



2

TECHNICAL REPORT BRL-TR-3239

# BRL

REPORT ON USING EXPERT SYSTEMS FOR  
COMPUTER-AIDED FORMULATION AND PROPERTY  
ESTIMATION OF PROPELLANTS. I.  
EXPERT SYSTEM SHELL SELECTION AND  
INITIAL PACKAGE DESIGN

JEFFREY B. MORRIS  
ROBERT A. FIFER

JUNE 1991



APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

U.S. ARMY LABORATORY COMMAND

BALLISTIC RESEARCH LABORATORY  
ABERDEEN PROVING GROUND, MARYLAND

91-03292



## NOTICES

Destroy this report when it is no longer needed. DO NOT return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

UNCLASSIFIED

REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302 and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 1991		3. REPORT TYPE AND DATES COVERED Final Oct 89 - Sep 90	
4. TITLE AND SUBTITLE REPORT ON USING EXPERT SYSTEMS FOR COMPUTER-AIDED FORMULATION AND PROPERTY ESTIMATION OF PROPELLANTS. I. EXPERT SYSTEM SHELL SELECTION AND INITIAL PACKAGE DESIGN				5. FUNDING NUMBERS 1L161102AH43	
6. AUTHOR(S) Jeffrey B. Morris, Robert A. Fifer					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Ballistic Research Laboratory ATTN: SLCBR-DD-T Aberdeen Proving Ground, MD 21005-5066				10. SPONSORING / MONITORING AGENCY REPORT NUMBER BRL-TR-3239	
11. SUPPLEMENTARY NOTES					
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) An expert system-based computer-assisted design (CAD) package for propellant formulation design has been created for use on an Intel 80286/80386-based workstation (PC). This CAD package is intended for use by propellant formulators as a tool to aid in the design of, and properties estimation for, new rocket and gun propellants. A backward-chaining expert system shell (Level 5), dBase III-compatible databases, and a PC version of the NASA-Lewis thermochemical code CET89 are linked together into one package. Properties estimation is accomplished via the use of both the expert system shell and the CET89 code; the thermochemical code options RKT=T (rocket calculation) and UV=T (constant volume) are supported. The initial version of this expert system has been implemented for a propellant formulation consisting of three ingredients (binder, plasticizer, and energetic material). Current research efforts are going into the design of a more comprehensive package for use with up to 21 propellant ingredients.					
14. SUBJECT TERMS Expert Systems, Propellant Formulation, Propellant Properties, Computer Aided Design				15. NUMBER OF PAGES 50	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified		18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified		19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	
				20. LIMITATION OF ABSTRACT UL	

NSN 7540-01-280-5500

UNCLASSIFIED

Standard Form 298 (Rev 2-89)  
Prescribed by ANSI Std Z39-18  
200-102

INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

	<u>PAGE</u>
LIST OF TABLES .....	v
ACKNOWLEDGEMENT .....	vii
I. INTRODUCTION .....	1
II. REQUIREMENTS AND EXPERT SYSTEM SHELL SELECTION .....	2
III. DESCRIPTION OF THE CAD PACKAGE .....	4
A. FORMULATION GENERATION .....	4
B. PROPERTIES ESTIMATION .....	7
C. SEARCHING .....	8
IV. CET89 MODIFICATIONS .....	11
V. EXPERT SYSTEMS .....	13
VI. FUTURE DIRECTIONS AND SUMMARY .....	16
REFERENCES .....	19
APPENDIX A: GLOSSARY .....	21
APPENDIX B: LEVEL 5 PRODUCTION RULE LANGUAGE (PRL) .....	25
APPENDIX C: SAMPLE KNOWLEDGE BASE .....	37
APPENDIX D: THERMOCHEMICAL DEFINITIONS AND CONSIDERATIONS .....	45
DISTRIBUTION LIST .....	49



<b>Accession For</b>	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

INTENTIONALLY LEFT BLANK

## LIST OF TABLES

<u>TABLE</u>		<u>PAGE</u>
1	Comparison of the Four Expert System Shells Evaluated .....	4
2	Fields in the Binder Data Base .....	6
3	Fields in the Propellant Formulation Data Base .....	8
4	Sample Extracted Record from Searching Program .....	10
5	Typical Product Species Exclusion List .....	12
6	Species Included in the Reduced H, C, N, O Thermo Library .....	12
7	Examples of Data Base Fields Used in Rules .....	14

INTENTIONALLY LEFT BLANK



## ACKNOWLEDGEMENT

This work is supported by the Strategic Defense Initiative Organization through the Office of Innovative Science and Technology, Leonard H. Caveny, Deputy Director. Thanks go to Tony Kotlar for doing a critical reading of this report.

INTENTIONALLY LEFT BLANK

## I. INTRODUCTION

We have been tasked to develop a propellant formulation expert system/CAD package. The goal of this project is the development of a "smart" PC-based tool to aid in the design of propellant formulations. This tool should be able to aid the propellant formulation scientist by flagging propellant ingredient incompatibilities and estimating properties of the propellant formulations. The approach taken for this task is to build the CAD package around an expert system shell which incorporates a user modified and maintained knowledge base of "rules." The CAD package needs to link together the expert system, external programs, and data bases into one cohesive system. A version of the CAD package has been developed which supports the use of a binder, a plasticizer, and an energetic material in either a gun or rocket propellant formulation. The output of this CAD package is a data base containing the properties of proposed propellant formulations. In the current implementation, these are either formulations suggested by the user, or automatically generated by the package from ingredients in the ingredient data bases (i.e., binders, plasticizers, and energetic materials). This project represents a major advance of an earlier attempt to develop a propellant formulation design package.<sup>1</sup>

Over the last few years, personal computers (PCs) have continued to develop into increasingly powerful workstations capable of handling tasks which could formerly be run only on mini/mainframe computers. One important area that has moved into the PC domain is that of expert systems and expert system shells. An expert system is an Artificial Intelligence (AI) computer software package which simulates the knowledge and reasoning power of a human expert. The computer queries the user for the necessary information and then based on this information, the expert system suggests an answer to the problem. Expert systems consist of two parts, the "knowledge base" and the "inference engine." The knowledge base and inference engine correspond, respectively, to the knowledge and reasoning power of the human expert. One type of commercially available AI software is that of the expert system "shell." An expert system shell is a tool which helps in the development of expert systems. One of the distinct features of expert system shells is that the knowledge base can be separated from the inference engine. These shells usually fall into one of two classifications depending on how the knowledge base is developed: example-based or rule-based. With example-based expert system shells, a human expert (who acts as the "knowledge engineer," or AI expert) enters examples which the shell uses to develop rules. Rule-based shells require the knowledge engineer to supply rules rather than examples. While the rule-based expert system shells are generally more difficult to learn than example based-shells (since the knowledge engineer must first develop the rules), they offer more flexibility and freedom. There are several reasons for using an expert system shell program rather than other AI software for this project. The use of the expert system shell eliminates the need to be familiar with AI computer languages (such as LISP or PROLOG) by both the expert system developer (primary knowledge engineer) and the person using the system. In general, the shells available today allow individuals with little or no AI background to function as knowledge engineers (for rule modification) once the framework of the expert system has been developed by the primary knowledge engineer. Also, many (non-shell) expert systems developed using an AI computer language require an AI-specific computer system. The

use of PC-based expert system shells results in a product which is more easily modified and has greater portability than other AI - or programming - options currently on the market.

## II. REQUIREMENTS AND EXPERT SYSTEM SHELL SELECTION

The primary goal of this project is to produce a package which is highly portable through the use of PC-based hardware. The proliferation of PCs also makes such a product more accessible than a piece of software which requires a mini or mainframe computer. The initial CAD package was developed and run on a Zenith ZWX-248 AT-compatible PC (Intel 80286-based) with 640k memory + 2560k extended memory and an Intel 80287 math coprocessor, or on a Gateway 2000 system (Intel 80386-based) with 640k memory + 3072k extended memory and an Intel 80387 math coprocessor. The PCs used to develop this package represent typical configurations of machines commonly available in government and industrial offices and laboratories today.

The software used for this project consists of an expert system shell, propellant ingredient and formulation data bases created using dBase III Plus,<sup>2</sup> a PC version (adapted in-house) of the NASA-Lewis thermochemical code CET89,<sup>3</sup> and additional computer programs written in Turbo Pascal and Microsoft Fortran. The use of Dbase III Plus is based upon its wide-spread use and support in the PC software industry.

The expert system shell selection process is based on a hands-on evaluation of four expert-system shells chosen by screening our previous expert system shell evaluation<sup>4</sup> and more recent literature sources.<sup>5-8</sup> The four shells chosen for evaluation are: Exsys Professional,<sup>9</sup> Guru Tutor,<sup>10</sup> Level 5,<sup>11</sup> and Personal Consultant Plus (PC Plus).<sup>12</sup> (Guru Tutor is an almost fully-functional demo version of the full expert system shell Guru.<sup>10</sup>) Exsys and Level 5 were among the five shells previously evaluated;<sup>4</sup> PC Plus and Guru Tutor had not been previously evaluated by our group at BRL. It should be noted that all of the expert system shells under consideration are predominantly backward-chaining rule-based shells. (Some of the shells are more adaptable to forward chaining than others.) A backward-chaining shell will work backward from the desired goal, evaluating rules as necessary until a logical assessment of that goal (usually "true" or "false") can be made; a forward-chaining shell will work forward from the initial data, drawing conclusions until a solution is reached. Level 5 was judged to be the most suitable shell for use in a propellant formulation expert system based on our previous work,<sup>4</sup> and also in this study. The primary requirements for the expert system shell are as follows: 1) the expert system shell must support data base access (preferably dBase) from within the shell; 2) it must be possible to run external programs from within the shell; 3) a means of linking (or "chaining") different expert system modules together must be available; and 4) the shell must not be unduly complex, since user support of the finished product will ultimately be required. (It is recognized that not many people are computer experts.) A secondary (but not absolute) requirement is that the shell should allow an ASCII source file to be used for input -- this allows the expert system to be understood and modified in a much more straight forward manner (i.e., via a word processor or text editor) than if a special rule editor is required.

