

2000

An Intelligent Failure Analysis System.

Claude Ray Mount

Louisiana State University and Agricultural & Mechanical College

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_disstheses

Recommended Citation

Mount, Claude Ray, "An Intelligent Failure Analysis System." (2000). *LSU Historical Dissertations and Theses*. 7379.

https://digitalcommons.lsu.edu/gradschool_disstheses/7379

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Historical Dissertations and Theses by an authorized administrator of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

AN INTELLIGENT FAILURE ANALYSIS SYSTEM

A Dissertation

**Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
In partial fulfillment of the
Requirements for the degree of
Doctor of Philosophy**

in

**The Interdepartmental Program in
Engineering Science**

by

**Claude Ray Mount
B.S., Louisiana State University, 1971
M.S., Louisiana State University, 1978
December 2000**

UMI Number: 9998698

Copyright 2000 by
Mount, Claude Ray

All rights reserved.

UMI[®]

UMI Microform 9998698

Copyright 2001 by Bell & Howell Information and Learning Company.

All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Copyright 2000
Claude Ray Mount
All rights reserved.

ACKNOWLEDGEMENTS

The greatest thanks and praise go to my Lord Jesus Christ. Except through His grace, I would not have had the courage to begin this work. But for His gifts, I would not have been able to complete the task.

I must express the great love felt for Debbie, my wife. She was my encourager. Without complaint, she made sacrifices and tolerated the many lonely hours.

Outside of academia, Eugene L. Gill willingly assumed an onerous duty. He ably dispensed relentless prodding for me to complete what I had begun. I owe him a tremendous debt of gratitude for his perseverance.

Any acknowledgement of personal accomplishment would be incomplete if it failed to mention parents. My father and mother, while raising me, laid the foundation to support me in my life. I literally would not be at this point but for their efforts.

Dr. T. Warren Liao (Department of Industrial and Manufacturing Systems Engineering) deserves immeasurable thanks. His patient encouragement kept me moving toward my goal. Dr. Ye-Sho Chen (Department of Information Systems and Decision Science) has been a staunch supporter. Special mention must be made of Dr. John M. Tyler (Department of Computer Science) who has graciously been a member of both of my graduate degree committees. Dr. Fereydoun Aghazadeh (Department of Industrial and Manufacturing Systems Engineering) suggested the germ of an idea from which this work grew. Dr. Armando B. Corripio (Department of Chemical Engineering) taught me about automatic process control and time series analysis, moving me along the path to become a failure analyst. A word of thanks also needs to be extended to Dr. Mark Davidson (Department of Mathematics) who served as the Dean's Representative for the Graduate School.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	iii
LIST OF TABLES	vii
LIST OF FIGURES.....	ix
LIST OF EQUATIONS.....	xii
ABSTRACT.....	xiii
 1. INTRODUCTION.....	 1
1.1 Explanation of Failure Analysis.	1
1.1.1 What Is Failure Analysis?.....	2
1.1.2 Why Is Failure Analysis Important?	4
1.1.3 Who Can Benefit from Failure Analysis?	6
1.2 Research History.....	7
1.2.1 Origin of the aIFAS Project.....	7
1.2.2 Chronology of aIFAS.....	8
1.3 Significant Discoveries.	10
1.4 The Next Step for aIFAS.....	13
1.5 Research Focus – The What, Why and How	14
1.5.1 A Prototype of aIFAS.....	15
1.5.2 Characterization of the Knowledge Base	15
1.5.3 Evaluation of Two Schemes for Case Representation	16
1.5.4 Comparison of Five Metrics for Case Selection.....	16
1.5.5 Exploration of Knowledge Base Size and Incremental Learning.....	17
 2. KNOWLEDGE ACQUISITION.....	 18
2.1 Acquisition Methods.	20
2.1.1 What Acquisition Tools Work.....	21
2.1.2 Building on Lessons Learned about Knowledge Acquisition	23
2.1.2.1 Attribute Set Selection.	24
2.1.2.2 Standardized Terminology.....	26
2.2 Information Sources.....	29
2.2.1 What Information Sources are Available.	29
2.2.1.1 Human Domain Experts.....	30
2.2.1.2 Publications.	31
2.2.1.3 Archived Reports.....	32
2.2.2 Building on Lessons Learned about Information Sources	32
2.2.2.1 Hierarchical Data Structure.....	33
2.2.2.2 Build Incrementally.	34
2.2.2.3 Completed Reports Offer the Best Data.	35
2.3 The Knowledge Engineer.....	35
2.3.1 What the Knowledge Engineer Must Do.	37
2.3.2 Building on Lessons Learned about Knowledge Engineers.....	38
2.3.2.1 Great Diversity of Required Disciplines.....	39

2.3.2.2 Expansive Scope Fosters Misunderstanding.....	39
2.3.3 The Same Person as Expert and Knowledge Engineer.....	40
2.3.3.1 Qualifications.....	41
2.3.3.2 Rationale for Assuming Both Roles.....	41
3. SYSTEM IMPLEMENTATION.....	43
3.1 General Methods for Knowledge Representation.....	45
3.2 The Case-Based Approach.....	46
3.2.1 Strengths of the Case-Based Approach.....	48
3.2.2 Weaknesses of the Case-Based Approach.....	49
3.2.3 Lessons Learned about the Case-Based Approach.....	50
3.3 The Rule-Based Approach.....	50
3.3.1 Strengths of the Rule-Based Approach.....	51
3.3.2 Weaknesses of the Rule-Based Approach.....	52
3.3.3 Lessons Learned about the Rule-Based Approach.....	52
3.4 Neural Networks.....	53
3.4.1 Strengths of the Neural Network Approach.....	53
3.4.2 Weaknesses of the Neural Network Approach.....	54
3.4.3 Lessons Learned about the Neural Network Approach.....	54
3.5 Metrics for Case Comparison.....	55
3.5.1 Conventional Distance and Similarity Measures.....	55
3.5.1.1 City Block (Manhattan) Distance.....	55
3.5.1.2 Cosine Correlation Similarity.....	56
3.5.1.3 Euclidean Distance.....	57
3.5.1.4 Hamming Distance.....	57
3.5.2 The Knowledge Graph Similarity.....	58
3.5.2.1 How KGS Works.....	59
3.5.2.2 A Modification of KGS.....	62
3.5.2.3 Justification of KGS.....	63
3.6 The Logic of Fuzzy Logic.....	64
4. THE aIFAS FRAMEWORK.....	66
4.1 Main Purpose of aIFAS.....	66
4.2 The Basic aIFAS Structure.....	67
4.2.1 The User Interface.....	67
4.2.2 Modular Architecture.....	68
4.3 Individual Functions of aIFAS.....	69
4.3.1 Support Parametric Studies.....	69
4.3.2 Direct Failure Investigations.....	70
4.3.3 Stepwise Entry of Test/Unknown Cases.....	71
4.3.4 Suggest Most-Likely Failure Modes.....	72
4.3.5 Queries of Stored Information.....	73
4.3.6 Retrieve/Display Information.....	73
4.3.7 Auxiliary Data Management.....	73
4.3.8 Data Entry Checking.....	74
4.3.9 Fuzzification of Case Attributes.....	74

5. TESTING THE SYSTEM	76
5.1 The aIFAS Knowledge Base.	77
5.1.1 Choosing Cases for aIFAS.	77
5.1.2 Knowledge Base Structure.	78
5.2 Verification of Capabilities	80
5.2.1 Directing a Failure Investigation	80
5.2.2 General Information Query.....	82
5.2.3 Learning from aIFAS	83
5.3 Validation of Performance Accuracy.....	84
5.3.1 Performance with Test Cases.....	84
5.3.1.1 One Time Testing	85
5.3.1.2 Multiple Trail Testing.....	85
5.3.2 Acceptable Results.	86
5.3.3 Measuring Performance.....	86
6. RESEARCH RESULTS	88
6.1 Extra Information.....	88
6.2 Metric Evaluation.	91
6.2.1 Distance or Similarity.....	92
6.2.2 Normalization	96
6.2.3 Extraneous Attributes	99
6.2.4 Combining KGS Terms.....	104
6.2.5 Modified Knowledge Graph Method	109
6.2.6 Step and Prune Operation.....	111
6.2.7 User Entry Simulation.....	116
6.2.8 Incremental Learning	120
7. CONCLUSIONS	128
7.1 Discussion	128
7.2 The aIFAS Prototype	130
7.2.1 Using aIFAS for Failure Analysis.....	130
7.2.2 Using aIFAS with Other Knowledge Domains	131
7.3 Recommendation for Future Work.....	131
REFERENCES.....	133
APPENDIX A: SCREEN VIEWS OF aIFAS	144
APPENDIX B: EXTRA INFORMATION TABULATIONS.....	151
APPENDIX C: METRIC COMPARISON TEST DATA.....	152
VITA.....	181

LIST OF TABLES

4.1	Example Attribute Possibility Values.....	75
5.1	Distribution of Failure Modes and Attributes in the Stored Cases.....	79
5.2	Number of Attributes Used to Describe a Failure Mode	79
6.1	Explanation of Parameter Settings.....	91
6.2	Parameter Settings for Distance/Similarity	92
6.3	6x15 TC Results for Distance/Similarity with Independent Attributes	95
6.4	6x15 TC Results for Distance/Similarity with Grouped Attributes.....	95
6.5	The Maximum Computed Metric Values.....	95
6.6	Sort Order Metrics Use for Selecting Matching Cases	96
6.7	Parameter Settings for AS-Computed/Normalized Metric	97
6.8	6x15 TC Results for As-Computed/Normalized with Independent Attributes ..	99
6.9	6x15 TC Results for As-Computed/Normalized with Grouped Attributes.....	99
6.10	Parameter Settings for Include/Exclude Attributes	100
6.11	6x15 TC Results for Include/Exclude with Independent Attributes.....	102
6.12	6x15 TC Results for Include/Exclude with Grouped Attributes	102
6.13	Knowledge Graph Times for Coefficient Set Computation.....	105
6.14	Parameter Settings for KG Combination and Weighting.....	105
6.15	6x15 TC Results for KG Combinations/Weights with Independent Attributes	108
6.16	6x15 TC Results for KG Combinations/Weights with Grouped Attributes	109
6.17	Parameter Settings for Modified KG Combination and Weighting	110
6.18	6x15 TC Results for Modified KG Combo/Weights with Grouped Attributes	111
6.19	Parameter Settings for Step and Prune Threshold	112
6.20	6x15 TC Results for 10%-10/25%-25 with Independent Attributes	115

6.21	6x15 TC Results for 10%-10/25%-25 with Grouped Attributes.....	115
6.22	Parameter Settings for User Entry Simulation	116
6.23	6x15 TC Results for User Entry Simulation with Independent Attributes	119
6.24	6x15 TC Results for User Entry Simulation with Grouped Attributes.....	120
6.25	Example Cases and Available Test Case Solutions.....	120
6.26	Parameter Settings for Incremental Learning.....	121
6.27	6x15 TC Results for 600 Example Cases.....	124
6.28	6x15 TC Results for 480 Example Cases.....	124
6.29	6x15 TC Results for 360 Example Cases.....	124
6.30	6x15 TC Results for 300 Example Cases.....	125
6.31	6x15 TC Results for 240 Example Cases.....	125
6.32	6x15 TC Results for 180 Example Cases.....	125
6.33	6x15 TC Results for 120 Example Cases.....	126
6.34	6x15 TC Results for 90 Example Cases.....	126
6.35	6x15 TC Results for 60 Example Cases.....	126
6.36	6x15 TC Results for 30 Example Cases.....	127

LIST OF FIGURES

1.1	The Failure Analysis Process.....	3
2.1	The Knowledge Acquisition Process.	36
4.1	The Basic aIFAS User Interface	68
4.2	Parameter Controls for aIFAS.....	69
4.3	Threshold Values for aIFAS	70
4.4	The Set of Program Control Buttons Used to Guide Analyses.....	71
4.5	aIFAS Produces Optional Real-Time Results.....	72
4.6	One of the Ways In Which the aIFAS Knowledge Base Can Be Queried	73
4.7	A Word Description, Combined with Graphic Documentation to Provide User Information	74
5.1	Representation of a Case Using Binary Attribute Values	78
5.2	Representation of a Case Using Grouped, Interval-Valued Attributes	78
5.3	How Control Buttons and Messages Direct aIFAS Operation	80
5.4	The Two Tabs that Can Be Accessed on the First Visit to <i>Data Entry</i>	81
5.5	The Third Visit to the <i>Data Entry</i> Module Allows Access to Four Tabs.....	81
5.6	aIFAS Message Generated IF Required Data Is Not Entered	81
5.7	An Error Message Resulting When Improper Data is Entered.....	82
5.8	Retrieval of a Report Underlying an Example Case in aIFAS for User Review	82
5.9	The Control Used to Pick the aIFAS Data Displays	83
5.10	An Example of the Information Retrieved By Selecting an Industry Type While In the User Entry Module.....	83
6.1	Failure Mode Classification System.....	89
6.2	Failure Mode Distribution by Division and Causal Classes	90
6.3	Failure Mode Distribution by Division and Mechanical Classes	90

6.4	Failure Mode Distribution by Divisions and Environmental Classes	90
6.5	Descending v. Ascending for CR – Independent Attributes.....	93
6.6	Descending v. Ascending for CR – Grouped Attributes	93
6.7	Descending v. Ascending for PS – Independent Attributes	94
6.8	Descending v. Ascending for PS – Grouped Attributes.....	94
6.9	As-Computed v. Normalized for CR – Independent Attributes	97
6.10	As-Computed v. Normalized for CR – Grouped Attributes	97
6.11	As-Computed v. Normalized for PS – Independent Attributes	98
6.12	As-Computed v. Normalized for PS – Grouped Attributes.....	98
6.13	Include v. Exclude Attributes for CR – Independent Attributes.....	100
6.14	Include v. Exclude Attributes for CR – Grouped Attributes	101
6.15	Include v. Exclude Attributes for PS – Independent Attributes.....	101
6.16	Include v. Exclude Attributes for PS – Grouped Attributes	102
6.17	KGS Term Combinations/Weighting for CR – Independent Attributes	106
6.18	KGS Term Combinations/Weighting for CR – Grouped Attributes.....	106
6.19	KGS Term Combinations/Weighting for PS – Independent Attributes	107
6.20	KGS Term Combinations/Weighting for PS – Grouped Attributes.....	107
6.21	Modified KGS Term Combinations/Weighting for CR – Grouped Attributes .	110
6.22	Modified KGS Term Combinations/Weighting for PS – Grouped Attributes ..	110
6.23	Threshold Testing of Step and Prune for CR – Independent Attributes.....	113
6.24	Threshold Testing of Step and Prune for CR – Grouped Attributes	113
6.25	Threshold Testing of Step and Prune for PS – Independent Attributes	114
6.26	Threshold Testing of Step and Prune for PS – Grouped Attributes.....	114
6.27	User Entry Simulation for CR – Independent Attributes	117

6.28	User Entry Simulation for CR – Grouped Attributes.....	117
6.29	User Entry Simulation for PS – Independent Attributes	118
6.30	User Entry Simulation for PS – Grouped Attributes.....	118
6.31	Incremental Learning for CR – Independent Attributes.....	121
6.32	Incremental Learning for CR – Grouped Attributes	122
6.33	Incremental Learning for PS – Independent Attributes.....	122
6.34	Incremental Learning for PS – Grouped Attributes	123

LIST OF EQUATIONS

3.1	City Block Distance	56
3.2	Cosine Correlation Similarity.....	56
3.3	Euclidean Distance.....	57
3.4	Hamming Distance.....	58
3.5	KGS Sensitivity Term.....	60
3.6	KGS Specificity Term.....	61
3.7	KGS Mode Selection	61
3.8	KGS for New and Example Cases.....	62
3.9	Modified KGS Sensitivity Term.....	63
5.1	Correctness Ratio	87
5.2	Relative Time Unit.....	87
5.3	Performance Score.....	87
6.1	KGS Combination Functions	104
6.2	Weighted Metric	104

ABSTRACT

The investigation of commercial/industrial failures is a vital, but complex task. This paper presents an Intelligent Failure Analysis System (aIFAS). It is a system designed by a failure analyst with the goal of making failure investigation easier.

The knowledge base for aIFAS comes from commercial laboratory reports. The methodologies employed represents the experience gained from over five years of development. One goal of aIFAS is to provide a case-based expert system tool to help find answers. Functionality ranges from matching a new case to stored example cases to extracting relational data from the aIFAS knowledge base.

This study focuses on two objectives beyond implementation of aIFAS. First, a more compact file structure to represent the failure mode/attribute data is explored. Second, five candidate metrics for case matching are compared.

Comparisons are accomplished using a parametric analytic engine built into aIFAS. Combinations of features are tested against a single set of fifty cases, as well as, with multiple trials of randomly selected cases. The Relative Time Unit and Performance Score measures are introduced. They offer a semi-quantitative yardstick that introduces both accuracy and speed into the assessment process.

A more compact, grouped format for attribute representation gave improved performance. It shows promise as a means to inject fuzzy logic into aIFAS. The City Block and Hamming distance algorithm were the most stable and efficient metrics.

1. INTRODUCTION

We are each one of us, from time to time, faced with the need to determine why something is not doing what it should. At the risk of over simplification, each of those riddles is a failure analysis. Such investigations might range from troubleshooting an air conditioner that quit cooling to dealing with a bed of wilting zinnias.

Magnify the scale of those problems to a size that can cost hundreds of thousands of dollars a day and you have the equivalent of an industrial/commercial failure analysis (henceforth referred to as failure analysis). That class of failure is the one considered by this work. Such failures demand thorough investigations that conclude by yielding quality answers. In some cases, lives literally depend on the accuracy of the failure analysis process.

The practice of conducting failure analyses is not new. What is new is the notion of automating the process in an innovative fashion that capitalizes on the synergism of man-machine interaction. The description, prototype structure, and test results for a knowledge system, dubbed as an Intelligent Failure Analysis System (aIFAS), are presented in this work. Collectively, the aIFAS project epitomizes the essence of a keynote address given before the 44th Annual National Metal Congress in 1962 by Admiral Rickover [Thielsch, 1977] wherein he said,

“We must accept the inexorably rising standards of technology, and we must relinquish comfortable routines and practices rendered obsolete because they no longer meet the new standards. This is our never-ending challenge.”

Admiral H.G. Rickover

1.1 Explanation of Failure Analysis

From a technical perspective, the old American Society for Metals (now called ASM International) Metals Handbook: Failure Analysis and Prevention [Powell, 1986] defines failure as the cessation of function or usefulness. (It then follows immediately

that a failure analysis is the process of investigating such a failure.) The rather terse definition might strike one as being overly broad in scope. It, however, quite accurately illustrates the character of the sort of failure analyses addressed by this work. Failure analysis, in its full and complete sense, is a wide-ranging knowledge domain that encompasses many technical disciplines and a variety of fields of scientific study.

A frequently used example for describing failure analysis is the familiar and generally well understood practice of medical diagnosis. The two methodologies closely parallel one another [Adlassnig, 1986] in their cycle of operation, information requirements, and level of expertise for successful performance. The goal of both kinds of investigation is to apply knowledge of cause-and-effect relationships to correctly link symptoms with causes [Becraft and Lee, 1993] for subsequent remediation.

Diagnostic tools very likely represent the largest area of application [Morales and Garcia, 1990] for artificial intelligence systems. The sheer number of implementations serves to indicate their scientific and commercial importance. Those systems are, however, extremely focused in their application. Except for the work proximate to this research, no fully implemented system has been discovered that addresses the full breadth of failure analysis. That is, a system that can support a failure investigation and subsequent analysis to determine the primary cause of a failure [Ryder *et al.*, 1975] in an unbounded operating environment.

1.1.1 What Is Failure Analysis

Failure analysis has been termed a means for linking effects to causes, much like medical diagnosis. The process is a methodical approach applied to situations for the express purpose of finding an answer to a problem. It is most often a sequential process, with subsequent results building on earlier discovery [Mount *et al.*, 1997].

Stepping through the process entails conducting examinations and tests in a logical order that hopefully minimizes the amount of work needed to provide an accurate assessment. The cycle of activities is usually iterative, repeating a pattern of steps [Milacic *et al.*, 1988] that includes information acquisition, hypothesis formulation, testing, and evaluation of results. Figure 1.1 is a process flow diagram illustrating the steps in a generic failure analysis type of investigation.

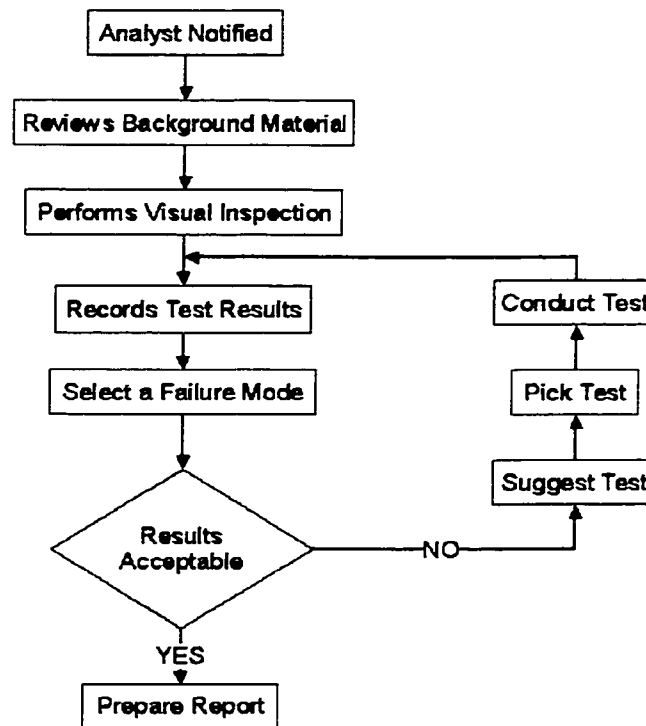


Figure 1.1: The Failure Analysis Process.

The preeminent objective of failure analysis is to discover why a piece of equipment, structural component, or virtually any engineered system or object failed [Zhang, 1998]. Intuitively this sounds like a relatively easy task. Contrarily, it may even be necessary to determine if a failure did in fact occur. For example, consider the case [Padalkar *et al.*, 1991] where a pipe has a small, through-wall hole – if the pipe transports fluid it has failed, yet if the pipe is merely a structural support it has not failed.

1.1.2 Why Is Failure Analysis Important

Failure analysis is in no way a modern day concept. There was concern regarding responsibility for engineering activity dating to more than 4000 years ago. Take note of the excerpt from the ancient rules of a Babylonian kingdom [Harper, 1994]:

"If a builder build a house for a man and do not make its construction firm and the house which he built collapse and cause the death of the owner of the house – that builder shall be put to death...

If it destroy property, he shall restore whatever it destroyed, and because he did not make the house which he built firm and it collapsed, he shall rebuild the house which collapsed at his own expense."

King Hammurabi (2297-2254 BC)

The response to lax construction methods is extreme by today's standards but serves to illustrate the importance of determining why failures occur. Failure analysis is a critical step in resolving matters ranging from material loss to loss of life. (The cited text actually speaks to the issue of root-cause failure analysis that identifies the exact reason for a failure. Root-cause analysis is a very comprehensive study, well beyond the goal of this work to specify a most likely a failure cause.)

In contemporary times, the practice of conducting failure analyses and reporting the results has a long formal tradition. The British Engine Insurance Co. has been reporting investigation details since 1879 [Hutchings and Unterweiser, 1981]. Their initial reason for doing so was to make collective experience available so that the many could profit from the misfortune of the few.

The educational value of failure analysis is another legitimate reason for which failure analyses are conducted. The results both teach the novice as well as document rare events. An excerpt from the Preface of the Chief Engineers Report of 1906 for the British Engine Insurance Co. [Hutchings and Unterweiser, 1981] states,

“...so long as there are young ones growing up to make the like mistakes and suffer the same troubles as their elders, unless warned by records of the experience of the past. And particularly is this true when the experiences recorded are of a kind that comparatively few have opportunities of gaining.”
M. Longridge, 1906

Retaining institutional memory is another reason to perform failure analyses.

Twenty years of personal experience in a commercial testing laboratory (that conducted failure analysis) saw many new faces with long time client companies. Turnover of staff can, and most often does, result in a loss of valuable institutional memory. Conducting and documenting failure analysis helps preserve that knowledge. In some cases it even provides a means for avoiding the repetition of mistakes [Klinger, 1994].

Failure analysis is not only important for determining the reasons why something fails, but also for developing a better understanding of how the device fails [Hui *et al.*, 1993]. The acquired knowledge not only helps prevent recurrence, but also is a valuable resource for formulating alternate, improved designs [French, 1983]. In other words, learning about the faults of a device can offer the bonus of learning how to build it better.

It is trite to say that we live in a litigious society. Yet, the potential for legal action over the past two decades has been the driving force behind many failure analyses [Esakul, 1992]. The strength of that motivation may have even overridden the impetus of safety or global competitiveness.

Safety and product reliability can be the very worthwhile rewards of failure analysis [Powell and Mahmoud, 1986]. The systematic identification and exploration of problems is the starting point. Once the mode of failure has been identified, correct measures can be taken – often in circumstances where lives literally hang on the accuracy of the failure analysis process [Wulpi, 1985].

Lastly, it is economical to conduct failure analyses. Prediction of when to make repairs or replace components is a perennial, universal problem [Stottler, 1994].

Typically, when the inevitable failure occurs, it is corrected in haste, and vigilance is relaxed until it happens again [Fincher, 1979]. The knowledge gained from failure analysis is a critical component when designing a preventative maintenance system.

1.1.3 Who Can Benefit from Failure Analysis

Everyone can benefit from failure analysis. To illustrate that, here are a few examples drawn from personal experience in conducting failure analysis. Discovering air bubbles in the leg of a plastic step stool helped an elderly woman gain compensation for medical expense incurred by a fall. Identifying oxygen pitting due to improper idle storage conditions in a church boiler helped design a new shutdown procedure that avoided frequent leak repairs. Proving that improper grounding of a large electrical motor was causing rust contamination helped return a well in a municipal water system to use. And the list goes on to include such things as a highway bridge support bearing, towers in an electric power transmission line, the cargo hold floor plates in a phosphate ore transport ship, or any one of numerous failures analyses from the past twenty years.

Formal references to “Who can benefit from failure analysis?” yields such instances as these. To a very large extent, the development of material science and engineering has resulted because of serious failures [Brostow and Corneliussen, 1986] (and their subsequent investigation). In the processing industries, plant operators depend on failure analysis results to efficiently and effectively control their systems [Becraft and Lee, 1993]. In addition, many students in undergraduate engineering courses would not be exposed to knowledge of how real world systems work without the results of exemplar failure analyses [Barer and Peters, 1970].

1.2 Research History

In the latter part of 1994, a project was begun to archive 18 years of industrial failure analysis records. The initial intent was to simply catalog the modes of failure and group them by broad industrial classes. For example, how many fatigue-cracking failures occurred in rotating equipment operating within the power generation industry?

The sheer volume of information pointed the investigation in another direction. The richness and variety of data would readily support the development of an expert system. Things progressed and a series of prototype systems were conceived that could resolve the causes for industrial failures. As time passed, capabilities were expanded; not only could problems be solved, but individuals could learn how to solve problems.

1.2.1 Origin of the aIFAS Project

The aIFAS project came into being after considerable research, trials, errors, and restarts. During its course, the effort has drawn on the resources of as many as twelve researchers and two corporate partners.

The project began with a suggestion made to the Author by Dr. Fereydoun Aghazadeh. His idea was to categorize the archived failure analysis reports from Scientific Testing Laboratories (the Author's employer at the time). The archived records spanned twenty years of service to a wide range of clients. As a minimum, the project would have merit as a source of statistical data for preventive maintenance.

With input from Dr. T. Warren Liao the scope expanded. An initial knowledge base was to be created by archiving records in electronic form. That resource provided the information for creation of a fledgling failure analysis expert system.

A natural next step was the growth of the project into an enterprise-wide information system. Adding business management functionality made sense. That effort

was aided by Dr. Ye-Sho Chen. He further assisted in a collaborative effort resulting in an LEQSF Grant (1995-97-RD-B-04) to fund the work.

The current state of the project is more focused. A best-of-breed approach is being used to glean valuable insights from earlier efforts. The goal is to produce an information resource tool for an individual; in particular, an independent consultant in the business of failure analysis or when offering expert legal testimony.

During system development, an important feature has been consistently adhered to and is a secondary driving force for the project. The decisions that human experts make, rely heavily on their working experience in the problem domain. Thus, a neophyte analyst may take longer time to conduct a failure analysis, or even worse, come up with a wrong conclusion because of inexperience [Zhan, 1998]. As with earlier generations of the aIFAS program, it is important that the knowledge of failure analysis experts is preserved through some form of an educational functionality within the system.

1.2.2 Chronology of aIFAS

Since the project began in 1994, there have been many lessons learned. Upon reflection, most pitfalls mentioned by other researchers have also been experienced.

Initially, time was given to researching methods for knowledge acquisition. The idea of simple interviews was explored and abandoned. It was determined that only a failure analysis expert can effectively interview a failure analysis expert. Questionnaires to elicit attributes of failure modes were studied. They worked; however, the method is slow in producing significant quantities of data. A further burden is the complexity of the questionnaires needed for the failure analysis knowledge domain. Cause-and-effect matrices were considered and found to present difficulties similar to the use of

questionnaires. Electronic scanning with optical character recognition followed by data mining of paper reports was left for future consideration.

Time was then spent examining what sort of expert system should be utilized. The choices fall into three classifications: rule-based, generated from cause-and-effect relationships; case-based, developed by classifying the attributes of solved cases; and, neural-nets trained from sets of stimulus-response data. Within those classes are the indeterminate variations which would invoke the use of fuzzy logic.

Rather than intellectualizing the perfect system, it was decided to begin with a simple data management application. The purpose was to create a vehicle for gathering case histories of failure analyses in a consistent fashion. The system could collect the expert's knowledge as it was being used to solve problems.

Difficulties arose with attempts at implementation. Discrete pieces of the system could be conceptualized, created, and, even tested. Those pieces were, nevertheless, not interconnected as a system and the merit of the concept could not be measured.

To connect the pieces as a functional system, a set of decision rules was needed. Brainstorming suggested a primitive, but effective approach. Use the very process people employ everyday for solving problems. That process can be simplified into: "Do I know the answer; if yes, then access that knowledge; if no, then choose where to find an answer." This is nothing more than applying the principles of scientific reasoning. The name Scientific Thinking Language (STL) was adopted. That term can refer equally well to the process-of-elimination approach used when solving failure analysis cases; or, when managing the convoluted structure of an expert system.

The STL concept now offered a scheme with which a viable failure analysis expert system could be developed. The first effort was a passive, data-gathering tool. It

was not intended to actually solve problems; rather, it captured the various data produced by failure analysis processes. It was assumed that from the mass of accumulating data, the necessary features and the semblance of a functional structure could be inferred. The groundwork would be established from which a full-featured expert system might evolve.

Serendipity prevailed. Using STL, an expert system is not constrained to narrowly defined problems with a limited knowledge base. A system can not only know its bounds, but also be able to respond appropriately when it met those restraints. Furthermore, the system could deal with a much more complex interplay of functionality. In a sense, such an expert system could ask questions outside its realm of knowledge, process the answers, and then proceed to do or learn more.

This version of the concept, called an Integrated Knowledge Engine (IKE), was a hyper expert system. IKE was envisioned as being: easily trainable; adaptable to a nearly unlimited range of activities; able to integrate multiple, complex functionality; and, ultimately capable of some level of independent operation. IKE would transcend a conventional expert system, offering the user comprehensive information processing.

The IKE concept was presented to the Artificial Neural Networks in Engineering Conference (Annie 1997) by the Author [Mount *et al.*, 1997]. Circumstances changed the imperative to implement such a comprehensive system. What was needed instead was a means to assist failure analysis efforts done on a less frequent basis. From the newly defined functionality emerged the idea of an Intelligent Failure Analysis System (aIFAS).

1.3 Significant Discoveries

In the course of previous efforts, issues dealing with rule-based systems, case-based systems, neural networks, the elicitation of knowledge, and management of domain

experts were encountered. The more memorable and hopefully significant of those experiences are synopsized in the following.

In a rule-based construct [Liao *et al.*, 2000a] a system using 477 cases and 59 attributes generated a large number of redundant rules with 18% of the cases being exceptional (no rule applicable). The cause was determined to stem from numerous training cases with noisy or overlapping data. This suggests that either the knowledge acquired was faulty (an issue not tested by the research) or the nature of failure analysis domain data may not be well suited (without extensive conditioning) to representation with a rule-base system.

The rules derived, however, tended to be simple and involving few attributes [Liao *et al.*, 2000b]. That finding is consistent with other work in rule-based systems [Polat and Guvenir, 1993] that indicated most systems have many short rule chains rather than a select few long rule chains. This is an encouraging discovery for a knowledge domain as complex as failure analysis. It shows that distinctions can be easily determined, requiring only a small set of attributes. Simple rules that do not compromise accuracy are desirable, because they are more likely to classify objects [Yaun and Shaw, 1995]. The inherent danger is creating an oversimplified representation [Johannsen and Alty, 1991] that might well be useless. In general, though, besides being easier to facilitate in a system, simple rules are more likely to tolerate missing or imprecise data [Yaun and Shaw, 1995].

Case-based systems appear to be a strong candidate for representation of failure analysis domain knowledge. Work in that area showed 100% accuracy could be achieved for standard cases and as high as 71% accuracy for exceptional cases [Liao *et al.*, 2000c]. This level of performance comes with a price. Case-based systems usually consist of a

large number of attributes, as illustrated by an implementation that used 90 attributes in 7 category groupings [Zhang, 1998]. Albeit there might be a perceived potential for selecting too many attributes, (in fact, verifying when a sufficient set exists is problematic), but it is easy to detect inappropriate ones [Liao *et al.*, 2000a]

The method employed to search through cases in the system for a solution is of prime importance. The efficiency of case retrieval can be greatly enhanced through attribute weighting and optimization of those weights can even serve to indicate superfluous attributes. By assigning threshold limits for similarity measures between test cases and nearest matches from the case base, standard and exceptional cases can be differentiated [Zhang *et al.*, 1997].

There is a need for caution in system design. For large systems, the similarity measurement algorithm needs to be computationally efficient. In one instance, computing the weightings for a genetic algorithm approach required 10 hours of computing time [Zhang, 1998].

Early on, it was obvious that with current technology, the application of neural networks to failure analysis was not feasible. Implementation of the methodology to create an engineering metal alloy identification system was, however, quite successful. The system identifies 90 commonly used/occurring materials with 100% accuracy when a 15% error tolerance was allowed. (That may seem excessive, however, materials can be within specification while having as much as $\pm 10\%$ variation in each one of their individual properties.) The final system architecture was comprised of specially tailored modular components [Garcia, 1997].

In the realm of knowledge elicitation, it is important to remember that the reasoning process of failure analysis does not follow an exact mathematical model and is

often accomplished with incomplete and ambiguous information [Zhan, 1998]. The net effect for the knowledge engineer is that there can be a very high percentage of unknown attribute values when considering historical data. Missing information is one problem, understanding what is available is another. Using pictures instead of text alone (combined with a wordbook of synonyms) to describe attributes would minimize identification errors [Liao *et al.*,2000a]. The information source is yet another matter for concern. There needs to be some means of conflict resolution among multiple experts for successful system development [Liao *et al.*,2000a].

Without argument, a wealth of knowledge was acquired getting to this point. Of those gems of wisdom, three are immediately applicable to the current effort.

- A case-based system implementation is the best choice for failure analysis.
- Simplicity is the secret for coping with the complexity of failure analysis.
- Good communication via both detailed text and graphic rendering is essential.

1.4 The Next Step for aIFAS

A prototype of the extant aIFAS concept is implemented as a hybridization of a research tool, comprehensive information management tool, and a problem solving aid. The system offers a variety of functions for studying performance with differing operating parameters, extracting knowledge from stored data, or finding the failure mode for a newly entered case. This is accomplished in aIFAS via a graphic-user-interface for communication with supporting computer-based modules.

The predecessor IKE system was to be an enterprise-wide method for managing everything – project receipt, scheduling and resource management, invoicing and collections, and even archival storage. IKE was to incorporate parallel reasoning systems and had a convoluted structure. That level of sophistication is no longer required.

What is needed, is an efficient personal assistant to maintain and support private consulting activities. That functionality can be provided by aIFAS. The system concept is derived using a best-of-breed approach combined with the expertise gained from several years of investigating the tools that can be used to support failure analysis.

The formalization of aIFAS creates a sort of artificial materials specialist that, if made public, would support the statement "...a cursory look by a materials specialist is more useful than many an irrelevant inquiry" [Naumann, 1983]. More pragmatically, such a system offers a many-fold improvement over traditional methods and can pay for itself if one error is avoided [Marra, 1997]. Any aIFAS-like system releases the power of being able to say, "Have we ever ...?" and accessing corporate memory or acquired expertise [Klinger, 1994] to find the answer.

Waxing philosophical, aIFAS represents a paradigm shift for performing failure analysis. It is envisioned as being at the heart of the notion of lifelong learning as it is expressed in the following [Goldstein, 1997],

"In a time of drastic change it is the learners who inherit the future. The learned usually find themselves equipped to live in a world that no longer exists."
Eric Hoffer

When all is said and done, with the proposition and development of any system, the bottom line is, "Will the system add value?" [Lee and O'Keefe, 1994]. The fully implemented aIFAS concept will add value to the process of failure analysis.

1.5 Research Focus – The What, Why, and How

This paper and the supporting research address six subjects.

- Foremost, is the presentation of a working model of aIFAS.
- It was necessary to know just what information content the aIFAS knowledge base would manage.

- The data structure to represent the example cases, specifically the failure mode and attribute information, needed to be decided.
- Of the numerous methods for comparing cases, an effective and efficient metric needed to be selected.
- Then, once the prototype existed, it needed to demonstrate that it worked and that it could learn.
- Lastly, this work chronicles the efforts that brought aIFAS to this point.

1.5.1 A Prototype of aIFAS

An augmented working version of aIFAS was produced. The set of system capabilities discussed in Chapter 4 (The aIFAS Framework) are implemented. Beyond those basic functions, a much wider range of extended user controls were incorporated for use. The additional system features were added as necessary features to support research activities dealing with the selection of a data structure for mode/attribute storage, for testing of candidate metrics for case comparison, and to study the effects of knowledge base size on system performance.

1.5.2 Characterization of the Knowledge Base

Until this point there has only been conjecture about the actual contents of the failure analysis knowledge base. It has been looked into in a limited manner to provide cause-and-effect data to test assorted hypotheses; that is, which sets of attributes correspond with what mode of failure. The knowledge base has not, however, been examined to see what other sorts of useful resource material it contains.

As information was assembled for this phase of aIFAS research, extra information was included for the individual cases. Those items included the report date, the industry/business group represented by the client, the component that failed, and the

material of construction. This additional information is studied to reveal what benefits might be derived from it that could improve the system's ability to aid a user.

1.5.3 Evaluation of Two Schemes for Case Representation

The scope of the failure analysis field has been characterized as broad. It should be understandable then, that even the least complex representation of a corresponding knowledge base would require a large numbers of data elements. Being able to manage a large set of failure modes and their related attributes is a very real concern. Two schemes for coping with that problem are considered. One approach uses independent, binary valued attributes. The other approach uses grouped sets of related attributes in which individual attributes can assume any value.

1.5.4 Comparison of Five Metrics for Case Selection

The size of the knowledge base for aIFAS has the potential of growing quite massive. Aside from the expansion produced when more and more example cases are included, as the system matures there will be an ever-increasing set of failure modes and corresponding attributes. The algorithm chosen to perform case selections that yield good solutions must be able to cope with those conditions. There is an imperative that the method not only discriminates well in the choices it makes, but is also computationally efficient.

Five metrics, representing the most common approaches used for case selection, are compared. The group includes the City Block (Manhattan) distance, Cosine Correlation similarity, Euclidean distance, Hamming distance, and a statistically based Knowledge Graph similarity. These algorithms are compared using the criteria of accuracy as well as efficiency (or speed).

To accomplish as fair as possible a comparison of the metrics, different ranking tools were considered. Drawing from literature, the conventional measure of the percentage of correct answers was used. Another measure, called a *Performance Score*, was devised to combine accuracy and computational time. The time measurement was made using a *Relative Time Unit* that scales time to a dimensionless, platform independent value.

1.5.5 Exploration of Knowledge Base Size and Incremental Learning

How many example cases is enough? This is a recurrent question when acquiring the case set for the knowledge base underlying an expert system. Obviously the set of included cases must be of sufficient size to answer as many questions posed the system as is reasonably possible. Yet the addition of each case imposes a computing load that can ultimately slow the response time to a crawl.

By applying the ranking tools used for comparison of the case selection metrics, some insight can be gained to answer the question of “How many example cases is enough?” A candidate metric can be applied to expanding sets of example cases and the resulting performance quantified. That is, the effects of incremental learning can be measured. The resulting information can be used in turn to identify an optimal set of example cases. If measures are made of not only the best match, but second, third, or more matches, then an estimate of an appropriate case density (number of stored cases representing each potential solution) can be determined.

2. KNOWLEDGE ACQUISITION

We learn through a process of knowledge acquisition, it is the means by which we capture knowledge and expertise [Parsaye and Chignell, 1985]. In a more contemporary sense, knowledge acquisition can be interpreted as the transfer and transformation of problem-solving expertise from some knowledge sources to a computer program [Lee-Post, 1994]. When applied to computerized systems, the process can include activities that involve direct elicitation from experts and/or machine induction from accumulated data [Johannsen and Alty, 1991].

