:

(CREATE-IMPLEMENTATION)

The function returns a slit containing all of the source code for the IMPLEMENTATION.LISP file. This list is very long and should not be displayed on the screen. It is meant to serve as the argument to SAVE-IMPLEMENTATION. The source code is produced from the generic network which must be present in memory. The results will have the following structure:

```
((DEFUN INITIALIZE-FRAMES ()
   (SET-FRAME-LIST
   '((SIZING-CRITERIA
      (OUTPUT-NODES
        (VALUE (generic-frame1 slot11) ........ (generic-framen slotnk)))
      (CHILDREN (VALUE child1 ....... childm)))
     (child1 (SPECIFICATION-SLOTS (VALUE slot11 ...... slot1q))
      (GENERIC-SLOTS (VALUE slot1 q+1 ...... slot1j))
      (slot11 (IF-NEEDED routine11) (RANGE min11 .... max11)) ......
      (slot1j (IF-NEEDED routine1j) (RANGE min1j ..... max1j)))
           .
           .
           .
     (childm (SPECIFICATION-SLOTS (VALUE slotm1 ..... slotmr))
      (GENERIC-SLOTS (VALUE slotm r+1 ..... slotmh))
      (slotm1 (IF-NEEDED routinem1) (RANGE minm1 .... maxm1)) ......
      (slotmh (IF-NEEDED routinemh) (RANGE minmh ..... maxmh)))
           .
           .
           .
  (DEFUN routineuv (FRAME SLOT &AUX VAL)
   (SETQ VAL cuv)
   (IFF 'generic-frame1 'slot11 'vuv11 (INC VAL cuv11))
                  .
                  .
                  .
   (IFF 'generic-framen 'slotnk 'vuvnk (INC VAL cuvnk))
   (fput-list FRAME SLOT VAL))
           .
           .
           .
 )
```

This structure contains a routine called INITIALIZE-FRAMES that creates a knowledge structure with frames describing the specifications and generic components. The structure contains a *children* slot which contains the names used for the frames representing specifications, generic components, or both. There is a frame for each of the design criteria that contains *range* and *if-needed* routines for its slots. The if-needed routines for the specification slots query the user. The if-needed routines for the generic components are named according to the convention that, if the frame name is *fname* and the slot name is *sname*, the if-needed routine name is *fname-sname*. The structure also contains the code for the generic if-needed routines. The routine calculates a value for the generic slot by initializing a variable to the bias of the corresponding node of the generic net and adding increments that depend on the specifications and the weights that link the specifications and generic components in the generic net.

(SAVE-IMPLEMENTATION source-code-file source-code-list)

This routine creates a source code file using a knowledge structure like the one described above. Both arguments are optional. The *source-code-file* defaults to IMPLEMENTATION.LISP, and the function will execute the CREATE-IMPLEMENTATION routine and use the results for *source-code-list* if it is not supplied. It prints a message for each of the generic if-needed routines recorded in the file:

ROUTINE routine-name FOR (frame-name slot-name)

The IMPLEMENTATION-CONTROL file contains functions to control the execution of the production system including initialization of the system and the if-needed routine for predicting code size. The two major functions are described below:

(INITIALIZE-SIZING)

Sets up the internal knowledge structures for the production system. It executes three routines that each initialize a different part of the system. INITIALIZE-FRAMES, which is actually part of the IMPLEMENTATION.LISP file, creates the knowledge structures for the specifications and generic software components. INITIALIZE-SIZING-NET loads the sizing-net from file SIZING-NET-STATE.LISP, and INITIALIZE-SPECIFICATION-RULES set up the specification dependencies (see Section 4) from the a list structure bound to the symbol IMPLICATION, with the following form:

((implying-specification-slot$_1$ implied-specification-slot$_1$) ....
(implying-specification-slot$_n$ implied-specification-slot$_n$))

(PROGRAM-PREDICTED-SIZE frame slot)

This is the if-needed routine for the *predicted-size* slot of the *program* frame and is called upon to predict the code size. It does this by determining the specifications supplied through user input, calculating the generic components by applying the rule base in the IMPLEMENTATION file, and estimating the size by feeding the generic components into the sizing net.

The user interface is modular and resides in one file created directly by the developers, named SCREEN.LISP, and in nine files automatically created through the use of the Action™ user interface generator. The generated files are named SCREEN-INTERFACE.LISP, RESULTS-INTERFACE.LISP, HELP-INTERFACE.LISP, TASK-INTERFACE.LISP, METHODS-INTERFACE.LISP, OBJECTS-INTERFACE.LISP, STRUCTURE-INTERFACE.LISP, USER-INTERFACE-INTERFACE.LISP, GRAPHICS-INTERFACE.LISP, and OTHER-INTERFACE.LISP. Some important aspects of the user interface are discussed below with emphasis on the parts that may need to be modified if the types of specifications or generic components are changed. Some references will be made to the interface tool and the manual should be consulted for an explanation when necessary. Because the Action™ tool was used to create the user interface, it is object oriented rather than frame based. It is therefore described in terms of objects, classes, and methods rather than frames, user defined inheritance mechanisms, and attached routines. The functionality is similar. Whenever necessary, the objects and frames are linked together by having a method execute one of the frame primitives or by having one of the routines attached to a frame execute a method.

The SCREEN file contains the objects and methods that were defined or written by the user (ie, the developer). This includes all of the objects in Figure 5.14 except for Application and ApplicationEdit, which are part of the Action™ package. Since Application Edit is a SuperClass for all of the user defined objects, they can be edited with the package. When the system has been finalized and is ready for distribution, the user defined objects are moved directly under the Application object which fixes them so that they cannot be improperly modified.

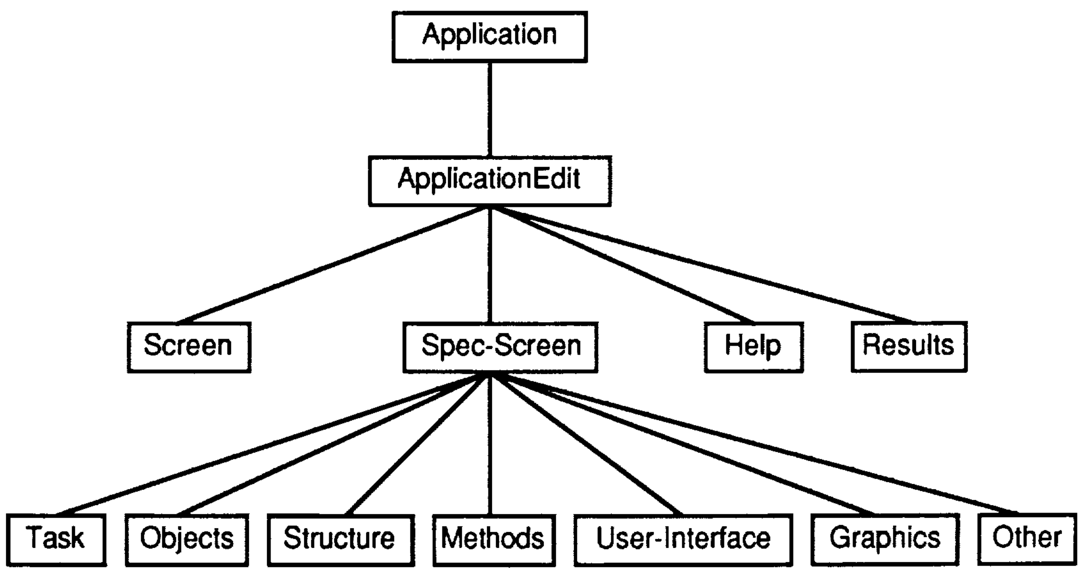


Figure 5.14
User Interface Objects

Each of the *leaf* nodes is the name of a display screen which is controlled by a file named according to the convention that a screen with the name *sname* is associated with a file named *sname-interface.lisp*. These files are created with the Action™ package. A sample extract from the TASK-INTERFACE file is shown in Figure 5.15. The specifications are divided among a set of seven screens. The objects associated with these screens are shown on the bottom row of the Figure 5.14. Each specification on a screen has a set of display objects from the user interface package associated with it including a TextObject containing the question for the specification, a ScrollBarObject for the certainty of the specification, and a ButtonObject for getting a *help* message on the specification. In addition, quantitative specifications have a ButtonGroupObject for selecting a quantitative value bin for the specification value (see Section 4). The frame-slot pair representing the specification is linked to the corresponding display objects so they are initialized to the correct values when the screen is displayed. There are also methods attached to the display objects so that the value of a corresponding frame-slot pair is changed when the position of the scroll bar or (when applicable) the button group selection is changed.

Figure 5.15

Extract from the Action™ Generated File
TASK-INTERFACE.LISP

```
(defexperinterface task (superclass spec-screen)
  (window
    (make-experobject 'exper-window :boundsrect
                 .
                 .
                 .
    )
  (menus
                 .      .
                 .
                 .
   )
  (experobjects
    (make-experobject 'scrollbarobject :boundsrect
                 .
                 .
                 .
    )
                 .
                 .
                 .
  ))
```

There is one help screen for each specification and one for each of the other screens in the user interface. The text for the screens in contained in a file HELP.LISP. As shown in Figure 5.16, the format for this file consists of a set of help messages each of which contains one line with the frame and slot names for the message, a blank line, a multi-line help message, and another blank line. The file is read when the user interface in initialized and each help message is added to the *help* facet of the corresponding frame and slot.

Figure 5.16
Extract from file HELP.LISP

.
.
.

screen task

The TASK screen contains specifications related to the type of applications performed by the program. The domain of the code sizing system is divided into Engineering Programs and database Programs, and various disciplines and subdisciplines are included.

task engineering-program

An Engineering program uses a set of laws or procedures to determine the evolution of a system over time and/or space. This includes simulations of physical or abstract systems. Programs of this type emphasize numerical calculation over input/output or transaction processing and are not dominated by real time (ie interrupt driven) considerations. This specification is present if a significant portion of the code will be devoted to the functions described above.

.
.
.

6.0 Systems Description

This section documents the system's structure and serves as a Programmers Manual. It, along with the listings, Operational Description, equipment manuals, and external software package manuals, will be used to maintain the system. This section contains a description of the hardware, a description of the delivery software and directory structure, an explanation of the notation used to document the software, and separate subsections with detailed documentation on the structure of the Production System, the Knowledge Acquisition System, and the Tools and Facilities developed as part of the project.

The hardware is based on a MicroExplorer system which includes a 68020 based Macintosh II with a special board from Texas Instruments that has a microprocessor designed with a LISP architecture intended for efficient symbolic processing. In addition to a large bank of physical memory, the MicroExplorer operating system has access to 100 MB of virtual memory from a page file residing on the external hard disk. When the MicroExplorer is running, the 68020 acts as a coprocessor for input and output, resulting in a highly efficient, graphics oriented system. The system is equipped with a full size tape drive, which was used for reading the software tapes from the COSMIC database, and a 300 dpi laser printer for high quality documentation. A system diagram is shown in Figure 6.1 and an equipment list in Table 6.2.

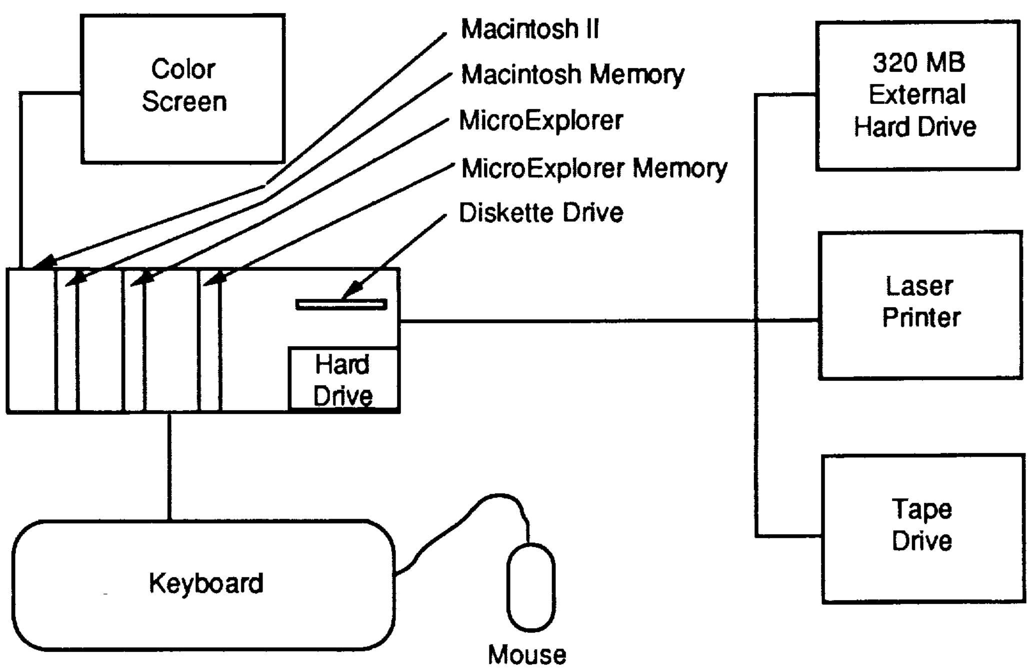


Figure 6.1
System Hardware

Figure 6.2 shows the structure of the delivered software and Tables 6.2 and 6.3 contain the definitions of the important folders (ie, directories) that make it up. We are using some of the Macintosh conventions and show an icon that looks like a folder to represent directories. The files developed at MCR are also included as a list of names enclosed in a rectangle under the appropriate folder. System software and general applications software are contained in the *system* and *utilities* folders, respectively. *HD* is the system disk, but a second copy of these folders are contained on *HD1* as backup. The files for the two major applications packages for the Code Sizing Tool, Action and the MicroExplorer system software, are contained in two separate directories. There is a backup copy of the MicroExplorer software on *HD1* that includes the (only copy of) a 100 MB page file for use by the virtual memory operation system. There is also a copy of an old version of the MicroExplorer software which was upgraded from version 5 to 6 during the course of the project. *HD1* contains source code from the COSMIC database and routine frame files resulting from parsing the source code. The *NASA* folder on *HD* contains the software and data developed at MCR in order to implement the Code Sizing Tool. The software needed to run the Code Sizing Tool (ie, the Production System) is included in this directory and consists of LISP and compiled LISP (machine) code. The source code and data files for the Production System and Knowledge Acquisition System are each contained in a subdirectory of *NASA*.

Table 6.1
Equipment List

Macintosh II	68020 based microcomputer used to house and assist the MicroExplorer
MicroExplorer	Specialized microprocessor for symbolic computing with a tagged memory architecture
MicroExplorer Memory	12 MB of physical memory dedicated to the MicroExplorer
Macintosh Memory	5 MB of physical memory dedicated to the Macintosh
Diskette Drive	Standard 800 KB microfloppy drive
Hard Drive	80 MB internal system drive, named HD
Color Screen	640 horizontal pixels X 480 vertical lines RGB monitor
Keyboard	Apple extended keyboard
Mouse	Standard Macintosh II mouse
External Hard Drive	High capacity (320 MB) drive with SCSI interface
Laser Printer	300 DPI Apple Laserwriter II
Tape Drive	Full size tape drive with 1600 and 3200 BPI densities and SCSI interface

Tables 6.2 and 6.3 use the convention that directory path names are separated by colons and a path name ending in a directory or subdirectory ends with a colon, whereas a path name ending with a file does not. The *NASA* folder contains both LISP (source) and XLD (compiled) code for the screens comprising the user interface. Both of these files are created by Action and therefore neither is source code in the normal sense. That is why they are both included at the top level of the *NASA* directory which also makes it slightly easier to invoke the Code Sizing Tool. There was some ambiguity in deciding where to include some of the source files, and it was decided that, if any function from a source file is used in running the Code Sizing Tool, that file is included in the *Production System* folder, otherwise it goes in the *Knowledge Acquisition* folder.

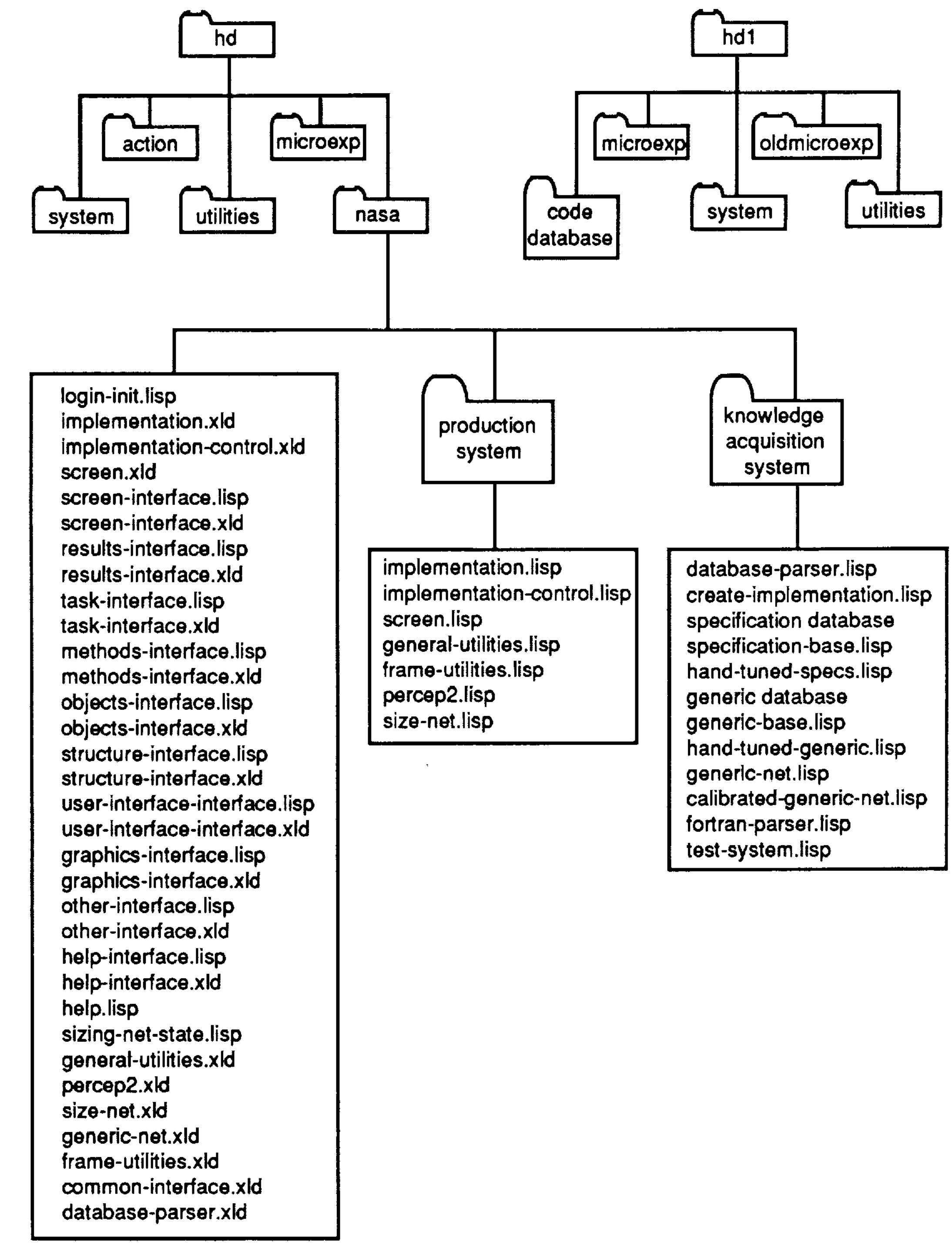


Figure 6.2
Directory Structure for Software Delivery

Table 6.2
Software Delivery Definitions - Part 1

hd:	The internal hard disk and system disk for the Macintosh II
hd:action:	Action™ software package, used for user interface
hd:utilities:	Contains general applications programs
hd:microexp:	Contains the MicroExplorer system software
hd:nasa:	Contains the software developed for the Code Sizing Project
login-init.lisp	Login file for Coding Sizing System
implementation.xld	Machine code for Production System knowledge base
implementation-control.xld	Machine code for Production System control
screen.xld	Machine code for user defined methods for user interface
results-interface.lisp	User interface to *results* screen
results-interface.xld	Machine code for *results* screen
screen-interface.lisp	User interface to *control* screen
screen-interface.xld	Machine code for *control* screen interface
task-interface.lisp	User interface to *task* screen
task-interface.xld	Machine code for *task* screen
methods-interface.lisp	User interface to *methods* screen
methods-interface.xld	Machine code for *methods* screen
objects-interface.lisp	User interface to *objects* screen
objects-interface.xld	Machine code for *objects* screen
structure-interface.lisp	User interface to *structure* screen
structure-interface.xld	Machine code for *structure* screen
user-interface-interface.lisp	User interface to *user-interface* screen
user-interface-interface.xld	Machine code for *user-interface* screen
graphics-interface.lisp	User interface to *graphics* screen
graphics-interface.xld	Machine code for *graphics* screen
other-interface.lisp	User interface to *other* screen
other-interface.xld	Machine code for *other* screen
help-interface.lisp	User interface to *help* screen
help-interface.xld	Machine code for *help* screen
help.lisp	Text for help screens
sizing-net-state.lisp	Network coefficients for sizing function
general-utilities.xld	Machine code for utility functions
frame-utilities.xld	Machine code for frame utilities
percep2.xld	Machine code for neural net functions
size-net.xld	Machine code for sizing function
database-parser.xld	Machine code for database access
common-interface.xld	Machine code for Common LISP user interface routines

Table 6.3
Software Delivery Definitions - Part 2

hd:nasa:production system:	Folder for Production System source
implementation.lisp	Production System knowledge base
implementation-control.lisp	Production System Control
screen.lisp	User defined methods for user interface
general-utilities.lisp	Utility functions of general applicability
frame-utilities.lisp	Utilities for the frame system
percep2.lisp	Neural network functions
size-net.lisp	Code sizing functions
hd:nasa:knowledge acquisition system:	Folder for Knowledge Acquisition source
database-parser.lisp	Input software database to frame structures
create-database.lisp	Create source code for generic knowledge base
specification database	Formatted specification database file
specification-base.lisp	Specification database for machine input
hand-tuned.lisp	Hand tuned specifications
generic database	Formatted generic database file
generic-base.lisp	Generic database for machine input
hand-tuned-generic.lisp	Hand tuned generic components
generic-net.lisp	Generic component regression functions
calibrated-generic-net.lisp	Generic component contribution coefficients
fortran-parser.lisp	Source code analysis functions
hd1:	High capacity external disk drive
hd1:microexp:	Backup copy of MicroExplorer system
hd1:oldmicroexp:	Old version of MicroExplorer system
hd1:code database:	Cosmic source code and parsed code
hd1:system:	Backup copy of Macintosh system
hd1:utilities:	Backup copy of utilities

The software is documented with transformation diagrams in this section. These diagrams show data and control flowing into and out of transformations, stores, and external terminators. The formalism is hierarchical because a single transformation in a diagram is documented by a transformation diagram at the next level of detail. Transformations at the lowest level are assumed to be understood or documented by the LISP source code. Figure 6.3 shows the elements that comprise the diagrams. Transformations are represented as labeled circles and usually refer to software transformations, although they are sometimes used to represent procedures carried out by people or mechanical devices. Control transformations have labels in italics and are used to turn other transformations off and on, rather than directly process the data. Terminators represent elements of the environment that interact with the system, such as the user or peripheral devices. Sets of similar transformations or terminators can be represented as shown. Stores are used to represent data or knowledge structures in memory. The Code Sizing System makes heavy use of stores because it is a frame based system. The three elements are connected by flows of data or control. There are two aspects to the reception of information by a transform: First, that a signal has been received, and second, the contents of the signal. Control flows are characterized by containing only the message that a signal has arrived. They are used to enable (turn on), disable (turn off), or trigger (execute once) other transformations. Flows are represented by directed, labeled links with control flows labeled in italics.

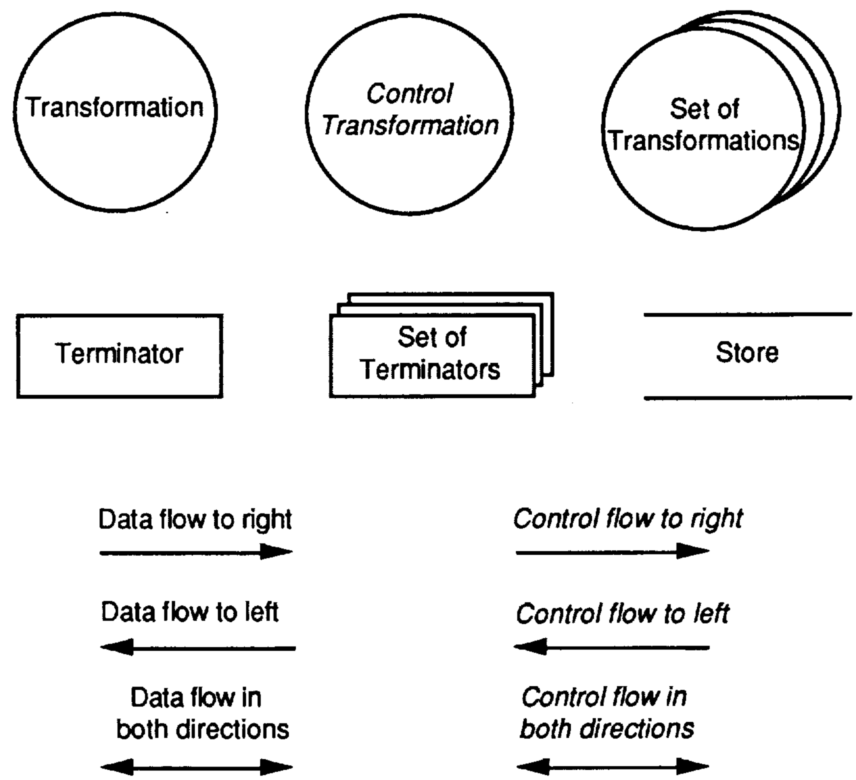


Figure 6.3
Elements of Transformation Diagrams

Systems are represented by a context schema at the highest level under this formalism. An example is shown in Figure 6.4 of an abstract system that accepts and provides messages to the user and an external storage device. The sample is shown in one further level of detail in Figure 6.5. The user messages are broken up into input and control parts. The Control transformation triggers an Initialization transform that initializes the Store from the data in external storage. It can enable or disable Transform based on instructions from the user. Transform allows the user to change the information in the Store. The Report transformation displays the current state of the Store and updates the information in external storage. The rest of Section 6 applies this formalism to the Production System, Knowledge Acquisition System, and Tools and Facilities.

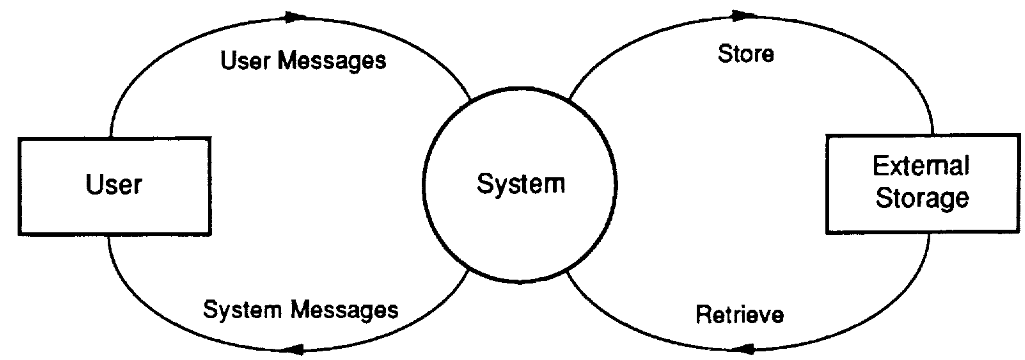


Figure 6.4
Sample Context Schema

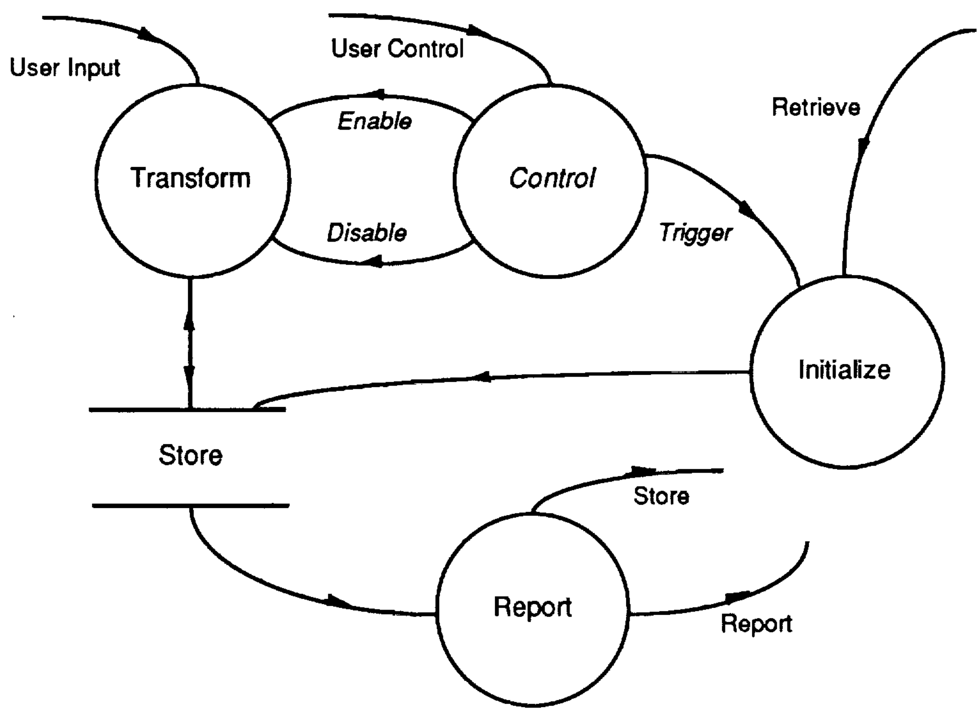


Figure 6.5
Sample Transformation Diagram

6.1 Production System

The production system is designed to produce code size estimates in response to program specifications supplied by the user. The context schema is shown in Figure 6.6. The user can issue one of the following commands to control the system:

SPECIFY	- Enter a complete set of specifications from the terminal
EDIT	- Edit a specific specification screen
STORE	- Write the current program specifications in a file.
RETRIEVE	- Read Program Specifications from an existing file.
HELP	- Display help text on the selected screen or specification
OK	- Remove the current screen
QUIT	- Leave the Code Sizing System
SIZE	- Perform a code size estimate

The user also supplies data in the form of program specifications and file names for the storage or retrieval of specifications. The interface is designed so that only legal responses can be input and there is therefore no explicit error handling. The exception to this is when the user specifies a bad specification file. The resulting errors will be caught by the internal LISP error handler. The system will provide the user with queries for software specifications, help messages, and size estimates. The compiled code for the production system, which contains the inferencing mechanisms and most of the knowledge base, is loaded into memory when the user logs into the system. The other files used by the system are the software specification files and a file called SIZING-NET-STATE.LISP, which contains knowledge for implementing the sizing function in the form of neural net coefficients, and HELP.LISP, which contains the help messages.

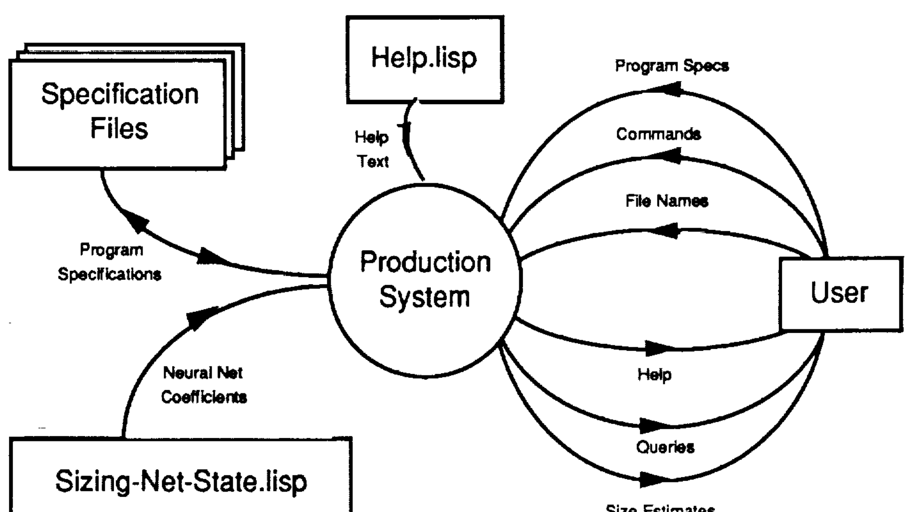


Figure 6.6
Context Schema for the Production System

The production system is defined in more detail in Figure 6.7 which contains the high level transformations. Almost all of the transformations are done on a global structure of frames that contain specifications for the software being estimated and the current internal state of the code sizing system itself. This is consistent with the implementation of the code sizing tool as a frame based system and is convenient for representing the highly structured knowledge within it. The only other data structure in the figure contains the sizing net coefficients.The screen handling mechanism is *object oriented* rather than frame based because it was created with the Action™ software tool. The initialization method for the initial control screen passes an initialization command to the Control Inference transformation which initializes the system. The User Interface stores the specifications in the Specification Frames when they are received from the user and passes a command to Control Inference to perform code size estimates when requested by the user. The Control Inference mechanism performs two basic functions. It initializes the system, mostly by modifying the Specification Frames store. It also performs code sizing estimates through the use of three other transformations: the Generic Rule Base produces generic component vectors based on the specifications; the Sizing Function produces size estimates based on generic component vectors, and Sizing Statistics produces estimates of the fluctuation factor based on a set of code size estimates.

The User Interface is defined in more detail in Figure 6.8. It contains a Control Screen transformation which creates and manages the initial screen of the Code Sizing Tool. This transformation accepts the user commands and either handles them itself or passes them to another transformation. It sends two commands to the Control Inference transformation: one is an initialization command which is sent automatically when the system is started up, and the other is a command to calculate sizing estimates in response to the user. The Edit and Specify commands activate one or more of the seven Specification Screens which display and receive the specifications of the program being analyzed. The Store or Retrieve commands initiate file I/O. Both the Control and Specification screens activate a help screen when the user clicks the mouse on a HELP button. The particular button chosen determines a frame and slot that is used to retrieve a help message that is displayed on a separate help screen. There are OK buttons to disable the help and specification screens when the user is finished with them.

The Control Inference mechanism is defined in more detail in Figure 6.9. The Control Initialization transform triggers three other transforms. Initialize Frames creates the frame structures for program specifications and sets slot values to their defaults. Initialize Specification Rules adds information about specification dependencies, and Initialize Sizing Net reads the neural net coefficients used in determining code size. These transformations comprise the upper half of the figure and are activated in response to an initialization command sent by User Interface when the system starts up. The bottom half of the figure contains the transforms that control the process of code size estimation. The Control Size Estimate transformation first sets the system state for deterministic sizing and activates the Generic Rule Base and Sizing Function to produce the most likely code size estimate. It then sets the system state for stocastic code sizing and activates the Generic Rule Base and Sizing Function a number of times (currently 20) to produce a set of size estimates which are used by the Sizing Statistic transform to estimate the fluctuation of the predictions.

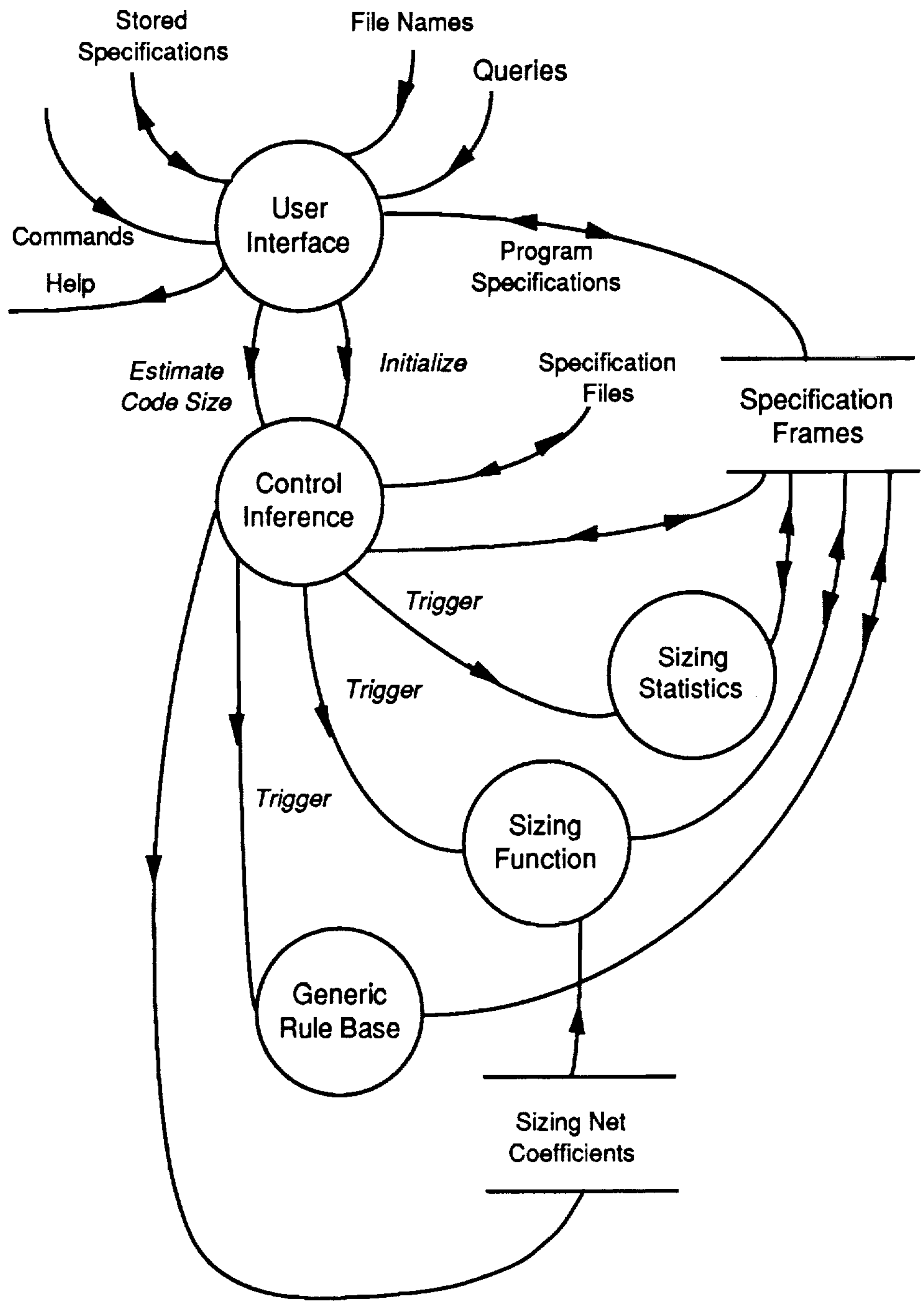


Figure 6.7
Preliminary Transformation Diagram for the Production System

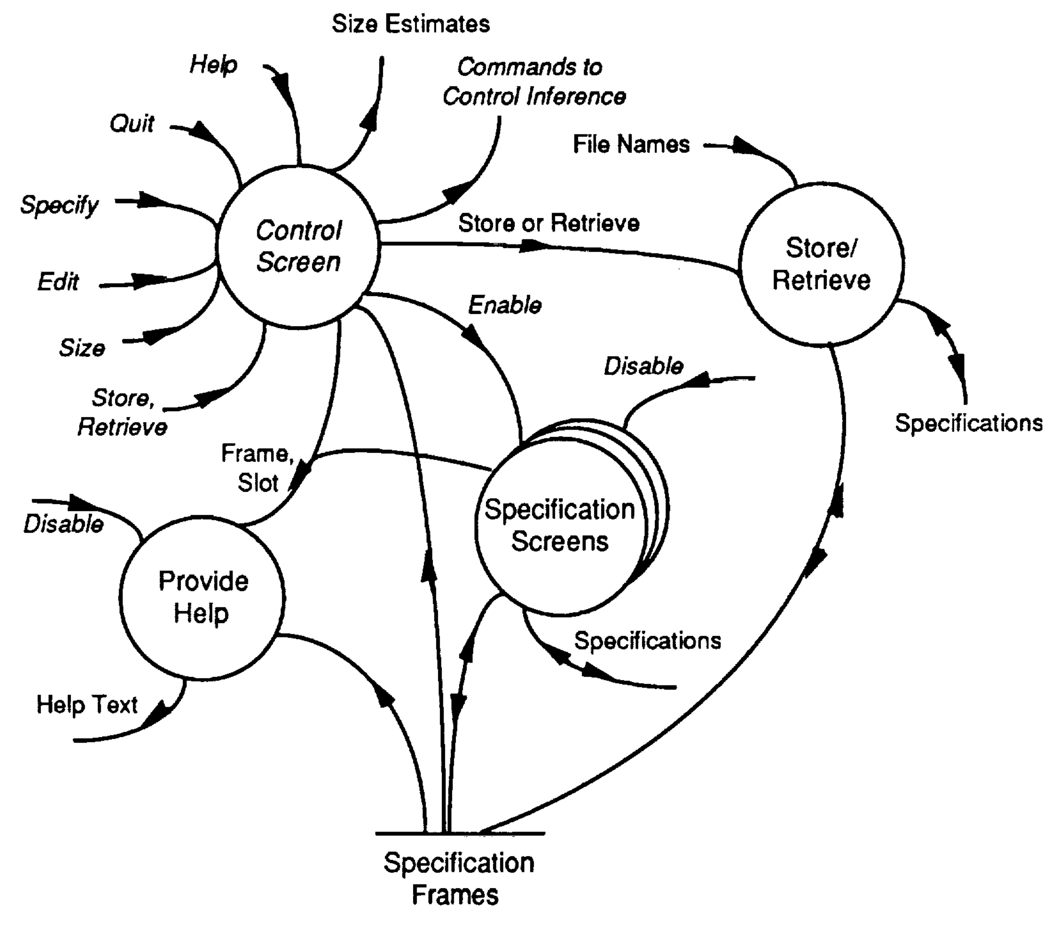


Figure 6.8
User Interface

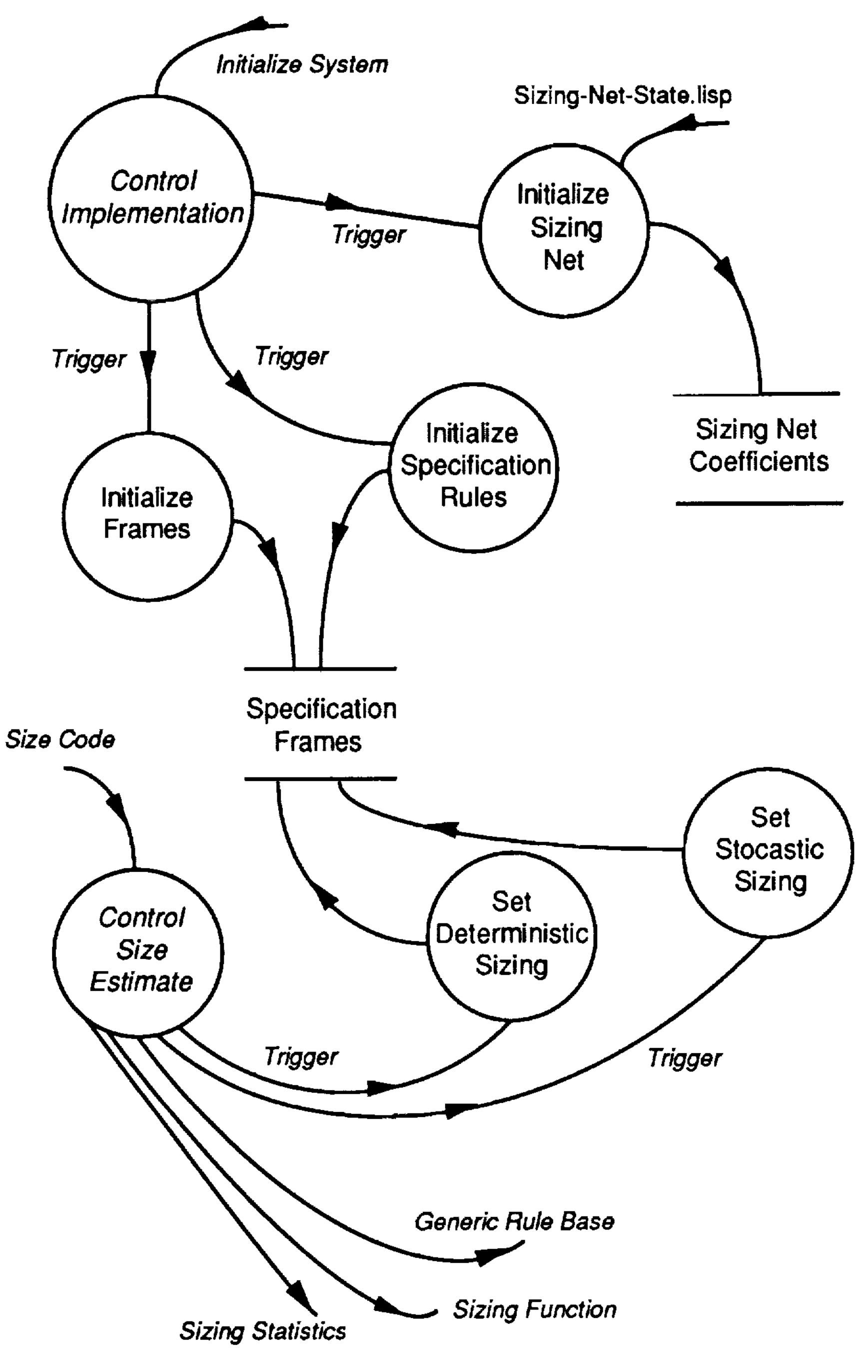


Figure 6.9
Control Inference

The mechanism of the Generic Rule Base is defined in Figure 6.10. A numeric value from 0 to 1 for each specification must be determined before the rule base can be used. This set of values is referred to as the Specification Vector. The process is initiated by Control Specification Assignments which sends each specification in turn to the Specification

Dependencies transformation. The specification dependencies can determine that a specification has the lowest possible value (ie it is absent). Otherwise either Get Deterministic Value or Get Stocastic Value is triggered, depending on the Sizing Type. After a value for each specification has been stored in the Specification Frames, Compute Generic Contribution is triggered to determine the generic components for the program.

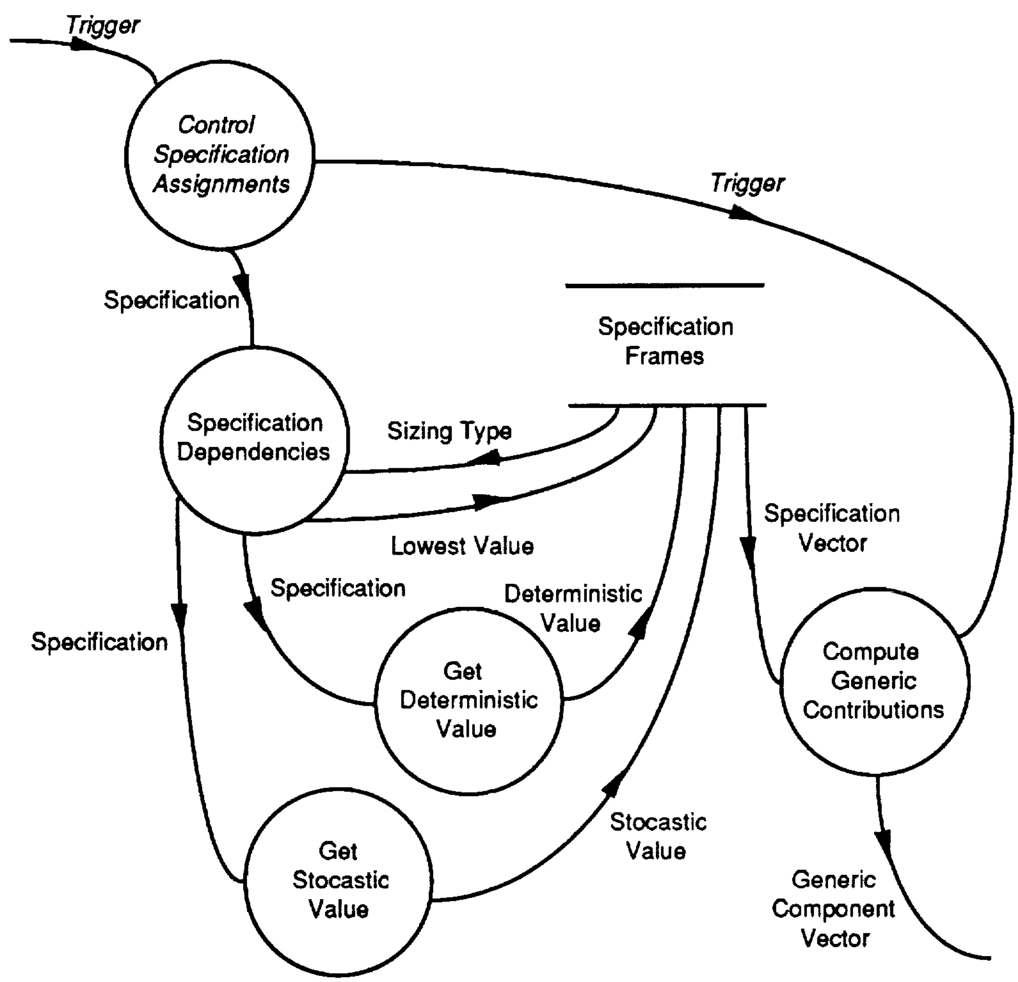


Figure 6.10
Generic Rule Base

6.2 Knowledge Acquisition System

The Context Schema for the knowledge Acquisition System is shown in Figure 11. The knowledge sources for the system are human expertise in the form of expert interviews, the literature, the experience of the developers, and existing software systems in the form of codes and documentation. The final product is the production system which is shown in three parts. The *implementation code* controls the inferencing and user interaction. The *generic rule base* contains the knowledge needed to turn the software specifications into a set of numbers (generic components) that measure stereotypical aspects of computer programs, and the *sizing net* which is a function, implemented as a neural network, that determines code size from generic components. There are also a number of intermediate results that are considered products of the Knowledge Acquisition System. These include the generic software components, software specifications, and a set of consistency rules called specification dependencies that relate the values of different specifications. The generic components and software specifications were used with the software database to create the *generic component database* and the *specification database*. A number of tools and facilities were developed and were used to create the sizing net and generic rule base from the generic component and specification databases and to test the accuracy of the code size prediction methods.

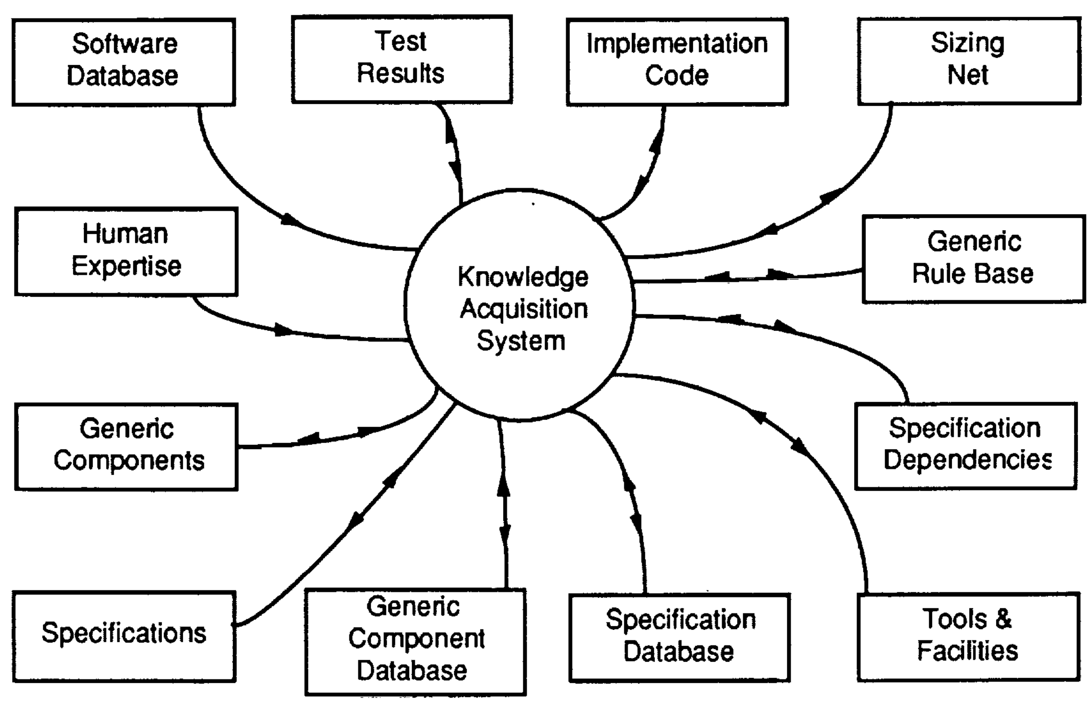


Figure 6.11
Context Schema for the Knowledge Acquisition System

The higher level procedures and their relationship to the knowledge acquisition products are shown in Figure 12. The knowledge sources were analyzed to determine the overall strategy including the system design and classification scheme. The necessary software was developed as was the generic component and specification databases. Finally the software and databases were used with the automatic analysis procedures to produce the test results, the sizing net, and the generic rule base. The test results were analyzed and used as feedback for the incremental design and implementation of the system. Of the four preceding processes, three, Conceptual Analysis, System Design, and Database Creation, were done "by hand" and will not be described further in this section. The Automatic Analysis process involves a software system which is described below.

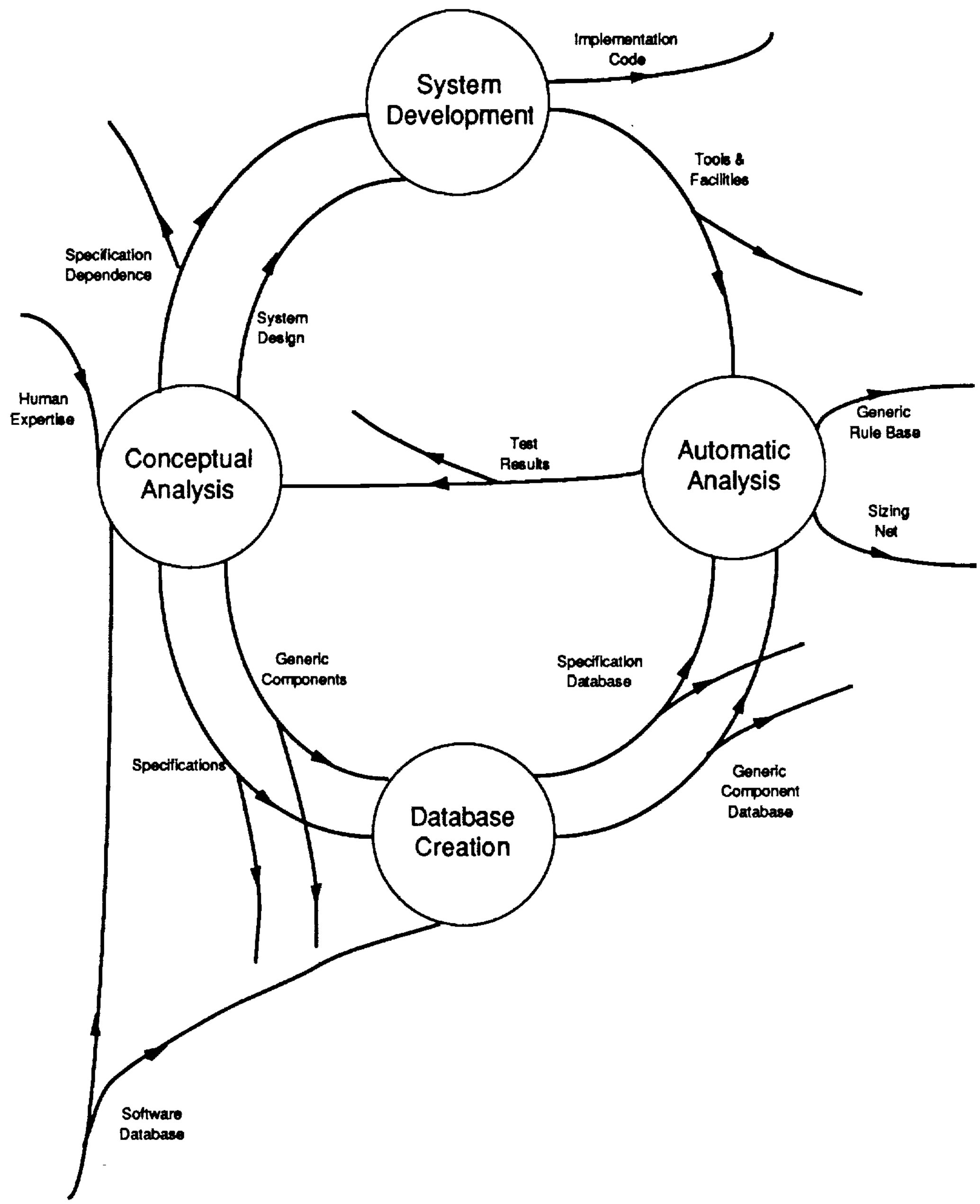


Figure 6.12
High Level Processes for the Knowledge Acquisition System

The high level transformations that comprise the automated part of the analysis are shown in Figure 6.13. The specification and generic component database files are read by Database Parser and converted to frame structures in Program Frames. This knowledge store also contains the current test results which can be stored in or retrieved from disk files by Get Results and Save Results. The test results can also be displayed on the screen in tabular form by Produce Test Results. There are two basic types of tests: tests of the sizing net, performed by Test Sizing, and tests of the generic net, performed by Test Generic. The networks can be stored and retrieved from disk files by Store Net and Retrieve Net. When the sizing tests are complete, the resulting network is stored in a file named SIZING-NET-STATE.LISP for use by the production system. When the generic component tests are complete, the generic network is converted to a rule base implemented in LISP, by Create-Implementation. This LISP code is compiled and is part of the Production System code.

The Database Parser, which initializes the knowledge structures in Program Frames, is shown in more detail in Figure 6.14. The initialization takes place in two stages: one for the specification and one for the generic components. Each stage accepts the database file name and an initialization flag. The previous structures are removed if the flag is set to *t*, through Set Up Features for the generic components and Initialize Specifications for the specifications. This feature allows multiple generic or specification database files to be used, if desired. Set Up System triggers both initializations with their default arguments: both initialization flags set to *t*, GENERIC-BASE.LISP for the generic component file, and SPECIFICATION-BASE.LISP for the specification file.

The Test Sizing transformation is shown in more detail in Figure 6.15. It does both calibrations and tests and, depending on the calibration parameters, will test a single program, test each program in turn, or calibrate the sizing net using all the programs. This information is passed to initialization and calibration routines. Initialize Sizing Test creates a network with random weights and uses the generic components and code sizes to create a training set which is passed to the calibration routine. If a program is being tested, it will not be included in the training set. Sizing Calibration uses the training set to calibrate the network. The current results are both displayed on the screen and recorded in the Program Frames knowledge store. The user can therefore disable the process (with CONTROL-ABORT on the keyboard) when satisfactory convergence has been achieved.

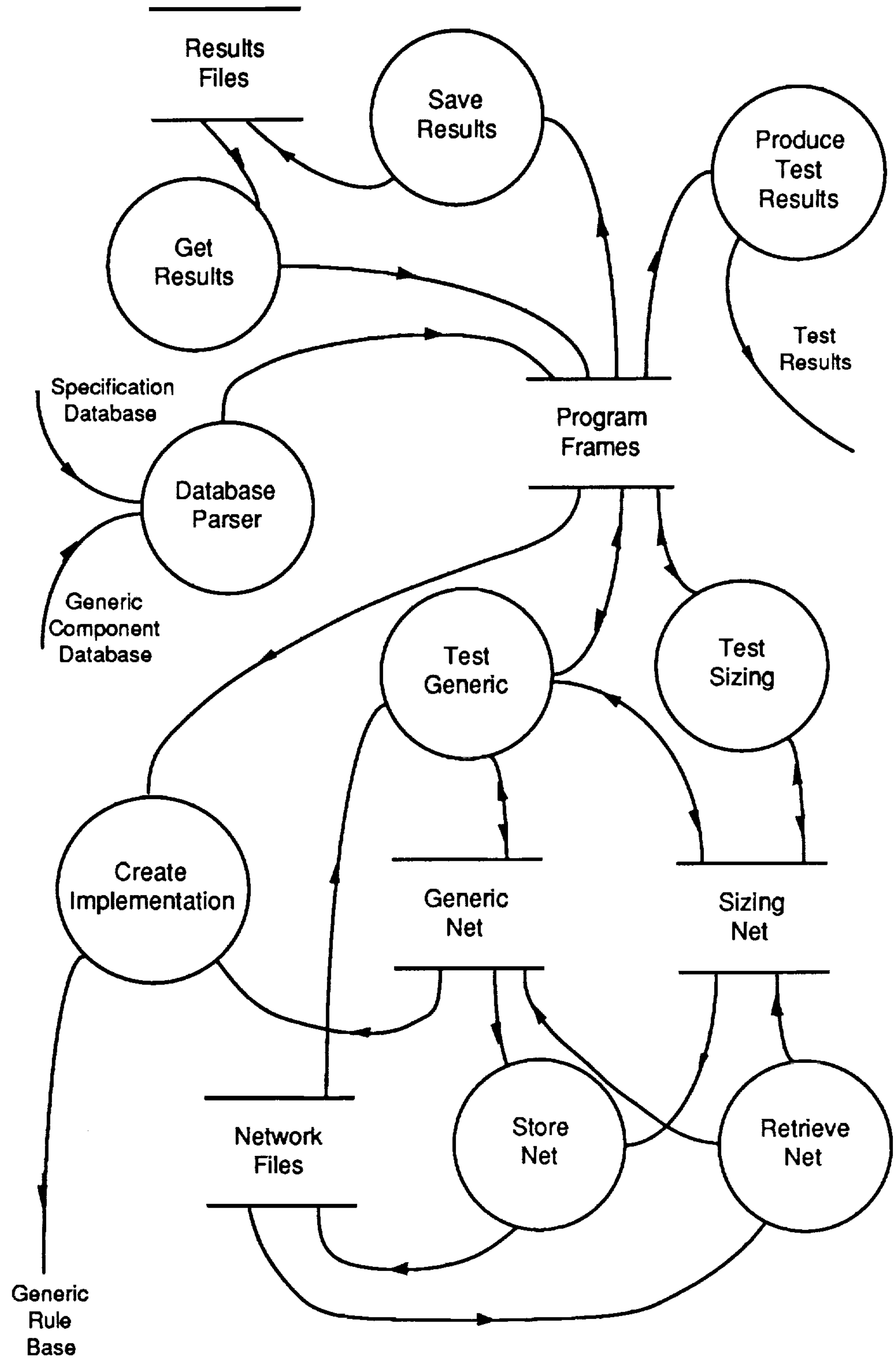


Figure 6.13
Automatic Analysis

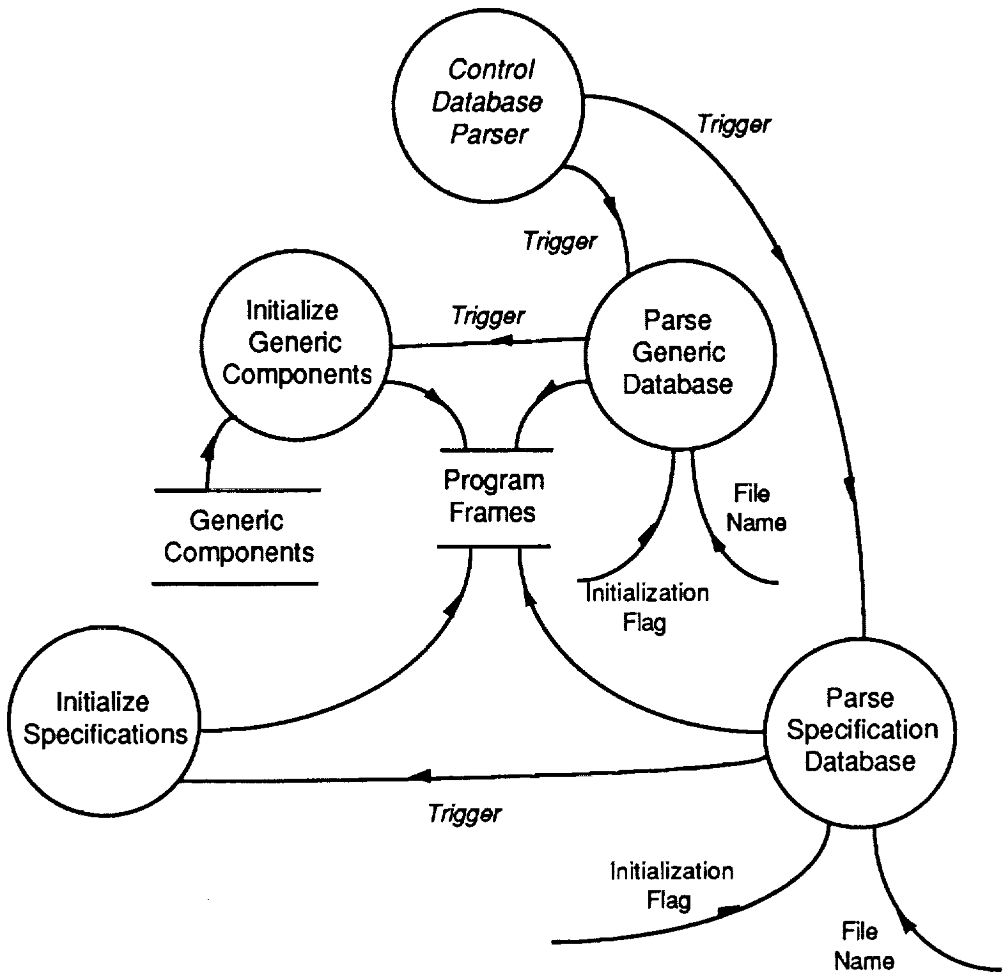


Figure 6.14
Database Parser

The Test Generic transformation (figure 6.16) is similar to Test Sizing except that both the sizing and generic nets are maintained in memory and accessed by the routines that display and store the current results. Another minor difference is that the transform uses its own store containing the types of specifications, which must be kept consistent with the specification database. The reason for this is to allow generic tests to be run from results files (after using Get Results) without reparsing the specification databases.

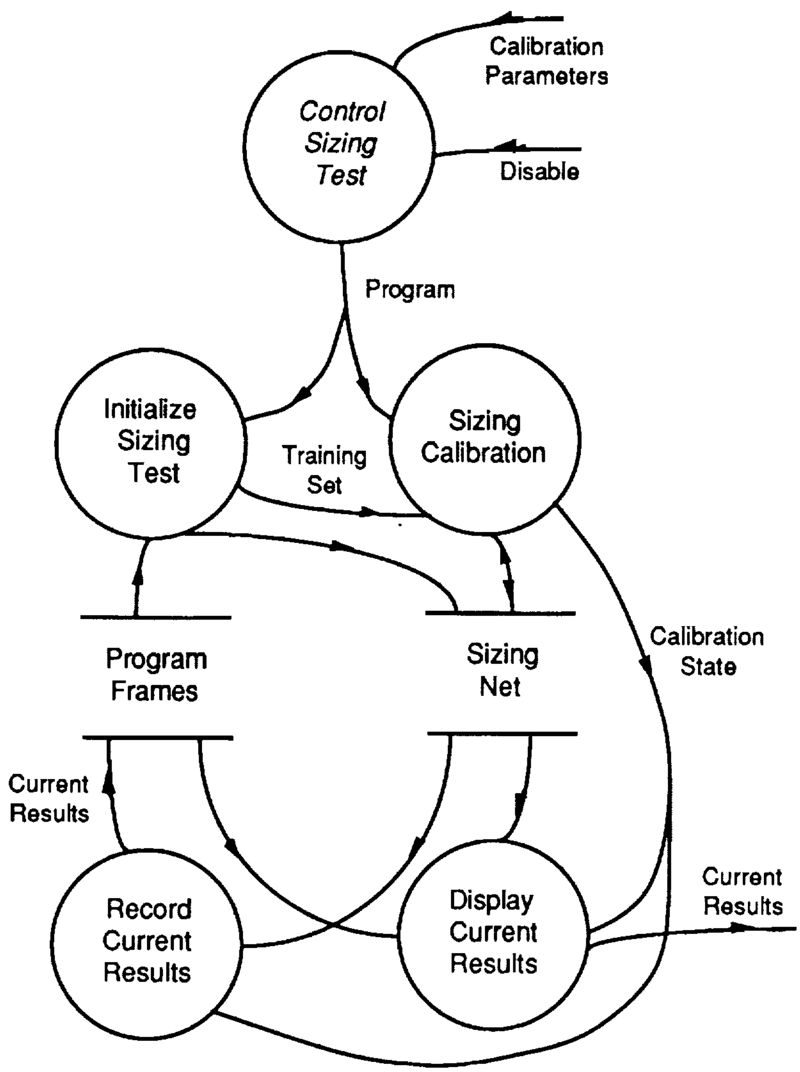


Figure 6.15
Sizing Test

The Create Implementation Transformation, which creates a rule base from the generic net coefficients that relates specifications to generic components, is shown in Figure 6.17. It is designed to be run after the final calibration of the generic net and uses the Specification Frames and Generic Net knowledge stores in their state after the calibration. Initialize Specification Frames readies Specification Frames for the creation procedures and produces lists of specification and generic component frames. Create If Needed Routines produces LISP code for the rule base and Save Implementation writes a source code file with functions to initialize the frame structures and execute the if-needed routines.

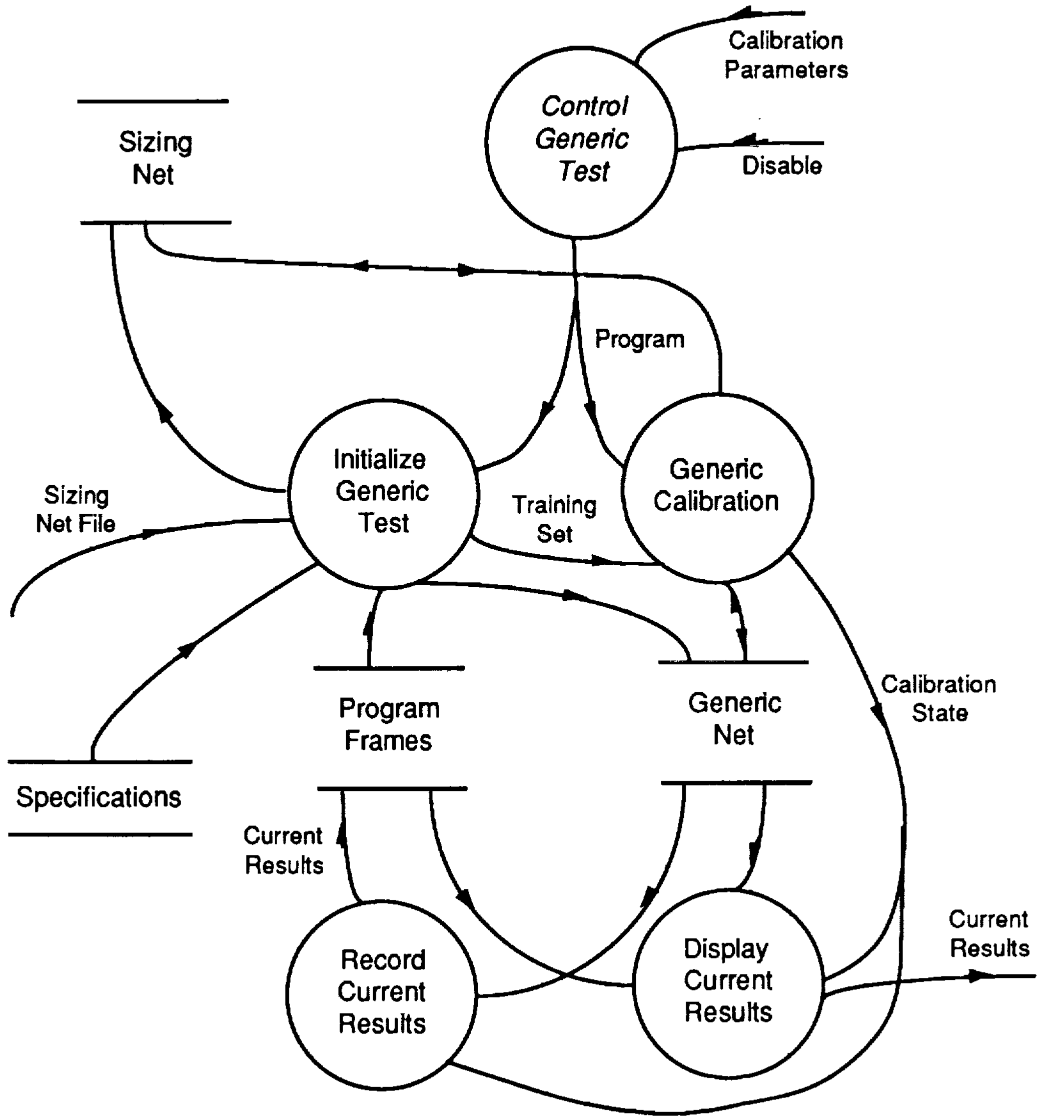


Figure 6.16
Generic Test

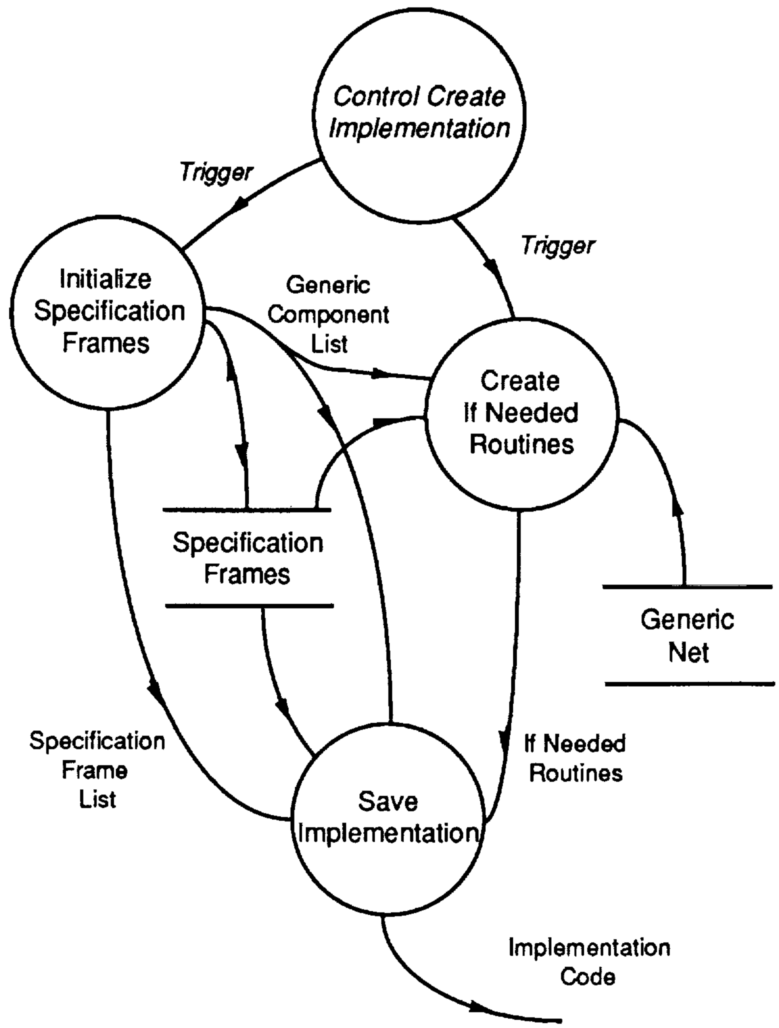


Figure 6.17
Create Implementation

6.3 Tools and Facilities

6.3.1 Frame Utilities

Figure 6.18 shows the context schema for the frame facility. The facility handles the interaction between the user and a data store with information structured as frames. The user can build or modify frame structures, or query the database. If a query is successful, the results will be a list of values. If not, *nil*, the empty list, is returned. The system achieves power and flexibility by allowing the user to associate side effects with any of these operations through the use of user specified routines called daemons, and by providing inheritance mechanisms that allow values and procedures to be inherited from more general to more specific classes.