The details of the expert system shell evaluation process will now be summarized. First of all, only the expert system shells Exsys Professional, Guru Tutor, and Level 5 were evaluated on a head-to-head basis, because of difficulties in procuring a copy of PC Plus in a reasonable amount of time. When a copy of PC Plus arrived, it was compared to Level 5 only. The comparison of the first three expert system shells involved extraction of information from simulated propellant ingredient data bases. The comparison between Level 5 and PC Plus primarily involved going through the PC Plus manuals and doing a hands-on examination of the way rules were written and data bases handled using PC Plus. A summary of key points will be discussed below.

Chaining. All four products support backward chaining. Exsys Professional supports forward and two other special chaining modes. Guru Tutor supports various chaining options which are set up using environmental variables in the rule set. Level 5 supports backward chaining only, but has parameters available to control how the rules are fired. PC Plus supports forward chaining to some degree. Guru Tutor has the most flexibility in its use of logical chaining.

Rule editing. Exsys Professional and Guru Tutor support the use of a rule editor or an ASCII file for creating rules. Level 5 supports the use of an ASCII file only and PC Plus supports the use of a rule editor only. Of the three shells which support the use of an ASCII file to edit rules, the least complex format is that supported by Level 5.

External Program Execution. We were unable to ascertain if Guru Tutor supports external program execution. The other three shells are known to support external program execution. Apparently (from its manuals) it is not difficult to destroy the computer's hard disk directory structure if PC Plus is used incorrectly in the execution of an external program.

Linking of Expert Systems. Linking (or "chaining") of different expert systems is not fully supported in Exsys Professional or Guru Tutor. PC Plus supports the linking of different expert systems in a hierarchical format similar to a DOS directory structure. Level 5 fully supports the linking together of different expert system modules.

Data Base Interface. All of the expert system shells evaluated have some sort of data base interface. Guru Tutor supports its own internal data base only while the other three shells contain some sort of dBase support. PC Plus supports dBase II, dBase III, and dBase III Plus by invoking the data base as an external program through special data base commands. Exsys Professional supports dBase III and dBase III Plus only, while Level 5 also supports dBase II formats. Neither Exsys Professional or Level 5 need dBase present to work with a data base. Exsys Professional, Level 5 and PC Plus all have certain limitations or problems working with data bases. The version of Exsys Plus we evaluated would crash if we tried to read in more than a few records, while Level 5 has a data integrity problem with very small data bases and it is very difficult to get PC Plus to cycle through and read or modify each record in a data base. Level 5 seems to present the fewest problems with data base access, and the problem encountered with Level 5 data base access can be avoided by working with a larger data base.

**Complexity.** Guru Tutor was the most complex of the four expert system shells evaluated. As a consequence, it would be expected that this product would offer the highest degree of versatility to a knowledgeable user and the highest level of frustration to the casual user or novice. In addition, its lack of dBase support eliminated it from consideration for those who must use dBase. Exsys Professional and PC Plus come in about equal on the complexity scale, while Level 5 seems to be the least complex of the four shells tested.

The results of present expert system shell evaluation are shown in Table 1. The selection of Level 5 as the most suitable expert system shell for this project is based upon the above considerations of the features currently available in these shells. Pluses for Level 5 include its relative ease of use, its dBase support, and its ability to link together different expert system modules. The Level 5 Production Rule Language and its knowledge base structure are discussed in detail in Appendices B and C of this report.

**Table 1.** Comparison of the Four Expert System Shells Evaluated.

Shell	Exsys Prof.	Guru	Level 5	PC Plus
Version	1.0	-- <sup>a</sup>	1.0	4.0
Price <sup>b</sup>	\$795	\$5,525	\$685	\$1,800
Backward Chaining	Yes	Yes	Yes	Yes
ASCII Rule Editor	Yes	Yes	Yes	No
External Program Execution	Yes	Unknown	Yes	Yes
Linking of Expert Systems	No	No	Yes	Yes
dBase Interface	Yes	No	Yes	Yes
Complexity <sup>c</sup>	3	4	1	2

a Due to the high cost of Guru, Guru Tutor (v1.1) was evaluated in its place.

b Prices reflect Government discounts of 15% for Guru and 39% for PC Plus.

c Subjective ranking: least complex = 1, most complex = 4.

### III. DESCRIPTION OF THE CAD PACKAGE

Figure 1 shows a flow chart of the CAD package. The CAD package consists of two distinct parts: formulation generation and properties estimation. For the current version, we decided to use formulations consisting of three ingredient types: binder, plasticizer, and energetic material. A data base has been set up for each type of propellant ingredient ("binder," "plast," and "energ," respectively). As an example, the structure of the binder data base is shown in Table 2; the plasticizer and energetic material data bases are similar. Ingredients are added or deleted from each ingredient data base by the use of dBase III Plus; in this way, each data base acts as a master list (which may be customized to a given propellant development program) for ingredients of that type.

#### A. FORMULATION GENERATION

Two modes of formulation generation are supported: Automatic and Manual. Automatic formulation generation is used when the user desires the CAD package to generate multiple formulations based on combinations of ingredients selected within each



data base. This mode of operation is useful when the user does not have a specific formulation in mind, but is interested in exploring a number of possibilities. The user selects the appropriate ingredients from each data base through the use of a menu. The propellant ingredients are combined according to two parameters contained in a configuration file: energetic material weight fraction, and binder-to-plasticizer ratio. Each of these parameters are assigned a numeric value corresponding to high, medium, or low (e.g., the values for the energetic material weight fraction may correspond to 80, 70, and 60 weight percent in the

**Table 2. Fields in the Binder Data Base.**

Field Name	Use
NAME	Ingredient Name
ABBREVIATN	Ingredient Abbreviation
MOL FORM	Molecular Formula
UNIT MW	Unit Molecular Weight
HT OF FORM	Heat of Formation (kJ/Unit-mole)
POLY MW	Polymer Molecular Weight
CHEM TYPE	Chemical Type
POLY TYPE	Polymer Type
ENERGETIC	Is Binder Energetic? (T/F)
PHASE 298K	Ingredient Phase at Ambient Conditions
COST	Ingredient Cost (\$/lb.)
DENSITY	Ingredient Density
DEG CRYST	Degree of Crystallinity
IMP SENS	Impact Sensitivity
FRICT SENS	Friction Sensitivity
ELECT SENS	Electrical Sensitivity
BURN RATE	Burning Rate
HT OF EXPL	Heat of Explosion (kJ/kg)
MELT PT	Melting Point
DECOMP T	Decomposition Point
SOFT PT	Softening Point
GLASS TRAN	Glass Transition Temperature
OXYGEN BAL	Oxygen Balance
PROCESSING	Processing Characteristics

formulation). For each set of three propellant ingredients, nine formulations are generated using the parameter levels specified in the configuration file. Each combination of ingredients (along with weight fractions and other data from the ingredient data bases) is written as a record in the propellant formulation data base. Potentially several thousand formulations may be produced using this method of formulation generation. The formulation data base is acted on by expert systems (Level 5 rule knowledge-bases) designed to screen for removal unwanted formulations and modify ingredient weight fractions. Formulation screening is done to remove formulations which contain ingredient combinations known to be incompatible, unstable, or unsafe. Weight fraction modification is done to adjust the ingredient weight fractions (which are set automatically by the CAD package) in cases where it may be advantageous to do so (e.g., in a case where the binder density is low, it may be desirable to increase the weight fraction of the energetic material). The means by which the expert system modifies the formulation data base will be discussed later on in this report.

Manual formulation generation is designed for when the user has one specific formulation (including weight fractions) in mind. The user selects the appropriate ingredient from each data base through the use of a menu. Once the ingredients are selected, the user is prompted to enter the weight fraction for each ingredient. Since the user has a definite idea of what the formulation should look like, no ingredient screening or weight fraction modification is done in this mode of formulation generation. Unlike automatic formulation generation, which produces at least nine formulations, manual



formulation generation produces a single formulation which is written into the formulation data base.

## B. PROPERTIES ESTIMATION

Once the formulations have been generated, the CAD package proceeds to properties estimation. Properties estimation consists of running thermochemical calculations, followed by expert system estimation of propellant properties. Table 3 shows the format of the propellant formulation data base (including the properties currently supported by this package). By running the thermochemical code prior to invoking the expert system, calculation results from CET89 can be used by expert system rules to estimate other properties.