The information being sought after comprises insight about causal relationships that have their basis in theory, statistical data, pure definitions, and personal judgement [Adlassnig, 1986]. That information can be classified into two categories – declarative facts or procedural strategies [Lee and O’Keefe, 1994]. The declarative facts can be explained as the descriptions of things and how they interact is called domain knowledge [Bergmann *et al.*, 1994] or the first-principle, basic facts of deep knowledge [Fink *et al.*, 1985]. The procedural strategies can be explained as the descriptions of the process for finding solutions [Bergmann *et al.*, 1994] or the rules-of-thumb derived from experience of shallow knowledge [Fink *et al.*, 1985]. Domain knowledge is usually much easier to obtain, especially in technical domains (such as failure analysis) [Bergmann *et al.*, 1994].

Gaining ownership of the knowledge is a major problem in building any knowledge-based system. That is the consensus of references on the subject, evoking such expressions as: the most difficult process [Scott, 1993]; one of the greatest difficulties [Polat and Guvenir, 1993]; the biggest bottleneck [Olson and Rueter, 1987]; or, the most time-consuming task [Lee-Post, 1994]. The magnitude of the issue can be

expressed by the statement [Johannsen and Alty, 1991] that few systems have progressed beyond the research or prototype phase mainly because of this inherent problem.

There are at least three immediate reasons that help explain the difficulties in knowledge acquisition. First is a lack of preparation. The knowledge domain may not have been researched in sufficient depth to satisfactorily support meaningful acquisition [Bench-Capon *et al.*, 1993]. Second is a lack of effort. There seems to be a natural tendency to underestimate the difficulty [Parsaye and Chignell, 1988] with attendant low levels of success because little effort is expended. Stated more emphatically, failing to appreciate the demands of the task is major reason for the failure of system development [Johannsen and Alty, 1991]. Third is a lack of communication. Evidence suggests that experts organize concepts differently than those not familiar with the knowledge domain [Olson and Rueter, 1987]. That fact amplifies the effects of inherent cognitive biases, subjectivity, and the often unstructured or ill-formulated nature of human knowledge to exacerbate useful dialogue [Lee-Post, 1994]. Furthermore, there may be national, cultural, religious, social status, or educational factors that separate the domain expert from the knowledge engineer [Adlassnig, 1986]. Simply put, if the expert and the knowledge engineer cannot effectively communicate, no knowledge is acquired.

In spite of the hindrances, knowledge acquisition remains a crucial element for any knowledge-based system [Nicholson, 1992]. The overall performance of a system literally depends upon the completeness of its knowledge base [Liao *et al.*, 2000a]. The quality, or correctness, of the acquired knowledge determines the ultimate success of the system [Parsaye and Chignell, 1988]. The vital importance of knowledge acquisition in system development is exemplified by MEKAS (Methodology for Knowledge Analysis);

a project dedicated entirely to giving the knowledge engineer a thorough understanding of knowledge domains [Bench-Capon *et al.*, 1993].

2.1 Acquisition Methods

There is no all-encompassing, unified theory of how to acquire knowledge, and probably never will be [Witten and MacDonald, 1988]. Devising a methodology is a complex problem. Consider the following comment [Johannsen and Alty, 1991] that goes on to explain that experts know what they know and what they do not know and can identify solution methods which will work or not work. Imagine coming up with a technique that can cope with that degree of variability in what information might be forthcoming? Adding to the conundrum, it is necessary to accommodate a variety of knowledge structures in the form of: simple lists, tables of data, flow diagrams, hierarchical relationships, nested categories, networks of associations, spatial maps, and physical models. Where each of those representations is suitable for a particular kind of reasoning or retrieval [Olson and Rueter, 1987].

There are some general methods available for consideration that serve to provide a basis for developing an approach tailored to meet the peculiarities of a chosen domain. They can be grouped into five areas: direct (interviews); observational (shadowing); indirect (repertory grids); machine learning (neural nets); and, document processing (data mining) [Liao *et al.*, 2000a]. Of those five areas, direct and indirect methods are best suited to failure analysis, while there might be some possible use of document processing to assess the content of archived reports.

Fitting a method to a knowledge domain is the beginning of the task. It is also necessary to motivate an expert to relinquish valuable time, assist with encoding the knowledge, and verify the completeness and consistency of the knowledge base [Low *et*

al., 1991]. While all those chores are being accomplished, it is also necessary to evoke meaningful responses that avoid exclusion of important facts from an individual who may be inept at articulating experiential knowledge [Olson and Rueter, 1987].

There are as many paradigms for knowledge acquisition as there are systems. This apparent richness and variety of approaches might give the impression of a field teeming with fruitful techniques [Witten and MacDonald, 1988]. In truth, there are many methods to use as models. The actuality is that each application has its own idiosyncrasies that must be dealt with aggressively and deliberately.

2.1.1 What Acquisition Tools Work

Of the direct, observational, indirect, machine learning, and document processing approaches, three offer the best fit with failure analysis. Experience has shown that the direct method of interviewing can be very productive, if for no other reason than it familiarizes the knowledge engineer with the knowledge domain. Other successes came from the indirect methods of completing charts of cause-and-effect relationships and the generation of repertory grids. Some manual document processing has been attempted (while producing the cause-and-effect charts) with encouraging success, sufficient success to warrant future investigation using digitized documents.

As alluded to, interviews are good for obtaining a sense of the knowledge domain [Johannsen and Alty, 1991]. The interview process, however, must be well directed to overcome a tendency by experts to describe interesting, complicated, or recent cases and omit the mundane and straightforward ones [Hart, 1985] (cases that are equally important for building a complete knowledge base). Interviews can also elicit distorted information that is an artifact from a perceived pressure of being interrogated versus questioned [Binaghi, 1990]. Other obstacles to be overcome during interviews are inarticulate

experts, forgotten facts, omission of information presumed to be common knowledge, and the ambiguities of technical language [Liao *et al.*, 2000a]. Notwithstanding those adversities, interviewing reveals the detailed structure of concepts better than other methods [Binaghi, 1990], yielding both general and specific information [Hart, 1985].

A very fruitful technique is the production of cause-and-effect charts. This indirect method can be formalized into a two-dimensional matrix array of causal relationships [Tansley and Hayball, 1993]. This sort of information display is particularly good for forcing consideration of combinations that would not otherwise have come to mind, and identifying attributes that would never co-occur or have no relationship. A bonus comes from the insight into the nature of the domain provided by requesting explanations of the non-occurrences of certain combinations.

Repertory grids have become somewhat of a standard form of indirect knowledge acquisition. They were introduced by George Kelly in 1955 [Parsaye and Chignell, 1988]. Based upon his theory of human thought, repertory grids capture how we make decisions by making selections from alternatives. This can be an extremely effectual technique for gathering the attributes of cases that experts consider in developing their solutions [Nicholson, 1992]. A repertory grid approach is well suited to accessing implicit knowledge (such as comprises much of the failure analysis domain). The technique can, however, be time consuming, produce difficult-to-interpret results, and become nearly unmanageable when considering more than ten elements [Tansley and Hayball, 1993]. As a parting consideration, it might be argued that in some situations experts may not even view problems in terms of the bipolar constructs of repertory grids.

Concept mapping [Osif, 1996] (or knowledge graphing) is another indirect approach that is apropos for the failure analysis domain. In this method, case attributes

would be arranged in hierarchical groupings from general to specific. Then linkages are constructed to establish relationships. Presentation in this fashion is an excellent way to expose misconceptions. Because alternatives do not necessarily have to be presented, elicitation may be less burdensome than with repertory grids. Concept mapping is somewhat of a hybridization of cause-and-effect matrices and repertory grids, combining the more positive aspects of each.

Document processing is a tantalizing, but inadequately explored technique of knowledge acquisition for failure analysis. Any serious undertaking could not use a find-whatever-there-is approach, but would need well-defined tasks and have procedures for dealing with missing or corrupt data [Holsheimer and Siebes, 1991]. (A major restriction has been the scarcity of failure analysis report documents that exist in electronic format.)

No one scheme of knowledge acquisition can sensibly be expected to suit all the circumstances arising in the failure analysis domain. It is plausible that part of the work would take a top-down (general-to-specific) approach starting with interviews and produce a somewhat hierarchical ordering of knowledge. Other efforts would use a bottom-up (specific-to-general) approach that might use statistical analysis to identify common/differentiating attributes of the cases being considered [Zahedi, 1993].

2.1.2 Building on Lessons Learned about Knowledge Acquisition

The problem of acquiring and assembling the required knowledge to support system development is very much different than is required for traditional computing domains (such as accounting, inventory, or job tracking) [Johannsen and Alty, 1991]. Selecting the essential set of attributes or striving to assimilate the lexicon of a technical field can easily become daunting tasks.

One good plan to contend with large-scale, complex applications is to develop the habit of maintaining a knowledge document [Prerau, 1987] for the project. That document provides a record of procedures and conventions followed in the knowledge acquisition process. Keeping such records is vital for facing the difficult task of obtaining an adequate training set of cases for a complicated real world problem; as well as, ensuring a sufficient number of counter examples [Liao *et al.*, 2000a].

It is equally important to remember that the process of knowledge acquisition cannot be rushed [Quinlan, 1986]. There are often demands for protracted interaction between domain experts and a knowledge engineer. Complex systems can take hundreds of person-days, or even years to finish the knowledge acquisition portion of their development [Johannsen and Alty, 1991].

2.1.2.1 Attribute Set Selection

The knowledge to be acquired, to a large degree, is contained in historical case studies. As the information is collected, it must be indexed in such a way that it can be reused easily. The indexing scheme must simultaneously satisfy two criteria: be general enough to apply to a significant number of cases and specific enough to discriminate significantly between cases [Allemang, 1994].

The indices that accomplish that important task of case classification will be referred to as attributes. These are the measures with which experts differentiate their selections in problem-solving situations. The characteristics of those attributes for failure analysis cases can be quite varied.

Attributes can convey predictive or descriptive information [Zhang, 1998]. The data values for attributes can be continuous, discrete, nominal, or Boolean [Dash and Liu, 1997]. The very same case attribute in different analyses, or different stages of the same

analysis, might have different relevance [Liao *et al.*, 1998]. It is also a good prospect that even a carefully selected set of attributes will contain irrelevant or redundant members [Dash and Liu, 1997]. All these factors emphasize that the process of selecting appropriate attributes is a complex matter.

Early efforts began with attempts to extract at least a starting set of attributes from selected case histories. That proved to be not only a lengthy, arduous task but also a not very effectual method. Subsequently, attributes were elicited from a group of experts in a brainstorming session. Through a consensus approach a good working set of attributes was developed. That set of attributes was utilized in a case-based failure analysis implementation [Zhang, 1998]. A further refinement was made using triad groupings in a fashion similar to repertory grid construction techniques to generate the attribute set for a rule-based failure analysis implementation [Zhan, 1998].

Neither of the attribute sets previously developed is considered optimum. They contain too many irrelevant and/or poorly described members. A new attribute set needs to be generated for aIFAS. It must be sufficient in content to allow discrimination of subtly different cases, but compact enough to support efficient performance. Using pictures depicting the failure mode being considered would function as an organizing tool to magnify relationships [Osif, 1996].

What to pick as attributes and how to select attributes are salient issues. Another, and much more elusive question, is the matter of deciding how many attributes is enough. It is unrealistic to attempt selection of a complete attribute set for a knowledge domain as complex as failure analysis. Rather, the system needs a compact set of attributes that is expansible to address new failure modes or variations of old ones.

A resolution to the predicament comes by considering Pareto's Rule. Vilfredo Pareto (1848-1923) was an Italian economist who observed that a relatively few people held the majority of the wealth. That is, roughly 80% of the assets were controlled by 20% of the populace [Schumpeter, 1952]. In the early 1950's, Dr. Joseph Juran expanded the notion to the stature of a universal principle. Pareto analysis has now become a widely accepted procedure of information study that separates the important few items from the many trivial items [Leong, 1996].

The 80-20 Pareto's Rule renders selection of an attribute set using the knowledge graphing approach a manageable prospect. The viability of this method when applied to failure analysis and related fields is confirmed in a fault diagnosis system for avionics components [Tan *et al.*, 1990] and the previously mentioned rule-based implementation of a failure analysis system [Zhan, 1998]. The situation is now distilled to one of selecting a starting subset of all failure analysis attributes that does not decrease accuracy and retains class distinctions for the complete domain. The indeterminate nature of the task is perhaps best explained by the observations that "the appropriate size of the attribute set is generally unknown in real-world problems" [Dash and Liu, 1997].

2.1.2.2 Standardized Terminology

Failure analysis is fraught with idiosyncratic terms related to particular industries or to the quirks of experts. With failure analysis terminology, some words can have three or four different alternative terms to describe the same thing [Graham-Jones and Mellor, 1996]. It should then be evident that speaking and understanding the language of the expert is crucial to successful knowledge acquisition [Parsaye and Chignell, 1988]. The task of knowledge elicitation can be utterly frustrating if there is faulty comprehension of either or both the knowledge domain and the domain expert.

The comprehension/communication barrier exists in all aspects of information gathering for failure analysis. Textual knowledge is a highly organized and systematic means of communication, yet it is inherently vague and ambiguous due to the variability of meanings [Ahmad *et al.*, 1991]. The process of interviewing is a common and useful tool for learning the jargon of an expert [Lee-Post, 1994], but that undertaking is hampered when the dialogue is technical [Plant, 1994]. The results of interviews can be ill-defined words or the unwitting documentation of synonymous terms [Hart, 1985]. Experts themselves are inconsistent. Two experts might use different terms to describe the same attribute or the same term for different attributes [Liao *et al.*, 2000a]. Each of these acquisition obstacles is related to the lack of standardized terminology.

The utilization of failure analysis domain knowledge fares no better in escaping the comprehension/communication confusion. In rule-based implementations, there is a certainty for ambiguous, poorly defined terms [Rissland and Skalak, 1991]. Imprecision in the semantics of abstract attributes used as indices for case-based implementations contribute to solution uncertainty [Dutta and Bonissone, 1993]. Essentially, the effects of ill-defined terms can insinuate themselves into the process of system development by confusing non-specialists in the knowledge domain [Hoppe, 1993].

Needless to say, failure analysis is not the only knowledge domain affected. This comprehension/communication roadblock is as pervasive as it is owing to the nature of how knowledge is structured. The knowledge of a domain is principally encoded in the language of that domain. An understanding of the language's terminology is crucial to an optimal application of the encoded knowledge [Ahmad *et al.*, 1991]. Crossing the boundaries between knowledge domains is generally difficult. A likely reason has been revealed by empirical studies that report considerable within-domain consistency of

terminology, but relatively low between-domain consistency. Bridging the chasm that exists between knowledge domains can be akin to learning an obscure foreign language.

The process of eliciting knowledge domain terminology has been formalized in software tools [Ahmad *et al.*, 1991]. Among those available are: KNACK by Klinker in 1988 that uses synonym rules to resolve terminology related conflicts; KEATS by Motta in 1990 that extracts concepts, statements, definitions, or relations; or, KRITON by Diederich in 1988 that exercises a fifteen step methodology for content analysis. Then there are also operational banks of terminology information such as EURODICAUTOM, LEXIS, and TEAM that can be accessed during the process of system development.

The most direct approach to standardize terminology starts with the creation of a terminology glossary [Ahmad *et al.*, 1991], a data dictionary [Hart, 1985], a concept database [Zahedi, 1993], or whatever name one might coin in reference to a lexicon of the knowledge domain vernacular. The next step is to incorporate unique descriptive names and explanatory clauses using standard domain jargon within the implemented system [Prerau, 1987]. Even the exemplar cases from which the system is developed can be accompanied by brief descriptions rendered in the language of the domain [Graham-Jones and Mellor, 1996]. As a final complement, graphics can be used in conjunction with text to describe terms and minimize identification errors [Liao *et al.*, 2000a].

The overriding goal of each part of the solution presented is to improve user interaction through the use of precise and consistent symbolic representations of terms [Liao *et al.*, 1998]. This framework for improving comprehension and communication is innovative in nature. It appears that in knowledge engineering, although terminology is often cited as an initial stage of knowledge acquisition, a systematic approach for achieving that goal is rarely adopted [Ahmad *et al.*, 1991].

2.2 Information Sources

A mistake commonly made that can adversely affect the ultimate performance of a system is to collect knowledge from the first available or most convenient source. In most problem domains, relevant information is scattered through multiple sources that may require aggressive search efforts for their discovery. Knowledge in many domains is available only to experts and may never have been written down in structured form [Parsaye and Chignell, 1988]. Failure analysis can certainly be characterized as one of those sorts of knowledge domains. The knowledge pertinent to successful failure investigation is poorly documented, as well as sparsely scattered throughout publications, historical case records, and the personal experience of domain experts [Zhan, 1998].

Being hard to find is just one issue; data quality is yet another. Inconsistent and incomplete data can lead to serious difficulties in system development and performance [Bort, 1996]. Any data that is elicited must be reliable and valid – reliable in the sense that a similar acquisition would yield similar results; and valid in that the data obtained are accurate [Tansley and Hayball, 1993].

Creativity and the benefit of experience are necessary for uncovering appropriate knowledge sources for failure analysis. This is very understandable since for the most part, the knowledge bases for systems such as is being presented in this work to perform failure analyses are by and large hand-crafted [Ahmad *et al.*, 1991].

2.2.1 What Information Sources Are Available

Failure analysis is a broad field, embracing many technical disciplines. Because of the breadth of the domain, one might be led to believe that such information sources are extensive and readily available. In some areas, sadly, the contrary is the case.

Interest here lies primarily with the amassing of experiential details. The primary sources for that sort of information are the individual recall of human domain experts, the scant number of volumes published in the field of failure analysis, and the archived records of actual failure analysis cases. The goal is to produce an efficient diagnostic system derived in large part from the historical content of the domain [Tan *et al.*, 1990].

2.2.1.1 Human Domain Experts

The domain expert is the essential ingredient for success in the knowledge-intensive and experience-based arena of failure analysis. Seasoned practitioners, through considerable experience, develop rules-of-thumb to swiftly guide them through an investigation and on to a solution. They possess capabilities quite often well beyond those of all but the most adept engineers and technicians [Doherty *et al.*, 1994].

There is a set of essential characteristics for an expert [Parsaye and Chignell, 1988]. Experts possess specialized knowledge. Experts know how to use that knowledge effectively. Experts can recognize the boundaries of their knowledge. Experts do not attempt to solve problems outside their expertise. Experts provide timely solutions.

Knowing what sort of resource to look for is one thing, finding a useful one is quite another. Assessing the competence of experts like determining their body weight is a fuzzy concept [Fourali, 1994]. To begin, because of the scope of issues in a complex domain it is difficult for an individual to be competent in all of the sub-domains. Further problems can arise from the degree of subjectivity evoked by experts as they regard a real world system [Druzdzal, 1997]. Then an expert's knowledge may simply be incomplete, inconsistent, or even erroneous [Suwa *et al.*, 1982].

Assuming that a suitable, competent expert can be found, there are still causes for concern that must be dealt with. One problem arises from the very experience that

creates an expert; long practice makes them less rigorous. As the need to perform intermediate steps in an investigation diminishes, the expert loses an awareness of the significance of those steps. With that erosion of methodology comes an inability to eloquently offer an account of how a decision is made [Parsaye and Chignell, 1988]. Then there is the more simplistic situation that the expert may be knowledgeable but just happens to have poor communication skills [Nicholson, 1992].

2.2.1.2 Publications

There are all too few sources of published material on the subject of failure analysis. Examples and explanations of failures are seldom published with the instances typically being brief, lacking in useful detail, and of limited variety [French, 1983].

The publication of significant numbers of references on failure analysis did not begin until the mid 1970's [Petzow, 1979]. Twenty years later, a discussion on the availability of literature on high temperature corrosion failures produced this comment, "...there has not been a single book covering up-to-date data..." [Lai, 1990]. The state of affairs still remains one in which there are few contemporary references. The available information still tends to be limited, dealing with basic failure modes [Esaklul, 1992].

The Author is aware of eight major references that have become somewhat the recognized authoritative source for failure analysis information. The principle sort of information offered by each is summarized in the following:

- [Barer and Peters, 1970] – explains with reasonable clarity the basic failure mechanisms and the tools used to perform failure investigations
- [Brooks and Choudhury, 1993] – the most contemporary of the references, including the utilization of modern technology

- [Esaklul, 1992] – two volumes of case studies from around the world presented in a well-organized, understandable format
- [Naumann, 1983] – provides examples of unexpected failure modes with excellent documentary photographs of unusual failures
- [Petzow, 1979] – twenty-five modes of failure are presented with good documentary illustrations, but the narrative is sketchy
- [Powell and Mahmoud, 1986] – examples of the most common failure modes and methodologies for studying failures
- [Ryder, 1975] – a comprehensive explanation of the failure investigation process combined with descriptions of the common modes of fracture experienced by metals
- [Wulpi, 1985] – offers an expansive list of questions to ask in the course of a failure investigation

2.2.1.3 Archived Reports

Historical files would be a prime source of information for developing a failure analysis system [Tan *et al.*, 1990]. The Author can access report files spanning 1989 to 1999, representing the work product of six failure analysts. Some three thousand reports are stored in electronic form; many with digitized graphics.

2.2.2 Building on Lessons Learned about Information Sources

It is crucial that the quality of the information source is assessed before any significant resources are invested using any of the acquired data to develop a system [Hinkle and Toomey, 1995]. Insuring the quality, or useability, of the knowledge as it is accumulated can be a daunting task unless it is obtained in a methodical fashion.

Earlier work indicated three methodologies that should improve the quality of the information. First, when asked to enumerate attributes, experts often supply too many of them [Hart, 1985]. Insisting on the formulation of a hierarchical data structure should overcome much of that problem. Second, attempting to deal concurrently with the full breadth of a domain as complex as failure analysis is at best unwieldy. Attempting to do so opens the door for contradictory and inconsistent information [Romaniuk and Hall, 1992]. By creating a system structure that fosters incremental growth, adequate time should be available to evaluate inputs and overcome the shear size problem. Third, is an issue concerning the case information – What data is actually suitable for use? The context of an historical report is often vital in interpreting the relevance of attributes or deciding between unknown and unimportant values [Zhan, 1998]. By using only complete case history reports (as opposed to adaptations or synthesized data), much of those concerns should be minimized if not alleviated.

2.2.2.1 Hierarchical Data Structure

A system using a hierarchical representation of its information is not only reasonable, but also resource efficient [Fink *et al.*, 1985]. In such a system, domain concepts and their respective relationships become easier to assess when they are decomposed into sub-components rather than when viewed as a whole [Tansley and Hayball, 1993]. In turn, the easier evaluation of knowledge possible in an hierarchical data structure assists with improved discrimination of classifications within the knowledge base [Allemang, 1994].

Practically speaking, this organization of the knowledge base works to reduce the number of search paths necessary and makes for answers that are more reliable. For example, if the component being considered is a lifting hook it is unlikely that it would

fail due to internal overpressurization [Graham-Jones and Mellor, 1995]. In a dynamic system, with a growing knowledge base, it becomes increasingly important that means are provided to search out information quickly with a combination of identifying features and in some hierarchy of critical facts [Graham-Jones and Mellor, 1996].

Functionally a hierarchical data structure for a knowledge base has merit, it also offers other advantages. A system design that partitions and organizes expert knowledge significantly increases speed [Perez and Koh, 1993]. As the knowledge base grows and matures, multistage retrieval is an efficient means of dealing with large numbers of example cases [Stottler, 1994]. And a final point, earlier work on this project identified an improvement to cope with the large number of unknown attribute values would be to formulate an hierarchical data structure for the failure analysis system.

2.2.2.2 Build Incrementally

Implementing a knowledge based system is understandably a difficult process, requiring special skills and often taking many person-years of effort [Watson and Marir, 1994]. Using human experts demands frequent communication over protracted intervals [Perez and Koh, 1993] to acquire all of the necessary knowledge. The resulting mass of information can be overwhelming.

Previous efforts suffered from that sort of an information glut. The system development process was bogged down trying to manage too many data attributes. Worse yet, the data was sometimes imperfect, since experts do not always provide the knowledge needed in complete and consistent chunks [Polat and Guvenir, 1993].

Another approach for getting data to support system development came from the consideration that in human learning, knowledge acquisition is typically an iterative process [Low *et al.*, 1991]. It would seem sensible, therefore, to have a system with an

inherent incremental learning capability that would serve to expand its knowledge base and improve overall performance [Liao *et al.*, 2000a].

2.2.2.3 Completed Reports Offer the Best Data

Different sources of historical information were explored; they included – direct evaluation of reports, narrative interviews with domain experts on their techniques for documenting case histories, the combination of raw attributes into hypothetical cases, and the extension of existing case histories into synthetic derivative cases. The effectiveness of each source for providing useful information varied.

Casual examination of case history reports often reflected the inexperience or subjectivity of the reviewer. The diversity of typical failure investigations was confusing to someone not familiar with the practice. Synthetic case histories were not reliable. Although utilitarian, they introduced biases regarding the mix of representative cases [O'Keefe and O'Leary, 1993]. The overriding common feature was the realization that identifying relevant data is a complex problem [Shortliffe and Buchanan, 1975].

All that withstanding, there is a great deal of important information and failure experience locked up in the engineering files of companies [Thielsch, 1977] that must in some way be captured. The method employed must adequately cope with the reality that reports written by experts tend to have a high degree of bias and reflective thought [Johannsen and Alty, 1991]. A sensible proposal would incorporate an objective analysis of actual case histories in the framework of a hierarchical data system, combined with a rigorously enforced use of standardized terminology.

2.3 The Knowledge Engineer

The process of knowledge acquisition can be logically divided into two tasks: the discovery phase and the revision phase. The process flow diagram in Figure 2.1 shows

the fundamental steps in that process [Ahmad *et al.*, 1991]. The knowledge engineer is the individual that is expected to ensure that all those steps are successfully completed.

The knowledge engineer is responsible for at least directing, if not actually performing, all of the activities central to building an expert system. The requirements for skills and expertise go well beyond those needed solely for knowledge acquisition and embrace the gamut of system implementation tasks.

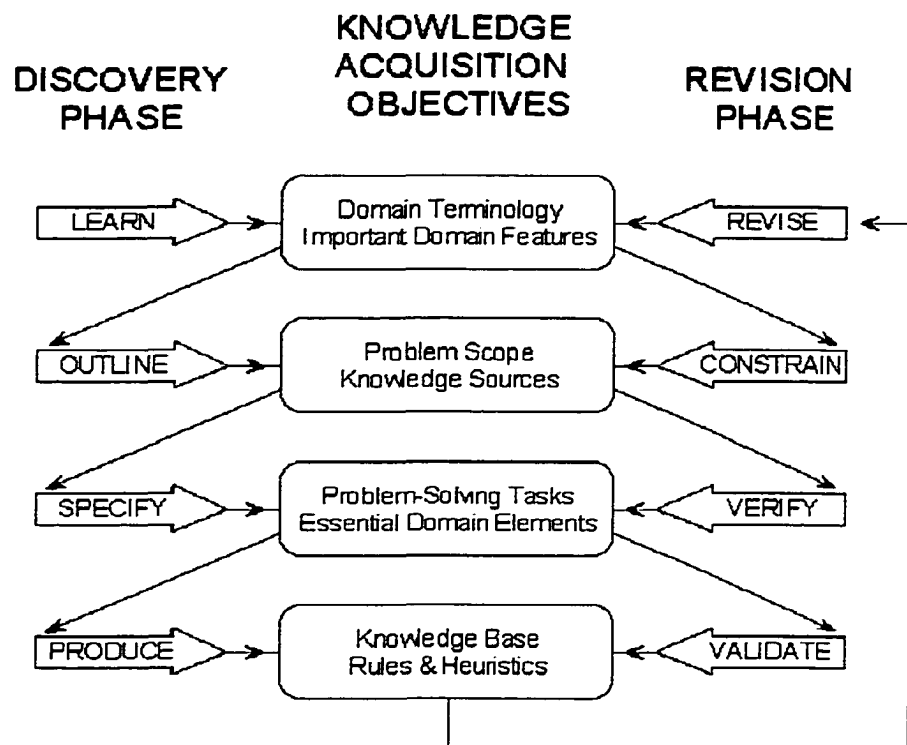


Figure 2.1: The Knowledge Acquisition Process.

The job demands a knowledge engineer expert. Certainly, there are automated tools to assist with the process, but they are merely tools and cannot supplant the need for a dedicated individual [Gallant, 1988]. It is crucially important that the judgement of a human knowledge engineer expert is intimately involved in determining the suitability of

knowledge elicitation methods, the structure of the knowledge base, and the methodology for manipulating that knowledge [Olson and Rueter, 1987].

2.3.1 What the Knowledge Engineer Must Do

What are the many specific things that the knowledge engineer must do to accomplish the goal of producing a viable expert system? The job title requires that an individual be able to wear many hats successfully. In the course of development, it may be necessary for the knowledge engineer to function as a system designer, translator, investigator, educator, mediator, interpreter, evaluator, innovator, observer, programmer, manager, or in some other unimagined skill classification that might arise.

To start the process, it is first necessary to conceive of a method to map human expertise into the parameters required by some acquisition scheme [Johannsen and Alty, 1991]. That is not a particularly easy task. Experts most often make decisions based on an abstract and intellectual process than even they themselves have difficulty understanding and verbalizing [Zahedi, 1993]. An initial task, then, of the knowledge engineer is to translate that abstract process into the knowledge base. Not much progress could be expected unless the knowledge engineer is at least moderately conversant in the vernacular of the knowledge domain. Proficient communication skills are needed to elicit the desired information and decipher the responses [Plant, 1994].

Determining how to get the information is only the start. Deciding what to get can be equally daunting. A common pitfall in knowledge acquisition is the collection of irrelevant knowledge [Parsaye and Chignell, 1988]. The knowledge engineer must be able to filter out the crucial tidbits from the greater mass of inconsequential data.

Assume that some scheme has evolved to begin acquiring what, for all practical considerations, seems to be reasonably good quality knowledge. A system that can

manipulate that information to solve problems must then be designed. The cognitive tasks that the knowledge engineer might make use of during system design involve a gamut of activities ranging from learning to problem-solving [Ahmad *et al.*, 1991].

To recap the requirements, a knowledge engineer is responsible for identifying a factual source of problem-solving knowledge; generate that knowledge into a useable form; then, validate such measures as accuracy, utility, benefit, speed, flexibility, and extensibility [Lee-Post, 1994]. This can most effectively be achieved by an iterative process of cycles that collect data, carefully scrutinize the information, and then thoughtfully make incremental refinements of the knowledge base [Andert, 1992]. To ensure the task of system development is completed requires above all that the knowledge engineer is a skillful resource manager.

2.3.2 Building on Lessons Learned about Knowledge Engineers

The process of knowledge elicitation is sometimes more tedious than it is laborious. Interaction with domain experts can be drawn out over extended periods, creating a perception of marginal progress. The inherently slow process of knowledge elicitation can become susceptible to errors [Morales and Garcia, 1990]. The knowledge engineer must be prepared to cope with that actuality through persistence and a dogged refusal to compromise quality for speed.

A very real stumbling block can be an assumption that the acquired information is consistent and complete. More often than not, domain experts function using a set of premises to which there are frequently numerous exceptions. Often times the revelation that such exceptions exist occurs long after the rudimentary knowledge is obtained [Johannsen and Alty, 1991]. The knowledge engineer must at all times remain flexible, working toward an adaptable, easily modified system to deal with changing conditions.

The knowledge engineer most of all needs to be a diplomat. Common sense appeals to the realization that more than one domain expert might be called upon to provide the knowledge base of a complex expert system. Conflict resolution of divergent views and opinions in a multi-expert system is an important task [Zahedi, 1993]. An effective knowledge engineer must be able to move forward even amidst dissension.

2.3.2.1 Great Diversity of Required Disciplines

How well any of those issues can be dealt with by the knowledge engineer is exacerbated by the complexity of the knowledge domain. Simply put, the field of failure analysis is vast [Wulpi, 1985]. The physical part of a failure investigation can entail an involved sequence of activities that embraces several disciplines [Ryder, 1975]. This very real diversity of domain sub-disciplines presents a factor that can significantly influence effective communication in the process of knowledge acquisition. The truly competent knowledge engineer would be expected to have communication skills with intellectual mastery adequate to disambiguate the technical jargon and combine the perspectives of those varied disciplines [Ahmad *et al.*, 1991].

2.3.2.2 Expansive Scope Fosters Misunderstanding

The tangled interdependency of skills, techniques, and background are very much a part of the way problems addressed by failure analyses are diagnosed [Barer and Peters, 1970]. The puzzle presents the investigator so many facets to be concerned with that the importance of knowledge elements can be misunderstood or misconstrued. The knowledge engineer must be attuned to the pressing need for exploring the full range of the domain expert's knowledge, to include divulging as many exceptions and unusual cases as possible [Parsaye and Chignell, 1988]. Only with that point of view would it be

reasonable to expect that the often unaddressed issues of conflicting experts and seemingly contradictory facts could be dealt with [Nazareth, 1993].

2.3.3 The Same Person as Expert and Knowledge Engineer

Developing an intelligent knowledge system for failure analysis requires capturing the expertise of domain experts. That expertise is primarily a skill of recognition, of seeing old patterns in the new problem [Olson and Rueter, 1987]. The skill attributed to a true expert is imbued by unique qualities and talents—

“An expert is a person who, because of training and experience, is able to do things the rest of us cannot; experts are not only proficient but also smooth and efficient in the actions they take. Experts know a great many things and have tricks and caveats for applying what they know to problems and tasks; they are also good at plowing through irrelevant information in order to get at basic issues, and they are good at recognizing problems they face as instances of types with which they are familiar. Underlying the behavior of experts is the body of operative knowledge we have termed expertise.” [Parsaye and Chignell, 1988]

Having access to the skill of an expert is important; however, human experts can present a problem — the human expert can effortlessly provide a sound answer, but have trouble explaining how that answer was determined [Graham-Jones and Mellor, 1996]. The knowledge engineer must bridge the gap to make knowledge both accessible and useful. Sadly, a major problem associated with knowledge acquisition is a lack of skilled knowledge engineers [Lee-Post, 1994]. The paucity of capable knowledge engineers increases proportional to the complexity of the knowledge domain.

An inventive approach would have the domain expert function in a dual role, also serving as knowledge engineer [Zahedi, 1993]. In that way a true-expert, one who understands the entire process, would be available during all phases of the system's design [Vargas and Raj, 1993]. That is the very solution proposed for the development and implementation of aIFAS.

2.3.3.1 Qualifications

In the realm of personal health, which would you prefer, a doctor fresh out of medical school with a state-of-the-art education or a veteran doctor with 20 years of practice? Most people would vote for experience over theory. Hands-on experience and decision making may be the most natural form of learning [Churbuck, 1992]

The logic for selecting domain experts and knowledge engineers should be no different than that expressed by the personal health illustration. Certainly ignorance or limited knowledge on the part of the expert could contribute to error [Nazareth, 1989]. Similarly, an effective knowledge engineer should be as familiar as possible with the domain [Tansley and Hayball, 1993]. The selection process to fill those roles should point to someone that has insight derived from experience and can communicate knowledge [Prerau, 1987].

The Author meets the requirements for assuming a dual role as domain expert and knowledge engineer. As regards domain knowledge, the Author is a registered engineer and has worked more than twenty years as a failure analyst. As regards knowledge engineering, the Author has served in varying capacities for knowledge elicitation over the duration of this project and has supported or co-authored published articles on the topic. Interestingly, only in the realm of industrial applications could a domain expert successfully assume the role of knowledge engineer [Johannsen and Alty, 1991].

2.3.3.2 Rationale for Assuming Both Roles

There are instances of both indirect and direct support found in a survey of literature for the domain-expert/knowledge-engineer hybrid. Some of the more salient aspects are presented in the following:

- Systems can only be developed if suitably experienced and willing experts can be found to provide the knowledge [Doherty *et al.*, 1994].
- Access to a human expert in the problem domain is essential [Zhan, 1998].
- Extensive experience and a familiarity with the work done by other experts in related fields are of utmost importance [Thielsch, 1977].
- It is a common observation that experts often have great difficulty in explaining the procedures which they use to arrive at decisions to knowledge engineers [Johannsen and Alty, 1991]
- Experts understand the relation of fundamentals and heuristics [Parsaye and Chignell, 1988]; that is, the expert knows how the knowledge base works.
- Accurate and complete knowledge may not be adequately transferred when an expert who does not understand computers works with a knowledge engineer who is unfamiliar with the problem domain [Suwa *et al.*, 1982].
- Communication problems in knowledge acquisition can be alleviated by the expert actually building the system himself [Hart, 1985].
- Having a single expert as a focal point for system development is often used to increase consistency and ensure completeness [Fink *et al.*, 1985].
- A strong interaction between the knowledge engineer and the domain expert is needed to maintain a system, even after deployment [Vargas and Raj, 1993].

When the situation presents itself, such as in the aIFAS project, the prudent option is to let the domain expert and knowledge engineer be the same person. In that way, the greatest economy of knowledge utilization can be achieved. That economy is derived in great part through the removal of communication barriers.

3. SYSTEM IMPLEMENTATION

Once a mass of knowledge has been acquired, or the acquisition process has at least begun, there needs to be a systematic construct for using that data. The form of that system is dependent somewhat upon the nature of the information being managed. How that is accomplished when dealing with the field of failure analysis must consider two separate and distinct concerns.

Reasoning within such disciplines as engineering, science, management, or medicine is usually based on formal methodology employing a logical treatment of causal relationships. Most technically trained individuals are reluctant to rely on heuristic approaches and ad-hoc reasoning schemes whenever the cost of making an error is high. As an extreme example, few people would choose to fly in airplanes put together based upon the designer's experiences over airplanes built using the laws of aerodynamics [Druzdzal, 1997]. A successful system implementation must, therefore, derive its answers and present a solution with sufficient technical authority to be palatable to the individuals who ultimately use those results.

On the other hand, the procedure by which experts in the domain of failure analysis use knowledge to infer solutions from observed conditions, test results, and history is a difficult to understand process. To some extent this is a subconscious activity, which is why it is probably often called an art [Adlassnig, 1986]. This means that the technologically authoritative system must also be able to cope with uncertain and imprecise data in an intangible manner.

Implementing an expert system to perform failure analysis is not a simple task. The system, however it is ultimately configured, must be able to provide expert problem-solving performance by exploiting a knowledge base and reasoning mechanism specific

to the chosen competence domain [Guida and Mauri, 1993]. That system's purpose is to meld the qualitative and non-quantifiable aspects of the problem-solving process with the fundamental or theoretical basis of the underlying knowledge domain [Zahedi, 1993].

Commercial expert systems of the sort described have been conceived, developed, and successfully implemented [Walker and Miller, 1989]. Some of the more mature systems are:

- EQUIPSELECT – commissioned by the American Welding Institute to assist in choosing welding equipment appropriate to a given task
- GARMAN – commissioned by ASEA Robotics to configure spot-welding robot systems
- WELDEX – commissioned by Battelle Columbus Laboratories to identify welding faults by examining features present in weld radiographs
- AMES – commissioned by the Nuclear Regulatory Commission to aid in the management planning of low probability, high consequence severe accidents
- XCANDIDATE – commissioned by Universal Technology Corporation to evaluate artificial intelligence applications in manufacturing

More recently, technical expert system have been developed that address such issues as medical diagnosis, DNA experimentation, weather forecasting, and process system control [Tzafestas *et al.*, 1994]. On a similar tract, a major area for contemporary artificial intelligence application interest is in the field of equipment troubleshooting [Low *et al.*, 1991]. To date, however, no expert system has been completely developed that comprehensively addresses the field of industrial failure analysis.

A comprehensive expert system must meet other criteria. The knowledge base representation must satisfy the requirement of being understandable, easily examined,

and readily updateable [Backer *et al.*, 1988]. Through some means the system must also be able to produce an improvement in its performance; that is, the comprehensive expert system must be able to learn [Jackson, 1988].

3.1 General Methods for Knowledge Representation

The sophistication of a knowledge base is a function of many factors including the actual task to be accomplished, the difficulty of that task to be accomplished, and the complexity of the knowledge domain. An example of a simple and very common use for a knowledge-based system is as a diagnostic tool that identifies faults given observed symptoms [Guan and Graham, 1994]. The fault-tree diagnostic approach for problem solving is uncomplicated but can be used to address moderately complicated situations.

Another specialized system developed for use with hydraulic devices is MIDAS (Model-based Intelligent system for Diagnosis, Animation and Simulation). A combination of cause-and-effect relationships, numerical models, and graphic presentations are put to use to present answers. This system has been extended beyond simple fault diagnosis and can furnish explanations of its results [Doherty, 1992].

Ultimately these type systems require a substantial investment in computational resources. Other, more flexible and economical methodologies need to be considered for applications involving comprehensive knowledge domains. The more common approaches are case-based reasoning, rule-based reasoning, and neural networks.

Simply put, case-based reasoning is a problem-solving paradigm using past experience to guide the solution process [Xu, 1994]. This approach is attractive since it allows the investigator to take advantage of what has happened before [Barletta, 1991]. One of the aspired benefits of case-based reasoning is to reduce the need to acquire and explicitly represent general knowledge of the problem domain. By not needing rules,

case-based reasoning can overcome the oft-spoken-of knowledge acquisition bottleneck [Bergmann *et al.*, 1994].

Rule-based reasoning is a mature technology and is very effective in well-understood domains [Watson and Marir, 1994]. The approach is intuitively attractive, as symptoms can be linked to causes explicitly. The distinct symptom-cause linkages can be expressed without any deep knowledge of the system structure, function, or principles of operation [Becraft and Lee, 1993]. The implementations of rule-based systems can in fact offer compact representations of diverse knowledge domains [Jackson, 1988].

Neural networks, or connectionist systems, find associations between inputs and outputs. The system need not know how and why they are related [Low *et al.*, 1991]. These systems are very good at handling noisy, or error-filled, data [Barletta, 1991]. The neural network approach also offers the possibility of discovering non-obvious information that might be inherent in the data being considered [Hudson *et al.*, 1991].