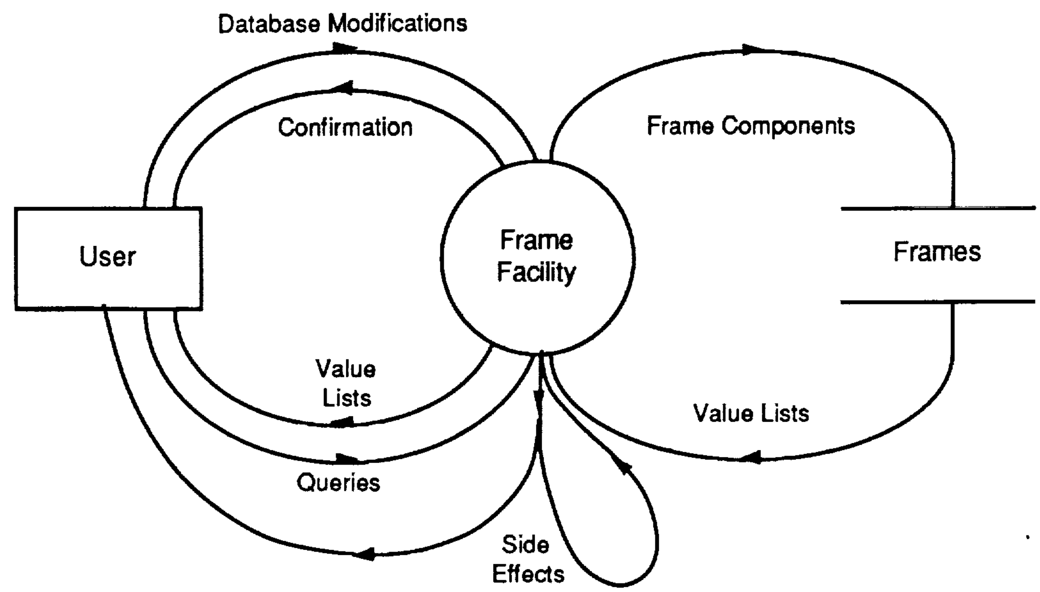


Figure 6.18
Context Schema for Frame Facility

Figure 6.19 shows the highest level transformations associated with the frame facility: queries and modifications. Queries require a frame and slot, and sometimes a facet (some queries automatically use the *value* facet). The query type is the name of the LISP function executed, which will determine what type of inheritance mechanism will be used and whether daemons are to be activated, if necessary, to aid in the search. If the frame, slot, and/or facet for an added value is missing, it is created. Modifications require a frame, slot, facet, and value. The modification type is the name of the LISP function executed, which determines whether the value is added or removed, and whether daemons are activated upon adding or removing the value.

Figure 6.20 shows the query mechanism in more detail. The Fget function will look for a value in a given frame, slot, and facet. Fget-classes uses this mechanism to find all of the "ancestors" of a given frame. The ancestors are found by tracing a frame's parents and parent's parents, etc. A frame's parents are defined as the contents of the value facet of its "a-kind-of" slot. This is the basic inheritance scheme and is used by Fget-i. Fget-z and Fget-n have somewhat more resourceful schemes in that they will also invoke daemons to help find values. Fget-z will look in the frame's value slot and, if no value is found, it will look in its default slot. If there is still no value, it will execute all of the routines in its "if-needed" slot, which will attempt to return values. This sequence is performed by Fget-v-d-p which is called by Fget-z. If a value still has not been found, Fget-z invokes Fget-v-d-p on each of the ancestor frames provided by Fget-classes until a value is found or all attempts have ended in failure. Fget-n is similar except that the search order is changed. The routine first looks through all the ancestor's value slots, then default slots, then if-needed slots in search of values. The "if needed" daemons can perform side effects including additional queries to the database, queries to the user, and modification to the database. The inheritance mechanism and daemons define the inference mechanism that is used in the code sizing system which primarily relies of Fget-z to control inference.

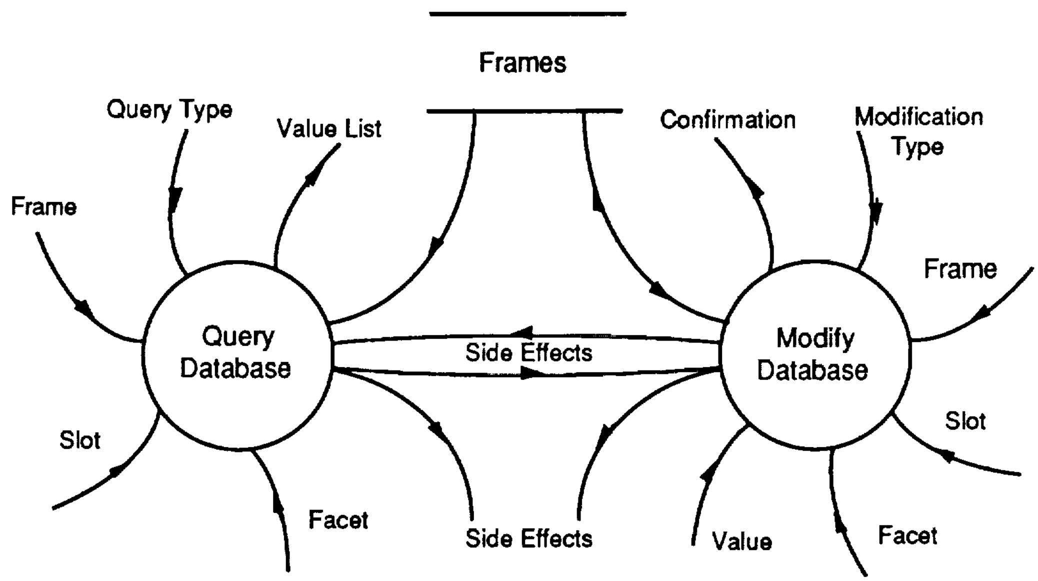


Figure 6.19
High Level Transformations for the Frame Facility

Figure 6.21 shown the database modification mechanism in more detail. The two lowest level routines, Fput and Fremove, handle all direct modifications of the database. Fput will put a given value in a given frame, slot, and facet, and Fremove will remove it. Fput-p and Fremove-p will activate any daemons assigned for the addition or removal of values by looking in the "if added" and "if removed" facets of the slot. Since these daemons are inherited, the ancestor frames, provided by Fget-classes, will also be checked. As in the case of queries, the daemons can produce side effects.

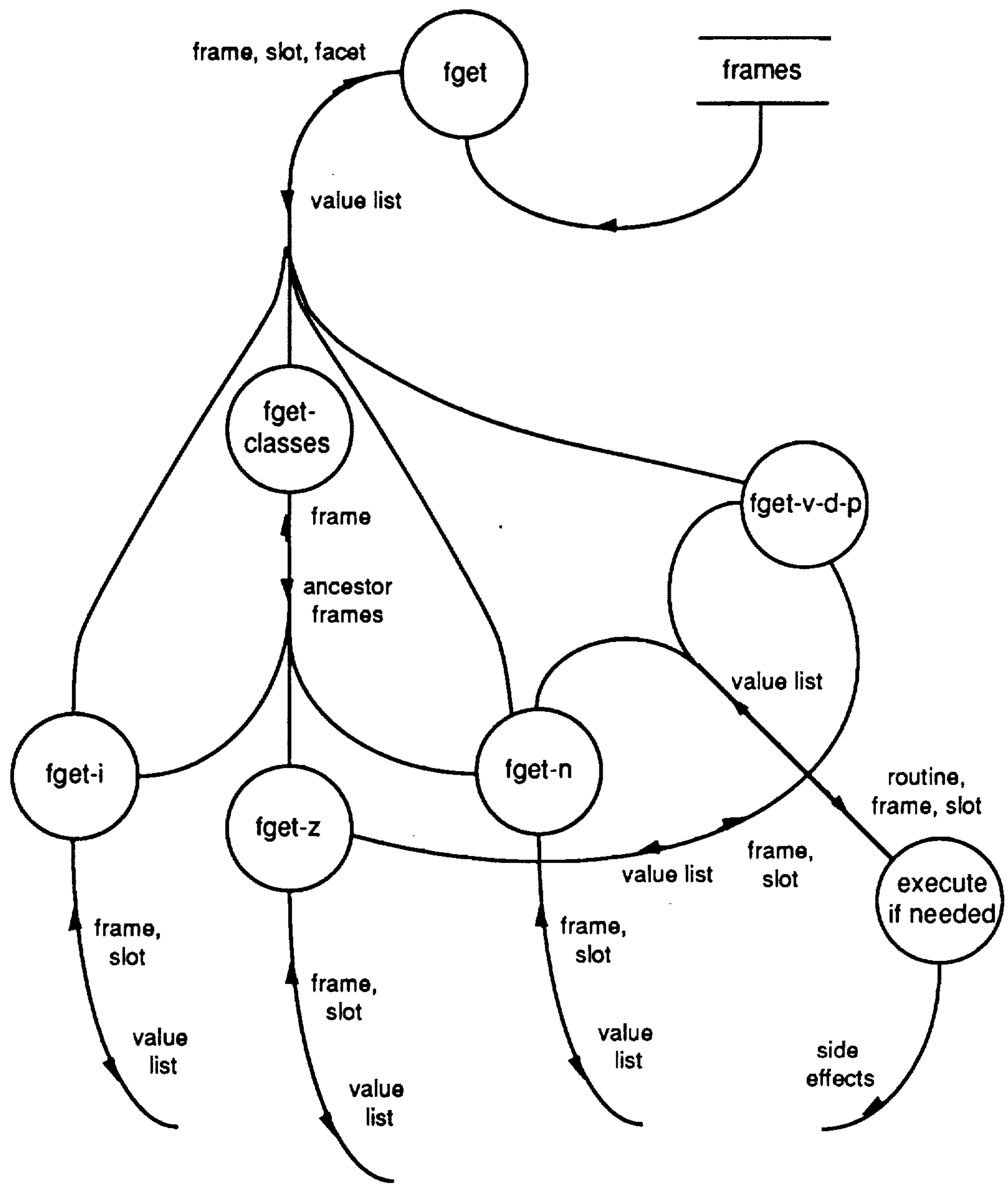


Figure 6.20
Query Mechanism

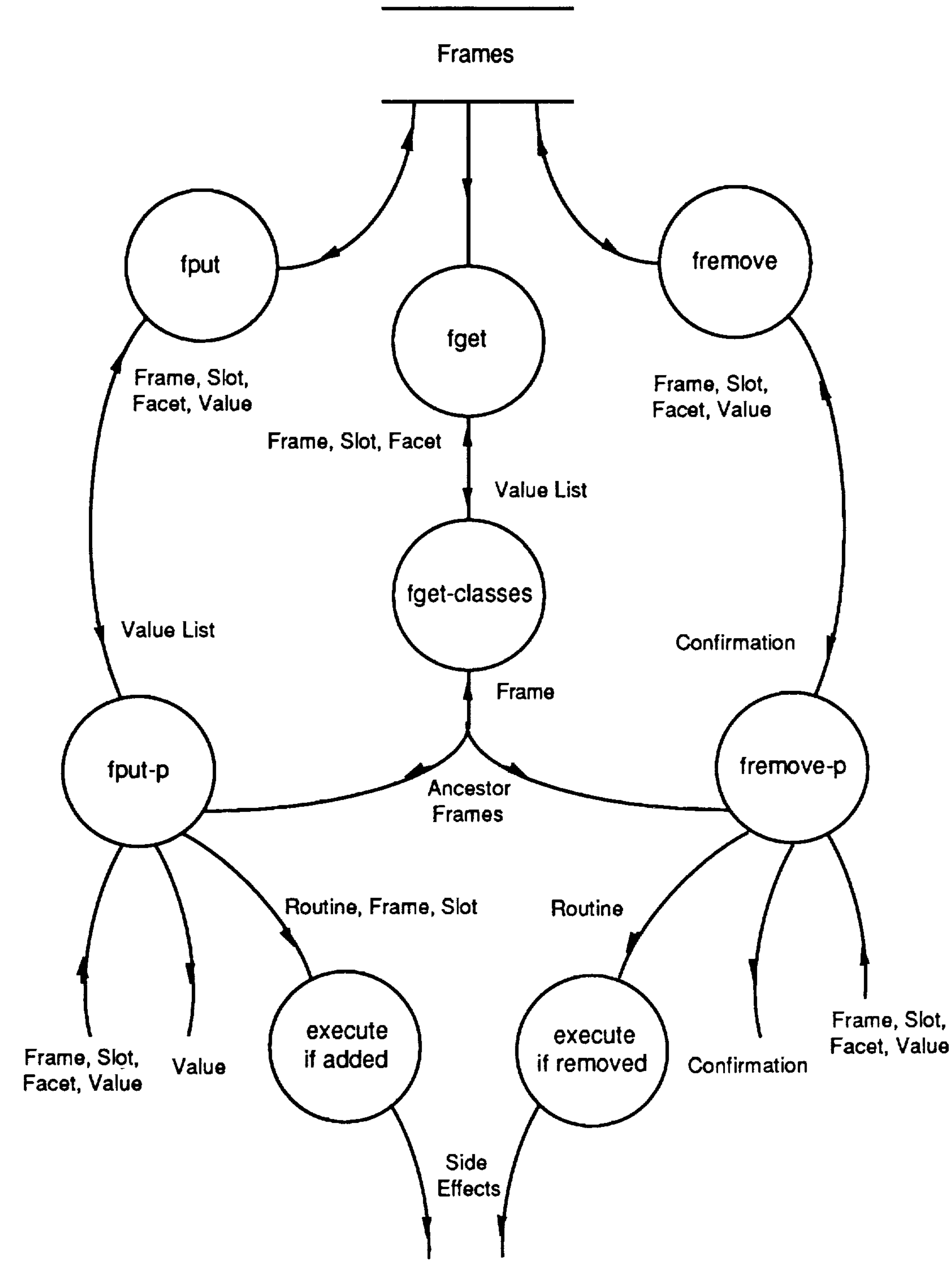


Figure 6.21
Database Modification Mechanism

6.3.2 Neural Net Facility

The neural net facility is described in Figure 6.22. The data store containing the nodes and connection weights for the network are represented as well as global parameters such as a flag that determines whether the net is linear or nonlinear, and the learning rates. Each of the transformations will follow computationally different, but conceptually analogous paths depending on whether the network is linear or nonlinear. The Learn transformation, for example, uses the perceptron convergence algorithm for linear nets and the backprop algorithm for nonlinear nets. The initialization routine, Set Up Net, creates the data structure for the network with random initial weights. The Learn transform uses a training set to adjust the weights and the Forward transform to propagate input vectors through the net and product output vectors.

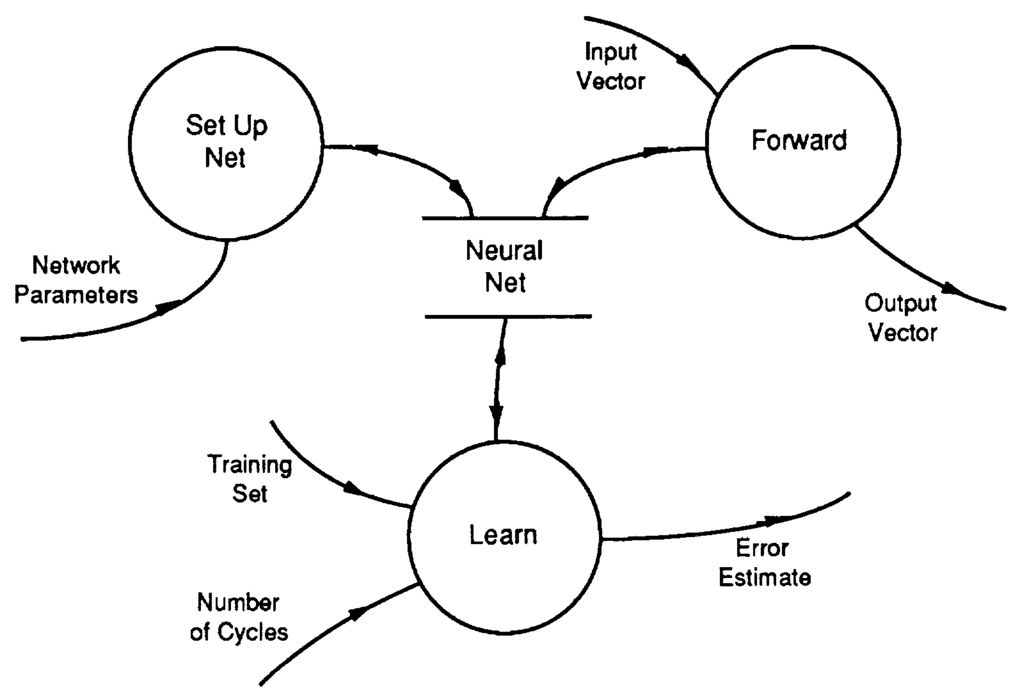


Figure 6.22
Neural Net Facility

The Learn transform is expanded in more detail in Figure 6.23. For each cycle it will go through the training set and, for each sample in the training set, it will propagate the input vector forward through the net, compare the calculated output with the desired output, and propagate the difference backward through the net, then it will update the weights according to the adjustments calculated for that cycle and print an estimate of the error (difference between calculated and desired outputs).

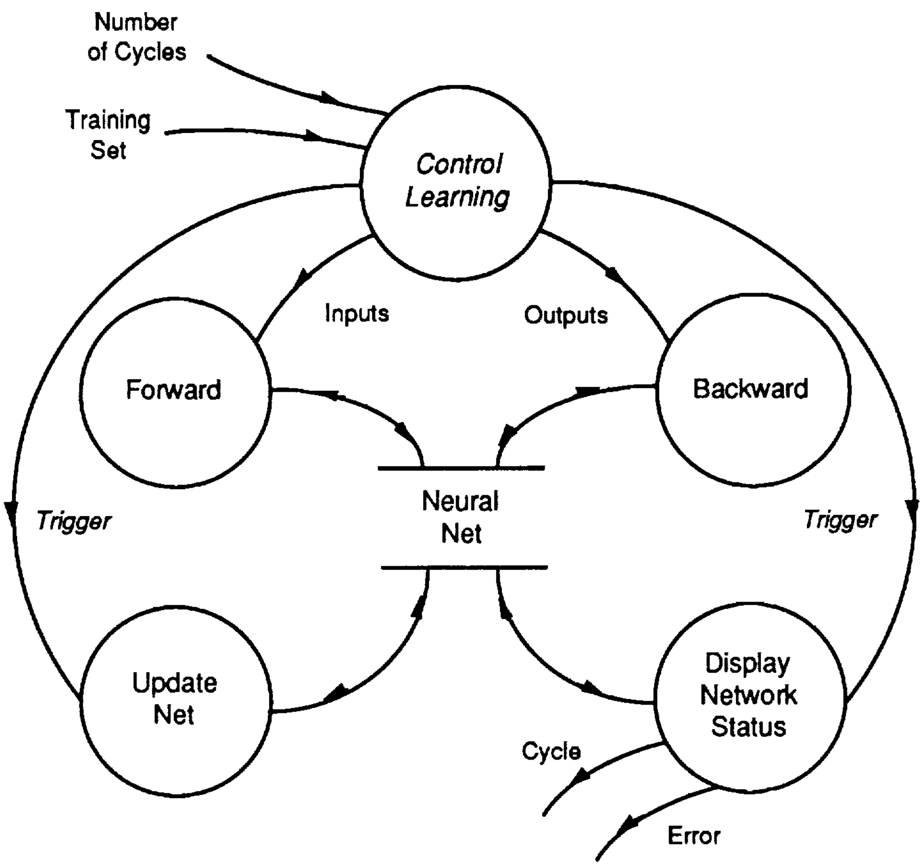


Figure 6.23
Learn

7.0 References

1. Albrecht, A.J., and Gaffney Jr., J.E., "Software Function, Source lines of Code, and Development Effort Prediction: A Software Science Validation, " IEEE Trans. on Software Eng., Vol SE-9, No. 6, Nov. 1983, pp 639-648.

2. Albrecht, A.J., "Measuring application development productivity," Proc. IBM Applications Development Symp., Monterey, CA Oct. 14-17, 1979; GUIDE int. and SHARE, Inc., IBM Corp.

3. Augustine, N.R., "Augustine's Laws and Major System Development Programs," Defense Systems Management Review, 1079, pp. 50-76.

4. Basili, V.R., "Metric Analysis and Data Validation Across Fortran Projects," IEEE Trans. Software Eng. Vol. SE-9, No. 6, 1983, pp 652-662.

5. Basili, V.R., "An Empirical Study of a Syntactic Complexity Family," IEEE Trans. Software Eng. Vol. SE-9, No. 6, 1983, pp 664-672.

6. Boehm, B.W., Software Engineering Economics, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1981.

7. Forsyth, R., Expert Systems, Principles and Case Studies, Chapman and Hall, New York, NY, 1984.

8. Halstead, M.H., Elements of Software Science, Elsevier, New York, 1977.

9. Levitin, A.V., "On Predicting Program Size by Progam Vocabulary," Proc. IEEE 9th International Computer Software and Applications Conference, Chicago, IL, Oct. 1985, pp. 98-103.

10. Lippmann, R.P., "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, April, 1987.

11. Rubin, H.A., "A Comparison of Cost Estimation Tools (A Panel Session)," Proc. IEEE 8th International Conference on Software Eng., London, UK, August 1985, pp. 174-180.

12. Rumelhart, D.E., Hinton, G.E., & Williams, R.J. "Learning Internal Representations by Error Propagation", Institute for Cognitive Science, ICS Report 8506, UCSD, September, 1985.

13. Shortliffe, E. Computer Based Medical Consultations: MYCIN, American Elsevier, New York, 1976.

14. Synott, D.J., and Tausworthe, R.C., "Software Task Statistics Derived from DSN Mark IV-A WBS System Data," Report JPL D-1982, Jet Propulsion Laboratory, Pasadena, CA, Oct. 1984.

15. Tauseworthe, R.C., "Software Acqusisition and Development Cost Forecasting Guidance," Jet Propulsion Laboratory, Pasadena, CA, Now. 8, 1985.

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

17. Winston, P.H., Artificial Intelligence, Second Edition, Addison-Wesley Publishing Coompany, Reading, MA, 1984.

Appendix 1 - Source Codes

The following source code files were delivered with the Code Sizing Tool and are included in this appendix:

1. hd:nasa:login-init.lisp
2. hd:nasa:production system:common-interface.lisp
3. hd:nasa:production system:frame-utilities.lisp
4. hd:nasa:production system:general-utilities.lisp
5. hd:nasa:production system:implementation-control.lisp
6. hd:nasa:production system:implementation.lisp
7. hd:nasa:production system:percep2.lisp
8. hd:nasa:production system:screen.lisp
9. hd:nasa:production system:size-net.lisp
10. hd:nasa:knowledge acquisition system:
    create-implementation.lisp
11. hd:nasa:knowledge acquisition system:database-parser.lisp
12. hd:nasa:knowledge acquisition system:fortran-parser.lisp
13. hd:nasa:knowledge acquitition system:generic-net.lisp
14. hd:nasa:knowledge acquitition system:test-system.lisp

File hd:nasa:login-init.lisp

```
;;; -*- Mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-

(load "lm:hd:nasa:frame-utilities")
(load "lm:hd:nasa:general-utilities")
(load "lm:hd:nasa:common-interface")
(load "lm:hd:nasa:percep2")
(load "lm:hd:nasa:size-net")
(load "lm:hd:nasa:generic-net")
(load "lm:hd:nasa:database-parser")
(load "lm:hd:nasa:implementation")
(load "lm:hd:nasa:implementation-control")
(load "lm:hd:nasa:screen")
(sys:add-printer-device "LaserWriter II NT" :mac-printer
                        'lm :stream :mac-printer :global nil)
(set-default-printer "LaserWriter II NT")
(set-default-image-printer "LaserWriter II NT")
(pkg-goto 'tb)
```

File hd:nasa:production system:common-interface.lisp

```
;;; -*- mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-

; This section contains user I/O functions

(defun ask-alt (frame slot &aux alternatives answer)
  (setq alternatives (fget-n1 slot (fget-classes frame) 'range))
  (while (not (member answer alternatives))
        (format t "~%~%   For frame ~g, slot ~g," frame slot)
        (format t "~%   Choose an alternative from ~G:~%~%   "
                alternatives)
        (setq answer (read))
        (format t "~%~%"))
  (iff 'recorder 'mode 'record
     (fput 'recorder 'record 'value (list frame slot answer)))
  (list (fput frame slot 'value answer)))

(defun ask-parts (category parts &aux answer done)
  (while (not done)
        (format t "~%~%   For category ~g, choose zero or more" category)
        (format t "~%   alternatives from ~G:~%~%   " parts)
        (setq answer (read))
        (format t "~%~%")
        (cond ((null answer) (setq done t))
            ((atom answer) (setq answer (list answer)))
            (t))
        (setq done (subset-of answer parts)))
  answer)

(defun subset-of (subset set)
  (cond ((null set) nil)
        ((null subset) t)
        (t (and (member (car subset) set)
                (subset-of (cdr subset) set)))))

(defun ask-subset (frame slot &aux subset set)
  (setq set (fget-n1 slot (fget-classes frame) 'range))
  (setq subset (ask-parts slot set))
  (dolist (l subset)
   (fput frame slot 'value l))
  subset)

(defun ask-range (frame slot &aux (range (fget frame slot 'range))
                        inp)
  (loop
   (format t "~%~%For frame ~s, slot ~s, select a value between"
          frame slot)
   (format t "~%~s and ~s:~%~%" (first range) (second range))
   (setq inp (read))
   (if (and (>= inp (first range)) (<= inp (second range)))
        (return-from ask-range (fput-list frame slot inp)))))
```

File hd:nasa:production system:frame-utilities.lisp

```
;;; -*- mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-


;;  *********************************************************************

;;  frame functions from Winston

;;  *********************************************************************

; fget gets the value of "facet" in "slot" of "frame".  It will return nil if
;  anything in the chain is missing or if "facet" has no value

(defun fget (frame slot facet)
  (cdr (assoc facet (cdr (assoc slot (cdr (get frame 'frame)))))))

; look for an entry under "key" in an assoc list "a-list".  If one is not
; add it in.

(defun extend (key a-list)
 (cond ((assoc key (cdr a-list)) )
     (t (cadr (rplacd (last a-list) (list (list key)))))))

; follow-path uses extend to push through the frame structure adding elements
; when necessary

(defun follow-path (path a-list)
 (cond ((null path) a-list)
     (t (follow-path (cdr path) (extend (car path) a-list)))))

; fput probes through the frame structure adding elements when necessary.
; it will eventually add "value" to "facet" of "slot" in "frame".  It
; returns the value if it is not already on the value list, nil otherwise.

(defun fput (frame slot facet value)
 (let ((value-list (follow-path (list slot facet)
                    (fget-frame frame))))
   (cond ((member value value-list :test #'equal) nil)
       (t (rplacd (last value-list) (list value))
        value))))

; FGET-FRAME gets, or, if necessary, makes a frame structure for "frame"

(defun fget-frame (frame)
 (cond ((get frame 'frame) )
     (t (putprop frame (list frame) 'frame))))

; FREMOVE will remove "value" from "facet" of "slot" in "frame"
; This has been revised (DSF) to allow LIST values to be removed.

(defun fremove (frame slot facet value &aux flag new-list)
 (let ((value-list (follow-path (list slot facet)
                    (fget-frame frame))))
   (dolist (v (cdr value-list))
    (if (equal v value) (setq flag t)
          (push-end v new-list)))
   (if flag (rplacd value-list new-list))
   flag))

; FGET-V-D-P will look at the VALUE, then the DEFAULT facets for a value.
; If nothing is found, it activates the IF-NEEDED facet.
; The routines are modified from Winston so that the procedure can be inherited
```

```
; but called with the original frame name as FRAME parameter.
; Another modification was made so that if an IF-NEEDED routine returns an atom,
; it will still be included in the results. Note the following convension for
; IF-NEEDED routines:
;   ATOM or (ATOM) -----> a single atomic value
;   (A1 .... An) -----> a set of values
;   ((A1 .... An)) -----> a single value that is a list.

(defun fget-v-d-p (frame slot)
 (fget-v-d-pp frame slot frame))

(defun fget-v-d-pp (frame slot original-frame)
 (cond ((fget frame slot 'value))
      ((fget frame slot 'default))
     (t
         (mapcan #'(lambda (demon &aux val)
                      (if (listp (setq val (funcall demon original-frame slot)))
                             val (list val)))
                   (fget frame slot 'if-needed)))))


; Similarly these routines can activate demons when values are addes or removed.
; The convension is that IF-ADDED routines take the frame and slot as arguements,
; but the IF-REMOVED routines have no arguements. In both cases, the return values
; are combined using MAPCAR (rather than MAPCAN as in the above example).

(defun fput-p (frame slot facet value)
 (cond ((fput frame slot facet value)
          (mapcar #'(lambda (e)
                      (mapcar #'(lambda (demon) (funcall demon frame slot))
                              (fget e slot 'if-added)))
                   (fget-classes frame))
          value)))

(defun fremove-p (frame slot facet value)
 (cond ((fremove frame slot facet value)
          (mapcar #'(lambda (e)
                      (mapcar #'(lambda (demon) (apply demon nil))
                              (fget e slot 'if-removed)))
                   (fget-classes frame))
          value)))

(defun fget-i (frame slot)
 (fget-i1 (fget-classes frame) slot)
 )

(defun fget-i1 (frames slot)
 (cond ((null frames)
          nil)
         ((fget (car frames) slot 'value))
         (t
          (fget-i1 (cdr frames) slot))
         )
 )

(defun fget-classes (start)
 (reverse (fget-classes1 (list start) nil))
 )

(defun fget-classes1 (queue classes)
 (cond ((null queue)
          classes)
         (t
          (fget-classes1 (append (fget (car queue) 'a-kind-of 'value)
```

```
                    (cdr queue))
            (cond ((member (car queue) classes)
                   classes)
                  (t
                   (cons (car queue) classes)
                   ))))))

; The routines are modified from Winston so that the procedure can be inherited
; but called with the original frame as a parameter.

(defun fget-z (frame slot)
  (fget-z1 slot (fget-classes frame) frame)
 )

(defun fget-z1 (slot classes frame)
  (cond ((null classes)
         nil)
        ((fget-v-d-pp (car classes) slot frame))
        (t
         (fget-z1 slot (cdr classes) frame)
         ))
 )

(defun ask-user (frame slot)
  (print `(please supply a value for the ,slot slot in the ,frame frame))
  (terpri)
  (let ((response (read)))
    (cond (response
           (list response))
          (t
           nil)
          )
   )
 )

; The routines are modified from Winston so that the procedure can be inherited
; but called with the original frame as a parameter.

(defun fget-n (frame slot)
  (let ((classes (fget-classes frame)))
    (cond ((fget-n1 slot classes 'value))
          ((fget-n1 slot classes 'default))
          ((fget-n2 slot classes frame))
          (t
           nil))))

(defun fget-n1 (slot classes key)
  (cond ((null classes)
         nil)
        ((fget (car classes) slot key))
        (t
         (fget-n1 slot (cdr classes) key)
         )))

(defun fget-n2 (slot classes original-frame)
  (cond ((null classes)
         nil)
        ((mapcan #'(lambda (demon) (funcall demon original-frame slot))
                 (fget (car classes) slot 'if-needed)))
        (t
         (fget-n2 slot (cdr classes) original-frame)
         )))
```

```
;; ********************************************************
;; These functions were added to Winston's basic set to increase the utility
;; of the frame system
;; ********************************************************


; These were used with the Fortran Parser to make routine calling structures
; from the source codes

(Defun make-frame (struct)
 `(,(first struct) (SIZE (VALUE ,(second struct)))
   (COMMENT (VALUE ,(third struct)))
   (CHILDREN (VALUE ,@(fourth struct)))))

(defun make-frame-list (st-list)
 (mapcar #'make-frame st-list))


; This macro makes defining the rules easier

(defmacro iff (fr sl vl success &optional failure)
 `(if (member ,vl (fget-z ,fr ,sl)) ,success ,failure))


; This routine will put a list of values in a slot

(defun fput-list (frame slot value-list)
 (if (null value-list) (return-from fput-list nil))
 (if (atom value-list) (setq value-list (list value-list)))
 (dolist (value value-list)
   (fput frame slot 'value value))
 value-list)


; This function displays the contents of a frame

(defun show (frame) (get frame 'frame))


; This routine deletes a frame structure

(defun delete-frame (frame)
 (putprop frame nil 'frame))


; set-frame sets symbol "frame" to "value" which should be in the form of a
; frame

(defun set-frame (value)
 (putprop (car value) (copy-tree value) 'frame)
 (car value))

; set-frame-list sets a list of frame structures.  It takes the car of the frame
; structure (the frame name) as the symbol to attach the frame structure to

(defun set-frame-list (frame-list)
 (mapcar 'set-frame frame-list))


; These functions can set a value to the only value in a slot or clear out all the
; values in a slot.
```

```lisp
(defun fset (frame slot facet value)
  (fclear frame slot facet)
  (fput frame slot facet value))

(defun fclear (frame slot facet)
  (rplacd (follow-path (list slot facet) (fget-frame frame)) nil)
  t)

; This function gets a whole slot

(defun sget (frame slot)
  (cdr (assoc slot (cdr (show frame)))))
```

File hd:nasa:production system:general-utilities.lisp

```
;;; -*- Mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-


; A file of general Lisp utilities

; This macro defines an increment function

(defmacro inc (sym n) `(setq ,sym (+ ,sym ,n)))


; These macros are used for extraction a random element, or set of
; elements from a list

(defmacro random-pop (l)
  (let ((ele (gensym)))
    `(if (null ,l) nil
         (let ((,ele (nth (random (length ,l)) ,l)))
           (setq ,l (remove ,ele ,l))
           ,ele))))

(defmacro random-sample (size l)
  (let ((sample (gensym)))
    `(let ((,sample))
       (dotimes (i (min ,size (length ,l)) ,sample)
         (push (random-pop ,l) ,sample)))))


; These routines do simple statistics on a set of numbers.

(defun mean (set)
  (/ (apply '+ set) (length set)))

(defun rms (set)
  (sqrt (/ (apply '+ (mapcar #'(lambda (x) (* x x)) set))
           (length set))))

(defun sigma (set &aux (ave (mean set)))
  (rms (mapcar #'(lambda (x) (- x ave)) set)))
```

file hd:nasa:production system:implementation-control.lisp

```lisp
;;; -*- Mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-


(defun initialize-sizing (&aux initial-frames)
  (setq initial-frames (initialize-frames))
  (fput-list 'sizing-criteria 'children (cdr initial-frames))
  (set-frame '(program (predicted-size (if-needed program-predicted-size))))
  (initialize-sizing-net)
  (initialize-specification-rules)
  (fset 'code-sizing-system 'specification-statistics 'value 'deterministic)
  (fset 'code-sizing-system 'number-of-iterations 'value 20)
  (fset 'code-sizing-system 'fluctuation 'value 1.8)
  )

(defun initialize-sizing-net (&optional (file "sizing-net-state"))
  (with-open-file (in file)
    (RestoreState (read in)))
  t)

(defun generic-features ()
  (fget-z 'sizing-criteria 'output-nodes)
  )

(defun specification-features (&aux features)
  (dolist (criterion (fget-z 'sizing-criteria 'children) features)
    (dolist (aspect (fget-z criterion 'specification-slots))
      (push-end (list criterion aspect) features))))

; This routine will assign values to the qualitative specifications from their
; certainty values

(defun assign-specification-values ()
  (dolist (spec (specification-features))
    (fset (first spec) (second spec) 'value
          (specification-value  (first spec) (second spec)))))

(defun program-predicted-size (frame slot)
  (fput-list frame slot (revoutfun (forward (input-vector)))))

(defun input-vector ()
  (mapcar #'(lambda (f-s) (car (fget-z (first f-s) (second f-s))))
          (generic-features)
          ))

(defun user-supplied (frame slot)
  (ask-alt frame slot)
  )

(defun sizing-system (&optional program)
  (initialize-sizing)
  (if program                 ; The data from an existing is needed
    (progn
          (if (not (framep 'nastran)) ; See if the data was loaded
             (progn
               (initialize-system)   ; if not load it.
               (initialize-sizing)))
          (if (framep program)
             (load-specifications program)
             (return-from sizing-system "BAD Program name"))))
  (fget-z 'program 'predicted-size)
  (sizing-report))
```

```
(defun load-specifications (program &aux val value frame slot)
 (dolist (f-s (specification-features) program)
  (setq frame (first f-s))
  (setq slot (second f-s))
  (setq val (fget program frame slot))
  (if (or (null val) (equal val '(n/a)))
        (fset frame slot 'value (car (last (fget frame slot 'range))))
        (fset frame slot 'value (car val)))
  (if (eq (setq value (car (fget frame slot 'value))) 'no)
        (fset frame slot 'certainty 0)
        (fset frame slot 'certainty 1))
  (if (neq value 'yes)
        (fset frame slot 'most-likely value))
 ))

(defun load-specification-certainties (program)
 (do* ((f-s (specification-features) (cdr f-s)) (f (caar f-s) (caar f-s))
        (s (cadar f-s) (cadar f-s)) val)
    ((null f-s) program)
  (setq val (car (fget program f s)))
  (if (or (null val) (eq val 'n/a)) (setq val (lowest-value f s)))
  (cond ((Vector-Spec s)
          (fset f s 'most-likely val)
          (fset f s 'certainty 1.))
         ((eq val 'yes)
          (fset f s 'certainty 1.))
         ((eq val 'no)
          (fset f s 'certainty 0.))
         (t
          (error "~s is invalid as a specification." val)))))

(defun sizing-report ()
 (format t "~%~30a           ~s" "Code Size (K lines)"
          (fget-z 'program 'predicted-size))
 (dolist (comp (fget-z 'sizing-criteria 'output-nodes))
  (format t "~% ~30s           ~s" comp
           (fget-z (first comp) (second comp)))))


;; This section contains the rules for controlling the determination of
;; the specifications.

(defvar implication
        '((structural-engineering structure) (turbulence fully-non-linear)
          (shock-waves turbulence) (video-images images) (essentially-none solids)
          (thin-shells solids) (shells-of-revol solids) (slabs solids)
          (thick-shells solids) (3D-structure present) (number-of-formats present)
          (perspective present) (hidden-line-removal present) (shading present)))

(defun initialize-specification-rules (&aux frames slots)
 (setq frames (remove-duplicates (mapcar 'car  (specification-features))))
 (setq slots  (remove-duplicates (mapcar 'cadr (specification-features))))
 (dolist (f frames)
  (dolist (s (fget-z f 'specification-slots))
   (fput f s 'if-added 'specification-dependence)))
 (dolist (imp implication)
  (fput (second imp) 'other-influences 'value (first imp))
  (fput (first imp) 'other-implications 'value (second imp)))
 )

(defun specification-dependence (frame slot &aux influences fr)
 (setq influences (union (fget slot 'specification-slots 'value)
                         (fget slot 'other-influences 'value)))
;(setq debug (instance 'debug))
```

```
;(fput-list debug 'args (list frame slot influences))
 (if (zerop (car (fget frame slot 'certainty))) ; If its definitely false
    (dolist (s influences)                  ; Look at the implications
         (setq fr (Find-Frame s))            ; Find the correct frame
         (cond ((vector-spec s)               ; For quantitative specs:
              (fset fr s 'most-likely        ; the lowest quantitative values
                        (car (last (fget fr s 'range))))
              (fset fr s 'certainty 1.))      ; are certain
             (t                              ; For qualitative specs:
              (fplace fr s 0.)))             ; they must also be false.
         (update-display fr s)))             ; Update if on screen
 )

(defun update-display (frame slot &aux Scroll Button App)
 (setq Scroll (car (fget frame slot 'ScrollBarObject)))
 (setq Button (car (fget frame slot 'ButtonGroupObject)))
 (setq App (car (fget 'code-sizing-system 'active-application 'value)))
 (cond ((and App Scroll (eq (send Scroll :ObjApp) App))
          (send Scroll :set-up frame slot)
          (if Button (send Button :set-up frame slot)))))


(defun fplace (frame slot certainty-value)
 (fclear frame slot 'certainty)
 (fput-p frame slot 'certainty certainty-value)
 )


(defun Vector-Spec (slot)
 (if (or (eq slot 'spatial-dimension)
          (equal "NUMBER" (subseq (string slot) 0 6))) t nil))

(defun Find-Frame (slot)
  (dolist (fs (specification-features))
   (if (eq slot (second fs)) (return-from Find-Frame (first fs)))))


;;; New additions as of 2 Mar 90

;; A utility for the lowest value

(defun lowest-value (f s)
 (car (last (fget f s 'range))))

;; Define routines for deterministic and stocastic values. The deterministic
;; version is first

(defun specification-value (f s)
 (dolist (imp (union f (fget s 'other-implications 'value)))
   (if (eq 'no (specification-value (Find-Frame imp) imp))
         (return-from specification-value (lowest-value f s))))
 (determine-spec-value f s))

(defun determine-spec-value (f s)
 (if (eq (car (fget-z 'code-sizing-system
                        'specification-statistics)) 'deterministic)
    (deterministic-value f s)
    (stocastic-value f s)))

(defun deterministic-value (f s)
 (if (Vector-Spec s)
    (car (fget f s 'most-likely))
    (if (> (car (fget f s 'certainty)) 0.5) 'Yes 'No)))
```

```
(defun stocastic-value (f s)
 (cond ((Vector-Spec s)
        (stocastic-vector-value f s))
       (t
        (if (> (car (fget f s 'certainty)) (random 1.0))
           'Yes No))))

(defun stocastic-vector-value (f s &aux (r (random 1.0)) cdist)
 (setq cdist (cdist f s))
 (dotimes (i (length cdist))
  (if (< r (nth i cdist))
       (return-from stocastic-vector-value
        (nth i (fget f s 'range))))))

(defun most-likely-value (f s &aux range likely)
 (setq range (fget f s 'range))
 (setq likely (car (fget f s 'most-likely)))
 (float (/ (position likely range) (1- (length range)))))

(defun cdist (f s &aux values cvalues (cum 0.) like cert del)
 (setq like (most-likely-value f s))
 (setq cert (car (fget f s 'certainty)))
 (setq del (/ 1 (1- (length (fget f s 'range)))))
 (dotimes (i (length (fget f s 'range)))
  (push-end (inc cum (num (* del i) like cert)) values))
 (dolist (v values cvalues)
  (push-end (/ v (car (last values))) cvalues)))

(defun display-disp (&optional (func 'pdf) &aux like val cert)
 (dotimes (i 5)                                    ; for each most likely value
  (setq like (* .25 i))
  (format t "~%~%~%  Most likely value is ~s" like)
  (format t "~%~%                certainty~%")
  (format t "~%value   ")
  (dotimes (c 5) (format t "~6,2f  " (* .25 c)))
  (format t "~%")
  (dotimes (v 5)
   (setq val (* .25 v))
   (format t "~%~6,2f  " val)
   (dotimes (c 5)                                  ; for each certainty value
       (setq cert (* .25 c))
       (format t "~6,4f  " (funcall func val like cert)))
  )))

(defun cdf (v l c &aux (cum 0.))
 (dotimes (i (1+ (* 4. v)) cum)
  (inc cum (pdf (/ i 4.) l c))))

(defun pdf (v l c) (/ (num v l c) (den l c)))

(defun num (value likely cert &aux (sharp 4.))
 (cond ((= cert 1)
        (if (= value likely) 1. 0.))
       (t
        (exp (* -1 (* (/ cert (- 1 cert)) sharp)
                (expt (- value likely) 2))))))

(defun den (likely cert &aux (tot 0.))
 (dolist (v '(0. 0.25 0.5 0.75 1.) tot) ; this is only valid for functions
  (inc tot (num v likely cert))))      ; that take on 5 different values
```