The thermochemical code selected is the NASA-Lewis CET89 code.<sup>3</sup> The Fortran code was adapted in-house for PC use. This PC version of CET89 was further modified (for size and speed, and the inclusion of an impetus calculation) for use within this CAD package. One of the advantages in using CET89 is that a single thermochemical code can be used which can both run rocket calculations (RKT=T option) and approximate gun conditions (UV=T option, constant volume). The CAD package takes care of processing CET89 input and output files: CET89 input files are automatically written by reading the required information from the ingredient data bases, the thermochemical code is run, the CET89 output files are automatically read for each of the formulations in the formulation data base, and selected data from the thermochemical code output are written into the formulation data base. The propellant combustion data currently imported from the CET89 output files into the formulation data base include: flame temperature (T), enthalpy (H), internal energy (U), free energy (G), entropy (S), product gas average molecular weight (M), product gas heat capacity ( $c_p$ ),  $\gamma_s$ , sonic velocity, specific impulse or impetus (depending on the type of calculation performed) and mole fractions (up to 15 species mole fractions are imported, including those for carbon monoxide, carbon dioxide, water, molecular hydrogen, and molecular nitrogen). The thermochemical quantities mentioned above are discussed in greater detail in Appendix D of this report. Once the thermo code has been run for each formulation, the CAD package proceeds to the properties estimation expert system.

At this point, the properties estimation expert system is run to evaluate propellant properties not calculated with CET89. Once the formulations have been run through the properties estimation expert system, the formulation data base is ready to be searched. Each property to be evaluated will require at least one rule in the expert system; in a final "trained" expert system, there may be dozens of rules for each property to take into account different types of possible ingredient interactions. Since rules for specific propellant properties will vary from propellant to propellant, the appropriate rules will need to be supplied by the user of the CAD package for each type of propellant being worked with, that is, the knowledge-base will in general be different for each propellant development program.

**Table 3.** Fields in the Propellant Formulation Data Base.

Field Name	Description
ENERG_NAME	Energetic Material Name
BIND_NAME	Binder Name
PLAST_NAME	Plasticizer Name
ADDIT_NAME	Additive Name -- Not Currently Supported
ENERG_ABBR	Energetic Material Abbreviation
BIND_ABBR	Binder Abbreviation
PLAST_ABBR	Plasticizer Abbreviation
ADDIT_ABBR	Additive Abbreviation -- Not Currently Supported
ENERG_FRAC	Energetic Material Weight Fraction
BIND_FRAC	Binder Weight Fraction
PLAST_FRAC	Plasticizer Weight Fraction
ADDIT_FRAC	Additive Weight Fraction
PHASE_298K	Estimated Phase
COMP_COST	Estimated Ingredient Cost
DENSITY	Estimated Density
BURN_RATE	Estimated Burning Rate
HT_OF_EXPL	Estimated Heat of Explosion
MELT_PT	Estimated Melting Point
DECOMP_T	Estimated Decomposition Temperature
GLASS_TRAN	Estimated Glass Transition Temperature
FRICT_SENS	Estimated Friction Sensitivity
ELECT_SENS	Estimated Electrical Sensitivity
IMP_SENS	Estimated Impact Sensitivity
OXYGEN_BAL	Estimated Oxygen Balance
VAP_PRESS	Estimated Vapor Pressure
CALC_TYPE	CET89 Calculation Type: Rocket or Constant V
FLAME_TEMP	Calculated Flame Temperature
H	Calculated Enthalpy
U	Calculated Internal Energy
G	Calculated Free Energy
S	Calculated Entropy
PRODUCT_MW	Calculated Product Gas Average Molecular Weight
CP	Calculated Product Gas Heat Capacity
GAMMA	Calculated Product Gas Gamma
SONIC_VEL	Calculated Product Gas Sonic Velocity
IMPETUS	Calculated Impetus if Constant V Calculation
SP_IMPULSE	Calculated Specific Impulse if Rocket Calculation

### C. SEARCHING

When the properties have been estimated for all of the formulations in the formulation data base, the data base records can be searched according to fields which contain the values for the propellant properties; this feature is required only for automatic formulation generation, where many formulations have been generated. Currently supported searching fields are: component cost, density, burn rate, heat of explosion, melting point, decomposition temperature, glass transition temperature, friction sensitivity, electrical sensitivity, impact sensitivity, oxygen balance, vapor pressure, flame temperature, H, U, G, S, product molecular weight,  $c_p$ ,  $\gamma_s$ , sonic velocity, impetus (for CET89 constant volume calculations), specific impulse (for CET89 rocket calculations), and mole fractions. Searches can be done in either ascending or descending order for

**Table 3.** Fields in Propellant Formulation Data Base. Continued.

Field Name	Description
X_CO	CO Mole Fraction
X_CO2	CO <sub>2</sub> Mole Fraction
X_H2	H <sub>2</sub> Mole Fraction
X_H2O	H <sub>2</sub> O Mole Fraction
X_N2	N <sub>2</sub> Mole Fraction
SPECIES_1	Additional Species 1 Name
X_1	Additional Species 1 Mole Fraction
SPECIES_2	.
X_2	.
SPECIES_3	.
X_3	.
SPECIES_4	.
X_4	.
SPECIES_5	.
X_5	.
SPECIES_6	.
X_6	.
SPECIES_7	.
X_7	.
SPECIES_8	.
X_8	.
SPECIES_9	.
X_9	.
SPECIES_10	Additional Species 10 Name
X_10	Additional Species 10 Mole Fraction
MECH_PROP1	Future Use: Mechanical Property
MECH_PROP2	Future Use: Mechanical Property
MECH_PROP3	Future Use: Mechanical Property
PROCESSING	Processing Characteristics (User-Defined)
PART SIZE	Particle Size: Small, Medium, or Large
FUTURE_EX2	Future Use: General Property
FUTURE_EX3	Future Use: General Property
ENERG CONC	Energetic Material Concentration: High, Medium, or Low
B TO P	Binder-to-Plasticizer Ratio: High, Medium, or Low
ADD CONC	Additive Concentration -- Not Currently Supported
KEEP	Used Internally to CAD Package
FLAGS	Used Internally to CAD Package

a given field (future versions of this CAD package will allow for multiple field searches with weighting factors assigned to the individual fields). The searching program will extract the twenty records with the highest (descending order) or lowest (ascending order) values of the selected field; for example, the twenty formulations with the lowest cost or with the highest impetus or specific impulse may be extracted. The searching program also supports two other modes of operation where all of the records in the data base may be viewed in a sequential manner, or an individual record may be viewed (indexed by its record number in the data base). A sample extracted record is shown in Table 4.

**Table 4.** Sample extracted record from searching program.

```

Record Number:      1          CET89 Calculation type: Rocket
*****
Type      Ingredient Name      Abbreviation      Wt. Fraction
-----
Energ. Mat.  RESEARCH DEPARTMENT EXPLOSIVE  RDX              0.600000
Binder      NITROCELLULOSE, 12.6% N      NC               0.133333
Plasticizer  1,2,4-BUTANETRIOL TRINITRATE  BTTN            0.266667
*****
Phase      Cost($/lb)  Densitya  Burn Rate  HEX(kJ/kg)  Oxygen Bal.(%)b
-----
S          38.00      1.705      1.273      5740.210      -1.000
*****
          Temperatures in K          Sensitivities
-----
          M.P.  Decomp. T  Glass. Tr.  Friction  Electrical  Impact
-----
          246.150  246.150      158.523      -1.000      -1.000      1.600
*****
Vap. Press  Energetic Material
(torr)      Particle Size      Processingc
-----
          -1.000      SMALL      ?????????????????????
*****
Flame T(K)  H(cal/g)  U(cal/g)  G(cal/g)  S(cal/g-K)  Sonic Vel(m/s)
-----
          1798.71  -828.07  -969.800  -5377.21      2.5291      855.5
*****
          Gas MW      Cp      gamma      Specific Impulse
-----
          25.220      0.4161      1.2341      278.100
*****
          X CO      X CO2      X H2      X H2O      X N2      Species 1      X 1
-----
          0.241700  0.148300  0.104800  0.243800  0.261400  H              0.000114
*****
          Species 2      X 2      Species 3      X 3      Species 4      X 4
-----
          OH              0.000018 NO              0.000001 HCN              0.000000
*****
          Species 5      X 5      Species 6      X 6      Species 7      X 7
-----
          HCO RAD      0.000000 O2              0.000000 O              0.000000
*****
          Species 8      X 8      Species 9      X 9      Species 10      X 10
-----
          HNCO              0.000000 NH2              0.000000              0.000000

```

Hit <Space Bar> for next record or <Esc> for new sort

- a. The units specifications on certain fields are left up to the user via expert system rules.
- b. Numerical properties with values of -1.000 have not been evaluated.
- c. Un-evaluated character fields are indicated with question marks.

#### IV. CET89 MODIFICATIONS

It is useful to document the modifications made to the CET89 thermochemical code for this project: a) adapting the full source code to PC, b) reducing the size of the PC source code, c) reducing the PC RAM (Random Access Memory) requirement of the internal data arrays of the executable (compiled and linked) program, and d) writing external programs to reduce the execution time of a CET89 run. The first step involved adapting the code to run on a PC - this primarily involved changing format statements in the code. The result of this adaptation is a fully functioning PC version of the code.