Beyond selection of the basic framework upon which to build the system, other matters must be considered. How will the knowledge base information be indexed? Will information be dichotomous values or the range of possibilities offered with fuzzy logic?

3.2 The Case-Based Approach

In the 1970's a computer science professor at Yale, Roger Schank, outlined a theory of artificial intelligence [Churbuck, 1992]. The basic question posed was, "Does thinking really involve thinking?" The response took the position that thinking is founded not on a complex reasoning process, but on the use of information from old situations in dealing with new situations [Allemang, 1994]. That assertion and the research it stimulated are widely held to be the origin of case-based reasoning [Watson and Marir, 1994].

Case-based reasoning systems are multipurpose vehicles for the knowledge engineer. They can serve as problem-solvers, to find a suitable plan or select a course of action; or, they can be precedent-setters, to justify/explain a particular solution [Dutta and Bonissone, 1993]. The versatility of case-based reasoning systems can be extended to include such features as: they help to repeat successes; they can assist in side-stepping previous errors; they can be used to teach novices; and, they can make experienced people more effective [Klinger, 1994].

Case-based reasoning systems attract attention because of their simplicity. Knowledge elicitation is the task of gathering case histories; implementation involves identifying significant features to describe the cases; existing database techniques can manage large case libraries; and, the system can learn as it acquires new cases [Watson and Marir, 1994]. The approach used by case-based reasoning is particularly useful where formal sets of rules are difficult to obtain, but examples of correct solutions are readily available [Graham-Jones and Mellor, 1996].

The most common approaches for the selection process in case-based reasoning systems can be fitted into two categories. One type uses a distance-based scheme where case similarity is derived computationally from attributes constituting the case. The other type uses some form of indexing structure to establish case similarity with a hierarchical arrangement of the attributes constituting the case [Liao *et al.*, 1998]. A third method for choosing similar cases is dependent upon the induction of selection relationships from the collection of example cases [Barletta, 1991].

Case-based reasoning systems offer an intuitive attraction to users. People solve many problems by recalling their experiences. Only a novice would attempt to solve problems by applying rules or knowledge that has only recently been acquired. Since

case-based reasoning systems rarely operate without human intervention, and actually encourage collaboration in the decision process, they are readily welcomed as a problem-solving tool [Watson and Marir, 1994].

The first case-based reasoning system to be developed is credited to Janet Kolodner. It was called CYRUS and contained travel itinerary history for once United States Secretary of State Cyrus Vance. Subsequently numerous systems have been developed in academia, serving many disciplines [Watson and Marir, 1994]:

- ARCHIE helps architects with conceptual design
- CADET functions as an assistant to mechanical designers
- COACH generates new football plays by improving old plays
- JUDGE presents a model for criminal sentencing
- MEDIATOR works in the domain of dispute resolution
- PROTOS was developed in the domain of clinical audiology
- TOTLEC solves complex manufacturing planning problems

3.2.1 Strengths of the Case-Based Approach

Case-based reasoning approaches work well in knowledge domains that are not very well understood [Barletta, 1991]. This is undoubtedly the case because case-based reasoning does not require an explicit domain knowledge model [Liao *et al.*, 1998]. By using the knowledge embodied in past cases, the knowledge-acquisition bottleneck common to expert systems is at least widened [Stottler, 1994]. Furthermore, proven systems already exist to efficiently manage even large volumes of historical cases that may accumulate in the knowledge base [Liao *et al.*, 1998].

To be considered successful, an expert system must be capable of generating adequate explanations. Case-based reasoning systems satisfy this requirement by using

previously learned cases as role models from which to infer an explanation [Vargas and Raj, 1993]. In short, the very information necessary to explain responses and justify answers is contained within the system [Barletta, 1991].

Case-based reasoning systems are unique because they can solve problems for which there are noisy or incomplete descriptions, provided there is a sufficiently rich library of cases. They can even be structured to accept new cases without major difficulties or the immediate assistance of the knowledge engineer [Vargas and Raj, 1993]. A serendipitous realization is that as a case-based reasoning system acquires information it begins to reflect the personality of its user [Watson and Raj, 1994].

Case-based reasoning systems also offer a built in degree of flexibility. A considerable amount of modification can be accomplished by altering the indexing scheme for case selection without the necessity of completely rebuilding the system [Barletta, 1991]. Finally, case-based reasoning systems can be developed much quicker, as much as four to eight times faster than other expert system implementation approaches [Watson and Marir, 1994].

3.2.2 Weaknesses of the Case-Based Approach

An issue worthy of some concern is that although case-based reasoning may alleviate the knowledge acquisition problem, it may do so by merely replacing that problem with an indexing problem [Allemang, 1994]. The example cases may be easy to obtain for incorporation within the case base of a system. However, creating a method for effectively and efficiently recalling them for problem-solving activities can be another issue all together. Particularly challenging issues are involved when trying to retrieve previous cases with vague and imprecise descriptions of either the example case and/or the new test case [Xu, 1994].

The case-based reasoning approach is straightforward and uncomplicated. There is a price to be paid for that simplicity. Searches of increasingly bigger cases libraries can become computationally expensive [Reategui *et al.*, 1997].

3.2.3 Lessons Learned about the Case-Based Approach

Case-based reasoning worked quite effectively in an implementation that applied a genetic algorithm search scheme to failure analysis. That effort did not use any of the commercial case-based reasoning development tools. The reasons cited were their difficulty in communicating with other kinds of databases and applications; inability to easily modify the case selection scheme; and, frankly put they were too expensive when compared with what they offered [Zhang, 1998].

3.3 The Rule-Based Approach

The DENDRAL Project, intended to deduce the likely molecular of organic compounds, was initiated in 1965 by Edward Feigenbaum. It is recognized by many as the genesis of expert systems, or rule-based reasoning with computers. The project predated by some seven years the much cited MYCIN program, developed to diagnose infectious blood diseases [Hughes *et al.*, 1999].

Rule-based reasoning expert systems have maintained a level of popularity for quite some time. Enthusiasm for them has been generated in part by a perceived relative advantage for computerization of specialized domains with perishable expertise [Nazareth, 1993]. Such systems typically are designed to move directly from raw data and facts to a correct answer [Lee and O'Keefe, 1994].

In general, induction techniques are viable if the problem is sufficiently simple and well defined [Nicholson, 1992]. To ensure those requirements are met, it is important to understand that rule-based systems work best in narrow application domains.

The rule-based reasoning approach is most useful in those applications with plentiful documented examples and where cause-and-effect type rules form a major part of the knowledge representation [Johannsen and Alty, 1991].

A sampling of contemporary rule-based reasoning system implementations includes the following assortment:

- Service Bay Diagnostic Systems for making the expertise of top vehicle diagnosticians available to general mechanics [Pepper, 1985]
- Central Office Maintenance Printout Analysis (COMPASS) for maintaining telephone switching systems [Prerau, 1987]
- Expert Maintenance System (EXMAS) for manufacturing fault diagnosis and failure identification in flexible manufacturing systems [Milacic *et al.*, 1988]
- DISPLAN a 1000 rule expert system used for planning geriatric patient hospital discharge [Preece, 1990]
- Final Approach Spacing Tool (FAST) a real-world expert system implemented to assist air traffic controllers [Isaacson *et al.*, 1997]

3.3.1 Strengths of the Rule-Based Approach

Rule-based reasoning expert systems present an explicit representation of the knowledge they contain. Because the governing relationships are present, they can generate explanations of their results. The approach also lends itself to the incorporation of rules, laws, or principles drawn from the fundamental basis of the knowledge domain [Becraft and Lee, 1993]. These characteristics are generally all true so long as the method is applied to small or medium size problems [Padalkar *et al.*, 1991].

Once the rule set has been generated, a rule-based reasoning system program is fast to execute. The resulting answers can be very accurate [Quinlan, 1986]. In fact,

given the appropriate set of rules, rule-based reasoning systems can yield 90 to 95 percent agreement with test questions [Fink, 1985].

3.3.2 Weaknesses of the Rule-Based Approach

In the industrial/commercial environment, outside of academia, people often make decisions without reference to fundamental principles and underlying cause-and-effect relationships [Watson and Marir, 1994]. The intuitive notions of experience take precedence over theory and rigorous proofs. Rule-based reasoning systems do not deal well with uncertainties such as the vagueness and ambiguity associated with that level of human thinking [Yuan and Shaw, 1995]. Moreover, these sorts of systems have difficulty recognizing the limits of their own knowledge and can, consequently, provide bad information [Fink *et al.*, 1985].

Stated very directly, rule-based reasoning systems have proven to be very difficult to create [Churbuck, 1992]. Major problems are often related to solutions that involve a combination of interrelated causes, rather than to a single cause. That yields a much too complicated rule structure which can take far too long to produce [Walton, 1991]. In some instances, just the process of accumulating the knowledge base from which a rule set can be produced can be costly [Stottler, 1994].

3.3.3 Lessons Learned about the Rule-Based Approach

The rule-based reasoning approach is a good, but high maintenance solution for providing assistance in the field of industrial failure analysis. Some very useful insights did come from the previous work applying the rule-based approach. A viable system must be able to deal with ill-formed new cases that have a large number of missing attributes. Hierarchical knowledge structure should be considered for the complex

domain of failure analysis. In addition, although complex and broad in scope, failure modes can be identified using simple relationships [Zhan, 1998].

3.4 Neural Networks

The idea of capturing knowledge using an artificial neural network (ANN) was proposed in the late 1950's [Scott, 1993]. Neural networks would be computer constructs that resemble human brain function. When presented with a problem, the neural network was intended to provide an answer, similar to a human expert.

Knowledge for a neural network is represented as sets of inputs and outputs, or causes and effects. The neural network data is assembled and trained based upon a burgeoning variety of computational algorithms. (Any detailed exploration of those stratagems is beyond the scope of this work.)

Regardless of the method, the goal of network training is twofold. Convergence should arrive at a point where solution error is at an acceptable level. Generalization should create a network that can successfully analyze new problems [Scott, 1993].

Neural networks have been applied to medical diagnosis as in MLP, a fuzzy multilayer system [Mitra, 1994]. Or for trouble shooting avionics instruments as in the Inertial Navigation System Interactive Diagnostic Expert (INSIDE) system [Tan *et al.*, 1990]. One of the largest networks is CPSC for internal medicine that consists of 33 descriptive and 375 feature nodes to identify 14 diseases [Lin and Druzdzal, 1998].

3.4.1 Strengths of the Neural Network Approach

One benefit of neural networks once they have been trained is that they are extremely fast [Becraft and Lee, 1993]. Of perhaps even greater importance is that neural networks can also produce accurate results [Barletta, 1991]. Where responsiveness is crucial, such as with the urgency of medical diagnosis, speed and accuracy are vital.

Neural network systems also tend to be tolerant of errors [Gallant, 1988]. This can be a feature of considerable worth for a knowledge-based system dealing with a complex knowledge domain. Of parallel value, most problems suitable for this sort of implementation can be solved with a simple 3-layer network of nodes [Scott, 1993].

3.4.2 Weaknesses of the Neural Network Approach

In the purist sense, a neural network can be characterized as a computational black box that offers nothing beyond the bare answer. Left by itself, the neural network has great difficulty in explaining its results [Stottler, 1994]. Probably as a reaction to such a stark response, early neural network approaches failed to gain acceptance because they did not appear to arrive at decisions through any recognized strategy that the user could understand [Hudson *et al.*, 1991].

Computationally, neural networks require two to three orders of magnitude more data and machine processing time than other schemes to produce an accurate system [Barletta, 1991]. Neural networks can quickly generate answers once they are trained, but they demand considerable resources in their creation. Assuming a useful system can be produced, the resulting neural network cannot be easily updated to incorporate new information as it becomes available [Low *et al.*, 1991].

3.4.3 Lessons Learned about the Neural Network Approach

Neural networks do not lend themselves to applications dealing with comprehensive knowledge domains that embrace a variety of disciplines, such as industrial failure analysis. The expense of data collection and computational resources is prohibitive. The concept does offer potential utility as a supporting subsystem for specialized tasks, such as the alloy selector [Garcia, 1997] discussed previously.

3.5 Metrics for Case Comparison

Case-based reasoning is an appealing choice for application in the field of failure analysis. Given a listing of attributes for a candidate case, a retrieval algorithm must find the most similar case in a set of stored examples. Among the well known methods for case retrieval are the nearest neighbor approach and knowledge guided induction. [Watson and Marir, 1994]. Both methods are candidates for aIFAS. A comparison of their performance and suitability as a retrieval algorithm is a major consideration of the research phase for this work.

3.5.1 Conventional Distance and Similarity Measures

The nearest neighbor approach can be developed along two differing lines of thought. It is possible to determine how different the cases are (distance decreases as case-match improves) or to determine how much the cases are alike (similarity increases as case-match improves). Numerous algorithms have been developed to provide measures of distance/similarity. The candidates that were selected represent the more common expressions of the logic paradigms used for such calculations. They include conventional Euclidean distance; as well as, representatives from non-Euclidean distance, semi-metric similarity, and pattern matching distance.

3.5.1.1 City Block (Manhattan) Distance

There are occasions when distances cannot be measured by straight lines between two points. The example commonly used to describe this metric (and hence its name) is the problem of measuring distance in a city. It is assumed that you cannot go diagonally (buildings block that path). Distance between points is measured from the origin by following the streets and making appropriate turns to arrive at the destination. When implemented, Equation 3.1 shows how City Block distance is computed.

$$City\ Block(X, Y) = \sum_{i=1}^n \|x_i - y_i\| \quad \text{Equation 3.1}$$

Consider two cases X and Y , each represented by a sets of n attributes denoted as x_i and y_i respectively. The City Block distance is a summation of the absolute values of the difference between the individual attributes representing the two cases.

This is a very simple metric that finds frequent application for case comparison [Wilson and Martinez, 1997]. The formulation can accept interval, discrete, or binary attribute values. This was selected as representative of non-Euclidean methods for comparing two cases.

3.5.1.2 Cosine Correlation Similarity

Rather than computing a distance, cases matching can be accomplished by comparing how closely the attributes agree. This method provides such a measure by calculating the cosine angle between two vectors. Case attributes correspond to the dimensions of vectors for the respective cases. When implemented, Equation 3.2 shows how Cosine Correlation similarity is computed.

$$Cosine\ Correlation(X, Y) = \frac{\sum_{i=1}^n (x_i \times y_i)}{\sqrt{\left(\sum_{i=1}^n x_i^2\right) \left(\sum_{i=1}^n y_i^2\right)}}$$

Equation 3.2

Consider two cases X and Y , each represented by a sets of n attributes denoted as x_i and y_i respectively. The Cosine Correlation similarity metric is computed as the sum of the products of the attributes, divided by the square root of the product of the sums of squares for the attributes. This metric has been extensively used for retrieving documents

stored in databases [Gupta and Montazemi, 1997]. The formulation can accept interval, discrete, or binary attribute values. This comparison tool was selected as representative of semi-metric methods.

3.5.1.3 Euclidean Distance

This is the scheme used in conventional calculations to determine point-to-point geometric separation. It is a multidimensional expression of the Pythagoras Theorem for determining straight line distance. When implemented, Equation 3.3 shows how Euclidean distance is computed.

$$Euclidean(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Equation 3.3

Consider two cases X and Y , each represented by a sets of n attributes denoted as x_i and y_i respectively. The Euclidean distance is computed as the square root of the sum of squares of the differences between the attributes. This is perhaps the most commonly used metric for general distance calculation and case comparison [Wilson and Martinez, 1997]. The formulation can accept interval, discrete, or binary attribute values. This was selected so that the Euclidean method would be represented.

3.5.1.4 Hamming Distance

The Hamming distance method grew from a need to compare the similarity of bit patterns. The distance value that is computed is a count of how many bits are different between the two patterns being considered. It can be extended to compare any two sets of ordered values. In those applications, the distance measure is the number of items that do not identically agree. When implemented, Equation 3.4 shows how Hamming distance is computed.

$$Hamming(X, Y) = \sum_{i=1}^n \begin{cases} 1, & x_i \neq y_i \\ 0, & x_i = y_i \end{cases}$$

Equation 3.4

Consider two cases X and Y , each represented by a sets of n attributes denoted as x_i and y_i respectively. The Hamming distance is computed as a summation over all of the attribute pairs, where the value added is unity if the attributes are not equal and zero if the attributes are identical. This is a straightforward algorithm, used in almost the same frequency as the conventional Euclidean distance [Liao *et al.*, 1998]. The formulation is intended for binary data, but can compare interval, discrete, or binary attribute values. This was selected as representative of binary pattern matching methods.

3.5.2 The Knowledge Graph Similarity

It is reasonable to accept that the importance of an attribute should have an effect on ranking the priority of cases [Xu, 1994]. In some rule-based reasoning systems, the importance of a rule is calculated as the ratio of the number of training examples covered by the rule to the total number of training examples [Agre, 1995]. If the significance of rules can be determined in such a manner, then why not apply a similar approach to help classify and select cases in a case-based reasoning system?

Probability theory is known to quite adequately model certain patterns of human plausible reasoning, such as diagnostic inference. References on statistics, however, rarely make the connection between causality and probabilistic approaches. Literature almost creates the impression that causality and probability are mutually exclusive concepts. Yet, statistical validation is generally accepted as a necessary condition for proving causality.

Using a statistically based inference approach for case selection would offer a readily constructed and modified tool that should also be easier for users to comprehend and accept [Druzdzet, 1997]. The statistical inference algorithm would represent any dependencies among the attributes and would reflect the causal structure of the case-base [Onisko *et al.*, 1998].

Knowledge Graph Similarity (KGS) is a recently emerging approach for applying statistics to case-based reasoning systems; as such, few examples could be found in literature. Besides an AIDS prevention system [Reategui *et al.*, 1997], there is the HEPAR system, which employs a statistical construct for clinical diagnosis and training in the area of liver disorders. Other systems have been successfully applied to natural language interpretation, planning, vision, robotics, and data mining [Onisko *et al.*, 1998]. The methodology has also been applied in the domains of machine learning and machine fault diagnosis [Lin and Druzdzet, 1997].

3.5.2.1 How KGS Works

Statistics are capable of encoding the qualitative knowledge about the relevance of relationships in a knowledge domain [vanLeijen and Druzdzet, 1998]. The KGS mechanism for information representation offers a technique for statistically classifying and selecting cases. The methodology outlined here is an extension of work presented by the case-based reasoning system ChartD₂ for medical diagnosis [Reategui *et al.*, 1997].

Extending the KGS concept for use in the domain of failure analysis is quite simple. There are minor changes in terminology. The case categories of KGS become failure modes, and category descriptors become failure mode attributes. And, a procedural simplification of the case selection process is used. This entails following an alternate path in KGS case selection that bypasses the use of specialized attributes.

As an introduction to the KGS concept, consider the following illustration of a means to classify and select toys. Suppose that the attributes of red, cubic, rubber, spherical, and wood are used to index a collection of toys. For balls, the set of descriptive features could include rubber, spherical, and red. For blocks, the set of descriptive features could include wood, cubic, and red. Intuitively, because of their greater value in identifying a toy that is a ball, only the attributes rubber and spherical would be needed in searches of the collection.

KGS provides a means for computing a statistical measure from a set of example cases that assigns importance to attributes, rather than relying on subjective decisions. The importance of an attribute in the KGS process of case indexing is determined from its sensitivity and specificity.

Applied to failure analysis, sensitivity is a measure of how many example cases for a given failure mode share a particular attribute. A more sensitive attribute would be common to a greater number of cases with a particular mode. Specificity is a measure of how many example cases representing all of the other failure modes share a particular attribute. A completely specific attribute would not be shared by any other failure mode. The two values of sensitivity and specificity are computed in the following manner.

$$Sensitivity(M, A) \equiv \frac{N_{MA}}{N_M}$$

Equation 3.5

where

M denotes failure mode
 A denotes attribute
 N_{MA} is number of cases with mode M
 that also have attribute A
 N_M is number of cases with mode M

$$Specificity(M, A) \equiv \frac{N_{w/oMA}}{N_{w/oM}}$$

Equation 3.6

where

M	denotes failure mode	$Sensitivity(M_j, A_i) + Specificity(M_j, A_i)$, if X_i exists
A	denotes attribute		
$N_{w/oMA}$	number of cases without mode M and without attribute A		
$N_{w/oM}$	number of cases without mode M		

When applied to a set of example cases, a matrices containing sensitivity and specificity terms are produced. Each matrix, sensitivity or specificity, has entries for each of the failure modes, (M_j , $j=1, \dots, m$); as well as, entries for each of the attributes, (A_i , $i=1, \dots, n$), used to describe those failure modes.

This statistical-based indexing method uses a three-step process to search the case base and generate a solution. First, a degree of similarity is computed between the new case and the set of KGS sensitivity/specificity terms for each failure mode. Let the new case be described as X_i , $i=1, \dots, n$. The calculation is performed as follows.

$$Similarity(M_j) = \sum_{j=1}^m \sum_{i=1}^n \begin{cases} Sensitivity(M_j, A_i) + Specificity(M_j, A_i) & , \text{if } X_i \text{ exists} \\ 0 & , \text{otherwise} \end{cases}$$

Equation 3.7

where

M	denotes failure mode	m	is the number of failure modes
A	denotes attribute	n	is the number of attributes
X_i	i^{th} attribute of case X		

In the second step, cases are retrieved from the most probable failure modes. That is, from failure modes that had a similarity with the new case above a threshold level. (Rather than a threshold value, aIFAS uses the five failure modes that offer the best

match to the test case.) All example cases with the selected failure modes are retrieved for comparison. Ideally, this process reduces the number of cases that need to be evaluated against a new case.

The third step entails calculating a similarity between each of the retrieved cases and the new case. The same similarity equation is used. A similarity is computed for each of the example cases retrieved. One difference applies, if the new case is represented by $(X_i, i=1, \dots, n)$ and an example case is represented by $(Y_i, i=1, \dots, n)$, then both X_i and Y_i must exist for the corresponding KGS terms to be included in the similarity summation. The pairing of example case and new case with the highest degree of similarity is the solution. The KGS value between new case X and example case Y is computed as follows:

$$Similarity(X,Y) = \sum_{i=1}^n \begin{cases} Sensitivity(M_Y, A_i) + Specificity(M_Y, A_i) & , X_i \text{ \& } Y_i \text{ exists} \\ 0 & , \text{ otherwise} \end{cases}$$

Equation 3.8

where	X	new case	n	is the number of attributes
	Y	example case	M_Y	mode of example case Y
	M	denotes failure mode	A_i	i^{th} attribute for mode M_Y
	A	denotes attribute	X_i	i^{th} attribute of new case X
			Y_i	i^{th} attribute of example case Y

3.5.2.2 A Modification of KGS

The Knowledge Graph approach as posited in the ChartD₂ program for medical diagnosis [Reategui *et al.*, 1997] relies on simple tallies. When computing the sensitivity and specificity terms, it is a matter of counting how many cases have or do not have a particular mode/attribute combination. A modification of that methodology is suggested to potentially enhance the applicability of the metric.

Typically, binary-valued attributes are considered. A means to deal with ambiguities and subjective uncertainties would be a worthwhile addition to the case-indexing scheme. The concept of fuzzy importance is proposed as a correspondence to the idea of statistical based importance. Fuzzy importance has proven to be useful tool for the assessment of contributions that might be made by attributes when they cannot be definitely specified [Furuta and Shiraishi, 1984]. To incorporate fuzzy importance into the KGS scheme should be no more complicated than using fuzzy attribute values.

Consider the sensitivity term. This numerator of that term is a summation of the number of example cases that share a common mode and attribute. Suppose that rather than simply counting those instances, that a total is accumulated of the attribute value for the included cases. When dichotomous attributes are involved nothing changes. This would, however, broaden the applicability of the metric to include ranked attributes or fuzzification of attributes. The form of the modified sensitivity term is,

$$Sensitivity_{Modified}(M, A) = \frac{\sum_{i=1}^k \begin{cases} X_i(A) & , \text{ case has mode } M \text{ \& attribute } A \\ 0 & , \text{ otherwise} \end{cases}}{N_M} \quad \text{Equation 3.9}$$

where	M	denotes failure mode	k	the number of example cases in the knowledge base
	A	denotes attribute		
	$X_i(A)$	is the value of attribute A for example case X_i	N_M	is number of cases with mode M

The specificity term would remain unchanged. Recall that the numerator of that term is a count of example cases without a mode and attribute.

3.5.2.3 Justification of the Method

Under what circumstances can knowledge be trusted? Only when it is corroborated by an accumulation of relevant cases [Allemang, 1994]. The method just

presented would allow for just that sort of information validation. Statistics drawn from the stored case base would accredit the solutions offered. Moreover, the biases and inconsistencies in human reasoning present a strong argument for using statistics based decision aids to help improve on unaided human intuition [Druzdzal, 1966]. It is important to remember that in case retrieval, it is important to get not just the most similar case, but the most relevant case [Nicholson, 1992].

3.6 The Logic of Fuzzy Logic

The majority of case-based reasoning systems still rely on binary logic or a crisp sets of attributes in their basic framework [Liao *et al.*, 1998]. Conventional rule-based reasoning systems generally continue to require a strict Boolean match for deciding their results [Dutta and Bonissone, 1993]. Yet, in the past decade, fuzzy logic has proved to be a powerful tool for decision making systems [Uebele *et al.*, 1993], recognized, but seldom exploited.

Using the principles of fuzzy logic in an expert system is quite sensible. Human beings in their day to day communications make use of imprecise information. They describe things as big or small, heavy or light, short or tall. That is, things belong to arbitrary, qualitative intervals [Fourali, 1994]. Fuzzy logic models the imprecise modes of reasoning which play an essential role in the remarkable human ability to make rational decisions in an environment of uncertainty and imprecision. That sort of intrinsically vague knowledge is typical of the decision processes involved in diagnosis. Experts deal with the problems in qualitative and non-deterministic ways using natural language instead of rigorous propositions of mathematical logic [Binaghi, 1990].

It is not that unusual for measures of various criteria to be linguistically oriented. For example, the finish produced on a surface by dried paint might be flat, matte, or

glossy. To be dealt with effectively, this type of attribute requires that fuzzy reasoning be somehow incorporated into the case selection algorithm [Xu, 1994]. A coincidental advantage of using fuzzy attribute values is that communication with the user can be in the same language that would naturally be used [Fox, 1977].

A powerful feature of fuzzy logic is realized when applied to case classification. Consider attributes related to washing clothes; the presence of bleach can damage colored clothes but is essential for producing white linens. Alternatively, think about the groupings of vehicular faults; a flat tire is a minor inconvenience when stationary but can be life threatening when moving. By using fuzzy logic it becomes possible for an attribute to have a different value based upon context; or, to place an example case in more than one category with differing degrees of membership [Yuan and Shaw, 1995].

The addition of fuzzy logic to an expert reasoning system has some demonstrable benefits. No contemporary computer program can replace the intuitive strengths of human judgement. However, when properly implemented by taking into consideration all available resources, computerized reasoning systems can contribute rigorously unbiased assistance in evaluating a huge number of complex problems [Binaghi, 1990].

4. THE aIFAS FRAMEWORK

The operation of aIFAS is not complicated. However, by virtue of its scope it might be considered complex. To successfully create such a system demanded a proactive approach with a clearly defined plan [Covey, 1989].

It is important to reassert that failure analysis is indeed an appropriate domain for an expert system application [Liao *et al.*, 2000b]. An expert system for failure analysis should, as a minimum, meet certain requirements [Walton, 1991]. The conditions required of aIFAS are: submit case data for subsequent problem solution; query the knowledge base for support information; and, offer training opportunities for novices. The implementation of aIFAS presented by this prototype has the additional function of supporting parametric studies of system performance.

4.1 Main Purpose of aIFAS

The goal of this reasoning system is not to replace the human, rather to provide a tool that allows the user to consistently obtain better results [Backer *et al.*, 1988]. To date, there is no other computer-aided failure analysis systems that incorporates the features available with aIFAS.

Most systems are developed as design tools, such as CRACK to diagnose steel bridge fracture. Other such systems usually focus on a particular failure mechanism. For the most part, these systems tend to concentrate on predicting rather than analyzing failures. Such as MIC.AB that finds examples of microbiologically influenced corrosion [Graham-Jones and Mellor, 1995].

The Failure Analysis Diagnostic Expert System (FADES) was developed at Southampton University with a hybrid approach for developing solutions to problems. A limitation of FADES is the fact it can only resolve problems with eleven of the more

common failure modes [Graham-Jones and Mellor, 1996]. No subsequent reports on recent advances with that system have been found in the literature.

The aIFAS package makes available appropriate, adequate, and reliable support beyond what is obtainable elsewhere. The eventual coverage of aIFAS will include both common and unusual failures in industrial, commercial, or residential settings. The aIFAS program has the facility to offer multiple levels of detail to maintain adequate data granularity for rendering meaningful responses with the necessary particulars. An intuitive user interface is an integral element of aIFAS. To accommodate changing requirements, aIFAS is easily extensible. Above all, aIFAS is an effective research and learning tool.

4.2 The Basic aIFAS Structure

Real-world systems should strive for simplistic design [Jackson, 1988]. Besides adhering to Ockham's Razor that the simplest answer is usually the best answer, reasoning systems need to exhibit acceptable performance levels. While meeting those criteria, the systems should also be usable and efficient [O'Keefe *et al.*, 1987]. In short, there are many factors to deal with while creating a viable reasoning system.

No system, however, will be accepted if it fails to yield understandable results [Adlassnig, 1986]. Ultimately, users do not care whether a system uses sophisticated techniques or random guesses to generate results. What they care about most is ease-of-use and if the system provides them tangible benefits [Hinkle, 1995].

4.2.1 The User Interface

In many cases the disappointing history that reasoning systems have had can be traced to a lack of respect for the user [Allemang, 1994]. The successful systems strive to use operational techniques and decision process with which users are comfortable.

Otherwise, frustrated users will shun the system and return to their old way of doing things [Klinger, 1994].

What qualities, then, should the user interface have? Needless to say, the user interface must be at least able to display and update system messages or problem results comprehensibly [Padalkar *et al.*, 1991]. A user interface that is so complex that deciphering it slows the problem-solution process is unusable [O'Keefe and O'Leary, 1993]. Put more succinctly, good interfaces are invisible [Cooper, 1995].

The aIFAS user is provided an informative interface that keeps them aware of system activity. Figure 4.1 is a greatly reduced, but typical view seen by a user. From this control panel, the various activities of aIFAS are managed.

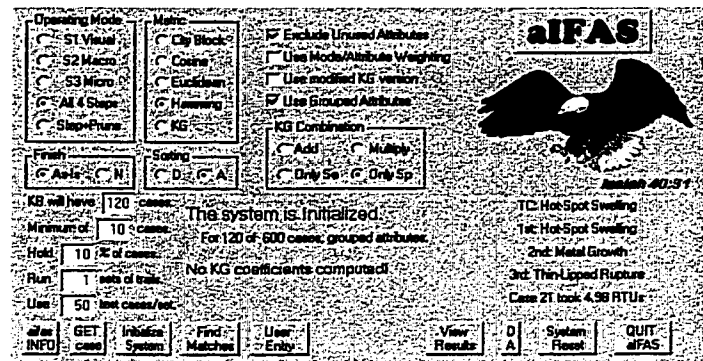


Figure 4.1: The basic aIFAS user interface.

An enlarged version of the figure and examples of other activities are presented in Appendix A. There are intermediate messages showing results as they are being produced, progress reports of protracted activities are made to keep the user informed that the program is still functioning, and a variety of opportunities for interactive decision to guide system operation or display data and results.

4.2.2 Modular Architecture

While designing an expert system it must be kept in mind that even the highly efficient implementation of an invalid system is absolutely useless [O'Keefe *et al.*, 1987]. Prior work on this project has shown aIFAS to be an intricate, but valid, concept. The system construct is based upon an architecture of specialized modules. That modular

structure for aIFAS scales the task of implementation into several manageable parts [Morales and Garcia, 1990].

There are secondary benefits to building a system from small pieces. Highly modular systems afford an advantage when maintenance is needed [Vargas and Raj, 1993], the logic of a small component is less convoluted than for a monolithic system. A modular system is also easier to reconfigure [Golding and Rosenbloom, 1996], only the affected part needs to be changed or replaced.

4.3 Individual Functions of aIFAS

First and foremost, aIFAS is a human-support-oriented system. Its operation keeps well in mind the need (and value) of user interaction in the problem solving process [Walton, 1991]. This prototypical implementation of aIFAS marks a transition from a machine-centered to a user-centered computing, it attempts to not make undue assumptions about what users are going to request of it [Allemang, 1994]. What aIFAS does instead, is provide basic support resources for directing a failure investigation, retrieving/displaying information, or providing for data entry.

4.3.1 Support Parametric Studies

This implementation of aIFAS has enhanced functionality. It was especially tailored to support studies of case comparison metrics, data structures, and the process of

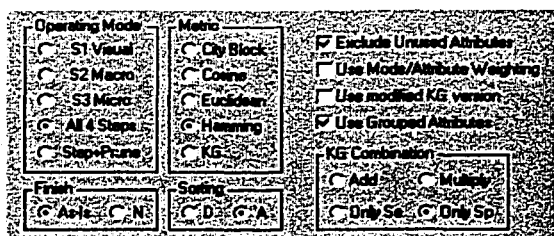


Figure 4.2: Parameter controls for research using aIFAS.

incremental learning. The set of parameters controls in Figure 4.2 allow a wide range of options to be explored. There are also three threshold values that the user can change. The quantities managed by those thresholds are illustrated in Figure 4.3. The program controls

that can be changed include the following: which of the five metrics will be used, will the metrics be used as-computed or normalized, will the computed metric value be ordered in ascending or descending, will unused case attributes be excluded from consideration, will weighting factors be used

KB will have:	120	cases
Minimum of:	10	cases
Hold:	10	% of cases
Run:	1	sets of trials
Use:	50	test cases/set

Figure 4.3: Threshold values for aIFAS.

with the metrics, is the independent or the grouped data structure to be used, is the modified KGS method to be used, which combination of KGS terms will used, how many cases of the 600 chronologically ordered possible cases are to be in the knowledge base, and what are the minimum number of cases to support analysis. There are quite a few control choices available to support research activities.

4.3.2 Direct Failure Investigations

The notion that human expertise can be divided into small chunks called rules, or instructions, suggests that a directed problem-solving methodology has advantages [Vargas and Raj, 1993]. Why not apply a procedural methodology to advantage in a failure investigation? Furthermore, by compelling the examination and testing process to track a logical path, the number of time-consuming and costly tests necessary to provide an accurate solution can be greatly reduced [Thielsch, 1977].

Working through a problem in an orderly manner has other advantages. It is easier to recognize when the boundaries of the system knowledge are reached and thereby avoid making faulty decisions [Fourali, 1994]. Also, in the event a solution cannot be reached, why an unresolved case response occurs can be better understood [Watson and Marir, 1994].

Directing the course of an investigation can also have educational value. Not everyone interested in a failure has the same level or kind of training [French, 1983].

This is compounded by the fact that in the particular field of fault investigation and diagnosis, it is difficult to train and then retain qualified personnel [Doherty *et al.*, 1994]. The insight gained from the interaction while going through a failure analysis may be even more important than the actual result that is obtained [Druzdzal, 1966].

The aIFAS program is designed to guide the user while accessing program features and, hence, indirectly controls failure investigations. Figure 4.4 shows the control buttons on the lower portion of the aIFAS user interface. The function of each button is revealed by a control tip that appears if the cursor pauses over the button. The aIFAS INFO button accesses queries of the knowledge base. The GET Case button will



Figure 4.4: The set of aIFAS program control buttons used to guide a failure analyses or parametric study.

recall the full report stored for one of the example cases. The *DA* button will display the analysis results for a series of tests in a parametric evaluation. The function of the remaining buttons is self-explanatory. See Section 5.2.1 for additional examples of how these functionalities are built into aIFAS.

4.3.3 Stepwise Entry of Test/Unknown Cases

In many instances, only a fraction of the facts is often sufficient to fully diagnose a problem and suggest its solution. For example, excessive internal pressure can cause a rupture, or cyclic loading can lead to fatigue cracking. Moreover, the failure analysis process, by its very nature, consists of incremental discovery. Depending upon the particulars of a specific failure, the depth of analysis required and the sequence of investigation for identifying failure causes might be quite different [Liao *et al.*, 2000b].

It is worthwhile understanding that in failure analysis the intermediate results and subsequent decisions made during an investigation are of extreme importance to the final solution. Few reasoning systems attempt to incorporate those intermediate steps into their structure [Esogbue and Elder, 1979]. Yet a task crucially important for improving the problem solving power of reasoning systems is to develop ways that they can behave smarter in the course of their operation [Agre, 1995]. What better approach to achieve that end than to proceed along the path of an investigation in a stepwise fashion?

The stepwise entry of case data can be done manually for a set of Test Cases, or an automated process if provided by accessing the User Entry module. See Section 5.2.1 numerical and verbal where this process is explained with examples.

4.3.4 Suggest Most-Likely Failure Modes

A case-based reasoning system can easily present listings of similar cases for review, even at intermediate stages of the solution process [Stottler, 1994]. The aIFAS program, like other systems, intentionally does not choose a final answer, possibilities are offered, leaving the ultimate decision to the user [Binaghi, 1990]. It does present a list of most closely matching cases ordered by their degree of similarity [Klinger, 1994].

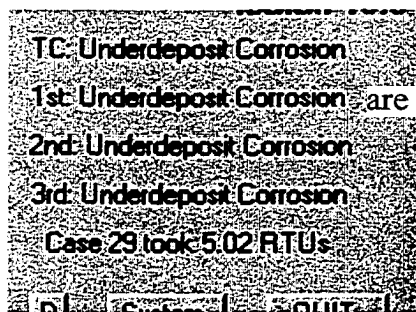


Figure 4.5: aIFAS produces optional real-time results.

The user interface, to be effective, includes numerical and verbal information about the cases that are offered for consideration [Druzdzal, 1966]. The human user can choose to accept the results as presented; reject them and start over; or, when allowable, continue on to the next step of the investigation [Isaacson *et al.*, 1997].

These capabilities are available in the prototype

aIFAS. Figure 4.5 shows a set of intermediate results that were displayed while the system analyzed a set of Test Cases. Additional screen views of methods for display of results are provided in Appendix A.

4.3.5 Queries of Stored Information

Provisions for accessing explanations and consulting the knowledge base are part of the system. There is a range of possibilities to accommodate everyone from the naïve user with little domain knowledge to the failure analysis expert [Lee and O’Keefe, 1994]. The user can retrieve a variety of auxiliary data and factor that information into the assessment of a solution [Stottler, 1994].

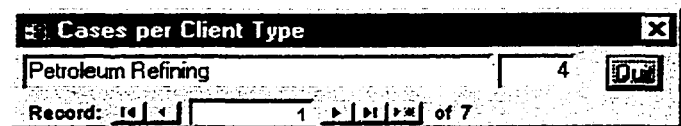


Figure 4.6: One of the ways in which the aIFAS knowledge base can be queried.

Figure 4.6 is one example of a variety of pre-defined queries that the aIFAS user can access. This display would allow finding the Case ID numbers for the seven failure cases representing the petroleum refining industry.

4.3.6 Retrieve/Display Information

Advantage is taken of the capability in case-based reasoning systems to readily display examples cases, complete with text, graphics, and even sound [Barletta, 1991]. This is an ability of intangible value. Humans often depend on such prompts to initiate information retrieval. The reader is referred to Section 5.2.2 that offers an example of retrieving the full report underlying an example case in the knowledge base.

4.3.7 Auxiliary Data Management

The minimum information to be stored consists of the attributes needed to identify and describe the example case [Stottler, 1994]. Associated report documents and data spreadsheets can be included [Churbuck, 1992]. A special benefit comes with the

inclusion of graphical objects in such form as diagrams, charts, photographs, or drawings [Perez and Koh, 1993]. All these types of supporting information are stored by aIFAS and are accessible via queries to the user.

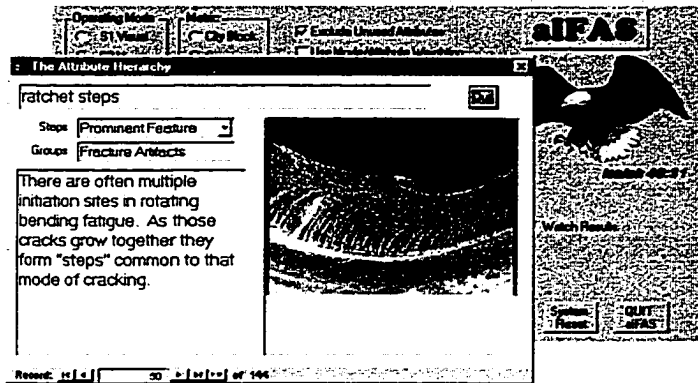


Figure 4.7: A word description, combined with graphic documentation to provide user information.

Figure 4.7 shows what is accessible to the aIFAS user if they have questions regarding failure mode attributes. The reader is referred to Appendix A for additional screen views of system operation that illustrate this and other features in more detail.

4.3.8 Data Entry Checking

Entry validation schemes are built into aIFAS to prevent as much error from creeping in through data entry as is feasible. This does not, and cannot, contend with flawed information resulting from reporting bias [Tzafestas *et al.*, 1994]. In Section 5.2.1 examples are provided of how the system addresses missing or inappropriate data entry.

4.3.9 Fuzzification of Case Attributes

Decision processes often begin by assuming relevant factors, or attributes, of the problem have only dichotomous values. That sort of binary-thinking is contrary to observation [Esogbue and Elder, 1979]. Human beings are more comfortable making imprecise verbal statements instead of quantitative estimations for their preference, judgement, or prediction. They use qualitative terms such as good and poor to grade materials, or specify properties with ranges of values. These are all fuzzy representation [Liao, 1996]

The process of fuzzification in a simplistic sense consists of assembling lists of attributes and assigning ranges of possible values [Renard *et al.*, 1993]. Actually, experience shows this to be a difficult and time-consuming process, especially for historical cases stored without that intent in mind [Zhan, 1998].