File hd:nasa:production system:implementation.lisp

```
;;; -*- Mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-

;;; This implementation was created on 09 May 90 12:22:36

;; The initialization function sets up frames for the
;; specifications and generic components of the program.

(defun initialize-frames nil
 (set-frame-list
  '((sizing-criteria
    (output-nodes
     (value (task complexity) (methods complexity) (methods generality) (methods accuracy)
          (methods efficiency) (methods redundancy) (objects complexity) (objects generality)
          (representation complexity) (representation capacity) (representation knowledge)
          (graphics complexity) (user-interface complexity) (data-management complexity)
          (sys-programming complexity) (hard-architecture complexity)))
    (children
     (value task engineering-program thermodynamics methods time-dependence linearity boundary
          objects solids structure continuous-volume shape-limitations fluids user-interface
          libraries graphics commercialization lumped-parameter database-program non-physical
          systems-programming representation data-management sys-programming hard-architecture)))
    (task (specification-slots (value engineering-program database-program))
    (engineering-program (if-needed user-supplied) (range yes no))
    (database-program (if-needed user-supplied) (range yes no))
    (generic-slots (value complexity)) (complexity (if-needed task-complexity)))
   (engineering-program
    (specification-slots
     (value structural-engineering electronic-circuit thermodynamics control-systems
          aerodynamics image-processing))
    (structural-engineering (if-needed user-supplied) (range yes no))
    (electronic-circuit (if-needed user-supplied) (range yes no))
    (thermodynamics (if-needed user-supplied) (range yes no))
    (control-systems (if-needed user-supplied) (range yes no))
    (aerodynamics (if-needed user-supplied) (range yes no))
    (image-processing (if-needed user-supplied) (range yes no)))
   (thermodynamics (specification-slots (value radiation conduction))
    (radiation (if-needed user-supplied) (range yes no))
    (conduction (if-needed user-supplied) (range yes no)))
   (methods (specification-slots (value number-redundant-choices))
    (number-redundant-choices (if-needed user-supplied)
     (range very-high high medium low very-low))
    (generic-slots (value complexity generality accuracy efficiency redundancy))
    (complexity (if-needed methods-complexity)) (generality (if-needed methods-generality))
    (accuracy (if-needed methods-accuracy)) (efficiency (if-needed methods-efficiency))
    (redundancy (if-needed methods-redundancy)))
   (time-dependence (specification-slots (value static quasi-static fully-dynamic))
    (static (if-needed user-supplied) (range yes no))
    (quasi-static (if-needed user-supplied) (range yes no))
    (fully-dynamic (if-needed user-supplied) (range yes no)))
   (linearity (specification-slots (value linear limited-non-linear fully-non-linear))
    (linear (if-needed user-supplied) (range yes no))
    (limited-non-linear (if-needed user-supplied) (range yes no))
    (fully-non-linear (if-needed user-supplied) (range yes no)))
   (boundary
    (specification-slots
     (value number-ways-to-specify 2d-field-on-3d-surface lumped-component-values))
    (number-ways-to-specify (if-needed user-supplied)
     (range very-high high medium low very-low))
    (2d-field-on-3d-surface (if-needed user-supplied) (range yes no))
    (lumped-component-values (if-needed user-supplied) (range yes no)))
   (objects (specification-slots (value solids fluids non-physical))
    (solids (if-needed user-supplied) (range yes no))
```

```
(fluids (if-needed user-supplied) (range yes no))
(non-physical (if-needed user-supplied) (range yes no))
(generic-slots (value complexity generality)) (complexity (if-needed objects-complexity))
(generality (if-needed objects-generality)))
(solids (specification-slots (value structure point-mass))
(structure (if-needed user-supplied) (range yes no))
(point-mass (if-needed user-supplied) (range yes no)))
(structure
 (specification-slots
  (value spatial-dimension lumped-parameter continuous-volume number-of-elements
      substructuring-capability))
 (spatial-dimension (if-needed user-supplied) (range 3 2 1 0))
 (lumped-parameter (if-needed user-supplied) (range yes no))
 (continuous-volume (if-needed user-supplied) (range yes no))
 (number-of-elements (if-needed user-supplied) (range very-high high medium low very-low))
 (substructuring-capability (if-needed user-supplied) (range yes no)))
(continuous-volume
 (specification-slots (value finite-elements grid number-of-known-shapes))
 (finite-elements (if-needed user-supplied) (range yes no))
 (grid (if-needed user-supplied) (range yes no))
 (number-of-known-shapes (if-needed user-supplied)
 (range very-high high medium low very-low)))
(shape-limitations
 (specification-slots
  (value essentially-none thin-shells shells-of-revol slabs thick-shells))
 (essentially-none (if-needed user-supplied) (range yes no))
 (thin-shells (if-needed user-supplied) (range yes no))
 (shells-of-revol (if-needed user-supplied) (range yes no))
 (slabs (if-needed user-supplied) (range yes no))
 (thick-shells (if-needed user-supplied) (range yes no)))
(fluids (specification-slots (value turbulence shock-waves))
 (turbulence (if-needed user-supplied) (range yes no))
 (shock-waves (if-needed user-supplied) (range yes no)))
(user-interface
 (specification-slots
  (value text-editor data-manipulator control-language user-specified-routines
      matrix-spec-language libraries))
 (text-editor (if-needed user-supplied) (range yes no))
 (data-manipulator (if-needed user-supplied) (range yes no))
 (control-language (if-needed user-supplied) (range yes no))
 (user-specified-routines (if-needed user-supplied) (range yes no))
 (matrix-spec-language (if-needed user-supplied) (range yes no))
 (libraries (if-needed user-supplied) (range yes no)) (generic-slots (value complexity))
 (complexity (if-needed user-interface-complexity)))
(libraries
 (specification-slots
  (value number-of-stored-shapes format-conversion geometric-conversion
      translate-for-other-progs))
 (number-of-stored-shapes (if-needed user-supplied)
 (range very-high high medium low very-low))
 (format-conversion (if-needed user-supplied) (range yes no))
 (geometric-conversion (if-needed user-supplied) (range yes no))
 (translate-for-other-progs (if-needed user-supplied) (range yes no)))
(graphics
 (specification-slots
  (value present 3d-structure number-of-formats perspective hidden-line-removal shading))
 (present (if-needed user-supplied) (range yes no))
 (3d-structure (if-needed user-supplied) (range yes no))
 (number-of-formats (if-needed user-supplied) (range very-high high medium low very-low))
 (perspective (if-needed user-supplied) (range yes no))
 (hidden-line-removal (if-needed user-supplied) (range yes no))
 (shading (if-needed user-supplied) (range yes no)) (generic-slots (value complexity))
 (complexity (if-needed graphics-complexity)))
(commercialization
```

```
    (specification-slots
     (value many-projects many-companies industry-standard many-upgrades many-computers))
    (many-projects (if-needed user-supplied) (range yes no))
    (many-companies (if-needed user-supplied) (range yes no))
    (industry-standard (if-needed user-supplied) (range yes no))
    (many-upgrades (if-needed user-supplied) (range yes no))
    (many-computers (if-needed user-supplied) (range yes no)))
   (lumped-parameter (specification-slots (value number-known-components))
    (number-known-components (if-needed user-supplied)
     (range very-high high medium low very-low)))
   (database-program (specification-slots (value video-images))
    (video-images (if-needed user-supplied) (range yes no)))
   (non-physical (specification-slots (value images))
    (images (if-needed user-supplied) (range yes no)))
   (systems-programming (specification-slots (value execute-commands code-to-save-memory))
    (execute-commands (if-needed user-supplied) (range yes no))
    (code-to-save-memory (if-needed user-supplied) (range yes no)))
   (representation (generic-slots (value complexity capacity knowledge))
    (complexity (if-needed representation-complexity))
    (capacity (if-needed representation-capacity))
    (knowledge (if-needed representation-knowledge)))
   (data-management (generic-slots (value complexity))
    (complexity (if-needed data-management-complexity)))
   (sys-programming (generic-slots (value complexity))
    (complexity (if-needed sys-programming-complexity)))
   (hard-architecture (generic-slots (value complexity))
    (complexity (if-needed hard-architecture-complexity))))))


;; This section contains the IF-NEEDED routines


;; The IF-NEEDED routine for frame TASK, slot COMPLEXITY

(defun task-complexity (frame slot &aux val)
 (setq val 0.079200014)
 (iff 'task 'engineering-program 'yes (inc val 0.15677664))
 (iff 'task 'database-program 'yes (inc val 0.15781765))
 (iff 'engineering-program 'structural-engineering 'yes (inc val 0.019139774))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val 0.042799994))
 (iff 'engineering-program 'thermodynamics 'yes (inc val 0.058112793))
 (iff 'engineering-program 'control-systems 'yes (inc val 0.13921365))
 (iff 'engineering-program 'aerodynamics 'yes (inc val -0.07263435))
 (iff 'engineering-program 'image-processing 'yes (inc val 0.06377641))
 (iff 'thermodynamics 'radiation 'yes (inc val -0.23982486))
 (iff 'thermodynamics 'conduction 'yes (inc val 0.027954046))
 (iff 'methods 'number-redundant-choices 'very-high (inc val 0.14098752))
 (iff 'methods 'number-redundant-choices 'high (inc val 0.011199996))
 (iff 'methods 'number-redundant-choices 'medium (inc val -0.13277595))
 (iff 'methods 'number-redundant-choices 'low (inc val -0.047721017))
 (iff 'methods 'number-redundant-choices 'very-low (inc val -0.08058988))
 (iff 'time-dependence 'static 'yes (inc val -0.024401616))
 (iff 'time-dependence 'quasi-static 'yes (inc val -0.07677092))
 (iff 'time-dependence 'fully-dynamic 'yes (inc val 0.0033372478))
 (iff 'linearity 'linear 'yes (inc val -0.061736364))
 (iff 'linearity 'limited-non-linear 'yes (inc val 0.105950244))
 (iff 'linearity 'fully-non-linear 'yes (inc val -0.03931454))
 (iff 'boundary 'number-ways-to-specify 'very-high (inc val 0.14740701))
 (iff 'boundary 'number-ways-to-specify 'high (inc val -0.0868))
 (iff 'boundary 'number-ways-to-specify 'medium (inc val 0.1928287))
 (iff 'boundary 'number-ways-to-specify 'low (inc val -0.037133746))
 (iff 'boundary 'number-ways-to-specify 'very-low (inc val -0.035601236))
 (iff 'boundary '2d-field-on-3d-surface 'yes (inc val 0.12747107))
 (iff 'boundary 'lumped-component-values 'yes (inc val -0.1020541))
```

```
(iff 'objects 'solids 'yes (inc val -0.09371934))
(iff 'objects 'fluids 'yes (inc val 0.17970333))
(iff 'objects 'non-physical 'yes (inc val 0.1668071))
(iff 'solids 'structure 'yes (inc val -0.08811934))
(iff 'solids 'point-mass 'yes (inc val -0.051429614))
(inc val (* (car (fget-z 'structure 'spatial-dimension)) 0.120865755))
(iff 'structure 'lumped-parameter 'yes (inc val -0.01242457))
(iff 'structure 'continuous-volume 'yes (inc val -0.015260225))
(iff 'structure 'number-of-elements 'very-high (inc val -0.030325852))
(iff 'structure 'number-of-elements 'high (inc val 0.029456824))
(iff 'structure 'number-of-elements 'medium (inc val 0.18135875))
(iff 'structure 'number-of-elements 'low (inc val 0.04979097))
(iff 'structure 'number-of-elements 'very-low (inc val 0.021620085))
(iff 'structure 'substructuring-capability 'yes (inc val 0.0060500293))
(iff 'continuous-volume 'finite-elements 'yes (inc val 0.026494816))
(iff 'continuous-volume 'grid 'yes (inc val 0.2353526))
(iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val 0.05121107))
(iff 'continuous-volume 'number-of-known-shapes 'high (inc val -0.04873027))
(iff 'continuous-volume 'number-of-known-shapes 'medium (inc val -0.12382611))
(iff 'continuous-volume 'number-of-known-shapes 'low (inc val 0.13640574))
(iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val -0.06475973))
(iff 'shape-limitations 'essentially-none 'yes (inc val -0.065857574))
(iff 'shape-limitations 'thin-shells 'yes (inc val 0.013648725))
(iff 'shape-limitations 'shells-of-revol 'yes (inc val -0.13156763))
(iff 'shape-limitations 'slabs 'yes (inc val -0.013161882))
(iff 'shape-limitations 'thick-shells 'yes (inc val 0.0104323365))
(iff 'fluids 'turbulence 'yes (inc val 0.00576568))
(iff 'fluids 'shock-waves 'yes (inc val 0.15635519))
(iff 'user-interface 'text-editor 'yes (inc val 0.14335294))
(iff 'user-interface 'data-manipulator 'yes (inc val 0.16736823))
(iff 'user-interface 'control-language 'yes (inc val -0.102388844))
(iff 'user-interface 'user-specified-routines 'yes (inc val -0.0132333245))
(iff 'user-interface 'matrix-spec-language 'yes (inc val 0.094693676))
(iff 'user-interface 'libraries 'yes (inc val -0.032224815))
(iff 'libraries 'number-of-stored-shapes 'very-high (inc val -0.106047876))
(iff 'libraries 'number-of-stored-shapes 'high (inc val 0.17377642))
(iff 'libraries 'number-of-stored-shapes 'medium (inc val -0.0071853255))
(iff 'libraries 'number-of-stored-shapes 'low (inc val 0.039833598))
(iff 'libraries 'number-of-stored-shapes 'very-low (inc val 0.01952392))
(iff 'libraries 'format-conversion 'yes (inc val 0.07133655))
(iff 'libraries 'geometric-conversion 'yes (inc val -0.124211825))
(iff 'libraries 'translate-for-other-progs 'yes (inc val 0.015730578))
(iff 'graphics 'present 'yes (inc val 0.0311605))
(iff 'graphics '3d-structure 'yes (inc val -0.04287205))
(iff 'graphics 'number-of-formats 'very-high (inc val -0.18548447))
(iff 'graphics 'number-of-formats 'high (inc val -0.1024))
(iff 'graphics 'number-of-formats 'medium (inc val 0.0015999973))
(iff 'graphics 'number-of-formats 'low (inc val -0.12234582))
(iff 'graphics 'number-of-formats 'very-low (inc val -0.11946896))
(iff 'graphics 'perspective 'yes (inc val 0.07888417))
(iff 'graphics 'hidden-line-removal 'yes (inc val -0.019601973))
(iff 'graphics 'shading 'yes (inc val 0.02678696))
(iff 'commercialization 'many-projects 'yes (inc val 0.08515973))
(iff 'commercialization 'many-companies 'yes (inc val 0.037021447))
(iff 'commercialization 'industry-standard 'yes (inc val -0.01137741))
(iff 'commercialization 'many-upgrades 'yes (inc val 0.010579169))
(iff 'commercialization 'many-computers 'yes (inc val 0.12892595))
(iff 'lumped-parameter 'number-known-components 'very-high (inc val 0.16705687))
(iff 'lumped-parameter 'number-known-components 'high (inc val -0.12))
(iff 'lumped-parameter 'number-known-components 'medium (inc val 0.08038825))
(iff 'lumped-parameter 'number-known-components 'low (inc val 0.1955552))
(iff 'lumped-parameter 'number-known-components 'very-low (inc val 0.06450044))
(iff 'database-program 'video-images 'yes (inc val 0.036917474))
(iff 'non-physical 'images 'yes (inc val -0.15429556))
```

```
 (iff 'systems-programming 'execute-commands 'yes (inc val 0.05912253))
 (iff 'systems-programming 'code-to-save-memory 'yes (inc val -0.070472054))
 (fput-list frame slot val))


;; The IF-NEEDED routine for frame METHODS, slot COMPLEXITY

(defun methods-complexity (frame slot &aux val)
 (setq val -0.1388)
 (iff 'task 'engineering-program 'yes (inc val -0.028632497))
 (iff 'task 'database-program 'yes (inc val -0.07830586))
 (iff 'engineering-program 'structural-engineering 'yes (inc val 0.08072675))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val -0.1996))
 (iff 'engineering-program 'thermodynamics 'yes (inc val 0.09706881))
 (iff 'engineering-program 'control-systems 'yes (inc val 0.030445073))
 (iff 'engineering-program 'aerodynamics 'yes (inc val -0.1532452))
 (iff 'engineering-program 'image-processing 'yes (inc val 0.08547639))
 (iff 'thermodynamics 'radiation 'yes (inc val -0.1706279))
 (iff 'thermodynamics 'conduction 'yes (inc val 0.054566722))
 (iff 'methods 'number-redundant-choices 'very-high (inc val -0.0051239072))
 (iff 'methods 'number-redundant-choices 'high (inc val 0.19000001))
 (iff 'methods 'number-redundant-choices 'medium (inc val -0.10803329))
 (iff 'methods 'number-redundant-choices 'low (inc val -0.06127314))
 (iff 'methods 'number-redundant-choices 'very-low (inc val 0.051157963))
 (iff 'time-dependence 'static 'yes (inc val 0.077936254))
 (iff 'time-dependence 'quasi-static 'yes (inc val 0.18425997))
 (iff 'time-dependence 'fully-dynamic 'yes (inc val 0.05408548))
 (iff 'linearity 'linear 'yes (inc val -0.058386214))
 (iff 'linearity 'limited-non-linear 'yes (inc val -0.16382642))
 (iff 'linearity 'fully-non-linear 'yes (inc val -0.06889983))
 (iff 'boundary 'number-ways-to-specify 'very-high (inc val -0.13217136))
 (iff 'boundary 'number-ways-to-specify 'high (inc val 0.08440001))
 (iff 'boundary 'number-ways-to-specify 'medium (inc val -0.117156096))
 (iff 'boundary 'number-ways-to-specify 'low (inc val -0.09296545))
 (iff 'boundary 'number-ways-to-specify 'very-low (inc val 0.16582043))
 (iff 'boundary '2d-field-on-3d-surface 'yes (inc val 0.09346862))
 (iff 'boundary 'lumped-component-values 'yes (inc val 0.025555577))
 (iff 'objects 'solids 'yes (inc val 0.12141921))
 (iff 'objects 'fluids 'yes (inc val -0.009548504))
 (iff 'objects 'non-physical 'yes (inc val -0.10988109))
 (iff 'solids 'structure 'yes (inc val 0.10981921))
 (iff 'solids 'point-mass 'yes (inc val 0.19901414))
 (inc val (* (car (fget-z 'structure 'spatial-dimension)) 0.039295632))
 (iff 'structure 'lumped-parameter 'yes (inc val 0.0429597))
 (iff 'structure 'continuous-volume 'yes (inc val 0.25432676))
 (iff 'structure 'number-of-elements 'very-high (inc val -0.08830488))
 (iff 'structure 'number-of-elements 'high (inc val -0.1444475))
 (iff 'structure 'number-of-elements 'medium (inc val -0.125098))
 (iff 'structure 'number-of-elements 'low (inc val -0.1699306))
 (iff 'structure 'number-of-elements 'very-low (inc val -0.08389172))
 (iff 'structure 'substructuring-capability 'yes (inc val 0.044191316))
 (iff 'continuous-volume 'finite-elements 'yes (inc val -0.079852015))
 (iff 'continuous-volume 'grid 'yes (inc val 0.11916547))
 (iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val -0.077800274))
 (iff 'continuous-volume 'number-of-known-shapes 'high (inc val 0.107544325))
 (iff 'continuous-volume 'number-of-known-shapes 'medium (inc val 0.005254544))
 (iff 'continuous-volume 'number-of-known-shapes 'low (inc val 0.09910422))
 (iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val 0.19382475))
 (iff 'shape-limitations 'essentially-none 'yes (inc val 0.033077274))
 (iff 'shape-limitations 'thin-shells 'yes (inc val 0.032758866))
 (iff 'shape-limitations 'shells-of-revol 'yes (inc val -0.07771117))
 (iff 'shape-limitations 'slabs 'yes (inc val 0.113793015))
 (iff 'shape-limitations 'thick-shells 'yes (inc val -0.028111165))
 (iff 'fluids 'turbulence 'yes (inc val 0.11395475))
```

```lisp
 (iff 'fluids 'shock-waves 'yes (inc val 0.056043297))
 (iff 'user-interface 'text-editor 'yes (inc val 0.17489976))
 (iff 'user-interface 'data-manipulator 'yes (inc val 0.06185607))
 (iff 'user-interface 'control-language 'yes (inc val 0.07541698))
 (iff 'user-interface 'user-specified-routines 'yes (inc val -0.04219431))
 (iff 'user-interface 'matrix-spec-language 'yes (inc val 0.19685344))
 (iff 'user-interface 'libraries 'yes (inc val -0.048570663))
 (iff 'libraries 'number-of-stored-shapes 'very-high (inc val -0.05513018))
 (iff 'libraries 'number-of-stored-shapes 'high (inc val 0.17107634))
 (iff 'libraries 'number-of-stored-shapes 'medium (inc val -0.11492747))
 (iff 'libraries 'number-of-stored-shapes 'low (inc val 0.044188038))
 (iff 'libraries 'number-of-stored-shapes 'very-low (inc val -0.06727925))
 (iff 'libraries 'format-conversion 'yes (inc val 0.0073317504))
 (iff 'libraries 'geometric-conversion 'yes (inc val 0.06369716))
 (iff 'libraries 'translate-for-other-progs 'yes (inc val -0.09920703))
 (iff 'graphics 'present 'yes (inc val -0.046199102))
 (iff 'graphics '3d-structure 'yes (inc val -0.014345068))
 (iff 'graphics 'number-of-formats 'very-high (inc val -0.06564317))
 (iff 'graphics 'number-of-formats 'high (inc val 0.0748))
 (iff 'graphics 'number-of-formats 'medium (inc val -0.1948))
 (iff 'graphics 'number-of-formats 'low (inc val -0.15065828))
 (iff 'graphics 'number-of-formats 'very-low (inc val 0.076629005))
 (iff 'graphics 'perspective 'yes (inc val -0.15897526))
 (iff 'graphics 'hidden-line-removal 'yes (inc val -0.08458962))
 (iff 'graphics 'shading 'yes (inc val 0.033210654))
 (iff 'commercialization 'many-projects 'yes (inc val -0.042342562))
 (iff 'commercialization 'many-companies 'yes (inc val 0.24450356))
 (iff 'commercialization 'industry-standard 'yes (inc val -0.061316688))
 (iff 'commercialization 'many-upgrades 'yes (inc val 0.1168513))
 (iff 'commercialization 'many-computers 'yes (inc val 0.11430409))
 (iff 'lumped-parameter 'number-known-components 'very-high (inc val -0.11484734))
 (iff 'lumped-parameter 'number-known-components 'high (inc val 0.039599985))
 (iff 'lumped-parameter 'number-known-components 'medium (inc val -0.068408296))
 (iff 'lumped-parameter 'number-known-components 'low (inc val -0.21275665))
 (iff 'lumped-parameter 'number-known-components 'very-low (inc val -0.05646009))
 (iff 'database-program 'video-images 'yes (inc val 0.11854644))
 (iff 'non-physical 'images 'yes (inc val -0.078642905))
 (iff 'systems-programming 'execute-commands 'yes (inc val 0.15579277))
 (iff 'systems-programming 'code-to-save-memory 'yes (inc val -0.11634206))
 (fput-list frame slot val))


;; The IF-NEEDED routine for frame METHODS, slot GENERALITY

(defun methods-generality (frame slot &aux val)
 (setq val 0.0888)
 (iff 'task 'engineering-program 'yes (inc val -0.005989452))
 (iff 'task 'database-program 'yes (inc val 0.19630213))
 (iff 'engineering-program 'structural-engineering 'yes (inc val -0.113902844))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val -0.13080001))
 (iff 'engineering-program 'thermodynamics 'yes (inc val 0.033416618))
 (iff 'engineering-program 'control-systems 'yes (inc val -0.041496042))
 (iff 'engineering-program 'aerodynamics 'yes (inc val -0.14834066))
 (iff 'engineering-program 'image-processing 'yes (inc val -0.09401288))
 (iff 'thermodynamics 'radiation 'yes (inc val 0.14131111))
 (iff 'thermodynamics 'conduction 'yes (inc val -0.08151603))
 (iff 'methods 'number-redundant-choices 'very-high (inc val -0.0011838083))
 (iff 'methods 'number-redundant-choices 'high (inc val 0.0944))
 (iff 'methods 'number-redundant-choices 'medium (inc val -0.0016423305))
 (iff 'methods 'number-redundant-choices 'low (inc val -0.096568964))
 (iff 'methods 'number-redundant-choices 'very-low (inc val 0.04028693))
 (iff 'time-dependence 'static 'yes (inc val 0.016413253))
 (iff 'time-dependence 'quasi-static 'yes (inc val 0.1359455))
 (iff 'time-dependence 'fully-dynamic 'yes (inc val -0.118272044))
```

```
(iff 'linearity 'linear 'yes (inc val -0.040885855))
(iff 'linearity 'limited-non-linear 'yes (inc val -0.10534706))
(iff 'linearity 'fully-non-linear 'yes (inc val 0.20289022))
(iff 'boundary 'number-ways-to-specify 'very-high (inc val 0.07296366))
(iff 'boundary 'number-ways-to-specify 'high (inc val -0.022799999))
(iff 'boundary 'number-ways-to-specify 'medium (inc val 0.09169702))
(iff 'boundary 'number-ways-to-specify 'low (inc val -2.1243904e-4))
(iff 'boundary 'number-ways-to-specify 'very-low (inc val 0.019643523))
(iff 'boundary '2d-field-on-3d-surface 'yes (inc val 0.07101585))
(iff 'boundary 'lumped-component-values 'yes (inc val -0.0907971))
(iff 'objects 'solids 'yes (inc val 0.10552213))
(iff 'objects 'fluids 'yes (inc val 0.17016484))
(iff 'objects 'non-physical 'yes (inc val -0.06800818))
(iff 'solids 'structure 'yes (inc val -0.012477847))
(iff 'solids 'point-mass 'yes (inc val 0.04162043))
(inc val (* (car (fget-z 'structure 'spatial-dimension)) -0.09920261))
(iff 'structure 'lumped-parameter 'yes (inc val 0.057116326))
(iff 'structure 'continuous-volume 'yes (inc val 0.15929717))
(iff 'structure 'number-of-elements 'very-high (inc val -0.09436259))
(iff 'structure 'number-of-elements 'high (inc val -0.12324172))
(iff 'structure 'number-of-elements 'medium (inc val 0.011332675))
(iff 'structure 'number-of-elements 'low (inc val -0.030606182))
(iff 'structure 'number-of-elements 'very-low (inc val -0.087030336))
(iff 'structure 'substructuring-capability 'yes (inc val -0.10060151))
(iff 'continuous-volume 'finite-elements 'yes (inc val 0.13527253))
(iff 'continuous-volume 'grid 'yes (inc val 0.14548151))
(iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val 0.04442904))
(iff 'continuous-volume 'number-of-known-shapes 'high (inc val 0.0829182))
(iff 'continuous-volume 'number-of-known-shapes 'medium (inc val 0.018090134))
(iff 'continuous-volume 'number-of-known-shapes 'low (inc val -0.19890563))
(iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val -0.03084001))
(iff 'shape-limitations 'essentially-none 'yes (inc val 0.13338448))
(iff 'shape-limitations 'thin-shells 'yes (inc val 0.016616052))
(iff 'shape-limitations 'shells-of-revol 'yes (inc val 0.15623766))
(iff 'shape-limitations 'slabs 'yes (inc val 0.14653209))
(iff 'shape-limitations 'thick-shells 'yes (inc val -0.06816234))
(iff 'fluids 'turbulence 'yes (inc val -0.09474069))
(iff 'fluids 'shock-waves 'yes (inc val 0.18892597))
(iff 'user-interface 'text-editor 'yes (inc val 0.057116974))
(iff 'user-interface 'data-manipulator 'yes (inc val 0.09492691))
(iff 'user-interface 'control-language 'yes (inc val 0.11259196))
(iff 'user-interface 'user-specified-routines 'yes (inc val 0.14128989))
(iff 'user-interface 'matrix-spec-language 'yes (inc val -0.10504371))
(iff 'user-interface 'libraries 'yes (inc val -0.072007746))
(iff 'libraries 'number-of-stored-shapes 'very-high (inc val -0.085434794))
(iff 'libraries 'number-of-stored-shapes 'high (inc val -0.073612876))
(iff 'libraries 'number-of-stored-shapes 'medium (inc val -0.17041384))
(iff 'libraries 'number-of-stored-shapes 'low (inc val -0.06367839))
(iff 'libraries 'number-of-stored-shapes 'very-low (inc val -0.052368395))
(iff 'libraries 'format-conversion 'yes (inc val -0.15220678))
(iff 'libraries 'geometric-conversion 'yes (inc val -0.09129375))
(iff 'libraries 'translate-for-other-progs 'yes (inc val 0.12006205))
(iff 'graphics 'present 'yes (inc val 0.13865939))
(iff 'graphics '3d-structure 'yes (inc val -0.095259696))
(iff 'graphics 'number-of-formats 'very-high (inc val 0.13397463))
(iff 'graphics 'number-of-formats 'high (inc val -0.056799993))
(iff 'graphics 'number-of-formats 'medium (inc val 0.03199999))
(iff 'graphics 'number-of-formats 'low (inc val 0.037125748))
(iff 'graphics 'number-of-formats 'very-low (inc val 0.061791446))
(iff 'graphics 'perspective 'yes (inc val 0.15634784))
(iff 'graphics 'hidden-line-removal 'yes (inc val -0.13654874))
(iff 'graphics 'shading 'yes (inc val -0.11097779))
(iff 'commercialization 'many-projects 'yes (inc val 0.1895556))
(iff 'commercialization 'many-companies 'yes (inc val 0.16852641))
```

```
 (iff 'commercialization 'industry-standard 'yes (inc val 0.15385455))
 (iff 'commercialization 'many-upgrades 'yes (inc val 0.1706947))
 (iff 'commercialization 'many-computers 'yes (inc val -0.03324728))
 (iff 'lumped-parameter 'number-known-components 'very-high (inc val -0.0036417688))
 (iff 'lumped-parameter 'number-known-components 'high (inc val 0.0996))
 (iff 'lumped-parameter 'number-known-components 'medium (inc val 0.08972102))
 (iff 'lumped-parameter 'number-known-components 'low (inc val -0.13467404))
 (iff 'lumped-parameter 'number-known-components 'very-low (inc val -0.06251345))
 (iff 'database-program 'video-images 'yes (inc val 0.16052331))
 (iff 'non-physical 'images 'yes (inc val 0.07634551))
 (iff 'systems-programming 'execute-commands 'yes (inc val 0.18354905))
 (iff 'systems-programming 'code-to-save-memory 'yes (inc val -0.22274339))
 (fput-list frame slot val))


;; The IF-NEEDED routine for frame METHODS, slot ACCURACY

(defun methods-accuracy (frame slot &aux val)
 (setq val -0.1688)
 (iff 'task 'engineering-program 'yes (inc val 0.07603234))
 (iff 'task 'database-program 'yes (inc val 0.12241937))
 (iff 'engineering-program 'structural-engineering 'yes (inc val 0.052200273))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val 0.11080001))
 (iff 'engineering-program 'thermodynamics 'yes (inc val -0.05048059))
 (iff 'engineering-program 'control-systems 'yes (inc val 0.018817648))
 (iff 'engineering-program 'aerodynamics 'yes (inc val 0.1600446))
 (iff 'engineering-program 'image-processing 'yes (inc val 0.112680875))
 (iff 'thermodynamics 'radiation 'yes (inc val -0.039222192))
 (iff 'thermodynamics 'conduction 'yes (inc val -0.008881149))
 (iff 'methods 'number-redundant-choices 'very-high (inc val -0.123594))
 (iff 'methods 'number-redundant-choices 'high (inc val -0.06799999))
 (iff 'methods 'number-redundant-choices 'medium (inc val -0.10267133))
 (iff 'methods 'number-redundant-choices 'low (inc val 0.14690791))
 (iff 'methods 'number-redundant-choices 'very-low (inc val 0.034445126))
 (iff 'time-dependence 'static 'yes (inc val -0.08690371))
 (iff 'time-dependence 'quasi-static 'yes (inc val 0.2227582))
 (iff 'time-dependence 'fully-dynamic 'yes (inc val 0.23380016))
 (iff 'linearity 'linear 'yes (inc val -0.11818694))
 (iff 'linearity 'limited-non-linear 'yes (inc val 0.06090113))
 (iff 'linearity 'fully-non-linear 'yes (inc val -0.022094158))
 (iff 'boundary 'number-ways-to-specify 'very-high (inc val 0.013670428))
 (iff 'boundary 'number-ways-to-specify 'high (inc val 0.039599985))
 (iff 'boundary 'number-ways-to-specify 'medium (inc val -0.15387535))
 (iff 'boundary 'number-ways-to-specify 'low (inc val 0.04443509))
 (iff 'boundary 'number-ways-to-specify 'very-low (inc val 0.08445763))
 (iff 'boundary '2d-field-on-3d-surface 'yes (inc val 0.12698595))
 (iff 'boundary 'lumped-component-values 'yes (inc val -0.080665894))
 (iff 'objects 'solids 'yes (inc val -0.05947474))
 (iff 'objects 'fluids 'yes (inc val 0.20398618))
 (iff 'objects 'non-physical 'yes (inc val 0.15020232))
 (iff 'solids 'structure 'yes (inc val -0.09587473))
 (iff 'solids 'point-mass 'yes (inc val 0.09835588))
 (inc val (* (car (fget-z 'structure 'spatial-dimension)) 0.17637646))
 (iff 'structure 'lumped-parameter 'yes (inc val 0.041296262))
 (iff 'structure 'continuous-volume 'yes (inc val -0.17899972))
 (iff 'structure 'number-of-elements 'very-high (inc val -0.12233887))
 (iff 'structure 'number-of-elements 'high (inc val 0.13093439))
 (iff 'structure 'number-of-elements 'medium (inc val -0.13519949))
 (iff 'structure 'number-of-elements 'low (inc val 0.009129278))
 (iff 'structure 'number-of-elements 'very-low (inc val 0.11256244))
 (iff 'structure 'substructuring-capability 'yes (inc val 0.18433678))
 (iff 'continuous-volume 'finite-elements 'yes (inc val -0.101839475))
 (iff 'continuous-volume 'grid 'yes (inc val 0.070884615))
 (iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val 0.14332508))
```

```
 (iff 'continuous-volume 'number-of-known-shapes 'high (inc val -0.13911773))
 (iff 'continuous-volume 'number-of-known-shapes 'medium (inc val -0.17319864))
 (iff 'continuous-volume 'number-of-known-shapes 'low (inc val 0.06279647))
 (iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val -0.02951745))
 (iff 'shape-limitations 'essentially-none 'yes (inc val -0.051895153))
 (iff 'shape-limitations 'thin-shells 'yes (inc val -0.0603809))
 (iff 'shape-limitations 'shells-of-revol 'yes (inc val 0.05884185))
 (iff 'shape-limitations 'slabs 'yes (inc val -0.08956168))
 (iff 'shape-limitations 'thick-shells 'yes (inc val 0.056841858))
 (iff 'fluids 'turbulence 'yes (inc val 0.04444464))
 (iff 'fluids 'shock-waves 'yes (inc val -0.13702382))
 (iff 'user-interface 'text-editor 'yes (inc val 0.023529653))
 (iff 'user-interface 'data-manipulator 'yes (inc val 0.034130026))
 (iff 'user-interface 'control-language 'yes (inc val -0.101721056))
 (iff 'user-interface 'user-specified-routines 'yes (inc val 0.017501237))
 (iff 'user-interface 'matrix-spec-language 'yes (inc val -0.14989397))
 (iff 'user-interface 'libraries 'yes (inc val 0.05444576))
 (iff 'libraries 'number-of-stored-shapes 'very-high (inc val 0.0055080005))
 (iff 'libraries 'number-of-stored-shapes 'high (inc val -0.17291915))
 (iff 'libraries 'number-of-stored-shapes 'medium (inc val -0.0563705))
 (iff 'libraries 'number-of-stored-shapes 'low (inc val 0.15143934))
 (iff 'libraries 'number-of-stored-shapes 'very-low (inc val 0.13022995))
 (iff 'libraries 'format-conversion 'yes (inc val 0.10419422))
 (iff 'libraries 'geometric-conversion 'yes (inc val -0.10615036))
 (iff 'libraries 'translate-for-other-progs 'yes (inc val -0.07047724))
 (iff 'graphics 'present 'yes (inc val 0.11480762))
 (iff 'graphics '3d-structure 'yes (inc val -0.006183359))
 (iff 'graphics 'number-of-formats 'very-high (inc val -0.12737112))
 (iff 'graphics 'number-of-formats 'high (inc val 0.002000004))
 (iff 'graphics 'number-of-formats 'medium (inc val -0.114))
 (iff 'graphics 'number-of-formats 'low (inc val 0.05666055))
 (iff 'graphics 'number-of-formats 'very-low (inc val 0.0025982661))
 (iff 'graphics 'perspective 'yes (inc val 0.09222456))
 (iff 'graphics 'hidden-line-removal 'yes (inc val -0.011434924))
 (iff 'graphics 'shading 'yes (inc val -0.11676))
 (iff 'commercialization 'many-projects 'yes (inc val 0.22130479))
 (iff 'commercialization 'many-companies 'yes (inc val -0.060107388))
 (iff 'commercialization 'industry-standard 'yes (inc val 0.25500524))
 (iff 'commercialization 'many-upgrades 'yes (inc val -0.017417356))
 (iff 'commercialization 'many-computers 'yes (inc val -0.06108398))
 (iff 'lumped-parameter 'number-known-components 'very-high (inc val 0.0597343))
 (iff 'lumped-parameter 'number-known-components 'high (inc val -0.049600005))
 (iff 'lumped-parameter 'number-known-components 'medium (inc val 0.19180804))
 (iff 'lumped-parameter 'number-known-components 'low (inc val -0.0958239))
 (iff 'lumped-parameter 'number-known-components 'very-low (inc val 0.045769222))
 (iff 'database-program 'video-images 'yes (inc val 0.076463796))
 (iff 'non-physical 'images 'yes (inc val -0.0086962255))
 (iff 'systems-programming 'execute-commands 'yes (inc val 0.10494009))
 (iff 'systems-programming 'code-to-save-memory 'yes (inc val -0.077667736))
 (fput-list frame slot val))


;; The IF-NEEDED routine for frame METHODS, slot EFFICIENCY

(defun methods-efficiency (frame slot &aux val)
 (setq val 0.10759999)
 (iff 'task 'engineering-program 'yes (inc val -0.06683519))
 (iff 'task 'database-program 'yes (inc val 0.016145082))
 (iff 'engineering-program 'structural-engineering 'yes (inc val -0.05050206))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val 0.0012000054))
 (iff 'engineering-program 'thermodynamics 'yes (inc val -0.14712231))
 (iff 'engineering-program 'control-systems 'yes (inc val -0.02432915))
 (iff 'engineering-program 'aerodynamics 'yes (inc val 0.11588017))
 (iff 'engineering-program 'image-processing 'yes (inc val -0.09693154))
```

```
(iff 'thermodynamics 'radiation 'yes (inc val -0.0012443416))
(iff 'thermodynamics 'conduction 'yes (inc val 0.10605081))
(iff 'methods 'number-redundant-choices 'very-high (inc val 0.1320479))
(iff 'methods 'number-redundant-choices 'high (inc val -0.075600006))
(iff 'methods 'number-redundant-choices 'medium (inc val 0.039588567))
(iff 'methods 'number-redundant-choices 'low (inc val -0.09849431))
(iff 'methods 'number-redundant-choices 'very-low (inc val -0.06747445))
(iff 'time-dependence 'static 'yes (inc val -0.15473932))
(iff 'time-dependence 'quasi-static 'yes (inc val 0.1523343))
(iff 'time-dependence 'fully-dynamic 'yes (inc val 0.14258836))
(iff 'linearity 'linear 'yes (inc val 0.04253627))
(iff 'linearity 'limited-non-linear 'yes (inc val -0.008571226))
(iff 'linearity 'fully-non-linear 'yes (inc val 0.22032627))
(iff 'boundary 'number-ways-to-specify 'very-high (inc val -0.015949003))
(iff 'boundary 'number-ways-to-specify 'high (inc val -0.1412))
(iff 'boundary 'number-ways-to-specify 'medium (inc val 0.11219448))
(iff 'boundary 'number-ways-to-specify 'low (inc val 0.02600634))
(iff 'boundary 'number-ways-to-specify 'very-low (inc val -0.14098419))
(iff 'boundary '2d-field-on-3d-surface 'yes (inc val 0.1471577))
(iff 'boundary 'lumped-component-values 'yes (inc val -4.1327953e-4))
(iff 'objects 'solids 'yes (inc val 0.0059080753))
(iff 'objects 'fluids 'yes (inc val -0.09839785))
(iff 'objects 'non-physical 'yes (inc val 0.1369036))
(iff 'solids 'structure 'yes (inc val -0.036891922))
(iff 'solids 'point-mass 'yes (inc val 0.031696748))
(inc val (* (car (fget-z 'structure 'spatial-dimension)) -0.06939353))
(iff 'structure 'lumped-parameter 'yes (inc val 0.13746154))
(iff 'structure 'continuous-volume 'yes (inc val 0.023897922))
(iff 'structure 'number-of-elements 'very-high (inc val 0.10145831))
(iff 'structure 'number-of-elements 'high (inc val -0.10182956))
(iff 'structure 'number-of-elements 'medium (inc val -0.07677313))
(iff 'structure 'number-of-elements 'low (inc val 0.019452458))
(iff 'structure 'number-of-elements 'very-low (inc val -0.025040355))
(iff 'structure 'substructuring-capability 'yes (inc val 0.1877441))
(iff 'continuous-volume 'finite-elements 'yes (inc val 0.11475524))
(iff 'continuous-volume 'grid 'yes (inc val 0.13542405))
(iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val -0.1474206))
(iff 'continuous-volume 'number-of-known-shapes 'high (inc val -0.14822783))
(iff 'continuous-volume 'number-of-known-shapes 'medium (inc val -0.01992266))
(iff 'continuous-volume 'number-of-known-shapes 'low (inc val -0.026193496))
(iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val 0.1794322))
(iff 'shape-limitations 'essentially-none 'yes (inc val -0.033140313))
(iff 'shape-limitations 'thin-shells 'yes (inc val -0.02803845))
(iff 'shape-limitations 'shells-of-revol 'yes (inc val -0.09636725))
(iff 'shape-limitations 'slabs 'yes (inc val -0.12936069))
(iff 'shape-limitations 'thick-shells 'yes (inc val 0.11163273))
(iff 'fluids 'turbulence 'yes (inc val -0.08611985))
(iff 'fluids 'shock-waves 'yes (inc val -0.09221114))
(iff 'user-interface 'text-editor 'yes (inc val -0.055205073))
(iff 'user-interface 'data-manipulator 'yes (inc val 0.008509341))
(iff 'user-interface 'control-language 'yes (inc val 0.15268995))
(iff 'user-interface 'user-specified-routines 'yes (inc val -0.08350555))
(iff 'user-interface 'matrix-spec-language 'yes (inc val 0.094205104))
(iff 'user-interface 'libraries 'yes (inc val 0.079101875))
(iff 'libraries 'number-of-stored-shapes 'very-high (inc val 0.19257195))
(iff 'libraries 'number-of-stored-shapes 'high (inc val 0.16786854))
(iff 'libraries 'number-of-stored-shapes 'medium (inc val 0.16407116))
(iff 'libraries 'number-of-stored-shapes 'low (inc val -0.10234634))
(iff 'libraries 'number-of-stored-shapes 'very-low (inc val 0.12230244))
(iff 'libraries 'format-conversion 'yes (inc val -0.07463047))
(iff 'libraries 'geometric-conversion 'yes (inc val 0.03752726))
(iff 'libraries 'translate-for-other-progs 'yes (inc val -0.10523188))
(iff 'graphics 'present 'yes (inc val 0.01654593))
(iff 'graphics '3d-structure 'yes (inc val 0.10852012))
```

```
 (iff 'graphics 'number-of-formats 'very-high (inc val 0.004384271))
 (iff 'graphics 'number-of-formats 'high (inc val 0.036400005))
 (iff 'graphics 'number-of-formats 'medium (inc val -0.1852))
 (iff 'graphics 'number-of-formats 'low (inc val -0.018812837))
 (iff 'graphics 'number-of-formats 'very-low (inc val 0.06609629))
 (iff 'graphics 'perspective 'yes (inc val -0.14577577))
 (iff 'graphics 'hidden-line-removal 'yes (inc val 0.05852331))
 (iff 'graphics 'shading 'yes (inc val -0.048456125))
 (iff 'commercialization 'many-projects 'yes (inc val 0.09687524))
 (iff 'commercialization 'many-companies 'yes (inc val -0.11879473))
 (iff 'commercialization 'industry-standard 'yes (inc val 0.0706483))
 (iff 'commercialization 'many-upgrades 'yes (inc val -0.03063732))
 (iff 'commercialization 'many-computers 'yes (inc val 0.041628294))
 (iff 'lumped-parameter 'number-known-components 'very-high (inc val -0.007429591))
 (iff 'lumped-parameter 'number-known-components 'high (inc val -0.18720001))
 (iff 'lumped-parameter 'number-known-components 'medium (inc val -0.15245347))
 (iff 'lumped-parameter 'number-known-components 'low (inc val -0.10261114))
 (iff 'lumped-parameter 'number-known-components 'very-low (inc val 0.10176187))
 (iff 'database-program 'video-images 'yes (inc val 0.07106094))
 (iff 'non-physical 'images 'yes (inc val -0.13939518))
 (iff 'systems-programming 'execute-commands 'yes (inc val 0.071497686))
 (iff 'systems-programming 'code-to-save-memory 'yes (inc val 0.14675224))
 (fput-list frame slot val))


;; The IF-NEEDED routine for frame METHODS, slot REDUNDANCY

(defun methods-redundancy (frame slot &aux val)
 (setq val -0.0572)
 (iff 'task 'engineering-program 'yes (inc val 0.21382213))
 (iff 'task 'database-program 'yes (inc val -0.07569032))
 (iff 'engineering-program 'structural-engineering 'yes (inc val -0.00720632))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val 0.0852))
 (iff 'engineering-program 'thermodynamics 'yes (inc val -0.050014555))
 (iff 'engineering-program 'control-systems 'yes (inc val -0.070801884))
 (iff 'engineering-program 'aerodynamics 'yes (inc val -0.17729628))
 (iff 'engineering-program 'image-processing 'yes (inc val 0.15028837))
 (iff 'thermodynamics 'radiation 'yes (inc val 0.061477546))
 (iff 'thermodynamics 'conduction 'yes (inc val -0.010745033))
 (iff 'methods 'number-redundant-choices 'very-high (inc val 0.016345214))
 (iff 'methods 'number-redundant-choices 'high (inc val -0.0119999945))
 (iff 'methods 'number-redundant-choices 'medium (inc val -0.07266162))
 (iff 'methods 'number-redundant-choices 'low (inc val -0.15090182))
 (iff 'methods 'number-redundant-choices 'very-low (inc val -0.03989061))
 (iff 'time-dependence 'static 'yes (inc val 0.013767125))
 (iff 'time-dependence 'quasi-static 'yes (inc val -0.12759677))
 (iff 'time-dependence 'fully-dynamic 'yes (inc val 0.053868454))
 (iff 'linearity 'linear 'yes (inc val 0.15399544))
 (iff 'linearity 'limited-non-linear 'yes (inc val 0.091097765))
 (iff 'linearity 'fully-non-linear 'yes (inc val -0.06161439))
 (iff 'boundary 'number-ways-to-specify 'very-high (inc val 0.08698144))
 (iff 'boundary 'number-ways-to-specify 'high (inc val 0.186))
 (iff 'boundary 'number-ways-to-specify 'medium (inc val 0.25323558))
 (iff 'boundary 'number-ways-to-specify 'low (inc val 0.09574672))
 (iff 'boundary 'number-ways-to-specify 'very-low (inc val -6.7262897e-4))
 (iff 'boundary '2d-field-on-3d-surface 'yes (inc val 0.08406843))
 (iff 'boundary 'lumped-component-values 'yes (inc val -0.017407551))
 (iff 'objects 'solids 'yes (inc val -0.16039771))
 (iff 'objects 'fluids 'yes (inc val -0.0015882566))
 (iff 'objects 'non-physical 'yes (inc val -0.061357427))
 (iff 'solids 'structure 'yes (inc val 0.109202266))
 (iff 'solids 'point-mass 'yes (inc val -0.07896923))
 (inc val (* (car (fget-z 'structure 'spatial-dimension)) -0.07248796))
 (iff 'structure 'lumped-parameter 'yes (inc val -0.07609992))
```

```
(iff 'structure 'continuous-volume 'yes (inc val 0.104793735))
(iff 'structure 'number-of-elements 'very-high (inc val -0.083680294))
(iff 'structure 'number-of-elements 'high (inc val 0.05167184))
(iff 'structure 'number-of-elements 'medium (inc val -0.06686951))
(iff 'structure 'number-of-elements 'low (inc val -0.17551972))
(iff 'structure 'number-of-elements 'very-low (inc val -0.111511156))
(iff 'structure 'substructuring-capability 'yes (inc val 0.1587955))
(iff 'continuous-volume 'finite-elements 'yes (inc val 0.020896912))
(iff 'continuous-volume 'grid 'yes (inc val 0.03685752))
(iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val 0.098456845))
(iff 'continuous-volume 'number-of-known-shapes 'high (inc val 0.1647793))
(iff 'continuous-volume 'number-of-known-shapes 'medium (inc val 0.0696853))
(iff 'continuous-volume 'number-of-known-shapes 'low (inc val 0.14387928))
(iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val 0.18569037))
(iff 'shape-limitations 'essentially-none 'yes (inc val 0.17815286))
(iff 'shape-limitations 'thin-shells 'yes (inc val -0.022768334))
(iff 'shape-limitations 'shells-of-revol 'yes (inc val 0.15688466))
(iff 'shape-limitations 'slabs 'yes (inc val -0.17923614))
(iff 'shape-limitations 'thick-shells 'yes (inc val -0.06471533))
(iff 'fluids 'turbulence 'yes (inc val 0.06310382))
(iff 'fluids 'shock-waves 'yes (inc val -0.12410146))
(iff 'user-interface 'text-editor 'yes (inc val -0.097921364))
(iff 'user-interface 'data-manipulator 'yes (inc val 0.028293092))
(iff 'user-interface 'control-language 'yes (inc val 0.13717386))
(iff 'user-interface 'user-specified-routines 'yes (inc val -0.05602517))
(iff 'user-interface 'matrix-spec-language 'yes (inc val 0.061290156))
(iff 'user-interface 'libraries 'yes (inc val 0.028134186))
(iff 'libraries 'number-of-stored-shapes 'very-high (inc val 0.15520096))
(iff 'libraries 'number-of-stored-shapes 'high (inc val -0.12291157))
(iff 'libraries 'number-of-stored-shapes 'medium (inc val -0.17003553))
(iff 'libraries 'number-of-stored-shapes 'low (inc val -0.24310055))
(iff 'libraries 'number-of-stored-shapes 'very-low (inc val 0.054137986))
(iff 'libraries 'format-conversion 'yes (inc val 0.1542234))
(iff 'libraries 'geometric-conversion 'yes (inc val 0.20134339))
(iff 'libraries 'translate-for-other-progs 'yes (inc val 0.1306556))
(iff 'graphics 'present 'yes (inc val 0.017665166))
(iff 'graphics '3d-structure 'yes (inc val -0.073260404))
(iff 'graphics 'number-of-formats 'very-high (inc val 0.019643009))
(iff 'graphics 'number-of-formats 'high (inc val 0.042400002))
(iff 'graphics 'number-of-formats 'medium (inc val 0.066))
(iff 'graphics 'number-of-formats 'low (inc val -0.20059201))
(iff 'graphics 'number-of-formats 'very-low (inc val -0.024159877))
(iff 'graphics 'perspective 'yes (inc val -0.003863127))
(iff 'graphics 'hidden-line-removal 'yes (inc val -0.122189485))
(iff 'graphics 'shading 'yes (inc val -0.12144625))
(iff 'commercialization 'many-projects 'yes (inc val -0.009052943))
(iff 'commercialization 'many-companies 'yes (inc val -0.060701884))
(iff 'commercialization 'industry-standard 'yes (inc val 0.03458374))
(iff 'commercialization 'many-upgrades 'yes (inc val 0.14307463))
(iff 'commercialization 'many-computers 'yes (inc val -0.11625518))
(iff 'lumped-parameter 'number-known-components 'very-high (inc val 0.27927178))
(iff 'lumped-parameter 'number-known-components 'high (inc val -0.17840001))
(iff 'lumped-parameter 'number-known-components 'medium (inc val 0.1024522))
(iff 'lumped-parameter 'number-known-components 'low (inc val 0.15789852))
(iff 'lumped-parameter 'number-known-components 'very-low (inc val 0.023268526))
(iff 'database-program 'video-images 'yes (inc val 0.043832436))
(iff 'non-physical 'images 'yes (inc val -0.023613838))
(iff 'systems-programming 'execute-commands 'yes (inc val -0.13648549))
(iff 'systems-programming 'code-to-save-memory 'yes (inc val 0.10503769))
(fput-list frame slot val))


;; The IF-NEEDED routine for frame OBJECTS, slot COMPLEXITY
```

```
(defun objects-complexity (frame slot &aux val)
 (setq val 0.04519999)
 (iff 'task 'engineering-program 'yes (inc val 0.21516515))
 (iff 'task 'database-program 'yes (inc val -0.046555318))
 (iff 'engineering-program 'structural-engineering 'yes (inc val 0.0366422))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val 0.0140000135))
 (iff 'engineering-program 'thermodynamics 'yes (inc val -0.094155446))
 (iff 'engineering-program 'control-systems 'yes (inc val -0.053079896))
 (iff 'engineering-program 'aerodynamics 'yes (inc val -0.08320488))
 (iff 'engineering-program 'image-processing 'yes (inc val -0.07650188))
 (iff 'thermodynamics 'radiation 'yes (inc val -0.08537224))
 (iff 'thermodynamics 'conduction 'yes (inc val 0.16986567))
 (iff 'methods 'number-redundant-choices 'very-high (inc val 0.12712917))
 (iff 'methods 'number-redundant-choices 'high (inc val 0.12000002))
 (iff 'methods 'number-redundant-choices 'medium (inc val -0.009048745))
 (iff 'methods 'number-redundant-choices 'low (inc val 0.11433482))
 (iff 'methods 'number-redundant-choices 'very-low (inc val 0.10942546))
 (iff 'time-dependence 'static 'yes (inc val 0.0911823))
 (iff 'time-dependence 'quasi-static 'yes (inc val -0.015631767))
 (iff 'time-dependence 'fully-dynamic 'yes (inc val -0.11153802))
 (iff 'linearity 'linear 'yes (inc val -0.028188974))
 (iff 'linearity 'limited-non-linear 'yes (inc val 0.06426294))
 (iff 'linearity 'fully-non-linear 'yes (inc val 0.10740984))
 (iff 'boundary 'number-ways-to-specify 'very-high (inc val 0.12915306))
 (iff 'boundary 'number-ways-to-specify 'high (inc val -0.1852))
 (iff 'boundary 'number-ways-to-specify 'medium (inc val -0.17007773))
 (iff 'boundary 'number-ways-to-specify 'low (inc val 0.07640339))
 (iff 'boundary 'number-ways-to-specify 'very-low (inc val 0.171162))
 (iff 'boundary '2d-field-on-3d-surface 'yes (inc val 0.22452389))
 (iff 'boundary 'lumped-component-values 'yes (inc val -0.024629401))
 (iff 'objects 'solids 'yes (inc val 0.0833406))
 (iff 'objects 'fluids 'yes (inc val 0.16721192))
 (iff 'objects 'non-physical 'yes (inc val 0.10112477))
 (iff 'solids 'structure 'yes (inc val 0.1885406))
 (iff 'solids 'point-mass 'yes (inc val -0.014989952))
 (inc val (* (car (fget-z 'structure 'spatial-dimension)) -0.016658407))
 (iff 'structure 'lumped-parameter 'yes (inc val -0.036622074))
 (iff 'structure 'continuous-volume 'yes (inc val 0.08184224))
 (iff 'structure 'number-of-elements 'very-high (inc val -0.070639506))
 (iff 'structure 'number-of-elements 'high (inc val -0.106502794))
 (iff 'structure 'number-of-elements 'medium (inc val 0.18517883))
 (iff 'structure 'number-of-elements 'low (inc val -0.11949584))
 (iff 'structure 'number-of-elements 'very-low (inc val -0.13029991))
 (iff 'structure 'substructuring-capability 'yes (inc val -0.042230725))
 (iff 'continuous-volume 'finite-elements 'yes (inc val -0.13273154))
 (iff 'continuous-volume 'grid 'yes (inc val -0.049780123))
 (iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val 0.0048310817))
 (iff 'continuous-volume 'number-of-known-shapes 'high (inc val 0.05968848))
 (iff 'continuous-volume 'number-of-known-shapes 'medium (inc val 0.021755436))
 (iff 'continuous-volume 'number-of-known-shapes 'low (inc val 0.06378857))
 (iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val -0.05382287))
 (iff 'shape-limitations 'essentially-none 'yes (inc val -0.049087122))
 (iff 'shape-limitations 'thin-shells 'yes (inc val -0.0641844))
 (iff 'shape-limitations 'shells-of-revol 'yes (inc val 0.16636667))
 (iff 'shape-limitations 'slabs 'yes (inc val -0.1062448))
 (iff 'shape-limitations 'thick-shells 'yes (inc val 0.020366618))
 (iff 'fluids 'turbulence 'yes (inc val -0.092004865))
 (iff 'fluids 'shock-waves 'yes (inc val 0.18090081))
 (iff 'user-interface 'text-editor 'yes (inc val 0.11889417))
 (iff 'user-interface 'data-manipulator 'yes (inc val -0.011614353))
 (iff 'user-interface 'control-language 'yes (inc val 0.16617233))
 (iff 'user-interface 'user-specified-routines 'yes (inc val 0.13074224))
 (iff 'user-interface 'matrix-spec-language 'yes (inc val 0.053052235))
 (iff 'user-interface 'libraries 'yes (inc val -0.021376679))
```

```
 (iff 'libraries 'number-of-stored-shapes 'very-high (inc val -0.05755708))
 (iff 'libraries 'number-of-stored-shapes 'high (inc val 0.025498124))
 (iff 'libraries 'number-of-stored-shapes 'medium (inc val 0.004338534))
 (iff 'libraries 'number-of-stored-shapes 'low (inc val 0.008106437))
 (iff 'libraries 'number-of-stored-shapes 'very-low (inc val 0.064254664))
 (iff 'libraries 'format-conversion 'yes (inc val 0.06638891))
 (iff 'libraries 'geometric-conversion 'yes (inc val 0.07673928))
 (iff 'libraries 'translate-for-other-progs 'yes (inc val 0.06334536))
 (iff 'graphics 'present 'yes (inc val 0.0034009404))
 (iff 'graphics '3d-structure 'yes (inc val -0.122660756))
 (iff 'graphics 'number-of-formats 'very-high (inc val -0.03143422))
 (iff 'graphics 'number-of-formats 'high (inc val -0.0024000108))
 (iff 'graphics 'number-of-formats 'medium (inc val 0.07520001))
 (iff 'graphics 'number-of-formats 'low (inc val -0.09255971))
 (iff 'graphics 'number-of-formats 'very-low (inc val -0.0061653866))
 (iff 'graphics 'perspective 'yes (inc val -0.046206743))
 (iff 'graphics 'hidden-line-removal 'yes (inc val -0.050544173))
 (iff 'graphics 'shading 'yes (inc val -0.022975259))
 (iff 'commercialization 'many-projects 'yes (inc val 0.022228852))
 (iff 'commercialization 'many-companies 'yes (inc val -0.112338))
 (iff 'commercialization 'industry-standard 'yes (inc val -0.091770954))
 (iff 'commercialization 'many-upgrades 'yes (inc val 0.015049095))
 (iff 'commercialization 'many-computers 'yes (inc val 0.0239514))
 (iff 'lumped-parameter 'number-known-components 'very-high (inc val -0.1985028))
 (iff 'lumped-parameter 'number-known-components 'high (inc val 0.044))
 (iff 'lumped-parameter 'number-known-components 'medium (inc val -0.115247905))
 (iff 'lumped-parameter 'number-known-components 'low (inc val 0.05930088))
 (iff 'lumped-parameter 'number-known-components 'very-low (inc val -0.008109517))
 (iff 'database-program 'video-images 'yes (inc val -0.004890043))
 (iff 'non-physical 'images 'yes (inc val 0.0077346903))
 (iff 'systems-programming 'execute-commands 'yes (inc val 0.045708653))
 (iff 'systems-programming 'code-to-save-memory 'yes (inc val 0.07980425))
 (fput-list frame slot val))


;; The IF-NEEDED routine for frame OBJECTS, slot GENERALITY

(defun objects-generality (frame slot &aux val)
 (setq val -0.055999994)
 (iff 'task 'engineering-program 'yes (inc val 0.10162884))
 (iff 'task 'database-program 'yes (inc val -0.03325494))
 (iff 'engineering-program 'structural-engineering 'yes (inc val -0.012774584))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val -0.072))
 (iff 'engineering-program 'thermodynamics 'yes (inc val 0.15103379))
 (iff 'engineering-program 'control-systems 'yes (inc val -0.10690186))
 (iff 'engineering-program 'aerodynamics 'yes (inc val -0.14443783))
 (iff 'engineering-program 'image-processing 'yes (inc val -0.12198767))
 (iff 'thermodynamics 'radiation 'yes (inc val 0.07243895))
 (iff 'thermodynamics 'conduction 'yes (inc val 0.12514801))
 (iff 'methods 'number-redundant-choices 'very-high (inc val 0.05409316))
 (iff 'methods 'number-redundant-choices 'high (inc val 0.037599996))
 (iff 'methods 'number-redundant-choices 'medium (inc val 0.11981277))
 (iff 'methods 'number-redundant-choices 'low (inc val 0.03278412))
 (iff 'methods 'number-redundant-choices 'very-low (inc val 0.027314818))
 (iff 'time-dependence 'static 'yes (inc val -0.03988077))
 (iff 'time-dependence 'quasi-static 'yes (inc val -0.028295638))
 (iff 'time-dependence 'fully-dynamic 'yes (inc val -0.07524317))
 (iff 'linearity 'linear 'yes (inc val -0.064337686))
 (iff 'linearity 'limited-non-linear 'yes (inc val 0.036316816))
 (iff 'linearity 'fully-non-linear 'yes (inc val 0.080736406))
 (iff 'boundary 'number-ways-to-specify 'very-high (inc val 0.14693101))
 (iff 'boundary 'number-ways-to-specify 'high (inc val -0.0036000013))
 (iff 'boundary 'number-ways-to-specify 'medium (inc val 0.03449942))
 (iff 'boundary 'number-ways-to-specify 'low (inc val -0.15724611))
```

```
(iff 'boundary 'number-ways-to-specify 'very-low (inc val -0.017379496))
(iff 'boundary '2d-field-on-3d-surface 'yes (inc val -0.021412456))
(iff 'boundary 'lumped-component-values 'yes (inc val 0.018298078))
(iff 'objects 'solids 'yes (inc val 0.12788956))
(iff 'objects 'fluids 'yes (inc val 0.11215701))
(iff 'objects 'non-physical 'yes (inc val 0.2327919))
(iff 'solids 'structure 'yes (inc val -0.06851049))
(iff 'solids 'point-mass 'yes (inc val -0.11665861))
(inc val (* (car (fget-z 'structure 'spatial-dimension)) 0.016403897))
(iff 'structure 'lumped-parameter 'yes (inc val 4.5195913e-4))
(iff 'structure 'continuous-volume 'yes (inc val 0.12762545))
(iff 'structure 'number-of-elements 'very-high (inc val -0.11794378))
(iff 'structure 'number-of-elements 'high (inc val 0.1155169))
(iff 'structure 'number-of-elements 'medium (inc val 0.13148569))
(iff 'structure 'number-of-elements 'low (inc val -0.14716928))
(iff 'structure 'number-of-elements 'very-low (inc val 0.07891526))
(iff 'structure 'substructuring-capability 'yes (inc val 0.0044966806))
(iff 'continuous-volume 'finite-elements 'yes (inc val 0.0011860424))
(iff 'continuous-volume 'grid 'yes (inc val -0.11896119))
(iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val 0.007280899))
(iff 'continuous-volume 'number-of-known-shapes 'high (inc val -0.15460299))
(iff 'continuous-volume 'number-of-known-shapes 'medium (inc val 0.16975631))
(iff 'continuous-volume 'number-of-known-shapes 'low (inc val 0.16311407))
(iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val 0.07205653))
(iff 'shape-limitations 'essentially-none 'yes (inc val 0.19642495))
(iff 'shape-limitations 'thin-shells 'yes (inc val 0.0553517))
(iff 'shape-limitations 'shells-of-revol 'yes (inc val 0.21519843))
(iff 'shape-limitations 'slabs 'yes (inc val -0.009287611))
(iff 'shape-limitations 'thick-shells 'yes (inc val 0.04159845))
(iff 'fluids 'turbulence 'yes (inc val 0.10636214))
(iff 'fluids 'shock-waves 'yes (inc val 0.14571068))
(iff 'user-interface 'text-editor 'yes (inc val -0.041743964))
(iff 'user-interface 'data-manipulator 'yes (inc val 0.23010847))
(iff 'user-interface 'control-language 'yes (inc val 0.2234146))
(iff 'user-interface 'user-specified-routines 'yes (inc val 5.495946e-4))
(iff 'user-interface 'matrix-spec-language 'yes (inc val 0.061563645))
(iff 'user-interface 'libraries 'yes (inc val -0.06681028))
(iff 'libraries 'number-of-stored-shapes 'very-high (inc val -0.019059034))
(iff 'libraries 'number-of-stored-shapes 'high (inc val -0.0019877357))
(iff 'libraries 'number-of-stored-shapes 'medium (inc val 0.16197108))
(iff 'libraries 'number-of-stored-shapes 'low (inc val -0.083079055))
(iff 'libraries 'number-of-stored-shapes 'very-low (inc val 0.1345595))
(iff 'libraries 'format-conversion 'yes (inc val 0.03666334))
(iff 'libraries 'geometric-conversion 'yes (inc val -0.16985352))
(iff 'libraries 'translate-for-other-progs 'yes (inc val -0.011306341))
(iff 'graphics 'present 'yes (inc val -0.10905321))
(iff 'graphics '3d-structure 'yes (inc val 0.0041846717))
(iff 'graphics 'number-of-formats 'very-high (inc val 0.20962992))
(iff 'graphics 'number-of-formats 'high (inc val -0.1196))
(iff 'graphics 'number-of-formats 'medium (inc val -0.0264))
(iff 'graphics 'number-of-formats 'low (inc val -0.086271614))
(iff 'graphics 'number-of-formats 'very-low (inc val 0.05664652))
(iff 'graphics 'perspective 'yes (inc val -0.06569221))
(iff 'graphics 'hidden-line-removal 'yes (inc val -0.08204249))
(iff 'graphics 'shading 'yes (inc val -0.06532336))
(iff 'commercialization 'many-projects 'yes (inc val 0.13954474))
(iff 'commercialization 'many-companies 'yes (inc val -0.07127242))
(iff 'commercialization 'industry-standard 'yes (inc val -0.03326635))
(iff 'commercialization 'many-upgrades 'yes (inc val -0.004313452))
(iff 'commercialization 'many-computers 'yes (inc val -0.036107674))
(iff 'lumped-parameter 'number-known-components 'very-high (inc val -0.20848309))
(iff 'lumped-parameter 'number-known-components 'high (inc val -0.0644))
(iff 'lumped-parameter 'number-known-components 'medium (inc val 0.14518543))
(iff 'lumped-parameter 'number-known-components 'low (inc val 0.106510624))
```

```
(iff 'lumped-parameter 'number-known-components 'very-low (inc val -0.098408155))
(iff 'database-program 'video-images 'yes (inc val -0.07552766))
(iff 'non-physical 'images 'yes (inc val 0.18034904))
(iff 'systems-programming 'execute-commands 'yes (inc val 0.16235644))
(iff 'systems-programming 'code-to-save-memory 'yes (inc val -0.060076125))
(fput-list frame slot val))


;; The IF-NEEDED routine for frame REPRESENTATION, slot COMPLEXITY

(defun representation-complexity (frame slot &aux val)
 (setq val 0.042799994)
 (iff 'task 'engineering-program 'yes (inc val 0.16686828))
 (iff 'task 'database-program 'yes (inc val 0.14919484))
 (iff 'engineering-program 'structural-engineering 'yes (inc val -0.047796763))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val 0.092800006))
 (iff 'engineering-program 'thermodynamics 'yes (inc val -0.09570364))
 (iff 'engineering-program 'control-systems 'yes (inc val 0.050326157))
 (iff 'engineering-program 'aerodynamics 'yes (inc val 0.16298206))
 (iff 'engineering-program 'image-processing 'yes (inc val 0.00604719))
 (iff 'thermodynamics 'radiation 'yes (inc val 0.17762595))
 (iff 'thermodynamics 'conduction 'yes (inc val -0.03927471))
 (iff 'methods 'number-redundant-choices 'very-high (inc val 0.17127965))
 (iff 'methods 'number-redundant-choices 'high (inc val -0.1556))
 (iff 'methods 'number-redundant-choices 'medium (inc val -0.11586908))
 (iff 'methods 'number-redundant-choices 'low (inc val 0.12558936))
 (iff 'methods 'number-redundant-choices 'very-low (inc val 0.11092863))
 (iff 'time-dependence 'static 'yes (inc val -0.07048193))
 (iff 'time-dependence 'quasi-static 'yes (inc val 0.013608872))
 (iff 'time-dependence 'fully-dynamic 'yes (inc val -0.1437356))
 (iff 'linearity 'linear 'yes (inc val 0.09606553))
 (iff 'linearity 'limited-non-linear 'yes (inc val -0.043462943))
 (iff 'linearity 'fully-non-linear 'yes (inc val -0.025767261))
 (iff 'boundary 'number-ways-to-specify 'very-high (inc val 0.18592554))
 (iff 'boundary 'number-ways-to-specify 'high (inc val -0.1588))
 (iff 'boundary 'number-ways-to-specify 'medium (inc val -0.098000005))
 (iff 'boundary 'number-ways-to-specify 'low (inc val -0.13446915))
 (iff 'boundary 'number-ways-to-specify 'very-low (inc val -0.01952804))
 (iff 'boundary '2d-field-on-3d-surface 'yes (inc val 0.15638651))
 (iff 'boundary 'lumped-component-values 'yes (inc val -0.15114854))
 (iff 'objects 'solids 'yes (inc val 0.082550004))
 (iff 'objects 'fluids 'yes (inc val -0.0011474579))
 (iff 'objects 'non-physical 'yes (inc val 0.031639628))
 (iff 'solids 'structure 'yes (inc val 0.17735003))
 (iff 'solids 'point-mass 'yes (inc val 0.18323308))
 (inc val (* (car (fget-z 'structure 'spatial-dimension)) 0.041050132))
 (iff 'structure 'lumped-parameter 'yes (inc val -0.032889683))
 (iff 'structure 'continuous-volume 'yes (inc val -0.06899676))
 (iff 'structure 'number-of-elements 'very-high (inc val 0.13538034))
 (iff 'structure 'number-of-elements 'high (inc val -0.023467286))
 (iff 'structure 'number-of-elements 'medium (inc val -0.17962898))
 (iff 'structure 'number-of-elements 'low (inc val 0.018265905))
 (iff 'structure 'number-of-elements 'very-low (inc val -0.14022157))
 (iff 'structure 'substructuring-capability 'yes (inc val 0.00904537))
 (iff 'continuous-volume 'finite-elements 'yes (inc val 0.0727515))
 (iff 'continuous-volume 'grid 'yes (inc val 0.09524327))
 (iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val 0.014032384))
 (iff 'continuous-volume 'number-of-known-shapes 'high (inc val 0.2025054))
 (iff 'continuous-volume 'number-of-known-shapes 'medium (inc val -0.040206045))
 (iff 'continuous-volume 'number-of-known-shapes 'low (inc val -0.13499816))
 (iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val -0.067405194))
 (iff 'shape-limitations 'essentially-none 'yes (inc val 0.050769247))
 (iff 'shape-limitations 'thin-shells 'yes (inc val -0.086162105))
 (iff 'shape-limitations 'shells-of-revol 'yes (inc val 0.17298305))
```

```
 (iff 'shape-limitations 'slabs 'yes (inc val -0.08721512))
 (iff 'shape-limitations 'thick-shells 'yes (inc val -0.11541689))
 (iff 'fluids 'turbulence 'yes (inc val 0.034582075))
 (iff 'fluids 'shock-waves 'yes (inc val -0.064106114))
 (iff 'user-interface 'text-editor 'yes (inc val -0.119057454))
 (iff 'user-interface 'data-manipulator 'yes (inc val 0.13250788))
 (iff 'user-interface 'control-language 'yes (inc val -0.06277002))
 (iff 'user-interface 'user-specified-routines 'yes (inc val 0.078964375))
 (iff 'user-interface 'matrix-spec-language 'yes (inc val -0.057779837))
 (iff 'user-interface 'libraries 'yes (inc val -0.016637133))
 (iff 'libraries 'number-of-stored-shapes 'very-high (inc val -0.11615891))
 (iff 'libraries 'number-of-stored-shapes 'high (inc val -0.20955285))
 (iff 'libraries 'number-of-stored-shapes 'medium (inc val -0.2394867))
 (iff 'libraries 'number-of-stored-shapes 'low (inc val -0.120701306))
 (iff 'libraries 'number-of-stored-shapes 'very-low (inc val 0.20462814))
 (iff 'libraries 'format-conversion 'yes (inc val -0.09084471))
 (iff 'libraries 'geometric-conversion 'yes (inc val 0.11643701))
 (iff 'libraries 'translate-for-other-progs 'yes (inc val 0.072339855))
 (iff 'graphics 'present 'yes (inc val 0.015883047))
 (iff 'graphics '3d-structure 'yes (inc val -0.04370362))
 (iff 'graphics 'number-of-formats 'very-high (inc val 0.15614952))
 (iff 'graphics 'number-of-formats 'high (inc val -0.0572))
 (iff 'graphics 'number-of-formats 'medium (inc val -0.0476))
 (iff 'graphics 'number-of-formats 'low (inc val 0.06013069))
 (iff 'graphics 'number-of-formats 'very-low (inc val 0.016848186))
 (iff 'graphics 'perspective 'yes (inc val 0.046853837))
 (iff 'graphics 'hidden-line-removal 'yes (inc val -0.0701064))
 (iff 'graphics 'shading 'yes (inc val -0.018938785))
 (iff 'commercialization 'many-projects 'yes (inc val 0.18571968))
 (iff 'commercialization 'many-companies 'yes (inc val 0.022658737))
 (iff 'commercialization 'industry-standard 'yes (inc val -0.026770685))
 (iff 'commercialization 'many-upgrades 'yes (inc val -0.092599876))
 (iff 'commercialization 'many-computers 'yes (inc val 0.008438659))
 (iff 'lumped-parameter 'number-known-components 'very-high (inc val -0.068267316))
 (iff 'lumped-parameter 'number-known-components 'high (inc val 0.09120001))
 (iff 'lumped-parameter 'number-known-components 'medium (inc val 0.08825782))
 (iff 'lumped-parameter 'number-known-components 'low (inc val -0.10890608))
 (iff 'lumped-parameter 'number-known-components 'very-low (inc val -0.11875599))
 (iff 'database-program 'video-images 'yes (inc val 0.08865587))
 (iff 'non-physical 'images 'yes (inc val -0.059482887))
 (iff 'systems-programming 'execute-commands 'yes (inc val 0.023548262))
 (iff 'systems-programming 'code-to-save-memory 'yes (inc val 0.03974405))
 (fput-list frame slot val))


;; The IF-NEEDED routine for frame REPRESENTATION, slot CAPACITY

(defun representation-capacity (frame slot &aux val)
 (setq val 0.12360002)
 (iff 'task 'engineering-program 'yes (inc val 0.20947029))
 (iff 'task 'database-program 'yes (inc val 0.12382602))
 (iff 'engineering-program 'structural-engineering 'yes (inc val 0.13676071))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val 0.11))
 (iff 'engineering-program 'thermodynamics 'yes (inc val 0.02681863))
 (iff 'engineering-program 'control-systems 'yes (inc val 0.04226289))
 (iff 'engineering-program 'aerodynamics 'yes (inc val -0.0024175558))
 (iff 'engineering-program 'image-processing 'yes (inc val -0.042103622))
 (iff 'thermodynamics 'radiation 'yes (inc val 0.0604303))
 (iff 'thermodynamics 'conduction 'yes (inc val -0.030268282))
 (iff 'methods 'number-redundant-choices 'very-high (inc val 0.15467131))
 (iff 'methods 'number-redundant-choices 'high (inc val 0.036799997))
 (iff 'methods 'number-redundant-choices 'medium (inc val -0.088236645))
 (iff 'methods 'number-redundant-choices 'low (inc val -0.04330272))
 (iff 'methods 'number-redundant-choices 'very-low (inc val 0.023356913))
```

```
(iff 'time-dependence 'static 'yes (inc val -1.289394e-4))
(iff 'time-dependence 'quasi-static 'yes (inc val -0.032063894))
(iff 'time-dependence 'fully-dynamic 'yes (inc val -0.09683384))
(iff 'linearity 'linear 'yes (inc val -0.22735152))
(iff 'linearity 'limited-non-linear 'yes (inc val -0.1724476))
(iff 'linearity 'fully-non-linear 'yes (inc val -0.07848824))
(iff 'boundary 'number-ways-to-specify 'very-high (inc val 0.046951666))
(iff 'boundary 'number-ways-to-specify 'high (inc val -0.062800005))
(iff 'boundary 'number-ways-to-specify 'medium (inc val 0.041503508))
(iff 'boundary 'number-ways-to-specify 'low (inc val -0.04803986))
(iff 'boundary 'number-ways-to-specify 'very-low (inc val 0.03687363))
(iff 'boundary '2d-field-on-3d-surface 'yes (inc val 0.12830366))
(iff 'boundary 'lumped-component-values 'yes (inc val -0.21488462))
(iff 'objects 'solids 'yes (inc val -0.013689235))
(iff 'objects 'fluids 'yes (inc val 0.017570779))
(iff 'objects 'non-physical 'yes (inc val -0.097927086))
(iff 'solids 'structure 'yes (inc val -0.086089246))
(iff 'solids 'point-mass 'yes (inc val 0.05069558))
(inc val (* (car (fget-z 'structure 'spatial-dimension)) 0.026740784))
(iff 'structure 'lumped-parameter 'yes (inc val -0.18308729))
(iff 'structure 'continuous-volume 'yes (inc val 0.05756072))
(iff 'structure 'number-of-elements 'very-high (inc val -0.014473833))
(iff 'structure 'number-of-elements 'high (inc val 0.17881878))
(iff 'structure 'number-of-elements 'medium (inc val 0.16068688))
(iff 'structure 'number-of-elements 'low (inc val 0.044878904))
(iff 'structure 'number-of-elements 'very-low (inc val 0.009378142))
(iff 'structure 'substructuring-capability 'yes (inc val 0.017610215))
(iff 'continuous-volume 'finite-elements 'yes (inc val 0.021912908))
(iff 'continuous-volume 'grid 'yes (inc val 0.12267725))
(iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val 0.04157497))
(iff 'continuous-volume 'number-of-known-shapes 'high (inc val -0.01280546))
(iff 'continuous-volume 'number-of-known-shapes 'medium (inc val 0.05299077))
(iff 'continuous-volume 'number-of-known-shapes 'low (inc val 0.02373507))
(iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val 0.082593545))
(iff 'shape-limitations 'essentially-none 'yes (inc val -0.055452295))
(iff 'shape-limitations 'thin-shells 'yes (inc val -0.034368105))
(iff 'shape-limitations 'shells-of-revol 'yes (inc val 0.124688536))
(iff 'shape-limitations 'slabs 'yes (inc val -0.13317645))
(iff 'shape-limitations 'thick-shells 'yes (inc val -0.07731149))
(iff 'fluids 'turbulence 'yes (inc val 0.11718244))
(iff 'fluids 'shock-waves 'yes (inc val -0.04190271))
(iff 'user-interface 'text-editor 'yes (inc val 0.06659781))
(iff 'user-interface 'data-manipulator 'yes (inc val -0.11011983))
(iff 'user-interface 'control-language 'yes (inc val 0.23126791))
(iff 'user-interface 'user-specified-routines 'yes (inc val -0.005489776))
(iff 'user-interface 'matrix-spec-language 'yes (inc val -0.10195692))
(iff 'user-interface 'libraries 'yes (inc val 0.12073451))
(iff 'libraries 'number-of-stored-shapes 'very-high (inc val 0.1444195))
(iff 'libraries 'number-of-stored-shapes 'high (inc val -0.093303606))
(iff 'libraries 'number-of-stored-shapes 'medium (inc val -0.15993859))
(iff 'libraries 'number-of-stored-shapes 'low (inc val -0.1357664))
(iff 'libraries 'number-of-stored-shapes 'very-low (inc val -0.06132197))
(iff 'libraries 'format-conversion 'yes (inc val 0.05646846))
(iff 'libraries 'geometric-conversion 'yes (inc val 0.069782384))
(iff 'libraries 'translate-for-other-progs 'yes (inc val -0.07577573))
(iff 'graphics 'present 'yes (inc val 0.09290389))
(iff 'graphics '3d-structure 'yes (inc val 0.08009502))
(iff 'graphics 'number-of-formats 'very-high (inc val -0.026189357))
(iff 'graphics 'number-of-formats 'high (inc val -0.0984))
(iff 'graphics 'number-of-formats 'medium (inc val -0.036799997))
(iff 'graphics 'number-of-formats 'low (inc val -0.07682752))
(iff 'graphics 'number-of-formats 'very-low (inc val 0.11230574))
(iff 'graphics 'perspective 'yes (inc val -0.008392328))
(iff 'graphics 'hidden-line-removal 'yes (inc val 0.13866976))
```

```
 (iff 'graphics 'shading 'yes (inc val -0.10210524))
 (iff 'commercialization 'many-projects 'yes (inc val -0.11635558))
 (iff 'commercialization 'many-companies 'yes (inc val 0.21442357))
 (iff 'commercialization 'industry-standard 'yes (inc val -0.14154269))
 (iff 'commercialization 'many-upgrades 'yes (inc val 0.13081498))
 (iff 'commercialization 'many-computers 'yes (inc val 0.12607463))
 (iff 'lumped-parameter 'number-known-components 'very-high (inc val 0.17601879))
 (iff 'lumped-parameter 'number-known-components 'high (inc val 0.07760002))
 (iff 'lumped-parameter 'number-known-components 'medium (inc val -0.059633717))
 (iff 'lumped-parameter 'number-known-components 'low (inc val 0.0028972898))
 (iff 'lumped-parameter 'number-known-components 'very-low (inc val -0.0015934804))
 (iff 'database-program 'video-images 'yes (inc val -0.13245916))
 (iff 'non-physical 'images 'yes (inc val 0.06743567))
 (iff 'systems-programming 'execute-commands 'yes (inc val -0.046228122))
 (iff 'systems-programming 'code-to-save-memory 'yes (inc val -0.12097398))
 (fput-list frame slot val))


;; The IF-NEEDED routine for frame REPRESENTATION, slot KNOWLEDGE

(defun representation-knowledge (frame slot &aux val)
 (setq val 0.13720001)
 (iff 'task 'engineering-program 'yes (inc val 0.2173163))
 (iff 'task 'database-program 'yes (inc val 0.08827213))
 (iff 'engineering-program 'structural-engineering 'yes (inc val -0.022969322))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val -0.0176))
 (iff 'engineering-program 'thermodynamics 'yes (inc val 0.23901685))
 (iff 'engineering-program 'control-systems 'yes (inc val -0.03760481))
 (iff 'engineering-program 'aerodynamics 'yes (inc val 0.033907894))
 (iff 'engineering-program 'image-processing 'yes (inc val -0.009357327))
 (iff 'thermodynamics 'radiation 'yes (inc val 0.16072051))
 (iff 'thermodynamics 'conduction 'yes (inc val -0.09498095))
 (iff 'methods 'number-redundant-choices 'very-high (inc val -0.047552332))
 (iff 'methods 'number-redundant-choices 'high (inc val 0.1328))
 (iff 'methods 'number-redundant-choices 'medium (inc val -0.058159836))
 (iff 'methods 'number-redundant-choices 'low (inc val -0.049738284))
 (iff 'methods 'number-redundant-choices 'very-low (inc val 0.1046461))
 (iff 'time-dependence 'static 'yes (inc val -0.061223034))
 (iff 'time-dependence 'quasi-static 'yes (inc val 0.06664058))
 (iff 'time-dependence 'fully-dynamic 'yes (inc val -0.075219825))
 (iff 'linearity 'linear 'yes (inc val 0.075415045))
 (iff 'linearity 'limited-non-linear 'yes (inc val -0.09451562))
 (iff 'linearity 'fully-non-linear 'yes (inc val 0.08628404))
 (iff 'boundary 'number-ways-to-specify 'very-high (inc val 0.039990295))
 (iff 'boundary 'number-ways-to-specify 'high (inc val -0.0536))
 (iff 'boundary 'number-ways-to-specify 'medium (inc val -0.03695123))
 (iff 'boundary 'number-ways-to-specify 'low (inc val -0.07507549))
 (iff 'boundary 'number-ways-to-specify 'very-low (inc val -0.089568034))
 (iff 'boundary '2d-field-on-3d-surface 'yes (inc val -0.07495986))
 (iff 'boundary 'lumped-component-values 'yes (inc val -0.07769095))
 (iff 'objects 'solids 'yes (inc val 0.18962619))
 (iff 'objects 'fluids 'yes (inc val -9.9577055e-4))
 (iff 'objects 'non-physical 'yes (inc val -0.0496323))
 (iff 'solids 'structure 'yes (inc val 0.014426237))
 (iff 'solids 'point-mass 'yes (inc val 0.11631391))
 (inc val (* (car (fget-z 'structure 'spatial-dimension)) -0.09096407))
 (iff 'structure 'lumped-parameter 'yes (inc val -0.12738419))
 (iff 'structure 'continuous-volume 'yes (inc val 0.24223068))
 (iff 'structure 'number-of-elements 'very-high (inc val -0.054458175))
 (iff 'structure 'number-of-elements 'high (inc val 0.023751868))
 (iff 'structure 'number-of-elements 'medium (inc val -0.1160022))
 (iff 'structure 'number-of-elements 'low (inc val -0.048865277))
 (iff 'structure 'number-of-elements 'very-low (inc val -0.106830664))
 (iff 'structure 'substructuring-capability 'yes (inc val 0.027690813))
```

```
 (iff 'continuous-volume 'finite-elements 'yes (inc val 0.037396647))
 (iff 'continuous-volume 'grid 'yes (inc val -0.051293727))
 (iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val 0.190605))
 (iff 'continuous-volume 'number-of-known-shapes 'high (inc val 0.116339475))
 (iff 'continuous-volume 'number-of-known-shapes 'medium (inc val -0.059427075))
 (iff 'continuous-volume 'number-of-known-shapes 'low (inc val 0.032987103))
 (iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val -0.082109004))
 (iff 'shape-limitations 'essentially-none 'yes (inc val 0.032173906))
 (iff 'shape-limitations 'thin-shells 'yes (inc val -0.012405593))
 (iff 'shape-limitations 'shells-of-revol 'yes (inc val -0.1135283))
 (iff 'shape-limitations 'slabs 'yes (inc val -0.10614117))
 (iff 'shape-limitations 'thick-shells 'yes (inc val -0.032328304))
 (iff 'fluids 'turbulence 'yes (inc val 0.105107896))
 (iff 'fluids 'shock-waves 'yes (inc val 0.073894195))
 (iff 'user-interface 'text-editor 'yes (inc val 0.12155227))
 (iff 'user-interface 'data-manipulator 'yes (inc val -0.094337575))
 (iff 'user-interface 'control-language 'yes (inc val -0.0342653))
 (iff 'user-interface 'user-specified-routines 'yes (inc val 0.023027934))
 (iff 'user-interface 'matrix-spec-language 'yes (inc val -0.09265282))
 (iff 'user-interface 'libraries 'yes (inc val 0.12920085))
 (iff 'libraries 'number-of-stored-shapes 'very-high (inc val 0.19993202))
 (iff 'libraries 'number-of-stored-shapes 'high (inc val -0.011357345))
 (iff 'libraries 'number-of-stored-shapes 'medium (inc val -0.066490225))
 (iff 'libraries 'number-of-stored-shapes 'low (inc val 0.06637471))
 (iff 'libraries 'number-of-stored-shapes 'very-low (inc val 0.01953635))
 (iff 'libraries 'format-conversion 'yes (inc val 0.15343758))
 (iff 'libraries 'geometric-conversion 'yes (inc val 0.1258015))
 (iff 'libraries 'translate-for-other-progs 'yes (inc val -0.042462017))
 (iff 'graphics 'present 'yes (inc val 0.21358623))
 (iff 'graphics '3d-structure 'yes (inc val -0.17962344))
 (iff 'graphics 'number-of-formats 'very-high (inc val -0.06342692))
 (iff 'graphics 'number-of-formats 'high (inc val -0.1104))
 (iff 'graphics 'number-of-formats 'medium (inc val -0.1632))
 (iff 'graphics 'number-of-formats 'low (inc val -0.0840753))
 (iff 'graphics 'number-of-formats 'very-low (inc val 0.017497722))
 (iff 'graphics 'perspective 'yes (inc val -0.16753446))
 (iff 'graphics 'hidden-line-removal 'yes (inc val 0.09820339))
 (iff 'graphics 'shading 'yes (inc val 0.10399837))
 (iff 'commercialization 'many-projects 'yes (inc val 0.06266849))
 (iff 'commercialization 'many-companies 'yes (inc val -0.06453058))
 (iff 'commercialization 'industry-standard 'yes (inc val 0.11933995))
 (iff 'commercialization 'many-upgrades 'yes (inc val 0.06292531))
 (iff 'commercialization 'many-computers 'yes (inc val 0.09787561))
 (iff 'lumped-parameter 'number-known-components 'very-high (inc val -0.03384813))
 (iff 'lumped-parameter 'number-known-components 'high (inc val -0.19240001))
 (iff 'lumped-parameter 'number-known-components 'medium (inc val 0.13544938))
 (iff 'lumped-parameter 'number-known-components 'low (inc val -0.077705786))
 (iff 'lumped-parameter 'number-known-components 'very-low (inc val -0.07069985))
 (iff 'database-program 'video-images 'yes (inc val 0.036369752))
 (iff 'non-physical 'images 'yes (inc val -0.026831888))
 (iff 'systems-programming 'execute-commands 'yes (inc val 0.030754123))
 (iff 'systems-programming 'code-to-save-memory 'yes (inc val 0.06478346))
 (fput-list frame slot val))


;; The IF-NEEDED routine for frame GRAPHICS, slot COMPLEXITY

(defun graphics-complexity (frame slot &aux val)
 (setq val 0.0448)
 (iff 'task 'engineering-program 'yes (inc val 0.13745873))
 (iff 'task 'database-program 'yes (inc val 0.051921915))
 (iff 'engineering-program 'structural-engineering 'yes (inc val -0.22406907))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val 0.1348))
 (iff 'engineering-program 'thermodynamics 'yes (inc val 0.09458021))
```

```
(iff 'engineering-program 'control-systems 'yes (inc val -0.1044173))
(iff 'engineering-program 'aerodynamics 'yes (inc val 0.063870996))
(iff 'engineering-program 'image-processing 'yes (inc val -0.057164595))
(iff 'thermodynamics 'radiation 'yes (inc val -0.18686852))
(iff 'thermodynamics 'conduction 'yes (inc val 0.008416869))
(iff 'methods 'number-redundant-choices 'very-high (inc val 0.081855066))
(iff 'methods 'number-redundant-choices 'high (inc val 0.1912))
(iff 'methods 'number-redundant-choices 'medium (inc val 0.20390984))
(iff 'methods 'number-redundant-choices 'low (inc val -0.05315444))
(iff 'methods 'number-redundant-choices 'very-low (inc val 0.09036513))
(iff 'time-dependence 'static 'yes (inc val -0.1633547))
(iff 'time-dependence 'quasi-static 'yes (inc val 0.1619585))
(iff 'time-dependence 'fully-dynamic 'yes (inc val -0.097325645))
(iff 'linearity 'linear 'yes (inc val 0.022856645))
(iff 'linearity 'limited-non-linear 'yes (inc val -0.07202876))
(iff 'linearity 'fully-non-linear 'yes (inc val 0.16916552))
(iff 'boundary 'number-ways-to-specify 'very-high (inc val -0.0010720321))
(iff 'boundary 'number-ways-to-specify 'high (inc val 0.16960002))
(iff 'boundary 'number-ways-to-specify 'medium (inc val 0.0882549))
(iff 'boundary 'number-ways-to-specify 'low (inc val -0.096906446))
(iff 'boundary 'number-ways-to-specify 'very-low (inc val -0.03810076))
(iff 'boundary '2d-field-on-3d-surface 'yes (inc val 0.0546883))
(iff 'boundary 'lumped-component-values 'yes (inc val -0.011718076))
(iff 'objects 'solids 'yes (inc val 0.15294707))
(iff 'objects 'fluids 'yes (inc val -0.026280334))
(iff 'objects 'non-physical 'yes (inc val 0.02776715))
(iff 'solids 'structure 'yes (inc val -0.07185299))
(iff 'solids 'point-mass 'yes (inc val 0.12253052))
(inc val (* (car (fget-z 'structure 'spatial-dimension)) -0.05187925))
(iff 'structure 'lumped-parameter 'yes (inc val 0.013790152))
(iff 'structure 'continuous-volume 'yes (inc val 0.110730976))
(iff 'structure 'number-of-elements 'very-high (inc val -0.15947407))
(iff 'structure 'number-of-elements 'high (inc val -0.15057854))
(iff 'structure 'number-of-elements 'medium (inc val -0.0358367))
(iff 'structure 'number-of-elements 'low (inc val -0.063163616))
(iff 'structure 'number-of-elements 'very-low (inc val 0.031628624))
(iff 'structure 'substructuring-capability 'yes (inc val -0.15507583))
(iff 'continuous-volume 'finite-elements 'yes (inc val 0.24870501))
(iff 'continuous-volume 'grid 'yes (inc val 0.09080952))
(iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val 0.18421967))
(iff 'continuous-volume 'number-of-known-shapes 'high (inc val -0.16499245))
(iff 'continuous-volume 'number-of-known-shapes 'medium (inc val 0.020492848))
(iff 'continuous-volume 'number-of-known-shapes 'low (inc val 0.1716559))
(iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val 0.019599661))
(iff 'shape-limitations 'essentially-none 'yes (inc val 0.25169984))
(iff 'shape-limitations 'thin-shells 'yes (inc val -0.035528794))
(iff 'shape-limitations 'shells-of-revol 'yes (inc val 0.14990298))
(iff 'shape-limitations 'slabs 'yes (inc val -0.13684109))
(iff 'shape-limitations 'thick-shells 'yes (inc val -0.16929705))
(iff 'fluids 'turbulence 'yes (inc val -0.24692906))
(iff 'fluids 'shock-waves 'yes (inc val -0.12298087))
(iff 'user-interface 'text-editor 'yes (inc val -0.030469086))
(iff 'user-interface 'data-manipulator 'yes (inc val 0.045080982))
(iff 'user-interface 'control-language 'yes (inc val 0.17504546))
(iff 'user-interface 'user-specified-routines 'yes (inc val 0.07372439))
(iff 'user-interface 'matrix-spec-language 'yes (inc val -0.008679777))
(iff 'user-interface 'libraries 'yes (inc val 4.099533e-4))
(iff 'libraries 'number-of-stored-shapes 'very-high (inc val 0.08043983))
(iff 'libraries 'number-of-stored-shapes 'high (inc val 0.12003534))
(iff 'libraries 'number-of-stored-shapes 'medium (inc val -0.10707315))
(iff 'libraries 'number-of-stored-shapes 'low (inc val 0.10175983))
(iff 'libraries 'number-of-stored-shapes 'very-low (inc val -0.095386274))
(iff 'libraries 'format-conversion 'yes (inc val 0.06275554))
(iff 'libraries 'geometric-conversion 'yes (inc val 0.019990982))
```

```
 (iff 'libraries 'translate-for-other-progs 'yes (inc val -0.1061834))
 (iff 'graphics 'present 'yes (inc val 0.016617678))
 (iff 'graphics '3d-structure 'yes (inc val 0.09830821))
 (iff 'graphics 'number-of-formats 'very-high (inc val 0.15021369))
 (iff 'graphics 'number-of-formats 'high (inc val -0.091199994))
 (iff 'graphics 'number-of-formats 'medium (inc val -0.13679999))
 (iff 'graphics 'number-of-formats 'low (inc val 0.11122601))
 (iff 'graphics 'number-of-formats 'very-low (inc val -0.11846418))
 (iff 'graphics 'perspective 'yes (inc val 0.211934))
 (iff 'graphics 'hidden-line-removal 'yes (inc val -0.09404178))
 (iff 'graphics 'shading 'yes (inc val 0.20693853))
 (iff 'commercialization 'many-projects 'yes (inc val 0.0019554754))
 (iff 'commercialization 'many-companies 'yes (inc val -0.11757948))
 (iff 'commercialization 'industry-standard 'yes (inc val 0.10731269))
 (iff 'commercialization 'many-upgrades 'yes (inc val -0.06290726))
 (iff 'commercialization 'many-computers 'yes (inc val -0.07365038))
 (iff 'lumped-parameter 'number-known-components 'very-high (inc val -0.16697855))
 (iff 'lumped-parameter 'number-known-components 'high (inc val 0.14240001))
 (iff 'lumped-parameter 'number-known-components 'medium (inc val 0.17701817))
 (iff 'lumped-parameter 'number-known-components 'low (inc val 0.08381913))
 (iff 'lumped-parameter 'number-known-components 'very-low (inc val 0.07431693))
 (iff 'database-program 'video-images 'yes (inc val -0.025744481))
 (iff 'non-physical 'images 'yes (inc val -0.05840595))
 (iff 'systems-programming 'execute-commands 'yes (inc val -0.13415068))
 (iff 'systems-programming 'code-to-save-memory 'yes (inc val 0.1943074))
 (fput-list frame slot val))


;; The IF-NEEDED routine for frame USER-INTERFACE, slot COMPLEXITY

(defun user-interface-complexity (frame slot &aux val)
 (setq val 0.07000001)
 (iff 'task 'engineering-program 'yes (inc val -0.13245223))
 (iff 'task 'database-program 'yes (inc val 0.18114536))
 (iff 'engineering-program 'structural-engineering 'yes (inc val -0.024334712))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val 0.1756))
 (iff 'engineering-program 'thermodynamics 'yes (inc val 0.19553381))
 (iff 'engineering-program 'control-systems 'yes (inc val -0.014402249))
 (iff 'engineering-program 'aerodynamics 'yes (inc val -0.11173894))
 (iff 'engineering-program 'image-processing 'yes (inc val 0.16490908))
 (iff 'thermodynamics 'radiation 'yes (inc val -3.0301845e-4))
 (iff 'thermodynamics 'conduction 'yes (inc val -0.18589935))
 (iff 'methods 'number-redundant-choices 'very-high (inc val -0.08028955))
 (iff 'methods 'number-redundant-choices 'high (inc val 0.088000014))
 (iff 'methods 'number-redundant-choices 'medium (inc val -0.0021219493))
 (iff 'methods 'number-redundant-choices 'low (inc val 0.0093895625))
 (iff 'methods 'number-redundant-choices 'very-low (inc val -0.053331327))
 (iff 'time-dependence 'static 'yes (inc val 0.13923872))
 (iff 'time-dependence 'quasi-static 'yes (inc val -0.17513557))
 (iff 'time-dependence 'fully-dynamic 'yes (inc val 0.061767865))
 (iff 'linearity 'linear 'yes (inc val 0.07737879))
 (iff 'linearity 'limited-non-linear 'yes (inc val 0.10258101))
 (iff 'linearity 'fully-non-linear 'yes (inc val 0.11315591))
 (iff 'boundary 'number-ways-to-specify 'very-high (inc val -0.023431506))
 (iff 'boundary 'number-ways-to-specify 'high (inc val -0.17))
 (iff 'boundary 'number-ways-to-specify 'medium (inc val -0.15108417))
 (iff 'boundary 'number-ways-to-specify 'low (inc val -0.11919303))
 (iff 'boundary 'number-ways-to-specify 'very-low (inc val -0.031444557))
 (iff 'boundary '2d-field-on-3d-surface 'yes (inc val 0.11455127))
 (iff 'boundary 'lumped-component-values 'yes (inc val 0.057881147))
 (iff 'objects 'solids 'yes (inc val 0.04454463))
 (iff 'objects 'fluids 'yes (inc val 0.16009775))
 (iff 'objects 'non-physical 'yes (inc val -0.0059253518))
 (iff 'solids 'structure 'yes (inc val -0.056255385))
```

```
(iff 'solids 'point-mass 'yes (inc val 0.0013907182))
(inc val (* (car (fget-z 'structure 'spatial-dimension)) 0.04774377))
(iff 'structure 'lumped-parameter 'yes (inc val 0.056670286))
(iff 'structure 'continuous-volume 'yes (inc val -0.08913474))
(iff 'structure 'number-of-elements 'very-high (inc val -0.14310122))
(iff 'structure 'number-of-elements 'high (inc val 0.0771822))
(iff 'structure 'number-of-elements 'medium (inc val -0.1437669))
(iff 'structure 'number-of-elements 'low (inc val -0.08736947))
(iff 'structure 'number-of-elements 'very-low (inc val 0.060702093))
(iff 'structure 'substructuring-capability 'yes (inc val -0.074873686))
(iff 'continuous-volume 'finite-elements 'yes (inc val 0.019569788))
(iff 'continuous-volume 'grid 'yes (inc val 0.090833575))
(iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val -0.038398672))
(iff 'continuous-volume 'number-of-known-shapes 'high (inc val -0.026146784))
(iff 'continuous-volume 'number-of-known-shapes 'medium (inc val 0.1256328))
(iff 'continuous-volume 'number-of-known-shapes 'low (inc val -0.06205655))
(iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val -0.08778411))
(iff 'shape-limitations 'essentially-none 'yes (inc val 0.03268828))
(iff 'shape-limitations 'thin-shells 'yes (inc val 0.09786448))
(iff 'shape-limitations 'shells-of-revol 'yes (inc val -0.07999943))
(iff 'shape-limitations 'slabs 'yes (inc val -0.082455985))
(iff 'shape-limitations 'thick-shells 'yes (inc val 0.17000063))
(iff 'fluids 'turbulence 'yes (inc val 0.011861053))
(iff 'fluids 'shock-waves 'yes (inc val 0.13395649))
(iff 'user-interface 'text-editor 'yes (inc val -0.1704554))
(iff 'user-interface 'data-manipulator 'yes (inc val 0.06823593))
(iff 'user-interface 'control-language 'yes (inc val 0.07052754))
(iff 'user-interface 'user-specified-routines 'yes (inc val 0.11484991))
(iff 'user-interface 'matrix-spec-language 'yes (inc val 0.079736754))
(iff 'user-interface 'libraries 'yes (inc val 0.04402837))
(iff 'libraries 'number-of-stored-shapes 'very-high (inc val 0.09014916))
(iff 'libraries 'number-of-stored-shapes 'high (inc val -0.05949088))
(iff 'libraries 'number-of-stored-shapes 'medium (inc val -0.023245724))
(iff 'libraries 'number-of-stored-shapes 'low (inc val -0.004369741))
(iff 'libraries 'number-of-stored-shapes 'very-low (inc val 0.26100376))
(iff 'libraries 'format-conversion 'yes (inc val 0.11492483))
(iff 'libraries 'geometric-conversion 'yes (inc val 0.1100038))
(iff 'libraries 'translate-for-other-progs 'yes (inc val -0.14592952))
(iff 'graphics 'present 'yes (inc val 0.10625532))
(iff 'graphics '3d-structure 'yes (inc val -0.022898072))
(iff 'graphics 'number-of-formats 'very-high (inc val -0.05556915))
(iff 'graphics 'number-of-formats 'high (inc val -0.076400004))
(iff 'graphics 'number-of-formats 'medium (inc val -0.1648))
(iff 'graphics 'number-of-formats 'low (inc val -0.07928447))
(iff 'graphics 'number-of-formats 'very-low (inc val -0.16029972))
(iff 'graphics 'perspective 'yes (inc val -0.0045090243))
(iff 'graphics 'hidden-line-removal 'yes (inc val -0.045826126))
(iff 'graphics 'shading 'yes (inc val 0.06657254))
(iff 'commercialization 'many-projects 'yes (inc val 0.06309887))
(iff 'commercialization 'many-companies 'yes (inc val 0.1254123))
(iff 'commercialization 'industry-standard 'yes (inc val -0.042816233))
(iff 'commercialization 'many-upgrades 'yes (inc val -0.031493403))
(iff 'commercialization 'many-computers 'yes (inc val -0.059063252))
(iff 'lumped-parameter 'number-known-components 'very-high (inc val 0.06518222))
(iff 'lumped-parameter 'number-known-components 'high (inc val 0.13080002))
(iff 'lumped-parameter 'number-known-components 'medium (inc val -0.024565415))
(iff 'lumped-parameter 'number-known-components 'low (inc val 0.007556507))
(iff 'lumped-parameter 'number-known-components 'very-low (inc val -0.05692658))
(iff 'database-program 'video-images 'yes (inc val 0.011456958))
(iff 'non-physical 'images 'yes (inc val 0.08002952))
(iff 'systems-programming 'execute-commands 'yes (inc val -0.021698138))
(iff 'systems-programming 'code-to-save-memory 'yes (inc val 0.12416149))
(fput-list frame slot val))
```

```
;; The IF-NEEDED routine for frame DATA-MANAGEMENT, slot COMPLEXITY

(defun data-management-complexity (frame slot &aux val)
 (setq val 0.030800015)
 (iff 'task 'engineering-program 'yes (inc val -0.1328848))
 (iff 'task 'database-program 'yes (inc val 0.11190756))
 (iff 'engineering-program 'structural-engineering 'yes (inc val -0.16719842))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val 0.0059999973))
 (iff 'engineering-program 'thermodynamics 'yes (inc val -0.08180122))
 (iff 'engineering-program 'control-systems 'yes (inc val -0.107553236))
 (iff 'engineering-program 'aerodynamics 'yes (inc val -0.06495628))
 (iff 'engineering-program 'image-processing 'yes (inc val -0.020205645))
 (iff 'thermodynamics 'radiation 'yes (inc val -0.08227366))
 (iff 'thermodynamics 'conduction 'yes (inc val 0.11679537))
 (iff 'methods 'number-redundant-choices 'very-high (inc val 0.14359973))
 (iff 'methods 'number-redundant-choices 'high (inc val -0.1596))
 (iff 'methods 'number-redundant-choices 'medium (inc val -0.049590323))
 (iff 'methods 'number-redundant-choices 'low (inc val -0.08642122))
 (iff 'methods 'number-redundant-choices 'very-low (inc val 0.070774995))
 (iff 'time-dependence 'static 'yes (inc val -0.13862693))
 (iff 'time-dependence 'quasi-static 'yes (inc val -0.044492967))
 (iff 'time-dependence 'fully-dynamic 'yes (inc val -0.06571752))
 (iff 'linearity 'linear 'yes (inc val 0.020146439))
 (iff 'linearity 'limited-non-linear 'yes (inc val 0.15141861))
 (iff 'linearity 'fully-non-linear 'yes (inc val -0.037130363))
 (iff 'boundary 'number-ways-to-specify 'very-high (inc val 0.16823906))
 (iff 'boundary 'number-ways-to-specify 'high (inc val -0.0964))
 (iff 'boundary 'number-ways-to-specify 'medium (inc val 0.050149847))
 (iff 'boundary 'number-ways-to-specify 'low (inc val -5.0564437e-4))
 (iff 'boundary 'number-ways-to-specify 'very-low (inc val -0.15072002))
 (iff 'boundary '2d-field-on-3d-surface 'yes (inc val 0.0657354))
 (iff 'boundary 'lumped-component-values 'yes (inc val -0.05918107))
 (iff 'objects 'solids 'yes (inc val 0.08242493))
 (iff 'objects 'fluids 'yes (inc val 0.09111613))
 (iff 'objects 'non-physical 'yes (inc val 0.1472881))
 (iff 'solids 'structure 'yes (inc val -0.18597515))
 (iff 'solids 'point-mass 'yes (inc val -0.07900209))
 (inc val (* (car (fget-z 'structure 'spatial-dimension)) -0.07240349))
 (iff 'structure 'lumped-parameter 'yes (inc val -0.13700584))
 (iff 'structure 'continuous-volume 'yes (inc val 0.16920157))
 (iff 'structure 'number-of-elements 'very-high (inc val -0.091537826))
 (iff 'structure 'number-of-elements 'high (inc val 0.060811214))
 (iff 'structure 'number-of-elements 'medium (inc val -0.09339666))
 (iff 'structure 'number-of-elements 'low (inc val 0.08614813))
 (iff 'structure 'number-of-elements 'very-low (inc val 0.026738307))
 (iff 'structure 'substructuring-capability 'yes (inc val 0.18323928))
 (iff 'continuous-volume 'finite-elements 'yes (inc val -0.11598683))
 (iff 'continuous-volume 'grid 'yes (inc val 0.2587935))
 (iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val 0.0038053894))
 (iff 'continuous-volume 'number-of-known-shapes 'high (inc val 0.19425584))
 (iff 'continuous-volume 'number-of-known-shapes 'medium (inc val 0.015351142))
 (iff 'continuous-volume 'number-of-known-shapes 'low (inc val 0.16324447))
 (iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val 0.040506262))
 (iff 'shape-limitations 'essentially-none 'yes (inc val -0.016459461))
 (iff 'shape-limitations 'thin-shells 'yes (inc val -0.15331402))
 (iff 'shape-limitations 'shells-of-revol 'yes (inc val -0.015236954))
 (iff 'shape-limitations 'slabs 'yes (inc val -0.041592427))
 (iff 'shape-limitations 'thick-shells 'yes (inc val -0.012036955))
 (iff 'fluids 'turbulence 'yes (inc val -0.026956264))
 (iff 'fluids 'shock-waves 'yes (inc val 0.14162615))
 (iff 'user-interface 'text-editor 'yes (inc val -0.07651837))
 (iff 'user-interface 'data-manipulator 'yes (inc val 0.17744781))
 (iff 'user-interface 'control-language 'yes (inc val 0.028189456))
```

```
 (iff 'user-interface 'user-specified-routines 'yes (inc val 0.06585189))
 (iff 'user-interface 'matrix-spec-language 'yes (inc val -0.024245817))
 (iff 'user-interface 'libraries 'yes (inc val -0.09547308))
 (iff 'libraries 'number-of-stored-shapes 'very-high (inc val 0.10910679))
 (iff 'libraries 'number-of-stored-shapes 'high (inc val -0.011005638))
 (iff 'libraries 'number-of-stored-shapes 'medium (inc val 0.089046106))
 (iff 'libraries 'number-of-stored-shapes 'low (inc val 0.17651431))
 (iff 'libraries 'number-of-stored-shapes 'very-low (inc val -0.18289831))
 (iff 'libraries 'format-conversion 'yes (inc val 0.1384533))
 (iff 'libraries 'geometric-conversion 'yes (inc val 0.19121297))
 (iff 'libraries 'translate-for-other-progs 'yes (inc val -0.04430574))
 (iff 'graphics 'present 'yes (inc val 0.14807615))
 (iff 'graphics '3d-structure 'yes (inc val 0.14292651))
 (iff 'graphics 'number-of-formats 'very-high (inc val 0.20825087))
 (iff 'graphics 'number-of-formats 'high (inc val -0.111600004))
 (iff 'graphics 'number-of-formats 'medium (inc val -0.1052))
 (iff 'graphics 'number-of-formats 'low (inc val -0.026589954))
 (iff 'graphics 'number-of-formats 'very-low (inc val 0.15950225))
 (iff 'graphics 'perspective 'yes (inc val -0.11946601))
 (iff 'graphics 'hidden-line-removal 'yes (inc val -0.16044487))
 (iff 'graphics 'shading 'yes (inc val 0.005749755))
 (iff 'commercialization 'many-projects 'yes (inc val 0.08386179))
 (iff 'commercialization 'many-companies 'yes (inc val -0.12378141))
 (iff 'commercialization 'industry-standard 'yes (inc val -0.060346447))
 (iff 'commercialization 'many-upgrades 'yes (inc val 0.1781645))
 (iff 'commercialization 'many-computers 'yes (inc val 0.051693242))
 (iff 'lumped-parameter 'number-known-components 'very-high (inc val 0.15481125))
 (iff 'lumped-parameter 'number-known-components 'high (inc val -0.063999996))
 (iff 'lumped-parameter 'number-known-components 'medium (inc val 0.08323041))
 (iff 'lumped-parameter 'number-known-components 'low (inc val 0.16842614))
 (iff 'lumped-parameter 'number-known-components 'very-low (inc val 0.06749542))
 (iff 'database-program 'video-images 'yes (inc val -0.029304251))
 (iff 'non-physical 'images 'yes (inc val -0.05195449))
 (iff 'systems-programming 'execute-commands 'yes (inc val -0.052845396))
 (iff 'systems-programming 'code-to-save-memory 'yes (inc val -0.011086686))
 (fput-list frame slot val))


;; The IF-NEEDED routine for frame SYS-PROGRAMMING, slot COMPLEXITY

(defun sys-programming-complexity (frame slot &aux val)
 (setq val -0.0908)
 (iff 'task 'engineering-program 'yes (inc val 0.16359632))
 (iff 'task 'database-program 'yes (inc val 0.08547161))
 (iff 'engineering-program 'structural-engineering 'yes (inc val 0.040477786))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val -0.12))
 (iff 'engineering-program 'thermodynamics 'yes (inc val -0.013192825))
 (iff 'engineering-program 'control-systems 'yes (inc val -0.0892433))
 (iff 'engineering-program 'aerodynamics 'yes (inc val -0.13405645))
 (iff 'engineering-program 'image-processing 'yes (inc val 0.10926937))
 (iff 'thermodynamics 'radiation 'yes (inc val 0.03620786))
 (iff 'thermodynamics 'conduction 'yes (inc val 0.21187292))
 (iff 'methods 'number-redundant-choices 'very-high (inc val -0.06884393))
 (iff 'methods 'number-redundant-choices 'high (inc val 0.09319998))
 (iff 'methods 'number-redundant-choices 'medium (inc val 0.07857715))
 (iff 'methods 'number-redundant-choices 'low (inc val -0.049253486))
 (iff 'methods 'number-redundant-choices 'very-low (inc val 0.14125526))
 (iff 'time-dependence 'static 'yes (inc val -0.04726411))
 (iff 'time-dependence 'quasi-static 'yes (inc val 0.025362883))
 (iff 'time-dependence 'fully-dynamic 'yes (inc val -0.12685674))
 (iff 'linearity 'linear 'yes (inc val -0.11672532))
 (iff 'linearity 'limited-non-linear 'yes (inc val -0.032994755))
 (iff 'linearity 'fully-non-linear 'yes (inc val -0.028932847))
 (iff 'boundary 'number-ways-to-specify 'very-high (inc val -0.17232803))
```

```
(iff 'boundary 'number-ways-to-specify 'high (inc val 0.1276))
(iff 'boundary 'number-ways-to-specify 'medium (inc val -0.03278401))
(iff 'boundary 'number-ways-to-specify 'low (inc val 0.063084155))
(iff 'boundary 'number-ways-to-specify 'very-low (inc val 0.19856298))
(iff 'boundary '2d-field-on-3d-surface 'yes (inc val 0.16505584))
(iff 'boundary 'lumped-component-values 'yes (inc val 0.015859857))
(iff 'objects 'solids 'yes (inc val -0.12382034))
(iff 'objects 'fluids 'yes (inc val 0.1565429))
(iff 'objects 'non-physical 'yes (inc val -0.05678254))
(iff 'solids 'structure 'yes (inc val 0.10937967))
(iff 'solids 'point-mass 'yes (inc val 0.18489724))
(inc val (* (car (fget-z 'structure 'spatial-dimension)) -0.07613311))
(iff 'structure 'lumped-parameter 'yes (inc val 0.23471975))
(iff 'structure 'continuous-volume 'yes (inc val -0.1879222))
(iff 'structure 'number-of-elements 'very-high (inc val -0.14820513))
(iff 'structure 'number-of-elements 'high (inc val -0.14660043))
(iff 'structure 'number-of-elements 'medium (inc val 0.09133425))
(iff 'structure 'number-of-elements 'low (inc val -0.1495489))
(iff 'structure 'number-of-elements 'very-low (inc val -0.010844522))
(iff 'structure 'substructuring-capability 'yes (inc val 0.113107875))
(iff 'continuous-volume 'finite-elements 'yes (inc val 0.14231206))
(iff 'continuous-volume 'grid 'yes (inc val -0.08039446))
(iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val -0.034513324))
(iff 'continuous-volume 'number-of-known-shapes 'high (inc val -0.18228686))
(iff 'continuous-volume 'number-of-known-shapes 'medium (inc val 0.14938244))
(iff 'continuous-volume 'number-of-known-shapes 'low (inc val 0.048873372))
(iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val -0.06732052))
(iff 'shape-limitations 'essentially-none 'yes (inc val 0.0028997844))
(iff 'shape-limitations 'thin-shells 'yes (inc val -0.003462799))
(iff 'shape-limitations 'shells-of-revol 'yes (inc val -0.005136408))
(iff 'shape-limitations 'slabs 'yes (inc val 0.11933698))
(iff 'shape-limitations 'thick-shells 'yes (inc val 0.059263613))
(iff 'fluids 'turbulence 'yes (inc val -0.014056447))
(iff 'fluids 'shock-waves 'yes (inc val -0.0776042))
(iff 'user-interface 'text-editor 'yes (inc val 0.042663723))
(iff 'user-interface 'data-manipulator 'yes (inc val -0.08011407))
(iff 'user-interface 'control-language 'yes (inc val -0.09336077))
(iff 'user-interface 'user-specified-routines 'yes (inc val 0.05792645))
(iff 'user-interface 'matrix-spec-language 'yes (inc val 0.05903819))
(iff 'user-interface 'libraries 'yes (inc val -0.0045040944))
(iff 'libraries 'number-of-stored-shapes 'very-high (inc val -0.07055829))
(iff 'libraries 'number-of-stored-shapes 'high (inc val -0.0823306))
(iff 'libraries 'number-of-stored-shapes 'medium (inc val -0.016149998))
(iff 'libraries 'number-of-stored-shapes 'low (inc val 0.09883577))
(iff 'libraries 'number-of-stored-shapes 'very-low (inc val -0.01526177))
(iff 'libraries 'format-conversion 'yes (inc val -0.08938202))
(iff 'libraries 'geometric-conversion 'yes (inc val 0.057058636))
(iff 'libraries 'translate-for-other-progs 'yes (inc val -0.10713856))
(iff 'graphics 'present 'yes (inc val -0.02903828))
(iff 'graphics '3d-structure 'yes (inc val 0.20216523))
(iff 'graphics 'number-of-formats 'very-high (inc val 0.13957314))
(iff 'graphics 'number-of-formats 'high (inc val -0.028400004))
(iff 'graphics 'number-of-formats 'medium (inc val -0.1176))
(iff 'graphics 'number-of-formats 'low (inc val 0.009577551))
(iff 'graphics 'number-of-formats 'very-low (inc val -0.10181556))
(iff 'graphics 'perspective 'yes (inc val -0.12706459))
(iff 'graphics 'hidden-line-removal 'yes (inc val -0.16325134))
(iff 'graphics 'shading 'yes (inc val 0.08966197))
(iff 'commercialization 'many-projects 'yes (inc val -0.115019925))
(iff 'commercialization 'many-companies 'yes (inc val 0.039512943))
(iff 'commercialization 'industry-standard 'yes (inc val -0.05559422))
(iff 'commercialization 'many-upgrades 'yes (inc val 0.088850394))
(iff 'commercialization 'many-computers 'yes (inc val 0.1045798))
(iff 'lumped-parameter 'number-known-components 'very-high (inc val -0.14420044))
```

```
 (iff 'lumped-parameter 'number-known-components 'high (inc val 0.1544))
 (iff 'lumped-parameter 'number-known-components 'medium (inc val 0.20111653))
 (iff 'lumped-parameter 'number-known-components 'low (inc val 0.12159582))
 (iff 'lumped-parameter 'number-known-components 'very-low (inc val 0.016023159))
 (iff 'database-program 'video-images 'yes (inc val -0.03277557))
 (iff 'non-physical 'images 'yes (inc val 0.1448865))
 (iff 'systems-programming 'execute-commands 'yes (inc val -0.07633767))
 (iff 'systems-programming 'code-to-save-memory 'yes (inc val 0.15317217))
 (fput-list frame slot val))


;; The IF-NEEDED routine for frame HARD-ARCHITECTURE, slot COMPLEXITY

(defun hard-architecture-complexity (frame slot &aux val)
 (setq val -0.154)
 (iff 'task 'engineering-program 'yes (inc val -0.0013482039))
 (iff 'task 'database-program 'yes (inc val 0.117435575))
 (iff 'engineering-program 'structural-engineering 'yes (inc val -0.119745724))
 (iff 'engineering-program 'electronic-circuit 'yes (inc val -0.13640001))
 (iff 'engineering-program 'thermodynamics 'yes (inc val 0.05105252))
 (iff 'engineering-program 'control-systems 'yes (inc val 0.119598225))
 (iff 'engineering-program 'aerodynamics 'yes (inc val 0.036167707))
 (iff 'engineering-program 'image-processing 'yes (inc val -0.09138998))
 (iff 'thermodynamics 'radiation 'yes (inc val 0.0225285))
 (iff 'thermodynamics 'conduction 'yes (inc val -0.08044807))
 (iff 'methods 'number-redundant-choices 'very-high (inc val -0.0014083743))
 (iff 'methods 'number-redundant-choices 'high (inc val -0.075600006))
 (iff 'methods 'number-redundant-choices 'medium (inc val 0.09931266))
 (iff 'methods 'number-redundant-choices 'low (inc val 0.12080977))
 (iff 'methods 'number-redundant-choices 'very-low (inc val -0.002474317))
 (iff 'time-dependence 'static 'yes (inc val 0.16035777))
 (iff 'time-dependence 'quasi-static 'yes (inc val -0.052254252))
 (iff 'time-dependence 'fully-dynamic 'yes (inc val -0.12496841))
 (iff 'linearity 'linear 'yes (inc val 0.12732668))
 (iff 'linearity 'limited-non-linear 'yes (inc val 0.08542358))
 (iff 'linearity 'fully-non-linear 'yes (inc val 0.12750362))
 (iff 'boundary 'number-ways-to-specify 'very-high (inc val -0.08902612))
 (iff 'boundary 'number-ways-to-specify 'high (inc val 0.006800011))
 (iff 'boundary 'number-ways-to-specify 'medium (inc val -0.15747282))
 (iff 'boundary 'number-ways-to-specify 'low (inc val -0.04211257))
 (iff 'boundary 'number-ways-to-specify 'very-low (inc val -0.030348841))
 (iff 'boundary '2d-field-on-3d-surface 'yes (inc val -0.013165859))
 (iff 'boundary 'lumped-component-values 'yes (inc val -0.056185313))
 (iff 'objects 'solids 'yes (inc val -0.013026827))
 (iff 'objects 'fluids 'yes (inc val -0.02490823))
 (iff 'objects 'non-physical 'yes (inc val -0.14028639))
 (iff 'solids 'structure 'yes (inc val 0.10497319))
 (iff 'solids 'point-mass 'yes (inc val -0.1179648))
 (inc val (* (car (fget-z 'structure 'spatial-dimension)) 0.060789485))
 (iff 'structure 'lumped-parameter 'yes (inc val -0.13399489))
 (iff 'structure 'continuous-volume 'yes (inc val 0.06785428))
 (iff 'structure 'number-of-elements 'very-high (inc val -0.024088793))
 (iff 'structure 'number-of-elements 'high (inc val 0.12677409))
 (iff 'structure 'number-of-elements 'medium (inc val -0.046499394))
 (iff 'structure 'number-of-elements 'low (inc val 0.09398723))
 (iff 'structure 'number-of-elements 'very-low (inc val 0.17246653))
 (iff 'structure 'substructuring-capability 'yes (inc val -0.18430778))
 (iff 'continuous-volume 'finite-elements 'yes (inc val 0.08664629))
 (iff 'continuous-volume 'grid 'yes (inc val 0.09461483))
 (iff 'continuous-volume 'number-of-known-shapes 'very-high (inc val 0.039981585))
 (iff 'continuous-volume 'number-of-known-shapes 'high (inc val 0.098168425))
 (iff 'continuous-volume 'number-of-known-shapes 'medium (inc val -0.123943664))
 (iff 'continuous-volume 'number-of-known-shapes 'low (inc val -0.05899225))
 (iff 'continuous-volume 'number-of-known-shapes 'very-low (inc val 0.13542567))
```

```
(iff 'shape-limitations 'essentially-none 'yes (inc val 0.10458109))
(iff 'shape-limitations 'thin-shells 'yes (inc val -0.13634907))
(iff 'shape-limitations 'shells-of-revol 'yes (inc val -0.0025769756))
(iff 'shape-limitations 'slabs 'yes (inc val 0.009630773))
(iff 'shape-limitations 'thick-shells 'yes (inc val 0.099823035))
(iff 'fluids 'turbulence 'yes (inc val 0.06416767))
(iff 'fluids 'shock-waves 'yes (inc val 0.16104586))
(iff 'user-interface 'text-editor 'yes (inc val 0.12005911))
(iff 'user-interface 'data-manipulator 'yes (inc val -0.035557877))
(iff 'user-interface 'control-language 'yes (inc val -0.009600595))
(iff 'user-interface 'user-specified-routines 'yes (inc val 0.0052227573))
(iff 'user-interface 'matrix-spec-language 'yes (inc val -0.07349289))
(iff 'user-interface 'libraries 'yes (inc val -0.03242931))
(iff 'libraries 'number-of-stored-shapes 'very-high (inc val -0.06433589))
(iff 'libraries 'number-of-stored-shapes 'high (inc val -0.1649899))
(iff 'libraries 'number-of-stored-shapes 'medium (inc val -0.05695246))
(iff 'libraries 'number-of-stored-shapes 'low (inc val 0.10026162))
(iff 'libraries 'number-of-stored-shapes 'very-low (inc val -0.0485436))
(iff 'libraries 'format-conversion 'yes (inc val -0.032308694))
(iff 'libraries 'geometric-conversion 'yes (inc val 0.085061125))
(iff 'libraries 'translate-for-other-progs 'yes (inc val -0.06991953))
(iff 'graphics 'present 'yes (inc val -0.03515198))
(iff 'graphics '3d-structure 'yes (inc val -0.15362635))
(iff 'graphics 'number-of-formats 'very-high (inc val 0.114321284))
(iff 'graphics 'number-of-formats 'high (inc val 0.048000008))
(iff 'graphics 'number-of-formats 'medium (inc val 0.154))
(iff 'graphics 'number-of-formats 'low (inc val 0.021870507))
(iff 'graphics 'number-of-formats 'very-low (inc val 0.020447895))
(iff 'graphics 'perspective 'yes (inc val -0.030375035))
(iff 'graphics 'hidden-line-removal 'yes (inc val 0.20122872))
(iff 'graphics 'shading 'yes (inc val -0.13915284))
(iff 'commercialization 'many-projects 'yes (inc val -0.035442844))
(iff 'commercialization 'many-companies 'yes (inc val 0.07679934))
(iff 'commercialization 'industry-standard 'yes (inc val -0.016351426))
(iff 'commercialization 'many-upgrades 'yes (inc val -0.019292513))
(iff 'commercialization 'many-computers 'yes (inc val -0.053422067))
(iff 'lumped-parameter 'number-known-components 'very-high (inc val -0.1116259))
(iff 'lumped-parameter 'number-known-components 'high (inc val -0.012800008))
(iff 'lumped-parameter 'number-known-components 'medium (inc val -0.13154343))
(iff 'lumped-parameter 'number-known-components 'low (inc val 0.067045875))
(iff 'lumped-parameter 'number-known-components 'very-low (inc val -0.040836886))
(iff 'database-program 'video-images 'yes (inc val -0.007307438))
(iff 'non-physical 'images 'yes (inc val 0.11873978))
(iff 'systems-programming 'execute-commands 'yes (inc val 0.0017242436))
(iff 'systems-programming 'code-to-save-memory 'yes (inc val 0.024214413))
(fput-list frame slot val))
```