The second set of modifications were done to reduce the size of the code. The full CET89 code (compiled and linked) requires approximately 1/2 Mbyte of PC RAM. After loading the operating system (MS-DOS), this leaves very little RAM left to run other programs (typically, CET89 would be called as a DOS application from within another program). The size of the source code (and also, the executable program size) can be reduced by eliminating those modules not required for the CAD package. Modules for calculations such as detonation or transport properties were removed; these modules can be supported if required in future versions of the CAD package. In addition, the modules which read the thermo and transport ASCII libraries were removed and written into a separate program in order to reduce the run time when multiple runs of CET89 are required. The CET89 treatment of libraries is discussed below.

About 1/2 of the RAM occupied by CET89 consists of arrays. The default array dimensions are 600 for the maximum number of species (MAXNS) and 400 for the maximum number of condensed species (MAXNC) which CET89 can handle. Since this many species will greatly exceed the expected number of important product species from a CET89 run, these values were reduced to MAXNS = 167 and MAXNC = 121. (The choices of MAXNS = 167 and MAXNC = 121 allow the Fortran source code to be searched, and these values modified, in an easier manner than if "rounder" numbers were chosen for these dimensions.) By reducing the size of the data arrays and removing some of the Fortran source code (described above), the RAM requirement of the CET89 executable program is cut in half.

The other modifications allow CET89 to execute faster, especially in the case of multiple runs. Pascal programs were written which screen the ingredient data base files in order to determine which atomic species were used in the propellant formulation. Once the atomic species are determined, the size of the thermo library can be reduced to only include products containing those elements. A product species exclusion list is also used to further reduce the size of the thermo library - for instance, benzene would not be expected as a combustion product. Table 5 shows a typical exclusion list. By extracting the atomic constituents from a propellant formulation and using an exclusion list of 122 species, the thermo library for a H, C, N, O propellant is reduced from about 1,100 species to 21. These 21 species are shown in Table 6. In the original PC modification of CET89, the thermo and transport ASCII libraries are read for each run of the code, written into binary files, and these binary files are then used. This file conversion was removed from CET89 and written as a separate Fortran program. Running this conversion program lets

**Table 5. Typical Product Species Exclusion List.**

C	CH	CH <sub>2</sub>	CH <sub>3</sub>
CN	NCN RAD	CNC RAD	CCO RAD
CNN RAD	C <sub>2</sub> H RAD	C <sub>2</sub> H <sub>3</sub> RAD	CH <sub>3</sub> CO RAD
CH <sub>2</sub> CHO RAD	C <sub>3</sub>	C <sub>3</sub> H <sub>3</sub> RAD	C <sub>3</sub> H <sub>5</sub> RAD
C <sub>4</sub>	C <sub>5</sub>	CH <sub>3</sub> C(CH <sub>3</sub> ) <sub>2</sub> CH <sub>3</sub>	HNO
HNO <sub>2</sub>	HNO <sub>3</sub>	HO <sub>2</sub>	H <sub>2</sub> N <sub>2</sub>
H <sub>2</sub> O <sub>2</sub>	N	NCO	NH
NH <sub>2</sub> OH	NO <sub>2</sub>	NO <sub>3</sub>	N <sub>2</sub> H <sub>2</sub>
N <sub>2</sub> O <sub>3</sub>	NH <sub>2</sub> OH	NH <sub>2</sub> NO <sub>2</sub>	N <sub>2</sub> O <sub>4</sub>
N <sub>2</sub> H <sub>4</sub>	N <sub>2</sub> O <sub>5</sub>	O <sub>3</sub>	N <sub>2</sub> O
N <sub>3</sub>	FORMALDEHYDE	FORMIC ACID	HYDROXYMETHYLENE
METHYLOXIDE	METHANOL	ACETYLENE	KETENE
METHYL CYANIDE	ETHYLENE	ACETALDEHYDE	ACETIC ACID
(FORMIC ACID) <sub>2</sub>	ETHYL RAD	ETHYL OXIDE RAD	ETHANE
AZOMETHANE	DIMETHYL ETHER	ETHANOL	CNC RAD
CYANOGEN	CYCLOPROPENE	PROPYNE	ALLENE
CYCLOPROPANE	PROPYLENE	PROPYLENE OXIDE	I-PROPYL RAD
N-PROPYL RAD	PROPANE	1-PROPANOL	CARBON SUBOXIDE
BUTADIYNE	CYCLOBUTADIENE	BUTAN-1EN-3YN	1,3-BUTADIENE
2-BUTYNE	2-BUTENE TRANS	2-BUTENE CIS	ISOBUTENE
1-BUTENE	(ACETIC ACID) <sub>2</sub>	T-BUTYL RAD	S-BUTYL RAD
N-BUTYL RAD	N-BUTANE	ISOBUTANE	CARBON SUBNITRID
CYCLOPENTADIENE	CYCLOPENTANE	1-PENTENE	T-PENTYL RAD
N-PENTYL RAD	PENTANE	ISOPENTANE	HEXATRIYNE
PHENYL RAD	PHENOXY RAD	BENZENE	PHENOL
CYCLOHEXENE	N-HEXYL RAD	TOLUENE	CRESOL
1-HEPTENE	N-HEPTYL RAD	N-HEPTANE	1-OCTENE
N-OCTYL RAD	OCTANE	ISO-OCTANE	N-NONYL RAD
NAPHTHLENE	AZULENE	N-DECYL RAD	O-BIPHENYL RAD
BIPHENYL	JET-A(G)	BENZENE(L)	TOLUENE(L)
OCTANE(L)	JET-A(L)		

**Table 6. Species Included in the Reduced H, C, N, O Thermo Library.**

CH <sub>4</sub>	CO	CO <sub>2</sub>	C <sub>2</sub>
H	HCN	HCO RAD	HNCO
H <sub>2</sub>	H <sub>2</sub> O	NH <sub>2</sub>	NH <sub>3</sub>
NO	N <sub>2</sub>	N <sub>3</sub> H	O
OH	O <sub>2</sub>	C(GR)	H <sub>2</sub> O(S)
H <sub>2</sub> O(L)			

the CET89 program use the binary thermo library; the binary library can be read in much faster than the ASCII version. Reducing the number of product species which need to be considered and doing the library ASCII-to-binary conversion results in a speed increase of greater than a factor of 10 for a CET89 run. For evaluation of many formulations (i.e., in the case of automatic formulation generation), the savings in execution time can be significant since the library conversion needs to be done only once for a given set of propellant ingredients.

## V. EXPERT SYSTEMS

Three expert system knowledge bases are implemented in the current version of the CAD package. These expert systems are responsible for deleting undesirable formulations and modifying weight fractions of propellant ingredients in the automatic formulation generation mode, and for predicting propellant formulation properties. The expert systems invoke user-supplied rules to perform a variety of tasks to the formulation data base. Taking advantage of user-supplied rules rather than programmed computer code results in the development of a much more versatile and flexible package. By making rule modification easily accessible to the user of this CAD package, adaptation to a specific propellant program is allowed in a straight forward manner. The formulator can modify a knowledge-base using a word processor without having to be familiar with a high-level computer language. An example of a typical rule for calculating a propellant formulation property would be:

```
RULE to predict glass transition temperature (I)
IF combine.flame_temp > 0
THEN got glass T
AND num1 := binder.glass_tran
AND num2 := combine.plast_frac/combine.bind_frac
AND num3 := plast.mol_weight
AND combine.glass_tran := num1 - (5000/num3)*num2
ELSE NOT got glass T
```

Rule I calculates the glass transition temperature according to the following formula:

$$T_g = T_{g(\text{binder})} - (C/MW_{\text{plast}})(X_{\text{plast}}/X_{\text{binder}}) \quad (1)$$

where  $T_g$  is the glass transition temperature of the propellant formulation (the desired property to be estimated),  $T_{g(\text{binder})}$  is the known glass transition temperature of the binder,  $MW_{\text{plast}}$  is the plasticizer molecular weight,  $X_{\text{plast}}$  is the plasticizer weight fraction,  $X_{\text{binder}}$  is the binder weight fraction, and  $C$  is a constant. Rule I is intended to be only an example of a rule for properties estimation (no validity for  $T_g$  is assumed), however it demonstrates several features of the Level 5 Production Rule Language (PRL). In addition to demonstrating the usage of several PRL reserved words (in upper case at the start of each line), Rule I also demonstrates data base access and the usage of algebraic expressions in Level 5. Data base access is accomplished by specifying the data base name (without the ".dbf" extension) followed by the data base field name; the two names are separated by a period. Examples of data base access (data base.fieldname) in Rule I are: "combine.flame\_temp", "combine.plast\_frac", "combine.bind\_frac", "combine.glass\_tran", "binder.glass\_tran", and "plast.mol\_weight". The data bases "combine," "binder," and "plast" are, respectively, the formulation, binder, and plasticizer data bases used for Rule I; the fields used for this rule are summarized in Table 7. The use of rules is discussed in Appendix B of this report. The antecedent of Rule I includes the supporting condition "combine.flame\_temp > 0"; this supporting condition will fire the rule in the event that a CET89 flame temperature has been calculated for the current formulation. If the supporting condition is true, then the conclusion "got glass T" is set to

**Table 7. Examples of Database Fields Used in Rules.**

Database	Field	Usage
Combine	Flame_Temp	Propellant Flame Temperature calculated using CET89
Combine	Energ_Abbr	Energetic Material abbreviation
Combine	Energ_Frac	Energetic Material weight fraction
Combine	Bind_Frac	Binder weight fraction
Combine	Plast_Frac	Plasticizer weight fraction
Combine	Glass_Tran	Propellant glass transition temperature
Combine	Burn_Rate	Propellant burning rate
Combine	Energ_Name	Energetic Material name
Combine	Comp_Cost	Propellant ingredient cost
Combine	Calc_Type	CET89 calculation type (Rocket or Constant V)
Bindtemp	Glass_Tran	Binder glass transition temperature
Bindtemp	Burn_Rate	Binder burning rate (set to 0 if not energetic)
Bindtemp	Cost	Binder cost
Plastemp	Mol_Weight	Plasticizer molecular weight
Plastemp	Burn_Rate	Plasticizer burning rate
Plastemp	Cost	Plasticizer cost
Energtemp	Burn_Rate	Energetic Material burning rate
Energtemp	Cost	Energetic Material cost

"True" and the  $T_g$  from Equation 1 is calculated and written to the formulation data base field "combine.glass\_tran" through the use of the four lines ANDed to the THEN statement. If the supporting condition is false, then the ELSE statement will set the conclusion "got glass T" to "False" and no glass transition temperature will be calculated.