The benefits of fuzzy logic can be realized by aIFAS. Rather than using binary attributes, the case selection algorithms can use interval-type values. This approach lets an attribute have values suited to individual case context, or to assume levels of relevance keyed to a failure mode. The calculation schemes can easily accept these fuzzy attributes with significance influenced by historical precedence, a method that offers a vast improvement over using purely crisp (binary) attribute values [Esogbue and Elder, 1980]

What remains is to determine the value to assign a particular attribute as it is entered into the knowledge base. Again, a straightforward scheme is adopted by aIFAS. Values are selected by the user from tables, such as Table 4.1, which link linguistic terms to possibility ranges. The possibility ranges are represented as triangular fuzzy numbers.

Table 4.1: Example Attribute Possibility Values.

Possibility Range	Adjective	Adverb
(0,0,0,0)	<i>Impossible</i>	<i>Never</i>
(0,0,0,.1)	<i>Very unlikely</i>	<i>Very rarely</i>
(.1,.25,.25,.4)	<i>Unlikely</i>	<i>Rarely</i>
(.25,.4,.4,.5)	<i>Fairly unlikely</i>	<i>Fairly rarely</i>
(.4,.5,.5,.6)	<i>Less likely than not</i>	<i>Less often than not</i>
(.5,.5,.5,.5)	<i>As likely as not</i>	<i>As often as not</i>
(.5,.6,.6,.75)	<i>More likely than not</i>	<i>More often than not</i>
(.6,.75,.75,.9)	<i>Fairly likely</i>	<i>Fairly often</i>
(.75,.9,.9,1)	<i>Likely</i>	<i>Commonly</i>
(.9,1,1,1)	<i>Very likely</i>	<i>Very commonly</i>
(1,1,1,1)	<i>Certain</i>	<i>Always</i>

This functionality alone has not yet been implemented, pending a decision on what data structure and case comparison metric will be incorporated into aIFAS.

5. TESTING THE SYSTEM

When considering how to test aIFAS, it was discovered that there is no generally accepted criterion for evaluating reasoning systems [Tzafestas *et al.*, 1994]. Further, of the various methods proposed, many are incomplete, poorly systematic, or not easily applied. Some thinking ran the risk of fashioning a measuring tool more complex than the system being evaluated [Guida and Mauri, 1993]. Another source even supported the concept of building a duplicate model system as an effective platform for knowledge verification and validation [Wu and Lee, 1997]. Before settling on a particular testing methodology, perhaps the why of testing, especially for aIFAS needs to be studied.

The paradox in applying reasoning systems is that we want them to do perfectly things that are not really understood. They are expected to provide advice where errors may lead to damage of facilities and equipment, economic loss, or even the loss of life [Lee and O'Keefe, 1994]. The potential for disaster due to faulty reasoning from inadequate testing grows with system complexity [Cragun, 1987].

The responsibility rests with the builders of reasoning systems to ensure that their creation will give its users accurate advice or correct solutions [Suwa *et al.*, 1982]. All the while taking care to avoid any perception of developer bias in their evaluation protocol or subsequent findings when testing the system. Credibility, the extent to which a system is believable, is important [O'Keefe and O'Leary, 1993].

Of premiere concern with aIFAS are performance validation and a usability evaluation. These types of tests are typically performed empirically, for example by using the system in its problem-solving mode on actual test cases [Mengshoel and DeLab, 1993]. Generally, testing tries in some way to measure the correspondence between the system results and those produced by human experts. The results are often

quantified by using statistical methods. In most cases, the empirical method approach can be expected to be the most effective means for system validation [Preece, 1990].

From a different perspective, it is important to realize that any reasoning system will inevitably make mistakes. Human experts also make mistakes, and the tendency is to tolerate them – especially because they are experts. Thus, the main questions for tolerance of mistakes are “What is the price to be paid for each mistake?” and “How often are the mistakes made?” Rather than aiming to verify the behavior of an expert system, it is a better idea to measure its performance [Parsaye and Chignell, 1988]

Taking all into consideration, but working mostly from the last premise, the testing plan decided upon for aIFAS does three things. First, it verifies that the system can do what it is supposed to do. Second, it measures the accuracy and performance. Third, it evaluates the ability of aIFAS to learn and grow.

5.1 The aIFAS Knowledge Base

The example cases for this research were drawn from electronic files. They are failure analysis reports produced by Scientific Testing Laboratories, Inc. during the years from 1989 through 1998. (During that period the Author was affiliated with the company. The information is being used with the permission of the current owner.) Of the available records, there were 1184 reports that were determined to be acceptable for consideration in this project.

5.1.1 Choosing Cases for aIFAS

The group of 1184 reports was ordered by date and report number. A set of 50 test cases was extracted from that list by choosing successive reports at a twenty-count interval. The reasoning behind that selection method was to produce a real world set of cases, similar to what might actually be encountered by aIFAS. This approach was

considered more reasonable for general evaluation purposes than random selection or an artificial construct of some predefined membership (*i.e.*, so many of each type of failure mode, or one example of each failure mode, or some such scheme).

The set of example cases for the knowledge base was taken from the reports remaining on the original ordered list. This group consisted of a 600-member knowledge base of stored cases for aIFAS. The knowledge base available for testing, therefore, represented roughly five years of failure analyses as they were performed in a commercial laboratory. Those stored cases would necessarily provide a representative mixture of failure modes that could be expected in a real world application of aIFAS. When needed, groups of example cases are selected as the “first-however-many-are-desired” from the basic group of 600 using a simple chronological ordering.

5.1.2 Knowledge Base Structure

With deliberate intent, no limited set of failure modes and attributes was chosen for consideration. A list containing 95 of the most-likely failure modes to be encountered was constructed. Matrices showing cause-and-effect relationships were formed from that basis. One matrix used 144 independent, binary attributes to characterize the various failure modes. Figure 5.1 shows

how a typical case might be represented using a string of binary

Case #52 000100001010000000100000...

Figure 5.1: Representation of a case using binary attribute values.



Figure 5.2: Representation of a case using grouped, interval-valued attributes.

values. A second matrix used those same 144 attributes, but arranged them in 32 groups of multi-valued attributes. In the resulting hierarchical structure, each of the original independent attributes was

assigned a subjective interval value. The values were based upon the Author's ranking of the attribute from professional experience as a failure analyst. (The ratio of descriptive attributes to individual failure modes may seem low; however, frequently a single attribute is sufficient identify a failure mode conclusively.) Table 5.1 presents the distribution of failure modes and attributes as was observed in the set of Stored Cases used with this research.

Table 5.1: Distribution of Failure Modes and Attributes in the Stored Cases

Number of Stored Cases	Failure Modes Represented	Independent Attributes Used	Grouped Attributes Used	Test Case Failure Modes Represented
First 120 cases	47	94	32	23
First 240 cases	61	105	32	27
First 360 cases	74	120	32	28
First 480 cases	80	126	32	28
All 600 cases	83	131	32	28

The majority of the failure modes and attributes had at least one representative in the set of stored cases. Only 12 of the failure modes and 13 of the independent attributes were unused. There must be 360 example cases in the knowledge base for all of the failure modes contained in the test case set to be represented in the stored cases. That presents a credible condition to aid in evaluating the effects of incremental learning.

Table 5.2: Number of Attributes Used To Describe a Failure Mode

Attributes to Describe a Failure Mode	Independent Attributes	Grouped Attributes
Minimum Number Used	1	1
Average Number Used	4	4
Maximum Number Used	8	8

The set of attributes was intentionally kept small. This was primarily to facilitate easier data entry of example cases. An interesting observation, nevertheless, was that although they were available, an expansive list of attributes was not necessary to describe a particular failure mode. (Subsequent results will show that even these small numbers of

attributes were sufficient to produce good answers.) Table 5.2 offers the statistics on attribute use in both sets of example cases.

5.2 Verification of Capabilities

Verification is proof that the system does the job right [Andert, 1992]. For the most part, verification investigates aspects that are not open to subjective appraisal, so objectivity is generally not an issue [O'Keefe and O'Leary, 1993]. Either a system does what is supposed to or it does not.

Verification testing of aIFAS considers two functions that the system is supposed to support. First, does the system appropriately direct a failure investigation? Second, can the user explore the system information and auxiliary material stored in the system knowledge base?

5.2.1 Directing a Failure Investigation

The core purpose of aIFAS is to be an aid to the skilled or average analyst as they go through the process of a failure investigation. It can also serve, however, as a training vehicle for the novice. The most direct means of learning is via access to the opinions of experts that aIFAS permits [Shortliffe and Buchanan, 1975].

In routine operation, aIFAS leads the user through the typical steps of a failure investigation. The system provides direction for the general course taken and makes suggestions regarding possible actions. In that manner aIFAS contains user activities within the limits that real human experts would impose on themselves [Witten and MacDonald, 1988].

This level of control is provided by the system in different fashions. Overall, direction is derived from the aIFAS Manager control buttons

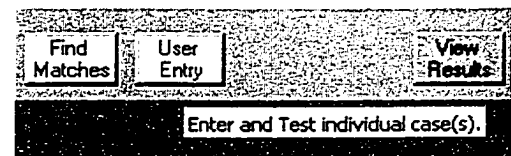


Figure 5.3: How control buttons and messages direct the way aIFAS operates.

at the bottom of the interface display. The button for the next operation that a user should perform is highlighted. Note that Find Matches and User Entry buttons are a lighter color in Figure 5.3. These are the next available choices after the system is initialized with a new set of conditions. Other actions are blocked and data entry of new parameters is locked out. Passing the cursor over the User Entry button caused the control tip “Enter and Test individual case(s)” to appear. Similar explanatory messages are available for other control features.

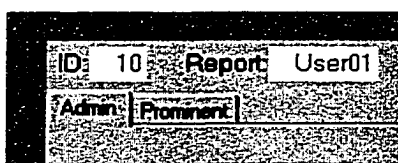


Figure 5.4: The two tabs that can be accessed on the first visit to Data Entry.

Assume the user elects to enter case data and test it against the system. Making that choice, opens a data entry form. The course of data entry is forced to along a sequential path, a sequential path that follows the methodology of traditional failure analysis. When the

first visit is made to the data entry form, only administrative and prominent feature information can be input. Subsequent visits to the form after analysis of the entered data, permit access to additional parts of the data entry form. Figures 5.4 and 5.5 show the addition of page tabs that can be accessed for data entry.

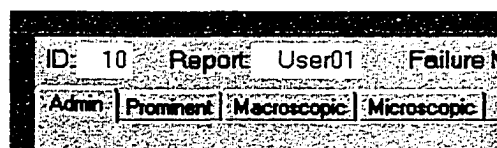


Figure 5.5: The third visit to the Data Entry module allows access to four tabs.

Other checks are applied to insure that all of the required information is entered

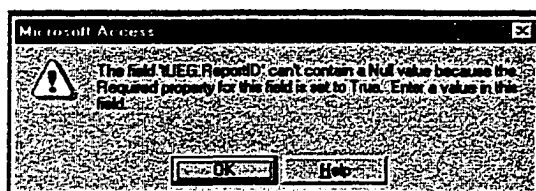


Figure 5.6: aIFAS message generated if required data is not entered.

for a case. Figure 5.6 shows the error message that results if the user forgets to enter an identifying name for a case being input for testing. There is also validation testing of critical data to insure that appropriate

information is provided aIFAS for analysis. A typical system message generated whenever improper data that is subject to validation testing is entered into aIFAS is shown in Figure 5.7.

Each time aIFAS guides a user through an investigation is an act of system verification. The completion of successive steps tests the consistency, completeness, and correctness of the system [O'Keefe and O'Leary, 1993].

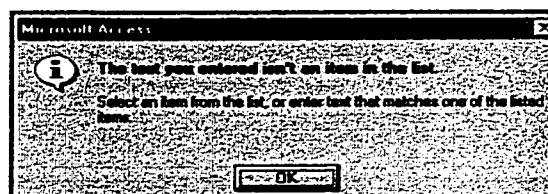


Figure 5.7: An error message resulting when improper data is entered into aIFAS.

5.2.2 General Informational Query

A reasoning system knowledge base should contain the necessary information and be able to present it in a form useable for helping solve problems [Suwa *et al.*, 1982]. To accommodate that requirement better, aIFAS provides for supplementary material to be archived. This is information that augments the contents of the knowledge base in a conventional case-based system.

It is possible for the user to initiate a request that retrieves and displays for review the full version of the report underlying an example case in the aIFAS knowledge base.

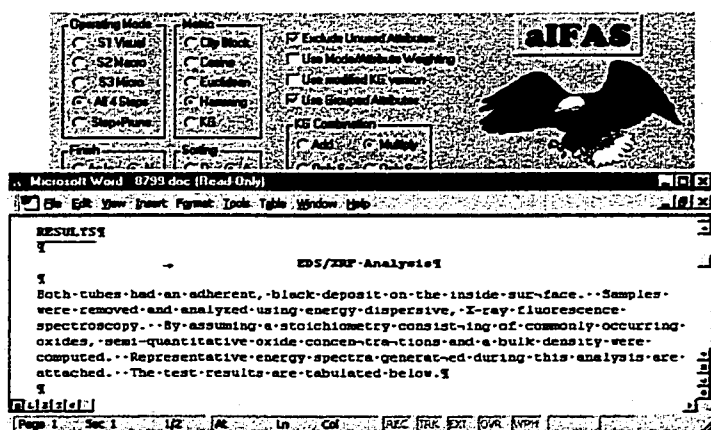


Figure 5.8: Retrieval for user review of a report underlying an example case in aIFAS.

The additional insight gained from studying a similar failure example can be invaluable in resolving a current problem. Figure 5.8 shows a screen view of what results from such a request.

The user can query the aIFAS knowledge base in other ways. Data can be displayed based using a hierarchical knowledge structure or as combinations of stored data. Figure 5.9 shows the possibilities.

(Screen views showing examples of the sorts of data

offered by each of the selections are provided in

Appendix A. Generally, the choices include

hierarchical display of failure mode or attribute

examples, listings of example cases fitting different

criteria, and a tabulation of the sorts of components or

materials that are represented in the knowledge base. This feature opens the possibility

of discovering underlying domain relationships of which the user might be unaware

[Barletta, 1991].

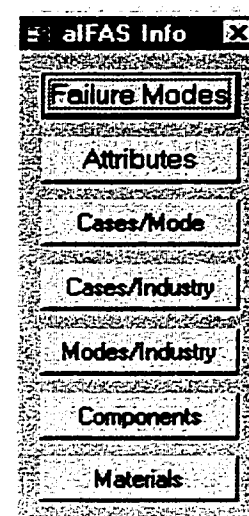


Figure 5.9: The control used to pick the data aIFAS displays.

5.2.3 Learning from aIFAS

The educational value of aIFAS has been alluded to throughout its development. This prototype is capable of demonstrating the sorts of information that can be conveyed to the novice by aIFAS. Perhaps the greatest wealth of information is available when the user is accessing the User Entry portion of aIFAS. In that module, the user can request that extra information be made available.

Initiating that request activates a series of pop up forms. Those forms are

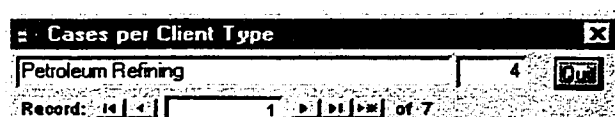


Figure 5.10: An example of the information retrieved by selecting an industry type while in User Entry mode.

displayed by selecting an industry type, failure mode, or even an individual attribute. Figure 5.10 illustrates the information that is displayed when an

industry (client type) is selected. In this example, there is a hierarchical description of the failures modes experienced within that industry as contained in the aIFAS knowledge base. Included is the Case ID number that can be used to recall the underlying report for each of the corresponding cases.

5.3 Validation of Performance Accuracy

Most definitions of reasoning systems mention their ability to perform close to human expert levels. Yet the testing to validate that level of performance has mostly been ad hoc, informal, and of dubious value [O’Keefe *et al.*, 1987]. That poses a strong contrast with the supposition that validation is supposed to be the process of assuring a system to be sound, producing the desired result, and sanctioned through systematic analysis and testing [Andert, 1992]. With aIFAS, validation is carried out to show that the system can performs the real-world tasks for which it was created [Preece, 1990].

Obviously, reasoning systems cannot be expected to perform better than the human experts they mimic do. The validation efforts should concentrate on ensuring performance at a realistic level [Andert, 1992]. The most reasonable approach is to define the level of expertise at which it should perform [O’Keefe and O’Leary, 1993].

5.3.1 Performance with Test Cases

Virtually all of the approaches used for evaluating (or testing) reasoning systems are derived from experimental use of the system in controlled situations [Guida and Mauri, 1993]. Generally that testing is conducted to ensure quality results and functional methodology [Martine-Mattei, 1992]. Exercising the reasoning system against test cases has been one of the most common methods [Andert, 1992].

Choosing the suite of test cases is a critical task; an improper selection can gravely bias the results. Some guidelines for picking the example case test set include:

they should span the knowledge domain by reflecting all the sorts of cases that will be encountered by the system; and, there should be enough cases to exercise system parameters and to establish some statistical measure of significance; plus [O’Keefe and O’Leary, 1993].

The scheme described in Section 5.1 meets those requirements. Another issue, however, is how are the test cases applied? Is a single test acceptable? Or, should multiple trails be attempted to establish statistical norms?

5.3.1.1 One Time Testing

A comparison of the performance of different metrics against different data structures under varying conditions is to be accomplished. It seems reasonable that to perform a fair comparison, the candidates should be “Asked the same question”. That is, apply all 50 of the test cases each time and only one time. (There is no reason to assume that subsequent testing would yield different results.)

Testing just one time has precedence in literature [Gonzales, *et al*, 1998], [Reategui and Campbell, 1997], [Xu, 1994], and [Preece, 1990]. This one-time method of testing is used by aIFAS.

5.3.1.2 Multiple Trail Testing

It is equally fair to say that a single set of results may not be representative. That the diversity of the knowledge base is not being adequately explored. The idea of multi-trail testing to acquire statistical norms also has precedence in literature [Agre, 1995], [Battiti, 1994], and [Smyth and Cunningham, 1994].

In aIFAS testing, 6 trails using sets of 15 test cases randomly selected from the pool of 50 test cases are also performed. The 6 trails is the average of the number of trails run by the various sources in literature that reported such data. The subset size of

15 was selected as large enough to provide reasonable coverage, but small enough to allow for anomalous behavior.

5.3.2 Acceptable Results

In failure analysis, the expectation for the human expert is 100% accuracy. This requirement is imposed by a credibility issue between failure analysts and their clients. No one places much value on the answers from a sometimes-correct failure analyst.

Just what level of success, then, is acceptable for aIFAS? In one instance, when comparing a case-based system against a human expert, an expectation of 90% accuracy was considered reasonable [Xu, 1994]. Another source drew on the expertise of real estate appraisers and reached a consensus that 80% accuracy was acceptable. A third source insisted that 100% accuracy would be an acceptable performance target [Golding and Rosenbloom, 1995]. For test results produced with aIFAS, the average of those values, 90% accuracy, will be used as the limit for acceptable performance.

5.3.3 Measuring Performance

What sort of measure can be used to quantify performance? One source infers that qualitative validation of case-based systems is the norm [O'Keefe and O'Leary, 1993]. Alternate, more prevalent, views tend to rely on a simple percent-correct measures of accuracy [Agre, 1995]. Two sources, [Hoppe and Meseguer, 1993] and [Golding and Rosenbloom, 1995], went so far as to include the amount of time needed to solve a problem in their results when they reported on performance.

In aIFAS, three quantities are used to help measure performance. The first is a basic performance measure called the Correctness Ratio (CR), [Gonzales *et al.*, 1998]. The CR is defined as a percentage ratio between the number of successful solutions and the number of attempts made trying to produce a solution.

$$\text{Correctness Ratio} = \frac{\text{Number Correct}}{\text{Number of Attempts}} \times 100\%$$

Equation 5.1

A more complete measure of performance needs to include the time required to produce an answer in its methodology. There is the matter of somehow capturing the differences between the candidate metrics in this series of evaluations.

Each of the metrics should have the potential for producing the same answer. Assuming that is true, then the time to produce a solution becomes important. In some instances, consideration of their respective computational time could be the only way to differentiate their performance.

Another time issue is the bias introduced by the speed of the computing platform used to perform the analysis. To ameliorate that problem, time is adjusted into a dimensionless Relative Time Unit (RTU). The RTU is defined as the ratio of measured time for an evaluation divided by the time to perform a standard looping operation. (Note: The RTU is used for time reporting in aIFAS, with adjustment made by the program for the system currently being used.)

$$\text{Relative Time Unit} = \frac{\text{Measured Time}}{\text{Time to Perform [For } i = 1 \text{ to } 250,000 : \text{Next } i]}$$

Equation 5.2

A measure called Performance Score (PS) is proposed to incorporate accuracy and speed. The PS is defined as the CR value divided by the RTU for the particular evaluation being assessed.

$$\text{Performance Score} = \frac{\text{Correctness Ratio}}{\text{Relative Time Unit}}$$

Equation 5.3

6. RESEARCH RESULTS

For a singular reason, aIFAS came into being – to assist in the conduct of failure investigations. The system is evolutionary with no expectation of ever being complete.

This research effort focused on two general areas. One area was to gain increased understanding of the requirements, structure, and potential of the aIFAS knowledge base. The other area was the identification of a suitable metric for case selection in a case-based reasoning system.

The research generated a curious mix of information. Some results highlighted the richness of information content that would be available in a fully implemented system. Other data tended to dispel seemingly accurate intuitive beliefs. An attractive algorithm was shown more of an intellectual curiosity than a practical tool for aIFAS. Operational insights for aIFAS were gained for using the system in a stepwise fashion. An indication of how large the set of example cases for the case-based reasoning portion of the system was revealed. It was shown that aIFAS can learn as its knowledge base and can readily expand to include new failure modes. As lagniappe, the Relative Time Unit (RTU) was introduced to let aIFAS measure operational timing independent of the platform on which it is running and a Performance Score was devised to evaluate results using a combination of accuracy and speed.

6.1 Extra Information

Part of the work in developing a list of candidate failure modes included creating a hierarchical structure for classifying the individual modes. That was not an easy task, especially since failures are rarely the result of a single cause, they are more often of a mixed-mode nature. Adding to the difficulty is the practice among failure analysts of using a failure cause to name a failure mode. (In welding, a lack-of-fusion produces a

weak joint that most often fails due to ductile overload, yet such an incident is frequently referred to as a lack-of-fusion failure.)

The classification system that resulted consisted of 3 general divisions that were divided into a group of 7 classes. The 95 individual failure modes were distributed within that hierarchy. Figure 6.1 shows the hierarchical structure that was produced.

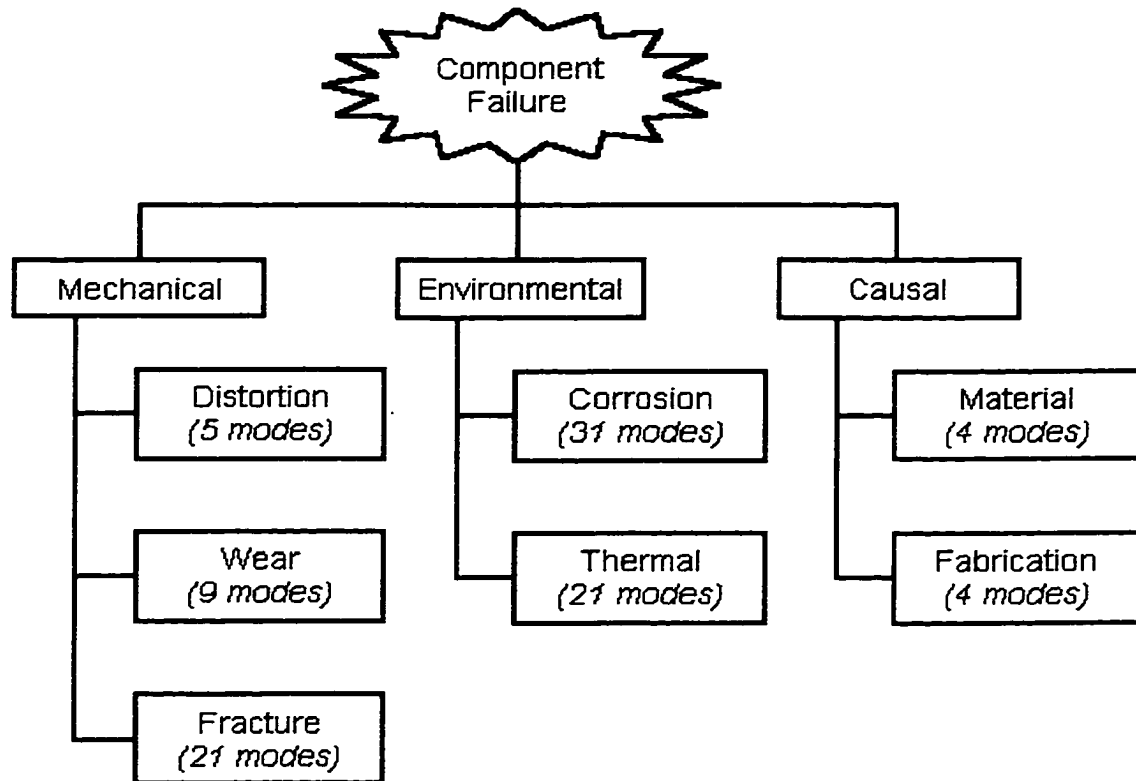


Figure 6.1: Failure Mode Classification System

The number of example cases in each of the classification categories was extracted from the aIFAS knowledge base. The results very closely matched what had heretofore only been conjectured. This might be a wobbly first step into the realm of data mining, but it clearly illustrates the untapped potential. The distribution of cases among the 3 divisions and 7 classes of failure modes are shown in Figures 6.2, 6.3, and 6.4.

The information about failure mode distribution is useful, but other figures were also made available. For the first time, more information was extracted from the example

cases than the essentials for identification and decision making. The extra data included the report date, a client type, the component that failed, and the material of construction for the failed component.

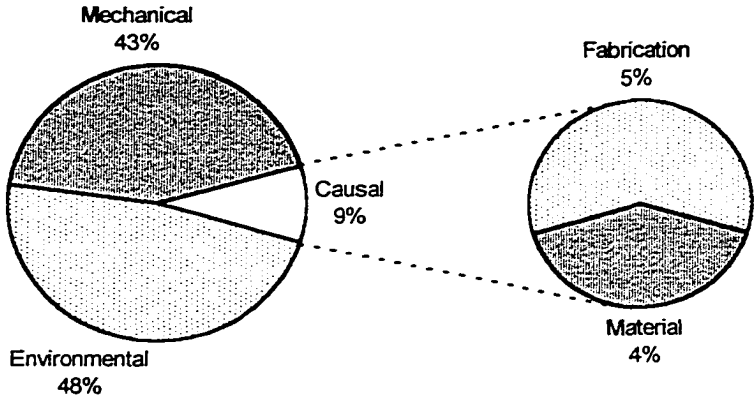


Figure 6.2: Failure Mode Distribution by Division and Causal Classes

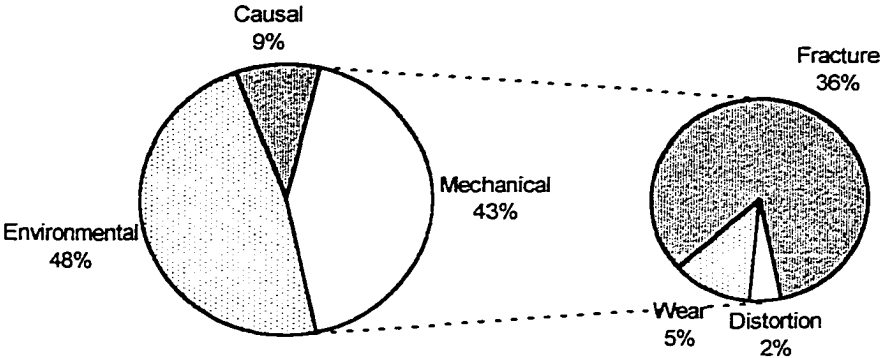


Figure 6.3: Failure Mode Distribution by Division and Mechanical Classes

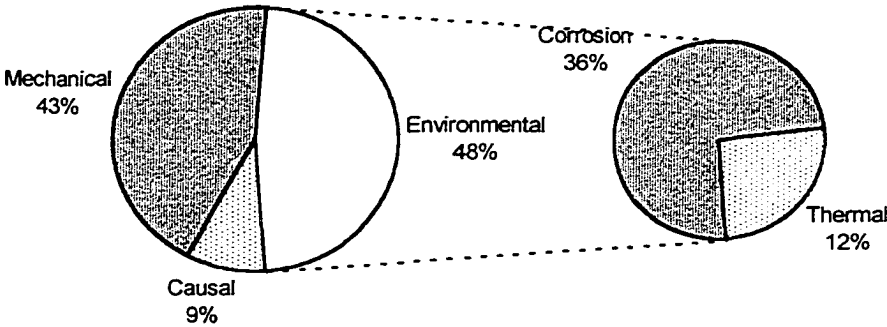


Figure 6.4: Failure Mode Distribution by Division and Environmental Classes

A summary of the data available includes that there were 24 different client types identified, the failures involved 340 different components, and the failed components represented 139 different materials of construction. A tabulation of failure mode distribution by failure class and client type is provided in Appendix B.

6.2 Metric Evaluation

The goal of this evaluation process is to select an efficient algorithm that yields the best possible case match between a test case and a set of example cases. The metric that is selected will be used in the case-based reasoning portion of aIFAS.

Table 6.1: Explanation of Parameter Settings

<i>Heading</i>	<i>Description</i>	<i>Choices</i>
Metric:	Method used to measure similarity or distance.	City Block, Cosine Correlation, Euclidean, Hamming, or KGS
Sorting:	Direction that results are ordered to select a match.	Ascending or Descending
Finish:	Form of the metric value.	As-Computed or Normalized
Mode:	Cycle of operation.	Input from 1, 2, 3, or 4 Steps used together; or a Step & Prune mode
Unused Attributes:	Action taken when EC and TC attribute value = 0 .	Include or Exclude them
Weights:	Are mode/attribute factors used with calculation?	Yes or No
KG Version:	Does KGS use the original tally or modified version?	Yes or No
KG Combo:	How are the KG terms combined for KGS or weighting factor use?	Added, Multiplied, Sensitivity alone, or Specificity alone
Attribute Format:	Which data structure is used for the attributes?	Independent or Grouped attributes
# EC in KB:	The number of Example Cases to use in the Knowledge Base?	1 to 600
# TC w/Modes:	The number of Test Cases that have solutions in the Knowledge Base.	1 to 50
Min # of EC:	The least number of Example Cases allowed.	1
Min % of EC:	The least percentage of Example Cases allowed.	0.1
Data Prefix:	Two character prefix for results stored in aIFAS	DA, MN, IE, WT, WM, SP, US, IL

Visual Basic for Applications code was generated that sequentially submitted the Test Cases for analysis against the example cases. The results are based upon 278 combinations of program parameters. The parameter settings for each evaluation series are given for reference when reviewing the comments and results. Table 6.1 lists the entry headings, their description, and the possible choices for those parameter settings. The raw data for individual parameter combinations are tabulated in Appendix C.

6.2.1 Distance or Similarity

Some of the candidate metrics compute a distance while others compute a similarity. It is assumed that when two cases are being compared, the distance between them is inversely related to their similarity. Those results, however, may not be a value in the interval from 0 to 1 (especially if other than binary attribute values are used).

Table 6.2 establishes the parameters used to obtain the data shown in Figures 6.5 through 6.8 for the 50 test case results and Tables 6.3 and 6.4 for the 6x15 test case results (see also DA001-DA020 in Appendix C).

Table 6.2: Parameter Settings for Distance/Similarity

<i>Parameter Settings for Distance/Similarity Series</i>					
Metric:	ALL	Unused Attributes:	Include	# EC in KB:	120
Sorting:	Variable	Weights:	No	# TC w/Modes:	45
Finish:	As-computed	KG Version:	Tally	Min # of EC:	10
Mode:	4-Together	KG Combo:	Add	Min % of EC:	10
		Attribute Format:	Both	Data Prefix:	DA

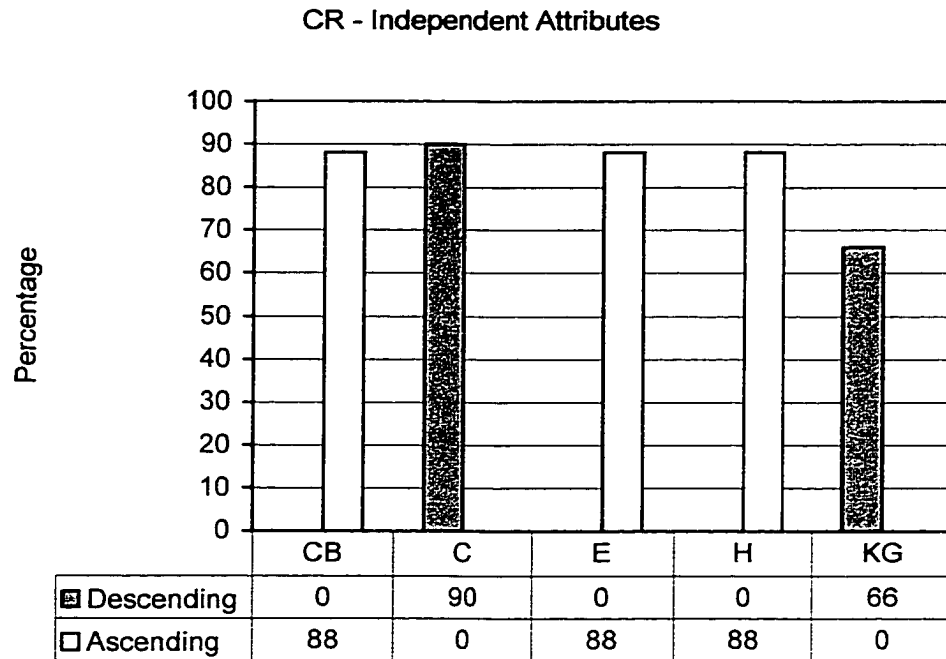


Figure 6.5: Descending v. Ascending for CR – Independent Attributes.

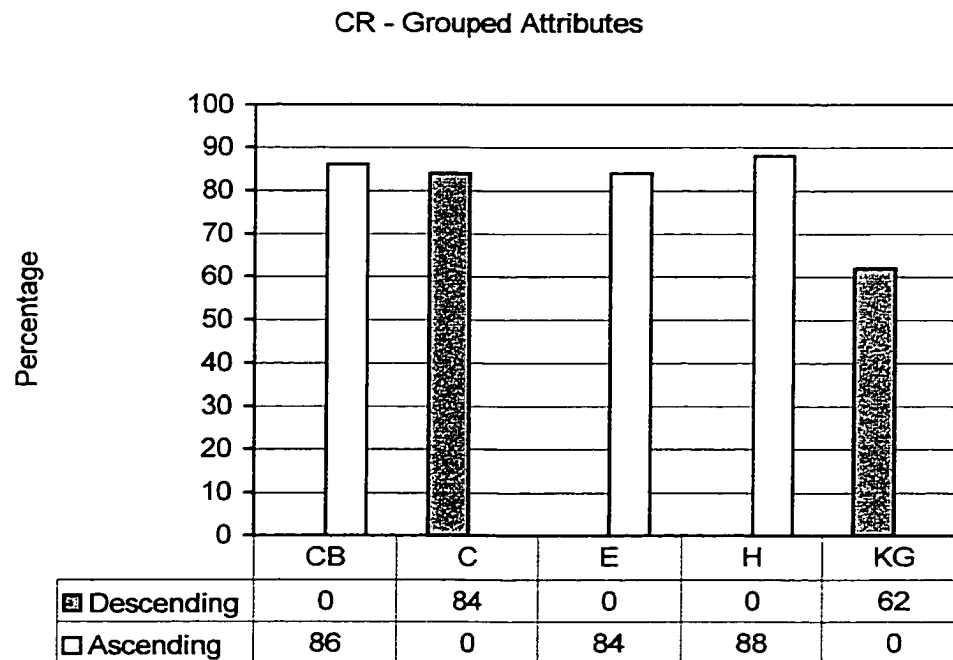


Figure 6.6: Descending v. Ascending for CR – Grouped Attributes.

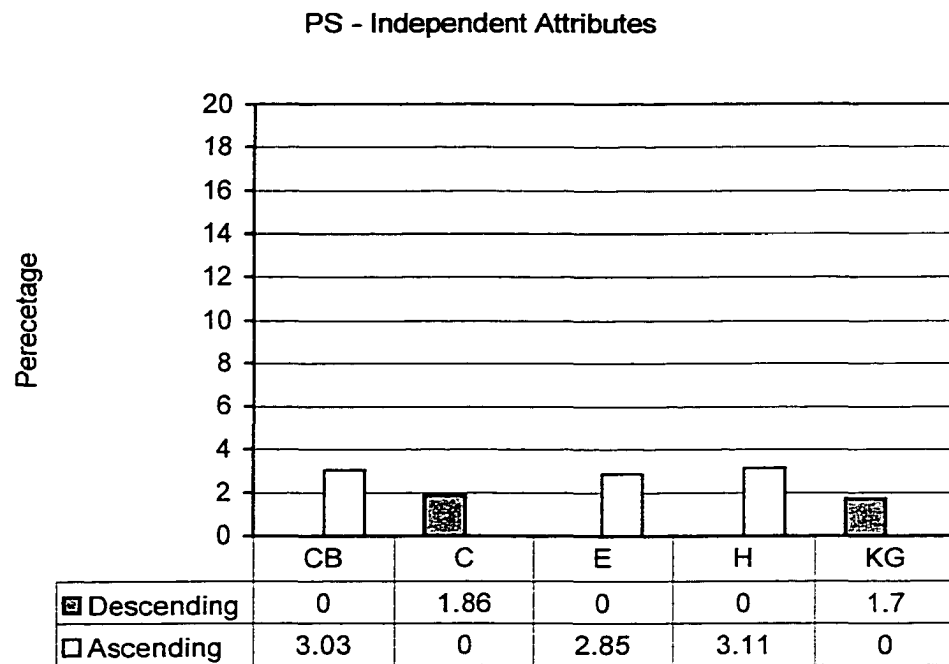


Figure 6.7: Descending v. Ascending for PS – Independent Attributes.

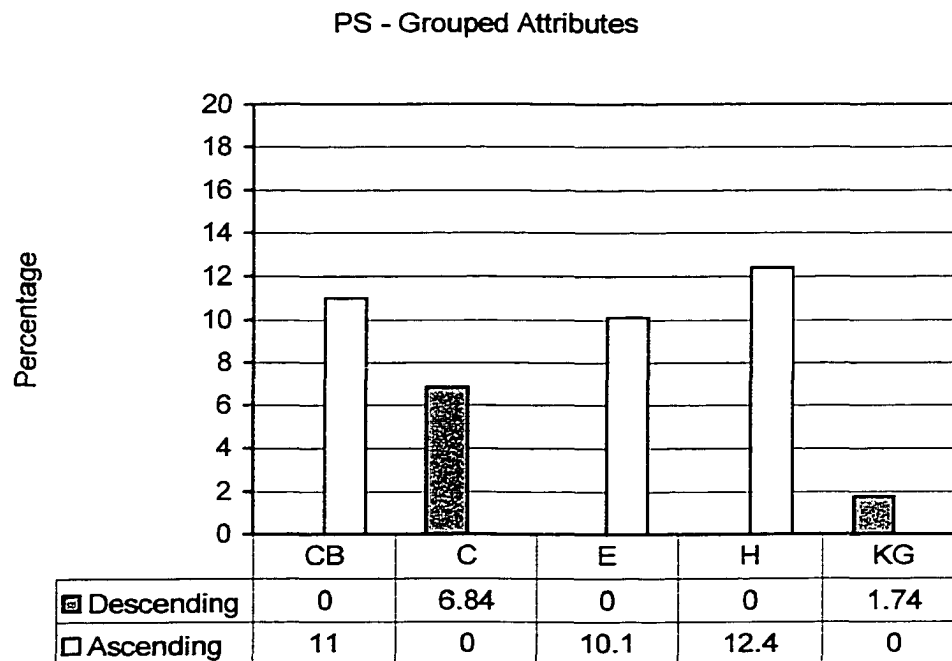


Figure 6.8: Descending v. Ascending for PS – Grouped Attributes.

Table 6.3: 6x15 TC Results for Distance/Similarity with Independent Attributes

	D/A	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	D	0	0	0	0	0	0
CB	A	100	73	87	4.0	2.9	3.4
C	D	93	73	87	2.4	1.9	2.2
C	A	0	0	0	0	0	0
E	D	0	0	0	0	0	0
E	A	93	73	84	3.5	2.8	2.2
H	D	0	0	0	0	0	0
H	A	100	87	92	4.3	3.6	3.9
KG	D	93	53	74	2.5	1.9	2.3
KG	A	0	0	0	0	0	0

Table 6.4: 6x15 TC Results for Distance/Similarity with Grouped Attributes

	D/A	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	D	0	0	0	0	0	0
CB	A	100	73	86	14	10	12
C	D	87	73	79	8.9	6.9	7.6
C	A	0	0	0	0	0	0
E	D	0	0	0	0	0	0
E	A	87	67	79	12	8.6	11
H	D	0	0	0	0	0	0
H	A	100	87	93	15	12	14
KG	D	73	47	64	2.4	1.6	2.2
KG	A	0	0	0	0	0	0

While gathering the data regarding the accuracy of matching the failure modes of the Test Cases, additional data was captured. The maximum value computed for the five metrics as they were used against the aIFAS knowledge base was recorded. Table 6.5 shows those results.