File hd:nasa:production system:percep2.lisp

```lisp
;;; -*- Mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-


; 14 Aug 89

(defvar alpha  0.9 "The momentum coefficient")
(defvar rscale 0.4  "The scale of the random weights")
(defvar eta    0.25 "Learning rate (.5 for no hidden layers)")

(defvar layers ()   "A list of the nodes in each layer starting at input")
(defvar nlayer 0    "The number of layers, including input and output")
(defvar ngap   0    "The number of gaps between layers")


; The nonlinearity of all nodes. The threshold is subtracted BEFORE
; calling SIGMOID.

(defun sigmoid (x)
  (cond ((> x 20) 1.)
        ((< x -20) 0.)
        (t (/ 1. (+ 1. (exp (- x)))))))


(defun learn (inputs outputs ncycle &optional (pnum (1+ ncycle)))
  (if (> nlayer 2)
     (learn-hid inputs outputs ncycle pnum)
     (learn-lin inputs outputs ncycle pnum)))

(defun learn-lin (inputs outputs ncycle &optional (pnum (1+ ncycle)))
  (dotimes (cyc ncycle)
   (do ((in inputs (cdr in)) (out outputs (cdr out))) ((null in))
    (forward-lin (car in))
    (back-lin (car out)))
   (update-net 0)
   (if (zerop (mod (1+ cyc) pnum)) (print-state cyc inputs outputs))))

(defun back-lin (des)
  (dolist (x (first layers))
   (mapcar #'(lambda (w d n)
               (setf (get w 'del)
                 (+ (get w 'del) (* eta (- d (get n 'excite))
                                      (get x 'excite)))))
            (get x 'to-weights) des (second layers))))

(defun learn-hid (inputs outputs ncycle &optional (pnum (1+ ncycle)))
  (dotimes (c ncycle)
   (do* ((in inputs) (out outputs)
         (i (pop in) (pop in)) (o (pop out) (pop out)))
        ((null i))
    (forward-hid i)
    (back (top-grads (nth ngap layers) o) ngap)
   (update-net 0)
    )
   (if (zerop (mod (1+ c) pnum)) (print-state c inputs outputs))))

(defun top-grads (nodes desired)
  (mapcar #'(lambda (n d &aux (e (get n 'excite))) (* e (- 1. e) (- d e)))
          nodes desired))

(defun backward (desired)
  (back (top-grads (nth ngap layers) desired) ngap))
```

```
(defun back (grads r)
 (update-row grads r)
 (if (> r 1) (back (get-grads (1- r) grads) (1- r))))

(defun update-row (grads r)
 (mapcar #'(lambda (n g) (update-wt n g 1.)) (nth r layers) grads)
 (mapcar #'(lambda (n) (update-node grads n)) (nth (1- r) layers)))

(defun update-node (grads n &aux (exe (get n 'excite)))
 (mapcar #'(lambda (w g) (update-wt w g exe)) (get n 'to-weights) grads))

(defun update-wt (w g e)
 (setf (get w 'del) (+ (get w 'del) (* eta g e) (* alpha (get w 'old)))))

(defun get-grads (r pgrads)
 (mapcar #'(lambda (n) (get-g (get n 'excite) pgrads (to-vals n)))
            (nth r layers)))

(defun get-g (e gr wt)
 (* e (- 1 e) (dot gr wt)))

(defun dot (x y)
 (apply '+ (mapcar '* x y)))

(defun forward (inputs)
 (if (> nlayer 2)
    (forward-hid inputs)
    (forward-lin inputs)))

(defun forward-lin (inputs)
 (mapcar #'(lambda (n e) (setf (get n 'excite) e)) (first layers) inputs)
 (mapcar #'(lambda (y) (setf (get y 'excite)
                                        (+ (dot inputs (from-vals y)) (get y 'val))))
            (second layers)))

(defun forward-hid (inputs)
 (mapcar #'(lambda (n i) (setf (get n 'excite) i)) (car layers) inputs)
 (forw 1 inputs))

(defun forw (r pexe &aux exe)
 (setq exe (mapcar #'(lambda (n) (set-exe n pexe (from-vals n)))
                         (nth r layers)))
 (if (< r ngap) (forw (1+ r) exe) exe))

(defun set-exe (n exe wts)
 (setf (get n 'excite) (sigmoid (+ (dot exe wts) (get n 'val)))))

(defun from-vals (node)
 (mapcar #'(lambda (w) (get w 'val)) (get node 'from-weights)))

(defun to-vals (node)
 (mapcar #'(lambda (w) (get w 'val)) (get node 'to-weights)))

(defun update-net (row &aux (nodes (nth row layers)))
 (mapcar #'(lambda (n) (update-n n row)) nodes)
 (if (<= row nlayer) (update-net (1+ row))))

(defun update-n (n row)
 (if (> row 0) (upd n))
 (if (< row nlayer) (mapcar 'upd (get n 'to-weights))))

(defun upd (x &aux (del (get x 'del)))
 (setf (get x 'val) (+ (get x 'val) del))
 (setf (get x 'del) 0.)
```

```
 (setf (get x 'old) del))

; The initialization routines

(defun set-up-net (layer-sizes &aux)
 (setq layers (make-names layer-sizes)) ; save this global info for display
 (setq nlayer (length layers))         ; number of layers in the network
 (setq ngap  (1- nlayer))              ; the number of gaps in the network
 (make-weights layers)                 ; the coupling between nodes
 )

(defun make-weights (net &aux wts)
 (do ((row (pop net) (pop net)))
    ((null net))
   (dolist (node (car net))
    (setq wts
           (mapcar #'(lambda (n)
                         (make-weight (get n 'number)
                                      (get node 'number))) row))
    (mapcar #'(lambda (n wt) (place-weight n node wt)) row wts))))

(defun make-names (layer-sizes &aux (n -1) l names)
 (dolist (layer layer-sizes)
  (setq l nil)
  (dotimes (node layer)
    (push-end (make-sym (inc n 1) names) l))
  (push-end l names))
 names)

(defun make-sym (n node-names &aux node)
 (setq node (intern (format nil "N~d" n)))
 (setf (get node 'number) n)          ; a count of the nodes
 (setf (get node 'to-weights) nil)
 (setf (get node 'from-weights) nil)
 (cond ((consp node-names)             ; it's not the input layer
          (init-element node))
         (t
          (setf (get node 'val) nil)      ; it is the input layer
          (setf (get node 'del) nil)
          (setf (get node 'old) nil)))
 node)

(defun make-weight (n1 n2 &aux weight)
 (cond ((> n1 n2) (make-weight n2 n1))
         (t
          (setq weight (intern (format nil "W_~D-~D" n1 n2)))
          (init-element weight))))

(defun init-element (ele)
 (setf (get ele 'old) 0.)
 (setf (get ele 'val) (make-rand))
 (setf (get ele 'del) 0.)
 ele)

(defun place-weight (node1 node2 wt)
 (cond ((> (get node1 'number) (get node2 'number))
          (place-weight node2 node1 wt))
         (t
          (setf (get node2 'from-weights)
              (append (get node2 'from-weights) (list wt)))
          (setf (get node1 'to-weights)
              (append (get node1 'to-weights) (list wt)))
          )))
```

```
(defun make-rand ()
 (- (* rscale (/ (random 1000) 1000))
    (/ rscale 2.))) ; a real random value


; Display functions

(defun display-weights (&optional (type 'val))
 (do* ((rows (reverse layers)) (row (pop rows) (pop rows))
          (l ngap (1- l))) ((null rows))
   (format t "~%                          to layer ~d~%~%" l)
   (do* ((ro (car rows)) (node (pop ro) (pop ro)))
           ((null node))
     (format t "from Layer ~d node ~s  " (1- l) node)
     (dolist (w (get node 'to-weights))
           (format t "~12,7F  " (get w type)))
     (format t "~%"))
   (format t "~%")))

(defun display-net (prop)
 (format t "~%          Network Property ~s~%~%" prop)
 (do* ((rows (reverse layers)) (row (pop rows) (pop rows))
          (r ngap (1- r)))
     ((null row))
   (format t "row ~d   " r)
   (dolist (n row)
     (format t "~12,7F  " (get n prop)))
   (format t "~%")))



(defun print-state (cycle inputs outputs &aux (err 0) dif)
 (do ((i (pop inputs) (pop inputs)) (o (pop outputs) (pop outputs)))
     ((null i))
   (setq dif (mapcar '- o (forward i)))
   (setq err (+ err (dot dif dif)))
   )
 (format t "~%Cycle~5D  ~s" cycle err))

(defun print-stuff (inputs outputs)
 (format t "~%")
 (do ((i (pop inputs) (pop inputs)) (o (pop outputs) (pop outputs)))
     ((null i))
   (format t "          ~S --> ~S ...... ~S~%"
             i o (forward i))))

;  Set up the input for the symmetry test

(defun make-inputs (&aux res)
 (dotimes (i 2 res)
   (dotimes (j 2)
     (dotimes (k 2)
           (dotimes (l 2)
             (dotimes (m 2)
               (dotimes (n 2)
                 (push-end (list i j k l m n) res))))))))

(defun stest (l &aux rl)
 (dotimes (i (/ (length l) 2))
   (push (pop l) rl))
 (if (equal rl l) 1 0))


; The SAVESTATE function stores the network parameters is a list structure,
```

```
; as follows:
;       state-list :== {entities}
;         entity :== name, {attributes}
;           attribute :== attribute-name, value
;             attribute-name :== NIL | property
; Where NIL means value binding and PROPERTY is used on the property list.

(defun savestate (&aux state)
 (setq state (list (list 'eta  (list nil eta))
                   (list 'alpha (list nil alpha))
                   (list 'layers (list nil layers))
                   (list 'nlayer (list nil nlayer))
                   (list 'ngap (list nil ngap))))
 (insert-nodes state)
 (insert-weights state))

(defun insert-nodes (state &aux nrec)
 (dolist (l layers state)
  (dolist (n l)
   (setq nrec (list n))
   (push-end (list 'val (get n 'val)) nrec)
   (push-end (list 'del (get n 'del)) nrec)
   (push-end (list 'old (get n 'old)) nrec)
   (push-end (list 'to-weights (get n 'to-weights)) nrec)
   (push-end (list 'from-weights (get n 'from-weights)) nrec)
   (push-end nrec state))))

(defun insert-weights (state &aux wrec)
 (dolist (l layers state)
  (dolist (n l)
   (dolist (w (get n 'to-weights))
        (setq wrec (list w))
        (push-end (list 'val (get w 'val)) wrec)
        (push-end (list 'del (get w 'del)) wrec)
        (push-end (list 'old (get w 'old)) wrec)
        (push-end wrec state)))))

(Defun Restorestate (state)
 (mapcar 'restoreentity state))

(defun restoreentity (entity &aux (ent (car entity)) (avlist (cdr entity)))
 (mapcar #'(lambda (av-pair)
            (restoreav ent (first av-pair) (second av-pair))) avlist))

(defun restoreav (ent attr val)
 (if attr
   (setf (get ent attr) val)
   (set ent val)))
```

File hd:nasa:production system:screen.lisp

```
;;; -*- Mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-

;; Define a flavor for each screen.

(defflavor screen ((FrameList nil)) (ApplicationEdit)
  :gettable-instance-variables :settable-instance-variables
  :initable-instance-variables)

(defflavor results () (ApplicationEdit))

(defflavor spec-screen () (ApplicationEdit))

(defflavor help () (ApplicationEdit))

(defflavor task () (spec-screen))

(defflavor objects () (spec-screen))

(defflavor structure () (spec-screen))

(defflavor methods () (spec-screen))

(defflavor User-Interface () (spec-screen))

(defflavor graphics () (spec-screen))

(defflavor other () (spec-screen))

(defmethod (help :Initialize) (&optional frame slot &rest ignore)
  (cond ((eq frame 'screen)
          (send (send self :NamedObject :Frame) :Set-Value slot)
          (send (send self :NamedObject :Slot)  :Set-Value 'screen))
         (frame
          (send (send self :NamedObject :Frame) :Set-Value frame)
          (send (send self :NamedObject :Slot)  :Set-Value slot)))
  (send (send self :NamedObject :Text)  :Set-Value
         (apply 'concatenate (cons 'string (fget frame slot 'help)))))

(defmethod (results :Initialize) (&rest ignore)
  (send (send self :NamedObject :Code-Size) :Set-Value
         (* (car (fget 'program 'predicted-size 'most-likely))
            (send (send self :NamedObject :Languages) :Value)))
  (send (send self :NamedObject :Fluctuation) :Set-Value
         (car (fget 'program 'predicted-size 'fluctuation)))
  (send self :generic-weights)
  )

(defmethod (results :rescale) (obj &rest ignore)
  (send (send self :NamedObject :Code-Size) :Set-Value
         (round (* (car (fget 'program 'predicted-size 'most-likely))
                   (send obj :value)))))


(defmethod (results :generic-weights) (&rest ignore &aux frames weights val total)
  (setq frames (remove-duplicates (mapcar 'car (generic-features))))
  (dolist (f frames)
   (setq val 0)
   (dolist (s (fget f 'generic-slots 'value))
    (setq val (+ val (max 0 (car (fget f s 'value)))))
   )
   (push-end val weights))
  (setq total (apply '+ weights))
```

```
 (set-frame '(debug))
 (dolist (f frames)
   (fput 'debug 'object 'value (send self :NamedObject (intern f 'keyword)))
   (setq val (read-from-string (format nil "~6,2f" (/ (pop weights) total))))
   (send (send self :NamedObject (intern f 'keyword)) :Set-Value val)))

(defmethod (screen :EditScreen) (Obj &rest ignore)
  (send self :GoSub (send Obj :Selection)))

(defmethod (spec-screen :NamedObject) (objectname &aux object)
  (dolist (ob displaylist object)
   (if (eq objectname (send ob :Name)) (setq object ob))))

(defmethod (screen :specify) (&rest ignore)
  (send self :GoSub 'task)
  (send self :GoSub 'methods)
  (send self :GoSub 'objects)
  (send self :GoSub 'structure)
  (send self :GoSub 'user-interface)
  (send self :GoSub 'graphics)
  (send self :GoSub 'other)
  )

(defmethod (screen :store) (&rest ignore &aux file)
  (setq file (send (send self :NamedObject :file) :Value))
  (with-open-file (out file :direction :output)
   (pprint (mapcar 'show FrameList) out)))

(defmethod (screen :load) (&rest ignore &aux file)
  (setq file (send (send self :NamedObject :file) :Value))
  (with-open-file (inp file)
   (setq FrameList (set-frame-list (read inp)))))


;; Determine Code Size

(defmethod (screen :size) (&rest ignore &aux size sizes fluct mean)
  (fset 'code-sizing-system 'specification-statistics 'value 'deterministic)
  (fclear 'program 'predicted-size 'value)
  (fclear-generic-component-vector)
  (assign-specification-values)
  (setq size (round (car (fget-z 'program 'predicted-size))))
  (fset 'program 'predicted-size 'most-likely size)
  (fset 'code-sizing-system 'specification-statistics 'value 'stocastic)
  (dotimes (i (car (fget-z 'code-sizing-system 'number-of-iterations)))
   (fclear 'program 'predicted-size 'value)
   (fclear-generic-component-vector)
   (assign-specification-values)
   (push-end (car (fget-z 'program 'predicted-size)) sizes))
  (fset 'program 'predicted-size 'statistical-sample sizes)
  (setq mean (mean sizes))
  (setq sizes (mapcar '(lambda (s) (/ s mean)) sizes)) ; normalize
  (setq fluct
        (exp (sqrt (+ (expt (sigma (mapcar 'log sizes)) 2)
                      (expt (log (car (fget 'code-sizing-system
                                            'fluctuation 'value))) 2)))))
  (setq fluct (read-from-string (format nil "~6,1f" fluct) t))
  (fset 'program 'predicted-size 'fluctuation fluct)
  (send self :GoSub 'results)
  )


;; Get a user specification. There are three types; for numeric specs,
```

```
;; pick an alternative (1) and set the certainty (2), and set the certainty
;; for qualitative specs (3). Certainties are input through ScrollBarObjects,
;; and alternatives are selected through ButtonGroupObjects.

(defmethod (Spec-Screen :User-Specs) (Obj &rest ignore)
  (send Obj :User-Specs (send Obj :spec-frame) (send Obj :slot)))

(defmethod (ButtonGroupObject :User-Specs) (frame slot)
  (fset frame slot 'most-likely (nth selection (fget frame slot 'range))))

(defmethod (ScrollBarObject :User-Specs) (frame slot)
  (fplace frame slot (send self :fraction))
  )

(defmethod (Spec-Screen :Get-Help) (Obj &rest ignore)
  (send self :GoSub 'Help (send Obj :spec-frame) (send Obj :slot)))

(defmethod (Spec-Screen :Screen-Help) (Obj &rest ignore)
  (send self :GoSub 'Help 'screen (send Obj :slot)))

(defmethod (Screen :Screen-Help) (Obj &rest ignore)
  (send self :GoSub 'Help 'screen (send Obj :slot)))

;; Initialize the user interface

(defmethod (screen :initialize) (&rest ignore)
  (fset 'code-sizing-system 'active-application 'value self)
  (initialize-sizing)
  (dolist (spec (specification-features))
    (if (vector-spec (second spec))
          (fset (first spec) (second spec) 'most-likely
               (lowest-value (first spec) (second spec))))
    (fset (first spec) (second spec) 'certainty 0.)
    (if (member (first spec) FrameList) t
          (push-end (first spec) FrameList)))
  (load-help))

(defun load-help (&optional (help-file "help") &aux frame slot line)
  (with-open-file (he help-file)
    (loop (setq frame (read he))
            (setq slot  (read he))
            (read-line he)
            (loop
              (setq line (read-line he nil))
              (if (null line) (return-from load-help))
              (if (equal line "") (return))
              (fput frame slot 'help line)))))

(defmethod (ApplicationEdit :before :initialize) (&rest ignore)
  (pkg-goto 'tb)
  )

(defmethod (Spec-Screen :initialize) (&rest ignore &aux frame slot)
  (fset 'code-sizing-system 'active-application 'value self)
  (dolist (object DisplayList)
    (cond ((setq frame (send object :spec-frame))
             (setq slot (send object :slot))
             (send object :set-up frame slot)
             (send object :put-object frame slot))))
  )

(defmethod (ButtonGroupObject :put-object) (frame slot)
  (fset frame slot 'ButtonGroupObject self))
```

```
(defmethod (ScrollBarObject :put-object) (frame slot)
 (fset frame slot 'ScrollBarObject self))

(defmethod (ButtonGroupObject :set-up) (frame slot)
 (send self :Set-Selection-Index nil)
 (send self :Set-Selection-Index
        (position (car (fget frame slot 'most-likely)) (fget frame slot 'range))))

(defmethod (ScrollBarObject :set-up) (frame slot &aux val)
 (setq val (round (+ (* (car (fget frame slot 'certainty))
                          (- (send self :max) (send self :min)))
                     (send self :min))))
 (send self :Set-Value val)
 )

(defmethod (ButtonObject :set-up) (frame slot))

(defmethod (ButtonObject :put-object) (frame slot))

;; Some routines for dealing with DisplayObjects

(defmethod (ScrollBarObject :fraction) ()
 (float (/ (- (send self :Value) (send self :Min))
           (- (send self :Max)   (send self :Min)))))

(defmethod (DisplayObject :slot) ()
 (intern (string Name) 'tb))

(defmethod (DisplayObject :spec-frame) ()
 (find-frame (send self :slot)))


(defun fclear-generic-component-vector ()
 (mapcar #'(lambda (f-s) (fclear (first f-s) (second f-s) 'value))
         (generic-features)))

(defun show-specs (&aux s f)
 (dolist (spec (specification-features))
  (setq f (car spec))
  (setq s (second spec))
  (format t "~% ~20s ~30s ~10s ~s" f s
           (car (fget f s 'value))
           (car (fget f s 'certainty)))))
```

File hd:nasa:production system:size-net.lisp

```lisp
;;; -*- Mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-

; Code to set up a neural net run from training set examples

; These globals are used to determine the characteristics of the network
; and the relationship between output node excitations and code size

(defvar hidden-layer t  "not nil if there is a hidden layer")
(defvar output-nodes 1  "the number of nodes in the output layer")
(defvar out-base 1.4    "for multiple output nodes, incremental factor")
(defvar out-linear nil  "t is its linear (with no hidden layers)")
(defvar net-features nil "features input to the net")

; this section creates and calibrates networks based on program frames
; with a set of 'sizing-criteria

(defun set-up-sizing (&optional (plist (fget-z 'program 'children)))
 (setq net-features (get-features))       ; Generic software components
 (if hidden-layer                          ; two or three layers
    (set-up-net (list (length net-features)
                              (/ (length net-features) 2) output-nodes))
    (set-up-net (list (length net-features) output-nodes)))
 (get-samples plist))

; get (list-of-input-vectors list-of-output-vectors)

(defun get-samples (programs &aux tail)
 (cond ((null programs) nil)
        (t (setq tail (get-samples (cdr programs)))
          (list (cons (in-vect (car programs))
                      (first tail))
                (cons (outfun (fget-z (car programs) 'total-size))
                      (second tail))))))

; These functions define the relationship between output node excitations
; and code sizes, depending on the global network parameters

(defun outfun (size-list)   ; node-excitations(code-size), always
 (if (= output-nodes 1)     ; takes and gives LISTS
    (if out-linear
          (outf (car size-list))
          (outf2 (car size-list)))
    (outf1 (car size-list))))

(defun outf (size)
 (list (/ size 1.0e2)))

(defun outf1 (size &aux pos)
 (setq pos (max 0 (round (log size out-base))))
 (if (>= pos output-nodes) "error, size too large"
    (ou pos 0)))

(defun outf2 (size)
 (list (max 0. (log size 1000))))

(defun ou (pos place &aux element)
 (cond ((= place output-nodes) nil)
        (t (if (= pos place)(setq element 1) (setq element 0))
          (cons element (ou pos (1+ place))))))

(defun revoutfun (size-list)   ; code-size(node-excitations), always
 (if (= output-nodes 1)        ; takes and gives LISTS
```

```
    (if out-linear
            (rev (car size-list))
            (rev2 (car size-list)))
    (rev1 size-list)))

(defun rev (size)
  (list (* size 1.0e2)))

(defun rev1 (size-list &aux (mval (car size-list)) (mpos 0))
  (do ((l (cdr size-list) (cdr l)) (pos 1 (1+ pos)))
     ((null l) (list (expt out-base mpos)))
   (cond ((> (car l) mval)
            (setq mpos pos)
            (setq mval (car l))))))

(defun rev2 (size)
  (list (expt 1000 size)))


; Determine the bias and fluction factors of the net for
; a set of programs

(defun show-results (&optional (proglist (fget-z 'program 'children))
                              &aux set)
  (setq set (mapcar 'log (view-results proglist)))
  (format t "~%~%  Bias factor = ~5,2f       Fluction factor = ~5,2f"
            (exp (mean set)) (exp (sigma set))))

(defun view-results (proglist &aux size pred res)
  (format t "~%~20s ~10s ~10s ~10s~%" 'program 'actual 'predicted 'p/a)
  (dolist (prog proglist res)
    (setq size (car (fget-z prog 'total-size)))
    (setq pred (predicted-size prog))
    (format t "~%~20s ~10s ~10s ~10s" prog size  pred (/ pred size))
    (push-end (/ pred size) res)))


; 17 NOV 89

; This code tests program bases with a slightly different organization than
; the code above. Sizing criteria are ENTITIES such as METHOD, GRAPHICS, and
; BOUNDARY. Each criterion has a set of ATTRIBUTES such as COMPLEXITY,
; VERSATILITY, and ACCURACY. Each attribute has a possible set of values such
; as VERY HIGH, HIGH, MEDIUM, etc. There is an input node in the NET for each
; (ENTITY ATTRIBUTE) pair with the excitation proportional to the VALUE.

; Program bases are tested calibrating nets with a CALIBRATION SET of
; programs and then testing the net with the TEST SET.

(defun test-prog-base (&optional (plist (fget-z 'program 'children))
                               (test-list plist) &aux io tlist frame)
  (setq frame (instance 'test-prog-base))
  (fput frame 'set-up 'start (time:print-current-time nil :|dd mmm yy|))
  (dolist (prog test-list)
    (setq tlist (remove prog plist))
    (setq io (set-up-sizing (setq tlist (remove prog plist))))
    (test-iteration io prog tlist frame)
   )
  (end-results frame plist test-list)
)

(defun end-results (frame plist test-list &aux iter err io)
  (fput frame 'set-up 'eta eta)
  (if hidden-layer (fput frame 'set-up 'alpha alpha))
```

```
 (setq io (set-up-sizing plist))
 (setq iter
         (dotimes (i 200 i)
           (learn (first io) (second io) 1)
           (if ( < (setq err (size-err plist)) .2) (return (1+ i)))
           (format t "~%System calibration iteration ~d error ~s" i err)))
 (fput frame 'set-up 'end (time:print-current-time nil :|dd mmm yy|))
 (fput frame 'set-up 'error err)
 (fput frame 'set-up 'iterations iter)
 (fput frame 'set-up 'calibration (get-calibration plist))
 (fput frame 'set-up 'net (mapcar 'length layers))
 (fput frame 'set-up 'random-results (get-random plist))
 (fput frame 'set-up 'test-results (get-test frame test-list))
 )

(defun test-iteration (io prog tlist frame &aux err iter pred act)
 (setq iter
         (dotimes (i 200 i)
           (learn (first io) (second io) 1)
           (if ( < (setq err (size-err tlist)) .2) (return (1+ i)))
           (format t "~%program ~s iteration ~d error ~s" prog i err)))
 (format t "~%Program ~s Predicted ~3,1f Actual ~3,1f" prog
           (setq pred (predicted-size prog))
           (Setq Act (Car (Fget-Z prog 'total-size))))
 (fput frame prog 'predicted-size pred)
 (fput frame prog 'actual-size act)
 (fput frame prog 'error err)
 (fput frame prog 'iterations iter)
 (fput frame prog 'calibration (get-calibration tlist))
 (fput frame prog 'net (mapcar 'length layers))
)

(defun get-test (frame plist &aux logs) ; changed push to push-end 17 NOV 89
 (dolist (prog plist)
  (push-end (log (/ (car (fget frame prog 'predicted-size))
                    (car (fget-z prog 'total-size)))) logs))
 (list 'bias (exp (mean logs)) 'fluct (exp (sigma logs))))

(defun get-random (plist &aux logs) ; changed push to push-end 17 NOV 89
 (dolist (prog1 plist)
  (dolist (prog2 plist)
   (push-end (log (/ (car (fget-z prog1 'total-size))
                     (car (fget-z prog2 'total-size)))) logs)))
 (list 'bias-factor (exp (mean logs))
         'fluctuation-factor (exp (sigma logs))))

(defun get-calibration (tlist &aux logs) ; changed push to push-end 17 NOV 89
 (dolist (prog tlist)
  (push-end (log (/ (predicted-size prog)
                    (car (fget-z prog 'total-size)))) logs))
 (list 'bias (exp (mean logs)) 'fluct (exp (sigma logs))))

; Determine the RMS value of LOG(PRED/ACT) from a list of programs

(defun size-err (plist)
 (rms (mapcar
         #'(lambda (p)
            (log (/ (predicted-size p)
                    (car (fget-z p 'total-size))))) plist)))

; The predicted size of a program by the NET

(defun predicted-size (prog)
 (car (revoutfun (forward (in-vect prog)))))
```

```
; This file redefines two functions for testing program databases:
; GET-FEATURES and IN-VECT. The idea is to have a net with fewer nodes.
; Instead of an input node for each (CRITERION ATTRIBUTE VALUE) triplet,
; there is only one for each (CRITERION ATTRIBTE) pair, with the excitation
; being a function of the VALUE.


(defun excitation (v)
 (second (assoc v '((very-high 1.) (high .75) (medium .5)
                    (low .25) (very-low 0.)))))

; We now use a system where all nodes are used

(defun get-features (&aux fs) ; don't use program list any more
 (dolist (f (fget-z 'design-criteria 'children) fs)
  (dolist (slot (cdr (show f)))
   (if (member (car slot) '(a-kind-of number-of-instances)) t
   (push-end (list f (car slot)) fs))))) ; changed push to push-end 17 NOV 89

(defun in-vect (prog &aux vect Cinstance)
 (if (null (fget-z prog 'design-criteria))
   (error "Program not in database"))
 (dolist (feat (get-features) vect)
  (setq Cinstance (get-Cinstance (car feat) prog))
  (push-end (excitation (car (fget-z Cinstance (second feat)))) vect)
 ))

(defun get-Cinstance (crit prog &aux Clist Plist)
 (setq Plist (fget-z prog 'design-criteria))
 (if (setq Clist
          (member crit Plist
                  :test #'(lambda (c p) (eq c (has-instance p)))))
   (car Clist) crit))



;  The first routine tests if an atom is a frame.
;  The second makes an instance of a category.
;  The third checks if A is an instance of B.
;  They should eventually be added to FRAME-UTILITIES.LISP

(defun framep (f)
 (if (show f) t nil))


; This routine is used to create an instance of a category

(defun instance (frame &aux inst)
 (setf inst (fget frame 'number-of-instances 'value))
 (cond (inst (setq inst (car inst))
            (fremove frame 'number-of-instances 'value inst)
            (inc  inst 1))
        (t   (setq inst 1)))
 (fput frame 'number-of-instances 'value inst)
 (setq inst (intern (format nil "~g~d" frame inst)))
 (set-frame `(,inst (a-kind-of (value ,frame))))
 inst)

(defun has-instance (inst &aux cat)
 (cond ((not (setq cat (car (fget inst 'a-kind-of 'value)))) nil)
       ((not (equal (string cat)
                    (subseq (string inst) 0 (length (string cat))))) nil)
```

```
        ((not (fget cat 'number-of-instances 'value)) nil)
        (t cat)))
```

```
; This function is used to create a table of reasons

(defun reason-table (&optional (plist (fget-z 'program 'children))
                              file &aux out)
  (if file
     (with-open-file (out file :direction :output) (reason-tab out plist))
     (progn (setq out t) (reason-tab out plist))))

(defun reason-tab (stream plist &aux entries)
  (setq entries (copy-tree (get-entries plist)))
  (stable-sort entries #'(lambda (x y) (> (excitation x) (excitation y)))
              :key 'third)
  (stable-sort entries #'string> :key 'second)
  (stable-sort entries #'string> :key 'car)
  (print-entries stream entries))

(defun print-entries (stream entries &aux e a v)
  (dolist (entry entries (length entries))
   (if (eq e (first entry))         (format stream "~%")
         (progn (setq e (first entry)) (format stream "~%~a" e)))
   (if (eq a (second entry))        (format stream "~c"  #\tab)
         (progn (setq a (second entry)) (format stream "~c~a" #\tab a)))
   (if (eq v (third entry))         (format stream "~c"  #\tab)
         (progn (setq v (third entry)) (format stream "~c~a" #\tab v)))
   (format stream "~c~a~c~a" #\tab (fourth entry) #\tab (fifth entry))))

(defun get-entries (plist &aux entries)
  (dolist (prog plist entries)
   (dolist (crit (fget-z prog 'design-criteria))
    (dolist (slot (cdr (show crit)))
        (if (eq (car slot) 'a-kind-of) t
           (push-end
            (list (car (fget-z crit 'a-kind-of))      ; criterion
                  (car slot)                            ; attribute
                  (car (fget-z crit (car slot)))      ; value
                  (car (fget crit (car slot) 'reason)) ; reason
                  prog) entries)))))                    ; program

;  This code is used to restructure Yingti's application tree for
;  display with the tree utility.


(defun restructure-tree (root)
  (dolist (ch (fget root 'children 'range))
   (fput root 'children 'value ch)
   (restructure-tree ch)))
```

File hd:nasa:knowledge acquisition system:create-implementation.lisp

```
;;; -*- Mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-

; This will be used to make rules & procedures out of the
; generic net correlations

(defun list-input-node-names (&aux (count 0))
  (dolist (f (fget-z 'specification 'criteria) count)
    (dolist (s (fget-z 'specification f))
     (format t "~%~35s ~s" s f )
     (inc count 1))))


(defun input-node-names (&aux nodes)
  (dolist (f (fget-z 'specification 'criteria) nodes)
    (dolist (s (fget-z 'specification f))
;     (if (vector-spec s)
;            (dolist (q '(very-high high medium low very-low))
;              (push-end (list q s f) nodes))
             (push-end (list s f) nodes)
;            )
    )))

(defun list-weights (&optional (outch t) &aux in out)
  (setq in (input-node-names))
  (setq out (get-features))
  (dotimes (j (length out))
   (dotimes (i (length in))
     (format
          outch "~%~s~c~s~c~s" (nth i in) #\tab (nth j out) #\tab
          (weight-val i (+ (length in) j))))))

(defun weight-val (in-node out-node)
  (get (intern (format nil "W_~s-~s" in-node out-node )) 'val))

(defun bias-val (node)
  (get (intern (format nil "N~s" node)) 'val))


(defun set-up-imp-1 (&aux glist plist inst)
  (set-frame '(test-system))
  (setq plist (fget-z 'program 'specified))  ; progs w. specifications
  (setq glist (fget-z 'program 'children))   ; progs w. generic components
  (setq inst (initialize-test plist))
  (set-up-net-architecture nonlinear-generic-net)
  (calibrate-generic-net inst plist (set-up-generic plist))
 )


(defun set-up-imp-2 (&aux glist plist (inst 'test-system1))
  (setq plist (fget-z 'program 'specified))  ; progs w. specifications
  (setq glist (fget-z 'program 'children))   ; progs w. generic components
  (set-up-net-architecture nonlinear-sizing-net)
  (calibrate-sizing-net inst (set-up-sizing glist) plist)
 )


(defun set-up-implementation ()
  (initialize-system)
  (setq max-iter 10)
  (set-up-net-architecture nonlinear-generic-net)
  (calibrate-generic-net
```

```
  'test (fget-z 'program 'specified) (set-up-generic)))

(defun save-implementation (&optional (file "implementation")
                                      (source (create-implementation))
                                      &aux (f-s (third source)))
  (with-open-file (out file :direction :output)
    (put-in-header out)
    (small-print `(defun initialize-frames ()
                    (set-frame-list ',(mapcar 'show (car source)))) out)
    (put-in-function-header out)
    (dolist (routine (second source))
      (format t "~% Routine ~s for ~s" (second routine) (car f-s))
      (put-in-routine-header (first (car f-s)) (second (car f-s)) out)
      (small-print routine out)
      (setq f-s (cdr f-s)))
   ))

(defun small-print (form &optional stream &aux (OldCase *print-case*))
  (setq *print-case* :downcase)
  (pprint form stream)
  (setq *print-case* OldCase))

(defun short-version (&optional (imp implementation))
  (list (first imp) (list (first (second imp)) (second (second imp)))
        (list (first (third imp)) (second (third imp)))))

(defun put-in-header (ch)
  (format ch ";;; -*- Mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-")
  (format ch "~%~%;;; This implementation was created on ~a"
          (time:print-current-time nil :|dd mmm yy|))
  (format ch "~%~%;;  The initialization function sets up frames for the")
  (format ch "~%;;  specifications and generic components of the program.~%")
  )

(defun put-in-function-header (ch)
  (format ch "~%~%~%;;  This section contains the IF-NEEDED routines")
  )


(defun put-in-routine-header (frame slot ch)
  (format ch "~%~%~%;;  The IF-NEEDED routine for frame ~s, slot ~s~%" frame slot))

(defun create-implementation (&aux frames generics)
  (setq frames (fget-z 'specification 'criteria))
  (dolist (spec (fget-z 'design-criteria 'children))
    (if (member spec frames) t (push-end spec frames)))
  (set-up-features)
  (setq generics (get-features))
  (fput-list 'sizing-criteria 'output-nodes generics)
  (push 'sizing-criteria frames)
  (Dolist (f (cdr frames))
    (set-frame (list f)))
  (dolist (spec (fget-z 'specification 'criteria))
    (dolist (slot (fget-z 'specification spec))
      (fput spec 'specification-slots 'value slot)
      (fput spec slot 'if-needed 'user-supplied)
      (dolist (val (determine-range slot))
        (fput spec slot 'range val))))
  (dolist (gen generics)
    (fput (first gen) 'generic-slots 'value (second gen))
    (fput (first gen) (second gen) 'if-needed
          (if-needed-name (first gen) (second gen))))
  (list frames (if-needed-routines generics) generics))
```

```
(defun determine-range (slot)
 (cond   ((eq slot 'spatial-dimension) '(3 2 1 0))
         ((vector-spec slot) '(very-high high medium low very-low))
         (t '(yes no))))

(defun if-needed-routines (generic-nodes &aux specs routines (count -1))
  (dolist (criterion (fget-z 'specification 'criteria))
    (dolist (aspect (fget-z 'specification criterion))
      (multiple-value-setq (specs count)
           (node-number specs count criterion aspect))))
  (dolist (gen generic-nodes routines)
   (push-end
     (create-routine (first gen) (second gen) (inc count 1) specs)
     routines))
 )

(defun create-routine (frame slot count specs &aux funct)
  (setq funct `(defun ,(if-needed-name frame slot)
                          (frame slot &aux val)
                       (setq val ,(bias-val count))))
  (dolist (sp specs)
    (setq funct (append funct (apply 'make-clauses (append sp (list count))))))
  (append funct '((fput-list frame slot val))))

(defun make-clauses (frame slot in-node out-node)
  (cond ((eq slot 'spatial-dimension)
          `((inc val (* (car (fget-z ',frame ',slot))
                        ,(weight-val in-node out-node)))))
      ((atom in-node)
          `((iff ',frame ',slot 'yes
                  (inc val ,(weight-val in-node out-node)))))
         (t
          (mapcar
            #'(lambda (resp)
               `(iff ',frame ',slot ',resp
                      (inc val ,(weight-val (pop in-node) out-node))))
            (determine-range slot)))
         ))

;;; Using the specification CRITERION and ASPECT, and a running node COUNT,
;;; update SPECS, a list of (CRITERION ASPECT NODE-NUMBER) triplets.
;;; NODE-NUMBER is a vector for "quantitative" specifications.

(defun node-number (specs count criterion aspect &aux num)
  (if (and (vector-spec aspect) (not (eq aspect 'spatial-dimension)))
     (dotimes (i (length (determine-range aspect)))
          (push-end (inc count 1) num))
     (setq num (inc count 1)))
  (push-end (list criterion aspect num) specs)
  (return-from node-number specs count))

(defun if-needed-name (frame slot)
  (intern (format nil "~s-~s" frame slot)))


;;; Code to create rule base description


(defun save-rule-base-description (&optional (file "rule-base")
                                             &aux specs (count -1))
  ;; Load in the names of the generic components and specifications.
;  (initialize-system)
  ;; code in IMPLEMENTATION-CONTROL get specification net
;  (initialize-sizing-net "generic-net-state")
```

```
;; create the list of (CRITERION ASPECT NODE-NUMBER) triplets for specifications
 (multiple-value-setq (specs count) (get-node-numbers specs count))
 (with-open-file (out file :direction :output)
  (put-in-top-header out)
  (dolist (generic (get-features)) ;rules for each generic component
   (put-in-generic-header (first generic) (second generic) out)
   (put-in-generic-rules (first generic) (second generic)
                         specs (inc count 1) out))
  )
)

(defun put-in-top-header (ch)
 (format ch "Code Sizing System Rule Base"))

(defun put-in-generic-header (frame slot ch)
 (format ch "~%~%~%~%Rules for ~s ~s" frame slot))

(defun get-node-numbers (specs count)
 (dolist (criterion (fget-z 'specification 'criteria))
  (dolist (aspect (fget-z 'specification criterion))
   (multiple-value-setq (specs count)
       (node-number specs count criterion aspect))))
 (return-from get-node-numbers specs count))

(defun put-in-generic-rules (frame slot specs node out)
 (format out "~%~%~%If the ~s ~s is being determined, " frame slot)
 (format out "Then initialize the value of the ~s ~s to ~s"
        frame slot (bias-val node))          ; The Bias rule
 (dolist (sp specs)
  (put-in-rule frame slot (first sp) (second sp)
               (third sp) node out)
  )
 )

(defun put-in-rule (frame slot criterion aspect in-node out-node ch)
 (cond ((eq aspect 'spatial-dimension)
        (put-in-r frame slot criterion aspect "X"
                  (format nil "~s * X" (weight-val in-node out-node)) ch))
       ((atom in-node)
        (put-in-r frame slot criterion aspect "YES"
                  (format nil "~s" (weight-val in-node out-node)) ch))
       (t
        (dolist (r (determine-range aspect))
         (put-in-r frame slot criterion aspect (string r)
                   (format nil "~s" (weight-val (pop in-node) out-node)) ch)))
       ))

(defun put-in-r (frame slot criterion aspect val-in val-out ch)
 (format ch "~%If the value of ~s ~s is ~a, " criterion aspect val-in)
 (format ch "Then increment the ~s ~s by ~a" frame slot val-out))
```

File hd:nasa:knowledge acquisition system:database-parser.lisp

```lisp
;;; -*- Mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-


(defvar spec-database-file
        "lm:hd:nasa:knowledge acquisition system:specification-base")
(DEFVAR DB-FILE "lm:hd:nasa:knowledge acquisition system:generic-base")


(defun initialize-system
    (&optional (sizing-file db-file) (generic-file spec-database-file))
  (init-db sizing-file)
  (make-specification-database generic-file)
  (set-frame '(test-sizing-system)))

; ***********************************************************

; This part is for the Program database... generic components to Size

; ***********************************************************


(DEFUN init-db (&optional (FILE db-file) (set-up-flag t)
                &aux inp toks pname feature)
  (if set-up-flag (set-up-features))
  (setq inp (open FILE))
  (loop
   (if (eq nil (setq toks (next-tokens inp))) (return))
;   (print toks)
   (cond ((= (length toks) 5)
          (if (framep pname) (print pname))          ; (show-prog pname)
          (setq pname (set-frame (list (pop toks))))  ; blank out prog frame
          (fput 'program 'children 'value pname)))    ; add to program list
   (if (= (length toks) 4)
       (cond ((eq (car toks) 'size)
              (fput pname 'total-size 'value (third toks))
              (setq toks nil))
             (t (fput pname 'design-criteria 'value
                      (setq feature (instance (pop toks)))))))
   (cond ((= (length toks) 3)
          (fput feature (first toks) 'value  (second toks))
          (fput feature (first toks) 'reason (third toks))))
   )
  (CLOSE INP)
  )

(defun next-tokens (inp &AUX ntok LINE EOF TOK (CNT 0) tlist)
  (LOOP
   (MULTIPLE-VALUE-SETQ (LINE EOF) (READ-LINE INP NIL T))
   (IF EOF (RETURN-FROM next-tokens nil))
   (setq line (string-trim '(#\space) line))
   (if (string-equal "end" line :end2 3) (return-from next-tokens nil))
   (if (and (> (length line) 3)
            (not (string-equal "program" line :end2 7)))
       (return)))
  (cond ((not (white-spacep (aref line 0))) (setq ntok 4))
        ((not (white-spacep (aref line 1))) (setq ntok 3))
        (t (setq ntok 2)))
  (dotimes (i ntok)
   (MULTIPLE-VALUE-SETQ (TOK CNT)
    (READ-FROM-STRING LINE NIL T :START CNT))
     (push-end tok tlist))
  (push-end (subseq line cnt) tlist)
```

```
 )

; SET-UP-FEATURES creates the feature hierarchy

(defun set-up-features ()
 (set-frame-list
  '((program
        (a-kind-of (value thing)))
   (design-criteria
        (a-kind-of (value thing))
        (children (value task methods objects representation graphics
                         user-interface data-management sys-programming
                         hard-architecture)))
   (task
        (a-kind-of  (value design-criteria))
        (complexity))
   (methods
        (a-kind-of  (value design-criteria))
        (complexity)
        (generality)
        (accuracy   (default medium))
        (efficiency (default medium))
        (redundancy (default very-low)))
   (objects
        (a-kind-of  (value design-criteria))
        (complexity)
        (generality))
   (representation
        (a-kind-of  (value design-criteria))
        (complexity (default medium))
        (capacity   (default medium))
        (knowledge))
   (graphics
        (a-kind-of  (value design-criteria))
        (complexity (default none)))
   (user-interface
        (a-kind-of  (value design-criteria))
        (complexity))
   (data-management
        (a-kind-of  (value design-criteria))
        (complexity (default very-low)))
   (sys-programming
        (a-kind-of  (value design-criteria))
        (complexity (default very-low)))
   (hard-architecture
        (a-kind-of  (value design-criteria))
        (complexity (default very-low)))
  )))

; This routine checks if a character is whitespace. May be put
; into UTILITIES.LISP at some point

(defun white-spacep (char)
 (if (member char '(#\space #\newline #\tab)) t))


;*****************

(defun show-prog (pname &aux (space ""))
 (format t "~%~10s" pname)
 (dolist (crit (fget-z pname 'design-criteria))
  (format t "~a~20s" space crit)
  (setq space "")
  (dolist (slot (cdr (show crit)))
```

```
    (format t "~a~15s~15s~%" space
             (car slot) (car (fget-z crit (car slot))))
    (setq space (make-string 30 :initial-element #\space)))
   (setq space (make-string 10 :initial-element #\space))))


; ************************************************************

; This part is for the Specification database
; ... specifications to generic components

; ************************************************************



; This is used to initialize the program specifications

(defun make-specification-database
      (&optional (file spec-database-file) (init-flag t) &aux toks programs)
  (if init-flag (set-frame '(specification)))
  (with-open-file (inp file)
   (loop
     (setq toks (next-toks inp))
     (cond ((or (null toks) (equal toks '(end)))
             (return-from make-specification-database programs))
            ((eq (car toks) 'program)
             (format t "~%~%~%Programs:  ~s~%" (setq programs (cdr toks)))
             (fput-list 'program 'specified programs)
             (setq toks (next-toks inp))))
     (cond ((= (length toks) (+ 2 (length programs)))
             (print toks)
             (fput 'specification 'criteria 'value (first toks))
             (fput 'specification (first toks) 'value (second toks))
             (dotimes (i (length programs))
               (fput (nth i programs) (first toks) (second toks)
                     (nth (+ 2 i) toks))))))))

(defun next-toks (inp &AUX LINE EOF TOK CNT tlist)
 (LOOP
  (MULTIPLE-VALUE-SETQ (LINE EOF) (READ-LINE INP NIL T)) ; read line
  (IF EOF (RETURN-FROM next-toks nil))                    ; eof test
  (setq cnt 0)                                ; place in line
  (loop                                       ; for each token
   (MULTIPLE-VALUE-SETQ (TOK CNT)
         (READ-FROM-STRING LINE NIL 'eof :START CNT)) ; read token
   (if (eq tok 'eof) (return)                 ; done with line
          (push-end tok tlist)))              ; list of tokens
  (if (consp tlist) (return-from next-toks tlist)) ; skip blank lines
  ))
```