Rule II is an example of a rule to calculate a propellant burning rate:

RULE to predict burning rate for RDX-based Rocket propellant (II)

```

IF combine.calc_type = Rocket
AND combine.energ_abbr = RDX
THEN got burning rate
AND num1 := -0.05957
AND num2 := 0.000147
AND num3 := combine.flame_temp
AND combine.burn_rate := num1 + num2*num3
ELSE NOT got burning rate

```

Rule II calculates the propellant burning rate as a function of flame temperature according to the following formula:

$$R = b + mT_f \quad (2)$$

where R is the propellant burning rate (cm/sec),  $T_f$  is the flame temperature (K) in the rocket chamber, and b and m are linear fitting parameters. Rule II applies to a rocket propellant being burned at 70 atmospheres. Burning rate data for RDX-Polyurethane propellants (RDX weight fractions: 0.75, 0.80, 0.85) are contained in Figure 20 of



Reference 13. The corresponding flame temperatures are calculated using CET89. A linear regression of burning rate versus flame temperature yields the parameters b (intercept) and m (slope) used in Equation 2. This trial rule is based on known correlations between burning rate and flame temperature (or heat of explosion). In Rule II, the antecedent requires the energetic material to be RDX and the CET89 calculation to be for rocket propellants in order for the rule to be fired. This rule demonstrates the use of a quantity calculated by CET89 (flame temperature) and stored in the formulation data base ("combine"). The conclusion for Rule II is "got burning rate" and when this conclusion is found to be true, the burning rate is calculated and stored in the formulation data base field "combine.burn\_rate". While Rule II is specific to a RDX rocket propellant with a non-energetic binder at 70 atmospheres, it could be one of many rules used to estimate propellant burning rates in a variety of situations.

While some properties (burning rate, sensitivity, etc.) can only be estimated with our current understanding of ingredient interactions, other properties (cost, density, etc.) can be evaluated more exactly. Rule III is a simple rule to calculate the materials cost for a formulation:

```

RULE to calculate ingredient cost                                     (III)
IF combine.energ_name = combine.energ_name
THEN got ingredient cost
AND num1 := combine.energ_frac*energ.cost
AND num2 := combine.bind_frac*binder.cost
AND num3 := combine.plast_frac*plast.cost
AND combine.comp_cost := num1 + num2 + num3
ELSE NOT got ingredient cost

```

This rule calculates the materials cost of the propellant by summing over the costs of the amounts of the individual ingredients used. The supporting statement ("combine.energ\_name = combine.energ\_name") used is chosen to ensure that this rule will fire every time. While this is a simple property to code into a computer program, the use of an expert system shell makes it easy to change the way this field is calculated (for example, a cost overhead factor could easily be inserted into or removed from this rule).

Rules can be linked into other rules by making the conclusion of one rule part of the antecedent of another rule. Consider Rule IV:

```

RULE to fire Rules I-III to calculate propellant properties        (IV)
IF got glass T
OR NOT got glass T
AND got burning rate
OR NOT got burning rate
AND got ingredient cost
OR NOT got ingredient cost
THEN fired all rules

```

The antecedent of Rule IV uses the conclusion statements of Rules I-III. This rule fires each rule (in order to evaluate "got glass T," "got burning rate," and "got ingredient cost") sequentially until all of the rules have been evaluated and the property data base has been updated. Using rules such as Rule IV provides an easy means for linking rules into - or out of - a knowledge-base.

## VI. FUTURE DIRECTIONS AND SUMMARY

This report has described an expert system-based CAD package for propellant formulation and property estimation which supports three ingredient propellant formulations. The development of this package has been primarily for demonstration purposes: we recognize the limitations of a package which supports only three propellant ingredient types and are currently working on a revised version of this propellant formulation CAD package which will allow the use of more ingredient types than are currently supported. There are several differences between this present version of the propellant formulation package and the improved version currently being worked on.

The most obvious difference to the user of the two packages is with the limitation to the number of ingredients and ingredient types supported in the two versions. In the improved version of this package, five ingredient types will be supported: energetic material, binder, plasticizer, metal fuel, and other additives. The maximum number of ingredient supported will break down as follows: six energetic materials, three binders, three plasticizers, three metals, and six other additives; this gives a total of 21 possible ingredients in the new version of this package. The main reason for supporting six possible energetic materials is the incorporation of identical energetic material ingredients with varied particle sizes (e.g. a formulation using RDX with 5, 20, and 70 micron particle sizes would use up three of the energetic material slots in the new CAD package). Due to the large number of combinations which would need to be generated, the ability to support 21 ingredients in the formulation essentially precludes any sort of automatic formulation generation, so this feature will be removed from the revised version of the package. In addition, only one propellant ingredient data base will be used. The reason for using a single ingredient data base is that Level 5 can only support four data bases in any single knowledge base module; the use of one data base for each ingredient type (energetic material, binder, plasticizer, metal fuel, and other additives) would result in five data bases being required to handle all of the propellant ingredients, plus another data base to handle the formulation (six data bases in all).

The other main difference between the two versions of the CAD package will be the inclusion of a knowledge base for screening propellant ingredient incompatibilities based upon safety in manufacture and environmental hazards in use. The idea behind such a knowledge base is the ability to warn the user of potentially dangerous propellant ingredient combinations. It would then be up to the user to decide whether or not to continue the formulation process using those ingredients.

The current propellant formulation expert system package allows for the easy modification of expert system rules which have the capability to screen for propellant

ingredient conflicts, adjust propellant weight fractions, and predict propellant properties. At this point, the "framework" for the package is largely completed for use with a three component system and the package is ready to be "tailored" to specific propellant development programs. This tailoring will involve the setting up of ingredient data bases and the input of knowledge base rules. By expanding the capabilities of this CAD package to support up to 21 varied propellant ingredients, we expect to be able to handle propellant formulations of HMX/AP/HTPB/Al, CL20-based formulations, Cubanes, or LOVA.

A large effort needs to be spent along the lines of gathering information suitable for incorporation into rules. This process involves the knowledge engineer collaborating with the human experts in order to develop rules, or at least to get useful information in a format which the knowledge engineer can use to write a rule. The co-operation of the human experts in the field of propellant formulation is crucial for the continued success of this project.

INTENTIONALLY LEFT BLANK

## REFERENCES

1. F. J. Shaw and R. A. Fifer, "An Expert System for Computer-Aided Design of Propellant Formulations," Proceedings of 24th JANNAF Combustion Meeting, CPIA Pub. 476, Vol. III, pp. 251-258, October 1987. See also BRL Technical Report BRL-TR-2895, March 1988.
2. dBase III Plus is a product of: Ashton-Tate, 20101 Hamilton Avenue, Torrance, CA 90502-1319.
3. The unmodified Fortran program CET89 is documented in the following references:  
a) R. A. Svehla and B. J. McBride, "Fortran IV Computer Program Calculation of Thermodynamic and Transport Properties of Complex Chemical Systems," NASA Technical Note TN D-7056, January 1973; b) S. Gordon and B. J. McBride, "Computer Program for Calculation of Complex Chemical Equilibrium Compositions, Rocket Performance, Incident and Reflected Shocks, and Chapman-Jouguet Detonations," NASA SP-273, March 1976; c) S. Gordon, B. J. McBride, and F. J. Zeleznik, "Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications Supplement I - Transport Properties," NASA Technical Memorandum 86885, October 1984.
4. F. J. Shaw and R. A. Fifer, "An Evaluation of Five PC-Based Expert System Shells," Ballistic Research Laboratory Technical Report BRL-TR-2908, March 1988.
5. K. Pedersen, "Connecting Expert Systems and Conventional Environments," AI Expert, pp. 26-35, May 1988.
6. T. Bodkin and I. Graham, "Case Studies of Expert System Development using Microcomputer Software Packages," Expert Systems, Vol. 6, pp. 12-15, February 1989.
7. R. G. Vedder, "PC-Based Expert System Shells: Some Desirable and Less Desirable Characteristics," Expert Systems, Vol. 6, pp. 28-42, February 1989.
8. "The 1st Annual AI Product Guide," PC AI, pp. 62-68, July/August 1989.
9. Exsys Professional is a product of: Exsys Inc., P.O. Box 11247, Albuquerque, NM 87192.
10. Guru and Guru Tutor are products of: mdba (Micro Data Base Systems), Inc., P.O. Box 248, Lafayette, IN 47902.
11. Level 5 Expert System Software is a product of: Information Builders, Inc., 1250 Broadway, New York, NY 10001.
12. Personal Consultant Plus is a product of: Texas Instruments Inc., Data Systems Group, M/S 2151, P.O. Box 2909, Austin, TX 78769-2909.