Table 6.5: The Maximum Computed Metric Values

	CB	C	E	H	KGS
Independent	15	1	3.9	15	11.2
Grouped	7.6	1	2.3	15	12.1

The common practice of using the relationship that [Similarity = 1 – Distance] would be an ineffective means of inverting the results (that is, ensuring that an ascending

list always has the best-match at the top). A better solution is to change the sort order for the metric to be in correspondence with whether it computes distance or similarity. The results of test combinations DA001 through DA020 confirm the sort orders to be used in subsequent analyses as they are presented in Table 6.6. (This may appear to be a trivial test, but it is included to avoid being a victim of a false assumption!)

Table 6.6: Sort Order Metrics Use for Selecting Matching Cases

Metric	Ascending Sort for Distance	Descending Sort for Similarity
City Block	X	
Cosine		X
Euclidean	X	
Hamming	X	
Knowledge Graph		X

6.2.2 Normalization

It is quite possible for a metric to use a different number of case attributes as it compares two different example cases against a test case. Similarly, two rival metrics could use different numbers of attributes as they compare an example case and a test case. In the extreme, a large number of poorly matching attributes could yield the same value for a metric as a few very good matches.

Intuitively, the normalizing process, of dividing the computed value by the number of attributes used to derive it, should level-the-playing-field when different cases or metrics are being compared. Just how effective that process is, was explored using the aIFAS set of candidate metrics and mode/attribute data structures.

Table 6.7 establishes the parameters used to obtain the data shown in Figures 6.9 through 6.12 for the 50 test case results and Tables 6.8 and 6.9 for the 6x15 test case results (see also MN021-MN040 in Appendix C).

Table 6.7: Parameter Settings for As-Computed/Normalized Metric

<i>Parameter Settings for As-Computed/Normalized Series</i>					
Metric:	ALL	Unused Attributes:	Include	# EC in KB:	120
Sorting:	Best Choice	Weights:	No	# TC w/Modes:	45
Finish:	Variable	KG Version:	Tally	Min # of EC:	10
Mode:	4-Together	KG Combo:	Add	Min % of EC:	10
		Attribute Format:	Both	Data Prefix:	MN

CR - Independent Attributes

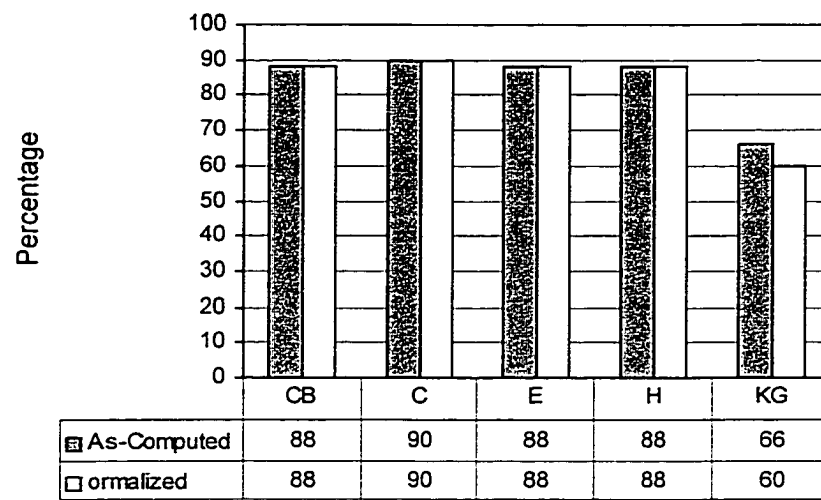


Figure 6.9: As-Computed v. Normalized for CR – Independent Attributes.

CR - Grouped Attributes

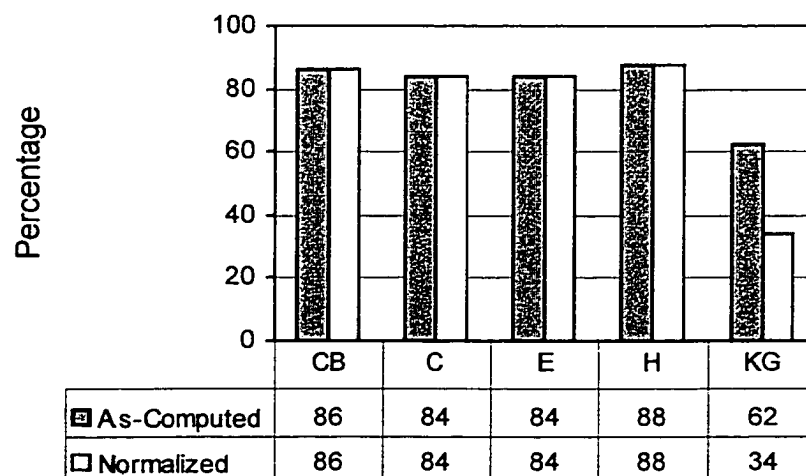


Figure 6.10: As-Computed v. Normalized for CR – Grouped Attributes.

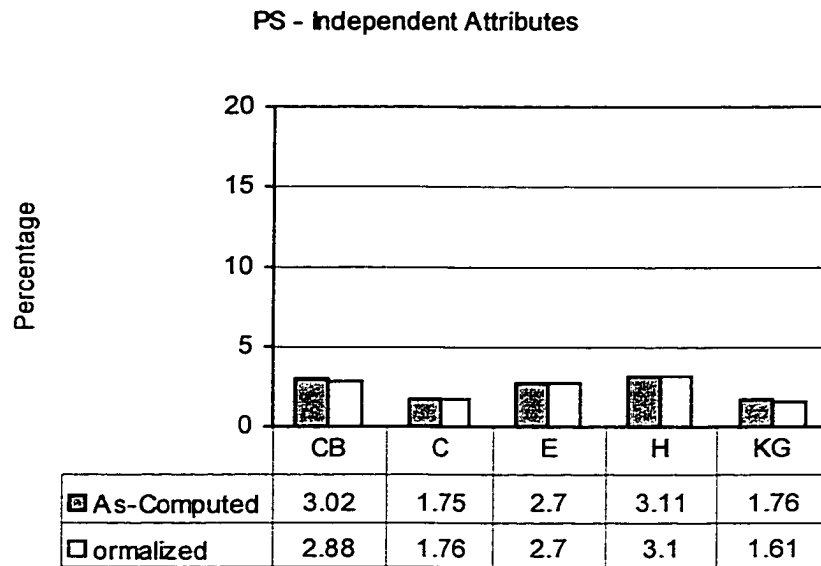


Figure 6.11: As-Computed v. Normalized for PS – Independent Attributes.

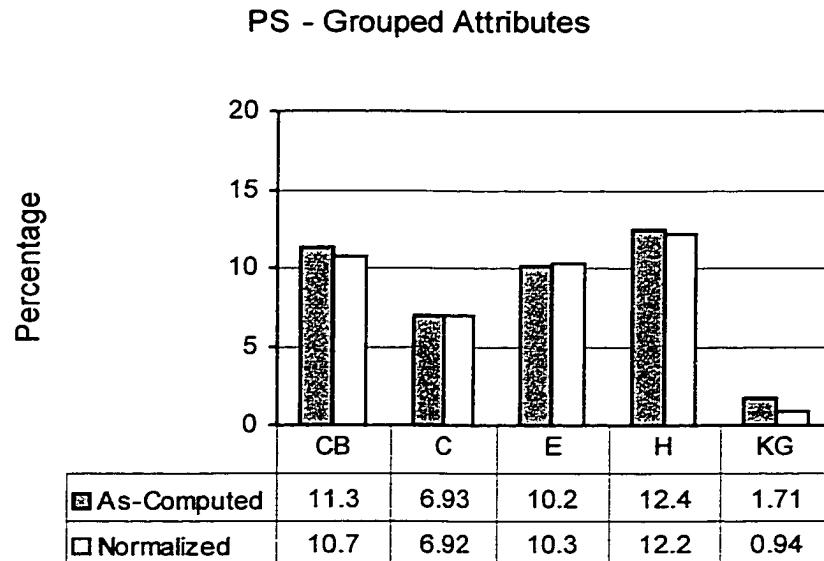


Figure 6.12: As-Computed v. Normalized for PS – Grouped Attributes.

The results of test combinations MN021 through MN040 indicate that there is no improvement in accuracy or performance to be gained by normalization of the computed results. For subsequent testing of aIFAS, the metric values will be used as-computed.

Table 6.8: 6x15 TC Results for As-Computed/Normalized with Independent Attributes

	As-Is/N	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	As-Is	100	80	90	4.0	3.2	3.6
CB	N	100	73	87	4.0	2.9	3.5
C	As-Is	93	73	87	2.5	1.8	2.2
C	N	93	80	87	2.4	2.0	2.2
E	As-Is	93	73	86	3.5	2.7	3.2
E	N	93	73	84	3.4	2.7	3.1
H	As-Is	93	87	91	4.0	3.5	3.8
H	N	100	87	92	4.2	3.5	3.8
KG	As-Is	93	53	74	2.5	1.8	2.2
KG	N	73	33	56	2.2	1.6	2.0

Table 6.9: 6x15 TC Results for As-computed/Normalized with Grouped Attributes

	As-Is/N	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	As-Is	93	80	89	14	11	12
CB	N	100	73	86	15	10	12
C	As-Is	87	73	79	8.8	6.9	7.8
C	N	87	67	78	8.7	6.8	7.6
E	As-Is	93	73	82	13	11	12
E	N	87	67	79	12	10	11
H	As-Is	93	80	88	14	11	13
H	N	100	87	93	14	12	13
KG	As-Is	73	47	64	2.4	1.6	2.2
KG	N	47	20	37	1.6	0.7	1.2

6.2.3 Extraneous Attributes

Only a small percentage of the total number of attributes available are used to characterize an individual failure case. With this particular set of failure mode and attribute set pairings, in the extreme case, only 16 attributes might be involved in the comparison of two cases. That leaves many unused attributes. The questions that arise are, “How computationally expensive is it to consider all attributes with each case comparison?” and “Are all of them necessary?” Connected with the answer for how to deal with the case of unused attributes is what response is appropriate to deal with unknown or missing attribute values.

Table 6.10 establishes the parameters used to obtain the data shown in Figures 6.13 through 6.16 for the 50 test case results and Tables 6.8 and 6.9 for the 6x15 test case results (see also IE041-IE058 in Appendix C).

Table 6.10: Parameter Settings for Include/Exclude Attributes

<i>Parameter Settings for Include/Exclude Series</i>					
Metric:	ALL	Unused Attributes:	Variable	# EC in KB:	120
Sorting:	Best Choice	Weights:	No	# TC w/Modes:	45
Finish:	As-Computed	KG Version:	Tally	Min # of EC:	10
Mode:	4-Together	KG Combo:	Add	Min % of EC:	10
		Attribute Format:	Both	Data Prefix:	IE

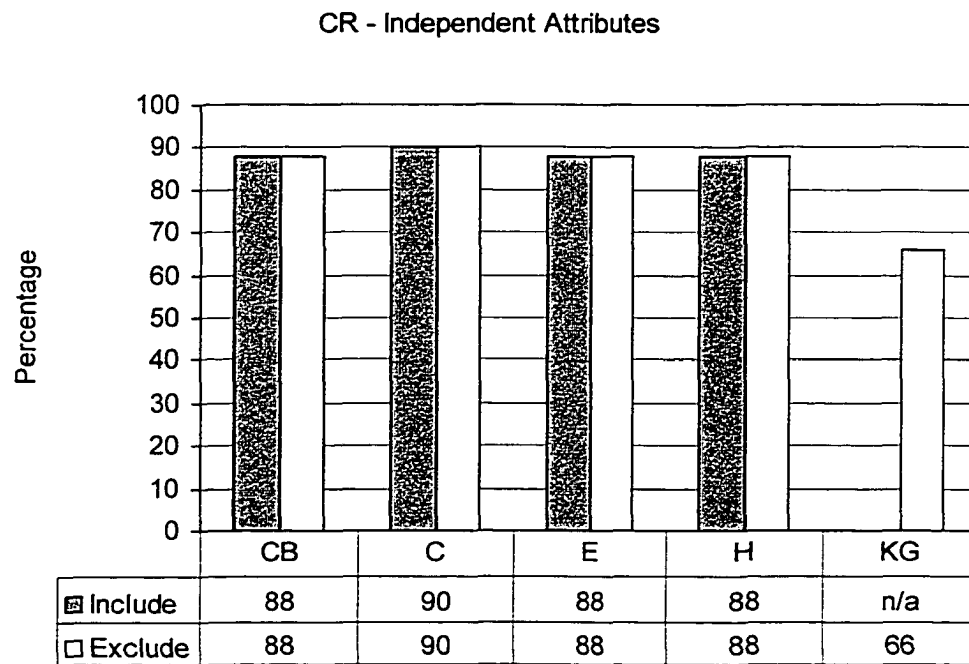


Figure 6.13: Include v. Exclude Attributes for CR – Independent Attributes.

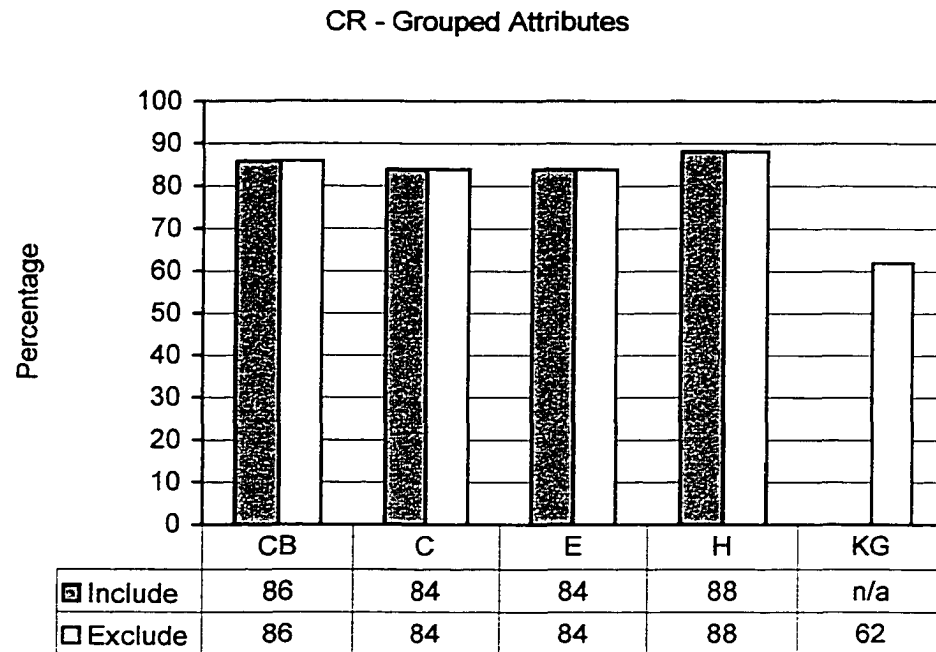


Figure 6.14: Include v. Exclude Attributes for CR – Grouped Attributes

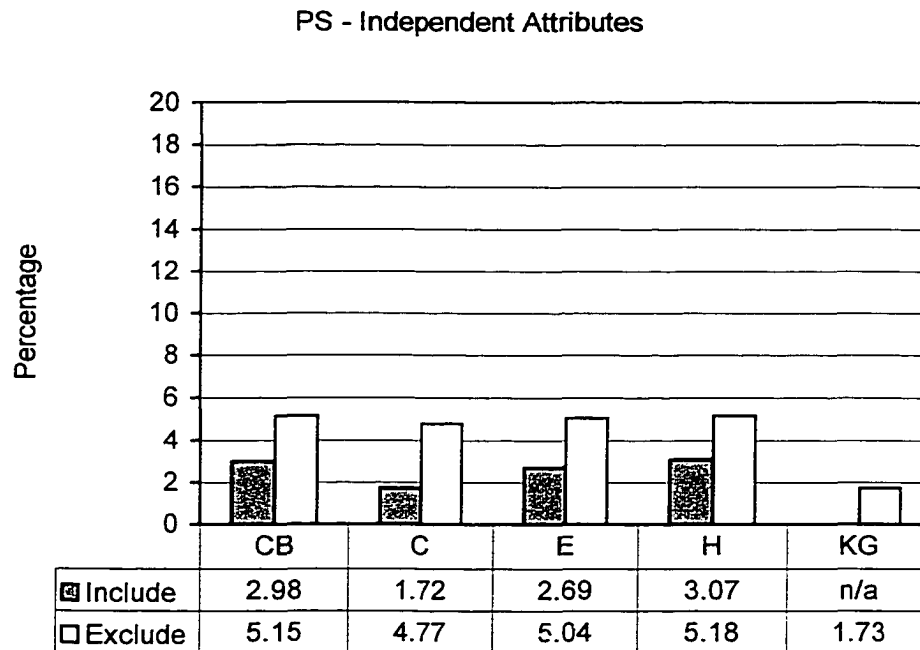


Figure 6.15: Include v. Exclude Attributes for PS – Independent Attributes.

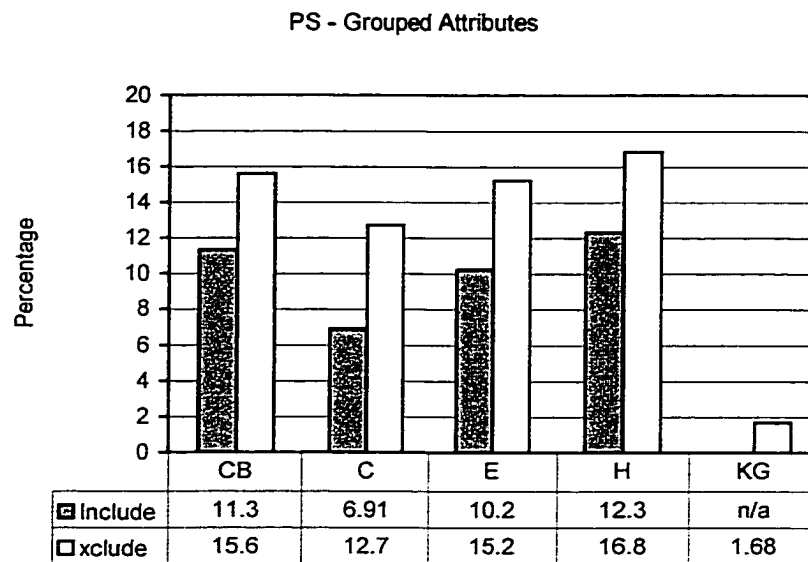


Figure 6.16: Include v. Exclude Attributes for PS – Grouped Attributes.

Table 6.11: 6x15 TC Results for Include/Exclude with Independent Attributes

	I/E	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	I	100	80	90	3.9	3.2	3.6
CB	E	100	73	87	7.0	5.3	6.2
C	I	93	73	87	2.4	1.8	2.2
C	E	93	80	87	6.3	5.0	5.6
E	I	93	73	86	3.4	2.8	3.2
E	E	93	73	84	6.8	5.2	6.0
H	I	93	87	91	3.9	3.4	3.7
H	E	100	87	92	7.2	6.1	6.6
KG	n/a						
KG	E	93	53	74	2.6	1.7	2.2

Table 6.12: 6x15 TC Results for Include/Exclude with Grouped Attributes

	I/E	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	I	93	80	89	13	11	12
CB	E	100	73	86	20	15	18
C	I	87	73	79	8.6	6.9	7.7
C	E	87	67	78	16	12	14
E	I	93	73	82	13	9.9	11
E	E	87	67	79	19	13	16
H	I	93	80	88	13	12	12
H	E	100	87	93	20	17	19
KG	n/a						
KG	E	73	47	64	2.5	1.5	2.2

The CR based results for test combinations IE041 through IE058 indicated that it made no difference whether the unused attributes were included in the calculations or not. The PS results, however, indicated a dramatic difference when unused attributes were excluded. There was nearly a 180% improvement in performance when the unused attributes were excluded while using the Cosine Correlation similarity with an independent attribute format. Of the City Block, Cosine Correlation, Euclidean, and Hamming metrics considered, all showed marked improvement when unused attributes were excluded for both independent or grouped attribute formats. (Note: By virtue of its mode of operation the KGS metric already excludes unused attributes. This is offered to explain the “n/a” entry in the data chart.)

The CR results illustrate that for the aIFAS knowledge base excluding unused attributes does not degrade the accuracy. The PS results show that including unused attributes, or those that convey no information, is expensive. This seems to infer two characteristics of useful attributes.

First, an attribute should convey information by its presence, not its absence. For instance, if an attribute indicates a crack exists, the results should not be expected to rely upon the absence of that attribute to confirm there is no cracking. Second, rather than devising schemes to estimate values for unknown or missing attributes, it makes more sense to simply exclude them. Alternatively, drop them from calculations until they can add to the problem solving process.

The results of test combinations IE041 through IE058 indicate that there is no degradation of accuracy and a tangible performance increase when unused attributes are excluded from consideration when performing metric calculations. For subsequent testing of aIFAS, the metric values will be computed with unused attributes excluded.

6.2.4 Combining KGS Terms

The original expression of the KGS metric adds the sensitivity and specificity terms to compute a similarity measure. This step of the evaluation compares the results derived from adding or multiplying the terms; as well as, using each of the terms individually. The choices can be expressed as,

$$\begin{aligned}
 KGS(X, Y) &= f(Se + Sp), & \text{where} & & X & \text{is a new case} \\
 & \text{or } f(Se \times Sp), & & & Y & \text{is an example case} \\
 & \text{or } f(Se), & & & Se & \text{the KGS Sensitivity term} \\
 & \text{or } f(Sp) & & & Sp & \text{the KGS Specificity term}
 \end{aligned}$$

Equation 6.1

This section also investigates using the various combinations of KGS terms as weighting factors for the other four metrics. The weighting is applied to all attributes. The value of the weight is the respective KGS term corresponding to the failure mode of the example case and the current attribute being compared. The weighting operation can be expressed as,

$$\text{Weighted Metric}(X, Y) = \sum_{i=1}^n g(M_y, i, Se, Sp) \times \text{Metric}(x_i, y_i)$$

Equation 6.2

where	X	is a new case	n	the number of attributes
	Y	is an example case	M _y	the example case failure mode
	Se	the KGS Sensitivity term	x _i	the new case's i th attribute
	Sp	the KGS Specificity term	y _i	the example case's i th attribute

The KGS metric must calculate a set of coefficient terms before it can compute a similarity. This set of terms must only be computed when the number of example cases, attribute data structure, or version of KGS being used changes. Nevertheless, there is a computational cost attached to performing that task. Table 6.13 shows the respective

times for generating sets of coefficients to be used in KGS computations. The data is reported using RTU times.

Table 6.13: Knowledge Graph Times for Coefficient Set Computation

Number of Stored Cases	Number of Failure Modes	Independent Attributes	Time When Independent	Grouped Attributes	Time When Grouped
120	47	94	870	32	230
240	61	105	2230	32	400
360	74	120	4320	32	850
480	80	126	5800	32	1240
600	83	131	5530	32	1610

This series of tests uses the original expression for the KGS metric. Because that is a simple tally of cases with or without certain failure modes or attributes, it is applied to both the independent and grouped forms of attribute representation. Table 6.14 establishes the parameters used to obtain the data shown in Figures 6.17 through 6.20 for the 50 test case results and Tables 6.15 and 6.16 for the 6x15 test case results (see also WT059-WT098 in Appendix C).

Table 6.14: Parameter Settings for KG Combination and Weighting

<i>Parameter Settings for KGS Combination/Weighting Series</i>					
Metric:	ALL	Unused Attributes:	Exclude	# EC in KB:	120
Sorting:	Best Choice	Weights:	Variable	# TC w/Modes:	45
Finish:	As-Computed	KG Version:	Tally	Min # of EC:	10
Mode:	4-Together	KG Combo:	Variable	Min % of EC:	10
		Attribute Format:	Both	Data Prefix:	WT

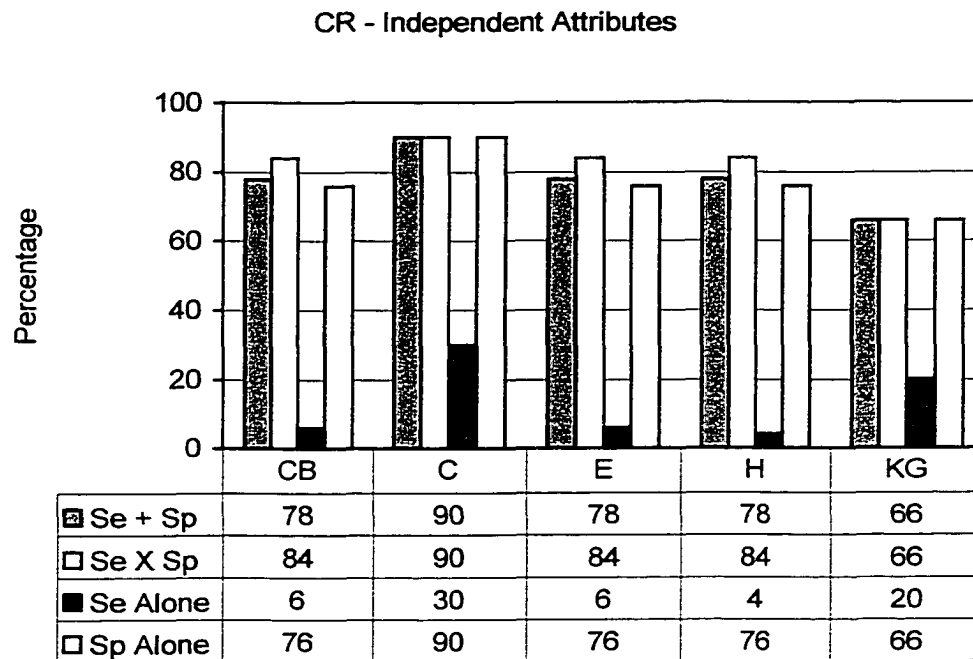


Figure 6.17: KGS Term Combinations/Weighting for CR – Independent Attributes.

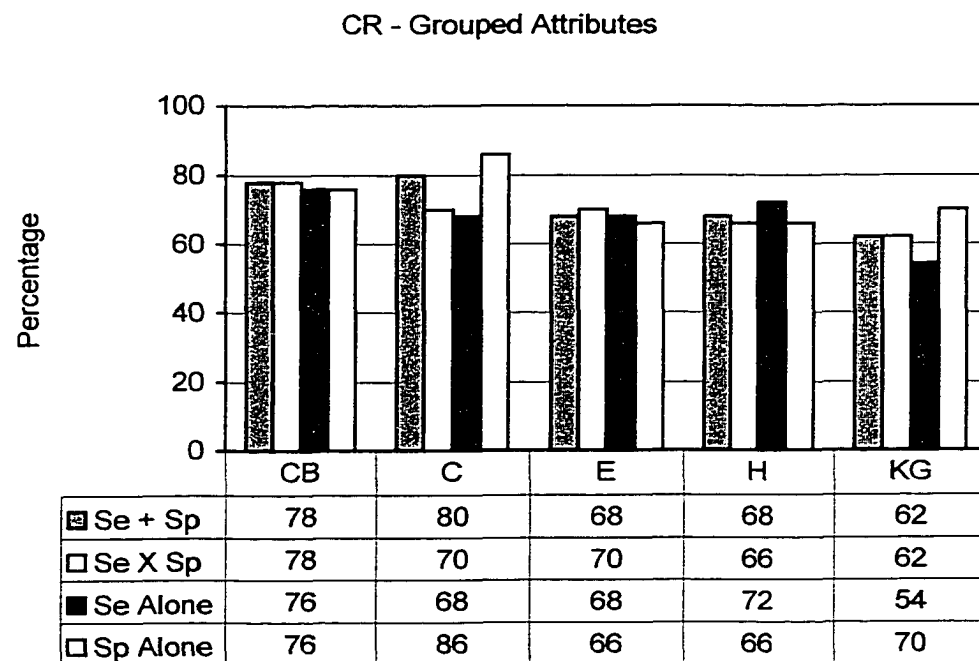


Figure 6.18: KGS Term Combinations/Weighting for CR – Grouped Attributes.

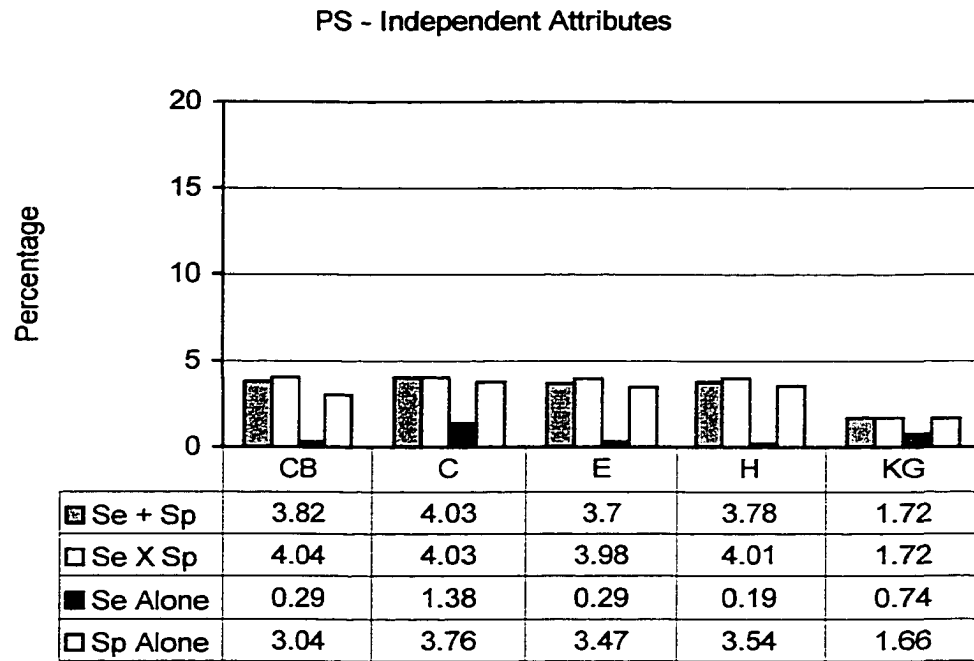


Figure 6.19: KGS Term Combinations/Weighting for PS – Independent Attributes.

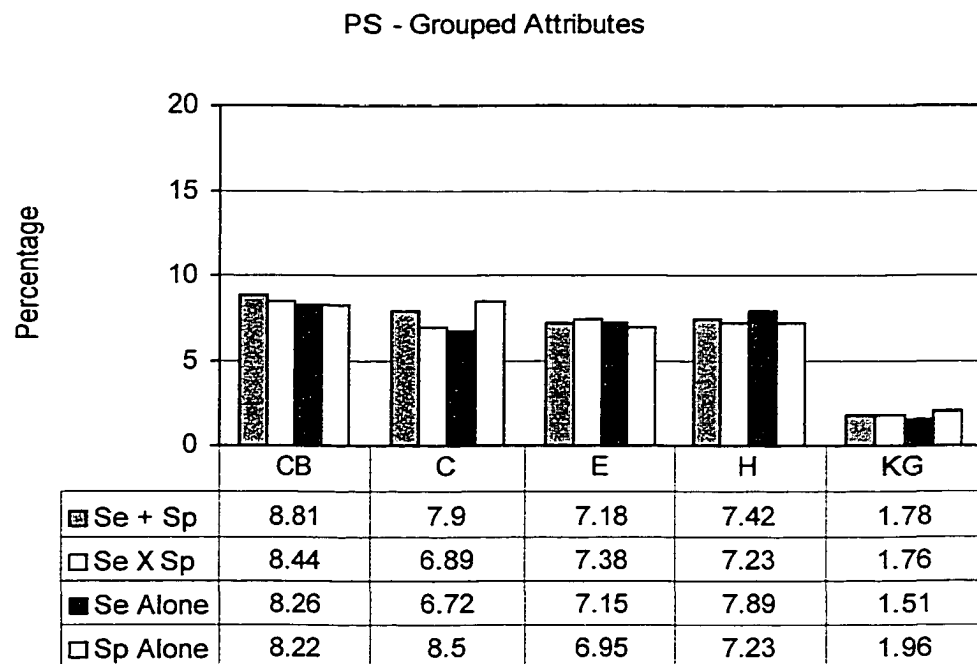


Figure 6.20: KGS Term Combinations/Weighting for PS – Grouped Attributes.

Table 6.15: 6x15 TC Results for KG Combination/Weights with Independent Attributes

	combo	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	Se +Sp	93	73	82	5.4	4.4	4.8
C	Se +Sp	100	73	88	5.6	4.2	4.9
E	Se +Sp	80	60	72	4.7	3.5	4.2
H	Se +Sp	87	67	77	5.2	3.9	4.5
KG	Se +Sp	73	53	62	2.3	2.0	2.2
CB	Se x Sp	93	80	88	5.5	4.7	5.1
C	Se x Sp	100	87	94	5.5	4.8	5.2
E	Se x Sp	100	67	87	5.9	4.1	5.1
H	Se x Sp	93	73	88	5.7	4.4	5.2
KG	Se x Sp	87	47	68	2.5	1.7	2.2
CB	Only Se	87	60	77	5.1	3.6	4.5
C	Only Se	100	73	88	5.5	4.2	5.0
E	Only Se	87	60	73	5.2	3.6	4.3
H	Only Se	80	60	73	4.9	3.5	4.4
KG	Only Se	73	47	66	2.5	2.1	2.3
CB	Only Sp	80	67	72	4.8	4.0	4.3
C	Only Sp	93	87	90	5.5	4.7	5.1
E	Only Sp	87	67	76	5.1	3.8	4.4
H	Only Sp	87	53	73	5.2	3.2	4.4
KG	Only Sp	87	47	70	2.4	1.8	2.1

The results of test combinations WT059 through WT098 indicated a slight degradation of accuracy. Most notably, the general performance suffered. Collectively, the grouped attribute data structure exhibited better accuracy and performance than did the independent attribute data structure. There is no clear evidence that the application of weighting factors of this sort was beneficial. The results for the KGS metric by itself were best for both data structures when the KGS Specificity term was used alone. The use of these weighting factors will not be used with aIFAS in routine operation. The best choice for the KGS metric thusfar is to use only the Specificity term in calculations.

Table 6.16: 6x15 TC Results for KG Combination and Weights with Grouped Attributes

	combo	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	Se +Sp	93	67	82	13	8.6	11
C	Se +Sp	100	67	82	11	8.3	9.7
E	Se +Sp	80	53	67	10	7.0	8.5
H	Se +Sp	73	53	67	9.6	7.5	8.9
KG	Se +Sp	87	47	62	3.1	1.6	2.2
CB	Se x Sp	87	67	76	11	8.8	10
C	Se x Sp	80	60	70	9.0	6.8	8.0
E	Se x Sp	87	53	73	11	7.2	9.6
H	Se x Sp	80	53	70	11	7.2	9.4
KG	Se x Sp	80	47	67	2.8	1.6	2.4
CB	Only Se	87	60	78	11	8.1	10
C	Only Se	67	53	60	7.3	6.5	7.0
E	Only Se	93	47	73	12	6.0	9.6
H	Only Se	87	60	72	11	7.8	9.6
KG	Only Se	73	53	59	2.6	1.7	2.0
CB	Only Sp	93	67	81	13	8.7	11
C	Only Sp	93	67	82	11	8.2	9.6
E	Only Sp	87	47	63	12	5.9	8.3
H	Only Sp	73	47	60	9.9	6.2	8.1
KG	Only Sp	87	47	62	2.8	1.5	2.0

6.2.5 Modified Knowledge Graph Method

A modification to the KGS methodology was suggested in Section 3.5.2.2 of the discussion on that metric. It was anticipated that it might be a vehicle for introducing other than binary attribute values into the method. Testing of that idea was conducted in a fashion parallel to the procedure outlined in Section 6.2.4, with two differences. First, the modified version of the Sensitivity term in Equation 3.9 is used. Second, only the grouped attribute data format is considered. (There is no difference between the two KGS method when binary data is involved.) Table 6.17 shows the parameters used to obtain the data for Figures 6.21 and 6.22 for the 50 test case results and Table 6.18 for the 6x15 test case results (see also WM099-WM118 in Appendix C).

Table 6.17: Parameter Settings for Modified KG Combination and Weighting

<i>Parameter Settings for Modified KGS Combination/Weighting Series</i>					
Metric:	ALL	Unused Attributes:	Exclude	# EC in KB:	120
Sorting:	Best Choice	Weights:	Variable	# TC w/Modes:	45
Finish:	As-Computed	KG Version:	Tally	Min # of EC:	10
Mode:	4-Together	KG Combo:	Variable	Min % of EC:	10
		Attribute Format:	Both	Data Prefix:	WM

CR - Grouped Attributes

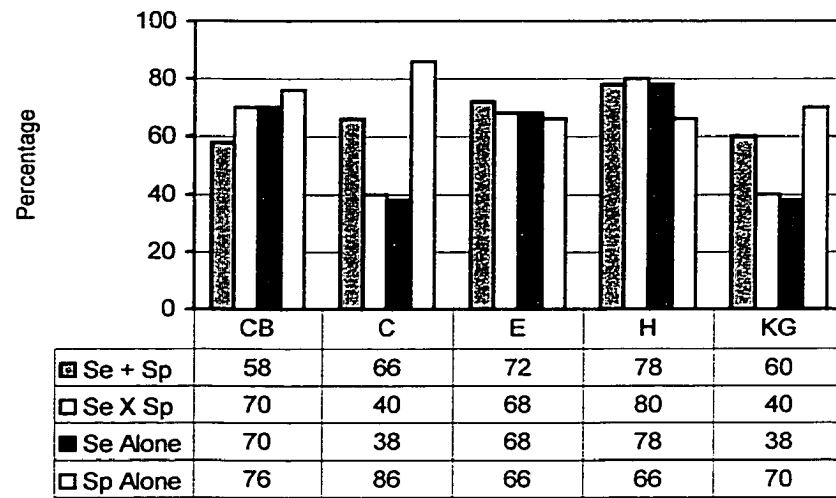


Figure 6.21: Modified KGS Term Combinations/Weighting for CR – Grouped Attributes.

PS - Grouped Attributes

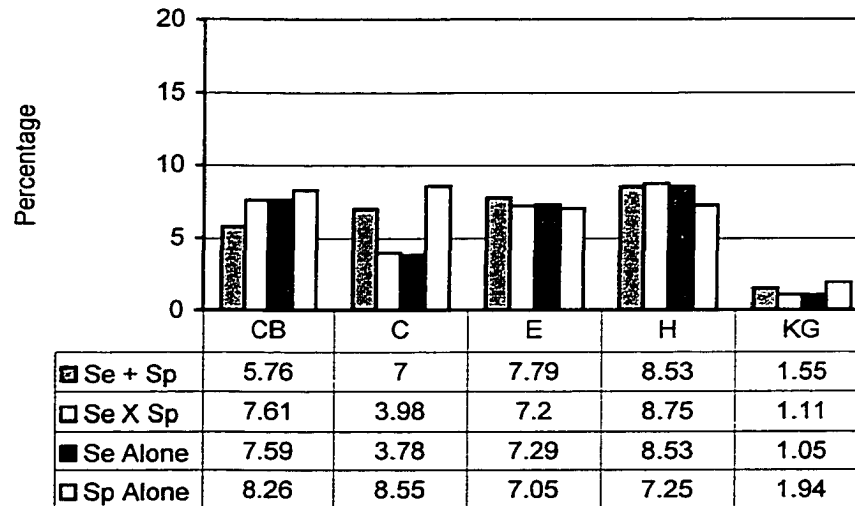


Figure 6.22: Modified KGS Combinations/Weighting for PS – Grouped Attributes.

Table 6.18: 6x15 TC Results for Modified KG Combo/Weights with Grouped Attributes

	combo	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	Se +Sp	80	60	69	11	7.9	9.3
C	Se +Sp	80	40	57	9.1	4.9	6.6
E	Se +Sp	67	53	61	8.6	6.8	7.8
H	Se +Sp	87	73	79	12	9.8	11
KG	Se +Sp	67	47	56	2.4	1.6	1.9
CB	Se x Sp	80	60	73	10	8.1	9.6
C	Se x Sp	67	13	44	8.0	1.5	5.2
E	Se x Sp	87	60	72	11	7.5	9.3
H	Se x Sp	93	73	86	13	10	12
KG	Se x Sp	60	27	43	2.0	0.9	1.5
CB	Only Se	80	60	68	11	7.7	9.2
C	Only Se	67	27	42	8.5	3.0	5.0
E	Only Se	80	60	68	10	7.8	8.8
H	Only Se	87	80	83	12	10	11
KG	Only Se	53	20	37	1.8	0.7	1.2
CB	Only Sp	87	67	73	11	8.6	9.6
C	Only Sp	93	80	84	11	8.8	9.6
E	Only Sp	73	47	58	10	6.3	7.6
H	Only Sp	80	47	66	11	6.3	8.8
KG	Only Sp	87	60	74	3.0	2.1	2.6

The results of test combinations WM099 through WT118 were similar to those generated by testing with the original KGS metric. Excepting the data for using the KGS Specificity term alone, a term by term comparison of the two sets of results yielded a general decrease in accuracy and performance. The results for using the KGS Specificity term alone were nearly identical for both the original and modified KGS method. As previously stated, the use of these sorts of weighting factors will not be used with aIFAS in routine operation. This data also reinforces the finding that the best choice for the KGS metric is to use only the Specificity term in similarity calculations.

6.2.6 Step and Prune Operation

Besides the grouped format for representing attributes in a more compact fashion, there is a hierarchical classification scheme in place. The process of failure analysis can be crudely divided into visual recording of prominent features, macroscopic study,

microscopic examination, and a collection of special tests that may or may not be performed dependent upon prior results. The attributes describing the stored cases are arranged in that general fashion.