C.3

File hd:nasa:knowledge acquisition system: fortran-parser.lisp

```
;;; -*- Mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-

(defvar treelist nil "The tree list generated by make-calling-tree.")

(defvar executable 0 "The number of executable lines of code.")
(defvar comments 0 "The number of comment lines in the code.")

(defvar source-line 0 "A count of the source lines read.")

(defvar white-space '(#\space #\tab #\newline) "White space.")

(defvar lineBuf "")

(defvar reservedwords
        '(BYTE LOGICAL INTEGER REAL DOUBLEPRECISION COMPLEX
          DOUBLECOMPLEX CHARACTER .TRUE. .FALSE. .LT. .LE. .EQ.
          .NE. .GT. .GE. .NOT. .AND. .OR. .XOR. .NEQV. .EQV.
          ACCEPT ASSIGN BACKSPACE BLOCK BLOCKDATA CALL CLOSE COMMON
          CONTINUE DATA DELETE DIMENSION DO DOWHILE ELSE ELSEIF
          END ENDDO ENDFILE ENDIF ENDMAP ENDSTRUCTURE ENDUNION
          ENTRY EQUIVALENCE EXTERNAL FILE FIND FORMAT FUNCTION GOTO
          IF IMPLICIT IMPLICITNONE INCLUDE INQUIRE INTRINSIC MAP
          NAMELIST NONE OPEN OPTIONS PARAMETER PAUSE PRINT PROGRAM
          READ RECORD RETURN REWIND REWRITE SAVE STOP STRUCTURE
          SUBROUTINE TYPE UNION UNLOCK VIRTUAL VOLATILE WHILE WRITE)
 "The list of the reserved words of FORTRAN as symbols.")

(defvar operators
        '(plus minus times divided power .LT. .LE. .EQ. .NE. .GT. .GE. .NOT. .AND.
          .OR. .XOR. .NEQV. .EQV. concatenation)
 "The list of FORTRAN operators.")

(defvar currentToken nil "The last token read from the file.")
(defvar currentLine "")
(defvar currentIndex 0)

(defvar modulelist nil "A list of the modules parsed.")
(defvar default-module-name 'program$main "default module name")
(defvar currentmodule default-module-name "The name of the module being parsed.")
(defvar arrays nil "The list of names of arrays in a module.")
(defvar calls nil "A list of the calls made in the current module.")
(defvar skip-lines nil "The number of lines of source code to skip")

(defun make-calling-tree (&optional
                           (inFile "lm:hd:action:fortran folder:nbod2.for"))
 (initparse)
 (with-open-file (inStrm inFile)
  (cond ((and (boundp 'skip-lines) (numberp skip-lines))
         (dotimes (jj skip-lines) (read-line inStrm nil))
         (setq source-line skip-lines)
         ))
  (parseStream inStrm)
  )
 (cleanupparse)
 (make-tree treelist)
)

(defun cleanupparse ()
)

(defun initparse ()
 (setq source-line 0)
```

```
 (setq treelist nil)
 (setq executable 0)
 (setq comments 0)
 (setq lineBuf "")
 (setq modulelist nil)
 (setq arrays nil)
 (setq calls nil)
 )


(defun parseStream (inStrm &aux eof)
 (while (not eof)
          (setq eof (getbigline inStrm))
          (gettoken)
          (parsestat)
          )
 )

(defun getbigline (inStrm &aux (more t) line eof)
 (if (string= "" lineBuf)
    (setq lineBuf (getline inStrm)))
 (setq line (copy-seq (subseq lineBuf 1)))
 (while (and lineBuf more)
          (multiple-value-setq (lineBuf eof) (getline inStrm))
          (cond ((and (not eof)
                        (char/= #\space (aref lineBuf 0)))
                    (setq line (string-append line (copy-seq (subseq lineBuf 1))))
                    )
             (t
                    (setq more nil))
             )
          )
 (setq currentLine line)
 (setq currentIndex 0)
 eof
 )

; getline reads the next non-comment line from inStrm, strips it via cleanupline,
; and returns the result.  At end-of-file, getline returns nil.

(defun getline (inStrm &aux line eof gotaline)
 (setq source-line (1+ source-line))
 (cond ((zerop (mod source-line 100))
         (print "reading line ")
         (princ source-line)))
 (while (not (or eof gotaline))
          (multiple-value-setq (line eof) (read-line inStrm nil))
          (setq line (string-upcase line))
          (cond ((not eof)
                    (cond ((or               ; modified by DSF 10-apr-89
                              (blank-line-p line)
                              (member (aref line 0) '(#\C #\D #\c #\d #\* #\!)))
                         (incf comments)
                        )
                        (t
                         (setq line (cleanupline line))
                         (incf executable)
                         (setq gotaline t)
                        )
                        )
                    )
             )
          )
 (values line eof)
```

```
 )

(defun blank-line-p (line)
 (zerop (length (string-left-trim white-space line))))


; cleanupline takes line and strips out all extraneous characters (any label,
; the first tab, spaces not appearing in strings, and all characters after
; column 72) and returns the resulting string. The result always looks like
; 1 character for continuation followed by the statement portion.

; NOTE: cleanupline changed 3-1-89 to remove all but one space from between
; tokens.

(defun cleanupline (line &aux tabPos newline c inString spaces)
 (setq tabPos (dotimes (i 6)
                 (if (char= (aref line i) #\tab)
                   (return i)))
       )
 (cond (tabPos
        (setq c (aref line (1+ tabPos)))
        (cond ((and (char>= c #\0)
                    (char<= c #\9))
               (setq newLine (string c))
               (setq line (subseq line (+ 2 tabPos)))
               )
          (t
               (setq newLine " ")
               (setq line (subseq line (1+ tabPos)))
               )
          )
        )
       (t
        (setq newLine (string (aref line 5)))
        (setq line (subseq line 6))
        ))
 (setq spaces 1)
 (dotimes (i (min (length line) 66))
  (setq c (aref line i))
  (if (char= #\' c)
       (setq inString (not inString)))
  (if (and (not inString)
           (char= #\space (aref line i)))
       (incf spaces)
       (setq spaces 0))
  (cond ((or inString
             (and (not inString)
                  (< spaces 2)))
          (setq newline (string-append newline (aref line i)))
          )
         )
  )
 newline
 )

(defun gettoken (&aux c)
 (setq currentToken nil)
 (setq c (next-nonwhite-Char))
 (if (char= #\: c)  (setq c (next-nonwhite-Char)))
 (cond ((char= c #\,)
        (setq currentToken 'comma))
       ((char= c #\()
        (setq currentToken 'lparen))
       ((char= c #\))
```

```
            (setq currentToken 'rparen))
           ((char= c #\+)
            (setq currentToken 'plus))
      ((char= c (code-char 138))
       (setq currentToken 'lf)
            (setq c (nextChar)))
           ((char= c #\-)
            (setq currentToken 'minus))
           ((char= c #\*)
            (readstar))
           ((char= c #\/)
            (readslash))
           ((char= c #\=)
            (setq currentToken 'equals))
           ((char= c #\')
            (readchars))
           ((char= c #\.)
            (readperiod))
           ((or (and (char<= #\0 c)
                     (char>= #\9 c))
              (char= #\& c))
            (readnum-H c))              ;changed by dsf 7-april-89
           ((or (and (char<= #\A c)
                     (char>= #\Z c))
              (and (char<= #\a c)
                     (char>= #\z c))
              (char= #\% c))
            (readword c))
           )
 currentToken
 )

(defun readstar (&aux c)
 (setq c (nextChar))
 (cond ((char= #\* c)
            (setq currentToken 'power))
           (t
            (setq currentToken 'times)
            (decf currentIndex)
            ))
 )

(defun readslash (&aux c)
 (setq c (nextChar))
 (cond ((char= #\/ c)
            (setq currentToken 'concatenation))
           (t
            (setq currentToken 'divided)
            (decf currentIndex))))

(defun readchars (&aux c)
 (setq c (nextChar))
 (while (char/= c #\')
            (setq c (nextChar))
            )
 (setq c (nextChar))
 (cond ((member c '(#\H #\O #\h #\o))
            (setq currentToken 'number))
           (t
            (setq currentToken 'chars)
            (unGetChar c))
           )
 )
```

```
(defun readnum (&aux c)
 (setq c (next-nonwhite-Char))
 (while (or (and (char<= #\0 c)
                 (char>= #\9 c))
            (decimal-pt c))
        (setq c (next-nonwhite-Char))
        )
 (cond ((member c '(#\E #\D #\Q #\e #\d #\q))
        (setq c (next-nonwhite-Char))
        (if (member c '(#\+ #\-))
            (setq c (next-nonwhite-Char)))
        (while (and (char<= #\0 c)
                    (char>= #\9 c))
               (setq c (next-nonwhite-Char))
               )
        )
       )
 (unGetChar c)
 (setq currentToken 'number)
 )

(defun readnum-H (c &aux numstr hol)       ; dsf 7-apr-89 for possible
 (setq numstr (string-append "" c))        ; HOLLERITH STATEMENTS
 (setq c (nextChar))
 (while (or (and (char<= #\0 c)
                 (char>= #\9 c))
            (decimal-pt c)
            (char= #\: c))              ; dsf 10-apr-89 for char substrings
        (setq numstr (string-append numstr c))
        (setq c (nextChar))
        )
 (cond ((member c '(#\H #\h))           ; a Holerith
        (dotimes (hol (read-from-string numstr))
         (nextChar))
        (setq hol t))
     ((member c '(#\E #\D #\Q #\e #\d #\q))
        (setq c (next-nonwhite-Char))
        (if (member c '(#\+ #\-))
            (setq c (next-nonwhite-Char)))
        (while (and (char<= #\0 c)
                    (char>= #\9 c))
               (setq c (nextChar))
               )
        )
       )
 (cond (hol (setq currentToken 'chars))
     (t  (setq currentToken 'number)
            (unGetChar c)))
 )


(defun decimal-pt (c &aux flag)
 (cond ((char= #\. c)
        (ungetChar (next-nonwhite-Char))
        (cond ((or (string= "LT."
                            (subseq currentLine currentIndex (+ currentIndex 3)))
                   (string= "LE."
                            (subseq currentLine currentIndex (+ currentIndex 3)))
                   (string= "EQ."
                            (subseq currentLine currentIndex (+ currentIndex 3)))
                   (string= "NE."
                            (subseq currentLine currentIndex (+ currentIndex 3)))
                   (string= "GT."
                            (subseq currentLine currentIndex (+ currentIndex 3)))
```

```
                    (string= "GE."
                              (subseq currentLine currentIndex (+ currentIndex 3)))
                    (string= "NOT."
                              (subseq currentLine currentIndex (+ currentIndex 4)))
                    (string= "AND."
                              (subseq currentLine currentIndex (+ currentIndex 4)))
                    (string= "OR."
                              (subseq currentLine currentIndex (+ currentIndex 3)))
                    (string= "XOR."
                              (subseq currentLine currentIndex (+ currentIndex 4)))
                    (string= "NEQV."
                              (subseq currentLine currentIndex (+ currentIndex 5)))
                    (string= "EQV."
                              (subseq currentLine currentIndex (+ currentIndex 4)))
                   )
                  (setq flag nil)
                  )
             (t
                  (setq flag t)
                  )
             )
         )
        )
 flag
 )

(defun readperiod (&aux c word)
 (setq c (next-nonwhite-Char))
 (cond ((and (char<= #\0 c)
              (char>= #\9 c))
         (readnum))
        (t
         (setq word (string-append "." c))
         (while (char/= #\. c)
                 (setq c (nextChar))
                 (setq word (string-append word c))
                 )
         (setq currentToken (intern word))
         )
        )
 )

(defun readword (c &aux (word ""))
 (while (or (and (char<= #\A c)
                    (char>= #\Z c))
             (and (char<= #\a c)
                    (char>= #\z c))
             (and (char<= #\0 c)
                    (char>= #\9 c))
             (char= #\$ c)
             (char= #\_ c)
             (char= #\: c)
             (char= #\% c))
         (cond ((char/= #\: c)
                 (setq word (string-append word c))))
         (setq c (nextChar))
;        (princ word) (print currentLine)
         )
 (unGetChar c)
 (if (member (intern word) reservedwords)
    (setq currentToken (intern word))
    (setq currentToken (list 'id (intern word))))
 )
```

```
(defun unGetChar (c)
 (cond ((char/= c 0)
          (decf currentIndex)
          ))
 )

(defun next-nonwhite-Char (&aux c)
 (while (member (setq c (nextChar)) white-space))
 c)

(defun nextChar ()
 (cond ((< currentIndex (length currentLine))
          (setq currentIndex (1+ currentIndex))
          (aref currentLine (1- currentIndex)))
         (t
          0)
         )
 )

; Individual statement parsers.

(defun parsestat ()
 (cond ((consp currentToken)
          (namestat))
         ((eq currentToken 'common)
          (commonstat))
         ((eq currentToken 'do)
          (dostat))
         ((eq currentToken 'dowhile)
          (dowhilestat))
         ((eq currentToken 'else)
          (elsestat))
         ((eq currentToken 'elseif)
          (elseifstat))
         ((eq currentToken 'if)
          (ifstat))
         ((eq currentToken 'program)
          (programstat))
         ((eq currentToken 'function)
          (functionstat))
         ((eq currentToken 'subroutine)
          (subroutinestat))
         ((eq currentToken 'end)
          (endstat))
         ((or (eq currentToken 'byte)
              (eq currentToken 'logical)
              (eq currentToken 'integer)
              (eq currentToken 'real)
              (eq currentToken 'doubleprecision)
              (eq currentToken 'complex)
              (eq currentToken 'doublecomplex)
              (eq currentToken 'character))
          (typestat))
         ((eq currentToken 'dimension)
          (dimensionstat))
         ((eq currentToken 'call)
          (callstat))
         ((member currentToken '(block blockdata))
          (blockdatastat))
         )
 )

(defun namestat ()
 (gettoken)
```

```
 (cond ((eq currentToken 'lparen)
          (parsearglist)
          (gettoken)))
 (gettoken)
 (expression)
 )

(defun commonstat ()
 (gettoken)
 (cond ((eq currentToken 'divided)
          (gettoken)
          (gettoken)
          (gettoken)))
 (parsedatalist)
 )

(defun dostat (&aux c)
 (setq c (next-nonwhite-char))
 (cond ((and (char<= #\0 c) (char>= #\9 c))
          (readnum))
         (t
          (ungetchar c)
          (gettoken)))
 (cond ((eq currentToken 'number)
          (gettoken)
          (if (eq currentToken 'comma)
             (gettoken))
          ))
 (cond ((eq currentToken 'while)
          (dowhilestat))
         (t
          (gettoken)    ; loop variable  (DSF 10-apr-89)
          (gettoken)    ; =
          (expression)  ; initial
          (gettoken)    ; ,
          (expression)  ; terminal
          (cond ((eq currentToken 'comma)
                   (gettoken)
                   (expression)))
          ))
 )

(defun dowhilestat ()
 (gettoken)
 (gettoken)
 (expression)
 )

(defun elsestat ()
 (gettoken)
 (if (eq currentToken 'IF)
    (elseifstat))
 )

(defun elseifstat ()
 (gettoken)      ; paren
 (gettoken)      ; first token of expression
 (expression)
 )

(defun ifstat ()
 (gettoken)      ; paren
 (gettoken)      ; first tok of expression
 (expression)
```

```
 (gettoken)
 (cond ((and (neq currentToken 'THEN)
             (neq currentToken 'number))
        (parsestat)))
 )

(defun programstat ()
 (format t "~%Parsing a program statement.")
 (gettoken)
 (setq currentModule (cadr currentToken))
 (push currentModule moduleList)
 (setq executable 1)
 (setq comments 0)
 (setq arrays nil)
 (setq calls nil)
 )

(defun blockdatastat (&aux (mod "block-data"))
 (format t "~%Parsing a block data statement.")
 (if (eq currentToken 'block) (gettoken))
 (gettoken)
 (if (consp currentToken)
   (setq mod (string-upcase
               (string-append mod "-"
                              (string (second currentToken))))))
 (setq currentToken (list 'ID (intern mod)))
 (setq currentModule (cadr currentToken))
 (push currentModule moduleList)
 (setq executable 1)
 (setq comments 0)
 (setq arrays nil)
 (setq calls nil)
 )

(defun functionstat ()
 (format t "~%Parsing a function statement.")
 (gettoken)
 (setq currentModule (cadr currentToken))
 (push currentModule moduleList)
 (setq executable 1)
 (setq comments 0)
 (setq arrays nil)
 (setq calls nil)
 )

(defun subroutinestat ()
 (format t "~%Parsing a subroutine statement.")
 (gettoken)
 (setq currentModule (cadr currentToken))
 (push currentModule moduleList)
 (setq executable 1)
 (setq comments 0)
 (setq arrays nil)
 (setq calls nil)
 )

(defun endstat ()
 (gettoken)
 (cond ((not currentToken)
        (push-end (list currentModule executable comments calls) treelist)
        (setq currentModule default-module-name)
        ))
 )
```

```
(defun typestat ()
 (gettoken)
 (cond ((eq currentToken 'FUNCTION)
          (functionstat))
         (t
          (cond ((eq currentToken 'times)
                   (gettoken)
                   (gettoken)
                   (cond ((eq currentToken 'times)  ;chaned by dsf 7-Apri-89
                       (gettoken)
                       (gettoken)))))
          (parsedatalist)
          ))
 )

(defun dimensionstat ()
 (gettoken)
 (parsedatalist)
 )

(defun callstat ()
 (gettoken)
 (if (not (member (second currentToken) calls))
    (push (second currentToken) calls))
 (gettoken)
 (cond (currentToken
          (parsearglist)
          ))
 )

(defun parsedatalist (&aux name)
 (while currentToken
          (if (member currentToken reservedwords)
             (setq currentToken (list 'id currentToken))) ;changed by dsf Apr/7/89
          (setq name (second currentToken))
          (gettoken)
;         (print name) (princ " ") (princ currenttoken)
          (cond ((eq currentToken 'times)
                   (gettoken)
                   (if (eq currentToken 'number) (getToken))))
          (cond ((eq currentToken 'lparen)
                   (if (not (member name arrays))
                      (push name arrays))
                   (parsearglist)
                   (gettoken)))
          (cond ((eq currentToken 'times)              ;chaned by dsf Apr/7/89
                   (gettoken)
                   (gettoken)))
          (gettoken)
          )
 )

; On entry, currentToken is opening parenthesis.
; At exit, currentToken is closing parenthesis.

(defun parsearglist ()
 (gettoken)
 (while (neq currentToken 'rparen)
          (expression)
          (if (eq currentToken 'comma)
             (gettoken))
          )
 )
```

```
; On entry, currentToken is first token of expression.
; At exit, currentToken is token after expression.

(defun expression (&aux name)
 (if (member currentToken '(plus minus .NOT.))
          (gettoken))
 (if (member currentToken reservedwords)
    (setq currentToken (list 'ID currentToken)))
 (cond ((consp currentToken)
          (setq name (second currentToken))
          (gettoken)
          (cond ((eq currentToken 'lparen)
                    (if (is-function name)
                       (push name calls))
                    (parsearglist)
                    (gettoken))
                    ))
         ((member currentToken '(.TRUE. .FALSE. number))
          (gettoken)
         )
         ((eq currentToken 'lparen)
          (parsearglist)
          (gettoken))
         ((member currentToken '(chars times)) (gettoken))
         ((null currenttoken) nil)
         (t (print "can't follow expression on start")
           (print currentline)
           (print currenttoken)
           (print currentindex))
         )
 (cond ((member currentToken '(number chars))
          (gettoken)))
 (cond ((eq currentToken 'lparen)        ; for "d(x)(3:3)"
          (if (is-function name)
             (push name calls))
          (parsearglist)
          (gettoken))
         )
 (cond ((member currentToken operators)
          (gettoken)
          (expression))
         ((not (member currentToken '(nil rparen comma chars)))
          (print "can't follow experssion of end")
          (print currentline)
          (print currenttoken)
          (print currentindex))))

(defun is-function (name)
 (and (not (member name arrays))
    (not (member name calls)))
 )

(defun newline (line)
 (setq currentLine line)
 (setq currentIndex 0)
 (gettoken)
 )

(defun make-node (node)
 (set-frame1 (car node) (list (car node)
                              (list 'children (cons 'value (fourth node)))
                              (list 'size (list 'value (second node)))
                              (list 'comment (list 'value (third node)))
                              ))
```

```
)

(defun make-tree (treelist)
 (mapcar 'make-node treelist)
 )

(defun set-frame1 (frame value)
 (putprop frame value 'frame))
```

File hd:nasa:knowledge acquisition system:generic-net.lisp

```lisp
;;; -*- Mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-

; this section creates and calibrates networks based on program frames
; with a set of specifications

(defun set-up-generic (&optional (plist (fget-z 'program 'specified))
                                 &aux nspec ngen)
;  (setq plist
;         (intersection (fget-z 'program 'specified)
;                       (fget-z 'program 'children)))
 (setq nspec (length (spec-vect (car plist))))
 (setq ngen  (length (get-features)))
 (if hidden-layer                   ; two or three layers
   (set-up-net (list nspec (/ nspec 2) ngen))
   (set-up-net (list nspec ngen)))
 (get-spec-samples plist))

(defun get-spec-samples (programs &aux ins outs)
 (dolist (prog programs (list ins outs))
  (push-end (spec-vect prog) ins)
  (push-end (in-vect prog)  outs)))

(defun spec-vect (prog &aux vect val)
 (dolist (criterion (fget-z 'specification 'criteria) vect)
  (dolist (aspect (fget-z 'specification criterion))
   (setq val (car (fget prog criterion aspect)))
   (dolist (resp (translate-response (list criterion aspect val)))
          (push-end resp vect)))))

(defun translate-response (specification
                                 &aux (response (third specification)))
 (if (null response) (setq response 'n/a))
 (cond ((eq response 'very-high) '(1 0 0 0 0))
       ((eq response 'high)      '(0 1 0 0 0))
       ((eq response 'medium)    '(0 0 1 0 0))
       ((eq response 'low)       '(0 0 0 1 0))
       ((eq response 'very-low)  '(0 0 0 0 1))
       ((eq response 'yes)          '(1))
       ((eq response 'probably)     '(0.75))
       ((eq response 'dont-know) (dont-know (second specification)))
       ((eq response 'maybe)        '(0.5))
       ((eq response 'probably-not)  '(0.25))
       ((eq response 'no)           '(0))
       ((eq response 'n/a) (not-applicable (second specification)))
       ((numberp response)    (list response))
       (t
        (cerror "~s is invalid" "Enter value if continued" response)
        (format t "~%Enter value :  ")
        (read))))

(defun not-applicable (aspect)
 (cond ((eq aspect 'spatial-dimension) '(0.))
       ((vector-spec aspect) '(0 0 0 0 1))
       (t '(0.))))

(defun dont-know (aspect)
 (if (vector-spec aspect)
   '(0 0 0 0 1) '(0.)))


; This routine tests the generic and sizing nets together.
```

```lisp
(defvar nonlinear-generic-net nil
  "Does the generic net have a hidden layer ?")
(defvar nonlinear-sizing-net  t
  "Does the sizing net have a hidden layer ?")

(defun test-sizing-system (&aux glist plist inst)
  (setq plist (fget-z 'program 'specified))  ; progs w. specifications
  (setq glist (fget-z 'program 'children))   ; progs w. generic components
  (setq inst (initialize-test plist))
  (set-up-net-architecture nonlinear-generic-net)
  (dolist (prog plist)
    (test-generic-net prog (set-up-generic (remove prog plist))))
  (calibrate-generic-net inst plist (set-up-generic plist))
  (set-up-net-architecture nonlinear-sizing-net)
  (dolist (prog plist)
    (test-sizing-net prog (set-up-sizing (remove prog glist))))
  (calibrate-sizing-net inst (set-up-sizing glist) plist)
  (finalize-test inst)
 )

(defun set-up-net-architecture (nonlinear-net)
  (cond (nonlinear-net
          (setq hidden-layer t)
          (setq eta .25)
          (setq alpha .9))
         (t
          (setq hidden-layer nil)
          (setq eta .005)          ; was .01
          (setq alpha .003))))     ; was .005

; Add the indicators of the sizing net's performance to the TEST-SYSTEM
; instance, after all of the network calculations have been done.

(defun finalize-test (inst &aux (plist (fget inst 'programs 'value)))
  (dolist (s '(test-sizing calibration-sizing))
   (dolist
     (f '(actual-components calibration-components test-components))
     (fput inst s f (sizing-evaluation plist s f))))
  (fput inst 'end 'value (time:print-current-time nil :|dd mmm yy|))
 )

; Determine bias and fluctuation factors from size estimated in SLOT
; and FACET of a list of frames, PLIST.

(defun sizing-evaluation (plist slot facet &aux dlist)
  (setq dlist
         (mapcar 'log (mapcar '/ (component-list plist slot facet)
                                 (component-list plist 'total-size 'value))))
  (list 'bias-factor (exp (mean dlist))
         'fluctuation-factor (exp (sigma dlist))))

; Determine a list of values in SLOT and FACET in the list of frames,
; FRAME-LIST.

(defun component-list (frame-list slot facet)
  (mapcar #'(lambda (f) (car (fget f slot facet))) frame-list))

(defvar max-iter 200  "Maximum backprop interations")
(defvar max-err .02  "Maximum error for nets")

(defun initialize-test (program-list &aux inst)
  (setq inst (instance 'test-system))
  (fput inst 'start 'value (time:print-current-time nil :|dd mmm yy|))
  (fput-list inst 'programs program-list)
```

```
 (fput inst 'learning 'eta eta)
 (fput inst 'learning 'alpha alpha)
 (fput inst 'learning 'max-iter max-iter)
 (fput inst 'learning 'max-err  max-err)
 (fput inst 'random-sizing 'value (get-random program-list))
 inst)

; Calibrate a net that has already been set up, given IO, a list whose
; first element is a list of input vectors and whose second element is
; a list of the desired output vectors.

(defun net-calibration (io &aux err iter)
 (setq iter
        (dotimes (i max-iter i)
         (learn (first io) (second io) 1)
         (if (< (setq err (vector-rms io)) max-err) (return (1+ i)))
         (format t "~%net calibration interation ~s error ~s" i err)))
 (list iter err))

; Calibrate a generic net and record its results on a specific program
; PROG, which was not used in the calibration.

(defun test-generic-net (prog io &aux cal)
 (format t "~%generic calibration for ~s" prog)
 (fput prog 'generic-calibration 'iterations
        (car (setq cal (net-calibration io))))
 (fput prog 'generic-calibration 'rms-error (second cal))
 (fput prog 'specifications 'actual (spec-vect prog))
 (fput prog 'generic-components 'actual (in-vect prog))
 (fput prog 'generic-components 'test (forward (spec-vect prog)))
 )

; Calibrate a generic net and test its results on the programs which
; were used in the calibration.

(defun calibrate-generic-net (inst glist io &aux cal)
 (format t "~%generic calibration for whole system")
 (fput inst 'net 'generic (mapcar 'length layers))
 (fput inst 'generic-calibration 'iterations
        (car (setq cal (net-calibration io))))
 (fput inst 'generic-calibration 'rms-error (second cal))
 (dolist (prog glist)
  (fput prog 'generic-components 'calibrated
         (forward (spec-vect prog)))))

; Calibrate a sizing net and record its results on a specific program
; PROG, which was not used in the calibration.

(defun test-sizing-net (prog io &aux cal vect)
 (format t "~%sizing calibration for ~s" prog)
 (fput prog 'sizing-calibration 'iterations
        (car (setq cal (net-calibration io))))
 (fput prog 'sizing-calibration 'rms-error (second cal))
 (fput prog 'test-sizing 'actual-components
        (car (revoutfun (forward (in-vect prog)))))
 (if (setq vect (car (fget prog 'generic-components 'calibrated)))
    (fput prog 'test-sizing 'calibration-components
          (car (revoutfun (forward vect)))))
 (if (setq vect (car (fget prog 'generic-components 'test)))
    (fput prog 'test-sizing 'test-components
          (car (revoutfun (forward vect)))))
 )

; Calibrate a sizing net and test its results on the programs which
```

```
; were used in the calibration.

(defun calibrate-sizing-net (inst io plist &aux cal vect)
 (format t "~%sizing calibration for whole system")
 (fput inst 'net 'sizing (mapcar 'length layers))
 (fput inst 'sizing-calibration 'iterations
         (car (setq cal (net-calibration io))))
 (fput inst 'sizing-calibration 'rms-error (second cal))
 (dolist (prog plist)
   (fput prog 'calibration-sizing 'actual-components
           (car (revoutfun (forward (in-vect prog)))))
   (if (setq vect (car (fget prog 'generic-components 'calibrated)))
         (fput prog 'calibration-sizing 'calibration-components
             (car (revoutfun (forward vect)))))
   (if (setq vect (car (fget prog 'generic-components 'test)))
         (fput prog 'calibration-sizing 'test-components
             (car (revoutfun (forward vect)))))
 ))

(defun vector-rms (io &aux (o (second io)) dif (err 0))
 (dolist (i (car io) (sqrt (/ err (length (car io)))))
   (setq dif (mapcar '- (forward i) (pop o)))
   (setq err (+ err (dot dif dif)))))


; This special test was used to calibrate the SIZING net with all programs
; except those used in the GENERIC net.

(defun special-test (&optional plist inst io cal)
 (setq plist (fget-z 'program 'specified))
 (setq inst (initialize-test plist))
 (set-up-net-architecture nonlinear-generic-net)
 (dolist (prog plist)
   (test-generic-net prog (set-up-generic (remove prog plist))))
 (calibrate-generic-net inst plist (set-up-generic plist))
 (set-up-net-architecture nonlinear-sizing-net)
 (setq io (set-up-sizing (remove-list plist (fget-z 'program 'children))))
 (fput inst 'sizing-calibration 'iterations
         (car (setq cal (net-calibration io))))
 (fput inst 'sizing-calibration 'rms-error (second cal))
 (dolist (prog plist)
   (format t "~%sizing calibration for ~s" prog)
   (fput prog 'test-sizing 'actual-components
           (car (revoutfun (forward (in-vect prog)))))
   (fput prog 'test-sizing 'calibration-components
           (car (revoutfun (forward (car (fget prog 'generic-components
                                                  'calibrated))))))
   (fput prog 'test-sizing 'test-components
           (car (revoutfun (forward (car (fget prog 'generic-components
                                                  'test))))))
 )
 (dolist
     (f '(actual-components calibration-components test-components))
     (fput inst 'test-sizing f (sizing-evaluation plist 'test-sizing f)))
 (fput inst 'end 'value (time:print-current-time nil :|dd mmm yy|))
 )

(defun remove-list (removable-items list)
 (dolist (item removable-items list)
   (setq list (remove item list))))

(defun date (&optional (format :|dd mmm yy|))
 (time:print-current-time nil format))
```

```
(defun latest-test ()
 (intern (format nil "TEST-SYSTEM~s"
                     (car (fget 'test-system 'number-of-instances 'value))) 'tb))

(defun save-results ()
 (with-open-file
  (out (concatenate 'string "test "
                     (substitute #\. #\: (date))) :direction :output)
  (pprint (show (latest-test)) out)
  (dolist (prog (fget (latest-test) 'programs 'value))
   (pprint (show prog) out))))
```

File hd:nasa:knowledge acquitition system:test-system.lisp

```
;;; -*- Mode:Common-Lisp; Package:MACTOOLBOX; Base:10 -*-

;; This part is used to restore the Sizing Net in a way so it can
;; be used with the generic net.

(defun restore-sizing-net (&optional (file "calibrated-sizing-net"))
  (with-open-file (in file)
    (mapcar 'restore-sizing-element (read in)))
  file)

(defun restore-sizing-element (element)
  (if (not (member (car element) '(eta alpha)))
     (mapcar #'(lambda (av-pair)
                        (restore-element (car element) (first av-pair) (second av-pair)))
             (cdr element))))

(defun restore-element (entity attribute value)
  (if attribute
     (setf (get (new-name entity) attribute) (new-name value))
     (set (new-name entity) (new-name value))))

(defun new-name (entity)
  (cond ((symbolp entity) (n-name entity))
          ((consp entity) (mapcar 'new-name entity))
          (t entity)))

(defun n-name (entity)
  (intern (format nil "S~a" (string entity)) 'tb))


;; This part is used to do forward propagation on the Sizing Net

(defvar slayers ()   "A list of the nodes in each layer starting at input")
(defvar snlayer  0   "The number of layers, including input and output")
(defvar sngap    0   "The number of gaps between layers")


(defun forward-size (inputs)
  (mapcar #'(lambda (n i) (setf (get n 'excite) i)) (car slayers) inputs)
  (for-size 1 inputs))

(defun for-size (r pexe &aux exe)
  (setq exe (mapcar #'(lambda (n) (set-exe n pexe (from-vals n)))
                        (nth r slayers)))
  (if (< r sngap) (for-size (1+ r) exe) exe))



;; This part tests a program's specifications


(defvar programs ())
(defvar test-file "test-system")
(defvar sizing-type nil)

(defun initialize-generic-test (&optional (sizing-file "calibrated-sizing-net"))
; (get-results test-file)
  (setq programs (fget-z (latest-test) 'test-set))
  (set-up-net-architecture nil)                        ; no hidden layer, linear
  (setq eta .002)
  (setq max-iter 350)
  (setq max-err .1)
```

```
 (restore-sizing-net sizing-file)
 (if (equal sizing-file "calibrated-sizing-net")
    (setq sizing-type 'calibration-sizing)
    (setq sizing-type 'test-sizing))
 )

(defun get-results (&optional (file test-file) &aux frame)
 (with-open-file (in file)
  (loop (setq frame (read in nil))
          (if frame (set-frame frame)
             (return-from get-results)))))

(defvar specs
          '((task engineering-program) (task database-program)
           (engineering-program structural-engineering)
           (engineering-program electronic-circuit)
           (engineering-program thermodynamics)
           (engineering-program control-systems)
           (engineering-program aerodynamics)
           (engineering-program image-processing) (thermodynamics radiation)
           (thermodynamics conduction) (methods number-redundant-choices)
           (time-dependence static) (time-dependence quasi-static)
           (time-dependence fully-dynamic) (linearity linear)
           (linearity limited-non-linear) (linearity fully-non-linear)
           (boundary number-ways-to-specify) (boundary 2d-field-on-3d-surface)
           (boundary lumped-component-values) (objects solids) (objects fluids)
           (objects non-physical) (solids structure) (solids point-mass)
           (structure spatial-dimension) (structure lumped-parameter)
           (structure continuous-volume) (structure number-of-elements)
           (structure substructuring-capability)
           (continuous-volume finite-elements) (continuous-volume grid)
           (continuous-volume number-of-known-shapes)
           (shape-limitations essentially-none) (shape-limitations thin-shells)
           (shape-limitations shells-of-revol) (shape-limitations slabs)
           (shape-limitations thick-shells) (fluids turbulence)
           (fluids shock-waves) (user-interface text-editor)
           (user-interface data-manipulator) (user-interface control-language)
           (user-interface user-specified-routines)
           (user-interface matrix-spec-language) (user-interface libraries)
           (libraries number-of-stored-shapes) (libraries format-conversion)
           (libraries geometric-conversion) (libraries translate-for-other-progs)
           (graphics present) (graphics 3d-structure) (graphics number-of-formats)
           (graphics perspective) (graphics hidden-line-removal) (graphics shading)
           (commercialization many-projects) (commercialization many-companies)
           (commercialization industry-standard) (commercialization many-upgrades)
           (commercialization many-computers)
           (lumped-parameter number-known-components)
           (database-program video-images) (non-physical images)
           (systems-programming execute-commands)
           (systems-programming code-to-save-memory)))

(defun specification-vector (prog &aux vect)
 (dolist (spec specs vect)
  (setq vect (append vect
                    (translate-response
                        (append spec (fget prog (first spec) (second spec)))))))

(defun generic-test (prog)
 (fput (latest-test) 'tested 'value prog)
 (fset prog 'generic-learning 'value
         (list 'eta eta 'max-iter max-iter 'max-err max-err))
 (generic-net-calibration prog (generic-set-up (remove prog programs)))
 )
```

```lisp
(defun final-generic-calibration (&aux (progs (fget-z (latest-test) 'test-set))
                                        io err est)
  (initialize-generic-test)
  (setq io (generic-set-up progs))
  (dotimes (i max-iter)
    (learn (first io) (second io) 1)
    (do ((p progs (cdr p)) (inp (first io) (cdr inp)) vec) ((null p))
      (setq vec (fset (car p) 'generic-components 'calibrated
                       (forward (car inp))))
      (fset (car p) 'calibration-sizing 'calibration-components
             (car (revoutfun (forward-size vec)))))
     (fset (latest-test) 'final-generic-calibration 'value
           (list 'iter (1+ i) 'err (setq err (vector-rms io))))
    (format t "~%Iteration ~4d Error ~6,3f Bias ~6,3f Fluct ~6,3f" i err
             (second (setq est (display-results progs 'calibration-sizing
                                                        'calibration-components t)))
             (fourth est))
    (if (< err max-err) (return-from final-generic-calibration)))
  )

(defun generic-set-up (prog-list &aux nspec ngen)
  (setq nspec (length (specification-vector (car prog-list))))
  (setq ngen (length (car (fget (car prog-list) 'generic-components 'actual))))
  (set-up-net (list nspec ngen))
  (list (mapcar 'specification-vector prog-list)
         (mapcar #'(lambda (p) (car (fget p 'generic-components 'actual)))
                  prog-list)))

(defun generic-net-calibration (prog io &aux err iter)
  (setq iter
         (dotimes (i max-iter i)
           (learn (first io) (second io) 1)
           (display-iteration prog i (setq err (vector-rms io)))
           (set-test-results prog i err)
           (if (< err max-err) (return (1+ i)))))
  (list iter err))

(defun set-test-results (prog iter err)
  (fset prog 'generic-calibration 'value (list 'iterations iter 'error err))
  (fset prog 'generic-components 'test (forward (specification-vector prog)))
  (fset prog sizing-type 'test-components
         (car (revoutfun
              (forward-size (car
                                  (fget prog 'generic-components 'test)))))))


;; These routines test the sizing net.



(defun initialize-sizing-test ()
; (get-results test-file)
  (setq programs (remove-list '(big-eng science-only)
                               (fget-z (latest-test) 'programs)))
  (set-up-net-architecture t)                                    ; nonlinear net.
  (setq alpha .8)
  (setq eta .2)
  (setq max-iter 300)
  (setq max-err .01)
  )

(defun sizing-test (prog)
  (fput (latest-test) 'tested 'value prog)
  (fset prog 'sizing-learning 'value
```

```
(list 'eta eta 'alpha alpha 'max-iter max-iter 'max-err max-err))
 (sizing-net-calibration prog (sizing-set-up (remove prog programs)))
 )


(defun final-sizing-calibration (&aux progs io err est)
 (setq progs (remove-list '(big-eng science-only)
                          (fget-z (latest-test) 'test-set)))
 (initialize-sizing-test)
 (setq io (sizing-set-up progs 'calibrated))
 (dotimes (i max-iter)
  (learn (first io) (second io) 1)
  (do ((vecs (first io) (cdr vecs)) (prs progs (cdr prs))) ((null vecs))
   (fset (car prs) 'calibration-sizing 'calibration-components
         (car (revoutfun (forward (car vecs))))))
  (fset (latest-test) 'final-sizing-calibration 'value
        (list 'iter (1+ i) 'err (setq err (vector-rms io))))
  (format t "~%Iteration ~4d Error ~6,3f Bias ~6,3f Fluct ~6,3f" i err
          (second (setq est (display-results progs 'calibration-sizing
                                              'calibration-components t)))
          (fourth est))
  (if (< err max-err) (return-from final-sizing-calibration)))
 )

(defun continue-sizing-test (prog &aux io prog-list)
 (setq prog-list (remove prog programs))
 (setq io
       (list (mapcar #'(lambda (p) (car (fget p 'generic-components 'test)))
                     prog-list)
             (mapcar #'(lambda (p) (outfun (fget-z p 'total-size)))
                     prog-list)))
 (sizing-net-calibration prog io))

(defun sizing-net-calibration (prog io &aux err iter)
 (setq iter
       (dotimes (i max-iter i)
        (learn (first io) (second io) 1)
        (display-sizing-iteration prog i (setq err (vector-rms io)))
        (set-sizing-results prog i err)
        (if (< err max-err) (return (1+ i)))))
 (list iter err))


(defun set-sizing-results (prog iter err)
 (fset prog 'sizing-calibration 'value (list 'iterations iter 'error err))
 (fset prog 'test-sizing 'test-components
       (car (revoutfun
             (forward (car
                       (fget prog 'generic-components 'test))))))
 (fset prog 'test-sizing 'actual-components
       (car (revoutfun
             (forward (car
                       (fget prog 'generic-components 'actual)))))))


(defun sizing-set-up (prog-list &optional (type 'actual) &aux ngen)
 (setq ngen (length (car (fget (car prog-list)
                               'generic-components type))))
 (if hidden-layer
   (set-up-net (list ngen (/ ngen 2) output-nodes))
   (set-up-net (list (length net-features) output-nodes)))
 (list (mapcar #'(lambda (p) (car (fget p 'generic-components 'test)))
               prog-list)
       (mapcar #'(lambda (p) (outfun (fget-z p 'total-size)))
```

```
                    prog-list)))

(defun overnight-sizing-test ()
 (setq max-iter 200)
 (setq max-err  .05)
 (dolist (prog (remove-list (fget-z (latest-test) 'tested)
                      (fget-z (latest-test) 'test-set)))
   (sizing-test prog)))

;; These routines are used to define, store, and display results

(defvar format-string
        "~%~8s iter ~4d  err ~9,5f  size: act ~8,4f  pred ~8,4f")

(defun display-iteration (prog iteration err)
 (format t format-string
          prog iteration err (car (fget-z prog 'total-size))
          (car (revoutfun (forward-size (forward (specification-vector prog)))))))


(defun display-sizing-iteration (prog iteration err)
 (format t format-string
          prog iteration err (car (fget-z prog 'total-size))
          (car (revoutfun (forward (car (fget prog 'generic-components 'test)))))))

(defun display-results (&optional plist
                                  (slot 'test-sizing) (facet 'test-components) no-print
                                  &aux set)
  (if (null plist)
     (setq plist (sizing-sort (fget-z (latest-test) 'test-set) slot facet)))
  (setq set (mapcar 'log (view-res plist slot facet no-print)))
  (if (null no-print)
    (format t "~%~%  Bias factor = ~5,2f       Fluctuation factor = ~5,2f"
           (exp (mean set)) (exp (sigma set)))
    (list 'bias (exp (mean set)) 'fluctuation (exp (sigma set)))))

(defun view-res (proglist &optional  (slot 'test-sizing)
                      (facet 'actual-components) no-print &aux size pred res)
  (if (null no-print)
    (format t "~%~20s ~10s ~10s ~10s~%" 'program 'actual 'predicted 'p/a))
  (dolist (prog proglist res)
   (setq size (car (fget-z prog 'total-size)))
   (setq pred (car (fget prog slot facet)))
   (if (null no-print)
         (format t "~%~20s ~10s ~10s ~10s" prog size  pred (/ pred size)))
   (push-end (/ pred size) res)))

(defun ratio (p &optional (slot 'calibration-sizing) (facet 'test-components))
  (/ (car (fget p slot facet))
    (car (fget-z p 'total-size))))

(defun sizing-sort (&optional (plist (fget-z (latest-test) 'test-set))
                          (slot 'calibration-sizing) (facet 'test-components))
  (sort (copy-tree plist)
        #'(lambda (a b) (> (abs (log (ratio a slot facet)))
                              (abs (log (ratio b slot facet)))))))


(defun save-current-results ()
  (with-open-file
    (out test-file :direction :output :if-exists :overwrite)
    (pprint (show 'test-system) out)
    (pprint (show (latest-test)) out)
    (dolist (prog (fget (latest-test) 'programs 'value))
```

```
(pprint (show prog) out))))
```

Appendix 2 - Generic Rule Base

This appendix contains the generic rule base organized by generic component

1. Rules for TASK COMPLEXITY
2. Rules for METHODS COMPLEXITY
3. Rules for METHODS GENERALITY
4. Rules for METHODS ACCURACY
5. Rules for METHODS EFFICIENCY
6. Rules for METHODS REDUNDANCY
7. Rules for OBJECTS COMPLEXITY
8. Rules for OBJECTS GENERALITY
9. Rules for REPRESENTATION COMPLEXITY
10. Rules for REPRESENTATION CAPACITY
11. Rules for REPRESENTATION KNOWLEDGE
12. Rules for GRAPHICS COMPLEXITY
13. Rules for USER-INTERFACE COMPLEXITY
14. Rules for DATA-MANAGEMENT COMPLEXITY
15. Rules for SYS-PROGRAMMING COMPLEXITY
16. Rules for HARD-ARCHITECTURE COMPLEXITY

Rules for TASK COMPLEXITY

If the TASK COMPLEXITY is being determined,
Then initialize the value of the TASK COMPLEXITY to 0.079200014
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the TASK COMPLEXITY by 0.15677664
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the TASK COMPLEXITY by 0.15781765
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the TASK COMPLEXITY by 0.019139774
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the TASK COMPLEXITY by 0.042799994
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the TASK COMPLEXITY by 0.058112793
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the TASK COMPLEXITY by 0.13921365
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the TASK COMPLEXITY by -0.07263435
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the TASK COMPLEXITY by 0.06377641
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the TASK COMPLEXITY by -0.23982486
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the TASK COMPLEXITY by 0.027954046
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the TASK COMPLEXITY by 0.14098752
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the TASK COMPLEXITY by 0.011199996
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the TASK COMPLEXITY by -0.13277595
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the TASK COMPLEXITY by -0.047721017
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the TASK COMPLEXITY by -0.08058988
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the TASK COMPLEXITY by -0.024401616
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the TASK COMPLEXITY by -0.07677092
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the TASK COMPLEXITY by 0.0033372478
If the value of LINEARITY LINEAR is YES,
Then increment the TASK COMPLEXITY by -0.061736364
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the TASK COMPLEXITY by 0.105950244
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the TASK COMPLEXITY by -0.03931454
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the TASK COMPLEXITY by 0.14740701
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the TASK COMPLEXITY by -0.0868
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the TASK COMPLEXITY by 0.1928287
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the TASK COMPLEXITY by -0.037133746
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the TASK COMPLEXITY by -0.035601236
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the TASK COMPLEXITY by 0.12747107
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the TASK COMPLEXITY by -0.1020541

If the value of OBJECTS SOLIDS is YES,
Then increment the TASK COMPLEXITY by -0.09371934
If the value of OBJECTS FLUIDS is YES,
Then increment the TASK COMPLEXITY by 0.17970333
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the TASK COMPLEXITY by 0.1668071
If the value of SOLIDS STRUCTURE is YES,
Then increment the TASK COMPLEXITY by -0.08811934
If the value of SOLIDS POINT-MASS is YES,
Then increment the TASK COMPLEXITY by -0.051429614
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the TASK COMPLEXITY by 0.120865755 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the TASK COMPLEXITY by -0.01242457
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the TASK COMPLEXITY by -0.015260225
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the TASK COMPLEXITY by -0.030325852
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the TASK COMPLEXITY by 0.029456824
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the TASK COMPLEXITY by 0.18135875
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the TASK COMPLEXITY by 0.04979097
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the TASK COMPLEXITY by 0.021620085
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the TASK COMPLEXITY by 0.0060500293
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the TASK COMPLEXITY by 0.026494816
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the TASK COMPLEXITY by 0.2353526
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the TASK COMPLEXITY by 0.05121107
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the TASK COMPLEXITY by -0.04873027
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the TASK COMPLEXITY by -0.12382611
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the TASK COMPLEXITY by 0.13640574
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the TASK COMPLEXITY by -0.06475973
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the TASK COMPLEXITY by -0.065857574
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the TASK COMPLEXITY by 0.013648725
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the TASK COMPLEXITY by -0.13156763
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the TASK COMPLEXITY by -0.013161882
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the TASK COMPLEXITY by 0.0104323365
If the value of FLUIDS TURBULENCE is YES,
Then increment the TASK COMPLEXITY by 0.00576568
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the TASK COMPLEXITY by 0.15635519
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the TASK COMPLEXITY by 0.14335294
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the TASK COMPLEXITY by 0.16736823
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the TASK COMPLEXITY by -0.102388844
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the TASK COMPLEXITY by -0.0132333245

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the TASK COMPLEXITY by 0.094693676
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the TASK COMPLEXITY by -0.032224815
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the TASK COMPLEXITY by -0.106047876
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the TASK COMPLEXITY by 0.17377642
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the TASK COMPLEXITY by -0.0071853255
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the TASK COMPLEXITY by 0.039833598
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the TASK COMPLEXITY by 0.01952392
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the TASK COMPLEXITY by 0.07133655
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the TASK COMPLEXITY by -0.124211825
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the TASK COMPLEXITY by 0.015730578
If the value of GRAPHICS PRESENT is YES,
Then increment the TASK COMPLEXITY by 0.0311605
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the TASK COMPLEXITY by -0.04287205
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the TASK COMPLEXITY by -0.18548447
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the TASK COMPLEXITY by -0.1024
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the TASK COMPLEXITY by 0.0015999973
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the TASK COMPLEXITY by -0.12234582
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the TASK COMPLEXITY by -0.11946896
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the TASK COMPLEXITY by 0.07888417
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the TASK COMPLEXITY by -0.019601973
If the value of GRAPHICS SHADING is YES,
Then increment the TASK COMPLEXITY by 0.02678696
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the TASK COMPLEXITY by 0.08515973
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the TASK COMPLEXITY by 0.037021447
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the TASK COMPLEXITY by -0.01137741
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the TASK COMPLEXITY by 0.010579169
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the TASK COMPLEXITY by 0.12892595
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the TASK COMPLEXITY by 0.16705687
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the TASK COMPLEXITY by -0.12
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the TASK COMPLEXITY by 0.08038825
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the TASK COMPLEXITY by 0.1955552
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the TASK COMPLEXITY by 0.06450044
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the TASK COMPLEXITY by 0.036917474
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the TASK COMPLEXITY by -0.15429556

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the TASK COMPLEXITY by 0.05912253
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the TASK COMPLEXITY by -0.070472054

Rules for METHODS COMPLEXITY

If the METHODS COMPLEXITY is being determined,
Then initialize the value of the METHODS COMPLEXITY to -0.1388
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the METHODS COMPLEXITY by -0.028632497
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the METHODS COMPLEXITY by -0.07830586
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the METHODS COMPLEXITY by 0.08072675
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the METHODS COMPLEXITY by -0.1996
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the METHODS COMPLEXITY by 0.09706881
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the METHODS COMPLEXITY by 0.030445073
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the METHODS COMPLEXITY by -0.1532452
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the METHODS COMPLEXITY by 0.08547639
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the METHODS COMPLEXITY by -0.1706279
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the METHODS COMPLEXITY by 0.054566722
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the METHODS COMPLEXITY by -0.0051239072
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the METHODS COMPLEXITY by 0.19000001
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the METHODS COMPLEXITY by -0.10803329
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the METHODS COMPLEXITY by -0.06127314
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the METHODS COMPLEXITY by 0.051157963
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the METHODS COMPLEXITY by 0.077936254
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the METHODS COMPLEXITY by 0.18425997
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the METHODS COMPLEXITY by 0.05408548
If the value of LINEARITY LINEAR is YES,
Then increment the METHODS COMPLEXITY by -0.058386214
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the METHODS COMPLEXITY by -0.16382642
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the METHODS COMPLEXITY by -0.06889983
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the METHODS COMPLEXITY by -0.13217136
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the METHODS COMPLEXITY by 0.08440001
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the METHODS COMPLEXITY by -0.117156096
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the METHODS COMPLEXITY by -0.09296545
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the METHODS COMPLEXITY by 0.16582043
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the METHODS COMPLEXITY by 0.09346862
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the METHODS COMPLEXITY by 0.025555577

If the value of OBJECTS SOLIDS is YES,
Then increment the METHODS COMPLEXITY by 0.12141921
If the value of OBJECTS FLUIDS is YES,
Then increment the METHODS COMPLEXITY by -0.009548504
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the METHODS COMPLEXITY by -0.10988109
If the value of SOLIDS STRUCTURE is YES,
Then increment the METHODS COMPLEXITY by 0.10981921
If the value of SOLIDS POINT-MASS is YES,
Then increment the METHODS COMPLEXITY by 0.19901414
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the METHODS COMPLEXITY by 0.039295632 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the METHODS COMPLEXITY by 0.0429597
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the METHODS COMPLEXITY by 0.25432676
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the METHODS COMPLEXITY by -0.08830488
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the METHODS COMPLEXITY by -0.1444475
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the METHODS COMPLEXITY by -0.125098
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the METHODS COMPLEXITY by -0.1699306
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the METHODS COMPLEXITY by -0.08389172
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the METHODS COMPLEXITY by 0.044191316
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the METHODS COMPLEXITY by -0.079852015
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the METHODS COMPLEXITY by 0.11916547
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the METHODS COMPLEXITY by -0.077800274
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the METHODS COMPLEXITY by 0.107544325
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the METHODS COMPLEXITY by 0.005254544
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the METHODS COMPLEXITY by 0.09910422
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the METHODS COMPLEXITY by 0.19382475
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the METHODS COMPLEXITY by 0.033077274
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the METHODS COMPLEXITY by 0.032758866
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the METHODS COMPLEXITY by -0.07771117
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the METHODS COMPLEXITY by 0.113793015
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the METHODS COMPLEXITY by -0.028111165
If the value of FLUIDS TURBULENCE is YES,
Then increment the METHODS COMPLEXITY by 0.11395475
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the METHODS COMPLEXITY by 0.056043297
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the METHODS COMPLEXITY by 0.17489976
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the METHODS COMPLEXITY by 0.06185607
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the METHODS COMPLEXITY by 0.07541698
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the METHODS COMPLEXITY by -0.04219431

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the METHODS COMPLEXITY by 0.19685344
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the METHODS COMPLEXITY by -0.048570663
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the METHODS COMPLEXITY by -0.05513018
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the METHODS COMPLEXITY by 0.17107634
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the METHODS COMPLEXITY by -0.11492747
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the METHODS COMPLEXITY by 0.044188038
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the METHODS COMPLEXITY by -0.06727925
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the METHODS COMPLEXITY by 0.0073317504
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the METHODS COMPLEXITY by 0.06369716
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the METHODS COMPLEXITY by -0.09920703
If the value of GRAPHICS PRESENT is YES,
Then increment the METHODS COMPLEXITY by -0.046199102
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the METHODS COMPLEXITY by -0.014345068
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the METHODS COMPLEXITY by -0.06564317
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the METHODS COMPLEXITY by 0.0748
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the METHODS COMPLEXITY by -0.1948
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the METHODS COMPLEXITY by -0.15065828
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the METHODS COMPLEXITY by 0.076629005
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the METHODS COMPLEXITY by -0.15897526
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the METHODS COMPLEXITY by -0.08458962
If the value of GRAPHICS SHADING is YES,
Then increment the METHODS COMPLEXITY by 0.033210654
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the METHODS COMPLEXITY by -0.042342562
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the METHODS COMPLEXITY by 0.24450356
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the METHODS COMPLEXITY by -0.061316688
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the METHODS COMPLEXITY by 0.1168513
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the METHODS COMPLEXITY by 0.11430409
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the METHODS COMPLEXITY by -0.11484734
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the METHODS COMPLEXITY by 0.039599985
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the METHODS COMPLEXITY by -0.068408296
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the METHODS COMPLEXITY by -0.21275665
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the METHODS COMPLEXITY by -0.05646009
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the METHODS COMPLEXITY by 0.11854644
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the METHODS COMPLEXITY by -0.078642905

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the METHODS COMPLEXITY by 0.15579277
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the METHODS COMPLEXITY by -0.11634206

Rules for METHODS GENERALITY

If the METHODS GENERALITY is being determined,
Then initialize the value of the METHODS GENERALITY to 0.0888
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the METHODS GENERALITY by -0.005989452
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the METHODS GENERALITY by 0.19630213
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the METHODS GENERALITY by -0.113902844
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the METHODS GENERALITY by -0.13080001
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the METHODS GENERALITY by 0.033416618
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the METHODS GENERALITY by -0.041496042
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the METHODS GENERALITY by -0.14834066
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the METHODS GENERALITY by -0.09401288
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the METHODS GENERALITY by 0.14131111
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the METHODS GENERALITY by -0.08151603
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the METHODS GENERALITY by -0.0011838083
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the METHODS GENERALITY by 0.0944
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the METHODS GENERALITY by -0.0016423305
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the METHODS GENERALITY by -0.096568964
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the METHODS GENERALITY by 0.04028693
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the METHODS GENERALITY by 0.016413253
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the METHODS GENERALITY by 0.1359455
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the METHODS GENERALITY by -0.118272044
If the value of LINEARITY LINEAR is YES,
Then increment the METHODS GENERALITY by -0.040885855
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the METHODS GENERALITY by -0.10534706
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the METHODS GENERALITY by 0.20289022
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the METHODS GENERALITY by 0.07296366
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the METHODS GENERALITY by -0.022799999
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the METHODS GENERALITY by 0.09169702
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the METHODS GENERALITY by -2.1243904e-4
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the METHODS GENERALITY by 0.019643523
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the METHODS GENERALITY by 0.07101585
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the METHODS GENERALITY by -0.0907971

If the value of OBJECTS SOLIDS is YES,
Then increment the METHODS GENERALITY by 0.10552213
If the value of OBJECTS FLUIDS is YES,
Then increment the METHODS GENERALITY by 0.17016484
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the METHODS GENERALITY by -0.06800818
If the value of SOLIDS STRUCTURE is YES,
Then increment the METHODS GENERALITY by -0.012477847
If the value of SOLIDS POINT-MASS is YES,
Then increment the METHODS GENERALITY by 0.04162043
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the METHODS GENERALITY by -0.09920261 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the METHODS GENERALITY by 0.057116326
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the METHODS GENERALITY by 0.15929717
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the METHODS GENERALITY by -0.09436259
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the METHODS GENERALITY by -0.12324172
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the METHODS GENERALITY by 0.011332675
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the METHODS GENERALITY by -0.030606182
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the METHODS GENERALITY by -0.087030336
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the METHODS GENERALITY by -0.10060151
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the METHODS GENERALITY by 0.13527253
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the METHODS GENERALITY by 0.14548151
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the METHODS GENERALITY by 0.04442904
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the METHODS GENERALITY by 0.0829182
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the METHODS GENERALITY by 0.018090134
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the METHODS GENERALITY by -0.19890563
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the METHODS GENERALITY by -0.03084001
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the METHODS GENERALITY by 0.13338448
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the METHODS GENERALITY by 0.016616052
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the METHODS GENERALITY by 0.15623766
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the METHODS GENERALITY by 0.14653209
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the METHODS GENERALITY by -0.06816234
If the value of FLUIDS TURBULENCE is YES,
Then increment the METHODS GENERALITY by -0.09474069
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the METHODS GENERALITY by 0.18892597
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the METHODS GENERALITY by 0.057116974
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the METHODS GENERALITY by 0.09492691
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the METHODS GENERALITY by 0.11259196
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the METHODS GENERALITY by 0.14128989

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the METHODS GENERALITY by -0.10504371
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the METHODS GENERALITY by -0.072007746
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the METHODS GENERALITY by -0.085434794
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the METHODS GENERALITY by -0.073612876
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the METHODS GENERALITY by -0.17041384
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the METHODS GENERALITY by -0.06367839
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the METHODS GENERALITY by -0.052368395
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the METHODS GENERALITY by -0.15220678
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the METHODS GENERALITY by -0.09129375
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the METHODS GENERALITY by 0.12006205
If the value of GRAPHICS PRESENT is YES,
Then increment the METHODS GENERALITY by 0.13865939
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the METHODS GENERALITY by -0.095259696
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the METHODS GENERALITY by 0.13397463
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the METHODS GENERALITY by -0.056799993
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the METHODS GENERALITY by 0.03199999
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the METHODS GENERALITY by 0.037125748
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the METHODS GENERALITY by 0.061791446
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the METHODS GENERALITY by 0.15634784
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the METHODS GENERALITY by -0.13654874
If the value of GRAPHICS SHADING is YES,
Then increment the METHODS GENERALITY by -0.11097779
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the METHODS GENERALITY by 0.1895556
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the METHODS GENERALITY by 0.16852641
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the METHODS GENERALITY by 0.15385455
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the METHODS GENERALITY by 0.1706947
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the METHODS GENERALITY by -0.03324728
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the METHODS GENERALITY by -0.0036417688
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the METHODS GENERALITY by 0.0996
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the METHODS GENERALITY by 0.08972102
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the METHODS GENERALITY by -0.13467404
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the METHODS GENERALITY by -0.06251345
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the METHODS GENERALITY by 0.16052331
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the METHODS GENERALITY by 0.07634551

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the METHODS GENERALITY by 0.18354905
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the METHODS GENERALITY by -0.22274339

Rules for METHODS ACCURACY

If the METHODS ACCURACY is being determined,
Then initialize the value of the METHODS ACCURACY to -0.1688
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the METHODS ACCURACY by 0.07603234
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the METHODS ACCURACY by 0.12241937
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the METHODS ACCURACY by 0.052200273
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the METHODS ACCURACY by 0.11080001
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the METHODS ACCURACY by -0.05048059
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the METHODS ACCURACY by 0.018817648
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the METHODS ACCURACY by 0.1600446
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the METHODS ACCURACY by 0.112680875
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the METHODS ACCURACY by -0.039222192
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the METHODS ACCURACY by -0.008881149
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the METHODS ACCURACY by -0.123594
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the METHODS ACCURACY by -0.06799999
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the METHODS ACCURACY by -0.10267133
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the METHODS ACCURACY by 0.14690791
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the METHODS ACCURACY by 0.034445126
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the METHODS ACCURACY by -0.08690371
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the METHODS ACCURACY by 0.2227582
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the METHODS ACCURACY by 0.23380016
If the value of LINEARITY LINEAR is YES,
Then increment the METHODS ACCURACY by -0.11818694
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the METHODS ACCURACY by 0.06090113
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the METHODS ACCURACY by -0.022094158
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the METHODS ACCURACY by 0.013670428
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the METHODS ACCURACY by 0.039599985
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the METHODS ACCURACY by -0.15387535
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the METHODS ACCURACY by 0.04443509
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the METHODS ACCURACY by 0.08445763
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the METHODS ACCURACY by 0.12698595
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the METHODS ACCURACY by -0.080665894

If the value of OBJECTS SOLIDS is YES,
Then increment the METHODS ACCURACY by -0.05947474
If the value of OBJECTS FLUIDS is YES,
Then increment the METHODS ACCURACY by 0.20398618
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the METHODS ACCURACY by 0.15020232
If the value of SOLIDS STRUCTURE is YES,
Then increment the METHODS ACCURACY by -0.09587473
If the value of SOLIDS POINT-MASS is YES,
Then increment the METHODS ACCURACY by 0.09835588
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the METHODS ACCURACY by 0.17637646 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the METHODS ACCURACY by 0.041296262
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the METHODS ACCURACY by -0.17899972
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the METHODS ACCURACY by -0.12233887
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the METHODS ACCURACY by 0.13093439
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the METHODS ACCURACY by -0.13519949
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the METHODS ACCURACY by 0.009129278
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the METHODS ACCURACY by 0.11256244
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the METHODS ACCURACY by 0.18433678
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the METHODS ACCURACY by -0.101839475
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the METHODS ACCURACY by 0.070884615
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the METHODS ACCURACY by 0.14332508
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the METHODS ACCURACY by -0.13911773
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the METHODS ACCURACY by -0.17319864
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the METHODS ACCURACY by 0.06279647
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the METHODS ACCURACY by -0.02951745
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the METHODS ACCURACY by -0.051895153
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the METHODS ACCURACY by -0.0603809
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the METHODS ACCURACY by 0.05884185
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the METHODS ACCURACY by -0.08956168
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the METHODS ACCURACY by 0.056841858
If the value of FLUIDS TURBULENCE is YES,
Then increment the METHODS ACCURACY by 0.04444464
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the METHODS ACCURACY by -0.13702382
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the METHODS ACCURACY by 0.023529653
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the METHODS ACCURACY by 0.034130026
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the METHODS ACCURACY by -0.101721056
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the METHODS ACCURACY by 0.017501237

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the METHODS ACCURACY by -0.14989397
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the METHODS ACCURACY by 0.05444576
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the METHODS ACCURACY by 0.0055080005
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the METHODS ACCURACY by -0.17291915
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the METHODS ACCURACY by -0.0563705
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the METHODS ACCURACY by 0.15143934
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the METHODS ACCURACY by 0.13022995
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the METHODS ACCURACY by 0.10419422
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the METHODS ACCURACY by -0.10615036
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the METHODS ACCURACY by -0.07047724
If the value of GRAPHICS PRESENT is YES,
Then increment the METHODS ACCURACY by 0.11480762
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the METHODS ACCURACY by -0.006183359
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the METHODS ACCURACY by -0.12737112
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the METHODS ACCURACY by 0.002000004
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the METHODS ACCURACY by -0.114
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the METHODS ACCURACY by 0.05666055
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the METHODS ACCURACY by 0.0025982661
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the METHODS ACCURACY by 0.09222456
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the METHODS ACCURACY by -0.011434924
If the value of GRAPHICS SHADING is YES,
Then increment the METHODS ACCURACY by -0.11676
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the METHODS ACCURACY by 0.22130479
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the METHODS ACCURACY by -0.060107388
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the METHODS ACCURACY by 0.25500524
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the METHODS ACCURACY by -0.017417356
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the METHODS ACCURACY by -0.06108398
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the METHODS ACCURACY by 0.0597343
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the METHODS ACCURACY by -0.049600005
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the METHODS ACCURACY by 0.19180804
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the METHODS ACCURACY by -0.0958239
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the METHODS ACCURACY by 0.045769222
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the METHODS ACCURACY by 0.076463796
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the METHODS ACCURACY by -0.0086962255

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the METHODS ACCURACY by 0.10494009
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the METHODS ACCURACY by -0.077667736

Rules for METHODS EFFICIENCY

If the METHODS EFFICIENCY is being determined,
Then initialize the value of the METHODS EFFICIENCY to 0.10759999
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the METHODS EFFICIENCY by -0.06683519
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the METHODS EFFICIENCY by 0.016145082
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the METHODS EFFICIENCY by -0.05050206
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the METHODS EFFICIENCY by 0.0012000054
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the METHODS EFFICIENCY by -0.14712231
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the METHODS EFFICIENCY by -0.02432915
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the METHODS EFFICIENCY by 0.11588017
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the METHODS EFFICIENCY by -0.09693154
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the METHODS EFFICIENCY by -0.0012443416
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the METHODS EFFICIENCY by 0.10605081
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the METHODS EFFICIENCY by 0.1320479
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the METHODS EFFICIENCY by -0.075600006
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the METHODS EFFICIENCY by 0.039588567
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the METHODS EFFICIENCY by -0.09849431
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the METHODS EFFICIENCY by -0.06747445
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the METHODS EFFICIENCY by -0.15473932
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the METHODS EFFICIENCY by 0.1523343
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the METHODS EFFICIENCY by 0.14258836
If the value of LINEARITY LINEAR is YES,
Then increment the METHODS EFFICIENCY by 0.04253627
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the METHODS EFFICIENCY by -0.008571226
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the METHODS EFFICIENCY by 0.22032627
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the METHODS EFFICIENCY by -0.015949003
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the METHODS EFFICIENCY by -0.1412
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the METHODS EFFICIENCY by 0.11219448
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the METHODS EFFICIENCY by 0.02600634
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the METHODS EFFICIENCY by -0.14098419
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the METHODS EFFICIENCY by 0.1471577
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the METHODS EFFICIENCY by -4.1327953e-4

If the value of OBJECTS SOLIDS is YES,
Then increment the METHODS EFFICIENCY by 0.0059080753
If the value of OBJECTS FLUIDS is YES,
Then increment the METHODS EFFICIENCY by -0.09839785
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the METHODS EFFICIENCY by 0.1369036
If the value of SOLIDS STRUCTURE is YES,
Then increment the METHODS EFFICIENCY by -0.036891922
If the value of SOLIDS POINT-MASS is YES,
Then increment the METHODS EFFICIENCY by 0.031696748
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the METHODS EFFICIENCY by -0.06939353 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the METHODS EFFICIENCY by 0.13746154
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the METHODS EFFICIENCY by 0.023897922
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the METHODS EFFICIENCY by 0.10145831
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the METHODS EFFICIENCY by -0.10182956
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the METHODS EFFICIENCY by -0.07677313
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the METHODS EFFICIENCY by 0.019452458
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the METHODS EFFICIENCY by -0.025040355
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the METHODS EFFICIENCY by 0.1877441
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the METHODS EFFICIENCY by 0.11475524
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the METHODS EFFICIENCY by 0.13542405
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the METHODS EFFICIENCY by -0.1474206
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the METHODS EFFICIENCY by -0.14822783
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the METHODS EFFICIENCY by -0.01992266
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the METHODS EFFICIENCY by -0.026193496
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the METHODS EFFICIENCY by 0.1794322
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the METHODS EFFICIENCY by -0.033140313
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the METHODS EFFICIENCY by -0.02803845
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the METHODS EFFICIENCY by -0.09636725
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the METHODS EFFICIENCY by -0.12936069
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the METHODS EFFICIENCY by 0.11163273
If the value of FLUIDS TURBULENCE is YES,
Then increment the METHODS EFFICIENCY by -0.08611985
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the METHODS EFFICIENCY by -0.09221114
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the METHODS EFFICIENCY by -0.055205073
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the METHODS EFFICIENCY by 0.008509341
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the METHODS EFFICIENCY by 0.15268995
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the METHODS EFFICIENCY by -0.08350555

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the METHODS EFFICIENCY by 0.094205104
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the METHODS EFFICIENCY by 0.079101875
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the METHODS EFFICIENCY by 0.19257195
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the METHODS EFFICIENCY by 0.16786854
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the METHODS EFFICIENCY by 0.16407116
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the METHODS EFFICIENCY by -0.10234634
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the METHODS EFFICIENCY by 0.12230244
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the METHODS EFFICIENCY by -0.07463047
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the METHODS EFFICIENCY by 0.03752726
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the METHODS EFFICIENCY by -0.10523188
If the value of GRAPHICS PRESENT is YES,
Then increment the METHODS EFFICIENCY by 0.01654593
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the METHODS EFFICIENCY by 0.10852012
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the METHODS EFFICIENCY by 0.004384271
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the METHODS EFFICIENCY by 0.036400005
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the METHODS EFFICIENCY by -0.1852
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the METHODS EFFICIENCY by -0.018812837
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the METHODS EFFICIENCY by 0.06609629
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the METHODS EFFICIENCY by -0.14577577
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the METHODS EFFICIENCY by 0.05852331
If the value of GRAPHICS SHADING is YES,
Then increment the METHODS EFFICIENCY by -0.048456125
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the METHODS EFFICIENCY by 0.09687524
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the METHODS EFFICIENCY by -0.11879473
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the METHODS EFFICIENCY by 0.0706483
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the METHODS EFFICIENCY by -0.03063732
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the METHODS EFFICIENCY by 0.041628294
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the METHODS EFFICIENCY by -0.007429591
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the METHODS EFFICIENCY by -0.18720001
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the METHODS EFFICIENCY by -0.15245347
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the METHODS EFFICIENCY by -0.10261114
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the METHODS EFFICIENCY by 0.10176187
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the METHODS EFFICIENCY by 0.07106094
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the METHODS EFFICIENCY by -0.13939518

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the METHODS EFFICIENCY by 0.071497686
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the METHODS EFFICIENCY by 0.14675224

Rules for METHODS REDUNDANCY

If the METHODS REDUNDANCY is being determined,
Then initialize the value of the METHODS REDUNDANCY to -0.0572
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the METHODS REDUNDANCY by 0.21382213
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the METHODS REDUNDANCY by -0.07569032
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the METHODS REDUNDANCY by -0.00720632
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the METHODS REDUNDANCY by 0.0852
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the METHODS REDUNDANCY by -0.050014555
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the METHODS REDUNDANCY by -0.070801884
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the METHODS REDUNDANCY by -0.17729628
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the METHODS REDUNDANCY by 0.15028837
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the METHODS REDUNDANCY by 0.061477546
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the METHODS REDUNDANCY by -0.010745033
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the METHODS REDUNDANCY by 0.016345214
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the METHODS REDUNDANCY by -0.0119999945
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the METHODS REDUNDANCY by -0.07266162
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the METHODS REDUNDANCY by -0.15090182
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the METHODS REDUNDANCY by -0.03989061
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the METHODS REDUNDANCY by 0.013767125
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the METHODS REDUNDANCY by -0.12759677
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the METHODS REDUNDANCY by 0.053868454
If the value of LINEARITY LINEAR is YES,
Then increment the METHODS REDUNDANCY by 0.15399544
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the METHODS REDUNDANCY by 0.091097765
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the METHODS REDUNDANCY by -0.06161439
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the METHODS REDUNDANCY by 0.08698144
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the METHODS REDUNDANCY by 0.186
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the METHODS REDUNDANCY by 0.25323558
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the METHODS REDUNDANCY by 0.09574672
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the METHODS REDUNDANCY by -6.7262897e-4
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the METHODS REDUNDANCY by 0.08406843
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the METHODS REDUNDANCY by -0.017407551

If the value of OBJECTS SOLIDS is YES,
Then increment the METHODS REDUNDANCY by -0.16039771
If the value of OBJECTS FLUIDS is YES,
Then increment the METHODS REDUNDANCY by -0.0015882566
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the METHODS REDUNDANCY by -0.061357427
If the value of SOLIDS STRUCTURE is YES,
Then increment the METHODS REDUNDANCY by 0.109202266
If the value of SOLIDS POINT-MASS is YES,
Then increment the METHODS REDUNDANCY by -0.07896923
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the METHODS REDUNDANCY by -0.07248796 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the METHODS REDUNDANCY by -0.07609992
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the METHODS REDUNDANCY by 0.104793735
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the METHODS REDUNDANCY by -0.083680294
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the METHODS REDUNDANCY by 0.05167184
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the METHODS REDUNDANCY by -0.06686951
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the METHODS REDUNDANCY by -0.17551972
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the METHODS REDUNDANCY by -0.111511156
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the METHODS REDUNDANCY by 0.1587955
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the METHODS REDUNDANCY by 0.020896912
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the METHODS REDUNDANCY by 0.03685752
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the METHODS REDUNDANCY by 0.098456845
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the METHODS REDUNDANCY by 0.1647793
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the METHODS REDUNDANCY by 0.0696853
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the METHODS REDUNDANCY by 0.14387928
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the METHODS REDUNDANCY by 0.18569037
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the METHODS REDUNDANCY by 0.17815286
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the METHODS REDUNDANCY by -0.022768334
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the METHODS REDUNDANCY by 0.15688466
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the METHODS REDUNDANCY by -0.17923614
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the METHODS REDUNDANCY by -0.06471533
If the value of FLUIDS TURBULENCE is YES,
Then increment the METHODS REDUNDANCY by 0.06310382
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the METHODS REDUNDANCY by -0.12410146
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the METHODS REDUNDANCY by -0.097921364
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the METHODS REDUNDANCY by 0.028293092
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the METHODS REDUNDANCY by 0.13717386
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the METHODS REDUNDANCY by -0.05602517

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the METHODS REDUNDANCY by 0.061290156
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the METHODS REDUNDANCY by 0.028134186
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the METHODS REDUNDANCY by 0.15520096
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the METHODS REDUNDANCY by -0.12291157
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the METHODS REDUNDANCY by -0.17003553
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the METHODS REDUNDANCY by -0.24310055
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the METHODS REDUNDANCY by 0.054137986
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the METHODS REDUNDANCY by 0.1542234
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the METHODS REDUNDANCY by 0.20134339
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the METHODS REDUNDANCY by 0.1306556
If the value of GRAPHICS PRESENT is YES,
Then increment the METHODS REDUNDANCY by 0.017665166
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the METHODS REDUNDANCY by -0.073260404
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the METHODS REDUNDANCY by 0.019643009
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the METHODS REDUNDANCY by 0.042400002
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the METHODS REDUNDANCY by 0.066
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the METHODS REDUNDANCY by -0.20059201
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the METHODS REDUNDANCY by -0.024159877
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the METHODS REDUNDANCY by -0.003863127
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the METHODS REDUNDANCY by -0.122189485
If the value of GRAPHICS SHADING is YES,
Then increment the METHODS REDUNDANCY by -0.12144625
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the METHODS REDUNDANCY by -0.009052943
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the METHODS REDUNDANCY by -0.060701884
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the METHODS REDUNDANCY by 0.03458374
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the METHODS REDUNDANCY by 0.14307463
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the METHODS REDUNDANCY by -0.11625518
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the METHODS REDUNDANCY by 0.27927178
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the METHODS REDUNDANCY by -0.17840001
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the METHODS REDUNDANCY by 0.1024522
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the METHODS REDUNDANCY by 0.15789852
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the METHODS REDUNDANCY by 0.023268526
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the METHODS REDUNDANCY by 0.043832436
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the METHODS REDUNDANCY by -0.023613838