13. N. Kubota, "Survey of Rocket Propellants and Their Combustion Characteristics," in Fundamentals of Solid-Propellant Combustion, K. K. Kuo and M. Summerfield, ed., American Institute of Aeronautics and Astronautics, Inc., New York, pp. 1-52, 1984.

## APPENDIX A: GLOSSARY

INTENTIONALLY LEFT BLANK



## APPENDIX A: GLOSSARY

This glossary contains terms used in the field of expert systems.

**ANTECEDENT:** The supporting condition or conditions of a rule (IF statement).

**CONCLUSION:** The result carried out when the antecedent of a rule is true (THEN statement).

**DOMAIN:** The body of knowledge which encompasses the subject matter of the expert system.

**EXPERT SYSTEM:** An inference engine (*vide infra*) coupled to a knowledge base which contains information on a given domain of knowledge. The expert system can be consulted to access this information.

**HUMAN EXPERT:** The person who has the actual knowledge which makes up the domain of the expert system.

**INFERENCE ENGINE:** The Artificial Intelligence means of reasoning which is coupled to a knowledge base (*vide infra*) to make an expert system. The inference engine corresponds to the reasoning ability of a human.

**KNOWLEDGE BASE:** A collection of rules (*vide infra*) which consist of the knowledge (domain) of a given field. An ideal knowledge base will correspond to the knowledge of one or more human experts.

**KNOWLEDGE ENGINEER:** The person who transforms the knowledge of the human expert into an expert system.

**RULE:** The means by which human knowledge is transformed into a knowledge base by the knowledge engineer. Rules generally consist of an antecedent followed by a conclusion.

INTENTIONALLY LEFT BLANK

## APPENDIX B: LEVEL 5 PRODUCTION RULE LANGUAGE (PRL)

INTENTIONALLY LEFT BLANK

## APPENDIX B: LEVEL 5 PRODUCTION RULE LANGUAGE (PRL)

The Level 5 Production Rule Language (PRL) is used to construct a knowledge base source file. Knowledge is stored in the expert system through the use of rules written by the human expert and/or the knowledge engineer. Typically, the knowledge engineer queries the human expert for the knowledge which is incorporated into a rule; the knowledge engineer would then write a rule using PRL. Rules generally consist of an antecedent followed by a conclusion. The antecedent of a rule consists of an "IF" statement (involving some condition to be tested). The conclusion consists of a "THEN" statement which is executed when the conditions of the antecedent are met. In the following rule:

RULE for HMX (B-I)  
IF the energetic material is HMX  
THEN energetic material is known

the antecedent is "IF the energetic material is HMX" and the conclusion is "THEN energetic material is known." In addition, all rules in PRL must have a rule statement (name). In the above example, the rule statement is "RULE for HMX." RULE, IF and THEN are reserved words in PRL. In addition to their decision-making capabilities, some typical uses for rules include mathematical calculations, data base manipulation, report generation, external program activation, and text display.

Information accumulated during a run of the expert system shell is stored as the facts for that run. Level 5 supports four fact types: simple fact, attribute-value fact, numeric fact, and string fact. A simple fact is a phrase which can be thought of as being true or false. An attribute-value fact is one where we can associate some characteristic to an object or event. A numeric fact is one used in numeric computations or to hold numeric values (i.e., a numeric fact corresponds to a numeric variable in a programming language such as FORTRAN). A string fact is analogous to a string (or character) variable in a programming language. An example of the use of simple facts in Level 5 is:

RULE for energetic material selection (B-II)  
IF energetic material is HMX  
OR energetic material is RDX  
THEN energmat is nitramine

In Rule B-II, the phrases "energetic material is HMX," "energetic material is RDX," and "energmat is nitramine" are simple facts. As simple facts, these phrases are either true or false as a whole. This usage is contrasted to that of an attribute-value fact which can associate one or more values to an attribute. Consider the following rule which makes use of an attribute-value fact:

RULE for energetic material selection (B-III)  
IF energetic material IS HMX  
OR energetic material IS RDX  
THEN energmat is nitramine

Rule B-III looks very similar to Rule B-II -- the difference is seen in the case of the word "IS" in Rule B-III. In this rule, a single attribute ("energetic material") is used in place of the two simple facts ("energetic material is HMX" and "energetic material is RDX") used in Rule B-II. When using attribute-value facts, one can think of using an attribute with a list of possible values. The attribute is a property or thing and the value describes that property or thing. The value is assigned to the attribute through the use of the PRL reserved words IS or ARE. In the case of Rule B-III, the attribute "energetic material" can be assigned the value "HMX" or "RDX". Another way of conceptualizing the difference between simple facts and attribute-value facts is to consider the translations (into English) of Rules B-II and B-III:

B-II Translation:     If the energetic material is HMX or the energetic material is RDX, then the energetic material is a nitramine.

B-III Translation:    If the energetic material is HMX or RDX, then the energetic material is a nitramine.

The remaining fact types are numeric facts and string facts. As was previously mentioned, these fact types function in a similar manner to their counterparts in more conventional programming languages such as Fortran or Pascal. An example of a rule using a numeric fact is:

```
RULE for high pressure                                     (B-IV)
IF pressure > 10
THEN pressure is high
```

In this rule, "pressure" is a numeric fact which is assigned a value. If that value is greater than 10 (atm), the simple fact "pressure is high" is true. An example of a rule using string facts is:

```
RULE for RDX                                             (B-V)
IF energetic material = "RDX"
THEN energmat := "nitramine"
```

This rule demonstrates the use of two string facts, "energetic material" and "energmat." These string facts are assigned, respectively, the values "RDX" and "nitramine."

A Level 5 Knowledge Base source file consists of the following sections (listed in the order they should appear in the source file): title, declaration of shared facts, declaration of data bases, declaration of fact types, parameter initialization and control, control element selectors, knowledge base goals, rules, text displays, and the reserved word END. The sections preceding the rules can be thought of as a header to the knowledge base.

The first line of the knowledge base will be the title. A title consists of the reserved word TITLE followed by the name of the knowledge base. The name of the knowledge base cannot contain any of the Level 5 reserved words. If the Level 5 reserved word

DISPLAY follows the name, then a preliminary text display can be added to the title; the name of the knowledge base is limited to 60 characters. Examples of title sections are:

TITLE Propellant Ingredient Incompatibilities

or:

TITLE PROPELLANT INGREDIENT INCOMPATIBILITIES DISPLAY

This expert system shell is designed to flag propellant ingredient incompatibilities. Press <Enter> to continue.

Facts shared by different knowledge bases linked together (through the use of a CHAIN command) need to be declared as shared facts (these fact values can then be passed between the different knowledge bases as they are invoked). Shared facts are indicated by the PRL reserved word SHARED followed by the fact type (SIMPLEFACT, NUMERIC, ATTRIBUTE, OR STRING). Additional facts are declared as shared facts through the use of the PRL reserved word AND. An example of the declaration of shared facts follows:

SHARED SIMPLEFACT the energetic material is HMX  
AND energetic material is known  
SHARED NUMERIC pressure

This shared fact declaration identifies "the energetic material is HMX" and "energetic material is known" as shared simple facts, and "pressure" as a shared numeric fact.

Data bases (dBase II or dBase III format) are declared to be opened with the following format:

OPEN (data base name) AS (DB2 or DB3) FOR (READ or WRITE)  
CALLED (Level 5 handle)

The data base name is the name of the dBase II or dBase III data base without the .dbf file extension. DB2 and DB3 are PRL reserved words -- the data base version indicated should agree with the version of dBase used. The PRL reserved words READ and WRITE refer to the actions required from the data base. If a data base has been opened for READ, then it can not be written to; a data base opened for WRITE can be read or written to. The reserved word CALLED is an optional Level 5 handle for use within the knowledge base; if the CALLED option is used, then the handle must always be used to reference that data base. An example of the data base declaration is:

OPEN binder AS DB3 FOR READ

In this case, the dBase III data base "binder.dbf" has been opened as a read-only file.

The declaration of fact types section of the knowledge base is used to pre-declare the facts used in the knowledge base. Only STRING fact types are required to be pre-declared (to keep them from being confused with NUMERIC fact type by the PRL compiler). An example fact type declaration would be:

```
SIMPLEFACT the energetic material is HMX
AND energetic material is known
NUMERIC pressure
```

The declaration of fact types has a similar format to the declaration of shared facts section of the knowledge base.