Working within that secondary structure, a procedure was devised to simulate the progressive use of aIFAS during a failure investigation. Basically, a subset of the attributes are used to attempt a solution and subsequently reduce the size of the set of example cases. After a step has been taken, the example cases are ranked by their respective calculated metric value. The greater number from a specified threshold minimum count or minimum percentage of the example cases taken from that ordered list are kept. The general idea is to speed up solving the problem by selectively removing poor choices from future consideration. The procedure is repeated until all four subsets of attributes (prominent features, macroscopic, microscopic, and special features) have been applied.

Two sets of threshold limits were applied, 10% and 10-count minimum or 25% and 25-count minimum. Table 6.19 shows the parameters used to obtain the data for Figures 6.23 and 6.26 for the 50 test case results and Tables 6.20 and 6.21 for the 6x15 test case results (see also SP119-SP138 in Appendix C).

Table 6.19: Parameter Settings for Step & Prune Threshold

<i>Parameter Settings for Step & Prune Threshold Series</i>					
Metric:	ALL	Unused Attributes:	Exclude	# EC in KB:	120
Sorting:	Best Choice	Weights:	No	# TC w/Modes:	45
Finish:	As-Computed	KG Version:	Tally	Min # of EC:	10/25
Mode:	4-Together	KG Combo:	Specificity	Min % of EC:	10/25
		Attribute Format:	Both	Data Prefix:	SP

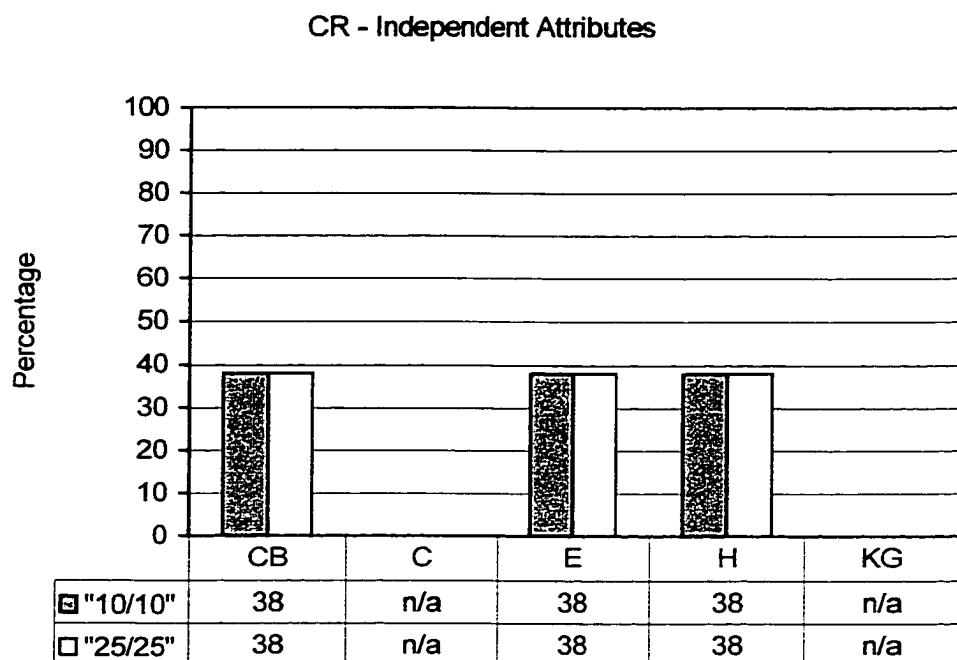


Figure 6.23: Threshold Testing of Step and Prune for CR – Independent Attributes.

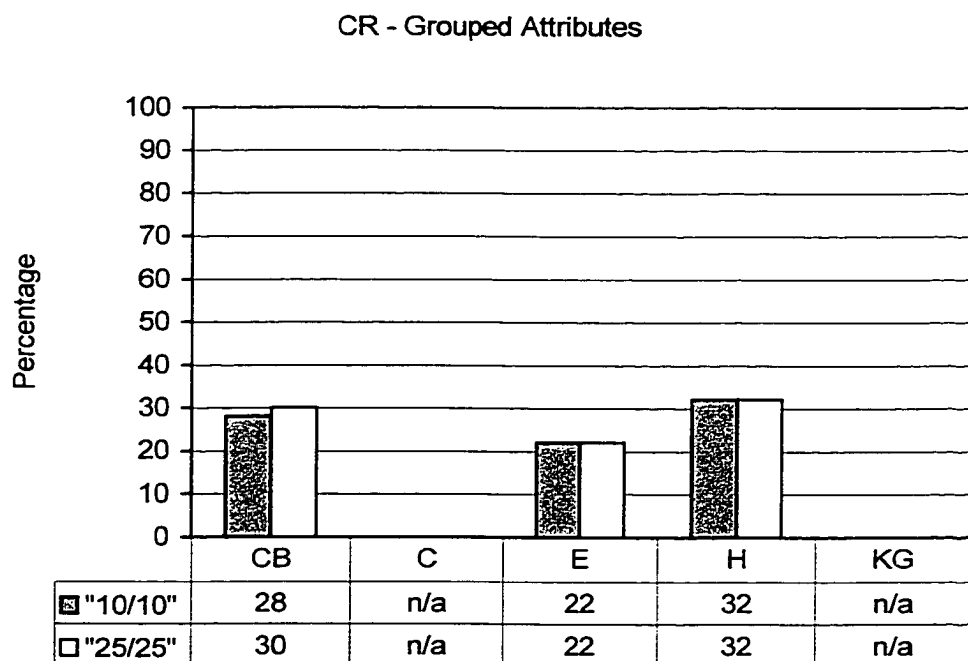


Figure 6.24: Threshold Testing of Step & Prune for CR – Grouped Attributes.

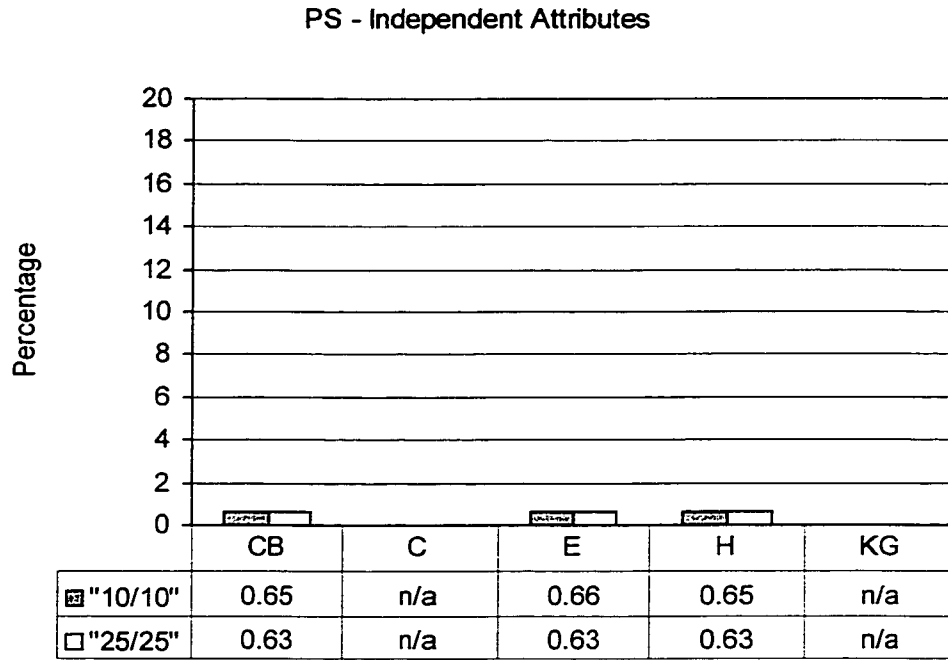


Figure 6.25: Threshold Testing of Step & Prune for PS – Independent Attributes.

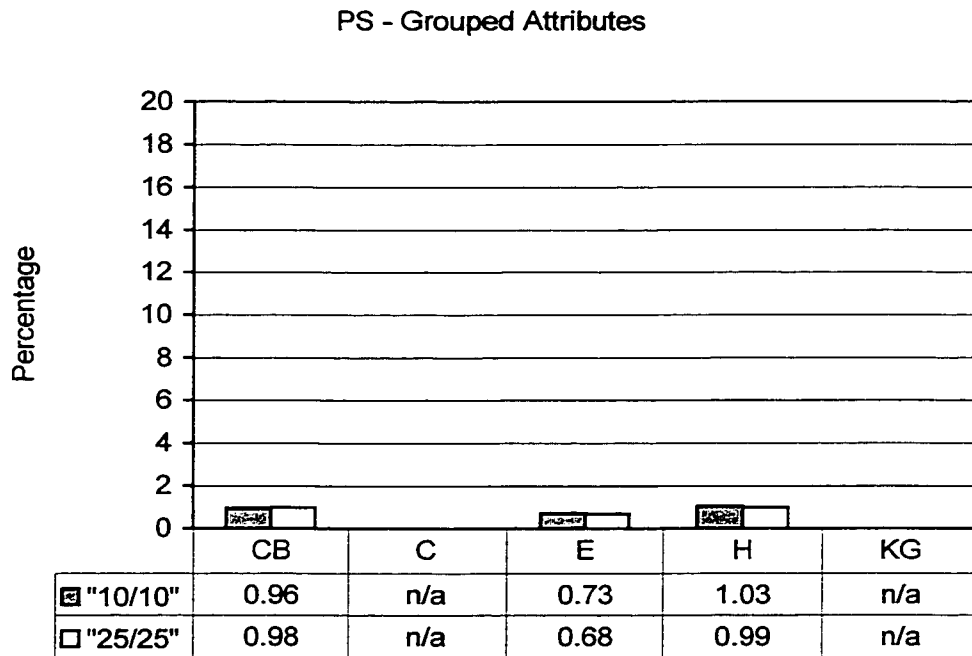


Figure 6.26: Threshold Testing of Step & Prune for PS – Grouped Attributes.

Table 6.20: 6x15 TC Results for 10%-10/25%-25 with Independent Attributes

	10/25	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	10	40	20	31	0.8	0.4	0.7
CB	25	33	20	29	0.7	0.4	0.6
C	10	0	0	0	---	---	---
C	25	0	0	0	---	---	---
E	10	47	33	41	1.0	0.7	0.9
E	25	67	20	36	1.3	0.4	0.7
H	10	47	27	39	1.0	0.6	0.8
H	25	47	13	32	1.0	0.3	0.6
KG	10	0	0	0	---	---	---
KG	25	0	0	0	---	---	---

Table 6.21: 6x15 TC Results for 10%-10/25%-25 with Grouped Attributes

	10/25	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	10	27	13	23	1.2	0.6	1.0
CB	25	27	13	20	1.1	0.6	0.8
C	10	0	0	0	---	---	---
C	25	0	0	0	---	---	---
E	10	33	13	27	1.4	0.6	1.1
E	25	33	6.7	22	1.5	0.3	1.0
H	10	40	20	34	1.6	0.8	1.4
H	25	40	13	28	1.6	0.5	1.1
KG	10	0	0	0	---	---	---
KG	25	0	0	0	---	---	---

Note: The “---” entries indicate that no results could be computed.

These results were not expected, but are quite understandable and explainable.

The step-and-prune process keeps a percentage of good matches and discards the rest of the stored cases in an attempt at streamlining and accelerating the selection process.

Because so few attributes are needed, and those are often well scattered throughout the data structure, the matching cases were being discarded prematurely. In an early step of the process if a case had no matching attribute to keep it in the pool of candidates, it was removed from later consideration. This sort of response influenced computations so strongly that the Cosine and KGS metrics were unable to successfully complete any comparison for any test case.

Interestingly, the results for test combinations SP119 through SP138 indicated no appreciable effect from changing the threshold limits. Understandably, the idea of using a step-and-prune mode of operation for aIFAS will not be pursued.

6.2.7 User Entry Simulation

There is another mode of stepwise operation to consider. When the User Entry option is selected, aIFAS forces the user to follow a sequential mode of data entry. The process closely resembles the fashion in which real-world failure investigations are conducted. A sequence of tests were contrived to demonstrate how aIFAS responds to the kind of problem solution scheme.

The set of test cases was solved in a four-step sequence. First, they could only access the Prominent Feature attributes. Next, both Prominent and Macroscopic Feature attributes were available. So on, until at last all four subsets of attributes could be used for case matching. These results should reflect the progressive improvement resulting as more and more information becomes available. Since this is a sequential sort of testing, it is reported with line charts rather than the bar charts which have been used previously. Table 6.22 shows the parameters used to obtain the data for Figures 6.27 and 6.30 for the 50 test case results and Tables 6.23 and 6.24 for the 6x15 test case results (see also US139-US178 in Appendix C).

Table 6.22: Parameter Settings for User Entry Simulation

<i>Parameter Settings for User Entry Simulation Series</i>					
Metric:	ALL	Unused Attributes:	Exclude	# EC in KB:	120
Sorting:	Best Choice	Weights:	No	# TC w/Modes:	45
Finish:	As-Computed	KG Version:	Tally	Min # of EC:	10
Mode:	1-by-1	KG Combo:	Specificity	Min % of EC:	10
		Attribute Format:	Both	Data Prefix:	US

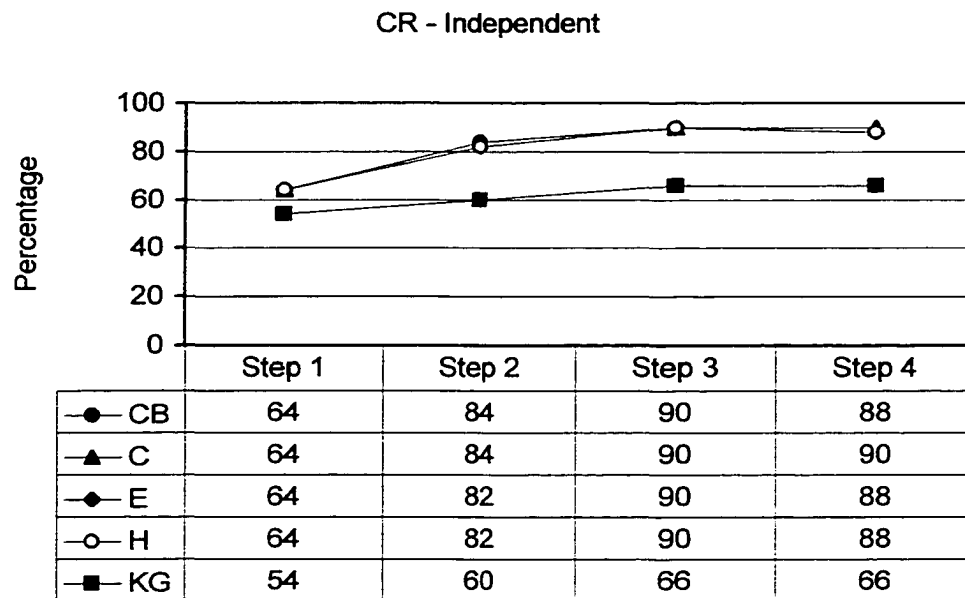


Figure 6.27: User Entry Simulation for CR – Independent Attributes.

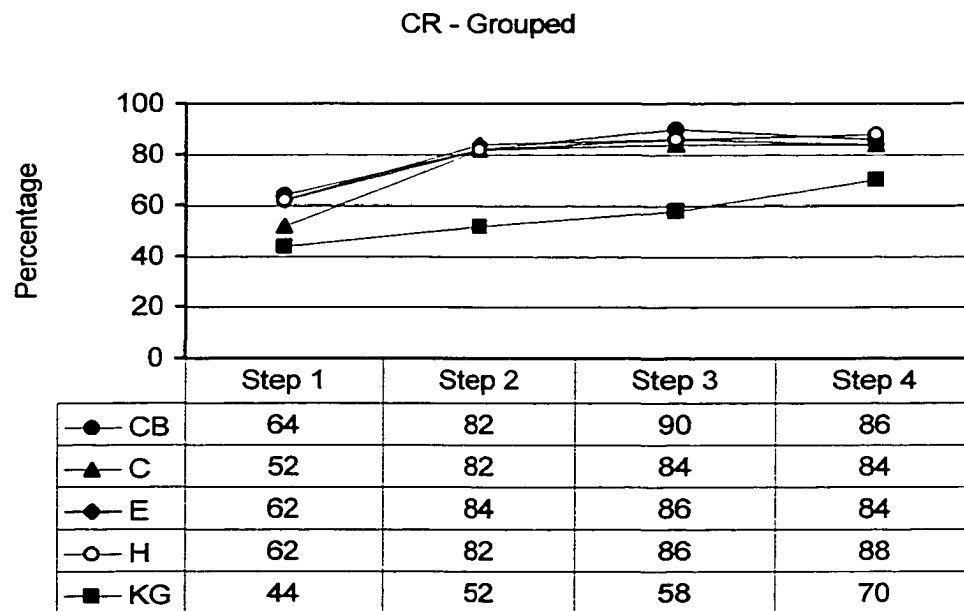


Figure 6.28: User Entry Simulation for CR- Grouped Attributes.

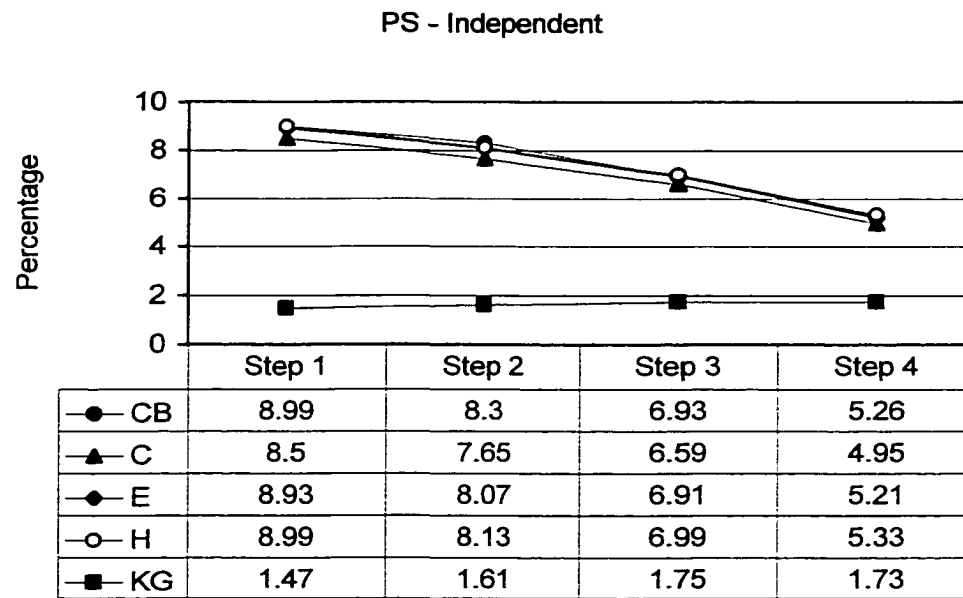


Figure 6.29: User Entry Simulation for PS – Independent Attributes.

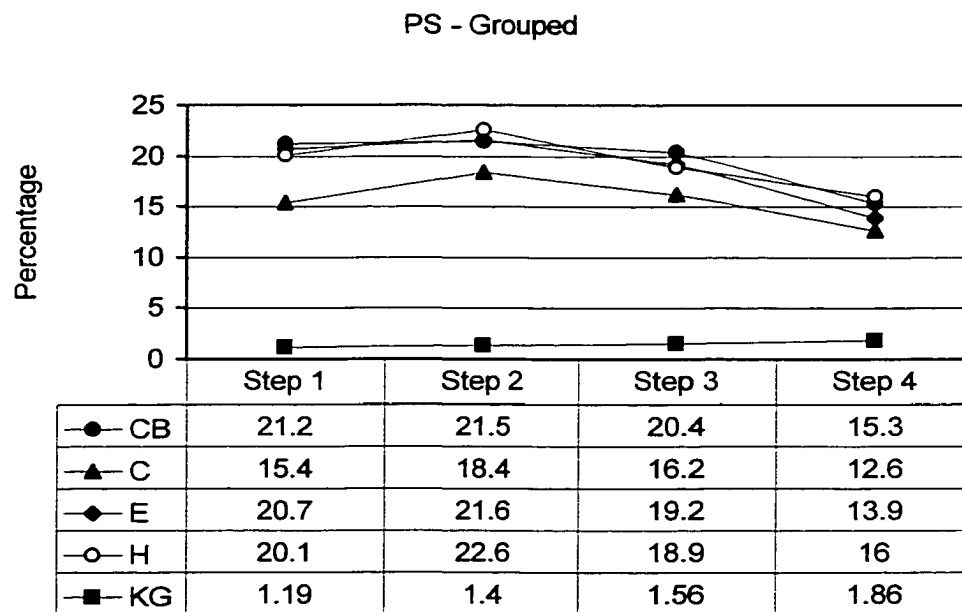


Figure 6.30: User Entry Simulation for CR- Grouped Attributes.

Table 6.23: 6x15 TC Results for User Entry Simulation with Independent Attributes

	Step	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	1	80	47	63	14	8.3	10
C	1	80	53	66	12	8.0	10
E	1	80	60	67	13	9.5	11
H	1	73	53	67	13	8.8	11
KG	1	73	40	53	2.5	1.3	1.9
CB	2	87	67	76	10	8.8	9.3
C	2	93	80	84	10	8.6	9.3
E	2	93	87	91	11	9.9	11
H	2	87	67	80	10	8.4	9.7
KG	2	73	40	59	2.4	1.7	2.1
CB	3	100	80	90	9.1	7.5	8.3
C	3	93	80	88	8.0	6.8	7.5
E	3	100	80	90	9.0	7.3	8.2
H	3	93	80	89	9.2	7.4	8.4
KG	3	80	53	69	2.5	2.2	2.3
CB	4	100	93	94	7.1	6.6	6.8
C	4	93	73	86	6.2	4.8	5.6
E	4	93	80	89	7.0	6.0	6.5
H	4	93	80	87	6.8	5.7	6.3
KG	4	87	53	68	2.5	1.8	2.1

The results for test combinations US139 through US178 illustrated, or perhaps proved, some characteristics of failure investigation. For some time, the assumption has been that roughly 85% of failure modes are determined by what the analyst sees. That would encompass the features found in the first two steps of the process. With the exception of the KGS metric, the average CR results for Step 2 were roughly 82%. If that number is adjusted for the fact that only 90% of the test cases had solutions, then the average Step 2 results are about 91%. That value meets the acceptance criteria that Section 5.3.2 established.

The declining values for the PS results is also understandable. The extra time required to utilize the little-bit-of-extra information offered by Special Features (i.e., nice to do, but not so important testing) is overshadowed by the computational (or, real-world

cost) of using it. Some failures can only be solved by such testing, most of them, however, behave like the CR results indicated.

Table 6.24: 6x15 TC Results for User Entry Simulation with Grouped Attributes

	Step	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	1	80	47	63	32	18	25
C	1	60	40	54	22	15	20
E	1	73	60	66	27	22	25
H	1	73	53	66	30	20	26
KG	1	47	27	39	1.6	0.9	1.4
CB	2	87	67	79	27	22	25
C	2	87	73	81	23	19	21
E	2	93	87	91	30	24	28
H	2	87	67	80	26	21	24
KG	2	67	33	52	2.4	1.1	1.8
CB	3	100	80	90	26	21	23
C	3	93	80	84	19	16	18
E	3	93	80	88	26	20	23
H	3	93	73	85	25	20	23
KG	3	87	40	57	2.9	1.4	1.9
CB	4	100	87	93	21	18	20
C	4	87	67	81	17	12	14
E	4	93	73	83	18	15	17
H	4	93	80	86	20	16	18
KG	4	87	47	62	2.8	1.6	2.1

6.2.8 Incremental Learning

Testing to this point has used a 120 member set of example cases that contained solutions for 90% of the test cases. The numbers in Table 6.25 show the number of test cases that have corresponding solutions in the aIFAS knowledge for each step in the series of incremental learning tests.

Table 6.25: Example Cases and Available Test Case Solutions

# Example Cases	30	60	90	120	180	240	300	360	480	600
# Test Cases w/solutions	34	39	44	45	48	49	50	50	50	50

This exercise demonstrates that the system can learn. Table 6.26 shows the parameters used to get data for Figures 6.31 to 6.34 for the 50 test case results and Tables 6.27 through 6.36 for the 6x15 test case results (see also IL179-IL278 in Appendix C).

Table 6.26: Parameter Settings for Incremental Learning

<i>Parameter Settings for Incremental Learning Series</i>					
Metric:	ALL	Unused Attributes:	Exclude	# EC in KB:	30-600
Sorting:	Best Choice	Weights:	No	# TC w/Modes:	34-50
Finish:	As-Computed	KG Version:	Tally	Min # of EC:	10
Mode:	1-by-1	KG Combo:	Specificity	Min % of EC:	10
		Attribute Format:	Both	Data Prefix:	US

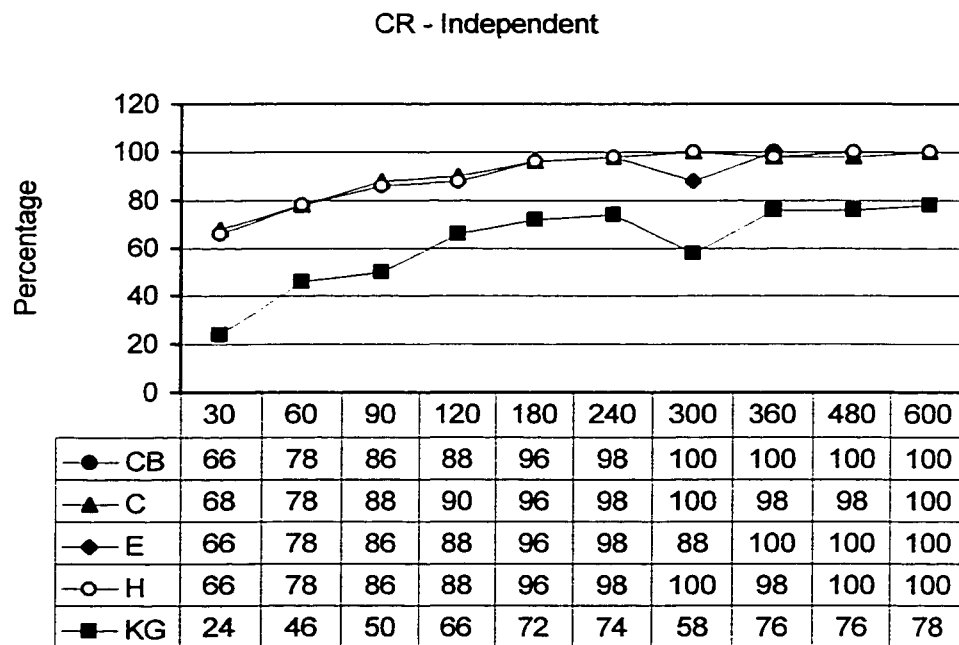


Figure 6.31: Incremental Learning for CR – Independent Attributes

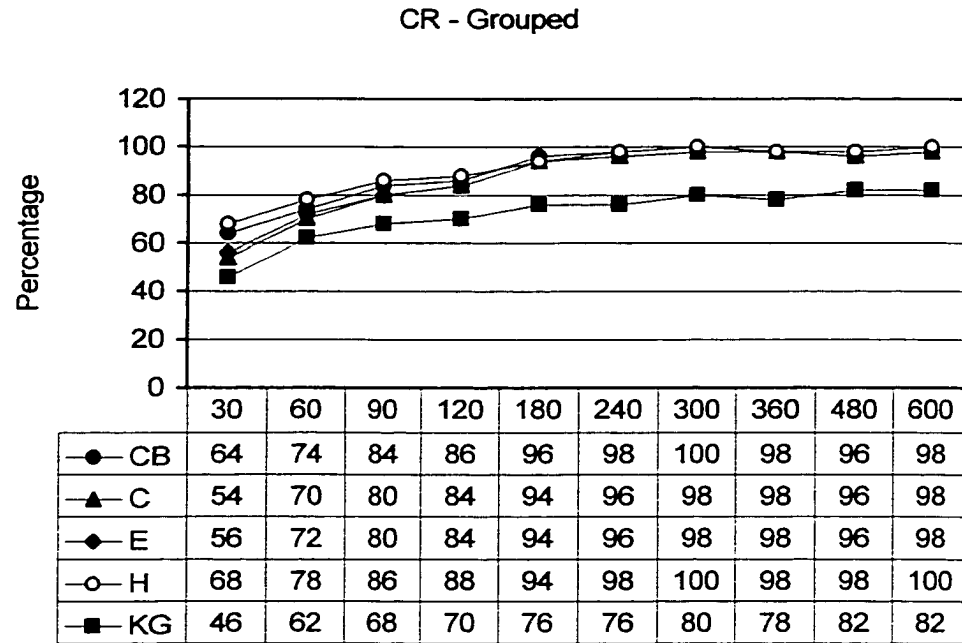


Figure 6.32: Incremental Learning for CR – Grouped Attributes.

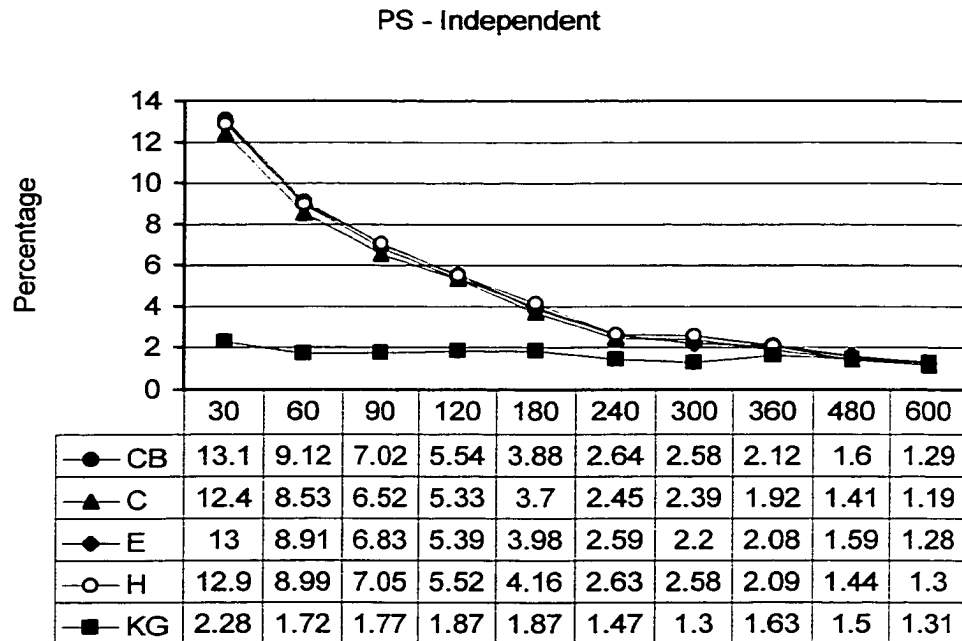


Figure 6.33: Incremental Learning for PS – Independent.

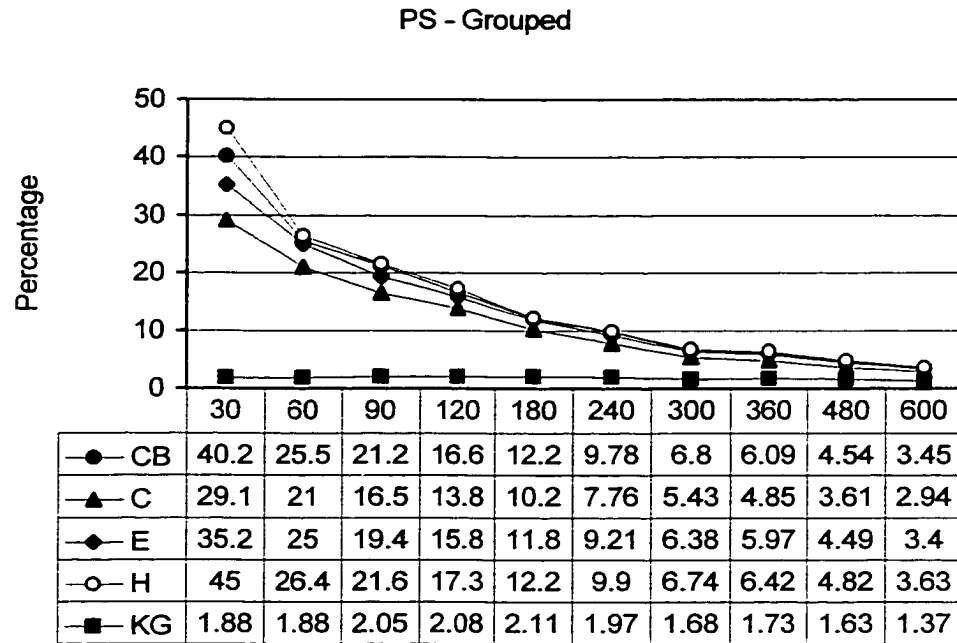


Figure 6.34: Incremental Learning for PS – Grouped Attributes.

Each of the metrics reported on in the test combinations from IL179 through IL278 improved its accuracy as cases were added to the example case set. While accuracy was on the increase, so was the time to arrive at a solution. The net effect was that in the vicinity of 200 example cases the CR results started to plateau and PS results appeared to be approaching an asymptotic limit. The results suggest that there is a relatively small, optimum set of example cases that would be sufficient to make consistent matches above the 90% level of matching accuracy.

The KGS metric, although not a strong performer in accuracy or speed, did show a curious trait. In the sequence of incremental learning episodes, the KGS metric managed to balance improved accuracy with slower speed to maintain a relatively constant PS result.

Table 6.27: 6x15 TC Results for 600 Example Cases

	I/G	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	I	100	100	100	1.6	1.5	1.5
C	I	100	100	100	1.5	1.4	1.4
E	I	100	100	100	1.6	1.5	1.5
H	I	100	100	100	1.6	1.5	1.6
KG	I	80	67	73	1.9	1.5	1.7
CB	G	100	93	99	4.2	3.9	4.1
C	G	100	93	98	3.5	3.3	3.4
E	G	100	93	98	4.1	3.8	3.9
H	G	100	100	100	4.3	4.1	4.2
KG	G	93	80	86	2.0	1.7	1.8

Table 6.28: 6x15 TC Results for 480 Example Cases

	I/G	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	I	100	100	100	2.0	1.8	1.9
C	I	100	93	99	1.8	1.6	1.7
E	I	100	100	100	1.9	1.8	1.9
H	I	100	100	100	2.0	1.8	1.8
KG	I	80	67	72	1.9	1.7	1.8
CB	G	100	87	97	6.2	5.0	5.7
C	G	100	93	98	4.6	4.0	4.2
E	G	100	87	96	6.0	4.8	5.5
H	G	100	93	99	6.5	5.6	5.9
KG	G	100	80	86	2.4	1.8	2.0

Table 6.29: 6x15 TC Results for 360 Example Cases

	I/G	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	I	100	93	99	4.0	3.7	3.9
C	I	100	93	99	3.7	3.4	3.6
E	I	100	100	100	3.9	3.8	3.8
H	I	100	100	100	4.0	3.8	3.9
KG	I	80	67	72	3.1	3.0	3.0
CB	G	100	93	98	11	10	11
C	G	100	100	100	9.6	8.9	9.2
E	G	100	93	98	11	9.9	10
H	G	100	93	99	12	10	11
KG	G	47	20	34	1.9	1.0	1.5

Table 6.30: 6x15 TC Results for 300 Example Cases

	I/G	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	I	100	100	100	3.2	3.0	3.1
C	I	100	100	100	2.9	2.7	2.7
E	I	100	100	100	3.2	3.0	3.1
H	I	100	100	100	3.5	3.0	3.2
KG	I	93	67	82	2.2	1.9	2.0
CB	G	100	100	100	9.2	8.3	8.7
C	G	100	93	96	7.6	6.3	7.1
E	G	100	93	98	8.7	7.8	8.2
H	G	100	100	100	9.1	8.5	8.8
KG	G	93	73	82	2.6	2.0	2.4

Table 6.31: 6x15 TC Results for 240 Example Cases

	I/G	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	I	100	93	98	3.8	3.4	3.6
C	I	100	93	98	3.5	3.2	3.4
E	I	100	100	100	3.8	3.6	3.7
H	I	100	93	97	3.7	3.5	3.6
KG	I	80	60	70	2.2	1.9	2.1
CB	G	100	93	97	11	9.8	10
C	G	100	87	94	8.7	7.4	8.1
E	G	100	93	97	11	9.8	10
H	G	100	93	98	11	10	10
KG	G	53	27	40	1.5	0.9	1.2

Table 6.32: 6x15 TC Results for 180 Example Cases

	I/G	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	I	100	93	97	5.5	4.6	5.0
C	I	100	87	96	4.9	4.1	4.6
E	I	100	93	98	5.1	4.5	4.8
H	I	100	87	94	5.1	4.4	4.7
KG	I	80	60	69	2.4	2.1	2.3
CB	G	100	87	94	15	13	14
C	G	100	93	97	12	10	11
E	G	100	87	93	14	12	13
H	G	100	87	93	14	12	13
KG	G	87	73	79	2.7	2.4	2.5

Table 6.33: 6x15 TC Results for 120 Example Cases

	I/G	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	I	100	80	90	7.2	5.7	6.6
C	I	100	73	88	6.6	4.9	5.8
E	I	93	73	84	7.2	5.4	6.2
H	I	93	73	84	6.6	5.4	6.0
KG	I	73	53	62	2.4	1.8	2.2
CB	G	87	67	81	19	13	17
C	G	93	80	87	16	15	16
E	G	93	80	90	20	15	18
H	G	100	80	89	23	17	19
KG	G	80	73	75	2.8	2.4	2.6

Table 6.34: 6x15 TC Results for 90 Example Cases

	I/G	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	I	100	73	88	9.5	7.1	8.5
C	I	100	73	88	8.8	6.4	7.9
E	I	87	73	80	8.9	6.9	7.7
H	I	93	73	82	8.8	7.0	7.9
KG	I	60	33	47	2.4	1.8	2.1
CB	G	87	60	78	24	16	21
C	G	93	73	84	23	17	20
E	G	93	80	90	25	20	24
H	G	100	67	86	27	17	23
KG	G	80	67	75	2.9	2.4	2.7

Table 6.35: 6x15 TC Results for 60 Example Cases

	I/G	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	I	87	60	76	12	8.3	10
C	I	93	60	80	12	7.5	10
E	I	87	67	75	12	9.0	10
H	I	93	67	80	13	9.0	11
KG	I	60	33	47	2.6	1.8	2.2
CB	G	87	47	64	32	18	26
C	G	87	60	78	28	18	25
E	G	87	73	79	36	26	31
H	G	100	47	79	40	20	32
KG	G	80	53	67	2.8	2.2	2.5

Table 6.36: 6x15 TC Results for 30 Example Cases

	I/G	maxCR	minCR	avgCR	maxPS	minPS	avgPS
CB	I	80	53	67	29	20	24
C	I	87	47	72	29	17	25
E	I	73	53	61	28	20	23
H	I	80	53	68	30	19	25
KG	I	33	20	24	5.7	4.0	4.6
CB	G	87	27	54	98	34	62
C	G	73	40	62	71	39	60
E	G	87	47	60	85	55	67
H	G	80	40	67	98	48	77
KG	G	53	33	42	5.4	3.3	3.9

7. CONCLUSIONS

In the course of this work, numerous occasions arose that suggested potential improvement or new avenues of exploration. Several of those were incorporated at the suggestion of individuals. As the results began to assembly themselves and were subject to review, their interpretation provided a few insights. They have been captured here for the reader's consideration. Besides the research effort, comments regarding the aIFAS prototype, its performance and future, are offered. Finally, since this has already been identified as a work in progress, some speculations for the future are made.

7.1 Discussion

The research results yielded both expected and unexpected information. Overall, the work represents a step forward in the development of aIFAS. The salient points are summarized in the following.

- When implemented, the aIFAS knowledge base will present a powerful research tool. The information content of a set of example cases from a single source and a moderate, but still limited, set of client types was revealing.
- Normalization of computations to account for differences in the number of elements involved in a calculation does not always offer improvement. The worth of the operation should be verified. In the metrics being considered, normalization not only added to the computational load, but either degraded or did nothing to improve performance.
- Including unused attributes in calculations imposes an undue burden on the computing resources. There is no value-added by doing so, only expense. An extension of this result is a consideration for coping with unknown or missing

attributes. An attribute that conveys no information should be excluded until it can contribute to the problem resolution.

- Grouping related attributes generally yields a significant improvement in performance. This compact format for attribute representation should allow expansion of the attribute set with minimal impact on the system. This finding is also indicative that a hierarchical data structure can be made to work in aIFAS.
- Weighting factors showed meager improvement in matching accuracy. That gain was overshadowed by the extra computational time necessary to apply them universally. It may be feasible to use weighting factors selectively with critical or highly sensitive attributes.
- Stepwise performance in aIFAS will necessarily mean the sequential addition of attribute information. Attempting to prune away poor matches based upon partial information is too likely to discard the case that provides a solution.
- Incremental learning does improve the results. It does so at the expense of rapidly increasing computational time. It appears that an optimum set of example cases can be found using a combination of the Correctness Ratio and Performance Score measuring tools. A first guess is that it would be in the neighborhood of twice the number of failure modes represented.
- The KGS method proved to be more an intellectual curiosity than a practical tool for aIFAS. It is quite likely very successful with complex cases described with a large number of attributes. The sort of cases that might be differentiated only by subtle variations. The method just cannot compete with the simpler metrics when using a case set such as aIFAS possesses.

- Two metrics were decided upon for future development. First, the Hamming Distance because it consistently yielded the best performance. Second, the City Block Distance certainly because it gave good performance, but more especially because it would allow easy introduction of other than binary attribute values.

7.2 The aIFAS Prototype

It was not the initial intent to implement the full aIFAS concept. Rather, a skeleton database was to be assembled that would be sufficient to support the intended research. As circumstances would have it, combined with a desire to automate as much of the analysis/data-gathering process as possible, the aIFAS prototype came into being. The system supports, or can easily be modified to support, all of the functionality discussed in this paper. Using the system to accumulate the research data has clarified two aspects of its utility. It certainly can support failure analyses, however, it is also a viable tool for the investigation of other sort of knowledge domains.