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the METHODS REDUNDANCY by -0.13648549
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the METHODS REDUNDANCY by 0.10503769

Rules for OBJECTS COMPLEXITY

If the OBJECTS COMPLEXITY is being determined,
Then initialize the value of the OBJECTS COMPLEXITY to 0.04519999
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the OBJECTS COMPLEXITY by 0.21516515
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the OBJECTS COMPLEXITY by -0.046555318
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the OBJECTS COMPLEXITY by 0.0366422
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the OBJECTS COMPLEXITY by 0.0140000135
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the OBJECTS COMPLEXITY by -0.094155446
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the OBJECTS COMPLEXITY by -0.053079896
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the OBJECTS COMPLEXITY by -0.08320488
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the OBJECTS COMPLEXITY by -0.07650188
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the OBJECTS COMPLEXITY by -0.08537224
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the OBJECTS COMPLEXITY by 0.16986567
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the OBJECTS COMPLEXITY by 0.12712917
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the OBJECTS COMPLEXITY by 0.12000002
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the OBJECTS COMPLEXITY by -0.009048745
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the OBJECTS COMPLEXITY by 0.11433482
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the OBJECTS COMPLEXITY by 0.10942546
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the OBJECTS COMPLEXITY by 0.0911823
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the OBJECTS COMPLEXITY by -0.015631767
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the OBJECTS COMPLEXITY by -0.11153802
If the value of LINEARITY LINEAR is YES,
Then increment the OBJECTS COMPLEXITY by -0.028188974
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the OBJECTS COMPLEXITY by 0.06426294
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the OBJECTS COMPLEXITY by 0.10740984
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the OBJECTS COMPLEXITY by 0.12915306
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the OBJECTS COMPLEXITY by -0.1852
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the OBJECTS COMPLEXITY by -0.17007773
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the OBJECTS COMPLEXITY by 0.07640339
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the OBJECTS COMPLEXITY by 0.171162
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the OBJECTS COMPLEXITY by 0.22452389
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the OBJECTS COMPLEXITY by -0.024629401

If the value of OBJECTS SOLIDS is YES,
Then increment the OBJECTS COMPLEXITY by 0.0833406
If the value of OBJECTS FLUIDS is YES,
Then increment the OBJECTS COMPLEXITY by 0.16721192
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the OBJECTS COMPLEXITY by 0.10112477
If the value of SOLIDS STRUCTURE is YES,
Then increment the OBJECTS COMPLEXITY by 0.1885406
If the value of SOLIDS POINT-MASS is YES,
Then increment the OBJECTS COMPLEXITY by -0.014989952
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the OBJECTS COMPLEXITY by -0.016658407 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the OBJECTS COMPLEXITY by -0.036622074
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the OBJECTS COMPLEXITY by 0.08184224
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the OBJECTS COMPLEXITY by -0.070639506
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the OBJECTS COMPLEXITY by -0.106502794
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the OBJECTS COMPLEXITY by 0.18517883
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the OBJECTS COMPLEXITY by -0.11949584
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the OBJECTS COMPLEXITY by -0.13029991
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the OBJECTS COMPLEXITY by -0.042230725
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the OBJECTS COMPLEXITY by -0.13273154
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the OBJECTS COMPLEXITY by -0.049780123
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the OBJECTS COMPLEXITY by 0.0048310817
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the OBJECTS COMPLEXITY by 0.05968848
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the OBJECTS COMPLEXITY by 0.021755436
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the OBJECTS COMPLEXITY by 0.06378857
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the OBJECTS COMPLEXITY by -0.05382287
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the OBJECTS COMPLEXITY by -0.049087122
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the OBJECTS COMPLEXITY by -0.0641844
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the OBJECTS COMPLEXITY by 0.16636667
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the OBJECTS COMPLEXITY by -0.1062448
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the OBJECTS COMPLEXITY by 0.020366618
If the value of FLUIDS TURBULENCE is YES,
Then increment the OBJECTS COMPLEXITY by -0.092004865
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the OBJECTS COMPLEXITY by 0.18090081
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the OBJECTS COMPLEXITY by 0.11889417
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the OBJECTS COMPLEXITY by -0.011614353
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the OBJECTS COMPLEXITY by 0.16617233
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the OBJECTS COMPLEXITY by 0.13074224

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the OBJECTS COMPLEXITY by 0.053052235
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the OBJECTS COMPLEXITY by -0.021376679
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the OBJECTS COMPLEXITY by -0.05755708
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the OBJECTS COMPLEXITY by 0.025498124
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the OBJECTS COMPLEXITY by 0.004338534
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the OBJECTS COMPLEXITY by 0.008106437
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the OBJECTS COMPLEXITY by 0.064254664
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the OBJECTS COMPLEXITY by 0.06638891
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the OBJECTS COMPLEXITY by 0.07673928
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the OBJECTS COMPLEXITY by 0.06334536
If the value of GRAPHICS PRESENT is YES,
Then increment the OBJECTS COMPLEXITY by 0.0034009404
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the OBJECTS COMPLEXITY by -0.122660756
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the OBJECTS COMPLEXITY by -0.03143422
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the OBJECTS COMPLEXITY by -0.0024000108
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the OBJECTS COMPLEXITY by 0.07520001
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the OBJECTS COMPLEXITY by -0.09255971
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the OBJECTS COMPLEXITY by -0.0061653866
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the OBJECTS COMPLEXITY by -0.046206743
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the OBJECTS COMPLEXITY by -0.050544173
If the value of GRAPHICS SHADING is YES,
Then increment the OBJECTS COMPLEXITY by -0.022975259
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the OBJECTS COMPLEXITY by 0.022228852
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the OBJECTS COMPLEXITY by -0.112338
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the OBJECTS COMPLEXITY by -0.091770954
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the OBJECTS COMPLEXITY by 0.015049095
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the OBJECTS COMPLEXITY by 0.0239514
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the OBJECTS COMPLEXITY by -0.1985028
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the OBJECTS COMPLEXITY by 0.044
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the OBJECTS COMPLEXITY by -0.115247905
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the OBJECTS COMPLEXITY by 0.05930088
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the OBJECTS COMPLEXITY by -0.008109517
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the OBJECTS COMPLEXITY by -0.004890043
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the OBJECTS COMPLEXITY by 0.0077346903

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the OBJECTS COMPLEXITY by 0.045708653
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the OBJECTS COMPLEXITY by 0.07980425

Rules for OBJECTS GENERALITY

If the OBJECTS GENERALITY is being determined,
Then initialize the value of the OBJECTS GENERALITY to -0.055999994
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the OBJECTS GENERALITY by 0.10162884
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the OBJECTS GENERALITY by -0.03325494
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the OBJECTS GENERALITY by -0.012774584
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the OBJECTS GENERALITY by -0.072
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the OBJECTS GENERALITY by 0.15103379
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the OBJECTS GENERALITY by -0.10690186
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the OBJECTS GENERALITY by -0.14443783
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the OBJECTS GENERALITY by -0.12198767
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the OBJECTS GENERALITY by 0.07243895
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the OBJECTS GENERALITY by 0.12514801
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the OBJECTS GENERALITY by 0.05409316
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the OBJECTS GENERALITY by 0.037599996
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the OBJECTS GENERALITY by 0.11981277
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the OBJECTS GENERALITY by 0.03278412
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the OBJECTS GENERALITY by 0.027314818
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the OBJECTS GENERALITY by -0.03988077
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the OBJECTS GENERALITY by -0.028295638
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the OBJECTS GENERALITY by -0.07524317
If the value of LINEARITY LINEAR is YES,
Then increment the OBJECTS GENERALITY by -0.064337686
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the OBJECTS GENERALITY by 0.036316816
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the OBJECTS GENERALITY by 0.080736406
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the OBJECTS GENERALITY by 0.14693101
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the OBJECTS GENERALITY by -0.0036000013
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the OBJECTS GENERALITY by 0.03449942
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the OBJECTS GENERALITY by -0.15724611
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the OBJECTS GENERALITY by -0.017379496
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the OBJECTS GENERALITY by -0.021412456
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the OBJECTS GENERALITY by 0.018298078

If the value of OBJECTS SOLIDS is YES,
Then increment the OBJECTS GENERALITY by 0.12788956
If the value of OBJECTS FLUIDS is YES,
Then increment the OBJECTS GENERALITY by 0.11215701
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the OBJECTS GENERALITY by 0.2327919
If the value of SOLIDS STRUCTURE is YES,
Then increment the OBJECTS GENERALITY by -0.06851049
If the value of SOLIDS POINT-MASS is YES,
Then increment the OBJECTS GENERALITY by -0.11665861
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the OBJECTS GENERALITY by 0.016403897 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the OBJECTS GENERALITY by 4.5195913e-4
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the OBJECTS GENERALITY by 0.12762545
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the OBJECTS GENERALITY by -0.11794378
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the OBJECTS GENERALITY by 0.1155169
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the OBJECTS GENERALITY by 0.13148569
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the OBJECTS GENERALITY by -0.14716928
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the OBJECTS GENERALITY by 0.07891526
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the OBJECTS GENERALITY by 0.0044966806
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the OBJECTS GENERALITY by 0.0011860424
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the OBJECTS GENERALITY by -0.11896119
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the OBJECTS GENERALITY by 0.007280899
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the OBJECTS GENERALITY by -0.15460299
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the OBJECTS GENERALITY by 0.16975631
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the OBJECTS GENERALITY by 0.16311407
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the OBJECTS GENERALITY by 0.07205653
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the OBJECTS GENERALITY by 0.19642495
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the OBJECTS GENERALITY by 0.0553517
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the OBJECTS GENERALITY by 0.21519843
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the OBJECTS GENERALITY by -0.009287611
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the OBJECTS GENERALITY by 0.04159845
If the value of FLUIDS TURBULENCE is YES,
Then increment the OBJECTS GENERALITY by 0.10636214
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the OBJECTS GENERALITY by 0.14571068
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the OBJECTS GENERALITY by -0.041743964
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the OBJECTS GENERALITY by 0.23010847
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the OBJECTS GENERALITY by 0.2234146
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the OBJECTS GENERALITY by 5.495946e-4

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the OBJECTS GENERALITY by 0.061563645
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the OBJECTS GENERALITY by -0.06681028
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the OBJECTS GENERALITY by -0.019059034
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the OBJECTS GENERALITY by -0.0019877357
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the OBJECTS GENERALITY by 0.16197108
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the OBJECTS GENERALITY by -0.083079055
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the OBJECTS GENERALITY by 0.1345595
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the OBJECTS GENERALITY by 0.03666334
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the OBJECTS GENERALITY by -0.16985352
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the OBJECTS GENERALITY by -0.011306341
If the value of GRAPHICS PRESENT is YES,
Then increment the OBJECTS GENERALITY by -0.10905321
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the OBJECTS GENERALITY by 0.0041846717
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the OBJECTS GENERALITY by 0.20962992
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the OBJECTS GENERALITY by -0.1196
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the OBJECTS GENERALITY by -0.0264
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the OBJECTS GENERALITY by -0.086271614
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the OBJECTS GENERALITY by 0.05664652
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the OBJECTS GENERALITY by -0.06569221
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the OBJECTS GENERALITY by -0.08204249
If the value of GRAPHICS SHADING is YES,
Then increment the OBJECTS GENERALITY by -0.06532336
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the OBJECTS GENERALITY by 0.13954474
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the OBJECTS GENERALITY by -0.07127242
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the OBJECTS GENERALITY by -0.03326635
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the OBJECTS GENERALITY by -0.004313452
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the OBJECTS GENERALITY by -0.036107674
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the OBJECTS GENERALITY by -0.20848309
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the OBJECTS GENERALITY by -0.0644
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the OBJECTS GENERALITY by 0.14518543
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the OBJECTS GENERALITY by 0.106510624
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the OBJECTS GENERALITY by -0.098408155
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the OBJECTS GENERALITY by -0.07552766
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the OBJECTS GENERALITY by 0.18034904

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the OBJECTS GENERALITY by 0.16235644
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the OBJECTS GENERALITY by -0.060076125

Rules for REPRESENTATION COMPLEXITY

If the REPRESENTATION COMPLEXITY is being determined,
Then initialize the value of the REPRESENTATION COMPLEXITY to 0.042799994
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.16686828
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.14919484
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.047796763
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.092800006
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.09570364
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.050326157
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.16298206
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.00604719
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.17762595
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.03927471
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the REPRESENTATION COMPLEXITY by 0.17127965
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the REPRESENTATION COMPLEXITY by -0.1556
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the REPRESENTATION COMPLEXITY by -0.11586908
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the REPRESENTATION COMPLEXITY by 0.12558936
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the REPRESENTATION COMPLEXITY by 0.11092863
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.07048193
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.013608872
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.1437356
If the value of LINEARITY LINEAR is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.09606553
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.043462943
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.025767261
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the REPRESENTATION COMPLEXITY by 0.18592554
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the REPRESENTATION COMPLEXITY by -0.1588
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the REPRESENTATION COMPLEXITY by -0.098000005
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the REPRESENTATION COMPLEXITY by -0.13446915
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the REPRESENTATION COMPLEXITY by -0.01952804
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.15638651
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.15114854

If the value of OBJECTS SOLIDS is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.082550004
If the value of OBJECTS FLUIDS is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.0011474579
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.031639628
If the value of SOLIDS STRUCTURE is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.17735003
If the value of SOLIDS POINT-MASS is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.18323308
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the REPRESENTATION COMPLEXITY by 0.041050132 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.032889683
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.06899676
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the REPRESENTATION COMPLEXITY by 0.13538034
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the REPRESENTATION COMPLEXITY by -0.023467286
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the REPRESENTATION COMPLEXITY by -0.17962898
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the REPRESENTATION COMPLEXITY by 0.018265905
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the REPRESENTATION COMPLEXITY by -0.14022157
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.00904537
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.0727515
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.09524327
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the REPRESENTATION COMPLEXITY by 0.014032384
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the REPRESENTATION COMPLEXITY by 0.2025054
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the REPRESENTATION COMPLEXITY by -0.040206045
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the REPRESENTATION COMPLEXITY by -0.13499816
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the REPRESENTATION COMPLEXITY by -0.067405194
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.050769247
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.086162105
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.17298305
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.08721512
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.11541689
If the value of FLUIDS TURBULENCE is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.034582075
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.064106114
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.119057454
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.13250788
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.06277002
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.078964375

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.057779837
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.016637133
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the REPRESENTATION COMPLEXITY by -0.11615891
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the REPRESENTATION COMPLEXITY by -0.20955285
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the REPRESENTATION COMPLEXITY by -0.2394867
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the REPRESENTATION COMPLEXITY by -0.120701306
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the REPRESENTATION COMPLEXITY by 0.20462814
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.09084471
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.11643701
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.072339855
If the value of GRAPHICS PRESENT is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.015883047
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.04370362
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the REPRESENTATION COMPLEXITY by 0.15614952
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the REPRESENTATION COMPLEXITY by -0.0572
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the REPRESENTATION COMPLEXITY by -0.0476
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the REPRESENTATION COMPLEXITY by 0.06013069
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the REPRESENTATION COMPLEXITY by 0.016848186
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.046853837
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.0701064
If the value of GRAPHICS SHADING is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.018938785
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.18571968
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.022658737
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.026770685
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.092599876
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.008438659
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the REPRESENTATION COMPLEXITY by -0.068267316
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the REPRESENTATION COMPLEXITY by 0.09120001
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the REPRESENTATION COMPLEXITY by 0.08825782
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the REPRESENTATION COMPLEXITY by -0.10890608
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the REPRESENTATION COMPLEXITY by -0.11875599
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.08865587
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the REPRESENTATION COMPLEXITY by -0.059482887

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.023548262
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the REPRESENTATION COMPLEXITY by 0.03974405

Rules for REPRESENTATION CAPACITY

If the REPRESENTATION CAPACITY is being determined,
Then initialize the value of the REPRESENTATION CAPACITY to 0.12360002
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the REPRESENTATION CAPACITY by 0.20947029
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the REPRESENTATION CAPACITY by 0.12382602
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the REPRESENTATION CAPACITY by 0.13676071
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the REPRESENTATION CAPACITY by 0.11
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the REPRESENTATION CAPACITY by 0.02681863
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the REPRESENTATION CAPACITY by 0.04226289
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the REPRESENTATION CAPACITY by -0.0024175558
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the REPRESENTATION CAPACITY by -0.042103622
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the REPRESENTATION CAPACITY by 0.0604303
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the REPRESENTATION CAPACITY by -0.030268282
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the REPRESENTATION CAPACITY by 0.15467131
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the REPRESENTATION CAPACITY by 0.036799997
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the REPRESENTATION CAPACITY by -0.088236645
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the REPRESENTATION CAPACITY by -0.04330272
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the REPRESENTATION CAPACITY by 0.023356913
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the REPRESENTATION CAPACITY by -1.289394e-4
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the REPRESENTATION CAPACITY by -0.032063894
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the REPRESENTATION CAPACITY by -0.09683384
If the value of LINEARITY LINEAR is YES,
Then increment the REPRESENTATION CAPACITY by -0.22735152
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the REPRESENTATION CAPACITY by -0.1724476
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the REPRESENTATION CAPACITY by -0.07848824
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the REPRESENTATION CAPACITY by 0.046951666
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the REPRESENTATION CAPACITY by -0.062800005
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the REPRESENTATION CAPACITY by 0.041503508
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the REPRESENTATION CAPACITY by -0.04803986
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the REPRESENTATION CAPACITY by 0.03687363
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the REPRESENTATION CAPACITY by 0.12830366
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the REPRESENTATION CAPACITY by -0.21488462

If the value of OBJECTS SOLIDS is YES,
Then increment the REPRESENTATION CAPACITY by -0.013689235
If the value of OBJECTS FLUIDS is YES,
Then increment the REPRESENTATION CAPACITY by 0.017570779
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the REPRESENTATION CAPACITY by -0.097927086
If the value of SOLIDS STRUCTURE is YES,
Then increment the REPRESENTATION CAPACITY by -0.086089246
If the value of SOLIDS POINT-MASS is YES,
Then increment the REPRESENTATION CAPACITY by 0.05069558
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the REPRESENTATION CAPACITY by 0.026740784 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the REPRESENTATION CAPACITY by -0.18308729
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the REPRESENTATION CAPACITY by 0.05756072
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the REPRESENTATION CAPACITY by -0.014473833
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the REPRESENTATION CAPACITY by 0.17881878
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the REPRESENTATION CAPACITY by 0.16068688
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the REPRESENTATION CAPACITY by 0.044878904
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the REPRESENTATION CAPACITY by 0.009378142
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the REPRESENTATION CAPACITY by 0.017610215
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the REPRESENTATION CAPACITY by 0.021912908
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the REPRESENTATION CAPACITY by 0.12267725
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the REPRESENTATION CAPACITY by 0.04157497
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the REPRESENTATION CAPACITY by -0.01280546
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the REPRESENTATION CAPACITY by 0.05299077
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the REPRESENTATION CAPACITY by 0.02373507
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the REPRESENTATION CAPACITY by 0.082593545
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the REPRESENTATION CAPACITY by -0.055452295
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the REPRESENTATION CAPACITY by -0.034368105
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the REPRESENTATION CAPACITY by 0.124688536
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the REPRESENTATION CAPACITY by -0.13317645
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the REPRESENTATION CAPACITY by -0.07731149
If the value of FLUIDS TURBULENCE is YES,
Then increment the REPRESENTATION CAPACITY by 0.11718244
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the REPRESENTATION CAPACITY by -0.04190271
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the REPRESENTATION CAPACITY by 0.06659781
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the REPRESENTATION CAPACITY by -0.11011983
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the REPRESENTATION CAPACITY by 0.23126791
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the REPRESENTATION CAPACITY by -0.005489776

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the REPRESENTATION CAPACITY by -0.10195692
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the REPRESENTATION CAPACITY by 0.12073451
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the REPRESENTATION CAPACITY by 0.1444195
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the REPRESENTATION CAPACITY by -0.093303606
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the REPRESENTATION CAPACITY by -0.15993859
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the REPRESENTATION CAPACITY by -0.1357664
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the REPRESENTATION CAPACITY by -0.06132197
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the REPRESENTATION CAPACITY by 0.05646846
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the REPRESENTATION CAPACITY by 0.069782384
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the REPRESENTATION CAPACITY by -0.07577573
If the value of GRAPHICS PRESENT is YES,
Then increment the REPRESENTATION CAPACITY by 0.09290389
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the REPRESENTATION CAPACITY by 0.08009502
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the REPRESENTATION CAPACITY by -0.026189357
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the REPRESENTATION CAPACITY by -0.0984
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the REPRESENTATION CAPACITY by -0.036799997
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the REPRESENTATION CAPACITY by -0.07682752
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the REPRESENTATION CAPACITY by 0.11230574
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the REPRESENTATION CAPACITY by -0.008392328
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the REPRESENTATION CAPACITY by 0.13866976
If the value of GRAPHICS SHADING is YES,
Then increment the REPRESENTATION CAPACITY by -0.10210524
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the REPRESENTATION CAPACITY by -0.11635558
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the REPRESENTATION CAPACITY by 0.21442357
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the REPRESENTATION CAPACITY by -0.14154269
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the REPRESENTATION CAPACITY by 0.13081498
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the REPRESENTATION CAPACITY by 0.12607463
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the REPRESENTATION CAPACITY by 0.17601879
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the REPRESENTATION CAPACITY by 0.07760002
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the REPRESENTATION CAPACITY by -0.059633717
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the REPRESENTATION CAPACITY by 0.0028972898
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the REPRESENTATION CAPACITY by -0.0015934804
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the REPRESENTATION CAPACITY by -0.13245916
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the REPRESENTATION CAPACITY by 0.06743567

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the REPRESENTATION CAPACITY by -0.046228122
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the REPRESENTATION CAPACITY by -0.12097398

Rules for REPRESENTATION KNOWLEDGE

If the REPRESENTATION KNOWLEDGE is being determined,
Then initialize the value of the REPRESENTATION KNOWLEDGE to 0.13720001
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.2173163
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.08827213
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.022969322
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.0176
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.23901685
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.03760481
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.033907894
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.009357327
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.16072051
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.09498095
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the REPRESENTATION KNOWLEDGE by -0.047552332
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the REPRESENTATION KNOWLEDGE by 0.1328
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the REPRESENTATION KNOWLEDGE by -0.058159836
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the REPRESENTATION KNOWLEDGE by -0.049738284
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the REPRESENTATION KNOWLEDGE by 0.1046461
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.061223034
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.06664058
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.075219825
If the value of LINEARITY LINEAR is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.075415045
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.09451562
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.08628404
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the REPRESENTATION KNOWLEDGE by 0.039990295
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the REPRESENTATION KNOWLEDGE by -0.0536
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the REPRESENTATION KNOWLEDGE by -0.03695123
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the REPRESENTATION KNOWLEDGE by -0.07507549
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the REPRESENTATION KNOWLEDGE by -0.089568034
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.07495986
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.07769095

If the value of OBJECTS SOLIDS is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.18962619
If the value of OBJECTS FLUIDS is YES,
Then increment the REPRESENTATION KNOWLEDGE by -9.9577055e-4
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.0496323
If the value of SOLIDS STRUCTURE is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.014426237
If the value of SOLIDS POINT-MASS is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.11631391
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the REPRESENTATION KNOWLEDGE by -0.09096407 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.12738419
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.24223068
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the REPRESENTATION KNOWLEDGE by -0.054458175
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the REPRESENTATION KNOWLEDGE by 0.023751868
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the REPRESENTATION KNOWLEDGE by -0.1160022
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the REPRESENTATION KNOWLEDGE by -0.048865277
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the REPRESENTATION KNOWLEDGE by -0.106830664
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.027690813
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.037396647
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.051293727
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the REPRESENTATION KNOWLEDGE by 0.190605
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the REPRESENTATION KNOWLEDGE by 0.116339475
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the REPRESENTATION KNOWLEDGE by -0.059427075
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the REPRESENTATION KNOWLEDGE by 0.032987103
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the REPRESENTATION KNOWLEDGE by -0.082109004
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.032173906
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.012405593
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.1135283
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.10614117
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.032328304
If the value of FLUIDS TURBULENCE is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.105107896
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.073894195
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.12155227
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.094337575
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.0342653
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.023027934

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.09265282
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.12920085
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the REPRESENTATION KNOWLEDGE by 0.19993202
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the REPRESENTATION KNOWLEDGE by -0.011357345
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the REPRESENTATION KNOWLEDGE by -0.066490225
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the REPRESENTATION KNOWLEDGE by 0.06637471
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the REPRESENTATION KNOWLEDGE by 0.01953635
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.15343758
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.1258015
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.042462017
If the value of GRAPHICS PRESENT is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.21358623
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.17962344
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the REPRESENTATION KNOWLEDGE by -0.06342692
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the REPRESENTATION KNOWLEDGE by -0.1104
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the REPRESENTATION KNOWLEDGE by -0.1632
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the REPRESENTATION KNOWLEDGE by -0.0840753
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the REPRESENTATION KNOWLEDGE by 0.017497722
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.16753446
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.09820339
If the value of GRAPHICS SHADING is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.10399837
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.06266849
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.06453058
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.11933995
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.06292531
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.09787561
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the REPRESENTATION KNOWLEDGE by -0.03384813
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the REPRESENTATION KNOWLEDGE by -0.19240001
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the REPRESENTATION KNOWLEDGE by 0.13544938
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the REPRESENTATION KNOWLEDGE by -0.077705786
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the REPRESENTATION KNOWLEDGE by -0.07069985
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.036369752
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the REPRESENTATION KNOWLEDGE by -0.026831888

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.030754123
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the REPRESENTATION KNOWLEDGE by 0.06478346

Rules for GRAPHICS COMPLEXITY

If the GRAPHICS COMPLEXITY is being determined,
Then initialize the value of the GRAPHICS COMPLEXITY to 0.0448
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the GRAPHICS COMPLEXITY by 0.13745873
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the GRAPHICS COMPLEXITY by 0.051921915
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the GRAPHICS COMPLEXITY by -0.22406907
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the GRAPHICS COMPLEXITY by 0.1348
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the GRAPHICS COMPLEXITY by 0.09458021
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the GRAPHICS COMPLEXITY by -0.1044173
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the GRAPHICS COMPLEXITY by 0.063870996
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the GRAPHICS COMPLEXITY by -0.057164595
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the GRAPHICS COMPLEXITY by -0.18686852
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the GRAPHICS COMPLEXITY by 0.008416869
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the GRAPHICS COMPLEXITY by 0.081855066
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the GRAPHICS COMPLEXITY by 0.1912
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the GRAPHICS COMPLEXITY by 0.20390984
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the GRAPHICS COMPLEXITY by -0.05315444
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the GRAPHICS COMPLEXITY by 0.09036513
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the GRAPHICS COMPLEXITY by -0.1633547
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the GRAPHICS COMPLEXITY by 0.1619585
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the GRAPHICS COMPLEXITY by -0.097325645
If the value of LINEARITY LINEAR is YES,
Then increment the GRAPHICS COMPLEXITY by 0.022856645
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the GRAPHICS COMPLEXITY by -0.07202876
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the GRAPHICS COMPLEXITY by 0.16916552
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the GRAPHICS COMPLEXITY by -0.0010720321
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the GRAPHICS COMPLEXITY by 0.16960002
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the GRAPHICS COMPLEXITY by 0.0882549
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the GRAPHICS COMPLEXITY by -0.096906446
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the GRAPHICS COMPLEXITY by -0.03810076
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the GRAPHICS COMPLEXITY by 0.0546883
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the GRAPHICS COMPLEXITY by -0.011718076

If the value of OBJECTS SOLIDS is YES,
Then increment the GRAPHICS COMPLEXITY by 0.15294707
If the value of OBJECTS FLUIDS is YES,
Then increment the GRAPHICS COMPLEXITY by -0.026280334
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the GRAPHICS COMPLEXITY by 0.02776715
If the value of SOLIDS STRUCTURE is YES,
Then increment the GRAPHICS COMPLEXITY by -0.07185299
If the value of SOLIDS POINT-MASS is YES,
Then increment the GRAPHICS COMPLEXITY by 0.12253052
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the GRAPHICS COMPLEXITY by -0.05187925 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the GRAPHICS COMPLEXITY by 0.013790152
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the GRAPHICS COMPLEXITY by 0.110730976
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the GRAPHICS COMPLEXITY by -0.15947407
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the GRAPHICS COMPLEXITY by -0.15057854
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the GRAPHICS COMPLEXITY by -0.0358367
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the GRAPHICS COMPLEXITY by -0.063163616
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the GRAPHICS COMPLEXITY by 0.031628624
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the GRAPHICS COMPLEXITY by -0.15507583
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the GRAPHICS COMPLEXITY by 0.24870501
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the GRAPHICS COMPLEXITY by 0.09080952
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the GRAPHICS COMPLEXITY by 0.18421967
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the GRAPHICS COMPLEXITY by -0.16499245
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the GRAPHICS COMPLEXITY by 0.020492848
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the GRAPHICS COMPLEXITY by 0.1716559
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the GRAPHICS COMPLEXITY by 0.019599661
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the GRAPHICS COMPLEXITY by 0.25169984
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the GRAPHICS COMPLEXITY by -0.035528794
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the GRAPHICS COMPLEXITY by 0.14990298
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the GRAPHICS COMPLEXITY by -0.13684109
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the GRAPHICS COMPLEXITY by -0.16929705
If the value of FLUIDS TURBULENCE is YES,
Then increment the GRAPHICS COMPLEXITY by -0.24692906
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the GRAPHICS COMPLEXITY by -0.12298087
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the GRAPHICS COMPLEXITY by -0.030469086
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the GRAPHICS COMPLEXITY by 0.045080982
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the GRAPHICS COMPLEXITY by 0.17504546
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the GRAPHICS COMPLEXITY by 0.07372439

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the GRAPHICS COMPLEXITY by -0.008679777
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the GRAPHICS COMPLEXITY by 4.099533e-4
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the GRAPHICS COMPLEXITY by 0.08043983
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the GRAPHICS COMPLEXITY by 0.12003534
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the GRAPHICS COMPLEXITY by -0.10707315
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the GRAPHICS COMPLEXITY by 0.10175983
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the GRAPHICS COMPLEXITY by -0.095386274
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the GRAPHICS COMPLEXITY by 0.06275554
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the GRAPHICS COMPLEXITY by 0.019990982
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the GRAPHICS COMPLEXITY by -0.1061834
If the value of GRAPHICS PRESENT is YES,
Then increment the GRAPHICS COMPLEXITY by 0.016617678
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the GRAPHICS COMPLEXITY by 0.09830821
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the GRAPHICS COMPLEXITY by 0.15021369
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the GRAPHICS COMPLEXITY by -0.091199994
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the GRAPHICS COMPLEXITY by -0.13679999
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the GRAPHICS COMPLEXITY by 0.11122601
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the GRAPHICS COMPLEXITY by -0.11846418
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the GRAPHICS COMPLEXITY by 0.211934
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the GRAPHICS COMPLEXITY by -0.09404178
If the value of GRAPHICS SHADING is YES,
Then increment the GRAPHICS COMPLEXITY by 0.20693853
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the GRAPHICS COMPLEXITY by 0.0019554754
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the GRAPHICS COMPLEXITY by -0.11757948
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the GRAPHICS COMPLEXITY by 0.10731269
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the GRAPHICS COMPLEXITY by -0.06290726
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the GRAPHICS COMPLEXITY by -0.07365038
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the GRAPHICS COMPLEXITY by -0.16697855
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the GRAPHICS COMPLEXITY by 0.14240001
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the GRAPHICS COMPLEXITY by 0.17701817
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the GRAPHICS COMPLEXITY by 0.08381913
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the GRAPHICS COMPLEXITY by 0.07431693
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the GRAPHICS COMPLEXITY by -0.025744481
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the GRAPHICS COMPLEXITY by -0.05840595

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the GRAPHICS COMPLEXITY by -0.13415068
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the GRAPHICS COMPLEXITY by 0.1943074

Rules for USER-INTERFACE COMPLEXITY

If the USER-INTERFACE COMPLEXITY is being determined,
Then initialize the value of the USER-INTERFACE COMPLEXITY to 0.07000001
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.13245223
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.18114536
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.024334712
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.1756
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.19553381
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.014402249
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.11173894
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.16490908
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the USER-INTERFACE COMPLEXITY by -3.0301845e-4
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.18589935
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the USER-INTERFACE COMPLEXITY by -0.08028955
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the USER-INTERFACE COMPLEXITY by 0.088000014
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the USER-INTERFACE COMPLEXITY by -0.0021219493
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the USER-INTERFACE COMPLEXITY by 0.0093895625
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the USER-INTERFACE COMPLEXITY by -0.053331327
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.13923872
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.17513557
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.061767865
If the value of LINEARITY LINEAR is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.07737879
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.10258101
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.11315591
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the USER-INTERFACE COMPLEXITY by -0.023431506
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the USER-INTERFACE COMPLEXITY by -0.17
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the USER-INTERFACE COMPLEXITY by -0.15108417
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the USER-INTERFACE COMPLEXITY by -0.11919303
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the USER-INTERFACE COMPLEXITY by -0.031444557
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.11455127
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.057881147

If the value of OBJECTS SOLIDS is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.04454463
If the value of OBJECTS FLUIDS is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.16009775
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.0059253518
If the value of SOLIDS STRUCTURE is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.056255385
If the value of SOLIDS POINT-MASS is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.0013907182
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the USER-INTERFACE COMPLEXITY by 0.04774377 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.056670286
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.08913474
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the USER-INTERFACE COMPLEXITY by -0.14310122
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the USER-INTERFACE COMPLEXITY by 0.0771822
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the USER-INTERFACE COMPLEXITY by -0.1437669
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the USER-INTERFACE COMPLEXITY by -0.08736947
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the USER-INTERFACE COMPLEXITY by 0.060702093
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.074873686
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.019569788
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.090833575
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the USER-INTERFACE COMPLEXITY by -0.038398672
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the USER-INTERFACE COMPLEXITY by -0.026146784
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the USER-INTERFACE COMPLEXITY by 0.1256328
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the USER-INTERFACE COMPLEXITY by -0.06205655
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the USER-INTERFACE COMPLEXITY by -0.08778411
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.03268828
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.09786448
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.07999943
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.082455985
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.17000063
If the value of FLUIDS TURBULENCE is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.011861053
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.13395649
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.1704554
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.06823593
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.07052754
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.11484991

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.079736754
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.04402837
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the USER-INTERFACE COMPLEXITY by 0.09014916
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the USER-INTERFACE COMPLEXITY by -0.05949088
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the USER-INTERFACE COMPLEXITY by -0.023245724
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the USER-INTERFACE COMPLEXITY by -0.004369741
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the USER-INTERFACE COMPLEXITY by 0.26100376
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.11492483
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.1100038
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.14592952
If the value of GRAPHICS PRESENT is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.10625532
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.022898072
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the USER-INTERFACE COMPLEXITY by -0.05556915
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the USER-INTERFACE COMPLEXITY by -0.076400004
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the USER-INTERFACE COMPLEXITY by -0.1648
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the USER-INTERFACE COMPLEXITY by -0.07928447
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the USER-INTERFACE COMPLEXITY by -0.16029972
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.0045090243
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.045826126
If the value of GRAPHICS SHADING is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.06657254
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.06309887
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.1254123
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.042816233
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.031493403
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.059063252
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the USER-INTERFACE COMPLEXITY by 0.06518222
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the USER-INTERFACE COMPLEXITY by 0.13080002
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the USER-INTERFACE COMPLEXITY by -0.024565415
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the USER-INTERFACE COMPLEXITY by 0.007556507
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the USER-INTERFACE COMPLEXITY by -0.05692658
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.011456958
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.08002952

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the USER-INTERFACE COMPLEXITY by -0.021698138
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the USER-INTERFACE COMPLEXITY by 0.12416149

Rules for DATA-MANAGEMENT COMPLEXITY

If the DATA-MANAGEMENT COMPLEXITY is being determined,
Then initialize the value of the DATA-MANAGEMENT COMPLEXITY to 0.030800015
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.1328848
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.11190756
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.16719842
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.0059999973
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.08180122
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.107553236
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.06495628
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.020205645
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.08227366
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.11679537
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.14359973
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.1596
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.049590323
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.08642122
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.070774995
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.13862693
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.044492967
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.06571752
If the value of LINEARITY LINEAR is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.020146439
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.15141861
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.037130363
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.16823906
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.0964
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.050149847
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the DATA-MANAGEMENT COMPLEXITY by -5.0564437e-4
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.15072002
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.0657354
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.05918107

If the value of OBJECTS SOLIDS is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.08242493
If the value of OBJECTS FLUIDS is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.09111613
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.1472881
If the value of SOLIDS STRUCTURE is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.18597515
If the value of SOLIDS POINT-MASS is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.07900209
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.07240349 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.13700584
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.16920157
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.091537826
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.060811214
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.09339666
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.08614813
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.026738307
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.18323928
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.11598683
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.2587935
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.0038053894
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.19425584
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.015351142
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.16324447
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.040506262
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.016459461
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.15331402
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.015236954
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.041592427
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.012036955
If the value of FLUIDS TURBULENCE is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.026956264
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.14162615
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.07651837
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.17744781
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.028189456
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.06585189

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.024245817
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.09547308
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.10910679
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.011005638
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.089046106
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.17651431
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.18289831
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.1384533
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.19121297
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.04430574
If the value of GRAPHICS PRESENT is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.14807615
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.14292651
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.20825087
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.111600004
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.1052
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.026589954
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.15950225
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.11946601
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.16044487
If the value of GRAPHICS SHADING is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.005749755
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.08386179
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.12378141
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.060346447
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.1781645
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.051693242
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.15481125
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.063999996
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.08323041
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.16842614
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the DATA-MANAGEMENT COMPLEXITY by 0.06749542
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.029304251
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.05195449

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.052845396
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the DATA-MANAGEMENT COMPLEXITY by -0.011086686

Rules for SYS-PROGRAMMING COMPLEXITY

If the SYS-PROGRAMMING COMPLEXITY is being determined,
Then initialize the value of the SYS-PROGRAMMING COMPLEXITY to -0.0908
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.16359632
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.08547161
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.040477786
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.12
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.013192825
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.0892433
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.13405645
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.10926937
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.03620786
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.21187292
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.06884393
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.09319998
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.07857715
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.049253486
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.14125526
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.04726411
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.025362883
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.12685674
If the value of LINEARITY LINEAR is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.11672532
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.032994755
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.028932847
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.17232803
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.1276
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.03278401
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.063084155
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.19856298
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.16505584
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.015859857

If the value of OBJECTS SOLIDS is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.12382034
If the value of OBJECTS FLUIDS is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.1565429
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.05678254
If the value of SOLIDS STRUCTURE is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.10937967
If the value of SOLIDS POINT-MASS is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.18489724
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.07613311 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.23471975
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.1879222
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.14820513
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.14660043
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.09133425
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.1495489
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.010844522
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.113107875
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.14231206
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.08039446
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.034513324
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.18228686
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.14938244
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.048873372
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.06732052
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.0028997844
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.003462799
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.005136408
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.11933698
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.059263613
If the value of FLUIDS TURBULENCE is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.014056447
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.0776042
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.042663723
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.08011407
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.09336077
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.05792645

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.05903819
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.0045040944
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.07055829
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.0823306
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.016149998
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.09883577
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.01526177
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.08938202
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.057058636
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.10713856
If the value of GRAPHICS PRESENT is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.02903828
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.20216523
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.13957314
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.028400004
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.1176
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.009577551
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.10181556
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.12706459
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.16325134
If the value of GRAPHICS SHADING is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.08966197
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.115019925
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.039512943
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.05559422
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.088850394
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.1045798
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.14420044
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.1544
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.20111653
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.12159582
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.016023159
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.03277557
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.1448865

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by -0.07633767
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the SYS-PROGRAMMING COMPLEXITY by 0.15317217

Rules for HARD-ARCHITECTURE COMPLEXITY

If the HARD-ARCHITECTURE COMPLEXITY is being determined,
Then initialize the value of the HARD-ARCHITECTURE COMPLEXITY to -0.154
If the value of TASK ENGINEERING-PROGRAM is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.0013482039
If the value of TASK DATABASE-PROGRAM is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.117435575
If the value of ENGINEERING-PROGRAM STRUCTURAL-ENGINEERING is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.119745724
If the value of ENGINEERING-PROGRAM ELECTRONIC-CIRCUIT is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.13640001
If the value of ENGINEERING-PROGRAM THERMODYNAMICS is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.05105252
If the value of ENGINEERING-PROGRAM CONTROL-SYSTEMS is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.119598225
If the value of ENGINEERING-PROGRAM AERODYNAMICS is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.036167707
If the value of ENGINEERING-PROGRAM IMAGE-PROCESSING is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.09138998
If the value of THERMODYNAMICS RADIATION is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.0225285
If the value of THERMODYNAMICS CONDUCTION is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.08044807
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-HIGH,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.0014083743
If the value of METHODS NUMBER-REDUNDANT-CHOICES is HIGH,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.075600006
If the value of METHODS NUMBER-REDUNDANT-CHOICES is MEDIUM,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.09931266
If the value of METHODS NUMBER-REDUNDANT-CHOICES is LOW,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.12080977
If the value of METHODS NUMBER-REDUNDANT-CHOICES is VERY-LOW,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.002474317
If the value of TIME-DEPENDENCE STATIC is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.16035777
If the value of TIME-DEPENDENCE QUASI-STATIC is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.052254252
If the value of TIME-DEPENDENCE FULLY-DYNAMIC is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.12496841
If the value of LINEARITY LINEAR is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.12732668
If the value of LINEARITY LIMITED-NON-LINEAR is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.08542358
If the value of LINEARITY FULLY-NON-LINEAR is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.12750362
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-HIGH,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.08902612
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is HIGH,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.006800011
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is MEDIUM,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.15747282
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is LOW,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.04211257
If the value of BOUNDARY NUMBER-WAYS-TO-SPECIFY is VERY-LOW,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.030348841
If the value of BOUNDARY 2D-FIELD-ON-3D-SURFACE is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.013165859
If the value of BOUNDARY LUMPED-COMPONENT-VALUES is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.056185313

If the value of OBJECTS SOLIDS is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.013026827
If the value of OBJECTS FLUIDS is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.02490823
If the value of OBJECTS NON-PHYSICAL is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.14028639
If the value of SOLIDS STRUCTURE is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.10497319
If the value of SOLIDS POINT-MASS is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.1179648
If the value of STRUCTURE SPATIAL-DIMENSION is X,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.060789485 * X
If the value of STRUCTURE LUMPED-PARAMETER is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.13399489
If the value of STRUCTURE CONTINUOUS-VOLUME is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.06785428
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-HIGH,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.024088793
If the value of STRUCTURE NUMBER-OF-ELEMENTS is HIGH,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.12677409
If the value of STRUCTURE NUMBER-OF-ELEMENTS is MEDIUM,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.046499394
If the value of STRUCTURE NUMBER-OF-ELEMENTS is LOW,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.09398723
If the value of STRUCTURE NUMBER-OF-ELEMENTS is VERY-LOW,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.17246653
If the value of STRUCTURE SUBSTRUCTURING-CAPABILITY is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.18430778
If the value of CONTINUOUS-VOLUME FINITE-ELEMENTS is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.08664629
If the value of CONTINUOUS-VOLUME GRID is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.09461483
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-HIGH,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.039981585
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is HIGH,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.098168425
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is MEDIUM,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.123943664
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is LOW,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.05899225
If the value of CONTINUOUS-VOLUME NUMBER-OF-KNOWN-SHAPES is VERY-LOW,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.13542567
If the value of SHAPE-LIMITATIONS ESSENTIALLY-NONE is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.10458109
If the value of SHAPE-LIMITATIONS THIN-SHELLS is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.13634907
If the value of SHAPE-LIMITATIONS SHELLS-OF-REVOL is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.0025769756
If the value of SHAPE-LIMITATIONS SLABS is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.009630773
If the value of SHAPE-LIMITATIONS THICK-SHELLS is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.099823035
If the value of FLUIDS TURBULENCE is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.06416767
If the value of FLUIDS SHOCK-WAVES is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.16104586
If the value of USER-INTERFACE TEXT-EDITOR is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.12005911
If the value of USER-INTERFACE DATA-MANIPULATOR is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.035557877
If the value of USER-INTERFACE CONTROL-LANGUAGE is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.009600595
If the value of USER-INTERFACE USER-SPECIFIED-ROUTINES is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.0052227573

If the value of USER-INTERFACE MATRIX-SPEC-LANGUAGE is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.07349289
If the value of USER-INTERFACE LIBRARIES is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.03242931
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-HIGH,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.06433589
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is HIGH,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.1649899
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is MEDIUM,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.05695246
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is LOW,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.10026162
If the value of LIBRARIES NUMBER-OF-STORED-SHAPES is VERY-LOW,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.0485436
If the value of LIBRARIES FORMAT-CONVERSION is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.032308694
If the value of LIBRARIES GEOMETRIC-CONVERSION is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.085061125
If the value of LIBRARIES TRANSLATE-FOR-OTHER-PROGS is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.06991953
If the value of GRAPHICS PRESENT is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.03515198
If the value of GRAPHICS 3D-STRUCTURE is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.15362635
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-HIGH,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.114321284
If the value of GRAPHICS NUMBER-OF-FORMATS is HIGH,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.048000008
If the value of GRAPHICS NUMBER-OF-FORMATS is MEDIUM,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.154
If the value of GRAPHICS NUMBER-OF-FORMATS is LOW,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.021870507
If the value of GRAPHICS NUMBER-OF-FORMATS is VERY-LOW,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.020447895
If the value of GRAPHICS PERSPECTIVE is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.030375035
If the value of GRAPHICS HIDDEN-LINE-REMOVAL is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.20122872
If the value of GRAPHICS SHADING is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.13915284
If the value of COMMERCIALIZATION MANY-PROJECTS is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.035442844
If the value of COMMERCIALIZATION MANY-COMPANIES is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.07679934
If the value of COMMERCIALIZATION INDUSTRY-STANDARD is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.016351426
If the value of COMMERCIALIZATION MANY-UPGRADES is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.019292513
If the value of COMMERCIALIZATION MANY-COMPUTERS is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.053422067
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-HIGH,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.1116259
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is HIGH,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.012800008
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is MEDIUM,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.13154343
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is LOW,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.067045875
If the value of LUMPED-PARAMETER NUMBER-KNOWN-COMPONENTS is VERY-LOW,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.040836886
If the value of DATABASE-PROGRAM VIDEO-IMAGES is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by -0.007307438
If the value of NON-PHYSICAL IMAGES is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.11873978

If the value of SYSTEMS-PROGRAMMING EXECUTE-COMMANDS is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.0017242436
If the value of SYSTEMS-PROGRAMMING CODE-TO-SAVE-MEMORY is YES,
Then increment the HARD-ARCHITECTURE COMPLEXITY by 0.024214413

Appendix 3 - Program Database

This appendix contains the software databases used by the Knowledge Acquisition System including specification files for the existing codes and hand tuned examples, and generic component files for the existing codes and hand tuned examples, in files:

1. hd:nasa:knowledge acquisition system:specification database
2. hd:nasa:knowledge acquisition system:hand-tunes-specs.lisp
3. hd:nasa:knowledge acquisition system:generic database
4. hd:nasa:knowledge acquisition system:hand-tuned-generic.lisp

hd:nasa:knowledge acquisition system:specification database

Program	nastran	trasys	stagsc-1	sspta
TASK				
Task Engineering-Program	yes	yes	yes	yes
Engineering-Program Structural-engineering	yes	no	yes	no
Engineering-Program electronic-circuit	no	no	no	no
Engineering-Program Thermodynamics	yes	yes	no	yes
Engineering-Program Control-systems	yes	no	no	no
Thermodynamics radiation	no	yes	no	yes
Thermodynamics conduction	yes	no	no	yes
METHODS				
methods number-redundant-choices	very-high	medium	very-low	low
time-dependence static	yes	yes	yes	no
time-dependence quasi-static	yes	yes	yes	yes
time-dependence fully-dynamic	yes	no	yes	no
linearity linear	yes	yes	yes	yes
linearity limited-non-linear	yes	yes	yes	yes
linearity fully-non-linear	no	no	yes	no
boundary number-ways-to-specify	very-high	medium	medium	medium
boundary 2D-field-on-3D-surface	yes	no	yes	yes
boundary lumped-component-values	yes	no	no	yes
OBJECTS & REPRESENTATIONS				
objects solids	yes	yes	yes	yes
objects fluids	yes	no	no	no
solids structure	yes	yes	yes	yes
solids point-mass	yes	no	no	no
structure spatial-dimension	3	3	3	3
structure lumped-parameter	no	yes	no	yes
structure continuous-volume	yes	no	yes	no
continuous-volume finite-elements	yes	no	yes	n/a
continuous-volume grid	no	no	no	n/a
structure number-of-elements	very-high	medium	low	low
structure substructuring-capability	yes	no	yes	no
continuous-volume number-of-known-shapes	very-high	n/a	medium	n/a
shape-limitations essentially-none	yes	n/a	no	no
shape-limitations thin-shells	yes	n/a	yes	yes
shape-limitations shells-of-revol	yes	n/a	yes	no
shape-limitations slabs	yes	n/a	yes	no
shape-limitations thick-shells	yes	n/a	yes	no
fluids turbulence	no	n/a	n/a	n/a
fluids shock-waves	no	n/a	n/a	n/a

USER INTERFACE

user-interface text-editor	no	yes	no	no
user-interface data-manipulator	yes	yes	no	no
user-interface control-language	yes	yes	no	yes
user-interface user-specified-routines	yes	yes	no	no
user-interface matrix-spec-language	yes	yes	no	no
user-interface libraries	yes	yes	yes	no
libraries number-of-stored-shapes	very-high	n/a	low	low
libraries format-conversion	yes	yes	no	yes
libraries geometric-conversion	yes	yes	no	yes
libraries translate-for-other-progs	no	yes	no	yes

GRAPHICS

graphics present	yes	yes	no	yes
graphics 3D-structure	yes	yes	n/a	yes
graphics number-of-formats	very-high	low	n/a	very-low
graphics perspective	yes	no	n/a	yes
graphics hidden-line-removal	yes	no	n/a	no
graphics shading	no	no	n/a	no

OTHER

commercialization many-projects	yes	yes	yes	yes
commercialization many-companies	yes	yes	yes	yes
commercialization industry-standard	yes	yes	no	no
commercialization many-upgrades	yes	yes	no	no
commercialization many-computers	yes	yes	yes	yes

Program	pasco	panair	sinda
TASK			
Task Engineering-Program	yes	yes	yes
Engineering-Program Structural-engineering	yes	yes	no
Engineering-program aerodynamics	no	yes	no
Engineering-Program electronic-circuit	no	no	no
Engineering-Program Thermodynamics	no	no	yes
Thermodynamics radiation	n/a	n/a	no
Thermodynamics conduction	n/a	n/a	yes
METHODS			
methods number-redundant-choices	low	very-low	low
time-dependence static	yes	no	yes
time-dependence quasi-static	yes	no	yes
time-dependence fully-dynamic	no	yes	no
linearity linear	yes	no	yes
linearity limited-non-linear	yes	no	yes
linearity fully-non-linear	no	yes	no
boundary number-ways-to-specify	medium	very-high	medium
boundary 2D-field-on-3D-surface	no	yes	no
boundary lumped-component-values	no	yes	yes
OBJECTS & REPRESENTATIONS			
objects solids	yes	yes	yes
objects fluids	no	yes	yes
solids structure	yes	yes	yes
solids point-mass	no	no	no
structure spatial-dimension	2	3	n/a
structure lumped-parameter	no	yes	yes
lumped-parameter number-known-components	n/a	low	very-high
structure continuous-volume	yes	yes	no
continuous-volume finite-elements	yes	no	n/a
continuous-volume grid	no	yes	n/a
structure number-of-elements	low	very-high	high
structure substructuring-capability	yes	yes	no
continuous-volume number-of-known-shapes	low	medium	n/a
shape-limitations essentially-none	no	yes	n/a
shape-limitations thin-shells	no	no	n/a
shape-limitations shells-of-revol	no	no	n/a
shape-limitations slabs	yes	no	n/a
shape-limitations thick-shells	no	no	n/a
fluids turbulence	n/a	yes	no
fluids shock-waves	n/a	yes	no

USER INTERFACE

user-interface text-editor	no	no	no
user-interface data-manipulator	no	no	no
user-interface control-language	yes	yes	yes
user-interface user-specified-routines	no	no	yes
user-interface matrix-spec-language	no	no	no
user-interface libraries	no	no	yes
libraries number-of-stored-shapes	very-low	low	very-low
libraries format-conversion	no	no	no
libraries geometric-conversion	no	no	no
libraries translate-for-other-progs	no	no	no

GRAPHICS

graphics present	yes	no	no
graphics 3D-structure	yes	n/a	n/a
graphics number-of-formats	low	n/a	n/a
graphics perspective	no	n/a	n/a
graphics hidden-line-removal	no	n/a	n/a
graphics shading	no	n/a	n/a

OTHER

commercialization many-projects	yes	yes	yes
commercialization many-companies	yes	yes	yes
commercialization industry-standard	no	yes	yes
commercialization many-upgrades	no	yes	yes
commercialization many-computers	yes	yes	yes

Program	vicar	jpldis
TASK		
Task Database-program	yes	yes
Database-program video-images	yes	no
Task Engineering-Program	yes	no
Engineering-Program Structural-engineering	no	n/a
Engineering-program aerodynamics	no	n/a
Engineering-Program electronic-circuit	no	n/a
Engineering-Program Thermodynamics	no	n/a
Thermodynamics radiation	n/a	n/a
Thermodynamics conduction	n/a	n/a
Engineering-Program Image-processing	yes	n/a
METHODS		
methods number-redundant-choices	very-high	very-low
time-dependence static	n/a	n/a
time-dependence quasi-static	n/a	n/a
time-dependence fully-dynamic	n/a	n/a
linearity linear	n/a	n/a
linearity limited-non-linear	n/a	n/a
linearity fully-non-linear	n/a	n/a
boundary number-ways-to-specify	n/a	n/a
boundary 2D-field-on-3D-surface	n/a	n/a
boundary lumped-component-values	n/a	n/a
OBJECTS & REPRESENTATIONS		
objects non-physical	yes	yes
non-physical images	yes	no
objects solids	no	no
objects fluids	no	no
solids structure	n/a	n/a
solids point-mass	n/a	n/a
structure spatial-dimension	n/a	n/a
structure lumped-parameter	n/a	n/a
structure continuous-volume	n/a	n/a
continuous-volume finite-elements	n/a	n/a
continuous-volume grid	n/a	n/a
structure number-of-elements	n/a	n/a
structure substructuring-capability	n/a	n/a
continuous-volume number-of-known-shapes	n/a	n/a
shape-limitations essentially-none	n/a	n/a
shape-limitations thin-shells	n/a	n/a
shape-limitations shells-of-revol	n/a	n/a
shape-limitations slabs	n/a	n/a
shape-limitations thick-shells	n/a	n/a
fluids turbulence	n/a	n/a
fluids shock-waves	n/a	n/a

USER INTERFACE

user-interface text-editor	yes	no
user-interface data-manipulator	yes	no
user-interface control-language	yes	yes
user-interface user-specified-routines	yes	yes
user-interface matrix-spec-language	no	no
user-interface libraries	yes	no
libraries number-of-stored-shapes	high	n/a
libraries format-conversion	yes	yes
libraries geometric-conversion	yes	no
libraries translate-for-other-progs	yes	no

GRAPHICS

graphics present	yes	no
graphics 3D-structure	no	n/a
graphics number-of-formats	very-high	n/a
graphics perspective	n/a	n/a
graphics hidden-line-removal	n/a	n/a
graphics shading	n/a	n/a

OTHER

commercialization many-projects	yes	yes
commercialization many-companies	yes	yes
commercialization industry-standard	no	no
commercialization many-upgrades	no	yes
commercialization many-computers	no	no
systems-programming execute-commands	yes	yes
systems-programming code-to-save-memory	yes	no

Program	nexus	care	rim	afile
TASK				
Task Database-program	yes	no	yes	no
Database-program video-images	no	no	no	no
Task Engineering-Program	yes	yes	no	yes
Engineering-Program Structural-engineering	yes	no	no	yes
Engineering-program aerodynamics	no	no	no	no
Engineering-Program electronic-circuit	no	no	no	no
Engineering-Program Thermodynamics	no	no	no	no
Thermodynamics radiation	no	no	no	no
Thermodynamics conduction	no	no	no	no
Engineering-Program Image-processing	no	no	no	no
METHODS				
methods number-redundant-choices	very-low	medium	very-low	very-low
time-dependence static	no	yes	no	yes
time-dependence quasi-static	no	yes	no	no
time-dependence fully-dynamic	no	no	no	no
linearity linear	no	no	no	no
linearity limited-non-linear	no	no	no	no
linearity fully-non-linear	no	no	no	no
boundary number-ways-to-specify	low	very-low	very-low	low
boundary 2D-field-on-3D-surface	yes	no	no	no
boundary lumped-component-values	yes	no	no	no
OBJECTS & REPRESENTATIONS				
objects non-physical	yes	yes	yes	no
non-physical images	yes	no	no	no
objects solids	yes	no	no	yes
objects fluids	no	no	no	no
solids structure	yes	no	no	yes
solids point-mass	no	no	no	no
structure spatial-dimension	3	0	0	3
structure lumped-parameter	no	no	no	yes
lumped-parameter number-known-components	very-low	very-low	very-low	medium
structure continuous-volume	yes	no	no	yes
continuous-volume finite-elements	yes	no	no	no
continuous-volume grid	yes	no	no	no
structure number-of-elements	very-high	very-low	very-low	low
structure substructuring-capability	yes	no	no	no
continuous-volume number-of-known-shapes	medium	very-low	very-low	very-low
shape-limitations essentially-none	yes	no	no	no
shape-limitations thin-shells	no	no	no	no
shape-limitations shells-of-revol	no	no	no	no
shape-limitations slabs	no	no	no	no
shape-limitations thick-shells	no	no	no	no
fluids turbulence	no	no	no	no
fluids shock-waves	no	no	no	no

USER INTERFACE

user-interface text-editor	yes	no	no	no
user-interface data-manipulator	yes	yes	yes	yes
user-interface control-language	yes	no	yes	yes
user-interface user-specified-routines	yes	no	yes	no
user-interface matrix-spec-language	no	no	no	no
user-interface libraries	yes	no	no	yes
libraries number-of-stored-shapes	medium	very-low	very-low	medium
libraries format-conversion	yes	no	no	no
libraries geometric-conversion	yes	no	no	yes
libraries translate-for-other-progs	yes	no	no	no

GRAPHICS

graphics present	yes	yes	no	yes
graphics 3D-structure	yes	no	no	yes
graphics number-of-formats	very-high	very-low	very-low	low
graphics perspective	yes	no	no	no
graphics hidden-line-removal	yes	no	no	no
graphics shading	yes	no	no	no

OTHER

commercialization many-projects	yes	yes	yes	yes
commercialization many-companies	yes	yes	yes	no
commercialization industry-standard	no	no	no	no
commercialization many-upgrades	no	no	yes	no
commercialization many-computers	no	yes	yes	no
systems-programming execute-commands	yes	no	no	no
systems-programming code-to-save-memory	no	no	no	no

Program	fpt	sap	discos	dids
TASK				
Task Database-program	no	no	no	yes
Database-program video-images	no	no	no	yes
Task Engineering-Program	no	no	yes	no
Engineering-Program Structural-engineering	no	no	no	no
Engineering-program aerodynamics	no	no	no	no
Engineering-Program electronic-circuit	no	no	no	no
Engineering-Program Thermodynamics	no	no	no	no
Engineering-Program Control-systems	no	no	yes	no
Thermodynamics radiation	no	no	no	no
Thermodynamics conduction	no	no	no	no
Engineering-Program Image-processing	no	no	no	no
METHODS				
methods number-redundant-choices	very-low	very-low	very-low	very-low
time-dependence static	no	no	no	no
time-dependence quasi-static	no	no	no	no
time-dependence fully-dynamic	no	no	yes	no
linearity linear	no	no	no	no
linearity limited-non-linear	no	no	no	no
linearity fully-non-linear	no	no	yes	no
boundary number-ways-to-specify	very-low	very-low	very-low	very-low
boundary 2D-field-on-3D-surface	no	no	no	no
boundary lumped-component-values	no	no	yes	no
OBJECTS & REPRESENTATIONS				
objects non-physical	yes	yes	no	yes
non-physical images	no	no	no	yes
objects solids	no	no	yes	no
objects fluids	no	no	no	no
solids structure	no	no	yes	no
solids point-mass	no	no	no	no
structure spatial-dimension	0	0	3	0
structure lumped-parameter	no	no	yes	no
lumped-parameter number-known-components	very-low	very-low	medium	very-low
structure continuous-volume	no	no	no	no
continuous-volume finite-elements	no	no	no	no
continuous-volume grid	no	no	no	no
structure number-of-elements	very-low	very-low	low	very-low
structure substructuring-capability	no	no	no	no
continuous-volume number-of-known-shapes	very-low	very-low	very-low	very-low
shape-limitations essentially-none	no	no	yes	no
shape-limitations thin-shells	no	no	no	no
shape-limitations shells-of-revol	no	no	no	no
shape-limitations slabs	no	no	no	no
shape-limitations thick-shells	no	no	no	no
fluids turbulence	no	no	no	no
fluids shock-waves	no	no	no	no

USER INTERFACE

user-interface text-editor	yes	no	no	yes
user-interface data-manipulator	no	no	no	yes
user-interface control-language	no	no	no	no
user-interface user-specified-routines	yes	no	yes	no
user-interface matrix-spec-language	no	no	no	no
user-interface libraries	yes	yes	no	yes
libraries number-of-stored-shapes	very-low	very-low	medium	very-high
libraries format-conversion	yes	no	yes	yes
libraries geometric-conversion	no	no	no	no
libraries translate-for-other-progs	no	no	yes	no

GRAPHICS

graphics present	no	no	yes	yes
graphics 3D-structure	no	no	no	no
graphics number-of-formats	very-low	very-low	very-low	very-high
graphics perspective	no	no	no	no
graphics hidden-line-removal	no	no	no	no
graphics shading	no	no	no	no

OTHER

commercialization many-projects	yes	yes	yes	no
commercialization many-companies	yes	yes	yes	yes
commercialization industry-standard	no	no	no	no
commercialization many-upgrades	no	no	yes	no
commercialization many-computers	no	yes	yes	no
systems-programming execute-commands	no	no	no	yes
systems-programming code-to-save-memory	no	no	no	yes

Program	samsan	spar	dyloflex	flexstab
TASK				
Task Database-program	no	yes	no	no
Database-program video-images	no	no	no	no
Task Engineering-Program	yes	yes	yes	yes
Engineering-Program Structural-engineering	no	yes	yes	yes
Engineering-program aerodynamics	no	no	yes	yes
Engineering-Program electronic-circuit	no	no	no	no
Engineering-Program Thermodynamics	no	yes	no	no
Engineering-Program Control-systems	yes	no	yes	yes
Thermodynamics radiation	no	no	no	no
Thermodynamics conduction	no	yes	no	no
Engineering-Program Image-processing	no	no	no	no
METHODS				
methods number-redundant-choices	very-low	medium	very-low	very-low
time-dependence static	no	yes	no	no
time-dependence quasi-static	yes	yes	yes	yes
time-dependence fully-dynamic	no	no	no	no
linearity linear	yes	yes	no	no
linearity limited-non-linear	no	yes	no	yes
linearity fully-non-linear	no	no	yes	no
boundary number-ways-to-specify	very-low	very-low	medium	medium
boundary 2D-field-on-3D-surface	no	yes	yes	yes
boundary lumped-component-values	no	no	yes	yes
OBJECTS & REPRESENTATIONS				
objects non-physical	yes	no	no	no
non-physical images	no	no	no	no
objects solids	no	yes	yes	yes
objects fluids	no	yes	yes	yes
solids structure	no	yes	yes	yes
solids point-mass	no	no	yes	no
structure spatial-dimension	0	3	3	3
structure lumped-parameter	no	no	yes	no
lumped-parameter number-known-components	very-low	very-low	low	very-low
structure continuous-volume	no	yes	yes	yes
continuous-volume finite-elements	no	yes	no	no
continuous-volume grid	no	no	yes	yes
structure number-of-elements	very-low	very-high	low	very-high
structure substructuring-capability	no	yes	yes	no
continuous-volume number-of-known-shapes	very-low	high	medium	medium
shape-limitations essentially-none	no	yes	yes	yes
shape-limitations thin-shells	no	no	no	no
shape-limitations shells-of-revol	no	no	no	no
shape-limitations slabs	no	no	no	no
shape-limitations thick-shells	no	no	no	no
fluids turbulence	no	no	yes	yes
fluids shock-waves	no	no	yes	no

USER INTERFACE

user-interface text-editor	no	no	no	no
user-interface data-manipulator	yes	yes	no	no
user-interface control-language	no	yes	no	yes
user-interface user-specified-routines	yes	no	no	yes
user-interface matrix-spec-language	yes	no	no	no
user-interface libraries	yes	yes	no	yes
libraries number-of-stored-shapes	very-low	medium	medium	medium
libraries format-conversion	no	yes	yes	no
libraries geometric-conversion	no	no	yes	no
libraries translate-for-other-progs	yes	no	yes	no

GRAPHICS

graphics present	no	yes	no	yes
graphics 3D-structure	no	yes	no	no
graphics number-of-formats	very-low	very-low	very-low	low
graphics perspective	no	yes	no	no
graphics hidden-line-removal	no	no	no	no
graphics shading	no	no	no	no

OTHER

commercialization many-projects	yes	yes	yes	yes
commercialization many-companies	yes	yes	yes	yes
commercialization industry-standard	no	no	no	no
commercialization many-upgrades	yes	yes	no	yes
commercialization many-computers	no	yes	no	yes
systems-programming execute-commands	no	no	no	no
systems-programming code-to-save-memory	no	no	no	yes

end

Program

TASK

Task Database-program
Database-program video-images
Task Engineering-Program
Engineering-Program Structural-engineering
Engineering-program aerodynamics
Engineering-Program electronic-circuit
Engineering-Program Thermodynamics
Thermodynamics radiation
Thermodynamics conduction
Engineering-Program Image-processing

METHODS

methods number-redundant-choices
time-dependence static
time-dependence quasi-static
time-dependence fully-dynamic
linearity linear
linearity limited-non-linear
linearity fully-non-linear
boundary number-ways-to-specify
boundary 2D-field-on-3D-surface
boundary lumped-component-values

OBJECTS & REPRESENTATIONS

objects non-physical
non-physical images
objects solids
objects fluids
solids structure
solids point-mass
structure spatial-dimension
structure lumped-parameter
lumped-parameter number-known-components
structure continuous-volume
continuous-volume finite-elements
continuous-volume grid
structure number-of-elements
structure substructuring-capability
continuous-volume number-of-known-shapes
shape-limitations essentially-none
shape-limitations thin-shells
shape-limitations shells-of-revol
shape-limitations slabs
shape-limitations thick-shells
fluids turbulence
fluids shock-waves

USER INTERFACE

user-interface text-editor
user-interface data-manipulator
user-interface control-language
user-interface user-specified-routines
user-interface matrix-spec-language
user-interface libraries
libraries number-of-stored-shapes
libraries format-conversion
libraries geometric-conversion
libraries translate-for-other-progs

GRAPHICS

graphics present
graphics 3D-structure
graphics number-of-formats
graphics perspective
graphics hidden-line-removal
graphics shading

OTHER

commercialization many-projects
commercialization many-companies
commercialization industry-standard
commercialization many-upgrades
commercialization many-computers
systems-programming execute-commands
systems-programming code-to-save-memory

hd:nasa:knowledge acquisition system:hand-tuned-specs.lisp

Program	big-eng	science-only
TASK		
Task Engineering-Program	yes	yes
Engineering-Program Structural-engineering	yes	no
Engineering-Program electronic-circuit	no	no
Engineering-Program Thermodynamics	yes	no
Engineering-Program Control-systems	yes	no
Thermodynamics radiation	yes	no
Thermodynamics conduction	yes	no
METHODS		
methods number-redundant-choices	very-high	very-low
time-dependence static	yes	no
time-dependence quasi-static	yes	no
time-dependence fully-dynamic	yes	yes
linearity linear	yes	no
linearity limited-non-linear	yes	no
linearity fully-non-linear	yes	yes
boundary number-ways-to-specify	very-high	low
boundary 2D-field-on-3D-surface	yes	no
boundary lumped-component-values	yes	yes
OBJECTS-&-REPRESENTATIONS		
objects solids	yes	yes
objects fluids	yes	no
solids structure	yes	yes
solids point-mass	yes	no
structure spatial-dimension	3	3
structure lumped-parameter	no	yes
lumped-parameter number-known-components	very-low	medium
structure continuous-volume	yes	no
continuous-volume finite-elements	yes	no
continuous-volume grid	no	no
structure number-of-elements	very-high	low
structure substructuring-capability	yes	no
continuous-volume number-of-known-shapes	very-high	very-low
shape-limitations essentially-none	yes	yes
shape-limitations thin-shells	yes	no
shape-limitations shells-of-revol	yes	no
shape-limitations slabs	yes	no
shape-limitations thick-shells	yes	no
fluids turbulence	no	no
fluids shock-waves	no	no

USER INTERFACE

user-interface text-editor	yes	no
user-interface data-manipulator	yes	no
user-interface control-language	yes	no
user-interface user-specified-routines	yes	no
user-interface matrix-spec-language	yes	no
user-interface libraries	yes	no
libraries number-of-stored-shapes	very-high	low
libraries format-conversion	yes	no
libraries geometric-conversion	yes	no
libraries translate-for-other-progs	yes	no

GRAPHICS

graphics present	yes	no
graphics 3D-structure	yes	no
graphics number-of-formats	very-high	very-low
graphics perspective	yes	no
graphics hidden-line-removal	yes	no
graphics shading	yes	no

OTHER

commercialization many-projects	yes	no
commercialization many-companies	yes	no
commercialization industry-standard	yes	no
commercialization many-upgrades	yes	no
commercialization many-computers	yes	no

end

hd:nasa:knowledge acquisition system:generic database

Program	Entity	Attribute	Value	Reason
panair	size	total	289	Cosmic catalogue
	task	complexity	very-high	aerodynamics: full sub and supersonic flows, restricted only by linear potential flow theory
	methods	complexity	very-high	Higher Order Panel Method
		generality	very-high	full sub and supersonic flows, restricted only by linear potential flow theory
		accuracy	very-high	Higher order panel method
		efficiency	high	****unknown****
		redundancy	very-low	****unknown****
	objects	complexity	very-high	handles arbitrary structures
		generality	very-high	handles arbitrary structures
	representation	complexity	high	set of surface grids
		capacity	high	assumed from context
		knowledge	high	good set of input options
	graphics	complexity	very-low	not mentioned
	user-interface	complexity	high	user has extensive control of input and output options, batch
	data-management	complexity	high	components communcate only through database of info. on disk: Systematic Software Development Methodology
pilot	size	total	47	Cosmic catalogue
	task	complexity	medium	experimental, proof of concept code for higher order panel mehtod in aerodynamics: full sub and supersonic flows
	methods	complexity	very-high	higher order panel method
		generality	very-high	full sub and supersonic flows, restricted only by linear potential flow theory
		accuracy	very-high	higher order panel method
		efficiency	medium	****unknown****
		redundancy	very-low	****unknown****
	objects	complexity	very-high	handles arbitrary structures
		generality	very-high	handles arbitrary structures
	representation	complexity	high	set of surface grids
		capacity	low	research program, not user oriented
		knowledge	very-low	research program, not user oriented
	graphics	complexity	very-low	not mentioned
	user-interface	complexity	very-low	research program, not user oriented
	data-management	complexity	very-low	research program, not user oriented
ussaero	size	total	21	Cosmic catalogue
	task	complexity	medium	Approximate sub and supersonic aerodynamics on restricted set of aircraft
	methods	complexity	high	Panel method
		generality	medium	panel method
		accuracy	medium	****default****
		efficiency	medium	****default****
		redundancy	very-low	****default****
	objects	complexity	low	restricted aircraft
		generality	very-low	restricted aircraft
	representation	complexity	medium	set of surface grids
		capacity	medium	****default****
		knowledge	very-low	few options
	graphics	complexity	low	one version can use DI-3000 grapics package
	user-interface	complexity	very-low	not mentioned
	data-management	complexity	very-low	****default****
	sys-programming	complexity	very-low	****default****
	hard-architecture	complexity	very-low	****default****
dyloflex	size	total	124	Cosmic catalogue
	task	complexity	very-high	aerodynamics: flexible aircraft with active controls
	methods	complexity	very-high	aircraft and control
		generality	high	can't do supersonic flow with unsteady aircraft
		accuracy	high	panel method
		efficiency	medium	****unknown****
		redundancy	medium	make up of eight individual programs
	objects	complexity	very-high	aircraft
		generality	high	not arbitrary
	representation	complexity	very-high	set of surface grids (aircraft) and control surface
		capacity	high	up to 70 modes, 70 degrees of freedom, and 400 aerodynamic singularities
		knowledge	high	can use output from structural programs
	graphics	complexity	very-low	not mentioned
	user-interface	complexity	low	control eight programs, batch execution
	data-management	complexity	high	handles many types in input data including output from struct. programs, manages files from intermeadiate results of 8 program system
eom-dylo	size	total	40	Cosmic catalogue