The parameter initialization and control section of the knowledge base is used to initialize facts when the knowledge base is invoked. When the knowledge base is run for the first time, the INIT statement is used to initialize those facts which need pre-assigned values. For example, if "the energetic material is HMX" is a simple fact and "pressure" is a numeric fact, then these facts can be initialized at the start of a knowledge base run:

```
INIT the energetic material is HMX
INIT pressure := 11
```

At this point, "the energetic material is HMX" is set to true and "pressure" is set to 11 (atm). The INIT statement has no effect upon restart of the knowledge base through the use of a CYCLE command or when a knowledge base is started by a CHAIN command. In order to initialize facts under these conditions, a REINIT statement needs to be used. The syntax for REINIT is similar to that of INIT:

```
REINIT the energetic material is HMX
REINIT pressure := 11
```

One of the features of PRL is the CYCLE command, which allows the knowledge base to start over again. When the CYCLE command is used, all of the facts are retained (i.e., they are not reset), with the exception of those affected by a REINIT command. In order to reset all of the facts, the following command is used:

```
FORGET ALL
```

In addition, specific facts can be singled out by the FORGET command, for instance:

```
FORGET pressure
FORGET energetic material is known
```

Upon the execution of a CYCLE command, the facts "pressure" and "energetic material is known" will be reset. Here is another example of the use of INIT, and FORGET:



INIT the energetic material is HMX  
INIT pressure := 11  
FORGET the energetic material is HMX  
FORGET energetic material is known

In this example, when the knowledge base is run for the first time, "the energetic material is HMX" is set to true and "pressure" is set to 11 (atm). Upon the running of a CYCLE command, "the energetic material is HMX" and the "energetic material is known" are reset by the FORGET statements. A subtle difference between FORGET and REINIT is that FORGET resets a fact to an unknown (or unevaluated) state, while REINIT will set a fact to a specific known value or state.

PRL supports the use of control element selectors for customizing the operation of the expert system shell. These features are activated by the following reserved words: FILE, THRESHOLD, CONFIDENCE, MULTI, EXHAUSTIVE, GOALSELECT, SUPPRESS, and UNKNOWN FAIL/CONTINUE/OFF. These eight control element selectors will be discussed in some detail over the next few pages of this appendix.

The FILE statement is used (along with FILE commands used within rules) for creating a custom report file for the knowledge base session. The file statement is used to specify the name of the report file:

FILE report.txt

The THRESHOLD statement is used to specify the lowest level of confidence required for Level 5 to reach a conclusion or goal. The THRESHOLD statement is dependent on the CONFIDENCE statement. The threshold can be an integer ranging from 0 to 100, which corresponds to a percent certainty required. The threshold is assigned by using the statement:

THRESHOLD = X

where X ranges from 0 to 100. If no value is assigned, a default value of 50 is assigned by Level 5. The threshold can also be changed from within a rule. If this is done, then the new threshold applies until it is changed again in another rule. Level 5 supports the use of confidence factors (CF) through the use of the CONFIDENCE statement. In order to enable confidence prompting, the statement:

CONFIDENCE ON

must be used. If it is not necessary to enable confidence prompting, then the statement:

CONFIDENCE OFF

can be used; if no CONFIDENCE statement is used, Level 5 defaults to CONFIDENCE OFF. If the confidence is off, Level 5 associates answers to true (CF = 100) or false (CF = 0). If the confidence is on, the user uses a graphic confidence bar to enter the

confidence factor for a fact. A CONFIDENCE statement can also be used to specify that confidence factors be used for specific facts with CONFIDENCE OFF for the rest of the facts in the knowledge base:

CONFIDENCE the energetic material is HMX  
AND pressure

In the above example, the simple fact "the energetic material is HMX" and the numeric fact "pressure" have confidence factors enabled. Confidence prompting can also be turned on or off from within a rule, and confidence factors can be assigned in the conclusion of a rule (or by using the INIT or REINIT statements). An example of a rule which assigns a confidence factor is:

RULE for LOVA (B-VI)  
IF energetic material IS RDX  
OR energetic material IS HMX  
THEN propellant is LOVA CF=70

In Rule B-VI, if the energetic material used is RDX or HMX, then the conclusion "propellant is LOVA" is assigned a confidence factor of 70. The assignment of a confidence factor in the conclusion of a rule (or in an INIT or REINIT statement) does not require the use of the confidence bar.

Once an attribute-value pair has been verified, Level 5 will not pursue additional values of the attribute (the attribute is single-valued). The use of the MULTI statement lets attribute-value facts be multivalued. The MULTI statement can be used to declare all attribute-value facts as multivalued:

MULTI ALL

or only specific facts as multivalued:

MULTI energetic material  
AND plasticizer

In the above case, only the attribute-value facts "energetic material" and "plasticizer" will be multivalued. The EXHAUSTIVE statement is used to indicate that all rules involving a simple fact or an attribute-value fact are to be used to determine the value of that fact which has the highest confidence. For example, suppose that the attribute-value pair:

energetic material IS HMX

is assigned a confidence factor of 60. If the THRESHOLD is set to 50, then the required threshold to come to a conclusion has been exceeded and no further values for the attribute "energetic material" are searched for. However, when an EXHAUSTIVE statement is used, all of the rules used to evaluate "energetic material" would be fired. Lets say that the following attribute-value pair is assigned a confidence factor of 70:

energetic material IS RDX

In this case, the value for "energetic material" with the highest CF would be RDX, since HMX was assigned a confidence factor of only 60. The EXHAUSTIVE statement can be used for all simple and attribute value facts:

EXHAUSTIVE ALL

or for individual facts:

EXHAUSTIVE energetic material  
AND plasticizer

The GOALSELECT statement is used to let a user interactively narrow down the domain of the expert system prior to firing rules. The use of GOALSELECT allows the more knowledgeable user to quickly narrow down the expert system to a particular subdomain. The format for GOALSELECT is:

GOALSELECT ON

or:

GOALSELECT OFF

The GOALSELECT statement can also be placed within a rule. The SUPPRESS statement is used to suppress the display of the intermediate and final conclusions arrived at, which are presented at the end of a Level 5 consultation. SUPPRESS can be used to suppress all or some of the facts or goals of the expert system. To suppress all facts and goals, use the statement:

SUPPRESS ALL

Individual facts can also be suppressed:

SUPPRESS energetic material is known  
AND finished

where "energetic material is known" is a simple fact and "finished" is a goal.

The UNKNOWN statement is used to set the processing mode Level 5 uses to handle unknown (or unevaluated) facts. The syntax is:

UNKNOWN FAIL

or:

UNKNOWN CONTINUE

In the UNKNOWN FAIL mode, Level 5 will drop a line of reasoning once the user indicates that a fact needed is unknown. In the UNKNOWN CONTINUE mode, Level 5 pursues the line of reasoning as if the unknown fact were actually true. The default mode is UNKNOWN FAIL. One way of indicating a fact to be unknown is to use the UNKN function key. The statement:

#### UNKNOWN OFF

disables the use of that function key. UNKNOWN OFF has no effect on either UNKNOWN FAIL or UNKNOWN CONTINUE.

After the control element selectors comes the goals section of the knowledge base. Every Level 5 knowledge base must have at least one goal statement. A goal statement consists of a conclusion which can be reached by a knowledge base. Each conclusion must be preceded by a goal outline number. For example:

#### 1. Finished

In this example, there is only one goal statement. The conclusion of the expert system is "Finished," which would be used to indicate that the expert system consultation is over. Consider the following goal statements:

1. The propellant is nitramine-based
  - 1.1 The nitramine is HMX
  - 1.2 The nitramine is RDX
  - 1.3 The nitramine is something else
2. The propellant is AP-based
3. The propellant is double-base
4. The propellant is based on something else

With these goal statements, the knowledge base is dealing with propellant ingredient identification based on nitramine, ammonium perchlorate, double base, or other. If the propellant is nitramine-based, then a set of sub-goals are defined to determine what type of nitramine is used. In this case, if the user knows that the propellant is nitramine based, then GOALSELECT could be used to select "1. The propellant is nitramine-based" as a starting point, which would eliminate the three other primary goals from further consideration.

If text statements are used, the body of the text statements must be entered after the rules section of the knowledge base. The primary means for outputting text to the user of a Level 5 expert system is the DISPLAY command. DISPLAY is used to supply text to be output to the user:

RULE for HMX

(B-VII)

IF the energetic material is HMX  
THEN energetic material is known  
AND DISPLAY text for hmx

DISPLAY text for hmx

The selected energetic material is HMX.

In Rule B-VII, if "the energetic material is HMX" then the message "The selected energetic material is HMX." will be displayed to the user. Fact values can also be embedded into the DISPLAY text through the use of [ ]:

RULE for energetic material selection

(B-VIII)

IF energetic material IS HMX  
OR energetic material IS RDX  
THEN energmat is nitramine  
AND DISPLAY selected energetic material

DISPLAY selected energetic material

The selected energetic material is [energetic material].

In this case, if "energetic material" is HMX, then the user will see the message "The selected energetic material is HMX." A similar message is printed out if RDX is selected.

Many of the PRL statements and commands used in a Level 5 knowledge base are optional. The only sections of the knowledge base source code which are required are the title, declaration of string facts, knowledge base goals, rules, and the END statement. The other sections of the knowledge base are only required if additional features (linking of knowledge bases, data base access, etc.) are used. While this appendix is by no means meant to replace the Level 5 manual, it is intended to give the reader some indication of the capabilities of a Level 5 expert system.

INTENTIONALLY LEFT BLANK

## APPENDIX C: SAMPLE KNOWLEDGE BASE

INTENTIONALLY LEFT BLANK



## APPENDIX C: SAMPLE KNOWLEDGE BASE

This Appendix contains a sample knowledge base written using the Level 5 Production Rule Language.

!Beginning of the Knowledge Base source file. This is a comment line.

!Comments lines begin with the exclamation point.

! This sample knowledge base evaluates propellant formulation properties for each  
! record (formulation) in the propellant formulation data base.

!=====

!Knowledge Base Header:

!-----

!Title of the Knowledge Base:

TITLE Properties Prediction Expert System Module

!Declaration of shared facts No shared facts are used in this Knowledge Base.

!Open the data bases in dBase III format:

! Combine - propellant formulation data base.

! Energ - energetic materials data base.

! Plast - plasticizer data base.

! Binder - binder data base.

OPEN combine AS DB3 FOR WRITE

OPEN energ AS DB3 FOR READ

OPEN plast AS DB3 FOR READ

OPEN binder AS DB3 FOR READ

!Declaration of fact types. Only string facts need to be declared.

NUMERIC num1

AND num2

AND num3

AND num4

!num1 - num4 are general use numeric variables for use in user-supplied rules.

!Parameter initialization and control. INIT, REINIT, and FORGET statements go here.

FORGET record is ready

AND time to cycle

AND need to continue

AND rules fired

AND got energetic material

AND got binder

AND got plasticizer

!=>Put conclusions from user rules below:

AND got glass T

AND got burning rate

AND got ingredient cost  
AND got density

!Control Element Selectors. FILE, THRESHOLD, CONFIDENCE, MULTI, EXHAUSTIVE,  
!GOALSELECT, SUPPRESS, and UNKNOWN statements go here.

!Defaults:

! THRESHOLD = 50  
! CONFIDENCE OFF  
! GOALSELECT OFF  
! UNKNOWN FAIL  
EXHAUSTIVE ALL

UNKNOWN CONTINUE

SUPPRESS ALL

!Goal.

1. Properties Estimated

!=====

! RULE SECTION

!-----

! The following four rules set up the data bases prior to firing any rules used for  
! property estimation.

! These rules should not be modified by the user of this expert system module.

!-----

RULE to check end-of-file

IF NOT EOF combine

THEN record is ready

!This rule checks for the end of the formulation data base

RULE to find energetic material record

POSITION 0 IN energ

LOCATE energ.abbreviatn = combine.energ\_abbr IN energ

THEN got energetic material

!This rule finds the current energetic material in the energetic material data base.

RULE to find binder record

POSITION 0 IN binder

LOCATE binder.abbreviatn = combine.bind\_abbr IN binder

THEN got binder

!This rule finds the current binder in the binder data base.

RULE to find plasticizer record  
 POSITION 0 IN plast  
 LOCATE plast.abbreviatn = combine.plast\_abbr IN plast  
 THEN got plasticizer  
 !This rule finds the current plasticizer in the plasticizer data base.

!-----  
 !     START OF USER-SUPPORTED RULE SECTION  
 !     The rules listed below may be modified by the user of this module  
 !     to suit a specific application.  
 !  
 !     When adding rules to this section, the conclusions from any new rules  
 !     should be added to the FORGET section of Parameter Initialization and Control  
 !     and also to the antecedent of the RULE to fire property estimation rules.  
 !     If rules are deleted from this section, similar changes should also be made.  
 !     In addition, any string facts used in these rules should be declared above.  
 !-----

RULE to predict glass transition temperature  
 IF combine.flame\_temp > 0  
 THEN got glass T  
 AND num1 := binder.glass\_tran  
 AND num2 := combine.plast\_frac/combine.bind\_frac  
 AND num3 := plast.mol\_weight  
 AND combine.glass\_tran := num1 - (5000/num3)\*num2  
 ELSE NOT got glass T

RULE to predict burning rate for RDX-based Rocket propellant  
 IF combine.calc\_type = Rocket  
 AND combine.energ\_abbr = RDX  
 THEN got burning rate  
 AND num1 := -0.05957  
 AND num2 := 0.000147  
 AND num3 := combine.flame\_temp  
 AND combine.burn\_rate := num1 + num2\*num3  
 ELSE NOT got burning rate

RULE to calculate ingredient cost  
 IF combine.energ\_name = combine.energ\_name  
 THEN got ingredient cost  
 AND num1 := combine.energ\_frac\*energ.cost  
 AND num2 := combine.bind\_frac\*binder.cost  
 AND num3 := combine.plast\_frac\*plast.cost  
 AND combine.comp\_cost := num1 + num2 + num3  
 ELSE NOT got ingredient cost

```

RULE to calculate density of formulation
IF combine.energ_name = combine.energ_name
THEN got density
AND num1 := combine.energ_frac/energ.density
AND num2 := combine.bind_frac/binder.density
AND num3 := combine.plast_frac/plast.density
AND num4 := num1 + num2 + num3
AND num1 := 1/num4
AND combine.density := num1
ELSE NOT got density

```

!The following rule controls the firing of the user-supplied rules

```

RULE to fire property estimation rules
IF record is ready
AND got energetic material
AND got binder
AND got plasticizer
!=>Link in conclusions from user rules below:
AND got glass T
    OR NOT got glass T
AND got burning rate
    OR NOT got burning rate
AND got ingredient cost
    OR NOT got ingredient cost
AND got density
    OR NOT got density
!=>End of user link
THEN rules fired

```

```

!-----
!      END OF USER-SUPPORTED RULE SECTION
!-----
!      The following two rules are used to control the firing of all of the above rules and
!      advances the propellant formulation data base to the next record.
!      These rules should not be modified by the user of this module.
!-----

```

```

RULE to fire all rules (above) and advance database
IF rules fired
THEN time to cycle
AND ADVANCE combine
!This rule advances the data base after the property estimation rules have been fired.

```

```

RULE to cycle
IF time to cycle
AND EOF combine
THEN properties estimated
AND EXIT

```

ELSE need to continue

AND CYCLE

!This rule checks for the end of the data base file (which indicates that all of the  
!propellant formulation records have been checked). Upon reaching the end of the data  
!base file, the goal is met and this module ends. If the data base is not at the end,  
!the knowledge base CYCLES and the rules are fired for the next record.

!-----  
!       END OF RULE SECTION  
!-----

!Any text displays would go here: DISPLAY, TEXT, or EXPAND.

!End of the Knowledge Base:

END

INTENTIONALLY LEFT BLANK

## APPENDIX D: THERMOCHEMICAL DEFINITIONS AND CONSIDERATIONS

INTENTIONALLY LEFT BLANK



## APPENDIX D: THERMOCHEMICAL DEFINITIONS AND CONSIDERATIONS

This Appendix is used to define the thermochemical quantities used in the CET89 computer code. In addition, considerations of these quantities will be made regarding the CET89 UV=T and RKT=T options. The following are some of the quantities calculated by the CET89 thermochemical code:

T	Calculated flame temperature
H	Enthalpy
U	Internal energy
G	Gibb's free energy
S	Entropy
$\gamma_s$	isentropic exponent for combustion products
M	average molecular weight for combustion product gas
$c_p$	heat capacity at constant pressure

A few comments are warranted for some of the above quantities. The quantities H, U, G, and S are calculated by CET89 for the combustion products. CET89 uses the ideal gas equation of state:

$$PV=nRT \quad (D-1)$$

and determines the equilibrium composition by minimizing the free energy of the system. Two cases should be considered: constant pressure (RKT=T) and constant volume (UV=T). In the case of constant pressure, the Gibb's free energy is minimized; for constant volume, the Helmholtz free energy is used. Similarly,  $c_p$  is suitable for constant pressure problems; for constant volume, the heat capacity at constant volume,  $c_v$ , should be used. It is important to keep in mind that the output from CET89 consists of the quantities applicable to constant pressure problems regardless of which calculation option is selected.

The quantity  $\gamma_s$  is related to the heat capacity ratio  $\gamma$ , but is a slightly different quantity. The heat capacity ratio is defined as follows:

$$\gamma = \frac{c_p}{c_v} \quad (D-2)$$

The isentropic exponent is defined as a function of the heat capacity ratio:

$$\gamma_s = \frac{-\gamma}{(\partial \ln V / \partial \ln P)_T} \quad (D-3)$$

As it turns out,  $(\partial \ln V / \partial \ln P)_T$  is within a few percent of -1.00, so  $\gamma_s$  is a relatively good approximation to  $\gamma$ . If an exact value for  $\gamma$  is required, then equation (D-3) should be

used; however, the quantity  $(\partial \ln V / \partial \ln P)_T$  is calculated by CET89, but not currently imported into the propellant formulation data base. (CET89 probably calculates  $\gamma_S$  because this quantity is used to calculate the sonic velocity.) One additional point which should be emphasized about the  $c_p$  and  $\gamma_S$  calculated by CET89 is that these quantities assume an equilibrium composition which allows for a change in the product composition as the system is perturbed (such as by the addition or removal of heat, changes in pressure, etc.). The trend is for the equilibrium composition  $c_p$  to be larger (by up to several factors) than the corresponding frozen composition  $c_p$ , resulting in smaller values of  $\gamma$  in the case of equilibrium composition. (For frozen composition,  $c_p = c_v + nR$ , and  $\gamma = \gamma_S$ .)

Two additional quantities are calculated by CET89 as propellant performance parameters. In the case of RKT=T, the code calculates the specific impulse,  $I_{sp}$ , for the motor configuration and propellant combustion conditions.  $I_{sp}$  is used as a performance parameter in the rocket community. A similar type of performance parameter used in the gun community is that of impetus:

$$I = \frac{RT}{M} \quad (D-4)$$

In the case of a constant volume calculation such as UV=T, the CET89 code has been modified to support this calculation.