7.2.1 Using aIFAS for Failure Analysis

The portion of the prototype that provided user information regarding the database and guided a failure investigation characterizes the goal of the project. The experienced user has access to knowledge base statistics to recall past work, or can enter attributes of a new case as they become available to aid in an investigation. The novice user can learn about failure modes and attributes, can read actual reports dealing with those issues, and can test hypothetical cases interactively. The aIFAS prototype does not represent a complete work, but it does indicate the real possibilities of the concept.

7.2.2 Using aIFAS with Other Knowledge Domains

The parametric study portion of the aIFAS prototype is not limited to just the failure analysis knowledge domain. The use of a modular architecture was intentional. With minor changes, the system could easily accept the case data from other knowledge domains. Even the suite of comparison metrics could be modified to accept more, or different types, of algorithms. As such, aIFAS presents a powerful tool for matching comparison metrics to knowledge base representations, or gaining a deeper understanding of the interrelationships of knowledge base size/structure with the comparison metric.

7.3 Recommendation for Future Work

There are several recognized (and certainly some not yet realized) issues that warrant additional investigation. Among them are the following:

- Expansion of the aIFAS knowledge base is a certain member of this list. There is a need to explore and grow beyond a single source. It is equally important to seek out more mundane records of failures (i.e. maintenance records from industry or shop records from service companies). This latter point is important because the failures that find their way into commercial laboratories represent only a small fraction of the total number. Only the special, difficult cases are sent out for study.
- Explore a better scheme for grouping attributes and assigning them values within those groups. It may be that some hybrid construct of critical independent attributes and lesser used grouped attributes will emerge. This would also be the most likely spot for introducing fuzzy attribute values.
- Weighting factors for emphasizing or diminishing the role of attributes may still have a use. It should prove worth the effort to identify the importance of

individual attributes. With that knowledge, weighting factors could be applied where they would produce the most benefit.

- It would be informative to conduct a parametric study to find an optimum size and mixture of example cases for the aIFAS knowledge base.
- Further exploration of the Hamming and City Block (or similar) metrics is planned to assess the advantages of fuzzification of attribute values.

REFERENCES

- Adlassnig, K., Fuzzy Set Theory in Medical Diagnosis, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-16, 1986.
- Agarwal, R., and Tanniru, M., A Petri-Net Based Approach for Verifying the Integrity of Production Systems, *International Journal of Man – Machine Studies*, 36, 1992.
- Agre, G., KBS Maintenance as Learning Two-Tiered Domain Representation, *Proceedings of the International Conference on Case-Based Reasoning Research and Development*, October 1995.
- Ahmad, K., Fulford, H., Griffin, S., and Holmes-Higgin, P., Text-Based Knowledge Acquisition – A Language for Special Purposes Perspective, Research and Development in Expert Systems VIII, Cambridge University Press, 1991.
- Allemang, D., Combining Case-Based Reasoning and Task-Specific Architectures, *IEEE Expert*, October 1994.
- Andert, E.P., Integrated Knowledge-Based System Design and Validation for Solving Problems in Uncertain Environments, *International Journal of Man – Machine Studies*, 36, 1992.
- Backer, E., Gerbrands, J.J., Reiber, J.H.C., Reijs, A.E.M., Krijgsman, H.J., and Van Den Herik, H.J., Modeling Uncertainty in ESATS by Classification Inference, *Pattern Recognition Letters*, #8, Elsevier Science Publishers, 1988.
- Barer, R.D. and Peters, B.F., Why Metals Fail, Gordon and Breach, 1970.
- Barletta, R., An Introduction to Case-Based Reasoning, *AI Expert*, August 1991.
- Battiti, R., Using Mutual Information for Selecting Features in Supervised Neural Net Learning, *IEEE Transactions on Neural Networks*, July 1994.
- Becker, L.A., and Peng, J., Network Processing of Hierarchical Knowledge for Classification and Diagnosis, Artificial Intelligence Research Group, Worcester Polytechnic Institute, Conference Proceedings, c1988.
- Becraft, W.R., and Lee, P.L., An Integrated Neural Network/Expert System Approach for Fault Diagnosis, *Computers in Chemical Engineering*, Volume 17, 1993.
- Bench-Capon, T., Coenen, F., Nwana, H., Paton, R., and Shave, M., Two Aspects of the Validation and Verification of Knowledge-Based Systems, *IEEE Expert*, June 1993.

Bergmann, R., Pews, G., and Wilke, W., Explanation-Based Similarity: A Unifying Approach for Integrating Domain Knowledge Into Case-Based Reasoning For Diagnosis And Planning Tasks, Topics in Case-Based Reasoning, Springer-Verlag, 1994.

Bergmann, R., Wilke, W., Vollrath, I., and Wess, S., Integrating General Knowledge with Object-Oriented Case Representation and Reasoning, *4th German Workshop: Case-Based Reasoning – System Development and Evaluation*, 1996.

Binaghi, E., A Fuzzy Logic Inference Model for a Rule-Based System in Medical Diagnosis, *Expert Systems*, Volume 7, 1990.

Bort, J., Data Mining's Midas Touch, *InfoWorld*, April 1996.

Brooks, C.R., and Choudhury, A., Metallurgical Failure Analysis, McGraw-Hill, 1993.

Brostow, W. and Corneliussen, R.D., Failure of Plastics, Hanser Publishers, 1986.

Chu, W.W., Cardenas, A.F., and Taira, R.K., KMeD: A Knowledge-Based Multimedia Medical Distributed Database System, *Information Systems*, Elsevier, 1995.

Churbuck, D.C., Learning by Example, *Forbes Magazine*, June 8, 1992.

Cooper, A., About Face: The Essentials of User Interface Design, IDG Books, 1995.

Covey, S.R., The 7 Habits of Highly Effective People, Simon & Schuster, 1989.

Cragun, B.J., A Decision-Table-Based Processor for Checking Completeness and Consistency in Rule-Based Expert Systems, *International Journal of Man—Machine Studies*, 26, 1987.

Dash, M., and Liu, H., Feature Selection For Classification, Intelligent Data Analysis, Elsevier Science, 1997.

Doherty, N.F., Knowledge-Based Approaches to Fault Diagnosis, PhD Thesis, University of Bradford, 1992.

Doherty, N.F., Kochhar, A.K., and Main, R., Knowledge-Based Approaches to Fault Diagnosis: A Practical Evaluation of the Relative Merits of Deep and Shallow Knowledge, *Proceedings of the Institution of Mechanical Engineers*, Volume 208, 1994.

Druzdzel, M.J., Qualitative Verbal Explanations in Bayesian Belief Networks, *Artificial Intelligence and Simulation of Behaviour Quarterly*, (Special issue on Bayesian Belief Networks), 1966.

Druzdzel, M.J., Five Useful Properties of Probabilistic Knowledge Representations from the Point of View of Intelligent Systems, *Fundamenta Informaticae*, 30, 1997.

- Dutta, S., and Bonissone, P.P., Integrating Case- and Rule-Based Reasoning, *International Journal of Approximate Reasoning*, 8, 1993.
- Esaklul, K.A., Handbook of Case Histories in Failure Analysis, ASM International, 1992.
- Esogbue, A.O., and Elder, R.C., Fuzzy Sets and the Modeling of Physician Decision Processes, Part I: The Initial Interview – Information Gathering Session, Fuzzy Sets and Systems, North-Holland Publishing, Volume 2, 1979.
- Esogbue, A.O., and Elder, R.C., Fuzzy Sets and the Modeling of Physician Decision Processes, Part II: Fuzzy Diagnosis Decision Models, Fuzzy Sets and Systems, North-Holland Publishing, Volume 3, 1980.
- Fincher, D.R., Corrosion Control in Petroleum Production, NACE, 1979.
- Fink, P.K., Luth, J.C. and Duran, J.W., A General Expert System Design for Diagnostic Problem Solving, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, September 1985.
- Fourali, C., Fuzzy Logic and the Quality of Assessment of Portfolios, *Fuzzy Sets and Systems* 68, Elsevier, 1994.
- Fox, J., Some Observations on Fuzzy Diagnosis and Medical Computing, *International Journal of Bio-Medical Computing*, Volume 8, Applied Science Publishers, 1977.
- Frappaola, C., Interviewing the Document, *Imaging Magazine*, September 1995.
- French, D.N., Metallurgical Failures in Fossil Fired Boilers, John Wiley & Sons, 1983.
- Furuta, H., and Shiraishi, N., Fuzzy Importance in Fault Tree Analysis, Fuzzy Sets and Systems 12, Elsevier, 1984.
- Gallant, S.I., Connectionist Expert Systems, *Communications of the ACM*, February 1988.
- Garcia, F.E., Alloy Identification Using Neural Networks and Fuzzy Systems, Research Report, Louisiana State University Department of Industrial and Manufacturing Systems Engineering, Baton Rouge, LA, 1997.
- Geissmann, J.R., and Schultz, R.D., Verification and Validation of Expert Systems, *AI Expert*, February 1988.
- Golding, A.R., and Rosenbloom, P.S., Improving Accuracy By Combining Rule-Based and Case-Based Reasoning, *Artificial Intelligence* 87, Elsevier, 1996.

Goldstein, R.J., Commentary—Learning for Life, *Mechanical Engineering*, ASME, January 1997.

Gonzalez, A.J., Xu, L., and Gupta, U.M., Validation Techniques for Case-Based Reasoning Systems, *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Volume 28, No. 4, July, 1998.

Graham-Jones, J., and Mellor, B.G., Expert and Knowledge-Based Systems in Failure Analysis, *Engineering Failure Analysis*, Volume 2, Elsevier, 1995.

Graham-Jones, J., and Mellor, B.G., The Development of a Generic Failure Analysis Expert System Based on Case-Based Reasoning, *Proceedings of Corrosion 96 Symposium*, Denver, CO, 1996.

Green, C.J.R., and Keyes, M.M., Verification and Validation of Expert Systems, *Proceedings of Westex 87: Western Conference on Expert Systems*, IEEE CS Press, Los Alamitos, CA, 1987.

Guan, J., and Graham, J.H., Diagnostic Reasoning with Fault Propagation Digraph and Sequential Testing, *IEEE Transactions On Systems, Man, and Cybernetics*, October 1994.

Guida, G., and Mauri, G., Evaluating Performance and Quality of Knowledge Based Systems: Foundation and Methodology, *IEEE Transactions on Knowledge and Data Engineering*, April 1993.

Gupta, K.M., and Montazemi, A.R., Empirical Evaluation of Retrieval in Case-Based Reasoning Systems Using Modified Cosine Matching Function, *IEEE Transactions on systems, Man, and Cybernetics – Part A: Systems and Humans*, Volume 27, September 1997.

Harper, R.F., The Code of Hammurabi, Gaunt & Sons, 1994.

Hart, A.E., The Role of Induction in Knowledge Elicitation, *Expert Systems*, January 1985.

Higgins, C.M., and Goodman, R.M., Incremental Learning with Rule-Based Neural Networks, *International Joint Conference on Neural Networks*, Seattle, 1991.

Hinkle, D., and Toomey, C., Applying Case-Based Reasoning to Manufacturing, *AI Magazine*, Spring 1995.

Holmes-Higgin, P.R., *Marvin Expert System Tool User Guide*, University of Surrey, 1988.

Holsheimer, M. and Siebes, A., Data Mining: The Search for Knowledge in Databases, *CWI*, 1991.

Hoppe, T., VVT Terminology: A Proposal, *IEEE Expert*, June 1993.

Hudson, D.L., Cohen, M.E., and Anderson, M.F., Use of Neural Network Techniques in a Medical Expert System, *International Journal of Intelligent Systems*, Volume 6, 1991.

Hughes, T., (Chairman), Developments in Artificial Intelligence, Funding a Revolution: Government Support for Computing Research, National Academy Press, 1999.

Hui, S.C., Goh, A., and Lau, L.H., A Multimedia Information System for IC Failure Analysis, *The Computer Journal*, Volume 36, 1993.

Hutchings, F.R. and Unterweiser, P.M., Failure Analysis: The British Engine Technical Reports, American Society for Metals, 1981.

Isaacson, D.R., Davis, T.J., and Robinson, J.E. III, Knowledge-Based Runway Assignment for Arrival Aircraft in the Terminal Area, *AIAA Guidance, Navigation, and Control Conference*, August 1997.

Jackson, A.H., Machine Learning, *Expert Systems*, Volume 5, 1988.

Jafar, M., and Bahill, A.T., Interactive Verification of Knowledge-Based Systems, *IEEE Expert*, February 1993.

Johannsen, G. and Alty, J.L., Knowledge Engineering for Industrial Expert Systems, *Automatica*, Volume 27, 1991.

Kane, R., and Milgram, M., Extraction of Semantic Rules form Trained Multilayer Neural Networks, *International Conference on Neural Networks*, 1993.

Klinger, D.W., A Decision Centered Design Approach to Case-Based Reasoning: Helping Engineers Prepare Bids and Solve Problems, Advances in Agile Manufacturing, IOS Press, 1994.

Kohno, T., Hamada, S., Araki, D., Kojima, S., and Tanaka, T., Error Repair and Knowledge Acquisition Via Case-Based Reasoning, *Artificial Intelligence 91*, Elsevier, 1997.

Lai, G.Y., High-Temperature Corrosion of Engineering Alloys, ASM International, 1990.

Lee, S., and O'Keefe, R.M., Developing a Strategy for Expert System Verification and Validation, *IEEE Transactions on Systems, Man, and Cybernetics*, April 1994.

Lee-Post, A., Knowledge Acquisition Automation: Research Issues and Directions, *Journal of Computer Information Systems*, Fall 1994.

van Leijen, H., and Druzdzal, M.J., Reversible Causal Mechanisms in Bayesian Networks, *Working Notes of the AAAI 1998 Symposium on Prospects for a Commonsense Theory of Causation*, March 1998.

Leong, M.C., Statistical Process Control and SPSS QI Analyst, *Proceedings of the SPSS Users' Conference*, November 1996.

Liao, T.W., A Fuzzy Multi-Criteria Decision Making Method for Material Selection, *Journal of Manufacturing Systems*, Vol. 15, No.1, 1996, 1-12.

Liao, T.W., and Lee, K.S., Integration of a Feature-Based CAD System and an ART1 Neural Model for GT Coding and Part Family Forming, *Computers in Industrial Engineering*, Volume 26, 1994.

Liao, T.W., Zhan, Z.H., Mount, C.R., An Integrated Database and Expert System for Failure Mechanism Identification: Part I – Automated Knowledge Acquisition, *Engineering Failure Analysis*, 2000a, to appear.

Liao, T.W., Zhan, Z.H., Mount, C.R., An Integrated Database and Expert System for Failure Mechanism Identification: Part II – The System and Performance Testing, *Engineering Failure Analysis*, 2000b, to appear.

Liao, T.W., Zhang, Z.M., and Mount, C.R., A Case-Based Reasoning Approach to Identifying Failure Mechanisms, *Engineering Applications of Artificial Intelligence*, 2000c, to appear

Liao, T.W., Zhang, Z.M., and Mount, C.R., Similarity Measures for Retrieval in Case-Based Reasoning Systems, *Applied Artificial Intelligence*, 1998.

Lin, Y., and Druzdzal, M.J., Computational Advantages of Relevance Reasoning in Bayesian Belief Networks, *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI-97)*, August 1997.

Lin, Y., and Druzdzal, M.J., Relevance-Based Sequential Evidence Processing in Bayesian Networks, *Proceedings of the Uncertain Reasoning in Artificial Intelligence Track of the Eleventh International Florida Artificial Intelligence Research Symposium (FLAIRS-98)*, May 1998.

Low, B.T., Lui, H.C., Tan, A.H., and Teh, H.H., Connectionist Expert System with Adaptive Learning Capability, *IEEE Transactions on Knowledge and Data Engineering*, June 1991.

Marra, J., An Expert System Eases Rotor Design, *Mechanical Engineering*, April 1997.

Martine-Mattei, C., Validation, Verification and Testing Procedures in the Life Cycle of Knowledge-Based Systems, EDF Research and Development Division, 1992.

Mengshoel, O.J., KVAT: A Tool for Incremental Knowledge Validation in a Knowledge Engineering Workbench, *Proceedings of the European Workshop on the Verification and Validation of Knowledge-Based Systems (EuroVAV 91)*, Cambridge, U.K., 1991.

Mengshoel, O.J., and Delab, S., Knowledge Validation: Principles and Practice, *IEEE Expert*, June 1993.

Milacic, V.R., Majstorovic, V.D. and Race, I.Z., Building Expert System for Diagnosis and Maintenance in FMS, *IEEE Transactions*, 1988.

Mitra, S., Fuzzy MLP Based Expert System for Medical Diagnosis, Fuzzy Seta and Systems, Elsevier, 1994.

Morales, E. and Garcia, H., A Modular Approach to Multiple Faults Diagnosis, Artificial Intelligence in Process Engineering, Academic Press, 1990.

Mount, C.R., Liao, T.W., and Chen, Y.S., Integrated Knowledge Engine for Failure Analysis, Smart Engineering Systems: Neural Networks, Fuzzy Logic, Data Mining, and Evolutionary Programming, ASME Press, 1997.

Naser, J.A., Nuclear Power Plant Expert System Verification and Validation, *Validation and Verification of Knowledge-Based Systems*, IEEE CS Press, 1991,

Naumann, F.K., Failure Analysis: Case Histories and Methodology, American Society for Metals, 1983.

Nazareth, D.L., Issues in the Verification of Knowledge in Rule-Based Systems, *International Journal in Man – Machine Studies*, 30, 1989.

Nazareth, D.L., Investigating the Applicability of Petri Nets for Rule-Based System Verification, *IEEE Transactions on Knowledge and Data Engineering*, June 1993.

Nicholson, C., Learning without Case Records: A Mapping of the Repertory Grid Technique onto Knowledge Acquisition from Examples, *Expert Systems*, May 1992.

Obrzut, J.J., Sleuthing Out Metal Identities, *Heat Treating*, April 1993.

O’Keefe, R.M., Balci, O., and Smith, E.P., Validating Expert System Performance, *IEEE Expert*, Winter 1987.

O’Keefe, R.M., and O’Leary, D.E., Expert System Verification and Validation: A Survey and Tutorial, *Artificial Intelligence Review*, 7, 1993.

Olson, J.R., and Rueter, H.H., Extracting Expertise from Experts: Methods for Knowledge Acquisition, *Expert Systems*, August 1987.

Onisko, A., and Druzdzal, M.J., Wasyluk, H., Graphical Probabilistic Models in Diagnosis of Liver Disorders, *Proceeding of the International Seminar on Statistics and Clinical Practice*, June 1998.

Osif, B., Teaching Research Skills: Innovative Strategies for Library Use Instruction, *Presented at the Special Libraries Association Annual Conference*, June 1996.

Padalkar, S., Karsai, G., Biegl, C., Sztipanovits, J., Okuda, K., and Miyasaka, N., Real-Time Fault Diagnostics, *IEEE Expert*, June 1991.

Parsaye, K., and M. Chignell, Knowledge Acquisition and Validation, Expert Systems for Experts, John Wiley & Sons, 1988.

Pepper, J., An Expert System for Automotive Diagnosis, *ACM Computing Surveys*, Volume 17, 1985.

Perez, R.A., and Koh, S.W., Integrating Expert Systems with a Relational Database in Semiconductor Manufacturing, *IEEE Transactions on Semiconductor Manufacturing*, August 1993.

Petzow, G., Case Histories in Failure Analysis, American Society for Metals, 1979.

Plant, R.T., Techniques for Knowledge Acquisition from Text, *Journal of Computer Information Systems*, Fall 1994.

Polat, F., and Guvenir, H.A., UVT: A Unification-Based Tool for Knowledge Base Verification, *IEEE Expert*, June 1993.

Powell, G.W. and Mahmoud, S.E., Metals Handbook: Failure Analysis and Prevention, 9th Edition, V11, American Society for Metals, 1986.

Prague, C.N., Amo, C.W., and Foxal, J.D., Access 97 Secrets, IDG Books Worldwide, 1997.

Preece, A.D., Towards a Methodology for Evaluating Expert Systems, *Expert Systems*, November 1990.

Prerau, D.S., Knowledge Acquisition in the Development of a Large Expert System, *AI Magazine*, Summer 1987.

Quinlan, J.R., Induction of Decision Trees, Machine Learning, Kluwar Academic Publishers, 1986.

Quinlan, J.R., Unknown Attribute Values in Induction, *International Journal of Man-Machine Studies*, Volume 27, 1987.

Reategui, E.B., Campbell, J.A., and Leao, B.F., A Case-Based Model that Integrates Specific and General Knowledge in Reasoning, *Applied Intelligence*, July 1997.

Renard, F.X., Sterling, L., and Brosilow, C., Knowledge Verification in Expert Systems Combining Declarative and Procedural Representations, *Computers in Chemical Engineering*, Volume 17, 1993.

Rissland, E.L., and Skalak, D.B., CABARET: Rule Interpretation in a Hybrid Architecture, *International Journal of Man – Machine Studies*, 34, 1991.

Romaniuk, S.G., and Hall, L.O., Decision Making on Creditworthiness, Using a Fuzzy Connectionist Model, *Fuzzy Sets and Systems*, Elsevier, 1992.

Ryder, D.A., Davies, T.J., Brough, I., and Hutchings, F.R., General Practice in Failure Analysis, *Metal Handbook: Volume 10 – Failure Analysis and Prevention*, 8th Edition, ASM, 1975.

Schumpeter, J., *Ten Great Economists*, George Allen & Unwin, London, 1952.

Scott, R., Artificial Intelligence: Its Use in Medical Diagnosis, *Journal of Nuclear Medicine*, March 1993.

Shortliffe, E.H. and Buchanan, B.G., A Model of Inexact Reasoning in Medicine, *Mathematical Biosciences*, Vol. 23, American Elsevier Publishing Co., Inc., 1975.

Siler, W., Tucker, D., and Buckley, J., A Parallel Rule Firing Fuzzy Production System With Resolution of Memory Conflicts By Weak Fuzzy Monotonicity, Applied to the Classification of Multiple Objects Characterized By Multiple Uncertain Features, *International Journal of Man – Machine Studies*, 26, 1987.

Siler, W., and Buckley, J.J., Fuzzy Reasoning: An Alternative to Approximate Reasoning Theory, Accepted for Publication by the *International Society for Uncertainty Management*, 1996, to appear.

Smyth, B., and Cunningham, P., A Comparison of Incremental Case-Based Reasoning and Inductive Learning, *Proceedings of the European Workshop on Advances in Case-Based Reasoning*, November 1994.

Stottler, R.H., CBR for Cost and Sales Prediction, *AI Expert*, August 1994.

Surma, J., and Vanhoof, K., Integrating Rules and Cases for the Classification Task, Proceedings of the International Conference for Case-Based Reasoning Research and Development, Springer, October 1995.

Suwa, M., Scott, A.C., and Shortliffe, E.H., An Approach to Verifying Completeness and Consistency in a Rule-Based Expert System, *AI Magazine*, Fall 1982.

Tan, A.H., Pan, Q., and Lui, H.C., The, H.H., Connectionist Expert System in Fault Diagnosis of Avionics Line Replaceable Units, *New Challenges in Aircraft Maintenance Conference*, Singapore, February 1990.

Tansley, D.S.W., and Hayball, C.C., Knowledge-Based Systems: Analysis and Design, Prentice-Hall, 1993.

Thielsch, H., Defects and Failures in Pressure Vessels and Piping, Krieger, 1977.

Tzafestas, S., Palios, L. and Cholin, F., Diagnostic Expert System Inference Engine Based On the Certainty Factors Model, *Knowledge-Based Systems*, Volume 7, 1994.

Uebele, V., Abe, S., and Lan, M.S., Extracting Fuzzy Rules from Pattern Classification Neural Networks, *IEEE International Conference on Systems, Man, and Cybernetics*, Volume 2, 1993.

Vargas, J.E., and Raj, S., Developing Maintainable Expert Systems Using Case-Base Reasoning, *Expert Systems*, November 1993.

Walker, T.C., and Miller, R.K., Expert System Handbook, The Fairmont Press, 1989.

Walton, H.W., Failure Diagnostics – Application of Expert Systems, *Conference Proceedings of the International Conference and Exhibits on Failure Analysis*, July 1991.

Watson, I., and Marir, F., Case-Based Reasoning: A Review, *The Knowledge Engineering Review*, Volume 9:4, 1994.

Wilson, D.R. and Martinez, T.R., Improved Hetrogeneous Distance Functions, *Journal of Artificial Intelligence Research*, Volume 6, 1997.

Witten, I.H., and MacDonald, B.A., Using Concept Learning for Knowledge Acquisition, *International Journal of Man-Machine Studies*, Volume 29, 1988.

Wu, C.H., and Lee, S.J., Enhanced High-Level Petri Nets with Multiple Colors for Knowledge Verification/Validation of Rule-Based Expert Systems, *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, Volume 27, October 1997.

Wulpi, D.J., Understanding How Components Fail, American Society for Metals, 1985.

Wulpi, D.J., Understanding How Components Fail, Second Edition, American Society for Metals, 1999.

Xu, L.D., Developing a Case-Based Knowledge System for AIDS Prevention, *Expert Systems*, August 1994.

Yuan, Y., and Shaw, M.J., Induction of Fuzzy Decision Trees, *Fuzzy Sets and Systems* 69, Elsevier, 1995.

Zahedi, F., Intelligent Systems for Business, Wadsworth, 1993.

Zhan, Z., An Integrated Database and Expert System for Failure Analysis with Knowledge Acquisition and Test Recommendation Capabilities, A Thesis Submitted to the Louisiana State University Department of Industrial and Manufacturing Systems Engineering, Baton Rouge, LA, 1998.

Zhang, Z., Applying Case-Based Reasoning and Genetic Algorithms to Failure Analysis, A Thesis Submitted to the Louisiana State University Department of Industrial and Manufacturing Systems Engineering, Baton Rouge, LA, 1998.

Zhang, Z., Liao, T.W., and Mount, C.R., Applying Case-Based Reasoning and Genetic Algorithms to Failure Mechanism Identification, Smart Engineering Systems: Neural Networks, Fuzzy Logic, Data Mining, and Evolutionary Programming, ASME Press, 1997.

Zimmerman, H.-J., Fuzzy Set Theory and Its Applications, 2nd Edition, Kluwer Academic Publishers, 1991.

APPENDIX A

SCREEN VIEWS OF aIFAS

Operating Mode <input type="radio"/> S1 Visual <input type="radio"/> S2 Macro <input type="radio"/> S3 Micro <input checked="" type="radio"/> All 4 Steps <input type="radio"/> Step+Prune		Metric <input type="radio"/> City Block <input type="radio"/> Cosine <input type="radio"/> Euclidean <input checked="" type="radio"/> Hamming <input type="radio"/> KG		<input checked="" type="checkbox"/> Exclude Unused Attributes <input type="checkbox"/> Use Mode/Attribute Weighting <input type="checkbox"/> Use modified KG version <input checked="" type="checkbox"/> Use Grouped Attributes	
Finish <input checked="" type="radio"/> As-Is <input type="radio"/> N		Sorting <input type="radio"/> D <input checked="" type="radio"/> A		KG Combination <input type="radio"/> Add <input type="radio"/> Multiply <input type="radio"/> Only Se <input checked="" type="radio"/> Only Sp	

KB will have cases. aIFAS not Setup!

Minimum of cases.

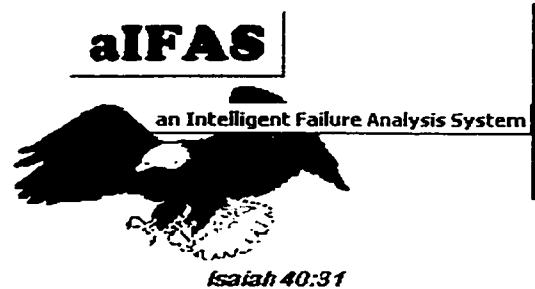
Hold % of cases..

Run sets of trials. No KG coefficients!

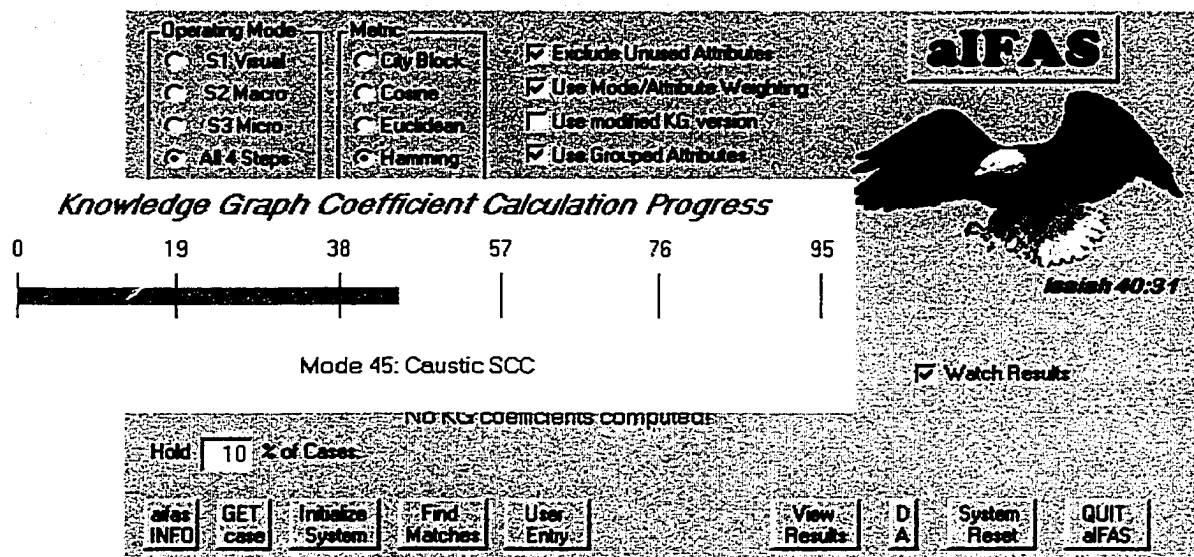
Use test cases/set.

☒ Watch Results

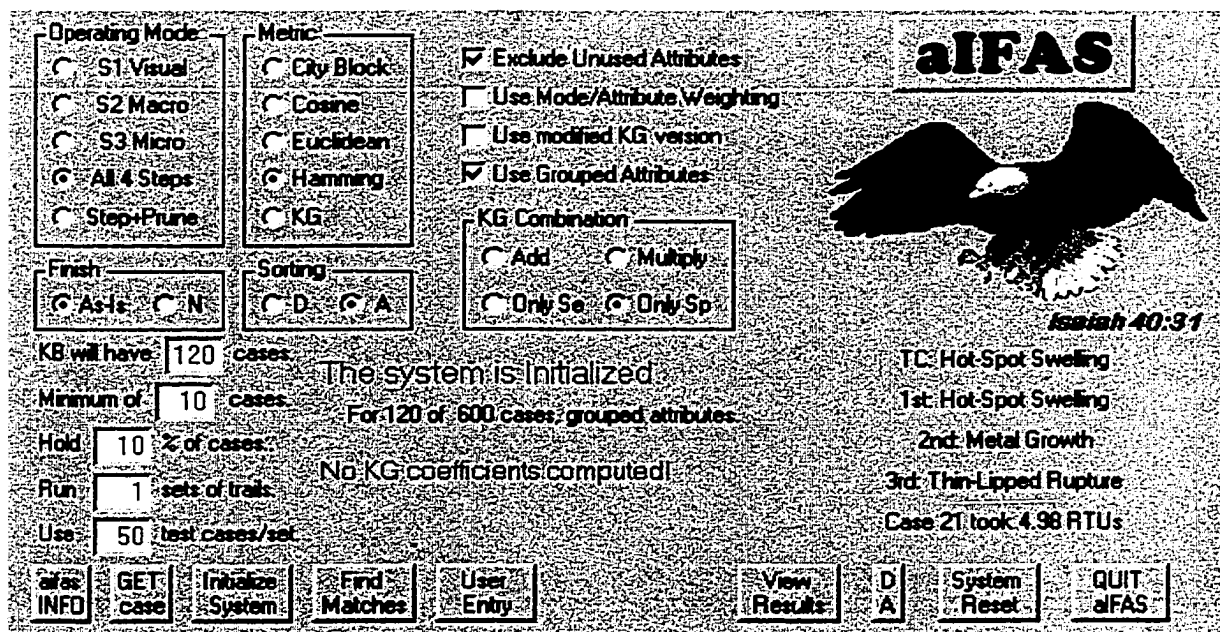
aifas INFO	GET case	Initialize System	Find Matches	User Entry	View Results	D A	System Reset	QUIT aIFAS
---------------	-------------	----------------------	-----------------	---------------	-----------------	--------	-----------------	---------------



The main control panel for aIFAS. There are pop up tips, like the one exposed in this view, to offer information and explain the function of different controls.



The system uses progress meters to keep the user aware that things are happening in the background. This message appears whenever coefficients are being



While a set of test cases are being evaluated the user is presented with intermediate results on a case-by-case basis.

Averaged Results

Evaluated with Test Case Set data.

Hamming metric scheme.

Used 120 of 600 stored cases.

Metric was used as calculated.

Sorting done in ascending order.

Unused case attributes were excluded.

No weighting factors used.

Grouped case attributes.

All four steps allowed.

Keep 10 % or at least 10 cases.

**aIFAS
Results**

15-Oct-00 17:48

Possible Match 90%

Best Match 88%

2nd Place Match 68%

3rd Place Match 54%

5th Place Match 44%

10th Place Match 12%

Average Time 4.89 RTU's

[Save This
Information](#)[Print This
Information](#)[Finished
Viewing](#)

Once a set of test case has been processed by aIFAS, the user has the option of viewing the results in two formats. This one shows the parameter settings that were used and the average performance for the set of test cases.

1st 2nd 3rd Place Results

Evaluated with Test Case Set data.

Hamming metric scheme.

Used 120 of 600 stored cases.

Metric was used as calculated.

Sorting done in ascending order.

Unused case attributes were excluded.

No weighting factors used.

Grouped case attributes.

All four steps allowed.

Keep 10 % or at least 10 cases.

**aIFAS
Results**

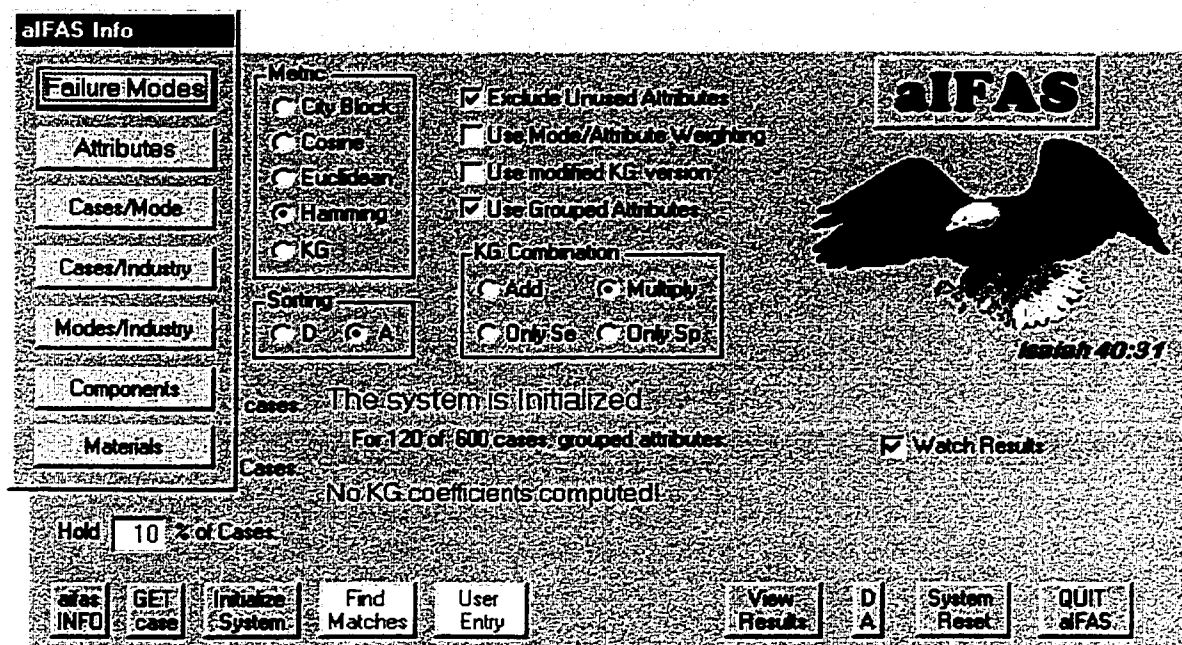
15-Oct-00 17:50

TC mode:	Hot-Spot Swelling	
1st	Hot-Spot Swelling	
2nd	Metal Growth	
3rd	Thin-Lipped Rupture	
	Metric	Case#
1st	1.000	54
2nd	3.000	14
3rd	3.000	74
Record: 14 of 60		

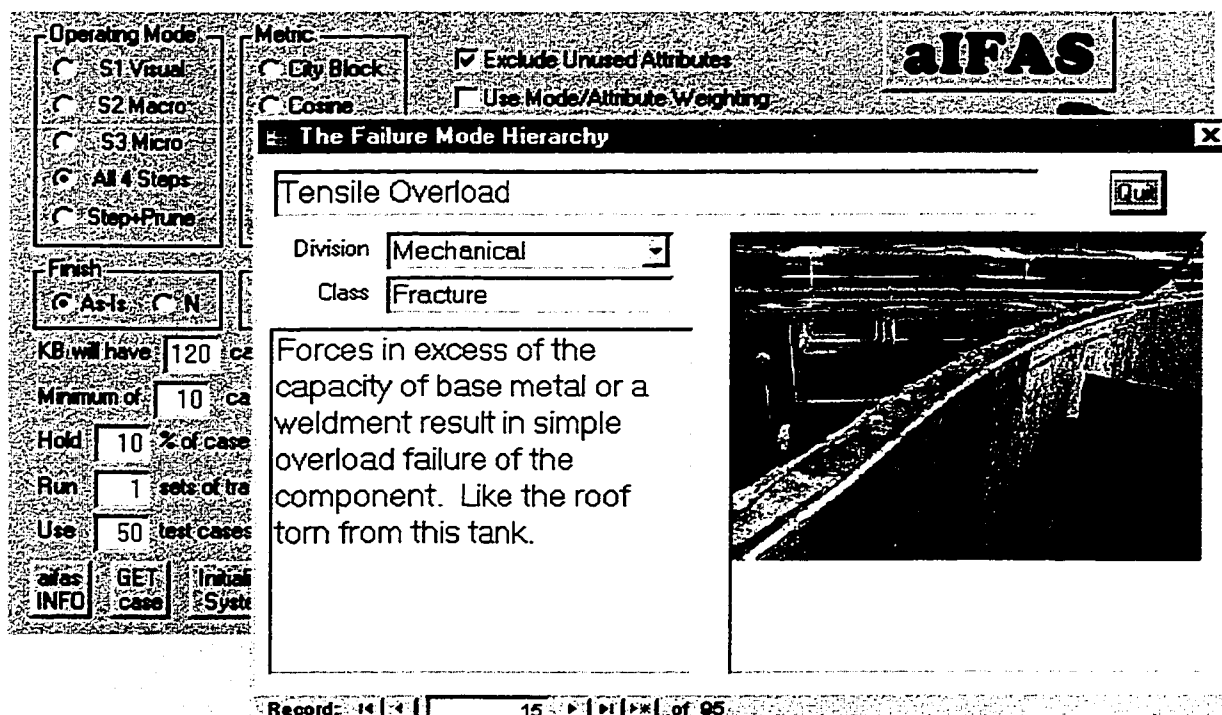
Average Time 4.89 RTU's

[Print This
Information](#)[Finished
Viewing](#)

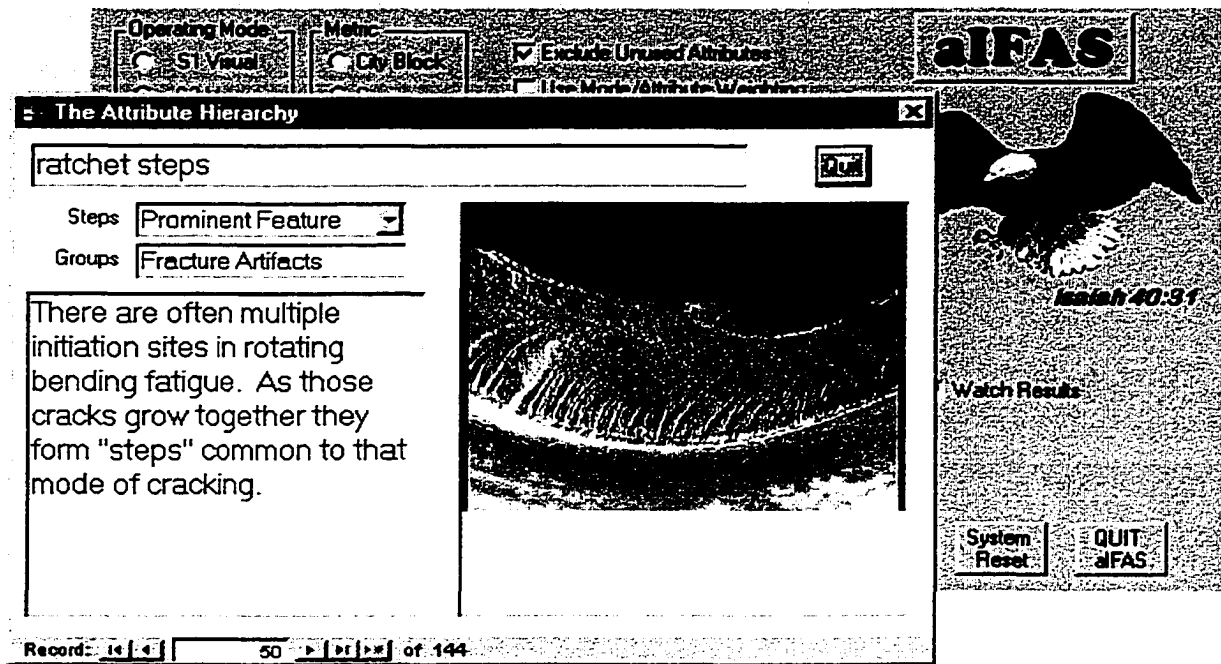
An alternate method for display results is to present the best three matches for a test case. The user can scroll through the results for each of the test cases that was analyzed. The parameters used for the analyses are also displayed.