ORIGINAL PAGE IS
OF POOR QUALITY

	task	complexity	medium	part of dyloflex: compute equations of motion matrices
	methods	complexity	medium	Langrangian energy approach
		generality	low	needs aerodynamic and structural data as input
		accuracy	high	part of dyloflex
		efficiency	medium	****unknown****
		redundancy	very-low	one component of another system
	objects	complexity	medium	works with matrix representations
		generality	medium	specialized component
	representation	complexity	medium	matrices
		capacity	high	100 dof, 400 aero. singularities, 35 gust zones, 20 reduced frequencies
		knowledge	medium	can use output from structural programs
	graphics	complexity	very-low	not mentioned
	user-interface	complexity	very-low	single component of dyloflex
	data-management	complexity	very-low	part of another system
corel	size	total	20	Cosmic catalogue
	task	complexity	low	approximate aerodynamics for supersonic wings
	methods	complexity	medium	panel method on highly restricted shapes
		generality	low	many restrictions and approximations
		accuracy	low	many restrictions and approximations
		efficiency	medium	****default****
		redundancy	low	at least one choice on similar calculation methods
	objects	complexity	low	wings with correction for idealized body
		generality	low	many restrictions
	representation	complexity	high	set of grids with refined mesh
		capacity	medium	****default****
		knowledge	low	a few options on shapes and calculation methods
	graphics	complexity	very-low	****default****
	user-interface	complexity	very-low	batch, no features mentioned
	data-management	complexity	very-low	****default****
	sys-programming	complexity	very-low	****default****
	hard-architecture	complexity	very-low	****default****
flexstab	size	total	100	Cosmic catalogue
	task	complexity	very-high	aerodynamics: predict stability and control characteristics of elastic airplane
	methods	complexity	medium	linear perturbation theory
		generality	medium	does sub and supersonic flow
		accuracy	medium	linear theory
		efficiency	high	procedures for both large and small problems
		redundancy	very-low	not mentioned
	objects	complexity	high	aircraft
		generality	medium	shapes seem somewhat restricted
	representation	complexity	high	surface grids
		capacity	very-high	not software limited
		knowledge	high	can interface with structural mech. programs
	graphics	complexity	medium	has plot capability
	user-interface	complexity	low	not discussed
	data-management	complexity	low	I/O files and some "out-of-core" matrix calculations
rhoiv	size	total	26	Cosmic catalogue
	task	complexity	medium	predict unsteady motion loadings on lifting surfaces caused by control surfaces
	methods	complexity	medium	kernel function method, subsonic only
		generality	low	significant restrictions on geometry and flow
		accuracy	high	good comparison with theory
		efficiency	medium	****unknown****
		redundancy	very-low	****unknown****
	objects	complexity	medium	lifting surfaces with leading and trailing edge control surfaces
		generality	low	lifting surfaces with leading and trailing edge control surfaces
	representation	complexity	low	determined by a handful of line segments and options
		capacity	low	representation too simple to include much detail
		knowledge	low	a few options and defaults included
	graphics	complexity	very-low	not mentioned
	user-interface	complexity	very-low	batch execution, details not given
	data-management	complexity	very-low	nothing mentioned
gtds	size	total	165	Cosmic catalogue
	task	complexity	very-high	Astrodyamics: mission support; communication, command, and tracking; navigation; orbital mechanics
	methods	complexity	very-high	complex filtering, data simulation, sophisticated data management and data and error analysis
		generality	very-high	completely covers domain, either operational or R&D tool
		accuracy	very-high	includes high-precision Cowell-type numerical integration
		efficiency	high	near real time support in interactive mode
		redundancy	high	8 programs, several orbital models, 4 filters in navigation

	objects	complexity	medium	predicted and actual trajectories and error analysis
		generality	high	completely covers domain with relatively simple representation
	representation	complexity	high	state vectors and complex adaptive filters
		capacity	very-high	from context
		knowledge	high	4 complex adaptive filters and error analysis
	graphics	complexity	very-low	no graphic output; interactive mode uses graphic screen
	user-interface	complexity	medium	batch or interactive; complex data management; control 8 programs; graphic screen for interactive use
	data-management	complexity	high	1 program for data mgt.; on-line database files including reports
radmas	size	total	63	Cosmic catalogue
	task	complexity	medium	R&D tool for mission analysis
	methods	complexity	medium	set of tools and independent cpu tasks for problem; automatic error recovery
		generality	very-high	user supplied "automatic sequence" in special high level language
		accuracy	medium	****unknown****
		efficiency	medium	****unknown****
		redundancy	very-low	****unknown****
	objects	complexity	medium	predicted and actual trajectories and error analysis
		generality	high	completely covers domain with relatively simple representation
	representation	complexity	high	trajectories and "automatic sequence"
		capacity	very-high	dynamically allocated arrays
		knowledge	medium	orbital mechanics, targeting, optimization, Monte Carlo error anal.
	graphics	complexity	medium	has graphics
	user-interface	complexity	high	user can control and redirect system in interactive mode (also has batch)
	data-management	complexity	medium	provides flexible data management and control structure
	sys-programming	complexity	low	set of tools and independent cpu tasks for problem; automatic error recovery
fsd	size	total	31	Cosmic catalogue
	task	complexity	medium	astrodynamics: dynamics of flexible spacecraft with control system
	methods	complexity	medium	orbital mechanics, control systems, flexible body dyn
		generality	low	versatile within its narrow domain, uses series of shape functions for flexible elements
		accuracy	high	used by NASA in 6 space programs
		efficiency	medium	****unknown****
		redundancy	very-low	not mentioned
	objects	complexity	high	spacecraft with active control systems
		generality	very-low	specific to narrow domain
	representation	complexity	medium	network of simple shapes, orbit and control characteristics
		capacity	very-low	up to 10 tubular elements in arbitrary orientation
		knowledge	medium	simple knowledge in earth orbits, control, and simple physical shapes
	graphics	complexity	very-low	not mentioned
	user-interface	complexity	very-low	batch, free format namelist construction
	data-management	complexity	very-low	nothing mentioned
post3d	size	total	30	Cosmic catalogue
	task	complexity	medium	astrodynamics: optimize and target spacecraft with control system
	methods	complexity	medium	orbit, optimization, control with 3 DOF
		generality	low	only 3 DOF (point mass) structures
		accuracy	medium	****unknown****
		efficiency	medium	****unknown****
		redundancy	very-low	****unknown****
	objects	complexity	very-high	spacecraft with active control systems
		generality	very-low	specific to narrow domain
	representation	complexity	low	point mass structure, aerodynamics, and guidance; planet model
		capacity	very-low	from context
		knowledge	medium	optimization, mechanics, control
	graphics	complexity	very-low	not mentioned
	user-interface	complexity	very-low	batch, free format namelist construction
	data-management	complexity	very-low	nothing mentioned
post6d	size	total	26	Cosmic catalogue
	task	complexity	medium	astrodynamics: optimize and target spacecraft with control system
	methods	complexity	medium	orbit, optimization, control with 6 DOF, rigid body
		generality	low	rigid body space craft only
		accuracy	medium	****unknown****
		efficiency	medium	****unknown****
		redundancy	very-low	****unknown****
	objects	complexity	very-high	spacecraft with active control systems
		generality	very-low	specific to narrow domain
	representation	complexity	low	rigid body structure, aerodynamics, and guidance; planet model
		capacity	very-low	from context
		knowledge	medium	optimization, mechanics, control
	graphics	complexity	very-low	not mentioned
	user-interface	complexity	very-low	batch, free format namelist construction

ORIGINAL PAGE IS
OF POOR QUALITY

	data-management	complexity	very-low	nothing mentioned
sort	size	total	46	Cosmic catalogue
	task	complexity	medium	astrodynamics: optimize missile, rocket, and spacecraft with control system
	methods	complexity	medium	optimization, control with 3 DOF
		generality	low	only 3 DOF
		accuracy	high	used on Space Shuttle
		efficiency	medium	****unknown****
		redundancy	medium	4 optimization techniques, 6 table types, 3 interpolation options
	objects	complexity	very-high	spacecraft with active control systems
		generality	very-low	specific to narrow domain
	representation	complexity	low	rigid body structure, aerodynamics, and guidance; planet model
		capacity	low	representation is simple so high capacity isn't needed
		knowledge	high	a lot of info on flight activities and vehicle characteristics; optimization, mechanics, control
	graphics	complexity	low	trajectory plot file generation in two formats
	user-interface	complexity	very-low	batch, free format namelist construction
	data-management	complexity	very-low	nothing mentioned
sinda	size	total	40	Cosmic catalogue (also 33 and 37 for other versions)
	task	complexity	medium	thermodynamics: without implicitly doing radiation
	methods	complexity	medium	general thermodynamics or similar work
		generality	very-high	can be used for any network based task with compatible mathematics
		accuracy	medium	****unknown****
		efficiency	medium	****unknown****
		redundancy	very-high	large choice of methods
	objects	complexity	medium	general RC network but no 3D geometry
		generality	high	general including fluid elements
	representation	complexity	medium	lumped parameter network
		capacity	medium	****unknown****
		knowledge	high	good knowledge of thermodynamic properties
	graphics	complexity	very-low	not mentioned
	user-interface	complexity	medium	user writes program in SINDA language calling Sinda library subroutines, his program is compiled into Fortran
	data-management	complexity	very-low	nothing mentioned
trasys	size	total	60	Cosmic catalogue
	task	complexity	high	thermodynamics: radiation analysis
	methods	complexity	medium	radiation analysis
		generality	very-high	anything representable as an RC network
		accuracy	medium	****unknown****
		efficiency	medium	****unknown****
		redundancy	high	choice of methods
	objects	complexity	high	general
		generality	high	general
	representation	complexity	medium	lumped parameter network
		capacity	medium	up to 1000 nodes
		knowledge	high	preprocessor to convert user's geometry, editor
	graphics	complexity	medium	requires DISSPLA plot package
	user-interface	complexity	high	editor, preprocessor, link user routines, batch w. interactive plots
	data-management	complexity	very-low	nothing mentioned
view	size	total	34	Cosmic catalogue
	task	complexity	medium	thermodynamics: radiation analysis; produce view factors only
	methods	complexity	medium	view factors only, using contour integration and double area integral techniques
		generality	high	anything representable surfaces
		accuracy	medium	****unknown****
		efficiency	medium	****unknown****
		redundancy	very-low	not mentioned
	objects	complexity	high	3D surfaces (can overlap)
		generality	medium	3D surfaces
	representation	complexity	medium	collection of flat surfaces (triangle or quadrilateral) in 3D
		capacity	medium	****unknown****
		knowledge	medium	some knowledge of solar radiation and 3D geometry
	graphics	complexity	high	interactive 3D graphics with hidden line removal
	user-interface	complexity	medium	interactive graphics and simple input manipulation
	data-management	complexity	very-low	nothing mentioned
necap	size	total	25	Cosmic catalogue
	task	complexity	low	energy usage predictions for buildings: thermal loading, energy concumption, heating/cooling system simulation, cost analysis
	methods	complexity	medium	simple model of complex process
		generality	low	only for typical heating systems in typical buildings
		accuracy	low	a lot of simplifying assumptions
		efficiency	medium	****default****

		redundancy	very-low	****default****
	objects	complexity	medium	simple representation of buildings and heating/cooling systems
		generality	low	typical instances are covered
	representation	complexity	medium	lumped parameter
		capacity	medium	****default****
		knowledge	high	13 types of heating systems and buildings with windows, people, lighting, insulation, etc
	graphics	complexity	very-low	****default****
	user-interface	complexity	medium	has input processor
	data-management	complexity	low	comes with weather data from 62 cities
	sys-programming	complexity	very-low	****default****
	hard-architecture	complexity	very-low	****default****
sspta	size	total	13.9	Cosmic catalogue
	task	complexity	medium	model thermodynamics of space shuttle
	methods	complexity	low	lumped parameter model with quasistatic conditions
		generality	low	radiation but no conduction
		accuracy	medium	****default****
		efficiency	medium	****default****
		redundancy	very-low	medium, 2 or 3 options in temperature calculation
	objects	complexity	medium	lumped parameter model of 3D surfaces
		generality	low	designed for shuttle but ok for other shapes if not needed in too much detail
	representation	complexity	medium	macroscopic surfaces with links parameteized rather than derrived form dynamics
		capacity	low	order of a few dozen shapes
		knowledge	low	knows a little geometery
	graphics	complexity	medium	wireframe 3D plots
	user-interface	complexity	low	batch input with commands and parameters, fixed format
	data-management	complexity	low	about input/output files to coordinate 6 models
	sys-programming	complexity	very-low	****default****
	hard-architecture	complexity	very-low	****default****
nastran	size	total	320	Cosmic catalogue
	task	complexity	very-high	structural-mech: general purpose structural analysis
	methods	complexity	very-high	finite element, displacement method: comprehensive treatment
		generality	very-high	very wide range of applications
		accuracy	very-high	its an industry standard
		efficiency	high	its an industry standard
		redundancy	very-high	was pieced together over many years
	objects	complexity	very-high	general
		generality	very-high	essentially arbitrary
	representation	complexity	very-high	volume of finite elements
		capacity	very-high	assumed from context
		knowledge	very-high	wide variety of options
	graphics	complexity	very-high	3D with hidden line removal, regular & deformed structures, many options
	user-interface	complexity	very-high	control many options, link user capabilities w matrix abstraction language
	data-management	complexity	medium	from context
spar	size	total	87	Cosmic catalogue
	task	complexity	very-high	structural-mech: Analysis and redesign
	methods	complexity	high	includes thermal analyzer
		generality	high	includes buckling and thermal
		accuracy	medium	****unknown****
		efficiency	medium	****unknown****
		redundancy	very-low	****unknown****
	objects	complexity	very-high	general
		generality	very-high	general
	representation	complexity	very-high	volume of finite elements
		capacity	very-high	> 50,000 elements
		knowledge	very-high	network generators and library
	graphics	complexity	medium	standard graphics features
	user-interface	complexity	high	batch or interactive, significant user control and flexibility
	data-management	complexity	low	25 communicate through unified database
stagsc-1	size	total	73	Cosmic catalogue : (26 Univac, 25 Vax, 73 CDC)
	task	complexity	high	structural-mech: general shells
	methods	complexity	high	skyline and other methods, bifurcation buckling
		generality	high	designed for general shells
		accuracy	high	exact solutions to non-linear equations
		efficiency	high	takes advantage of zeros in matrix
		redundancy	very-low	no redundant features mentioned
	objects	complexity	medium	general shells and some other shapes
		generality	medium	general shells and some other shapes
	representation	complexity	medium	general shells, finite elements

		knowledge	medium	not many shapes
	graphics	complexity	very-low	not mentioned
	user-interface	complexity	low	batch execution, no features mentioned
	data-management	complexity	very-low	nothing mentioned
stars2s	size	total	25	Cosmic catalogue
	task	complexity	high	structural-mech: shells of revolution
	methods	complexity	medium	Love-Reissner first order shell theory
		generality	medium	linear for nonsymmetrically loaded shells and nonlinear for symmetrically loaded shells only
		accuracy	medium	****unknown****
		efficiency	medium	****unknown****
		redundancy	very-low	no redundant features mentioned
	objects	complexity	low	shells of revolution
		generality	low	shells of revolution
	representation	complexity	low	shell elements, details not specified
		knowledge	low	no mention of element library
	graphics	complexity	very-low	not mentioned
	user-interface	complexity	low	batch execution, no features mentioned
	data-management	complexity	very-low	nothing mentioned
plans	size	total	53	Cosmic catalogue
	task	complexity	high	structural-mech: variety of nonlinear structures
	methods	complexity	high	some restrictions on nonlinear effects on general structures
		generality	medium	structure restrictions
		accuracy	medium	estimated from context
		efficiency	medium	****unknown****
		redundancy	high	set of 5 programs with overlap
	objects	complexity	high	general shapes with some restrictions
		generality	high	general shapes with some restrictions
	representation	complexity	medium	fair selection of element shapes
		capacity	low	estimated from context
		knowledge	high	preprocessor: debug and resequence
	graphics	complexity	medium	plots of input structures
	user-interface	complexity	medium	preprocessor checks bugs, batch execution
	data-management	complexity	very-low	nothing mentioned
pasco	size	total	19.1	Cosmic catalogue
	task	complexity	medium	structural-mech: sizing and analyzing stiffened panels
	methods	complexity	medium	stress equations simplified due to restrictions
		generality	low	only prismatic structures
		accuracy	medium	****default****
		efficiency	medium	****default****
		redundancy	very-low	****default****
	objects	complexity	low	elements form panels
		generality	low	no variation alone one axis
	representation	complexity	medium	****default****
		capacity	medium	****default****
		knowledge	low	only rectangular plate elements
	graphics	complexity	low	only two plots of input only
	user-interface	complexity	very-low	fixed format batch input
	data-management	complexity	very-low	****default****
	sys-programming	complexity	very-low	****default****
	hard-architecture	complexity	very-low	****default****
nascap	size	total	45	Cosmic catalogue
	task	complexity	high	electromagnetics: spacecraft in charged particle environment, incl effect of spacecraft's charge on local environment
	methods	complexity	high	quasi static approximation using finite elements
		generality	high	general objects in plasma environment
		accuracy	medium	****unknown****
		efficiency	medium	****unknown****
		redundancy	very-low	not mentioned
	objects	complexity	very-high	general object in plasma environment
		generality	very-high	general object in plasma environment
	representation	complexity	high	geometrically, materially, and electrically complex objects
		capacity	medium	****unknown****
		knowledge	medium	geometry, material, and electrical properties
	graphics	complexity	very-low	none mentioned
	user-interface	complexity	very-low	batch
	data-management	complexity	very-low	not mentioned
sigmaii	size	total	32	Cosmic catalogue
	task	complexity	medium	electromagnetics: simulation and optimization of radiation shielding
	methods	complexity	medium	fluxes --> attenuation kernels --> radiation levels --> optimum shielding

ORIGINAL PAGE IS OF POOR QUALITY

		generality	low	needs to be modified for use in any radiation environment
		accuracy	medium	used on Mariner Jupiter/Saturn spacecraft
		efficiency	medium	****unknown****
		redundancy	very-low	not mentioned
	objects	complexity	medium	shielding geometry
		generality	low	needs to be modified for use in any radiation environment
	representation	complexity	medium	****unknown****
		capacity	medium	****unknown****
		knowledge	medium	geometry, flux, mass distribution, radiation transport
	graphics	complexity	low	has some plot capabilities
	user-interface	complexity	very-low	batch
	data-management	complexity	very-low	not mentioned
vicar	size	total	150	Cosmic catalogue
	task	complexity	very-high	combination image processing system and image database
	methods	complexity	medium	many image transformations and image handling tasks
		generality	very-high	covers two general topics
		accuracy	medium	****unknown****
		efficiency	high	special purpose I/O routines for efficient image handling
		redundancy	high	consists of two separate large systems put together
	objects	complexity	very-high	(possible overlayed and transformed) images with (possibly) associated tabular data
		generality	very-high	both gereral images, vectors, and tabular data
	representation	complexity	medium	vectors, matricies, and tables
		capacity	very-high	high volume application
		knowledge	high	wide variety of transformations and command interpreter for user
	graphics	complexity	high	image overlays (2 D)
	user-interface	complexity	high	interactive: command interpreter
	data-management	complexity	very-high	high volume database application
larsfris	size	total	60	Cosmic catalogue
	task	complexity	high	data analysis and pattern recognition for multispectral remote sensing data
	methods	complexity	medium	classic statistical and pattern classification
		generality	medium	works with any 2D image data
		accuracy	medium	****default****
		efficiency	medium	****default****
		redundancy	very-low	****default****
	objects	complexity	very-high	multispectral images and classification criteria
		generality	very-high	multispectral images and classification criteria
	representation	complexity	very-high	multispectral images and classification criteria
		capacity	medium	****default****
		knowledge	high	statistics and pattern recognition
	graphics	complexity	very-low	****default****
	user-interface	complexity	medium	batch with command interpreter
	data-management	complexity	medium	tape database
	sys-programming	complexity	very-low	****default****
	hard-architecture	complexity	very-low	****default****
dids	size	total	40	Cosmic catalogue
	task	complexity	low	database and color display for data geographically distributed over US
	methods	complexity	low	simple database and statistical operations
		generality	medium	any type of data can be displayed
		accuracy	medium	****default****
		efficiency	medium	built for speed
		redundancy	very-low	****default****
	objects	complexity	low	geographically distributed scalar data, histograms, color codes
		generality	low	suitable for many types of data
	representation	complexity	low	simple tables
		capacity	medium	****default****
		knowledge	medium	knowledge of US geography and statistical representations
	graphics	complexity	medium	color coded maps
	user-interface	complexity	medium	interactive color graphics, but simple manipulations
	data-management	complexity	medium	creation of simple image files
	sys-programming	complexity	medium	interacting processes, multiusers
	hard-architecture	complexity	low	host (archive, store, and retrieve) and satellite (retrieve only)
jpldis	size	total	39	Cosmic catalogue
	task	complexity	medium	unsophisticated gerneral database system
	methods	complexity	medium	small set of powerful commands: sort, merge, display, modify structure, create reports, etc.
		generality	very-high	high level task
		accuracy	very-high	this type of task is essentially perfectly accurate
		efficiency	medium	****unknown****
		redundancy	very-low	****unknown****

	objects	complexity	low	structrued records
		generality	very-high	very abstract objects
	representation	complexity	low	objects and representations are essentailly identical for this task
		capacity	high	implied by task, but no specifics given
		knowledge	low	representation too abstract for much knowledge to be used
	graphics	complexity	very-low	not mentioned or appropriate
	user-interface	complexity	high	command interpreter with macro capability
	data-management	complexity	high	central to task, but doesn't seem to be a sophisticated DB system
gempak	size	total	50	Cosmic catalogue
	task	complexity	medium	meteorology subroutine library, make met. maps
	methods	complexity	low	map and grid generation, simple data analysis
		generality	medium	covers much of weather forcasting, knows 3 input formats and 8 graphics devices
		accuracy	medium	****default****
		efficiency	medium	****default****
		redundancy	low	library of subroutines from different sources
	objects	complexity	low	2D vector and scalar fields
		generality	medium	weather parameters
	representation	complexity	high	grids and maps
		capacity	medium	****default****
		knowledge	high	geography for maps, and grid generators
	graphics	complexity	high	interactively generate maps with vector fields
	user-interface	complexity	high	interactive graphics, uses Transportable Applications Executive program for interface
	data-management	complexity	medium	handles a fair amount of vector field data
	sys-programming	complexity	very-low	simple sequential process
	hard-architecture	complexity	very-low	single user single cpu
dams	size	total	44	Cosmic catalogue
	task	complexity	high	detect and map surface water (or other resources) using Landsat multispectral scanner data
	methods	complexity	medium	uses spectral filter point by point
		generality	low	tied to Landsat
		accuracy	medium	****default****
		efficiency	medium	****default****
		redundancy	very-low	****default****
	objects	complexity	medium	maps, spectra, simple filters
		generality	medium	covers narrow domain
	representation	complexity	medium	details not given, seems relatively simple
		capacity	medium	handles map and spectral data
		knowledge	low	spectral cutoffs for water, Landsat formats
	graphics	complexity	low	prepares data for map but doesn't do graphics itself
	user-interface	complexity	low	batch or interactive, nothing complex mentioned
	data-management	complexity	medium	handles map and spectral data
	sys-programming	complexity	very-low	****default****
	hard-architecture	complexity	very-low	****default****
nexus	size	total	100	
	task	complexity	high	
	methods	complexity	medium	
		generality	very-high	
		accuracy	medium	****default****
		efficiency	medium	****default****
		redundancy	low	
	objects	complexity	very-high	
		generality	very-high	
	representation	complexity	very-high	
		capacity	very-high	
		knowledge	medium	
	graphics	complexity	very-high	
	user-interface	complexity	medium	
	data-management	complexity	high	
	sys-programming	complexity	low	
	hard-architecture	complexity	very-low	****default****
care	size	total	37	
	task	complexity	high	
	methods	complexity	medium	
		generality	high	
		accuracy	high	
		efficiency	medium	****default****
		redundancy	very-low	****default****
	objects	complexity	medium	
		generality	high	

	representation	complexity	medium	****default****
		capacity	high	
		knowledge	low	
	graphics	complexity	low	
	user-interface	complexity	low	
	data-management	complexity	low	
	sys-programming	complexity	very-low	****default****
	hard-architecture	complexity	very-low	****default****
rim	size	total	51	
	task	complexity	medium	
	methods	complexity	medium	
		generality	very-high	
		accuracy	medium	****default****
		efficiency	medium	****default****
		redundancy	very-low	****default****
	objects	complexity	very-high	
		generality	very-high	
	representation	complexity	medium	****default****
		capacity	very-high	
		knowledge	very-low	
	graphics	complexity	very-low	
	user-interface	complexity	high	
	data-management	complexity	very-high	
	sys-programming	complexity	very-low	****default****
	hard-architecture	complexity	very-low	****default****
afile	size	total	36	
	task	complexity	low	
	methods	complexity	low	
		generality	low	
		accuracy	medium	****default****
		efficiency	medium	****default****
		redundancy	very-low	****default****
	objects	complexity	medium	
		generality	medium	
	representation	complexity	medium	****default****
		capacity	medium	****default****
		knowledge	medium	
	graphics	complexity	low	
	user-interface	complexity	low	
	data-management	complexity	low	
	sys-programming	complexity	very-low	****default****
	hard-architecture	complexity	very-low	****default****
fpt	size	total	18	
	task	complexity	low	
	methods	complexity	low	
		generality	very-low	
		accuracy	medium	****default****
		efficiency	medium	****default****
		redundancy	very-low	****default****
	objects	complexity	low	
		generality	medium	
	representation	complexity	low	
		capacity	medium	****default****
		knowledge	medium	
	graphics	complexity	very-low	****default****
	user-interface	complexity	low	
	data-management	complexity	very-low	****default****
	sys-programming	complexity	very-low	****default****
	hard-architecture	complexity	very-low	****default****
sap	size	total	12	
	task	complexity	very-low	
	methods	complexity	low	
		generality	low	
		accuracy	medium	****default****
		efficiency	medium	****default****
		redundancy	very-low	****default****
	objects	complexity	low	
		generality	low	

```
          representation      complexity  very-low
                              capacity    medium      ****default****
                              knowledge   very-low
          graphics            complexity  very-low    ****default****
          user-interface      complexity  very-low
          data-management     complexity  very-low    ****default****
          sys-programming     complexity  very-low    ****default****
          hard-architecture   complexity  very-low    ****default****

discos    size                total       20
          task                complexity  medium
          methods             complexity  high
                              generality  high
                              accuracy    very-high
                              efficiency  very-low
                              redundancy  very-low    ****default****
          objects             complexity  medium
                              generality  medium
          representation      complexity  medium      ****default****
                              capacity    low         ****default****
                              knowledge   medium
          graphics            complexity  low
          user-interface      complexity  very-low
          data-management     complexity  very-low    ****default****
          sys-programming     complexity  very-low    ****default****
          hard-architecture   complexity  very-low    ****default****

samsan    size                total       20
          task                complexity  medium
          methods             complexity  medium
                              generality  medium
                              accuracy    medium      ****default****
                              efficiency  medium      ****default****
                              redundancy  very-low    ****default****
          objects             complexity  medium
                              generality  medium
          representation      complexity  medium      ****default****
                              capacity    very-low
                              knowledge   very-low
          graphics            complexity  very-low    ****default****
          user-interface      complexity  very-low
          data-management     complexity  very-low    ****default****
          sys-programming     complexity  very-low    ****default****
          hard-architecture   complexity  very-low    ****default****

spar      size                total       87
          task                complexity  very-high
          methods             complexity  very-high
                              generality  very-high
                              accuracy    high
                              efficiency  medium      ****default****
                              redundancy  medium
          objects             complexity  very-high
                              generality  very-high
          representation      complexity  high
                              capacity    very-high
                              knowledge   high
          graphics            complexity  medium
          user-interface      complexity  medium
          data-management     complexity  high
          sys-programming     complexity  very-low    ****default****
          hard-architecture   complexity  very-low    ****default****


end
          size                total
          task                complexity
          methods             complexity
                              generality
                              accuracy    medium      ****default****
                              efficiency  medium      ****default****
```

	redundancy	very-low	****default****
objects	complexity		
	generality		
representation	complexity	medium	****default****
	capacity	medium	****default****
	knowledge		
graphics	complexity	very-low	****default****
user-interface	complexity		
data-management	complexity	very-low	****default****
sys-programming	complexity	very-low	****default****
hard-architecture	complexity	very-low	****default****

hd:nasa:knowledge acquisition system:hand-tuned-generic.lisp

big-eng	size	total	500
	task	complexity	very-high
	methods	complexity	very-high
		generality	very-high
		accuracy	very-high
		efficiency	very-high
		redundancy	very-high
	objects	complexity	very-high
		generality	very-high
	representation	complexity	very-high
		capacity	very-high
		knowledge	very-high
	graphics	complexity	very-high
	user-interface	complexity	very-high
	data-management	complexity	very-high
science-only	size	total	10
	task	complexity	medium
	methods	complexity	high
		generality	high
		accuracy	very-high
		efficiency	very-low
		redundancy	very-low
	objects	complexity	medium
		generality	medium
	representation	complexity	medium
		capacity	very-low
		knowledge	medium
	graphics	complexity	very-low
	user-interface	complexity	very-low
	data-management	complexity	very-low
	sys-programming	complexity	very-low
	hard-architecture	complexity	very-low

Appendix 4 - Test Results

This appendix contains the contents of file hd:nasa:knowledge acquisition system:test-results.lisp which contains the detailed results of the testing in frame form:

```
(TEST-SYSTEM (NUMBER-OF-INSTANCES (VALUE 1)))

(TEST-SYSTEM1 (A-KIND-OF (VALUE TEST-SYSTEM))
 (PROGRAMS
  (VALUE NASTRAN TRASYS STAGSC-1 SSPTA PASCO PANAIR SINDA VICAR JPLDIS NEXUS CARE RIM AFILE FPT
   SAP DISCOS DIDS SAMSAN SPAR DYLOFLEX FLEXSTAB BIG-ENG SCIENCE-ONLY PILOT USSAERO EOM-DYLO
   COREL RHOIV GTDS RADMAS FSD POST3D POST6D SORT VIEW NECAP STARS2S PLANS NASCAP SIGMAII
   LARSFRIS GEMPAK DAMS))
 (LEARNING (ETA) (ALPHA)) (RANDOM-SIZING (VALUE (BIAS-FACTOR 1.0 FLUCTUATION-FACTOR 4.437556)))
 (TEST-SET
  (VALUE NASTRAN TRASYS STAGSC-1 SSPTA PASCO PANAIR SINDA VICAR JPLDIS NEXUS CARE RIM AFILE FPT
   SAP DISCOS DIDS SAMSAN SPAR DYLOFLEX FLEXSTAB))
 (TESTED
  (VALUE NASTRAN TRASYS STAGSC-1 SSPTA PASCO PANAIR SINDA VICAR JPLDIS NEXUS CARE RIM AFILE FPT
   SAP DISCOS DIDS SAMSAN SPAR DYLOFLEX FLEXSTAB))
 (SIZING-LEARNING (ALPHA 0.4) (ETA 0.2)) (GENERIC-LEARNING (ETA 0.002))
 (FINAL-GENERIC-CALIBRATION (VALUE (ITER 84 ERR 0.4190327)))
 (FINAL-SIZING-CALIBRATION (VALUE (ITER 220 ERR 0.025613084))))

(NASTRAN (TOTAL-SIZE (VALUE 320))
 (DESIGN-CRITERIA
  (VALUE TASK20 METHODS20 OBJECTS20 REPRESENTATION20 GRAPHICS20 USER-INTERFACE20
   DATA-MANAGEMENT20))
 (TASK (ENGINEERING-PROGRAM YES))
 (ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING YES) (ELECTRONIC-CIRCUIT NO) (THERMODYNAMICS YES)
  (CONTROL-SYSTEMS YES))
 (THERMODYNAMICS (RADIATION NO) (CONDUCTION YES)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-HIGH))
 (TIME-DEPENDENCE (STATIC YES) (QUASI-STATIC YES) (FULLY-DYNAMIC YES))
 (LINEARITY (LINEAR YES) (LIMITED-NON-LINEAR YES) (FULLY-NON-LINEAR NO))
 (BOUNDARY (NUMBER-WAYS-TO-SPECIFY VERY-HIGH) (2D-FIELD-ON-3D-SURFACE YES)
  (LUMPED-COMPONENT-VALUES YES))
 (OBJECTS (SOLIDS YES) (FLUIDS YES)) (SOLIDS (STRUCTURE YES) (POINT-MASS YES))
 (STRUCTURE (SPATIAL-DIMENSION 3) (LUMPED-PARAMETER NO) (CONTINUOUS-VOLUME YES)
  (NUMBER-OF-ELEMENTS VERY-HIGH) (SUBSTRUCTURING-CAPABILITY YES))
 (CONTINUOUS-VOLUME (FINITE-ELEMENTS YES) (GRID NO) (NUMBER-OF-KNOWN-SHAPES VERY-HIGH))
 (SHAPE-LIMITATIONS (ESSENTIALLY-NONE YES) (THIN-SHELLS YES) (SHELLS-OF-REVOL YES) (SLABS YES)
  (THICK-SHELLS YES))
 (FLUIDS (TURBULENCE NO) (SHOCK-WAVES NO))
 (USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR YES) (CONTROL-LANGUAGE YES)
  (USER-SPECIFIED-ROUTINES YES) (MATRIX-SPEC-LANGUAGE YES) (LIBRARIES YES))
 (LIBRARIES (NUMBER-OF-STORED-SHAPES VERY-HIGH) (FORMAT-CONVERSION YES)
  (GEOMETRIC-CONVERSION YES) (TRANSLATE-FOR-OTHER-PROGS NO))
 (GRAPHICS (PRESENT YES) (3D-STRUCTURE YES) (NUMBER-OF-FORMATS VERY-HIGH) (PERSPECTIVE YES)
  (HIDDEN-LINE-REMOVAL YES) (SHADING NO))
 (COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD YES)
  (MANY-UPGRADES YES) (MANY-COMPUTERS YES))
 (GENERIC-COMPONENTS
  (TEST
   (1.2751708 0.8198107 1.1691252 1.1412227 0.9513452 0.6909006 0.5685283 1.1370099 0.7120148
    0.99244916 0.8017662 1.0109091 1.1377828 0.68481284 -0.12369816 -0.08889696))
  (ACTUAL (1 1 1 1 0.75 1 1 1 1 1 1 1 1 0.5 0 0))
  (CALIBRATED
   (1.0334704 1.0411793 0.98189485 0.98563266 0.7712983 0.95388347 0.9624302 1.0044041 1.0214492
    0.9861505 0.9655042 1.004624 0.9562767 0.5076866 -0.042902477 -0.10574649)))
 (CALIBRATION-SIZING (ACTUAL-COMPONENTS 235.81723) (TEST-COMPONENTS 273.62283)
  (CALIBRATION-COMPONENTS 229.15096))
 (GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.086743094)))
 (SIZING-CALIBRATION (VALUE (ITERATIONS 192 ERROR 0.22981052)))
 (TEST-SIZING (TEST-COMPONENTS 156.14948) (ACTUAL-COMPONENTS 159.57074))
 (SIZING-LEARNING (VALUE (ETA 0.2 ALPHA 0.8 MAX-ITER 400 MAX-ERR 0.01))))

(TRASYS (TOTAL-SIZE (VALUE 60))
 (DESIGN-CRITERIA
  (VALUE TASK16 METHODS16 OBJECTS16 REPRESENTATION16 GRAPHICS16 USER-INTERFACE16
   DATA-MANAGEMENT16))
 (TASK (ENGINEERING-PROGRAM YES))
 (ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING NO) (ELECTRONIC-CIRCUIT NO) (THERMODYNAMICS YES)
  (CONTROL-SYSTEMS NO))
```

(THERMODYNAMICS (RADIATION YES) (CONDUCTION NO)) (METHODS (NUMBER-REDUNDANT-CHOICES MEDIUM))
(TIME-DEPENDENCE (STATIC YES) (QUASI-STATIC YES) (FULLY-DYNAMIC NO))
(LINEARITY (LINEAR YES) (LIMITED-NON-LINEAR YES) (FULLY-NON-LINEAR NO))
(BOUNDARY (NUMBER-WAYS-TO-SPECIFY MEDIUM) (2D-FIELD-ON-3D-SURFACE NO)
(LUMPED-COMPONENT-VALUES NO))
(OBJECTS (SOLIDS YES) (FLUIDS NO)) (SOLIDS (STRUCTURE YES) (POINT-MASS NO))
(STRUCTURE (SPATIAL-DIMENSION 3) (LUMPED-PARAMETER YES) (CONTINUOUS-VOLUME NO)
(NUMBER-OF-ELEMENTS MEDIUM) (SUBSTRUCTURING-CAPABILITY NO))
(CONTINUOUS-VOLUME (FINITE-ELEMENTS NO) (GRID NO) (NUMBER-OF-KNOWN-SHAPES N/A))
(SHAPE-LIMITATIONS (ESSENTIALLY-NONE N/A) (THIN-SHELLS N/A) (SHELLS-OF-REVOL N/A) (SLABS N/A)
(THICK-SHELLS N/A))
(FLUIDS (TURBULENCE N/A) (SHOCK-WAVES N/A))
(USER-INTERFACE (TEXT-EDITOR YES) (DATA-MANIPULATOR YES) (CONTROL-LANGUAGE YES)
(USER-SPECIFIED-ROUTINES YES) (MATRIX-SPEC-LANGUAGE YES) (LIBRARIES YES))
(LIBRARIES (NUMBER-OF-STORED-SHAPES N/A) (FORMAT-CONVERSION YES) (GEOMETRIC-CONVERSION YES)
(TRANSLATE-FOR-OTHER-PROGS YES))
(GRAPHICS (PRESENT YES) (3D-STRUCTURE YES) (NUMBER-OF-FORMATS LOW) (PERSPECTIVE NO)
(HIDDEN-LINE-REMOVAL NO) (SHADING NO))
(COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD YES)
(MANY-UPGRADES YES) (MANY-COMPUTERS YES))
(GENERIC-COMPONENTS
(TEST
(0.47948068 0.20325413 0.9985753 0.34528643 0.8170441 -0.011420786 0.79440755 0.7340747
0.74895555 0.87004644 0.47808763 0.360407 0.40167683 0.3880681 -0.09053253 -0.1365945))
(ACTUAL (0.75 0.5 1 0.5 0.5 0.75 0.75 0.75 0.5 0.5 0.75 0.5 0.75 0 0 0))
(CALIBRATED
(0.7629907 0.46175537 0.9072276 0.37445438 0.5201953 0.65401894 0.6737648 0.74198276
0.44536638 0.48041087 0.7174295 0.5464649 0.62948567 -0.14859799 0.073859036 0.06320888)))
(CALIBRATION-SIZING (ACTUAL-COMPONENTS 62.023563) (TEST-COMPONENTS 59.986786)
(CALIBRATION-COMPONENTS 68.81633))
(GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.08331497)))
(SIZING-CALIBRATION (VALUE (ITERATIONS 199 ERROR 0.053455487)))
(TEST-SIZING (TEST-COMPONENTS 59.642075) (ACTUAL-COMPONENTS 48.770435)))

(STAGSC-1 (TOTAL-SIZE (VALUE 73))
(DESIGN-CRITERIA
(VALUE TASK22 METHODS22 OBJECTS22 REPRESENTATION22 GRAPHICS22 USER-INTERFACE22
DATA-MANAGEMENT22))
(TASK (ENGINEERING-PROGRAM YES))
(ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING YES) (ELECTRONIC-CIRCUIT NO) (THERMODYNAMICS NO)
(CONTROL-SYSTEMS NO))
(THERMODYNAMICS (RADIATION NO) (CONDUCTION NO)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-LOW))
(TIME-DEPENDENCE (STATIC YES) (QUASI-STATIC YES) (FULLY-DYNAMIC YES))
(LINEARITY (LINEAR YES) (LIMITED-NON-LINEAR YES) (FULLY-NON-LINEAR YES))
(BOUNDARY (NUMBER-WAYS-TO-SPECIFY MEDIUM) (2D-FIELD-ON-3D-SURFACE YES)
(LUMPED-COMPONENT-VALUES NO))
(OBJECTS (SOLIDS YES) (FLUIDS NO)) (SOLIDS (STRUCTURE YES) (POINT-MASS NO))
(STRUCTURE (SPATIAL-DIMENSION 3) (LUMPED-PARAMETER NO) (CONTINUOUS-VOLUME YES)
(NUMBER-OF-ELEMENTS LOW) (SUBSTRUCTURING-CAPABILITY YES))
(CONTINUOUS-VOLUME (FINITE-ELEMENTS YES) (GRID NO) (NUMBER-OF-KNOWN-SHAPES MEDIUM))
(SHAPE-LIMITATIONS (ESSENTIALLY-NONE NO) (THIN-SHELLS YES) (SHELLS-OF-REVOL YES) (SLABS YES)
(THICK-SHELLS YES))
(FLUIDS (TURBULENCE N/A) (SHOCK-WAVES N/A))
(USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR NO) (CONTROL-LANGUAGE NO)
(USER-SPECIFIED-ROUTINES NO) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES YES))
(LIBRARIES (NUMBER-OF-STORED-SHAPES LOW) (FORMAT-CONVERSION NO) (GEOMETRIC-CONVERSION NO)
(TRANSLATE-FOR-OTHER-PROGS NO))
(GRAPHICS (PRESENT NO) (3D-STRUCTURE N/A) (NUMBER-OF-FORMATS N/A) (PERSPECTIVE N/A)
(HIDDEN-LINE-REMOVAL N/A) (SHADING N/A))
(COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
(MANY-UPGRADES NO) (MANY-COMPUTERS YES))
(GENERIC-COMPONENTS
(TEST
(0.1435066 0.86518717 -0.433296 0.8514948 0.4510905 0.45480716 1.3713033 0.014908761
0.9948333 0.43931264 0.4847242 0.28188527 0.07538243 0.32003683 0.4767031 -0.09333875))
(ACTUAL (0.75 0.75 0.75 0.75 0.75 0 0.5 0.5 0.5 0.5 0.5 0 0.25 0 0 0))
(CALIBRATED
(0.60899746 0.74203026 0.6791901 0.69057834 0.6855666 -0.03136365 0.52202195 0.3904127
0.3930834 0.43073392 0.42583275 -0.023100503 0.22497074 -0.082200825 0.09149502 0.09563881)))
(CALIBRATION-SIZING (ACTUAL-COMPONENTS 76.33486) (TEST-COMPONENTS 60.59274)
(CALIBRATION-COMPONENTS 66.71014))
(GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.09828651)))
(SIZING-CALIBRATION (VALUE (ITERATIONS 33 ERROR 0.14178926)))
(TEST-SIZING (TEST-COMPONENTS 43.330788) (ACTUAL-COMPONENTS 39.216076))
(SIZING-LEARNING (VALUE (ETA 0.2 ALPHA 0.8 MAX-ITER 100 MAX-ERR 0.1))))

(SSPTA (TOTAL-SIZE (VALUE 13.9))
(DESIGN-CRITERIA
(VALUE TASK19 METHODS19 OBJECTS19 REPRESENTATION19 GRAPHICS19 USER-INTERFACE19
DATA-MANAGEMENT19 SYS-PROGRAMMING5 HARD-ARCHITECTURE4))
(TASK (ENGINEERING-PROGRAM YES))

(ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING NO) (ELECTRONIC-CIRCUIT NO) (THERMODYNAMICS YES)
(CONTROL-SYSTEMS NO))
(THERMODYNAMICS (RADIATION YES) (CONDUCTION YES)) (METHODS (NUMBER-REDUNDANT-CHOICES LOW))
(TIME-DEPENDENCE (STATIC NO) (QUASI-STATIC YES) (FULLY-DYNAMIC NO))
(LINEARITY (LINEAR YES) (LIMITED-NON-LINEAR YES) (FULLY-NON-LINEAR NO))
(BOUNDARY (NUMBER-WAYS-TO-SPECIFY MEDIUM) (2D-FIELD-ON-3D-SURFACE YES)
(LUMPED-COMPONENT-VALUES YES))
(OBJECTS (SOLIDS YES) (FLUIDS NO)) (SOLIDS (STRUCTURE YES) (POINT-MASS NO))
(STRUCTURE (SPATIAL-DIMENSION 3) (LUMPED-PARAMETER YES) (CONTINUOUS-VOLUME NO)
(NUMBER-OF-ELEMENTS LOW) (SUBSTRUCTURING-CAPABILITY NO))
(CONTINUOUS-VOLUME (FINITE-ELEMENTS N/A) (GRID N/A) (NUMBER-OF-KNOWN-SHAPES N/A))
(SHAPE-LIMITATIONS (ESSENTIALLY-NONE NO) (THIN-SHELLS YES) (SHELLS-OF-REVOL NO) (SLABS NO)
(THICK-SHELLS NO))
(FLUIDS (TURBULENCE N/A) (SHOCK-WAVES N/A))
(USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR NO) (CONTROL-LANGUAGE YES)
(USER-SPECIFIED-ROUTINES NO) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES NO))
(LIBRARIES (NUMBER-OF-STORED-SHAPES LOW) (FORMAT-CONVERSION YES) (GEOMETRIC-CONVERSION YES)
(TRANSLATE-FOR-OTHER-PROGS YES))
(GRAPHICS (PRESENT YES) (3D-STRUCTURE YES) (NUMBER-OF-FORMATS VERY-LOW) (PERSPECTIVE YES)
(HIDDEN-LINE-REMOVAL NO) (SHADING NO))
(COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
(MANY-UPGRADES NO) (MANY-COMPUTERS YES))
(GENERIC-COMPONENTS
(TEST
(0.42056486 0.4894415 0.97136384 0.23336713 -0.17567904 0.61932814 0.6584353 0.4238025
0.7611632 0.8285405 0.5244411 0.46153885 0.14723304 -0.0074608326 -0.38612527 0.2195806))
(ACTUAL (0.5 0.25 0.25 0.5 0.5 0 0.5 0.25 0.5 0.25 0.25 0.5 0.25 0.25 0 0))
(CALIBRATED
(0.5508115 0.37349698 0.3981396 0.67439514 0.48920202 0.22041018 0.49793077 0.30706525
0.6146174 0.40570676 0.3059947 0.49620473 0.22956978 0.3639368 -0.09089134 -0.05246365)))
(CALIBRATION-SIZING (ACTUAL-COMPONENTS 27.257244) (TEST-COMPONENTS 6.6802006)
(CALIBRATION-COMPONENTS 10.869377))
(GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.07829353)))
(SIZING-CALIBRATION (VALUE (ITERATIONS 199 ERROR 0.067516506)))
(TEST-SIZING (TEST-COMPONENTS 29.092657) (ACTUAL-COMPONENTS 34.799614)))

(PASCO (TOTAL-SIZE (VALUE 19.1))
(DESIGN-CRITERIA
(VALUE TASK25 METHODS25 OBJECTS25 REPRESENTATION25 GRAPHICS25 USER-INTERFACE25
DATA-MANAGEMENT25 SYS-PROGRAMMING6 HARD-ARCHITECTURE5))
(TASK (ENGINEERING-PROGRAM YES))
(ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING YES) (AERODYNAMICS NO) (ELECTRONIC-CIRCUIT NO)
(THERMODYNAMICS NO))
(THERMODYNAMICS (RADIATION N/A) (CONDUCTION N/A)) (METHODS (NUMBER-REDUNDANT-CHOICES LOW))
(TIME-DEPENDENCE (STATIC YES) (QUASI-STATIC YES) (FULLY-DYNAMIC NO))
(LINEARITY (LINEAR YES) (LIMITED-NON-LINEAR YES) (FULLY-NON-LINEAR NO))
(BOUNDARY (NUMBER-WAYS-TO-SPECIFY MEDIUM) (2D-FIELD-ON-3D-SURFACE NO)
(LUMPED-COMPONENT-VALUES NO))
(OBJECTS (SOLIDS YES) (FLUIDS NO)) (SOLIDS (STRUCTURE YES) (POINT-MASS NO))
(STRUCTURE (SPATIAL-DIMENSION 2) (LUMPED-PARAMETER NO) (CONTINUOUS-VOLUME YES)
(NUMBER-OF-ELEMENTS LOW) (SUBSTRUCTURING-CAPABILITY YES))
(LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS N/A))
(CONTINUOUS-VOLUME (FINITE-ELEMENTS YES) (GRID NO) (NUMBER-OF-KNOWN-SHAPES LOW))
(SHAPE-LIMITATIONS (ESSENTIALLY-NONE NO) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS YES)
(THICK-SHELLS NO))
(FLUIDS (TURBULENCE N/A) (SHOCK-WAVES N/A))
(USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR NO) (CONTROL-LANGUAGE YES)
(USER-SPECIFIED-ROUTINES NO) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES NO))
(LIBRARIES (NUMBER-OF-STORED-SHAPES VERY-LOW) (FORMAT-CONVERSION NO) (GEOMETRIC-CONVERSION NO)
(TRANSLATE-FOR-OTHER-PROGS NO))
(GRAPHICS (PRESENT YES) (3D-STRUCTURE YES) (NUMBER-OF-FORMATS LOW) (PERSPECTIVE NO)
(HIDDEN-LINE-REMOVAL NO) (SHADING NO))
(COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
(MANY-UPGRADES NO) (MANY-COMPUTERS YES))
(GENERIC-COMPONENTS
(TEST
(0.6522589 0.2653798 0.5605383 -0.09349618 0.5820937 -6.107539e-4 0.67525715 0.24664858
0.36795023 0.25727636 0.13324875 0.47085863 0.4285 0.140006 0.36331007 -0.036286965))
(ACTUAL (0.5 0.5 0.25 0.5 0.5 0 0.25 0.25 0.5 0.5 0.25 0.25 0 0 0 0))
(CALIBRATED
(0.692163 0.4032512 0.3788594 0.45397532 0.5587638 0.032834478 0.2807214 0.3366832 0.60827285
0.5109307 0.39906973 0.28692234 0.031162973 0.06776053 -0.10175448 -0.040513843)))
(CALIBRATION-SIZING (ACTUAL-COMPONENTS 25.265142) (TEST-COMPONENTS 31.032166)
(CALIBRATION-COMPONENTS 19.555958))
(GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.092116594)))
(SIZING-CALIBRATION (VALUE (ITERATIONS 27 ERROR 0.12434305)))
(TEST-SIZING (TEST-COMPONENTS 35.431423) (ACTUAL-COMPONENTS 27.735744))
(SIZING-LEARNING (VALUE (ETA 0.2 ALPHA 0.8 MAX-ITER 400 MAX-ERR 0.01))))

(PANAIR (TOTAL-SIZE (VALUE 289))
(DESIGN-CRITERIA

```
(VALUE TASK1 METHODS1 OBJECTS1 REPRESENTATION1 GRAPHICS1 USER-INTERFACE1 DATA-MANAGEMENT1))
(TASK (ENGINEERING-PROGRAM YES))
(ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING YES) (AERODYNAMICS YES) (ELECTRONIC-CIRCUIT NO)
 (THERMODYNAMICS NO))
(THERMODYNAMICS (RADIATION N/A) (CONDUCTION N/A)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-LOW))
(TIME-DEPENDENCE (STATIC NO) (QUASI-STATIC NO) (FULLY-DYNAMIC YES))
(LINEARITY (LINEAR NO) (LIMITED-NON-LINEAR NO) (FULLY-NON-LINEAR YES))
(BOUNDARY (NUMBER-WAYS-TO-SPECIFY VERY-HIGH) (2D-FIELD-ON-3D-SURFACE YES)
 (LUMPED-COMPONENT-VALUES YES))
(OBJECTS (SOLIDS YES) (FLUIDS YES)) (SOLIDS (STRUCTURE YES) (POINT-MASS NO))
(STRUCTURE (SPATIAL-DIMENSION 3) (LUMPED-PARAMETER YES) (CONTINUOUS-VOLUME YES)
 (NUMBER-OF-ELEMENTS VERY-HIGH) (SUBSTRUCTURING-CAPABILITY YES))
(LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS LOW))
(CONTINUOUS-VOLUME (FINITE-ELEMENTS NO) (GRID YES) (NUMBER-OF-KNOWN-SHAPES MEDIUM))
(SHAPE-LIMITATIONS (ESSENTIALLY-NONE YES) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS NO)
 (THICK-SHELLS NO))
(FLUIDS (TURBULENCE YES) (SHOCK-WAVES YES))
(USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR NO) (CONTROL-LANGUAGE YES)
 (USER-SPECIFIED-ROUTINES NO) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES NO))
(LIBRARIES (NUMBER-OF-STORED-SHAPES LOW) (FORMAT-CONVERSION NO) (GEOMETRIC-CONVERSION NO)
 (TRANSLATE-FOR-OTHER-PROGS NO))
(GRAPHICS (PRESENT NO) (3D-STRUCTURE N/A) (NUMBER-OF-FORMATS N/A) (PERSPECTIVE N/A)
 (HIDDEN-LINE-REMOVAL N/A) (SHADING N/A))
(COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD YES)
 (MANY-UPGRADES YES) (MANY-COMPUTERS YES))
(GENERIC-COMPONENTS
 (TEST
  (0.9479831 1.153682 1.189929 1.2583556 0.49530473 0.81115544 1.0982032 1.3823335 0.9501823
   0.88016313 0.6645733 -0.08619308 0.25793803 1.4496528 0.17116198 0.5072423))
 (ACTUAL (1 1 1 1 0.75 0 1 1 0.75 0.75 0.75 0 0.75 0.75 0 0))
 (CALIBRATED
  (1.0188699 1.0589675 1.0022615 0.975085 0.62295675 0.15484509 1.0659341 1.0977899 0.7800848
   0.7710583 0.73738354 0.025694616 0.4767716 0.7373127 -0.033514455 0.008537844)))
(CALIBRATION-SIZING (ACTUAL-COMPONENTS 196.8014) (TEST-COMPONENTS 240.05965)
 (CALIBRATION-COMPONENTS 220.413))
(GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.08531721)))
(SIZING-CALIBRATION (VALUE (ITERATIONS 349 ERROR 0.25131726)))
(TEST-SIZING (TEST-COMPONENTS 171.68573) (ACTUAL-COMPONENTS 175.18758))
(SIZING-LEARNING (VALUE (ETA 0.25 ALPHA 0.5 MAX-ITER 350 MAX-ERR 0.01))))

(SINDA (TOTAL-SIZE (VALUE 40))
 (DESIGN-CRITERIA
  (VALUE TASK15 METHODS15 OBJECTS15 REPRESENTATION15 GRAPHICS15 USER-INTERFACE15
   DATA-MANAGEMENT15))
 (TASK (ENGINEERING-PROGRAM YES))
 (ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING NO) (AERODYNAMICS NO) (ELECTRONIC-CIRCUIT NO)
  (THERMODYNAMICS YES))
 (THERMODYNAMICS (RADIATION NO) (CONDUCTION YES)) (METHODS (NUMBER-REDUNDANT-CHOICES LOW))
 (TIME-DEPENDENCE (STATIC YES) (QUASI-STATIC YES) (FULLY-DYNAMIC NO))
 (LINEARITY (LINEAR YES) (LIMITED-NON-LINEAR YES) (FULLY-NON-LINEAR NO))
 (BOUNDARY (NUMBER-WAYS-TO-SPECIFY MEDIUM) (2D-FIELD-ON-3D-SURFACE NO)
  (LUMPED-COMPONENT-VALUES YES))
 (OBJECTS (SOLIDS YES) (FLUIDS YES)) (SOLIDS (STRUCTURE YES) (POINT-MASS NO))
 (STRUCTURE (SPATIAL-DIMENSION N/A) (LUMPED-PARAMETER YES) (CONTINUOUS-VOLUME NO)
  (NUMBER-OF-ELEMENTS HIGH) (SUBSTRUCTURING-CAPABILITY NO))
 (LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS VERY-HIGH))
 (CONTINUOUS-VOLUME (FINITE-ELEMENTS N/A) (GRID N/A) (NUMBER-OF-KNOWN-SHAPES N/A))
 (SHAPE-LIMITATIONS (ESSENTIALLY-NONE N/A) (THIN-SHELLS N/A) (SHELLS-OF-REVOL N/A) (SLABS N/A)
  (THICK-SHELLS N/A))
 (FLUIDS (TURBULENCE NO) (SHOCK-WAVES NO))
 (USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR NO) (CONTROL-LANGUAGE YES)
  (USER-SPECIFIED-ROUTINES YES) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES YES))
 (LIBRARIES (NUMBER-OF-STORED-SHAPES VERY-LOW) (FORMAT-CONVERSION NO) (GEOMETRIC-CONVERSION NO)
  (TRANSLATE-FOR-OTHER-PROGS NO))
 (GRAPHICS (PRESENT NO) (3D-STRUCTURE N/A) (NUMBER-OF-FORMATS N/A) (PERSPECTIVE N/A)
  (HIDDEN-LINE-REMOVAL N/A) (SHADING N/A))
 (COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD YES)
  (MANY-UPGRADES YES) (MANY-COMPUTERS YES))
 (GENERIC-COMPONENTS
  (TEST
   (0.45942342 -0.42581564 0.17623568 0.60225344 0.22078496 -0.23664671 0.09279606 0.18790704
    0.01963333 1.0027336 0.48070848 0.63430136 0.1397767 0.2185854 0.26920015 -0.1565734))
  (ACTUAL (0.5 0.5 1 0.5 0.5 1 0.5 0.75 0.5 0.5 0.75 0 0.5 0 0 0))
  (CALIBRATED
   (0.42784935 0.4364939 0.948451 0.53064555 0.51919794 0.8307121 0.60171765 0.8052182
    0.33206338 0.4639923 0.7372322 -0.08082323 0.6523174 0.037354752 -0.042308643 0.014266387)))
 (CALIBRATION-SIZING (ACTUAL-COMPONENTS 41.816307) (TEST-COMPONENTS 15.303929)
  (CALIBRATION-COMPONENTS 28.192623))
 (GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.097088955)))
 (SIZING-CALIBRATION (VALUE (ITERATIONS 33 ERROR 0.09989633)))
 (TEST-SIZING (TEST-COMPONENTS 39.586308) (ACTUAL-COMPONENTS 44.042698))
```

(SIZING-LEARNING (VALUE (ETA 0.25 ALPHA 0.5 MAX-ITER 100 MAX-ERR 0.1))))

(VICAR (TOTAL-SIZE (VALUE 150))
(DESIGN-CRITERIA
(VALUE TASK28 METHODS28 OBJECTS28 REPRESENTATION28 GRAPHICS28 USER-INTERFACE28
DATA-MANAGEMENT28))
(TASK (DATABASE-PROGRAM YES) (ENGINEERING-PROGRAM YES)) (DATABASE-PROGRAM (VIDEO-IMAGES YES))
(ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING NO) (AERODYNAMICS NO) (ELECTRONIC-CIRCUIT NO)
(THERMODYNAMICS NO) (IMAGE-PROCESSING YES))
(THERMODYNAMICS (RADIATION N/A) (CONDUCTION N/A))
(METHODS (NUMBER-REDUNDANT-CHOICES VERY-HIGH))
(TIME-DEPENDENCE (STATIC N/A) (QUASI-STATIC N/A) (FULLY-DYNAMIC N/A))
(LINEARITY (LINEAR N/A) (LIMITED-NON-LINEAR N/A) (FULLY-NON-LINEAR N/A))
(BOUNDARY (NUMBER-WAYS-TO-SPECIFY N/A) (2D-FIELD-ON-3D-SURFACE N/A)
(LUMPED-COMPONENT-VALUES N/A))
(OBJECTS (NON-PHYSICAL YES) (SOLIDS NO) (FLUIDS NO)) (NON-PHYSICAL (IMAGES YES))
(SOLIDS (STRUCTURE N/A) (POINT-MASS N/A))
(STRUCTURE (SPATIAL-DIMENSION N/A) (LUMPED-PARAMETER N/A) (CONTINUOUS-VOLUME N/A)
(NUMBER-OF-ELEMENTS N/A) (SUBSTRUCTURING-CAPABILITY N/A))
(CONTINUOUS-VOLUME (FINITE-ELEMENTS N/A) (GRID N/A) (NUMBER-OF-KNOWN-SHAPES N/A))
(SHAPE-LIMITATIONS (ESSENTIALLY-NONE N/A) (THIN-SHELLS N/A) (SHELLS-OF-REVOL N/A) (SLABS N/A)
(THICK-SHELLS N/A))
(FLUIDS (TURBULENCE N/A) (SHOCK-WAVES N/A))
(USER-INTERFACE (TEXT-EDITOR YES) (DATA-MANIPULATOR YES) (CONTROL-LANGUAGE YES)
(USER-SPECIFIED-ROUTINES YES) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES YES))
(LIBRARIES (NUMBER-OF-STORED-SHAPES HIGH) (FORMAT-CONVERSION YES) (GEOMETRIC-CONVERSION YES)
(TRANSLATE-FOR-OTHER-PROGS YES))
(GRAPHICS (PRESENT YES) (3D-STRUCTURE NO) (NUMBER-OF-FORMATS VERY-HIGH) (PERSPECTIVE N/A)
(HIDDEN-LINE-REMOVAL N/A) (SHADING N/A))
(COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
(MANY-UPGRADES NO) (MANY-COMPUTERS NO))
(SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS YES) (CODE-TO-SAVE-MEMORY YES))
(GENERIC-COMPONENTS
(TEST
(1.0681171 0.025971338 0.3230194 0.3345483 1.1398964 -0.24951836 0.2876497 0.40102836
0.92103416 0.5303075 0.38791782 0.3254585 1.3670759 0.533749 0.79391867 0.09500097))
(ACTUAL (1 0.5 1 0.5 0.75 0.75 1 1 0.5 1 0.75 0.75 0.75 1 0 0))
(CALIBRATED
(0.8897602 0.47800544 0.92042464 0.4084196 0.62014973 0.64227253 0.9694961 0.7874404
0.59618217 0.7915079 0.6110095 0.7616864 0.6617759 0.80959404 0.20312394 0.08914299)))
(CALIBRATION-SIZING (ACTUAL-COMPONENTS 181.66124) (TEST-COMPONENTS 187.21155)
(CALIBRATION-COMPONENTS 134.56613))
(GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.10547809)))
(SIZING-CALIBRATION (VALUE (ITERATIONS 199 ERROR 0.06004925)))
(TEST-SIZING (TEST-COMPONENTS 108.015976) (ACTUAL-COMPONENTS 97.149635)))

(JPLDIS (TOTAL-SIZE (VALUE 39))
(DESIGN-CRITERIA
(VALUE TASK31 METHODS31 OBJECTS31 REPRESENTATION31 GRAPHICS31 USER-INTERFACE31
DATA-MANAGEMENT31))
(TASK (DATABASE-PROGRAM YES) (ENGINEERING-PROGRAM NO)) (DATABASE-PROGRAM (VIDEO-IMAGES NO))
(ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING N/A) (AERODYNAMICS N/A) (ELECTRONIC-CIRCUIT N/A)
(THERMODYNAMICS N/A) (IMAGE-PROCESSING N/A))
(THERMODYNAMICS (RADIATION N/A) (CONDUCTION N/A)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-LOW))
(TIME-DEPENDENCE (STATIC N/A) (QUASI-STATIC N/A) (FULLY-DYNAMIC N/A))
(LINEARITY (LINEAR N/A) (LIMITED-NON-LINEAR N/A) (FULLY-NON-LINEAR N/A))
(BOUNDARY (NUMBER-WAYS-TO-SPECIFY N/A) (2D-FIELD-ON-3D-SURFACE N/A)
(LUMPED-COMPONENT-VALUES N/A))
(OBJECTS (NON-PHYSICAL YES) (SOLIDS NO) (FLUIDS NO)) (NON-PHYSICAL (IMAGES NO))
(SOLIDS (STRUCTURE N/A) (POINT-MASS N/A))
(STRUCTURE (SPATIAL-DIMENSION N/A) (LUMPED-PARAMETER N/A) (CONTINUOUS-VOLUME N/A)
(NUMBER-OF-ELEMENTS N/A) (SUBSTRUCTURING-CAPABILITY N/A))
(CONTINUOUS-VOLUME (FINITE-ELEMENTS N/A) (GRID N/A) (NUMBER-OF-KNOWN-SHAPES N/A))
(SHAPE-LIMITATIONS (ESSENTIALLY-NONE N/A) (THIN-SHELLS N/A) (SHELLS-OF-REVOL N/A) (SLABS N/A)
(THICK-SHELLS N/A))
(FLUIDS (TURBULENCE N/A) (SHOCK-WAVES N/A))
(USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR NO) (CONTROL-LANGUAGE YES)
(USER-SPECIFIED-ROUTINES YES) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES NO))
(LIBRARIES (NUMBER-OF-STORED-SHAPES N/A) (FORMAT-CONVERSION YES) (GEOMETRIC-CONVERSION NO)
(TRANSLATE-FOR-OTHER-PROGS NO))
(GRAPHICS (PRESENT NO) (3D-STRUCTURE N/A) (NUMBER-OF-FORMATS N/A) (PERSPECTIVE N/A)
(HIDDEN-LINE-REMOVAL N/A) (SHADING N/A))
(COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
(MANY-UPGRADES YES) (MANY-COMPUTERS NO))
(SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS YES) (CODE-TO-SAVE-MEMORY NO))
(GENERIC-COMPONENTS
(TEST
(0.43061173 0.6631439 0.73389655 0.5882627 0.26302525 -0.17896326 0.8580383 0.820793
0.39751613 0.90882164 0.1420773 -0.22389877 0.42164248 0.50919217 0.14725482 0.33913976))
(ACTUAL (0.5 0.5 1 1 0.5 0 0.25 1 0.25 0.75 0.25 0 0.75 0.75 0 0))
(CALIBRATED

```
(0.35664666 0.46817365 0.9200645 0.7530614 0.5102373 0.06482123 0.5801659 0.8841848
 0.27480507 0.81599283 0.20233002 0.08729164 0.61276144 0.63928866 -0.089322515 0.016387329)))
(CALIBRATION-SIZING (ACTUAL-COMPONENTS 66.053024) (TEST-COMPONENTS 37.85231)
 (CALIBRATION-COMPONENTS 33.100643))
(GENERIC-CALIBRATION (VALUE ((ITERATIONS 200 ERROR 0.07869448)))
(SIZING-CALIBRATION (VALUE ((ITERATIONS 199 ERROR 0.06972669)))
(TEST-SIZING (TEST-COMPONENTS 48.241173) (ACTUAL-COMPONENTS 64.810295)))

(NEXUS (TOTAL-SIZE (VALUE 100))
 (DESIGN-CRITERIA
 (VALUE TASK34 METHODS34 OBJECTS34 REPRESENTATION34 GRAPHICS34 USER-INTERFACE34
  DATA-MANAGEMENT34 SYS-PROGRAMMING11 HARD-ARCHITECTURE10))
 (TASK (DATABASE-PROGRAM YES) (ENGINEERING-PROGRAM YES)) (DATABASE-PROGRAM (VIDEO-IMAGES NO))
 (ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING YES) (AERODYNAMICS NO) (ELECTRONIC-CIRCUIT NO)
  (THERMODYNAMICS NO) (IMAGE-PROCESSING NO))
 (THERMODYNAMICS (RADIATION NO) (CONDUCTION NO)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-LOW))
 (TIME-DEPENDENCE (STATIC NO) (QUASI-STATIC NO) (FULLY-DYNAMIC NO))
 (LINEARITY (LINEAR NO) (LIMITED-NON-LINEAR NO) (FULLY-NON-LINEAR NO))
 (BOUNDARY (NUMBER-WAYS-TO-SPECIFY LOW) (2D-FIELD-ON-3D-SURFACE YES)
  (LUMPED-COMPONENT-VALUES YES))
 (OBJECTS (NON-PHYSICAL YES) (SOLIDS YES) (FLUIDS NO)) (NON-PHYSICAL (IMAGES YES))
 (SOLIDS (STRUCTURE YES) (POINT-MASS NO))
 (STRUCTURE (SPATIAL-DIMENSION 3) (LUMPED-PARAMETER NO) (CONTINUOUS-VOLUME YES)
  (NUMBER-OF-ELEMENTS VERY-HIGH) (SUBSTRUCTURING-CAPABILITY YES))
 (LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS VERY-LOW))
 (CONTINUOUS-VOLUME (FINITE-ELEMENTS YES) (GRID YES) (NUMBER-OF-KNOWN-SHAPES MEDIUM))
 (SHAPE-LIMITATIONS (ESSENTIALLY-NONE YES) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS NO)
  (THICK-SHELLS NO))
 (FLUIDS (TURBULENCE NO) (SHOCK-WAVES NO))
 (USER-INTERFACE (TEXT-EDITOR YES) (DATA-MANIPULATOR YES) (CONTROL-LANGUAGE YES)
  (USER-SPECIFIED-ROUTINES YES) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES YES))
 (LIBRARIES (NUMBER-OF-STORED-SHAPES MEDIUM) (FORMAT-CONVERSION YES) (GEOMETRIC-CONVERSION YES)
  (TRANSLATE-FOR-OTHER-PROGS YES))
 (GRAPHICS (PRESENT YES) (3D-STRUCTURE YES) (NUMBER-OF-FORMATS VERY-HIGH) (PERSPECTIVE YES)
  (HIDDEN-LINE-REMOVAL YES) (SHADING YES))
 (COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
  (MANY-UPGRADES NO) (MANY-COMPUTERS NO))
 (SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS YES) (CODE-TO-SAVE-MEMORY NO))
 (GENERIC-COMPONENTS
 (TEST
  (0.85368806 0.57809573 0.43191415 0.60210043 1.2169985 0.83660614 0.76886386 0.78794444
   1.6151308 0.64545137 0.34874147 0.418124 0.94660187 0.8610997 -0.5722993 0.2241874))
  (ACTUAL (0.75 0.5 1 0.5 0.5 0.25 1 1 1 1 0.5 1 0.5 0.75 0.25 0))
  (CALIBRATED
  (0.7997466 0.4039193 0.99121743 0.50555897 0.6486532 0.2175319 0.996769 1.0377867 0.97271925
   1.0539267 0.6046084 0.89704067 0.5983577 0.7703494 0.16366906 -0.064409785)))
 (CALIBRATION-SIZING (ACTUAL-COMPONENTS 116.32308) (TEST-COMPONENTS 169.24077)
  (CALIBRATION-COMPONENTS 105.86646))
 (GENERIC-CALIBRATION (VALUE ((ITERATIONS 200 ERROR 0.070239246)))
 (SIZING-CALIBRATION (VALUE ((ITERATIONS 199 ERROR 0.054675683)))
 (TEST-SIZING (TEST-COMPONENTS 153.16548) (ACTUAL-COMPONENTS 127.97199)))

(CARE (TOTAL-SIZE (VALUE 37))
 (DESIGN-CRITERIA
 (VALUE TASK35 METHODS35 OBJECTS35 REPRESENTATION35 GRAPHICS35 USER-INTERFACE35
  DATA-MANAGEMENT35 SYS-PROGRAMMING12 HARD-ARCHITECTURE11))
 (TASK (DATABASE-PROGRAM NO) (ENGINEERING-PROGRAM YES)) (DATABASE-PROGRAM (VIDEO-IMAGES NO))
 (ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING NO) (AERODYNAMICS NO) (ELECTRONIC-CIRCUIT NO)
  (THERMODYNAMICS NO) (IMAGE-PROCESSING NO))
 (THERMODYNAMICS (RADIATION NO) (CONDUCTION NO)) (METHODS (NUMBER-REDUNDANT-CHOICES MEDIUM))
 (TIME-DEPENDENCE (STATIC YES) (QUASI-STATIC YES) (FULLY-DYNAMIC NO))
 (LINEARITY (LINEAR NO) (LIMITED-NON-LINEAR NO) (FULLY-NON-LINEAR NO))
 (BOUNDARY (NUMBER-WAYS-TO-SPECIFY VERY-LOW) (2D-FIELD-ON-3D-SURFACE NO)
  (LUMPED-COMPONENT-VALUES NO))
 (OBJECTS (NON-PHYSICAL YES) (SOLIDS NO) (FLUIDS NO)) (NON-PHYSICAL (IMAGES NO))
 (SOLIDS (STRUCTURE NO) (POINT-MASS NO))
 (STRUCTURE (SPATIAL-DIMENSION 0) (LUMPED-PARAMETER NO) (CONTINUOUS-VOLUME NO)
  (NUMBER-OF-ELEMENTS VERY-LOW) (SUBSTRUCTURING-CAPABILITY NO))
 (LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS VERY-LOW))
 (CONTINUOUS-VOLUME (FINITE-ELEMENTS NO) (GRID NO) (NUMBER-OF-KNOWN-SHAPES VERY-LOW))
 (SHAPE-LIMITATIONS (ESSENTIALLY-NONE NO) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS NO)
  (THICK-SHELLS NO))
 (FLUIDS (TURBULENCE NO) (SHOCK-WAVES NO))
 (USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR YES) (CONTROL-LANGUAGE NO)
  (USER-SPECIFIED-ROUTINES NO) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES NO))
 (LIBRARIES (NUMBER-OF-STORED-SHAPES VERY-LOW) (FORMAT-CONVERSION NO) (GEOMETRIC-CONVERSION NO)
  (TRANSLATE-FOR-OTHER-PROGS NO))
 (GRAPHICS (PRESENT YES) (3D-STRUCTURE NO) (NUMBER-OF-FORMATS VERY-LOW) (PERSPECTIVE NO)
  (HIDDEN-LINE-REMOVAL NO) (SHADING NO))
 (COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
  (MANY-UPGRADES NO) (MANY-COMPUTERS YES))
```

```
(SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS NO) (CODE-TO-SAVE-MEMORY NO))
 (GENERIC-COMPONENTS
  (TEST
   (0.34327382 0.15115997 1.096796 0.6282853 0.5570892 0.1476478 0.9634936 0.4991774 0.19526768
    0.30385855 0.12207735 0.23398663 0.18265322 -0.027103484 0.05588656 -0.05130118))
  (ACTUAL (0.75 0.5 0.75 0.75 0.5 0 0.5 0.75 0.5 0.75 0.25 0.25 0.25 0.25 0 0))
  (CALIBRATED
   (0.5042857 0.43761447 0.57262254 0.68576896 0.56318426 -0.10452509 0.39063954 0.7096672
    0.30933928 0.49987602 0.15523043 0.15855783 0.18279366 0.110414594 0.01195392 0.03761694)))
 (CALIBRATION-SIZING (ACTUAL-COMPONENTS 51.97947) (TEST-COMPONENTS 20.055557)
  (CALIBRATION-COMPONENTS 31.9528))
 (GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.09075029)))
 (SIZING-CALIBRATION (VALUE (ITERATIONS 199 ERROR 0.051158965)))
 (TEST-SIZING (TEST-COMPONENTS 24.024239) (ACTUAL-COMPONENTS 41.8817)))

(RIM (TOTAL-SIZE (VALUE 51))
 (DESIGN-CRITERIA
  (VALUE TASK36 METHODS36 OBJECTS36 REPRESENTATION36 GRAPHICS36 USER-INTERFACE36
   DATA-MANAGEMENT36 SYS-PROGRAMMING13 HARD-ARCHITECTURE12))
 (TASK (DATABASE-PROGRAM YES) (ENGINEERING-PROGRAM NO)) (DATABASE-PROGRAM (VIDEO-IMAGES NO))
 (ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING NO) (AERODYNAMICS NO) (ELECTRONIC-CIRCUIT NO)
  (THERMODYNAMICS NO) (IMAGE-PROCESSING NO))
 (THERMODYNAMICS (RADIATION NO) (CONDUCTION NO)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-LOW))
 (TIME-DEPENDENCE (STATIC NO) (QUASI-STATIC NO) (FULLY-DYNAMIC NO))
 (LINEARITY (LINEAR NO) (LIMITED-NON-LINEAR NO) (FULLY-NON-LINEAR NO))
 (BOUNDARY (NUMBER-WAYS-TO-SPECIFY VERY-LOW) (2D-FIELD-ON-3D-SURFACE NO)
  (LUMPED-COMPONENT-VALUES NO))
 (OBJECTS (NON-PHYSICAL YES) (SOLIDS NO) (FLUIDS NO)) (NON-PHYSICAL (IMAGES NO))
 (SOLIDS (STRUCTURE NO) (POINT-MASS NO))
 (STRUCTURE (SPATIAL-DIMENSION 0) (LUMPED-PARAMETER NO) (CONTINUOUS-VOLUME NO)
  (NUMBER-OF-ELEMENTS VERY-LOW) (SUBSTRUCTURING-CAPABILITY NO))
 (LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS VERY-LOW))
 (CONTINUOUS-VOLUME (FINITE-ELEMENTS NO) (GRID NO) (NUMBER-OF-KNOWN-SHAPES VERY-LOW))
 (SHAPE-LIMITATIONS (ESSENTIALLY-NONE NO) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS NO)
  (THICK-SHELLS NO))
 (FLUIDS (TURBULENCE NO) (SHOCK-WAVES NO))
 (USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR YES) (CONTROL-LANGUAGE YES)
  (USER-SPECIFIED-ROUTINES YES) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES NO))
 (LIBRARIES (NUMBER-OF-STORED-SHAPES VERY-LOW) (FORMAT-CONVERSION NO) (GEOMETRIC-CONVERSION NO)
  (TRANSLATE-FOR-OTHER-PROGS NO))
 (GRAPHICS (PRESENT NO) (3D-STRUCTURE NO) (NUMBER-OF-FORMATS VERY-LOW) (PERSPECTIVE NO)
  (HIDDEN-LINE-REMOVAL NO) (SHADING NO))
 (COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
  (MANY-UPGRADES YES) (MANY-COMPUTERS YES))
 (SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS NO) (CODE-TO-SAVE-MEMORY NO))
 (GENERIC-COMPONENTS
  (TEST
   (0.29931754 0.25261575 1.1057416 0.8139257 0.49929065 -0.15005387 0.44082102 0.7174021
    0.38272506 0.8171468 -0.10273352 0.1419791 0.51834434 0.3847705 -0.05369599 0.034388866))
  (ACTUAL (0.5 0.5 1 0.5 0.5 0 1 1 0.5 1 0 0 0.75 1 0 0))
  (CALIBRATED
   (0.5224818 0.48120925 0.95040184 0.5169731 0.56350774 -0.04087875 0.48040545 0.8791658
    0.48304808 0.8217073 0.02167634 0.13011737 0.52870744 0.7828218 0.100862905 -0.042008147)))
 (CALIBRATION-SIZING (ACTUAL-COMPONENTS 56.80058) (TEST-COMPONENTS 22.614279)
  (CALIBRATION-COMPONENTS 49.16212))
 (GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.08293925)))
 (SIZING-CALIBRATION (VALUE (ITERATIONS 199 ERROR 0.05609662)))
 (TEST-SIZING (TEST-COMPONENTS 40.26175) (ACTUAL-COMPONENTS 70.75111)))

(AFILE (TOTAL-SIZE (VALUE 36))
 (DESIGN-CRITERIA
  (VALUE TASK37 METHODS37 OBJECTS37 REPRESENTATION37 GRAPHICS37 USER-INTERFACE37
   DATA-MANAGEMENT37 SYS-PROGRAMMING14 HARD-ARCHITECTURE13))
 (TASK (DATABASE-PROGRAM NO) (ENGINEERING-PROGRAM YES)) (DATABASE-PROGRAM (VIDEO-IMAGES NO))
 (ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING YES) (AERODYNAMICS NO) (ELECTRONIC-CIRCUIT NO)
  (THERMODYNAMICS NO) (IMAGE-PROCESSING NO))
 (THERMODYNAMICS (RADIATION NO) (CONDUCTION NO)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-LOW))
 (TIME-DEPENDENCE (STATIC YES) (QUASI-STATIC NO) (FULLY-DYNAMIC NO))
 (LINEARITY (LINEAR NO) (LIMITED-NON-LINEAR NO) (FULLY-NON-LINEAR NO))
 (BOUNDARY (NUMBER-WAYS-TO-SPECIFY LOW) (2D-FIELD-ON-3D-SURFACE NO) (LUMPED-COMPONENT-VALUES
NO))
 (OBJECTS (NON-PHYSICAL NO) (SOLIDS YES) (FLUIDS NO)) (NON-PHYSICAL (IMAGES NO))
 (SOLIDS (STRUCTURE YES) (POINT-MASS NO))
 (STRUCTURE (SPATIAL-DIMENSION 3) (LUMPED-PARAMETER YES) (CONTINUOUS-VOLUME YES)
  (NUMBER-OF-ELEMENTS LOW) (SUBSTRUCTURING-CAPABILITY NO))
 (LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS MEDIUM))
 (CONTINUOUS-VOLUME (FINITE-ELEMENTS NO) (GRID NO) (NUMBER-OF-KNOWN-SHAPES VERY-LOW))
 (SHAPE-LIMITATIONS (ESSENTIALLY-NONE NO) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS NO)
  (THICK-SHELLS NO))
 (FLUIDS (TURBULENCE NO) (SHOCK-WAVES NO))
 (USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR YES) (CONTROL-LANGUAGE YES)
```

(USER-SPECIFIED-ROUTINES NO) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES YES))
(LIBRARIES (NUMBER-OF-STORED-SHAPES MEDIUM) (FORMAT-CONVERSION NO) (GEOMETRIC-CONVERSION YES)
(TRANSLATE-FOR-OTHER-PROGS NO))
(GRAPHICS (PRESENT YES) (3D-STRUCTURE YES) (NUMBER-OF-FORMATS LOW) (PERSPECTIVE NO)
(HIDDEN-LINE-REMOVAL NO) (SHADING NO))
(COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES NO) (INDUSTRY-STANDARD NO)
(MANY-UPGRADES NO) (MANY-COMPUTERS NO))
(SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS NO) (CODE-TO-SAVE-MEMORY NO))
(GENERIC-COMPONENTS
(TEST
(0.5912612 0.6507159 0.316742 1.0399702 -0.030442625 0.99077046 0.2343714 0.4603753
0.72774976 0.016817018 0.8127148 1.0566217 0.07451641 -0.053072885 0.13188629 0.122585975))
(ACTUAL (0.25 0.25 0.25 0.5 0.5 0 0.5 0.5 0.5 0.5 0.5 0.25 0.25 0.25 0 0))
(CALIBRATED
(0.18394423 0.33524784 0.20940477 0.569829 0.40770948 0.05136521 0.5772465 0.49701798
0.5362124 0.49931735 0.4228416 0.33328694 0.115986116 0.32237685 -0.010831654 -0.10510914)))
(CALIBRATION-SIZING (ACTUAL-COMPONENTS 30.965376) (TEST-COMPONENTS 32.611668)
(CALIBRATION-COMPONENTS 40.390915))
(GENERIC-CALIBRATION (VALUE (ITERATIONS 74 ERROR 0.47531757)))
(SIZING-CALIBRATION (VALUE (ITERATIONS 195 ERROR 0.049997628)))
(TEST-SIZING (TEST-COMPONENTS 45.644943) (ACTUAL-COMPONENTS 36.962128)))

(FPT (TOTAL-SIZE (VALUE 18))
(DESIGN-CRITERIA
(VALUE TASK38 METHODS38 OBJECTS38 REPRESENTATION38 GRAPHICS38 USER-INTERFACE38
DATA-MANAGEMENT38 SYS-PROGRAMMING15 HARD-ARCHITECTURE14))
(TASK (DATABASE-PROGRAM NO) (ENGINEERING-PROGRAM NO)) (DATABASE-PROGRAM (VIDEO-IMAGES NO))
(ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING NO) (AERODYNAMICS NO) (ELECTRONIC-CIRCUIT NO)
(THERMODYNAMICS NO) (CONTROL-SYSTEMS NO) (IMAGE-PROCESSING NO))
(THERMODYNAMICS (RADIATION NO) (CONDUCTION NO)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-LOW))
(TIME-DEPENDENCE (STATIC NO) (QUASI-STATIC NO) (FULLY-DYNAMIC NO))
(LINEARITY (LINEAR NO) (LIMITED-NON-LINEAR NO) (FULLY-NON-LINEAR NO))
(BOUNDARY (NUMBER-WAYS-TO-SPECIFY VERY-LOW) (2D-FIELD-ON-3D-SURFACE NO)
(LUMPED-COMPONENT-VALUES NO))
(OBJECTS (NON-PHYSICAL YES) (SOLIDS NO) (FLUIDS NO)) (NON-PHYSICAL (IMAGES NO))
(SOLIDS (STRUCTURE NO) (POINT-MASS NO))
(STRUCTURE (SPATIAL-DIMENSION 0) (LUMPED-PARAMETER NO) (CONTINUOUS-VOLUME NO)
(NUMBER-OF-ELEMENTS VERY-LOW) (SUBSTRUCTURING-CAPABILITY NO))
(LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS VERY-LOW))
(CONTINUOUS-VOLUME (FINITE-ELEMENTS NO) (GRID NO) (NUMBER-OF-KNOWN-SHAPES VERY-LOW))
(SHAPE-LIMITATIONS (ESSENTIALLY-NONE NO) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS NO)
(THICK-SHELLS NO))
(FLUIDS (TURBULENCE NO) (SHOCK-WAVES NO))
(USER-INTERFACE (TEXT-EDITOR YES) (DATA-MANIPULATOR NO) (CONTROL-LANGUAGE NO)
(USER-SPECIFIED-ROUTINES YES) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES YES))
(LIBRARIES (NUMBER-OF-STORED-SHAPES VERY-LOW) (FORMAT-CONVERSION YES) (GEOMETRIC-CONVERSION NO)
(TRANSLATE-FOR-OTHER-PROGS NO))
(GRAPHICS (PRESENT NO) (3D-STRUCTURE NO) (NUMBER-OF-FORMATS VERY-LOW) (PERSPECTIVE NO)
(HIDDEN-LINE-REMOVAL NO) (SHADING NO))
(COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
(MANY-UPGRADES NO) (MANY-COMPUTERS NO))
(SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS NO) (CODE-TO-SAVE-MEMORY NO))
(GENERIC-COMPONENTS
(TEST
(-0.096894376 0.28895563 0.5463048 0.71393067 0.6352075 0.47627056 0.12243645 0.79603356
0.17642331 0.57324976 0.15664724 0.2856045 0.32144424 0.09286008 0.040235296 -0.11586778))
(ACTUAL (0.25 0.25 0 0.5 0.5 0 0.25 0.5 0.25 0.5 0.5 0 0.25 0 0 0))
(CALIBRATED
(0.34264427 0.32474753 0.24203588 0.72281575 0.3244387 -0.073038585 0.49730873 0.4274279
0.12173728 0.5636444 0.30539685 0.02732306 0.28785312 0.20188113 -0.05578645 0.013750434)))
(CALIBRATION-SIZING (ACTUAL-COMPONENTS 25.256855) (TEST-COMPONENTS 25.674088)
(CALIBRATION-COMPONENTS 18.158997))
(GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.08398175)))
(SIZING-CALIBRATION (VALUE (ITERATIONS 199 ERROR 0.05496426)))
(TEST-SIZING (TEST-COMPONENTS 41.025497) (ACTUAL-COMPONENTS 23.622879)))

(SAP (TOTAL-SIZE (VALUE 12))
(DESIGN-CRITERIA
(VALUE TASK39 METHODS39 OBJECTS39 REPRESENTATION39 GRAPHICS39 USER-INTERFACE39
DATA-MANAGEMENT39 SYS-PROGRAMMING16 HARD-ARCHITECTURE15))
(TASK (DATABASE-PROGRAM NO) (ENGINEERING-PROGRAM NO)) (DATABASE-PROGRAM (VIDEO-IMAGES NO))
(ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING NO) (AERODYNAMICS NO) (ELECTRONIC-CIRCUIT NO)
(THERMODYNAMICS NO) (CONTROL-SYSTEMS NO) (IMAGE-PROCESSING NO))
(THERMODYNAMICS (RADIATION NO) (CONDUCTION NO)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-LOW))
(TIME-DEPENDENCE (STATIC NO) (QUASI-STATIC NO) (FULLY-DYNAMIC NO))
(LINEARITY (LINEAR NO) (LIMITED-NON-LINEAR NO) (FULLY-NON-LINEAR NO))
(BOUNDARY (NUMBER-WAYS-TO-SPECIFY VERY-LOW) (2D-FIELD-ON-3D-SURFACE NO)
(LUMPED-COMPONENT-VALUES NO))
(OBJECTS (NON-PHYSICAL YES) (SOLIDS NO) (FLUIDS NO)) (NON-PHYSICAL (IMAGES NO))
(SOLIDS (STRUCTURE NO) (POINT-MASS NO))
(STRUCTURE (SPATIAL-DIMENSION 0) (LUMPED-PARAMETER NO) (CONTINUOUS-VOLUME NO)

```
(NUMBER-OF-ELEMENTS VERY-LOW) (SUBSTRUCTURING-CAPABILITY NO))
(LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS VERY-LOW))
(CONTINUOUS-VOLUME (FINITE-ELEMENTS NO) (GRID NO) (NUMBER-OF-KNOWN-SHAPES VERY-LOW))
(SHAPE-LIMITATIONS (ESSENTIALLY-NONE NO) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS NO)
 (THICK-SHELLS NO))
(FLUIDS (TURBULENCE NO) (SHOCK-WAVES NO))
(USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR NO) (CONTROL-LANGUAGE NO)
 (USER-SPECIFIED-ROUTINES NO) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES YES))
(LIBRARIES (NUMBER-OF-STORED-SHAPES VERY-LOW) (FORMAT-CONVERSION NO) (GEOMETRIC-CONVERSION NO)
 (TRANSLATE-FOR-OTHER-PROGS NO))
(GRAPHICS (PRESENT NO) (3D-STRUCTURE NO) (NUMBER-OF-FORMATS VERY-LOW) (PERSPECTIVE NO)
 (HIDDEN-LINE-REMOVAL NO) (SHADING NO))
(COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
 (MANY-UPGRADES NO) (MANY-COMPUTERS YES))
(SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS NO) (CODE-TO-SAVE-MEMORY NO))
(GENERIC-COMPONENTS
(TEST
 (0.31109712 0.11023791 -0.24679561 0.45595354 0.63338965 -0.44586706 0.06334501 0.117771894
  0.37051755 0.55778563 0.19828658 0.09558204 0.01038596 0.25065914 -0.22090356 0.0259742))
(ACTUAL (0 0.25 0.25 0.5 0.5 0 0.25 0.25 0 0.5 0 0 0 0 0 0))
 (CALIBRATED
 (0.27011406 0.29901445 0.1625885 0.5165067 0.5794081 -0.18957064 0.20523474 0.39585125
  0.2611137 0.57214254 0.105254695 -0.15233816 0.1694705 0.12578756 0.037585184 -0.13264482)))
(CALIBRATION-SIZING (ACTUAL-COMPONENTS 11.038157) (TEST-COMPONENTS 18.985033)
 (CALIBRATION-COMPONENTS 10.353344))
(GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.082792476)))
(SIZING-CALIBRATION (VALUE (ITERATIONS 199 ERROR 0.06063182)))
(TEST-SIZING (TEST-COMPONENTS 29.048874) (ACTUAL-COMPONENTS 17.927038)))

(DISCOS (TOTAL-SIZE (VALUE 20))
(DESIGN-CRITERIA
 (VALUE TASK40 METHODS40 OBJECTS40 REPRESENTATION40 GRAPHICS40 USER-INTERFACE40
  DATA-MANAGEMENT40 SYS-PROGRAMMING17 HARD-ARCHITECTURE16))
(TASK (DATABASE-PROGRAM NO) (ENGINEERING-PROGRAM YES)) (DATABASE-PROGRAM (VIDEO-IMAGES NO))
(ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING NO) (AERODYNAMICS NO) (ELECTRONIC-CIRCUIT NO)
 (THERMODYNAMICS NO) (CONTROL-SYSTEMS YES) (IMAGE-PROCESSING NO))
(THERMODYNAMICS (RADIATION NO) (CONDUCTION NO)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-LOW))
(TIME-DEPENDENCE (STATIC NO) (QUASI-STATIC NO) (FULLY-DYNAMIC YES))
(LINEARITY (LINEAR NO) (LIMITED-NON-LINEAR NO) (FULLY-NON-LINEAR YES))
(BOUNDARY (NUMBER-WAYS-TO-SPECIFY VERY-LOW) (2D-FIELD-ON-3D-SURFACE NO)
 (LUMPED-COMPONENT-VALUES YES))
(OBJECTS (NON-PHYSICAL NO) (SOLIDS YES) (FLUIDS NO)) (NON-PHYSICAL (IMAGES NO))
(SOLIDS (STRUCTURE YES) (POINT-MASS NO))
(STRUCTURE (SPATIAL-DIMENSION 3) (LUMPED-PARAMETER YES) (CONTINUOUS-VOLUME NO)
 (NUMBER-OF-ELEMENTS LOW) (SUBSTRUCTURING-CAPABILITY NO))
(LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS MEDIUM))
(CONTINUOUS-VOLUME (FINITE-ELEMENTS NO) (GRID NO) (NUMBER-OF-KNOWN-SHAPES VERY-LOW))
(SHAPE-LIMITATIONS (ESSENTIALLY-NONE YES) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS NO)
 (THICK-SHELLS NO))
(FLUIDS (TURBULENCE NO) (SHOCK-WAVES NO))
(USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR NO) (CONTROL-LANGUAGE NO)
 (USER-SPECIFIED-ROUTINES YES) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES NO))
(LIBRARIES (NUMBER-OF-STORED-SHAPES MEDIUM) (FORMAT-CONVERSION YES) (GEOMETRIC-CONVERSION NO)
 (TRANSLATE-FOR-OTHER-PROGS YES))
(GRAPHICS (PRESENT YES) (3D-STRUCTURE NO) (NUMBER-OF-FORMATS VERY-LOW) (PERSPECTIVE NO)
 (HIDDEN-LINE-REMOVAL NO) (SHADING NO))
(COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
 (MANY-UPGRADES YES) (MANY-COMPUTERS YES))
(SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS NO) (CODE-TO-SAVE-MEMORY NO))
(GENERIC-COMPONENTS
(TEST
 (1.1148465 1.0000348 0.5173342 0.7496426 -0.0057442524 0.5741864 1.1926603 0.71048063
  0.1994492 0.69778895 -0.0150009245 -0.30467254 0.20625187 0.6814102 0.65108943 -0.04486663))
(ACTUAL (0.5 0.75 0.75 1 0 0 0.5 0.5 0.5 0.25 0.5 0.25 0 0 0 0))
 (CALIBRATED
 (0.52889 0.68612957 0.7439897 0.90554345 0.05448331 -0.024491832 0.5176872 0.5113161
  0.44871336 0.2305316 0.563324 0.08382335 0.19684431 0.0306089 -0.0052065924 0.02052708)))
(CALIBRATION-SIZING (ACTUAL-COMPONENTS 18.664589) (TEST-COMPONENTS 59.22189)
 (CALIBRATION-COMPONENTS 21.401623))
(GENERIC-CALIBRATION (VALUE (ITERATIONS 197 ERROR 0.182)))
(SIZING-CALIBRATION (VALUE (ITERATIONS 22 ERROR 0.099680424)))
(TEST-SIZING (TEST-COMPONENTS 48.065277) (ACTUAL-COMPONENTS 44.842545))
(SIZING-LEARNING (VALUE (ETA 0.25 ALPHA 0.5 MAX-ITER 100 MAX-ERR 0.1))))

(DIDS (TOTAL-SIZE (VALUE 40))
(DESIGN-CRITERIA
 (VALUE TASK30 METHODS30 OBJECTS30 REPRESENTATION30 GRAPHICS30 USER-INTERFACE30
  DATA-MANAGEMENT30 SYS-PROGRAMMING8 HARD-ARCHITECTURE7))
(TASK (DATABASE-PROGRAM YES) (ENGINEERING-PROGRAM NO)) (DATABASE-PROGRAM (VIDEO-IMAGES YES))
(ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING NO) (AERODYNAMICS NO) (ELECTRONIC-CIRCUIT NO)
 (THERMODYNAMICS NO) (CONTROL-SYSTEMS NO) (IMAGE-PROCESSING NO))
```

```
(THERMODYNAMICS (RADIATION NO) (CONDUCTION NO)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-LOW))
(TIME-DEPENDENCE (STATIC NO) (QUASI-STATIC NO) (FULLY-DYNAMIC NO))
(LINEARITY (LINEAR NO) (LIMITED-NON-LINEAR NO) (FULLY-NON-LINEAR NO))
(BOUNDARY (NUMBER-WAYS-TO-SPECIFY VERY-LOW) (2D-FIELD-ON-3D-SURFACE NO)
 (LUMPED-COMPONENT-VALUES NO))
(OBJECTS (NON-PHYSICAL YES) (SOLIDS NO) (FLUIDS NO)) (NON-PHYSICAL (IMAGES YES))
(SOLIDS (STRUCTURE NO) (POINT-MASS NO))
(STRUCTURE (SPATIAL-DIMENSION 0) (LUMPED-PARAMETER NO) (CONTINUOUS-VOLUME NO)
 (NUMBER-OF-ELEMENTS VERY-LOW) (SUBSTRUCTURING-CAPABILITY NO))
(LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS VERY-LOW))
(CONTINUOUS-VOLUME (FINITE-ELEMENTS NO) (GRID NO) (NUMBER-OF-KNOWN-SHAPES VERY-LOW))
(SHAPE-LIMITATIONS (ESSENTIALLY-NONE NO) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS NO)
 (THICK-SHELLS NO))
(FLUIDS (TURBULENCE NO) (SHOCK-WAVES NO))
(USER-INTERFACE (TEXT-EDITOR YES) (DATA-MANIPULATOR YES) (CONTROL-LANGUAGE NO)
 (USER-SPECIFIED-ROUTINES NO) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES YES))
(LIBRARIES (NUMBER-OF-STORED-SHAPES VERY-HIGH) (FORMAT-CONVERSION YES)
 (GEOMETRIC-CONVERSION NO) (TRANSLATE-FOR-OTHER-PROGS NO))
(GRAPHICS (PRESENT YES) (3D-STRUCTURE NO) (NUMBER-OF-FORMATS VERY-HIGH) (PERSPECTIVE NO)
 (HIDDEN-LINE-REMOVAL NO) (SHADING NO))
(COMMERCIALIZATION (MANY-PROJECTS NO) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
 (MANY-UPGRADES NO) (MANY-COMPUTERS NO))
(SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS YES) (CODE-TO-SAVE-MEMORY YES))
(GENERIC-COMPONENTS
(TEST
 (0.14639632 0.07108798 0.48942095 0.53922176 0.38841677 0.10720483 0.7808202 0.7021467
  -0.2391152 1.2843144 0.8203115 0.6118797 0.82239544 0.5827514 -0.22501843 0.021385536))
(ACTUAL (0.25 0.25 0.5 0.5 0.5 0 0.25 0.25 0.25 0.5 0.5 0.5 0.5 0.5 0.5 0.25))
 (CALIBRATED
 (0.30686426 0.29655886 0.5785048 0.59687114 0.5099994 0.09605509 0.27158037 0.58472776
  0.06338241 0.62761915 0.6322575 0.4851675 0.59456056 0.68700874 0.3518101 0.3074916)))
(CALIBRATION-SIZING (ACTUAL-COMPONENTS 52.88096) (TEST-COMPONENTS 43.20947)
 (CALIBRATION-COMPONENTS 48.67561))
(GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.09579681)))
(SIZING-CALIBRATION (VALUE (ITERATIONS 199 ERROR 0.054026626)))
(TEST-SIZING (TEST-COMPONENTS 41.44719) (ACTUAL-COMPONENTS 57.512726)))

(SAMSAN (TOTAL-SIZE (VALUE 20))
 (DESIGN-CRITERIA
 (VALUE TASK41 METHODS41 OBJECTS41 REPRESENTATION41 GRAPHICS41 USER-INTERFACE41
   DATA-MANAGEMENT41 SYS-PROGRAMMING18 HARD-ARCHITECTURE17))
 (TASK (DATABASE-PROGRAM NO) (ENGINEERING-PROGRAM YES)) (DATABASE-PROGRAM (VIDEO-IMAGES NO))
 (ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING NO) (AERODYNAMICS NO) (ELECTRONIC-CIRCUIT NO)
  (THERMODYNAMICS NO) (CONTROL-SYSTEMS YES) (IMAGE-PROCESSING NO))
 (THERMODYNAMICS (RADIATION NO) (CONDUCTION NO)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-LOW))
 (TIME-DEPENDENCE (STATIC NO) (QUASI-STATIC YES) (FULLY-DYNAMIC NO))
 (LINEARITY (LINEAR YES) (LIMITED-NON-LINEAR NO) (FULLY-NON-LINEAR NO))
 (BOUNDARY (NUMBER-WAYS-TO-SPECIFY VERY-LOW) (2D-FIELD-ON-3D-SURFACE NO)
  (LUMPED-COMPONENT-VALUES NO))
 (OBJECTS (NON-PHYSICAL YES) (SOLIDS NO) (FLUIDS NO)) (NON-PHYSICAL (IMAGES NO))
 (SOLIDS (STRUCTURE NO) (POINT-MASS NO))
 (STRUCTURE (SPATIAL-DIMENSION 0) (LUMPED-PARAMETER NO) (CONTINUOUS-VOLUME NO)
  (NUMBER-OF-ELEMENTS VERY-LOW) (SUBSTRUCTURING-CAPABILITY NO))
 (LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS VERY-LOW))
 (CONTINUOUS-VOLUME (FINITE-ELEMENTS NO) (GRID NO) (NUMBER-OF-KNOWN-SHAPES VERY-LOW))
 (SHAPE-LIMITATIONS (ESSENTIALLY-NONE NO) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS NO)
  (THICK-SHELLS NO))
 (FLUIDS (TURBULENCE NO) (SHOCK-WAVES NO))
 (USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR YES) (CONTROL-LANGUAGE NO)
  (USER-SPECIFIED-ROUTINES YES) (MATRIX-SPEC-LANGUAGE YES) (LIBRARIES YES))
 (LIBRARIES (NUMBER-OF-STORED-SHAPES VERY-LOW) (FORMAT-CONVERSION NO) (GEOMETRIC-CONVERSION NO)
  (TRANSLATE-FOR-OTHER-PROGS YES))
 (GRAPHICS (PRESENT NO) (3D-STRUCTURE NO) (NUMBER-OF-FORMATS VERY-LOW) (PERSPECTIVE NO)
  (HIDDEN-LINE-REMOVAL NO) (SHADING NO))
 (COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
  (MANY-UPGRADES YES) (MANY-COMPUTERS NO))
 (SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS NO) (CODE-TO-SAVE-MEMORY NO))
 (GENERIC-COMPONENTS
 (TEST
  (1.1058773 0.36601207 0.8062235 0.86513126 0.43998566 0.61421704 0.4760372 0.41716897
   0.3853284 0.66234595 0.8731527 -0.17886399 0.44287872 0.4979682 -0.54288346 -0.1517324))
 (ACTUAL (0.5 0.5 0.5 0.5 0.5 0 0.5 0.5 0.5 0 0 0 0 0 0 0))
  (CALIBRATED
  (0.57433087 0.54676783 0.6664868 0.59145844 0.5257423 0.40629125 0.5494422 0.6108882
   0.7150388 0.2780841 0.18715663 0.08116226 0.07071208 0.1632645 -0.06546661 -0.080140844)))
 (CALIBRATION-SIZING (ACTUAL-COMPONENTS 20.125961) (TEST-COMPONENTS 46.121834)
  (CALIBRATION-COMPONENTS 17.024487))
 (GENERIC-CALIBRATION (VALUE (ITERATIONS 53 ERROR 0.58417714)))
 (SIZING-CALIBRATION (VALUE (ITERATIONS 199 ERROR 0.054569155)))
 (TEST-SIZING (TEST-COMPONENTS 43.572563) (ACTUAL-COMPONENTS 23.292809)))
```

```
(SPAR (TOTAL-SIZE (VALUE 87))
 (DESIGN-CRITERIA
  (VALUE TASK42 METHODS42 OBJECTS42 REPRESENTATION42 GRAPHICS42 USER-INTERFACE42
   DATA-MANAGEMENT42 SYS-PROGRAMMING19 HARD-ARCHITECTURE18))
 (TASK (DATABASE-PROGRAM YES) (ENGINEERING-PROGRAM YES)) (DATABASE-PROGRAM (VIDEO-IMAGES NO))
 (ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING YES) (AERODYNAMICS NO) (ELECTRONIC-CIRCUIT NO)
  (THERMODYNAMICS YES) (CONTROL-SYSTEMS NO) (IMAGE-PROCESSING NO))
 (THERMODYNAMICS (RADIATION NO) (CONDUCTION YES)) (METHODS (NUMBER-REDUNDANT-CHOICES MEDIUM))
 (TIME-DEPENDENCE (STATIC YES) (QUASI-STATIC YES) (FULLY-DYNAMIC NO))
 (LINEARITY (LINEAR YES) (LIMITED-NON-LINEAR YES) (FULLY-NON-LINEAR NO))
 (BOUNDARY (NUMBER-WAYS-TO-SPECIFY VERY-LOW) (2D-FIELD-ON-3D-SURFACE YES)
  (LUMPED-COMPONENT-VALUES NO))
 (OBJECTS (NON-PHYSICAL NO) (SOLIDS YES) (FLUIDS YES)) (NON-PHYSICAL (IMAGES NO))
 (SOLIDS (STRUCTURE YES) (POINT-MASS NO))
 (STRUCTURE (SPATIAL-DIMENSION 3) (LUMPED-PARAMETER NO) (CONTINUOUS-VOLUME YES)
  (NUMBER-OF-ELEMENTS VERY-HIGH) (SUBSTRUCTURING-CAPABILITY YES))
 (LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS VERY-LOW))
 (CONTINUOUS-VOLUME (FINITE-ELEMENTS YES) (GRID NO) (NUMBER-OF-KNOWN-SHAPES HIGH))
 (SHAPE-LIMITATIONS (ESSENTIALLY-NONE YES) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS NO)
  (THICK-SHELLS NO))
 (FLUIDS (TURBULENCE NO) (SHOCK-WAVES NO))
 (USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR YES) (CONTROL-LANGUAGE YES)
  (USER-SPECIFIED-ROUTINES NO) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES YES))
 (LIBRARIES (NUMBER-OF-STORED-SHAPES MEDIUM) (FORMAT-CONVERSION YES) (GEOMETRIC-CONVERSION NO)
  (TRANSLATE-FOR-OTHER-PROGS NO))
 (GRAPHICS (PRESENT YES) (3D-STRUCTURE YES) (NUMBER-OF-FORMATS VERY-LOW) (PERSPECTIVE YES)
  (HIDDEN-LINE-REMOVAL NO) (SHADING NO))
 (COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
  (MANY-UPGRADES YES) (MANY-COMPUTERS YES))
 (SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS NO) (CODE-TO-SAVE-MEMORY NO))
 (GENERIC-COMPONENTS
  (TEST
   (0.03504813 1.2590215 0.40161705 0.730369 0.75442576 0.66420305 0.34104657 1.047965 0.9790008
    0.75297046 0.05610972 0.2931307 0.15534315 1.0323933 0.04892406 0.20627415))
  (ACTUAL (1 1 1 0.75 0.5 0.5 1 1 0.75 1 0.75 0.5 0.5 0.75 0 0))
  (CALIBRATED
   (1.0041741 1.0033312 1.000428 0.7439978 0.41671985 0.50309867 1.0333672 0.93818665 0.73952866
    1.0158776 0.7182078 0.5205191 0.5748605 0.7503561 0.11815056 0.1528223)))
 (CALIBRATION-SIZING (ACTUAL-COMPONENTS 156.4173) (TEST-COMPONENTS 127.77358)
  (CALIBRATION-COMPONENTS 72.96775))
 (GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.10505067)))
 (SIZING-CALIBRATION (VALUE (ITERATIONS 199 ERROR 0.05009777)))
 (TEST-SIZING (TEST-COMPONENTS 84.55742) (ACTUAL-COMPONENTS 108.66265)))

(DYLOFLEX (TOTAL-SIZE (VALUE 124))
 (DESIGN-CRITERIA
  (VALUE TASK4 METHODS4 OBJECTS4 REPRESENTATION4 GRAPHICS4 USER-INTERFACE4 DATA-MANAGEMENT4))
 (TASK (DATABASE-PROGRAM NO) (ENGINEERING-PROGRAM YES)) (DATABASE-PROGRAM (VIDEO-IMAGES NO))
 (ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING YES) (AERODYNAMICS YES) (ELECTRONIC-CIRCUIT NO)
  (THERMODYNAMICS NO) (CONTROL-SYSTEMS YES) (IMAGE-PROCESSING NO))
 (THERMODYNAMICS (RADIATION NO) (CONDUCTION NO)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-LOW))
 (TIME-DEPENDENCE (STATIC NO) (QUASI-STATIC YES) (FULLY-DYNAMIC NO))
 (LINEARITY (LINEAR NO) (LIMITED-NON-LINEAR NO) (FULLY-NON-LINEAR YES))
 (BOUNDARY (NUMBER-WAYS-TO-SPECIFY MEDIUM) (2D-FIELD-ON-3D-SURFACE YES)
  (LUMPED-COMPONENT-VALUES YES))
 (OBJECTS (NON-PHYSICAL NO) (SOLIDS YES) (FLUIDS YES)) (NON-PHYSICAL (IMAGES NO))
 (SOLIDS (STRUCTURE YES) (POINT-MASS YES))
 (STRUCTURE (SPATIAL-DIMENSION 3) (LUMPED-PARAMETER YES) (CONTINUOUS-VOLUME YES)
  (NUMBER-OF-ELEMENTS LOW) (SUBSTRUCTURING-CAPABILITY YES))
 (LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS LOW))
 (CONTINUOUS-VOLUME (FINITE-ELEMENTS NO) (GRID YES) (NUMBER-OF-KNOWN-SHAPES MEDIUM))
 (SHAPE-LIMITATIONS (ESSENTIALLY-NONE YES) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS NO)
  (THICK-SHELLS NO))
 (FLUIDS (TURBULENCE YES) (SHOCK-WAVES YES))
 (USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR NO) (CONTROL-LANGUAGE NO)
  (USER-SPECIFIED-ROUTINES NO) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES NO))
 (LIBRARIES (NUMBER-OF-STORED-SHAPES MEDIUM) (FORMAT-CONVERSION YES) (GEOMETRIC-CONVERSION YES)
  (TRANSLATE-FOR-OTHER-PROGS YES))
 (GRAPHICS (PRESENT NO) (3D-STRUCTURE NO) (NUMBER-OF-FORMATS VERY-LOW) (PERSPECTIVE NO)
  (HIDDEN-LINE-REMOVAL NO) (SHADING NO))
 (COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
  (MANY-UPGRADES NO) (MANY-COMPUTERS NO))
 (SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS NO) (CODE-TO-SAVE-MEMORY NO))
 (GENERIC-COMPONENTS
  (TEST
   (0.9263165 0.1870482 0.47051477 0.617027 0.45919302 -0.8150745 0.5141034 0.8209049 1.4907271
    0.4658824 1.1072906 0.5891058 -0.09071378 0.27873105 -0.024999112 0.06996849))
  (ACTUAL (1 1 0.75 0.75 0.5 0.5 1 0.75 1 0.75 0.75 0 0.25 0.75 0 0))
  (CALIBRATED
   (1.0427557 0.91840875 0.70534444 0.68972194 0.57682365 0.25608653 0.83364975 0.7305551
    0.92179424 0.6618962 0.74480116 0.11216675 0.27894294 0.63066804 0.053990915 0.05790077)))
```

```
(CALIBRATION-SIZING (ACTUAL-COMPONENTS 126.39902) (TEST-COMPONENTS 69.85614)
 (CALIBRATION-COMPONENTS 135.52574))
(GENERIC-CALIBRATION (VALUE (ITERATIONS 25 ERROR 0.8684077)))
(SIZING-CALIBRATION (VALUE (ITERATIONS 30 ERROR 0.10438016)))
(TEST-SIZING (TEST-COMPONENTS 50.75594) (ACTUAL-COMPONENTS 49.395554))
(SIZING-LEARNING (VALUE (ETA 0.02 ALPHA 0.9 MAX-ITER 100 MAX-ERR 0.1)))
(GENERIC-LEARNING (VALUE (ETA 0.002 MAX-ITER 350 MAX-ERR 0.1))))

(FLEXSTAB (TOTAL-SIZE (VALUE 100))
 (DESIGN-CRITERIA
  (VALUE TASK7 METHODS7 OBJECTS7 REPRESENTATION7 GRAPHICS7 USER-INTERFACE7 DATA-MANAGEMENT7))
 (TASK (DATABASE-PROGRAM NO) (ENGINEERING-PROGRAM YES)) (DATABASE-PROGRAM (VIDEO-IMAGES NO))
 (ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING YES) (AERODYNAMICS YES) (ELECTRONIC-CIRCUIT NO)
  (THERMODYNAMICS NO) (CONTROL-SYSTEMS YES) (IMAGE-PROCESSING NO))
 (THERMODYNAMICS (RADIATION NO) (CONDUCTION NO)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-LOW))
 (TIME-DEPENDENCE (STATIC NO) (QUASI-STATIC YES) (FULLY-DYNAMIC NO))
 (LINEARITY (LINEAR NO) (LIMITED-NON-LINEAR YES) (FULLY-NON-LINEAR NO))
 (BOUNDARY (NUMBER-WAYS-TO-SPECIFY MEDIUM) (2D-FIELD-ON-3D-SURFACE YES)
  (LUMPED-COMPONENT-VALUES YES))
 (OBJECTS (NON-PHYSICAL NO) (SOLIDS YES) (FLUIDS YES)) (NON-PHYSICAL (IMAGES NO))
 (SOLIDS (STRUCTURE YES) (POINT-MASS NO))
 (STRUCTURE (SPATIAL-DIMENSION 3) (LUMPED-PARAMETER NO) (CONTINUOUS-VOLUME YES)
  (NUMBER-OF-ELEMENTS VERY-HIGH) (SUBSTRUCTURING-CAPABILITY NO))
 (LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS VERY-LOW))
 (CONTINUOUS-VOLUME (FINITE-ELEMENTS NO) (GRID YES) (NUMBER-OF-KNOWN-SHAPES MEDIUM))
 (SHAPE-LIMITATIONS (ESSENTIALLY-NONE YES) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS NO)
  (THICK-SHELLS NO))
 (FLUIDS (TURBULENCE YES) (SHOCK-WAVES NO))
 (USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR NO) (CONTROL-LANGUAGE YES)
  (USER-SPECIFIED-ROUTINES YES) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES YES))
 (LIBRARIES (NUMBER-OF-STORED-SHAPES MEDIUM) (FORMAT-CONVERSION NO) (GEOMETRIC-CONVERSION NO)
  (TRANSLATE-FOR-OTHER-PROGS NO))
 (GRAPHICS (PRESENT YES) (3D-STRUCTURE NO) (NUMBER-OF-FORMATS LOW) (PERSPECTIVE NO)
  (HIDDEN-LINE-REMOVAL NO) (SHADING NO))
 (COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD NO)
  (MANY-UPGRADES YES) (MANY-COMPUTERS YES))
 (SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS NO) (CODE-TO-SAVE-MEMORY YES))
 (GENERIC-COMPONENTS
 (TEST
  (0.13249312 0.5457227 0.8276131 0.66837287 0.86339307 0.7083652 1.378714 0.6317245 0.15991887
   -0.15189558 0.105110645 0.83321035 0.9477229 -0.19191897 0.30306646 0.10110332))
 (ACTUAL (1 0.5 0.5 0.5 0.75 0 0.75 0.5 0.75 1 0.75 0.5 0.25 0.25 0 0))
  (CALIBRATED
  (0.8620417 0.5752711 0.5530717 0.6544486 0.7489988 0.0864477 0.79158473 0.42447326 0.8352638
   0.9967352 0.6929237 0.48220032 0.40455008 0.3281815 0.01920189 -0.038586088)))
 (CALIBRATION-SIZING (ACTUAL-COMPONENTS 110.91214) (TEST-COMPONENTS 68.3348)
  (CALIBRATION-COMPONENTS 79.2085))
 (GENERIC-CALIBRATION (VALUE (ITERATIONS 200 ERROR 0.091643915)))
 (SIZING-CALIBRATION (VALUE (ITERATIONS 199 ERROR 0.050526295)))
 (TEST-SIZING (TEST-COMPONENTS 45.657494) (ACTUAL-COMPONENTS 63.9332)))

(BIG-ENG (TOTAL-SIZE (VALUE 500))
 (DESIGN-CRITERIA
  (VALUE TASK43 METHODS43 OBJECTS43 REPRESENTATION43 GRAPHICS43 USER-INTERFACE43
   DATA-MANAGEMENT43))
 (TASK (ENGINEERING-PROGRAM YES))
 (ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING YES) (ELECTRONIC-CIRCUIT NO) (THERMODYNAMICS YES)
  (CONTROL-SYSTEMS YES))
 (THERMODYNAMICS (RADIATION YES) (CONDUCTION YES))
 (METHODS (NUMBER-REDUNDANT-CHOICES VERY-HIGH))
 (TIME-DEPENDENCE (STATIC YES) (QUASI-STATIC YES) (FULLY-DYNAMIC YES))
 (LINEARITY (LINEAR YES) (LIMITED-NON-LINEAR YES) (FULLY-NON-LINEAR YES))
 (BOUNDARY (NUMBER-WAYS-TO-SPECIFY VERY-HIGH) (2D-FIELD-ON-3D-SURFACE YES)
  (LUMPED-COMPONENT-VALUES YES))
 (OBJECTS (SOLIDS YES) (FLUIDS YES)) (SOLIDS (STRUCTURE YES) (POINT-MASS YES))
 (STRUCTURE (SPATIAL-DIMENSION 3) (LUMPED-PARAMETER NO) (CONTINUOUS-VOLUME YES)
  (NUMBER-OF-ELEMENTS VERY-HIGH) (SUBSTRUCTURING-CAPABILITY YES))
 (CONTINUOUS-VOLUME (FINITE-ELEMENTS YES) (GRID NO) (NUMBER-OF-KNOWN-SHAPES VERY-HIGH))
 (SHAPE-LIMITATIONS (ESSENTIALLY-NONE YES) (THIN-SHELLS YES) (SHELLS-OF-REVOL YES) (SLABS YES)
  (THICK-SHELLS YES))
 (FLUIDS (TURBULENCE NO) (SHOCK-WAVES NO))
 (USER-INTERFACE (TEXT-EDITOR YES) (DATA-MANIPULATOR YES) (CONTROL-LANGUAGE YES)
  (USER-SPECIFIED-ROUTINES YES) (MATRIX-SPEC-LANGUAGE YES) (LIBRARIES YES))
 (LIBRARIES (NUMBER-OF-STORED-SHAPES VERY-HIGH) (FORMAT-CONVERSION YES)
  (GEOMETRIC-CONVERSION YES) (TRANSLATE-FOR-OTHER-PROGS YES))
 (GRAPHICS (PRESENT YES) (3D-STRUCTURE YES) (NUMBER-OF-FORMATS VERY-HIGH) (PERSPECTIVE YES)
  (HIDDEN-LINE-REMOVAL YES) (SHADING YES))
 (COMMERCIALIZATION (MANY-PROJECTS YES) (MANY-COMPANIES YES) (INDUSTRY-STANDARD YES)
  (MANY-UPGRADES YES) (MANY-COMPUTERS YES))
 (GENERIC-COMPONENTS
  (CALIBRATED
```

```
(0.9411489 0.910926 1.3921202 0.7596223 0.7819247 0.8644963 1.1426878 1.0394086 1.1071641
  0.857352 1.3951648 1.0565795 0.8187169 0.2725457 -0.010894686 -0.045779258)))
 (CALIBRATION-SIZING (CALIBRATION-COMPONENTS 223.41252)))

(SCIENCE-ONLY (TOTAL-SIZE (VALUE 10))
 (DESIGN-CRITERIA
  (VALUE TASK44 METHODS44 OBJECTS44 REPRESENTATION44 GRAPHICS44 USER-INTERFACE44
   DATA-MANAGEMENT44 SYS-PROGRAMMING20 HARD-ARCHITECTURE19))
  (TASK (DATABASE-PROGRAM NO) (ENGINEERING-PROGRAM YES)) (DATABASE-PROGRAM (VIDEO-IMAGES NO))
  (ENGINEERING-PROGRAM (STRUCTURAL-ENGINEERING NO) (AERODYNAMICS NO) (ELECTRONIC-CIRCUIT NO)
  (THERMODYNAMICS NO) (CONTROL-SYSTEMS NO) (IMAGE-PROCESSING NO))
  (THERMODYNAMICS (RADIATION NO) (CONDUCTION NO)) (METHODS (NUMBER-REDUNDANT-CHOICES VERY-LOW))
  (TIME-DEPENDENCE (STATIC NO) (QUASI-STATIC NO) (FULLY-DYNAMIC YES))
  (LINEARITY (LINEAR NO) (LIMITED-NON-LINEAR NO) (FULLY-NON-LINEAR YES))
  (BOUNDARY (NUMBER-WAYS-TO-SPECIFY LOW) (2D-FIELD-ON-3D-SURFACE NO)
   (LUMPED-COMPONENT-VALUES YES))
  (OBJECTS (NON-PHYSICAL NO) (SOLIDS YES) (FLUIDS NO)) (NON-PHYSICAL (IMAGES NO))
  (SOLIDS (STRUCTURE YES) (POINT-MASS NO))
  (STRUCTURE (SPATIAL-DIMENSION 3) (LUMPED-PARAMETER YES) (CONTINUOUS-VOLUME NO)
   (NUMBER-OF-ELEMENTS LOW) (SUBSTRUCTURING-CAPABILITY NO))
  (LUMPED-PARAMETER (NUMBER-KNOWN-COMPONENTS MEDIUM))
  (CONTINUOUS-VOLUME (FINITE-ELEMENTS NO) (GRID NO) (NUMBER-OF-KNOWN-SHAPES VERY-LOW))
  (SHAPE-LIMITATIONS (ESSENTIALLY-NONE YES) (THIN-SHELLS NO) (SHELLS-OF-REVOL NO) (SLABS NO)
   (THICK-SHELLS NO))
  (FLUIDS (TURBULENCE NO) (SHOCK-WAVES NO))
  (USER-INTERFACE (TEXT-EDITOR NO) (DATA-MANIPULATOR NO) (CONTROL-LANGUAGE NO)
   (USER-SPECIFIED-ROUTINES NO) (MATRIX-SPEC-LANGUAGE NO) (LIBRARIES NO))
  (LIBRARIES (NUMBER-OF-STORED-SHAPES LOW) (FORMAT-CONVERSION NO) (GEOMETRIC-CONVERSION NO)
   (TRANSLATE-FOR-OTHER-PROGS NO))
  (GRAPHICS (PRESENT NO) (3D-STRUCTURE NO) (NUMBER-OF-FORMATS VERY-LOW) (PERSPECTIVE NO)
   (HIDDEN-LINE-REMOVAL NO) (SHADING NO))
  (COMMERCIALIZATION (MANY-PROJECTS NO) (MANY-COMPANIES NO) (INDUSTRY-STANDARD NO)
   (MANY-UPGRADES NO) (MANY-COMPUTERS NO))
  (SYSTEMS-PROGRAMMING (EXECUTE-COMMANDS NO) (CODE-TO-SAVE-MEMORY NO))
  (GENERIC-COMPONENTS
  (CALIBRATED
   (0.06871896 0.30382785 0.1289039 0.8055429 0.2363187 -0.13235594 0.2682919 0.29011923
    0.20058946 -0.29440707 0.24225022 0.5418814 -0.14628364 -0.12068774 0.11227903 0.20935439)))
  (CALIBRATION-SIZING (CALIBRATION-COMPONENTS 12.197316)))
```

Appendix 5 - Summaries of Expert Interviews

The domain experts were taken from the University of California, Santa Barbara. Professor Yoshihiko Nakamura of the Robotics Laboratory is an expert in the design of software for mechanical control systems. Interviews were conducted with him were conducted from February through March of 1989. Mr. David Girardot of the Chemical & Nuclear Engineering department provided expertise in the area of software for process control. Interviews were conducted with him were conducted during April and May of 1989. One purpose of the interviews was to determine the knowledge structures needed to represent real time programs. The interviews suggest that the strategies of the Code Sizing Tool will work with real time programs when they become available for analysis in the field testing program. Summaries of the interviews with the domain experts are given below.

2.2.2.2.1 Professor Yohihiko Nakamura

The structures and techniques for knowledge acquisition and representation were formalized on knowledge from the design of a robot arm developed by Professor Nakamura. A set of examples are included in this section. Figure 5.5 shows a classification scheme for real time programs. There are three types of real time programs at the first level:

- Control Systems
- Data Collection
- Computer Operating System

The control system is distinguished from the data collection systems in that it controls actuator devices that act on the outside world. These systems typically have a relatively high level of feedback and a relatively low bandwidth. Data collection systems are designed to observe the outside world rather than act on it. They have little or no feedback and a typically high bandwidth.

Figure 5.6 shows a generalized architecture for a real time program. Not all real time programs will have all of the components represented. In particular, the data collection system will not, in general, contain a control system since it does not necessarily employ feedback. A brief definition of each components is given below:

- Top Level Control: Selects among various possible modes of the system and can provide global services such as initialization and scheduling.

- User Interface: The part of the system that interacts with the user. It is present in flexible real time systems that can be programmed for different functions or in systems that serve monitoring function.

- Task Oriented Routines: A hierarchy of routines that contain the application oriented logic of the system.

- The System Model: In control systems, this component transforms the variables that are measured into the variables needed to control the system.

- The Control System: This component provides a feedback mechanism by comparing the measured state with the desired state and estimating actuator commands that will move the system toward the desired state.

- Sensor Processing: For systems with large amounts of complex data, such as vision systems, a separate component is provided to numerically process the data before it is applied to any higher level components.

- Device Drivers: These components translate functional commands into primitive hardware commands for the computer peripherals.

- Diagnostics: This component is used to test for error conditions in the software or malfunctions of the hardware and notify the user of the system.

- Error Handling: This component is designed to correct errors or other abnormal conditions detected by the diagnostic modules.

Figure 5.7 shows an architecture for a simple robotic system with Fortran code estimates for each component. This system is constrained to operate with only three degrees of freedom and was designed for a research environment. In this case, there are no diagnostic or error handling components. There are two levels of task oriented routines. At the higher level, there is one component for each of two modes of the system, the TEACH mode and the RUN mode. The (essentially trivial) user interface is contained in the teach mode component. At the lower level, there is a set of task oriented routines that make up the basic capabilities or "vocabulary" of the robot. The vocabulary contains commands such as INITIALIZE, MOVE, GRASP, and RELEASE, which comprise the basic capabilities of the system. It appears that this organization of the task oriented routines applies to a wide variety of programs. This particular robot was designed in a research environment to test advanced control algorithms. The system model (ie, the Dynamics and Kinematics) is larger than would be the case in constrained commercial robots, and is estimated at 500 lines of Fortran code. The actual control system module is relatively simple and is estimated as a single page (50 lines) of code. The device drivers shown are typical for a robot arm: a D to A (digital to analogue) converter to control the motors, an A to D (analogue to digital) converter for the force sensor for the robot hand, an encoder-counter to read the angular displacement of the joints, and a device driver for the Teach Pendent, the user interface device for teaching the robot arm.

Table 5.1 shows how the code estimates change as the level of requirements for the robot are increased. The rows represent each of the components shown in the figure 5.7. The first column contains the estimate for the original robot. In the second column, estimates are given for a similar robot with a more complex, three finger, manipulator. The user interface, task oriented, and control system modules are increase additional functionality is required. The specifications in the third column are for the same physical mechanism as in the second column but additional functionality is required from the system. There are two additional increases, a large (500 line) constant increase in the size of the control system and a relative small (120 line) additional task oriented routine for each additional vocabulary (primitive function) that is added.

Table 5.2 shows a similar analysis for an unconstrained (six degree of freedom) robot designed for commercial applications. The first column shows estimates for a robot that does simple assembly tasks and whose only sensing mechanism is a force sensor on the manipulator. There is a large component for task oriented routines in the commercial case. The system model is large even though advanced control techniques are not used. This is because the arm is unconstrained (ie, operates with six degrees freedom). Also, a component for diagnostics is introduced in the commercial robot that may not be necessary in a research environment. There is still no error handling term because, unlike other types of systems, a robot arm can be reset when an error is encountered. In the second column, a vision system has been added for feedback purposes only. A sensor processing component has to be added at this point due to the much larger flow of data into the system. There is also an increase in the device driver component due to the introduction of a new device. The user interface and task oriented routines are increased to control the new device, and the diagnostic component is increased to monitor the device. In the third column, pattern recognition is added to the requirements for the vision system. Instead of just being used to verify the position of the arm, the vision system also used to recognize shapes. This adds small increases to the user interface and task oriented routines. A large addition occurs in the sensor processing component. This increase contains the pattern recognition algorithms that are used to identify shapes.

The information summarized above was used to determine the rules shown in Tables 5.3 to 5.6. There are a total of 47 rules that are divided into 10 groups. The first, and largest group, contains 13 rules that specify robotics systems in general. Each of the other nine groups corresponds to one of the generic components of a robotic system. Some of the components are simple. The *top level control*, for example, is sized as a constant of approximately one page (50 lines) of code by rule 14. The *task oriented* rules (rules 24 to 32) are the largest of the component groups reflecting the complexity and variability of this aspect of the software. The code size does not depend completely on complexity, however. Extremely complicated mathematics can often be done by small routines whereas a conceptually simple but flexible report writer or user interface may be much larger. These rules will be implemented either explicitly as rules which control the development of a set of frames describing the new program, or implicitly as the logic for *if needed* routines in the frame base.

Figure 5.5
Classification of Real Time Program

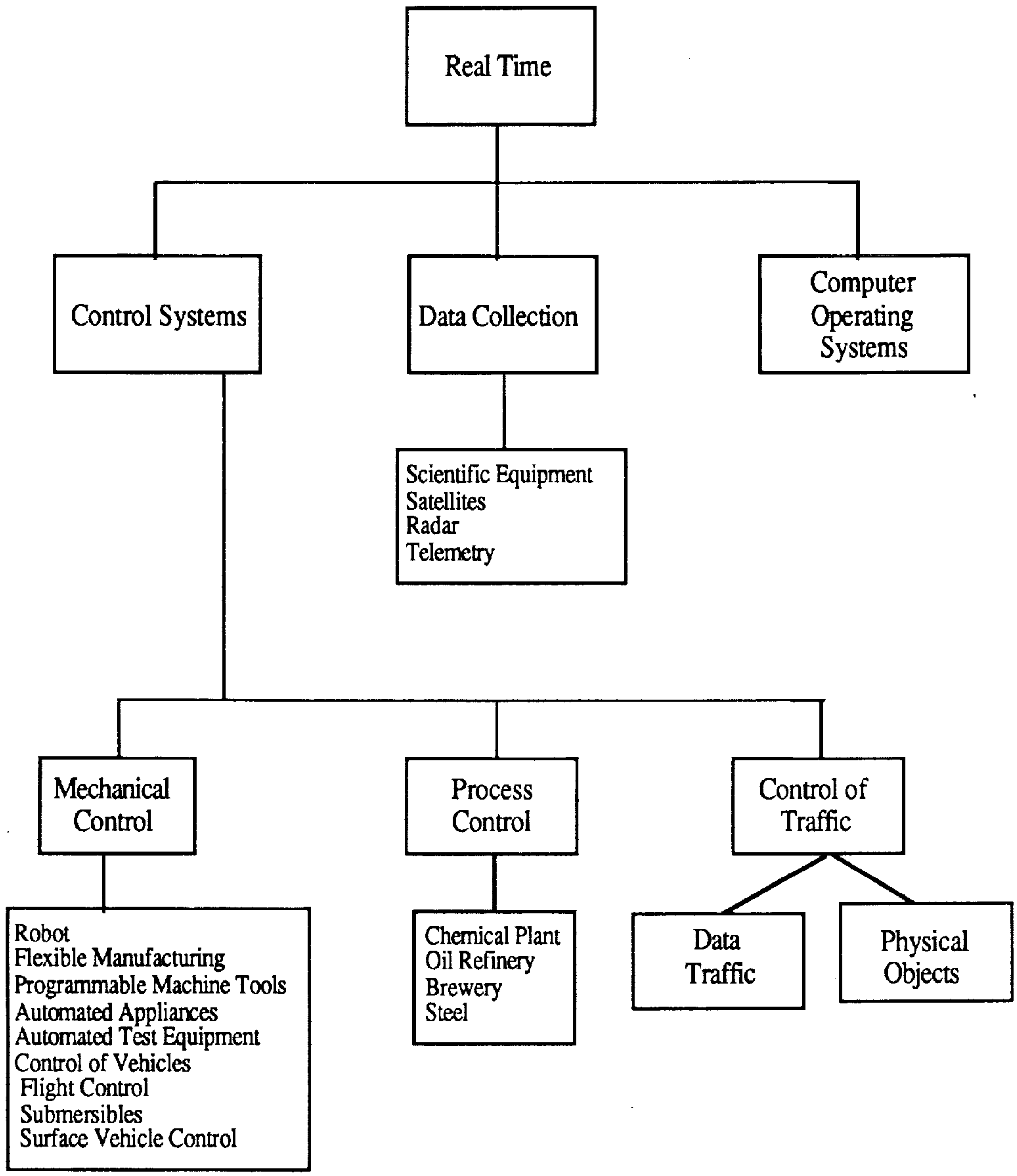


Figure 5.6

Generalized Real Time Program

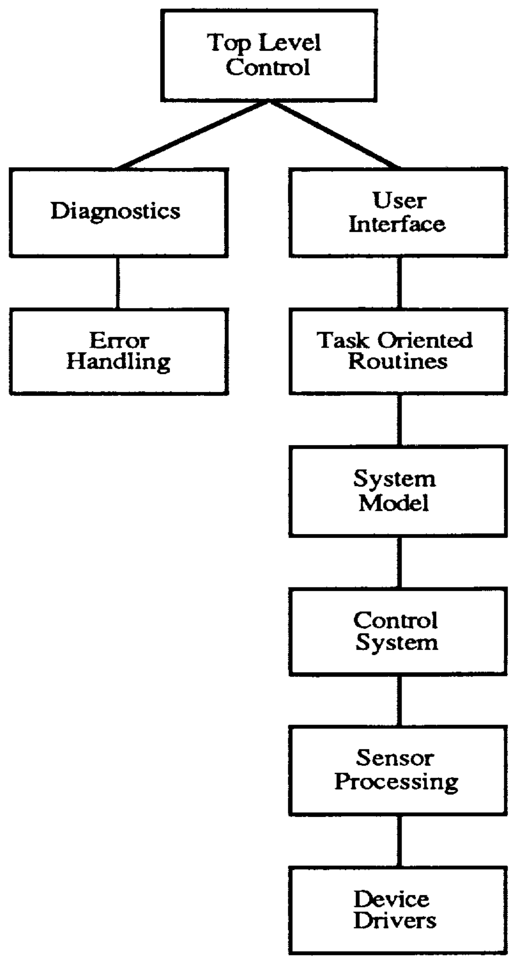


Table 5.1

Analysis of Robotic System Components I

	Constrained One Finger Robot	Three Finger Robot (same functions)	Three Finger Robot (add functions)
Top Level Control	50	same	up
Diagnostics	50	same	same
User Interface	150	Add 150 (coordination)	up
Task Oriented Teach Mode Run Mode Commands	 150 50 120	Add 120 (coordination)	120 for each additional command
System Model*	500	same	same
Control System	50	Add 150 (coordination)	500
Sensor Processing	0	0	0
Device Drivers	250	same	same

* This is for an advanced system. A simple system would be 50 lines.

Table 5.2

Analysis of Robotic System Components II

	6 Joint Simple Assembly (Blind)	Add Vision (for Feedback only)	Add Recognition (Image Understanding)
Top Level Control	50	same	same
Diagnostics	100	200	200
User Interface	150	300	400
Task Oriented	1700	2200	2320
System Model	600	same	same
Control System	100	same	same
Sensor Processing	0	200	3000
Device Drivers	250	450	450

Table 5.3

Sizing Rules for Robotic Systems, I

1. General rules

Rule1:	Most robots are blind
Rule2:	Some robots have VISION
Rule3:	Most robots are PROGRAMMABLE
Rule4:	A CONSTRAINED robot has 3-4 degrees of freedom
Rule5:	An UNCONSTRAINED robot has 5-6 degrees of freedom
Rule6:	A CONSTRAINED robot has a SIMPLE-FUNCTION
Rule7:	An UNCONSTRAINED robot has a COMPLEX FUNCTION
Rule8:	COMMERCIAL robots are CONSTRAINED
Rule9:	RESEARCH robots are either CONSTRAINED or UNCONSTRAINED
Rule10:	Robots with VISION may have PATTERN-RECOGNITION
Rule11:	Robots with PATTERN-RECOGNITION must have VISION
Rule12:	A robot generally has JOINTS and a MANIPULATOR
Rule13:	A robot with JOINTS or a MANIPULATOR has MOTORS

2. Top-level-control rules

Rule14:	TOP-LEVEL-CONTROL is constant at 50 lines (small)

3. Diagnostic rules

Rule15:	A BLIND robot with a SIMPLE-FUNCTION has a 50 line (small) DIAGNOSTIC module
Rule16:	If the robot has a COMPLEX-FUNCTION add 50 lines to the DIAGNOSTIC module
Rule17:	If the robot has VISION, add 100 lines to the DIAGNOSTIC module

Table 5.4

Sizing Rules for Robotic Systems, II

4. Error-handling rules

Rule18:	Robots have a 0 line (INSIGNIFICANT) ERROR-HANDLING component

5. User-interface rules

Rule19:	Most robots have a USER-INTERFACE
Rule20:	Programmable robots must have a USER-INTERFACE
Rule21:	A BLIND robot with a SIMPLE-FUNCTION has a 150 lines USER INTERFACE
Rule22:	If the robot has a COMPLEX-MANIPULATOR, add 150 lines to USER INTERFACE
Rule23:	If the robot has PATTERN-RECOGNITION, add 100 lines to the USER INTERFACE

Table 5.5

Sizing Rules for Robotic Systems, III

6. Task-oriented rules

Rule24:	There are two levels of TASK-ORIENTED-ROUTINES: HIGH-LEVEL-TASKS and LOW-LEVEL-TASKS
Rule25:	PROGRAMMABLE robots have a HIGH-LEVEL-TASK called TEACH-MODE that is 150 line in size
Rule26:	All robots with a HIGH-LEVEL-TASK called RUN-MODE that is 50 lines in size
Rule27:	A BLIND robot with a SIMPLE-FUNCTION has the following low level tasks: INITIALIZE at 50 lines MOVE at 50 lines GRASP at 10 lines RELEASE at 10 lines
Rule28:	For each additional FUNCTION is a LOW-LEVEL-TASK of 120 lines
Rule29:	Add a FUNCTION (coordination) if the robot has a COMPLEX MANIPULATOR
Rule30:	Add 9 FUNCTION if the robot is UNCONSTRAINED
Rule31:	Add 4 FUNCTION if the robot has VISION
Rule32:	Add a FUNCTION if the robot has PATTERN RECOGNITION

7. System model rules

Rule33:	A CONSTRAINED robot has a 50 line SYSTEM-MODEL
Rule34:	A CONSTRAINED robot with ADVANCED-CONTROL has a 500 line SYSTEM-MODEL
Rule35:	If the robot has a COMPLEX-MANIPULATOR add 100 lines to the SYSTEM MODEL

Table 5.6

Sizing Rules for Robotic Systems, IV

8. Control system rules

Rule36:	A CONSTRAINED robot has a 50 line CONTROL-SYSTEM
Rule37:	Add 100 lines if the robot has a COMPLEX-MANIPULATOR
Rule38:	Add 50 lines if the robot is UNCONSTRAINED

9. Sensor processing rules

Rule39:	A BLIND robot has a 0 line SENSOR-PROCESSING component
Rule40:	A robot with VISION has a 200 line SENSOR-PROCESSING component
Rule41:	If the robot has PATTERN-RECOGNITION, add 2800 lines to the SENSOR PROCESSING component

10. Device-driver rules

Rule42:	A robot with MOTORS has a 50 line D-TO-A DEVICE-DRIVER
Rule44:	A robot with JOINTS has a 50 line ENCODER-COUNTER DEVICE-DRIVER
Rule45:	A robot with a FORCE-SENSOR has a 50 line A-TO-D DEVICE-DRIVER
Rule46:	A PROGRAMMABLE robot has a user interface device
Rule47:	A typical user interface device is a TEACH-PENDENT with a 100 line DEVICE DRIVER

2.2.2.2.2 Mr. David Girardot

Two of the major divisions of real time programming are Process Control and Mechanical Control. This subsection summarizes four interviews with Mr. David Girardot, a graduate student of Professor Melichamp at the University of California, Santa Barbara. Mr. Girardot is a graduate student in Process Engineering and has been doing research on the *Multivariable pH Neutralization System* shown in Figure 3.1.

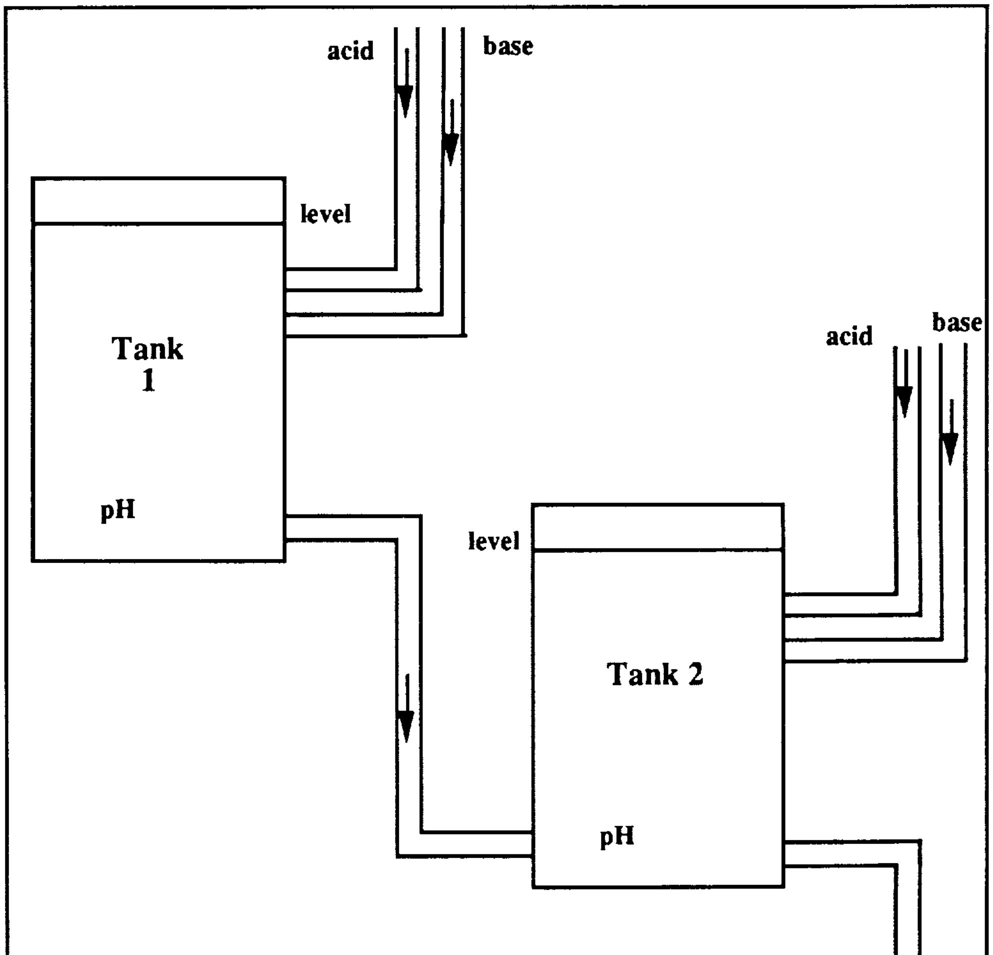


Figure 3.1
Real Time Process Control System

This system was designed as a demonstration unit for the evaluation of advanced process control strategies such as adaptive control and multivariable control. The process is highly nonlinear and contains time-varying gain characteristics. The system contains two coupled tanks with a pH neutralization reaction in each. It has four controlled variables, the pH and liquid level in each tank, and four manipulated variables, the acid and base streams in each tank with strong interactions between the controlled and manipulated variables.

In addition to the interviews, we have obtained the codes for the system. The system is controlled on an IBM PC under a specialized real time operating system. The obtained code includes both programs to run the system, control the apparatus, and record the results and programs to simulate the entire process without the apparatus. It is informative to have both types of codes because we have reason to believe that there is close correspondence between simulation and control and the code used to simulate a process in similar to the code used to control it. This would be useful for the project because Engineering software is, in large part, simulation software and real-time software is, in large part, control software. There is therefore good reason to believe that these two types of knowledge can be integrated in the code sizing knowledge based system in a smooth way rather than as two loosely coupled systems.

Our examination of the process control system confirms that our determination of the components of real time programs:

- Top Level Control
- Diagnostics
- Error Handling
- User Interface
- Task Oriented Routines
- Estimator
- Control System
- Sensor Processing
- Device Drivers,

also applies to process control systems. It should be noted that we now use the term *estimator* rather than *system model* to signify the same component. Both terms are used in real time systems but the first one is more descriptive of the component's function and less likely to be confused with other functions of typical software systems.

There are, however, some differences to be found between process and mechanical control that will result in different relative sizes of components and perhaps some components in Process Control systems that are absent in some mechanical control systems such as robots. In particular, Process Control will have significant Error Handling components to correct errors in addition to Diagnostic components to detect them. This is due to the fact that the time scale for a typical process control application is much longer than that of mechanical control and because it may involve a whole chemical plant or refinery which cannot be shut down and reset because of a fault the way a robot usually can. Process control applications therefore typically have more sophisticated error handling capabilities because there is both the time and motivation for them. Although these are currently the only two real time applications that have been analyzed, similar logic can be applied to other real time applications. In Flight Control, for example, there is a very strong motivation for error correction, but a very short time scale in which to take action. Under these conditions, error handling would be included in the system but would be costly. There would be a number of costs associated with a very safe system including, perhaps, additional hardware for redundancy, and additional software to perform this difficult function in the limited reaction time of the system. Optimization for speed will, in general, increase the code size just as minimizing the code size will sometimes slow the system down.

Another difference between the Process Control and Robotics systems in the user interface. Mechanical Control systems are more often embedded systems than Process Control systems and therefore tend to have less elaborate user interfaces, file handling, and reporting systems. Process Control systems, especially if they are large in scale, often provide the user with a graphic display of the state of the system and record statistics on system performance. A similar system may be present on Flight Control and Data Collection systems. Teachable robotic systems usually have a very simple user interface for guiding the arm through a flexible series motions. Programmable machinery generally is programmed off-line on a general purpose computer and the programs are down loaded into the machinery. In a complex real time system, the diagnostic software component can be large and contain tests and displays of many hardware components. The growth of software size as the system gets large in both size and complexity is an important question of this project. The current hypothesis is a meta rule that relevant software components of a system will grow linearly as the number of types of mechanical devices in a real time system are increased and logarithmically as the number of devices of the same type are increased. The second, more gradual, type of increase in software size might, at first glance, appear not to be present since repeated processes in software can be handled with loops and arrays. However, there is generally a cost to tracking and managing large amounts of data of any kind, and this may, at some point, necessitate a changes in the design of the methods for handling such data.

In general, our experience with the first two samples of real time systems is encouraging and leads us to believe that, if we have a theory of how such systems are structured and a few examples of actual systems, we can create a knowledge base for sizing the codes. We feel, therefore, that it is not necessary to have a large statistical sample of different programs in order to obtain this knowledge. We are, in some sense, using the existing codes to determine parameters and coupling strengths between components of the theory. The components themselves are implementations of stereo-typical concepts that are used over and over again in software development.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Final Report for An Expert System Based Software Sizing Tool, Phase II		5. Report Date 7/90
		6. Performing Organization Code
7. Author(s) David Friedlander		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address		11. Contract or Grant No. NAS7-1033
		13. Type of Report and Period Covered Final Report 7/88 - 7/90
12. Sponsoring Agency Name and Address MCR Technology, Inc. 55 Depot Road Goleta, CA 93117		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The Phase II goal has been achieved and is documented in the final report. The field prototype knowledge based system can predict code sizes based on the functional specifications of the software with an accuracy similar to that of human experts but without their optimistic bias. The Code Sizing Tool accepts specifications from the user and converts them into a set of quantitative generic software components through the user of a rule base. It then sends the generic components to a nonlinear sizing function which predicts code size. The system was calibrated and tested on programs from NASA's COSMIC database and made predictions with a fluctuation factor of 1.77 and a bias factor of 1.06 compared to the actual sizes. The commercial system, planned for Phase III, will allow users to make unbiased size predictions at the proposal stage of software intensive systems projects. Such predictions are important in making budget and schedule estimates.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Code Sizing, CASE, Expert Systems, Artificial Intelligence, Neural Networks	SBIR - Proprietary

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
none	none	340	

NASA FORM 1626 OCT 86