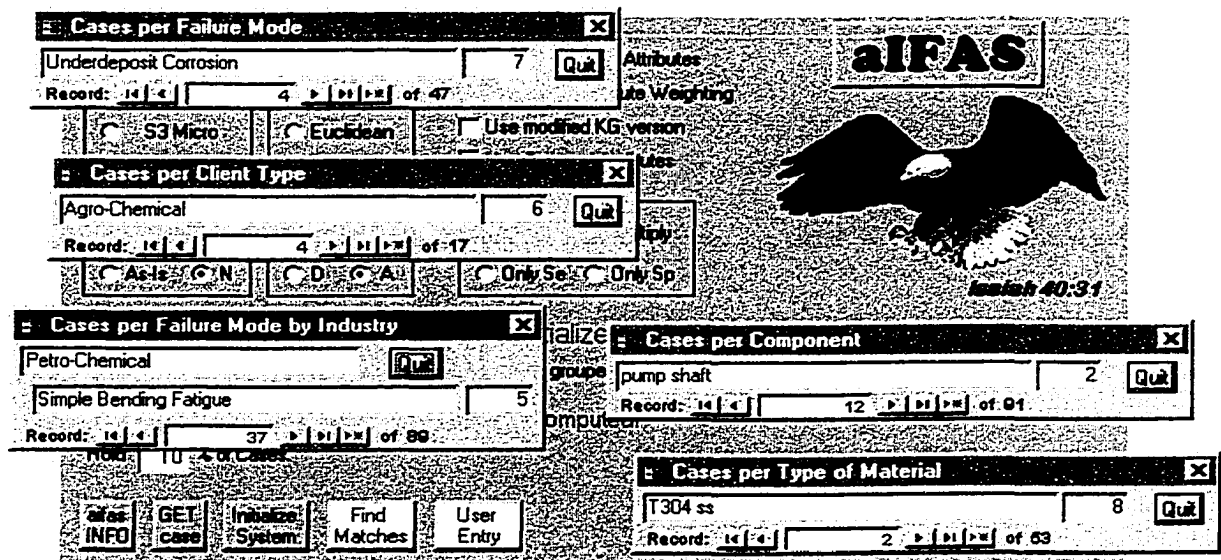
Aside from analyzing test cases, aIFAS can provide the user with information contained in its knowledge base. This view shows the control that is presented for making a selection. Information about failure modes, attributes, or queries of the numbers of cases in different categories are possible.



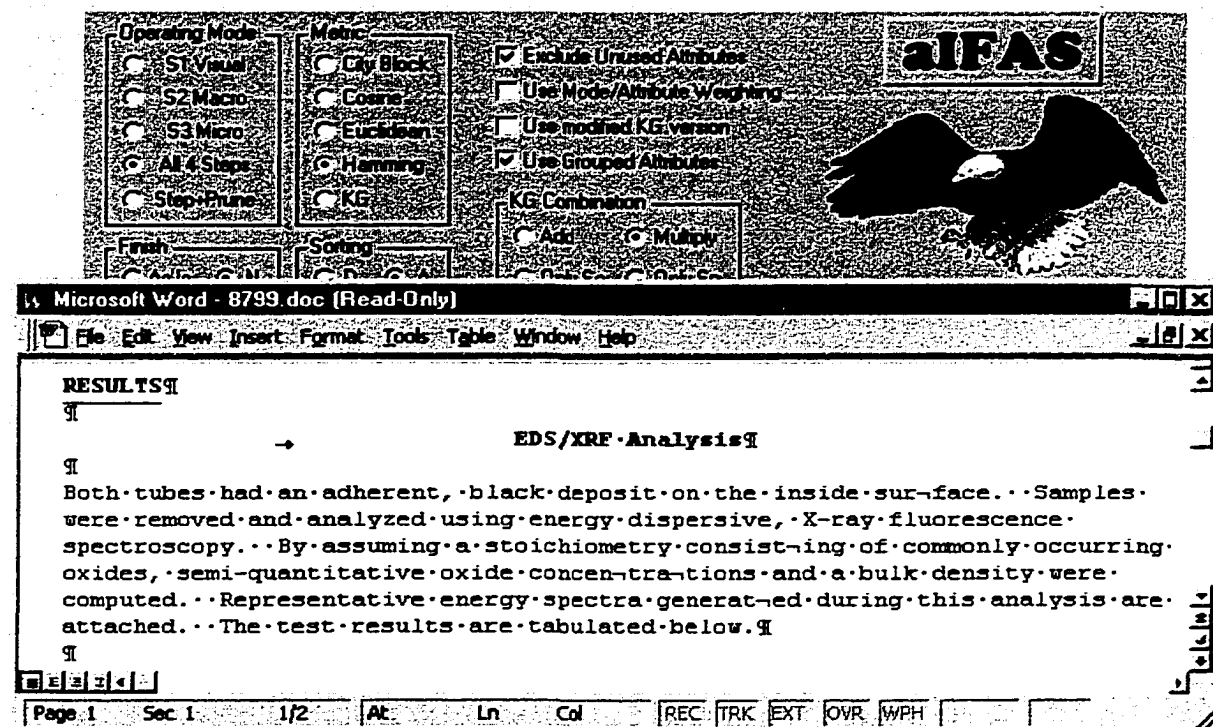
For each failure mode represented in aIFAS, the user can obtain a text definition accompanied by an image of a typical such failure. There are also references to the example cases in aIFAS that correspond to that failure mode.



Information is presented to the user to explain the various case attributes. This offers a common reference to avoid the confusion of unfamiliar, or inconsistent, terminology.



If aIFAS is queried for data on the distribution of cases in particular categories, the user is provided that listing, along with the identification numbers of the individual cases.



The case identification number is sufficient information to recall the full report underlying an example case in the knowledge base. This view illustrates the recall of such an analysis report.

ID: 9 Report: Test Failure Mode: unknown

Admin | Prominent | Macroscopic | Microscopic |

Bulk Microstructure: pearlite & cementite

Microstructural Damage:

Microstructural Change:

Microstructural Artifacts: oxide-filled cracks

Crack Path:

Crack Profile:

What Mode 4 Attribute

Case #: 32

Hot Cracking

Record: 1 of 4

Delete | 1 Records in UEG | Add Record | First | <<< | >>> | Last | FINISHED

While a user is entering data for an individual case (a trial case not in the routine test set), aIFAS forces data entry along a controlled path that parallels the process of a conventional failure investigation.

ID: 9 Report Failure Mode: Chloride SCC

Admin Prominent

What Attributes 4 Mode

Prominent Features: Chloride SCC

Case#	DL	DO	DA	TD	WP	SAP	SAC	SAR	FAP	FAR
71										

Record: 14 of 6

Macro Features

Case#	SD	SF	FP	BD	DE	DP	CSF
71			0.95	0.60			

Record: 14 of 6

Micro Features

ID	BM	MD	MC	MA	CPa	CPi
71	0.45				0.90	0.25

Record: 14 of 6

Special Features

ID	Part	MoC	PS	SS	TC	FF	CA	C	FSD
71			0.80					0.10	

Record: 14 of 6

What Mode 4 Industry

Client Type: Agro-Chemical Case #: 71

General Division: Environmental

Class of Failure: Corrosion

Failure Mode: Chloride SCC

Record: 14 of 6

FINISHED

There are different sorts of information that can be provided the user while entering a new case. One sort of display will indicate the list of attributes most commonly used for a particular failure mode.

Operating Mode: S1-Visual Metric: Exclude Unused Attributes

aIFAS

Basic Results

Case ID	Metric	CR1	PS1	RTU%	BP%
MNO22	CB	88.0	3.02	29.10	90
MNO23	C	90.0	1.75	51.51	90
MNO24	C	90.0	1.76	51.06	90
MNO25	E	88.0	2.70	32.59	90
MNO26	E	88.0	2.71	32.51	90
MNO27	H	88.0	3.11	28.31	90
MNO28	H	88.0	3.08	28.55	90
MNO29	KG	66.0	1.76	37.47	90
MNO30	KG	60.0	1.61	37.20	90

Hold: 10 % of Cases For 120 of 600 cases, grouped attributes.

aIFAS GET case Initialize System Find Matches User Entry Look Here >> View Results D/A System Reset QUIT aIFAS

Isaiah 40:31

Watch Results

The basic results presented by aIFAS when a group of parametric data has been analyzed. Recall the CR is Correctness Ratio, PS is Performance Score, and RTU is the Reference Time Unit. The BP value is the best possible score, or a value for how many of the test cases actually have solutions in the knowledge base.

APPENDIX B

EXTRA INFORMATION TABULATIONS

Client Type	Mechanical			Environmental		Causal		Totals
	Distortion	Wear	Fracture	Corrosion	Thermal	Material	Fabrication	
Agro-Chemical			10	16	2	1		29
Air Liquefaction		1	4	1	1			7
Automotive			2	1	1		1	5
Aviation			1					1
School		1						1
Federal Agency	1		2	1				4
Food Processing				2				2
Trucking	1		1					2
Individual			3					3
Legal Testimony	1		8	3	1		1	14
Manufacturer		3	8	3		3	2	19
Maritime			2			1		3
Municipality		3		1				4
Oil Exploration		3	30	16		1	3	53
Petro-Chemical	1	7	39	59	17	5	7	135
Oil Refining	4	3	48	54	24	5	8	146
Plastics	2	3	20	23	5		3	56
Power Generation	1	3	6	3	12	2	1	28
Pulp & Paper			5	5	2			12
Service Company	1	3	15	13	5	2	5	44
Specialty Minerals			9	8	4	1	1	23
State Agency		1						1
Sugar Refining			1			1		2
Tobacco			2	4				6
Totals by Class	12	31	216	213	74	22	32	600
Totals by Division	259			287		54		

APPENDIX C
METRIC COMPARISON TEST DATA

DA001 – DA020	Determine sorting order for selecting best matching cases from computed metric values
MN021 – MN040	Explore the effects of using normalization or simply accepting the metric value as-computed
IE041 – IE058	Explore the effects of including or excluding unused attributes
WT059 – WT098	Explore the combination form for the Knowledge Graph Similarity and evaluate the effectiveness of that number as a weighting term for the other metrics
WM099 – WM118	Similar to above, except using the Modified form for the KGS Sensitivity term
SP119 – SP138	Explore the performance when a step-and-prune mode of operation is used to compare cases
US139 – US178	Explore the results when User Entry operation is simulated
IL179 – IL278	Explore the effects of incrementally increasing the size of the knowledge base (incremental learning)

	DA 001	DA 002	DA 003	DA 004	DA 005	DA 006	DA 007	DA 008	DA 009	DA 010
City Block Metric	X	X								
Cosine Metric			X	X						
Euclidean Metric					X	X				
Hamming Metric							X	X		
Knowledge Graph Metric									X	X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes	X	X	X	X	X	X	X	X	X	X
Grouped Attributes										
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort	X		X		X		X		X	
Ascending Sort		X		X		X		X		X
Include All Attributes	X	X	X	X	X	X	X	X		
Exclude Unused Attributes									X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	X	X
Modified KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a		
No Weighting Coefficients	X	X	X	X	X	X	X	X		
Add KG Terms									X	X
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity										
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	0	88	90	0	0	88	0	88	66	0
2 nd Match, AVG% Correct	0	70	72	0	0	70	0	70	56	0
3 rd Match, AVG% Correct	0	58	58	0	0	58	0	58	46	0
5 th Match, AVG% Correct	0	42	44	0	0	42	0	42	34	0
10 th Match, AVG% Correct	0	8	4	0	0	8	0	8	8	0
Average Time/Case	29.5	29.0	48.3	49.4	31.8	30.9	33.7	28.3	38.7	36.6

	DA 011	DA 012	DA 013	DA 014	DA 015	DA 016	DA 017	DA 018	DA 019	DA 020
City Block Metric	X	X								
Cosine Metric			X	X						
Euclidean Metric					X	X				
Hamming Metric							X	X		
Knowledge Graph Metric									X	X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes										
Grouped Attributes	X	X	X	X	X	X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort	X		X		X		X		X	
Ascending Sort		X		X		X		X		X
Include All Attributes	X	X	X	X	X	X	X	X		
Exclude Unused Attributes									X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	X	X
Modified KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a		
No Weighting Coefficients	X	X	X	X	X	X	X	X		
Add KG Terms									X	X
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity										
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	0	86	84	0	0	84	0	88	62	0
2 nd Match, AVG% Correct	0	70	68	0	0	70	0	68	52	0
3 rd Match, AVG% Correct	0	56	52	0	0	52	0	54	44	0
5 th Match, AVG% Correct	0	50	30	0	0	28	0	44	46	0
10 th Match, AVG% Correct	0	4	12	0	0	16	0	12	10	0
Average Time/Case	7.61	7.80	12.3	11.9	8.29	8.33	7.20	7.08	35.7	35.9

	MN 021	MN 022	MN 023	MN 024	MN 025	MN 026	MN 027	MN 028	MN 029	MN 030
City Block Metric	X	X								
Cosine Metric			X	X						
Euclidean Metric					X	X				
Hamming Metric							X	X		
Knowledge Graph Metric									X	X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes	X	X	X	X	X	X	X	X	X	X
Grouped Attributes										
Metric as-computed	X		X		X		X		X	
Normalized		X		X		X		X		X
Descending Sort			X	X					X	X
Ascending Sort	X	X			X	X	X	X		
Include All Attributes	X	X	X	X	X	X	X	X		
Exclude Unused Attributes									X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	X	X
Modified KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a		
No Weighting Coefficients	X	X	X	X	X	X	X	X		
Add KG Terms									X	X
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity										
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	88	88	90	90	88	88	88	88	66	60
2 nd Match, AVG% Correct	70	70	72	72	70	70	70	70	56	36
3 rd Match, AVG% Correct	58	58	58	58	58	58	58	58	46	24
5 th Match, AVG% Correct	42	42	44	44	42	42	44	42	34	18
10 th Match, AVG% Correct	8	8	4	4	8	8	10	8	8	10
Average Time/Case	29.1	30.6	51.5	51.1	32.6	32.5	28.3	28.6	37.5	37.2

	MN 031	MN 032	MN 033	MN 034	MN 035	MN 036	MN 037	MN 038	MN 039	MN 040
City Block Metric	X	X								
Cosine Metric			X	X						
Euclidean Metric					X	X				
Hamming Metric							X	X		
Knowledge Graph Metric									X	X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes										
Grouped Attributes	X	X	X	X	X	X	X	X	X	X
Metric as-computed	X		X		X		X		X	
Normalized		X		X		X		X		X
Descending Sort			X	X					X	X
Ascending Sort	X	X			X	X	X	X		
Include All Attributes	X	X	X	X	X	X	X	X	X	X
Exclude Unused Attributes										
Tallying KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	X	X
Modified KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a		
No Weighting Coefficients	X	X	X	X	X	X	X	X		
Add KG Terms									X	X
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity										
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	86	86	84	84	84	84	88	88	62	34
2 nd Match, AVG% Correct	70	70	68	68	70	70	68	68	52	36
3 rd Match, AVG% Correct	56	56	52	52	52	52	54	54	44	38
5 th Match, AVG% Correct	50	50	30	30	28	28	44	44	46	34
10 th Match, AVG% Correct	4	4	12	12	14	16	12	12	10	16
Average Time/Case	7.59	8.04	12.1	12.1	8.20	8.16	7.10	7.18	36.3	36.2

	IE 041	IE 042	IE 043	IE 044	IE 045	IE 046	IE 047	IE 048	IE 049
City Block Metric	X	X							
Cosine Metric			X	X					
Euclidean Metric					X	X			
Hamming Metric							X	X	
Knowledge Graph Metric									X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120
Independent Attributes	X	X	X	X	X	X	X	X	X
Grouped Attributes									
Metric as-computed	X	X	X	X	X	X	X	X	X
Normalized									
Descending Sort			X	X					X
Ascending Sort	X	X			X	X	X	X	
Include All Attributes	X		X		X		X		X
Exclude Unused Attributes		X		X		X		X	
Tallying KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X	X	X	X	X	
Add KG Terms									X
Multiply KG Terms									
Use ONLY KG Sensitivity									
Use ONLY KG Specificity									
Sections Used in Operation	4	4	4	4	4	4	4	4	4
Step & Prune Operation									
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	88	88	90	90	88	88	88	88	66
2 nd Match, AVG% Correct	70	70	72	72	70	70	70	70	56
3 rd Match, AVG% Correct	58	58	58	58	58	58	58	58	46
5 th Match, AVG% Correct	42	42	44	44	42	42	42	44	34
10 th Match, AVG% Correct	8	8	4	4	8	8	8	10	8
Average Time/Case	29.6	17.1	52.4	18.9	32.8	17.5	28.7	17.0	38.1

	IE 050	IE 051	IE 052	IE 053	IE 054	IE 055	IE 056	IE 057	IE 058
City Block Metric	X	X							
Cosine Metric			X	X					
Euclidean Metric					X	X			
Hamming Metric							X	X	
Knowledge Graph Metric									X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120
Independent Attributes									
Grouped Attributes	X	X	X	X	X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X
Normalized									
Descending Sort			X	X					X
Ascending Sort	X	X			X	X	X	X	
Include All Attributes	X		X		X		X		
Exclude Unused Attributes		X		X		X		X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	X	X
Modified KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a		
No Weighting Coefficients	X	X	X	X	X	X	X		
Add KG Terms								X	X
Multiply KG Terms									
Use ONLY KG Sensitivity									
Use ONLY KG Specificity									
Sections Used in Operation	4	4	4	4	4	4	4	4	4
Step & Prune Operation									
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	86	86	84	84	84	84	88	88	62
2 nd Match, AVG% Correct	70	70	68	68	70	70	68	68	52
3 rd Match, AVG% Correct	56	56	52	52	52	52	54	54	44
5 th Match, AVG% Correct	50	50	30	30	28	28	44	44	46
10 th Match, AVG% Correct	4	4	12	12	16	16	12	12	10
Average Time/Case	7.61	5.53	12.2	6.59	8.27	5.51	7.16	5.22	36.8

	WT 059	WT 060	WT 061	WT 062	WT 063	WT 064	WT 065	WT 066	WT 067	WT 068
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes	X	X	X	X	X	X	X	X	X	X
Grouped Attributes										
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	X	X	X	X	X	X	X	X	X	X
Modified KG calculation										
No Weighting Coefficients										
Add KG Terms	X	X	X	X	X					
Multiply KG Terms						X	X	X	X	X
Use ONLY KG Sensitivity										
Use ONLY KG Specificity										
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	78	90	78	78	66	84	90	84	84	66
2 nd Match, AVG% Correct	60	72	60	60	56	68	72	68	68	56
3 rd Match, AVG% Correct	28	58	28	28	46	54	58	54	54	46
5 th Match, AVG% Correct	10	46	10	10	34	18	46	18	18	32
10 th Match, AVG% Correct	16	8	16	16	8	26	10	26	26	10
Average Time/Case	20.4	22.3	21.1	20.6	38.3	20.8	22.3	21.1	21.0	38.3

	WT 069	WT 070	WT 071	WT 072	WT 073	WT 074	WT 075	WT 076	WT 077	WT 078
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes	X	X	X	X	X	X	X	X	X	X
Grouped Attributes										
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	X	X	X	X	X	X	X	X	X	X
Modified KG calculation										
No Weighting Coefficients										
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity	X	X	X	X	X					
Use ONLY KG Specificity						X	X	X	X	X
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	6	30	6	4	20	76	90	76	76	66
2 nd Match, AVG% Correct	2	8	2	2	16	58	72	58	58	56
3 rd Match, AVG% Correct	2	12	2	2	12	30	58	30	30	46
5 th Match, AVG% Correct	4	4	4	4	4	6	46	8	6	36
10 th Match, AVG% Correct	6	2	6	8	4	14	8	14	14	8
Average Time/Case	20.9	21.8	21.0	20.6	27.0	25.0	23.9	21.9	21.5	39.9

	WT 079	WT 080	WT 081	WT 082	WT 083	WT 084	WT 085	WT 086	WT 087	WT 088
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes										
Grouped Attributes	X	X	X	X	X	X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	X	X	X	X	X	X	X	X	X	X
Modified KG calculation										
No Weighting Coefficients										
Add KG Terms	X	X	X	X	X					
Multiply KG Terms						X	X	X	X	X
Use ONLY KG Sensitivity										
Use ONLY KG Specificity										
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	78	80	68	68	62	78	70	70	66	62
2 nd Match, AVG% Correct	60	68	46	60	52	66	58	52	70	52
3 rd Match, AVG% Correct	42	48	32	38	44	44	52	30	46	44
5 th Match, AVG% Correct	8	38	18	14	46	22	42	18	18	44
10 th Match, AVG% Correct	18	8	4	6	10	22	16	8	8	8
Average Time/Case	8.86	10.1	9.47	9.16	34.8	9.24	10.2	9.49	9.12	35.2

	WT 089	WT 090	WT 091	WT 092	WT 093	WT 094	WT 095	WT 096	WT 097	WT 098
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes										
Grouped Attributes	X	X	X	X	X	X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	X	X	X	X	X	X	X	X	X	X
Modified KG calculation										
No Weighting Coefficients										
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity	X	X	X	X	X					
Use ONLY KG Specificity						X	X	X	X	X
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	76	68	68	72	54	76	86	66	66	70
2 nd Match, AVG% Correct	66	58	48	52	52	60	70	46	50	60
3 rd Match, AVG% Correct	46	46	36	32	42	34	50	34	36	54
5 th Match, AVG% Correct	12	38	20	16	44	14	36	14	16	34
10 th Match, AVG% Correct	10	8	6	8	12	4	10	4	6	4
Average Time/Case	9.20	10.1	9.51	9.12	35.7	9.24	10.1	9.49	9.12	35.8

	WM 099	WM 100	WM 101	WM 102	WM 103	WM 104	WM 105	WM 106	WM 107	WM 108
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes										
Grouped Attributes	X	X	X	X	X	X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation										
Modified KG calculation	X	X	X	X	X	X	X	X	X	X
No Weighting Coefficients										
Add KG Terms	X	X	X	X	X					
Multiply KG Terms						X	X	X	X	X
Use ONLY KG Sensitivity										
Use ONLY KG Specificity										
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	72	58	66	78	60	70	40	68	80	40
2 nd Match, AVG% Correct	56	68	44	52	44	52	38	50	62	40
3 rd Match, AVG% Correct	40	52	36	38	40	34	42	34	52	40
5 th Match, AVG% Correct	10	34	16	18	44	22	42	12	24	30
10 th Match, AVG% Correct	6	22	4	13	12	10	22	12	12	16
Average Time/Case	9.24	10.6	9.43	9.14	38.6	9.20	10.6	9.45	9.14	36.0

	WM 109	WM 110	WM 111	WM 112	WM 113	WM 114	WM 115	WM 116	WM 117	WM 118
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes										
Grouped Attributes	X	X	X	X	X	X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation										
Modified KG calculation	X	X	X	X	X	X	X	X	X	X
No Weighting Coefficients										
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity	X	X	X	X	X					
Use ONLY KG Specificity						X	X	X	X	X
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	70	38	68	78	38	76	86	66	66	70
2 nd Match, AVG% Correct	52	38	44	66	40	60	70	46	50	60
3 rd Match, AVG% Correct	36	42	38	44	40	34	50	34	36	54
5 th Match, AVG% Correct	18	30	12	24	30	14	36	14	16	34
10 th Match, AVG% Correct	8	24	6	18	16	4	10	4	6	4
Average Time/Case	9.22	10.1	9.33	9.14	36.0	9.20	10.1	9.37	9.10	36.1

	SP 119	SP 120	SP 121	SP 122	SP 123	SP 124	SP 125	SP 126	SP 127	SP 128
City Block Metric	X	X								
Cosine Metric			X	X						
Euclidean Metric					X	X				
Hamming Metric							X	X		
Knowledge Graph Metric									X	X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes	X	X	X	X	X	X	X	X	X	X
Grouped Attributes										
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort			X	X					X	X
Ascending Sort	X	X			X	X	X	X		
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	X	X
Modified KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a		
No Weighting Coefficients	X	X	X	X	X	X	X	X		
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity									X	X
Sections Used in Operation										
Step & Prune Operation	X	X	X	X	X	X	X	X	X	X
Minimum # Cases in KB	10	25	10	25	10	25	10	25	10	25
% of KB Cases Kept	10	25	10	25	10	25	10	25	10	25
Best Match, AVG% Correct	38	38	--	--	38	38	38	38	--	--
2 nd Match, AVG% Correct	32	32	--	--	32	32	32	32	--	--
3 rd Match, AVG% Correct	26	30	--	--	26	30	30	26	--	--
5 th Match, AVG% Correct	14	16	--	--	14	16	16	14	--	--
10 th Match, AVG% Correct	4	4	--	--	4	4	4	4	--	--
Average Time/Case	58.0	60.1	--	--	58.0	60.6	58.1	60.5	--	--

Note: The characters "--" in the table indicate instances in which the Cosine Correlation similarity method failed to provide any kind of answer.

	SP 129	SP 130	SP 131	SP 132	SP 133	SP 134	SP 135	SP 136	SP 137	SP 138
City Block Metric	X	X								
Cosine Metric			X	X						
Euclidean Metric					X	X				
Hamming Metric							X	X		
Knowledge Graph Metric									X	X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes										
Grouped Attributes	X	X	X	X	X	X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort			X	X					X	X
Ascending Sort	X	X			X	X	X	X		
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	X	X
Modified KG calculation	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a		
No Weighting Coefficients	X	X	X	X	X	X	X	X		
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity									X	X
Sections Used in Operation										
Step & Prune Operation	X	X	X	X	X	X	X	X	X	X
Minimum # Cases in KB	10	25	10	25	10	25	10	25	10	25
% of KB Cases Kept	10	25	10	25	10	25	10	25	10	25
Best Match, AVG% Correct	28	30	--	--	22	22	32	32	--	--
2 nd Match, AVG% Correct	26	26	--	--	18	20	34	34	--	--
3 rd Match, AVG% Correct	26	30	--	--	16	18	26	30	--	--
5 th Match, AVG% Correct	16	16	--	--	20	18	14	12	--	--
10 th Match, AVG% Correct	0	2	--	--	0	0	2	6	--	--
Average Time/Case	29.2	30.8	--	--	30.2	32.1	31.0	32.2	--	--

Note: The characters “- -” in the table indicate instances in which the Cosine Correlation similarity method failed to provide any kind of answer.

	US 139	US 140	US 141	US 142	US 143	US 144	US 145	US 146	US 147	US 148
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes	X	X	X	X	X	X	X	X	X	X
Grouped Attributes										
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	X	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a		n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X		X	X	X	X	
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity					X					X
Sections Used in Operation	1	1	1	1	1	2	2	2	2	2
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	64	64	64	64	54	84	84	82	82	60
2 nd Match, AVG% Correct	58	58	58	56	42	70	72	72	72	54
3 rd Match, AVG% Correct	48	52	48	48	38	56	56	56	56	44
5 th Match, AVG% Correct	32	40	32	34	30	40	48	40	40	32
10 th Match, AVG% Correct	2	4	2	2	18	14	8	14	14	10
Average Time/Case	7.12	7.53	7.16	7.12	36.6	10.1	11.0	10.2	10.1	37.2

	US 149	US 150	US 151	US 152	US 153	US 154	US 155	US 156	US 157	US 158
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes	X	X	X	X	X	X	X	X	X	X
Grouped Attributes										
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	X	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a		n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X		X	X	X	X	
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity					X					X
Sections Used in Operation	3	3	3	3	3	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	90	90	90	90	66	88	90	88	88	66
2 nd Match, AVG% Correct	70	72	70	70	54	70	72	70	70	56
3 rd Match, AVG% Correct	54	56	54	54	44	58	58	58	58	46
5 th Match, AVG% Correct	40	46	40	40	32	42	44	42	42	36
10 th Match, AVG% Correct	10	4	10	10	10	8	4	8	8	8
Average Time/Case	13.0	13.6	13.0	12.9	37.6	16.7	18.2	16.9	16.5	38.1

	US 159	US 160	US 161	US 162	US 163	US 164	US 165	US 166	US 167	US 168
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes										
Grouped Attributes	X	X	X	X	X	X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	X	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a		n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X		X	X	X	X	
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity					X					X
Sections Used in Operation	1	1	1	1	1	2	2	2	2	2
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	64	52	62	62	44	82	82	84	82	52
2 nd Match, AVG% Correct	56	48	54	52	44	70	58	68	66	44
3 rd Match, AVG% Correct	52	36	46	48	44	58	44	50	52	44
5 th Match, AVG% Correct	32	18	28	32	38	40	30	34	40	46
10 th Match, AVG% Correct	4	16	4	8	8	4	14	4	10	10
Average Time/Case	3.02	3.37	3.00	3.08	36.9	3.82	4.47	3.90	3.63	37.2

	US 169	US 170	US 171	US 172	US 173	US 174	US 175	US 176	US 177	US 178
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes										
Grouped Attributes	X	X	X	X	X	X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	X	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a		n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X		X	X	X	X	
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity					X					X
Sections Used in Operation	3	3	3	3	3	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	90	84	86	86	58	86	84	84	88	70
2 nd Match, AVG% Correct	66	60	64	64	52	70	68	70	68	60
3 rd Match, AVG% Correct	50	34	42	56	48	56	52	52	54	54
5 th Match, AVG% Correct	40	30	36	36	40	50	30	28	44	34
10 th Match, AVG% Correct	8	10	2	14	4	4	12	16	12	4
Average Time/Case	4.41	5.18	4.47	4.55	37.3	5.63	6.69	6.04	5.49	37.6

	IL 179	IL 180	IL 182	IL 183	IL 184	IL 185	IL 186	IL 187	IL 188	IL 189
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	600	600	600	600	600	600	600	600	600	600
Independent Attributes	X	X	X	X	X					
Grouped Attributes						X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	X	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a		n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X		X	X	X	X	X
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity					X					X
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	100	100	100	100	78	98	98	98	100	82
2 nd Match, AVG% Correct	96	96	96	96	74	96	96	96	96	82
3 rd Match, AVG% Correct	90	92	90	90	74	92	88	90	92	80
5 th Match, AVG% Correct	86	86	86	86	70	80	78	80	84	68
10 th Match, AVG% Correct	58	60	58	58	44	56	52	52	56	50
Average Time/Case	77.8	84.0	77.9	76.7	59.4	28.4	33.3	28.8	27.5	59.9

	IL 189	IL 190	IL 191	IL 192	IL 193	IL 194	IL 195	IL 196	IL 196	IL 198
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	480	480	480	480	480	480	480	480	480	480
Independent Attributes	X	X	X	X	X					
Grouped Attributes						X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	X	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a		n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X		X	X	X	X	X
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity					X					X
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	100	98	100	100	70	96	96	96	98	82
2 nd Match, AVG% Correct	92	94	92	92	72	92	92	92	92	78
3 rd Match, AVG% Correct	86	84	86	86	70	90	84	84	80	74
5 th Match, AVG% Correct	76	78	76	76	58	72	70	70	82	62
10 th Match, AVG% Correct	54	58	54	54	44	52	46	42	50	50
Average Time/Case	62.6	69.3	62.8	69.2	50.8	21.1	26.6	21.4	20.4	50.2

	IL 199	IL 200	IL 201	IL 202	IL 203	IL 204	IL 205	IL 206	IL 207	IL 208
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	360	360	360	360	360	360	360	360	360	360
Independent Attributes	X	X	X	X	X					
Grouped Attributes						X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	X	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a		n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X		X	X	X	X	X
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity					X					X
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	100	98	100	98	76	98	98	98	98	78
2 nd Match, AVG% Correct	90	90	90	92	70	90	84	86	90	72
3 rd Match, AVG% Correct	82	82	82	84	66	76	74	74	80	62
5 th Match, AVG% Correct	60	64	60	60	46	64	58	60	62	54
10 th Match, AVG% Correct	50	50	50	48	42	50	38	32	46	38
Average Time/Case	47.2	51.0	48.2	46.8	46.6	16.1	20.2	16.4	15.3	45.2

	IL 209	IL 210	IL 211	IL 212	IL 213	IL 214	IL 215	IL 216	IL 217	IL 218
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	300	300	300	300	300	300	300	300	300	300
Independent Attributes	X	X	X	X	X					
Grouped Attributes						X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	X	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a		n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X		X	X	X	X	X
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity					X					X
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	100	100	88	100	58	100	98	98	100	80
2 nd Match, AVG% Correct	92	92	76	92	56	92	84	86	86	76
3 rd Match, AVG% Correct	82	82	64	82	54	78	74	74	78	72
5 th Match, AVG% Correct	62	60	48	62	42	62	64	64	68	56
10 th Match, AVG% Correct	38	40	20	38	26	38	20	14	38	24
Average Time/Case	38.7	41.8	40.0	38.8	44.5	14.7	18.1	15.4	14.8	47.5

	IL 219	IL 220	IL 221	IL 222	IL 223	IL 224	IL 225	IL 226	IL 227	IL 228
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	240	240	240	240	240	240	240	240	240	240
Independent Attributes	X	X	X	X	X					
Grouped Attributes						X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	X	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a		n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X		X	X	X	X	X
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity					X					X
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	98	98	98	98	74	98	96	96	98	76
2 nd Match, AVG% Correct	86	88	86	86	68	88	82	82	82	76
3 rd Match, AVG% Correct	80	78	80	80	64	72	70	68	76	64
5 th Match, AVG% Correct	58	58	58	58	46	58	62	62	56	54
10 th Match, AVG% Correct	36	40	34	36	30	34	18	18	36	28
Average Time/Case	37.1	40.0	37.8	37.3	50.5	10.0	12.4	10.4	9.90	38.6

	IL 229	IL 230	IL 231	IL 232	IL 233	IL 234	IL 235	IL 236	IL 237	IL 238
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	180	180	180	180	180	180	180	180	180	180
Independent Attributes	X	X	X	X	X					
Grouped Attributes						X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	X	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a		n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X		X	X	X	X	X
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity					X					X
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	96	96	96	96	72	96	94	94	94	76
2 nd Match, AVG% Correct	88	88	88	88	68	82	80	80	82	68
3 rd Match, AVG% Correct	64	64	64	64	52	58	58	56	60	56
5 th Match, AVG% Correct	50	50	50	50	38	44	36	34	46	50
10 th Match, AVG% Correct	18	22	16	18	20	20	18	20	20	26
Average Time/Case	24.7	25.9	24.1	23.1	38.5	7.90	9.27	7.94	7.67	36.0

	IL 239	IL 240	IL 241	IL 242	IL 243	IL 244	IL 245	IL 246	IL 247	IL 248
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	120	120	120	120	120	120	120	120	120	120
Independent Attributes	X	X	X	X	X					
Grouped Attributes						X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	X	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a		n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X		X	X	X	X	X
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity					X					X
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	88	90	88	88	66	86	84	84	88	70
2 nd Match, AVG% Correct	70	72	70	70	56	70	68	70	68	60
3 rd Match, AVG% Correct	58	58	58	58	46	56	52	52	54	54
5 th Match, AVG% Correct	42	44	42	42	36	50	30	28	44	34
10 th Match, AVG% Correct	8	4	8	8	8	4	12	16	12	4
Average Time/Case	15.9	16.9	16.3	15.9	35.2	5.18	6.08	5.31	5.08	33.7

	IL 249	IL 250	IL 251	IL 252	IL 253	IL 254	IL 255	IL 256	IL 257	IL 258
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	90	90	90	90	90	90	90	90	90	90
Independent Attributes	X	X	X	X	X					
Grouped Attributes						X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	X	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a		n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X		X	X	X	X	X
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity					X					X
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	86	88	86	86	50	84	80	80	86	68
2 nd Match, AVG% Correct	68	70	68	68	40	72	64	68	66	62
3 rd Match, AVG% Correct	58	56	58	58	36	50	54	56	52	56
5 th Match, AVG% Correct	30	34	30	30	20	30	18	18	28	24
10 th Match, AVG% Correct	0	4	0	0	8	4	14	4	8	12
Average Time/Case	12.2	13.5	12.6	12.2	28.2	3.96	4.84	4.12	3.98	33.1

	IL 259	IL 260	IL 261	IL 262	IL 263	IL 264	IL 265	IL 266	IL 267	IL 268
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	60	60	60	60	60	60	60	60	60	60
Independent Attributes	X	X	X	X	X					
Grouped Attributes						X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	X	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a		n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X		X	X	X	X	X
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity					X					X
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	78	78	78	78	46	74	70	72	78	62
2 nd Match, AVG% Correct	56	58	56	56	34	54	48	50	48	44
3 rd Match, AVG% Correct	46	46	46	46	28	48	44	48	48	42
5 th Match, AVG% Correct	16	14	16	16	10	16	16	8	18	12
10 th Match, AVG% Correct	0	0	0	0	0	0	2	2	4	10
Average Time/Case	8.55	9.14	8.76	8.67	26.8	2.90	3.33	2.88	2.96	32.9

	IL 269	IL 270	IL 271	IL 272	IL 273	IL 274	IL 275	IL 276	IL 277	IL 278
City Block Metric	X					X				
Cosine Metric		X					X			
Euclidean Metric			X					X		
Hamming Metric				X					X	
Knowledge Graph Metric					X					X
Cases in the Knowledge Base	30	30	30	30	30	30	30	30	30	30
Independent Attributes	X	X	X	X	X					
Grouped Attributes						X	X	X	X	X
Metric as-computed	X	X	X	X	X	X	X	X	X	X
Normalized										
Descending Sort		X			X		X			X
Ascending Sort	X		X	X		X		X	X	
Include All Attributes										
Exclude Unused Attributes	X	X	X	X	X	X	X	X	X	X
Tallying KG calculation	n/a	n/a	n/a	n/a	X	n/a	n/a	n/a	n/a	X
Modified KG calculation	n/a	n/a	n/a	n/a		n/a	n/a	n/a	n/a	
No Weighting Coefficients	X	X	X	X		X	X	X	X	X
Add KG Terms										
Multiply KG Terms										
Use ONLY KG Sensitivity										
Use ONLY KG Specificity					X					X
Sections Used in Operation	4	4	4	4	4	4	4	4	4	4
Step & Prune Operation										
Minimum # Cases in KB	10	10	10	10	10	10	10	10	10	10
% of KB Cases Kept	10	10	10	10	10	10	10	10	10	10
Best Match, AVG% Correct	66	68	66	66	24	64	54	56	68	46
2 nd Match, AVG% Correct	34	30	34	34	12	30	34	36	30	32
3 rd Match, AVG% Correct	12	16	12	12	8	12	12	10	12	12
5 th Match, AVG% Correct	4	0	4	4	0	4	0	6	0	0
10 th Match, AVG% Correct	0	0	0	0	0	0	0	0	0	4
Average Time/Case	5.04	5.47	5.06	5.10	10.5	1.59	1.86	1.59	1.51	24.4

VITA

Claude Ray Mount was born on February 3, 1948, in Parkersburg, West Virginia. After traveling extensively throughout the continental United States and Canada, his family settled in Louisiana in the 1950's. Except for an active duty tour with the military, he has lived there since.

In May of 1971, he received a bachelor of science in physics from Louisiana State University in Baton Rouge, Louisiana. He served as a commissioned officer in the Armor Branch of the United States Army for five years. He resigned his commission with the rank of Captain. In August of 1978, he received a master of science in mechanical engineering from Louisiana State University in Baton Rouge, Louisiana. In 1989, he became a Registered Professional Engineer in the field of metallurgy.

He is married to the former Debra Rae Louks. Between them, they have two daughters, two sons, and two grandsons.

He is currently on the faculty of the Baton Rouge Community College as the Coordinator for the Process Technology Program. He also works as an independent consultant in the areas of failure analysis and expert legal testimony.

He is a candidate for the degree of Doctor of Philosophy in the Engineering Science Department with a major in the field of industrial engineering and a minor in the field of computer science.

DOCTORAL EXAMINATION AND DISSERTATION REPORT

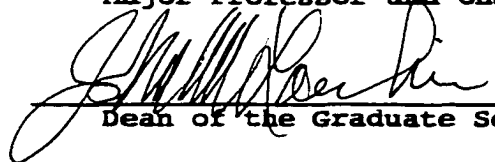
Candidate: Claude R. Mount

Major Field: Engineering Science

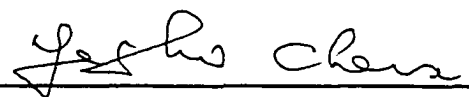
Title of Dissertation: An Intelligent Failure Analysis System

Approved:

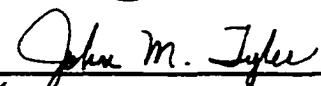

Major Professor and Chairman

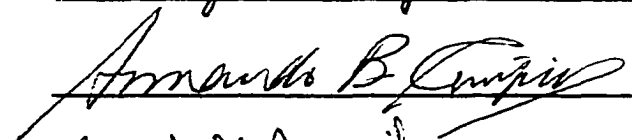

Dean of the Graduate School


EXAMINING COMMITTEE:


James Chen


H. Ghazadeh


John M. Tyler


Annand B. Camp


Mark S. Davidson

Date of Examination:

10/27/00

: