The Fifth Annual AI Systems In Government Conference

Proceedings

The Fifth Annual AI Systems in Government Conference

Editors:
Barry Silverman
Vincent Hwang
Stephen Post

May 6-11, 1990
Washington, D.C.

IEEE Computer Society Press
Los Alamitos, California

Washington • Brussels • Tokyo

Published by

IEEE Computer Society Press
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1264

Cover designed by Jack I. Ballestero

Printed in the United States of America

IEEE Computer Society Press Order Number 2044
Library of Congress Number 90-81057
IEEE Catalog Number 90CH2875-3
ISBN 0-8186-2044-7 (paper)
ISBN 0-8186-6044-9 (microfiche)
ISBN 0-8186-9044-5 (case)
SAN 264-620X

Additional copies can be ordered from:

IEEE Computer Society Press
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1264

IEEE Computer Society
13, Avenue de l'Aquilon
B-1200 Brussels
BELGIUM

IEEE Computer Society
Ooshima Building
2-19-1 Minami-Aoyama,
Minato-Ku
Tokyo 107, JAPAN

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

MESSAGE FROM THE AISIG'90 CONFERENCE CHAIR

During this, the fifth running of the Artificial Intelligence Systems in Government (AISIG) Conference, it is interesting to note the role the conference now plays in the community it serves: those doing research, development, and applications of AI in government settings or under government sponsorship. Last year the conference drew participants from over two dozen states and from six countries. It has been a very rewarding year as Chair to benefit from those participants' notes, calls, work to help prepare this year's conference, and attendance once again. To those of you who are newcomers, welcome to our eclectic community. To the rest of you welcome back.

This year's conference offerings are full to the brim. The tutorial program is fifty percent larger, the vendor exhibits now include third party demonstrations, the technical program has several innovations, and a research workshop has been added for the first time. Also we continue to attract respected guest speakers for plenary and other sessions. To a superb organizing committee, thanks for coming through again and again. Each member truly made my job a pleasure and an honor.

The tutorial program reflects the interests of last year's attendees, as evidenced by encore tutorials on machine learning, AI and databases, and object oriented programming. Moreover the tutorial program reflects the newest interests of the AI community with tutorials on such subjects as knowledge acquisition, case based reasoning, genetic algorithms, hypertext, and natural language interfaces. We are pleased to welcome the tutorial presenters many of whose names I am sure you all recognize such as John Boose, Al Davis, Brian Gaines, and Ryszard Michalski to mention just a few.

For those more interested in sharing research results, in parallel with the tutorial program is the Full-Sized Knowledge Based Systems Workshop. Little did we realize how timely a workshop theme we had chosen as revealed by the large number of high quality submissions received from the likes of Ed Feigenbaum, Doug Lenat, Dick McGammon (Prospector II), and John Sowa, among many others including those from Japan and West Germany. Also, some of the leading government AI sponsors from DARPA, NASA, Navy, NIH, and NSF will be critiquing and discussing the presentations and the state of the art as they see it.

The official opening of the conference begins on Wednesday with the plenary session. We are priviliged this year to have as guest speakers Dr. Ralph Weischedel from Bolt, Beraneck, and Newman plus Dr. Michael Genesereth from Stanford University. The plenary will be chaired by Dr. Oscar Garcia, Past President of the IEEE Computer Society and will include a wrap up summary of the Workshop as well. These distinguished speakers will provide keen insights into the problems and successes of AI systems in government.

The acceptance rate for submitted papers was stringent again this year and I believe the reviewers looked closely at the submissions, offering you only the best written and most mature of the efforts submitted. This year an effort has been made to include several application papers rather than strictly technical developments as that is what past attendees have expressed interest in seeing. In selecting those applications, care has been taken to try and select papers with interesting viewpoints and insights. This set of papers is supplemented by the panels track which offers the insights of many of the leaders of government labs, industry doing government work, and academicians pioneering new techniques of interest to the AISIG community. The technical program has been further broadened to include summaries of other conferences and, as it did last year, vendor exhibits and demonstrations. The overall conference offers an array of topics. We hope you enjoy it.

Barry G. Silverman

Conference Committee

Conference Chair
Barry Silverman
George Washington University

Conference Vice Chairs
Henry Hamburger
George Mason University

Alan Meyrowitz
U.S. Army

Ted Senator
U.S. Navy

Robert Simpson
DARPA

Technical Program Co-Chairs
Vincent Hwang
MITRE

Stephan Post
Advanced Information Technology

Panels Co-Chairs
Ralph Wachter
ONR

Harry Seigel
JACOR

Workshop Co-Chairs
Greg Wenig
IntelliTek, Inc.

Art Murray
McDonnell Douglas

Publicity Co-Chairs
William Carpenter
MITRE

Rey Wong
CACI

Finance Chair
Joe Blumberg
PRC

Standing Committee
Peter Bonasso
MITRE

Jude Franklin
PRC

Kamal Kama
CCGA

Tutorials Chair
Thomas Beckman
IRS

Vendor Contacts
Umar Kahn
U.S. Air Force

Franz Hatfield
Advanced Decision Systems

List of Reviewers

Ted Albert
Jim Antonisse
Thomas Beckman
Jack Benoit
Paul Birkel
Ben Bloom
Joe Blumberg
Pete Bonasso
James Carnes
William Carpenter
H.H. Chen
John Davidson
Chris Elsaesser
Martha Farinacci
Carole Hafner
Henry Hamburger
Jim Hendler
Vincent Hwang
Garry Jacyna
Pam Jordan
Martha Kahn
Umar Kahn
Francois-Michel Lang
Sharon Laskowski
Elaine Lusher
H. Madala
Elaine Marsh
Dick McCammon
Alex Meng
Richard Michalski
Betty Miller
Richard Modjeski
Paul Morawski
Authur Murray
Paula Mutchler
Steven Post
Jim Sanborn
Len Seligman
Harry Siegel
Barry Silverman
Lisa Sokol
Doyle Smith
Eric Summers
Beth Sundheim
Rick Tucker
David Vogel
Ragendra Wall
Ralph Weischedel
Gregory Wenig

TABLE OF CONTENTS

Message from the Conference Chair v

Conference Committee vi

List of Reviewers vii

Expert System Development and Management

Expert Systems Catching on at the Navy Finance Center 2
T. Tubalkain and J.W. Griesser
Bechmark Programs for Evaluating Knowledge-Based Performance Requirements 6
M.M. Hayes and K.L. Rueff
Methods for Selecting Promising Expert System Applications .14
T.J. Beckman

Planning and Scheduling

Operational Planning and Monitoring with Envelopes 24
G.M. Powell and P.R. Cohen
Artificial Intelligence Scheduling for NASA's Hubble Space Telescope 33
M.D. Johnston and G. Miller
A Constraint Directed Approach for Communications Network Design/Planning 40
P.W. Jordan and E.P. Lusher

Expert Database Systems

A Decentralized AI System for Transcript Applications 48
H. Hamburger, A. Lodgher, T. Maney, C. Jardine, and F Siff
Intelligent Information Management in a Distributed Environment 54
C.L. Ortiz and M.A. Bienkowski
A Multi-User, Relationally Based Inference Engine 61
W.A. Carpenter and W.N. McCrosky

Knowledge Acquisition and Management Systems

Knowledge Base Change Monitor 72
K. Michels, D. Jackoway, and B. Perrin
Knowledge Management: An Abstraction of Knowlege Base and Database Management Systems 80
J.D. Riedesel
Technology Transfer Utilizing Automated Knowledge Acquisition Tools 88
G.L. Atkinson

Human Computer Interaction

KINDEN: Knowledge-based INtelligent VLSI Design ENvironment 96
A. Hekmatpour and P. Chau
Services and Information Management for Decision Support 104
Y. Arens
A Knowledge-Based Multi-Modal Interface 112
Y. Arens

Expert Advisory Systems

IWSAS - An Intelligent Interviewing Assistant for Collecting Data on Small Quantity Hazardous Waste Generation 122
D. Coursey, S. Bretschneider, and J. Blair
IAQES: An Expert System for Indoor Air Quality Problem Resolution 127
G.B. Shoom and D.G. Bowen
STAR-Eagle: An Expert Decision Support System for Assessing Tactical Air Capability 134
K.M. German

Natural Language

A Linguistically Sound Approach to Content Analysis of Natural Language Text 142
C.D. Hafner
A Three-Tiered Parsing Approach for Operational Systems 150
A. Kehler, H.R. Blejer, S. Flank, and D. McKee
PUNDIT Apprend le Francais: The PRATTFALL Machine-Translation Module 157
F-M Lang

Knowledge Representation

Diagnostic Reasoning at Multiple Levels of Abstraction 168
B-T.B. Chu and J.A. Reggia
Object-Oriented Inventories for Simulations of Manufacturing Processes 176
R.R. Silbar, P.D. Knudsen, C.A. Hodge, and J.W. Jackson
A Parallel Implementation of a Belief Maintenance System 180
C.L. Ramsey and L.B. Booker

Learning and Knowledge Fusion (And Vision)

Integrating Spatial Data From Text and Vision 188
B.J. Vivier and P.E. Nielsen
Learning by Examples with Uncertainty for the Adaptive Diagnostic System 195
J.F. Blumberg
Motion Estimation of a Rigid Planar Patch: A Robust Computational Technique 199
C-H Lee

Cognitive Modelling, Learning and Problem Solving

A New Level of Language Generation Technology: Capabilities and Possibilities 208
E.H. Hovy
Control and Valuation in Human Cognition 217
T.J. Beckman
Investigation of a Simulator-Trained Machine Discovery System 225
B.G. Silverman, M. Hieb, V. Kosnett, and L. Wu

Author Index 233

EXPERT SYSTEM DEVELOPMENT AND MANAGEMENT

Expert Systems Catching on at the Navy Finance Center

Toomas Tubalkain and John W. Griesser

Navy Finance Center
Applications Technology Division
Cleveland, OH 44199-2055

Abstract

This paper discusses the evolution of the Expert System Program at the Navy Finance Center from its conception to its successful establishment. The management decisions and actions leading to a successful pilot project in Former Spouse Survivor Benefit Plan Adjudication, the training of domain experts to create their own expert systems, creating an environment that will promote the success of expert systems technology, and a cost benefit analysis of an expert system project are covered. The objective of this paper is to show how easily expert system technology can be introduced into the culture of an organization.

INTRODUCTION

How do you introduce expert systems into an organization without spending a great deal of time and money for research, development, and prototyping? How do you measure the expert system benefits in tangible and intangible savings? The Navy Finance Center's Systems Department came up with a four step process that drastically cut start-up time while keeping costs to a bare minimum. Better yet, less than a year after introducing the expert systems concept here, the first system has already achieved proven productivity gains, improved the quality of employee work-life, and inspired twenty additional expert system projects.

STEP ONE: SELL THE IDEA

The Systems Department began promoting the concept of expert systems during the summer of 1988. At that time, there was no in-house expertise, no budget, and no program. A video training program on Artificial Intelligence (AI) applications was obtained, and shown to Navy Finance Center managers to give them an appreciation of what the technology could and could not do. During this training, the Director of the Retired Pay Department realized that he had a problem that was perfectly suited to expert system development. Adjudication of Former Spouse Survivor Benefit Plan (FSSBP) cases, one of his most complex functions, was entirely dependent on a single subject matter expert, and that person was becoming increasingly swamped with a backlog of cases. His critical need in this area and the ready adaptability of AI technology to this type of problem, raised expert systems development to a high-priority project. The Director of the Retired Pay Department became the "Champion" of the FSSBP development effort.

STEP TWO: TAKE A TEAM APPROACH

From the beginning, the development of an expert system to meet the needs of the Retired Pay Department was a team project, involving not only departments of the Navy Finance Center, but the Air Force Logistics Center in Dayton, Ohio as well. The Air Force Logistics Center Artificial Intelligence Program Office provided a copy of the M.1 expert system shell and a two week Knowledge

CH2875-3/90/0000/0002$1.00

Engineering/M.1 Programming course. With this background, development was started in November 1988, with a scheduled completion date of March 3, 1989. One programer from the data base administration area was assigned part-time as a Knowledge Engineer (expert system developer/programer) to work with the FSSBP expert (Domain Expert) in Retired Pay to develop this system. Development of the system followed a prototyping methodology. The Knowledge Engineer delivered a system in progress to the Domain Expert at the beginning of each prototyping session. The Domain Expert was able to test the system throughout the development and by the time the knowledge base was complete, felt confident that all rules had been accurately captured. The knowledge base was essentially complete by the end of January 1989. The remaining month was spent making the system user friendly, writing a user's manual, and developing a training program for the adjudication staff.

The FSSBP system was designed to determine the eligibility of a member or former spouse or both to elect former spouse coverage under the Survivor Benefits Plan. The system directs a clerk to send a form letter(s), explains how to update the member's Retired Pay account, and how to compute FSSBP deductions. The explanation for each ineligibility determination contains a reference to the paragraph of the base statute or amendment that applies to the case. A printed copy of each consultation is produced that serves as documentation of the decision and is kept in the member's case jacket.

The system went into production on 7 March 1989. The backlog in FSSBP had grown to 380 cases by this time, but within a month had been reduced to less than 50 cases. Response time to answer a former spouse inquiry dropped from over four months to less than four weeks. With the work being handled by clerical employees, the need to hire additional high graded Military Pay Specialists to work the cases was eliminated. The clerical employees are pleased to be working more complex and interesting cases and the Domain Expert is delighted to get rid of his backlog and be able to concentrate on very difficult cases.

STEP THREE: SHARE THE SUCCESS

As soon as the FSSBP expert system was put into production, on 7 March 1989 a series of system demonstration briefings were begun to reinforce management awareness of the value of this technology. These briefings were carefully choreographed and included presentation of the concept and development by the knowledge engineer, demonstration of the product by the domain expert, and a summary of the benefits by the Retired Pay Department Director. This cast was capable of answering all questions and provided three distinct viewpoints of expert system usage. Based on the encouraging results, demand for this technology in other parts of the command became overwhelming. The Systems department staffed an Expert Systems Group, procured a site license for the M.1 expert system shell, and brought in an instructor from the Air Force Logistics Command to train analysts and programers (including the FSSBP Domain Expert) from all Navy Finance Center departments. New proposals for expert system development are coming in rapidly enough to ensure new projects are continuously available.

The following benefits were publicized to both management and staff in a series of briefings, group discussions and employee newspaper articles.
The system produced saved operational costs, avoided future operational costs and had numerous intangible benefits. The backlog was reduced and the response time to answer a former spouse inquiry dropped to less than four weeks. The FSSBP system ensured that legal requirements were met and were

consistently applied. The clerical employees learned additional skills, such as reading legal documents, worked more complex and interesting cases and had increased job satisfaction. This was the first PC based application for the clerks. The PC user interface is much more friendly than the mainframe interface and nearly all of the clerks were eager to begin using the PC. The system shifted the work from the expert to clerical employees thus eliminating the need to hire, train, and develop another "Expert". The Domain Expert is now able to take vacations and not work on Saturdays. The risk of him being unavailable is reduced. The Expert could now work on the extremely complex cases and other parts of the FSSBP Expert System Project. A planned enhancement to the system is to handle the adjudication of election changes, in addition to determining election eligibility. The Domain Expert has now assumed the duties of maintaining the FSSBP Expert System and in his spare time is developing a new expert system to help compute Social Security Offsets.

STEP FOUR: SHARE THE TECHNOLOGY

Sharing the success of the first expert system project led to a large demand for expert systems by operational management. To satisfy this demand, an instructor from the Air Force Logistics Command AI Program Office was brought in and a two week long in-house training course in knowledge engineering and M.1 programing was provided to twenty functional user personnel. Each of the trainees was required to bring a project to class that would benefit the operations of the Navy Finance Center. The projects ranged from five day efforts to two month projects. In an effort to provide the new knowledge engineers with an environment in which they could succeed, a technical support team was formed in the systems department. However, most knowledge engineers were able to bring up their systems with minimal technical assistance.

Expert Systems In Production:

*Deceased Account Settlement -- Assist clerical personnel in closing a deceased Navy member's payroll account.

*LAN Troubleshooting -- Help technical person determine the cause of LAN failure.

*Zenith-248 Hardware Diagnostics -- Aid end user to diagnose PC problems.

*Successor Check adjudication -- Lead clerical personnel through process of determining when to send out a new check for one that has been reported lost.

*Dual Compensation Determination -- Determine when a retired Navy member who is working is exempt from dual compensation and pay cap deductions.

A month after completion of the class, an Expert Systems Users Group was formed at NAVFINCEN. The group is open to knowledge engineers, domain experts, and interested management. The goals of the group are to share the successes, transfer procedural and technical information, and lead the development of Expert System Quality Standards. Each expert system completed at NAVFINCEN gets demonstrated at a users group meeting. This ensures that the knowledge engineers know what is going on with expert systems and aids them in the selection, development, and design of new systems.

THE BOTTOM LINE

Former Spouse Survivor Benefit Plan Cost Benefit Analysis:

DEVELOPMENT COSTS:

1 Knowledge Engineer (4 months half-time)	$ 7,230
1 Domain Expert (4 months half-time)	$ 5,453
1 Site license for the Expert System Shell	$ 5,000
TOTAL COSTS	$17,683

BENEFITS: (annual)
Reduction in need for overtime.. $ 2,800
Elimination of need to hire
two Military Pay Specialists...$56,086
TOTAL...........................$58,886

SAVINGS:
YEAR 1 ($58,886-$17683).........$41,203

CONTINUING ANNUAL SAVINGS........$58,886

TANGIBLE BENEFITS:
1) Reduction of backlog from 380 cases to less than 40 cases.
2) Improvement in turn-around time from over 4 months to two weeks or less.
3) Audit trail, due to print out of consultation.

INTANGIBLE BENEFITS:
1) Risk reduction. If the only expert becomes unavailable, processing can continue.
2) Consistent application of Former Spouse Survivor Benefit Plan laws.
3) Improved job satisfction for both the Military Pay Specialist and Adjudicators.
4) More Timely Management Information on Backlogs
5) Additional Information - Case Status
6) Legal Requirements Standardized
7) Processing Standardized
8) Image Improved with Customers

Note: The savings for the first year would be higher if the cost of the M.1 software were amortized over many projects, rather than included in the cost of this first project.

CONCLUSIONS

The keys to success in the Navy Finance Center's expert system program are management awareness, project selection, and project team composition. The video training was enough to convince NAVFINCEN management of the potential benefits of expert systems but a project champion must be courted from the upper management of the users organization. The "Champion" is capable of convincing end user management of the benefits of the new technology. His voice is much more convincing than a technician's. User and expert system management then chose projects that were suitable, feasible and stood a good chance of becoming successful expert systems. Project teams consist of two or three people, a Knowledge Engineer and one or two Domain Experts. Project teams worked together in prototyping sessions and knowledge acquisition sessions. This makes projects easy to manage and ensures that expert system development rather than project management is the primary project team activity.

BENCHMARK PROGRAMS FOR EVALUATING KNOWLEDGE-BASED PERFORMANCE REQUIREMENTS

Mary M. Hayes Karen L. Ruoff

Artificial Intelligence Laboratory
McDonnell Douglas Electronic Systems Company
Huntington Beach, California 92647

Abstract

In order to effectively develop full-sized knowledge-based systems, a set of standard benchmark programs and a methodology for evaluating knowledge-based system performance are needed. Currently used benchmark programs such as the "Farmer's Dilemma" and "Monkeys and Bananas" are too trivial and unrepresentative of real applications to provide useful results. In this paper, we present a set of benchmark programs that have many of the characteristics of full-sized, complex knowledge-based systems. Furthermore, we present a methodology for characterizing knowledge-based system software and evaluating its performance.

Introduction

As we move to the development of larger, more mature knowledge-based systems, the expected performance of such systems is becoming a major issue. Commonly used knowledge-based system benchmark programs such as the "Farmer's Dilemma" and "Monkeys and Bananas" are too small and unrepresentative to provide insight into the performance of a full-sized knowledge-based system. Performance statistics that we examined for these benchmark programs revealed that most of the processing time was due to initialization of the knowledge-based tool data structures. In order to obtain more meaningful results, we selected a set of three benchmark programs from Carnegie Mellon University and a benchmark program in use at Artificial Intelligence Technologies, Incorporated (see Table 2-1), which do not exhibit this problem.

An analysis of the coding style and run-time performance of this set of four benchmark programs has allowed us to better understand the resource utilization of knowledge-based systems, including use of the memory, scheduler, instruction set, etc. It has also allowed us to select original equipment manufacturer (OEM) hardware that best meets the knowledge-based tool needs and helped us to identify areas in which processors could be tailored to give better performance. These areas include large cache memory and machine instruction set extensions that perform extensive branching more efficiently. By using benchmark programs large enough to stress many aspects of the knowledge-based tool, we have been able to make a more confident and informed choice of the best knowledge-based tool and hardware platform for development of full-sized knowledge-based systems.

Knowledge-Based System Performance Analysis

In order to accurately assess the performance of knowledge-based system software and hardware, performance characteristics of the benchmark programs must be defined. Work done previously at Carnegie Mellon University [1] identified characteristics of the application program that affect the performance of a knowledge-based system. We developed an analysis tool to gather statistics on the key performance indicators for the set of four proposed benchmark programs. We found that this set of benchmark programs varied sufficiently in the amount and complexity of matching being done in the match cycle (see section 2.2) to test the software and hardware in diverse conditions. The benchmark programs were written in OPS5 and could easily be rewritten using other knowledge-based tools such as Inference Corporation's ART or IntelliCorp's KEE to evaluate the performance of other knowledge-based tools.

Overview of Benchmark Programs

The set of benchmark programs we are using is shown in Table 2-1. The "Monkeys and Bananas" program is also included for comparison purposes.

Table 2-1. Set of Five Benchmark Programs.

PROGRAM NAME	DESCRIPTION	NUMBER OF RULES
Weaver	A VLSI routing program by Rostom Joobbani [2, 3]	637
Rubik	A program that solves the Rubik's cube by James Allen [2]	70
Tourney	A program that assigns match schedules for a tournament by Bill Barabash from DEC [2]	17
Big Conflict	A program designed by Artificial Intelligence Tech., Inc. to stress the Rete matching algorithm	2
Bananas and Monkeys	A program which gives instructions to a monkey that will allow the monkey to grab a bunch of bananas	20

"Weaver" is significant because it is a full-sized knowledge-based system [2,3]. "Rubik" performs a large number of changes to working memory causing many invocations of the match cycle. This characterizes a condition in which the Rete matching algorithm is not designed to perform optimally [4]. The processing requirements of "Tourney" are comparative to many larger knowledge-based systems, though the number of rules is small. The "Big Conflict" program has many partial matches that lead to a lengthy match process each time the match cycle is performed.

Role of the Match Process In The Rete Matching Algorithm

During execution, the inference engine of a knowledge-based tool determines which instantiation of a rule should fire next. The functions performed by the inference engine can be broken into three steps with the functional definitions varying somewhat between knowledge-based tools. The three steps performed are as follows:

1. Match: This step takes place after each change to working memory. It determines if the change makes any new instantiations of the rule eligible to fire. New instantiations of the rule are added to the conflict set (agenda). It also determines if the change means that one or more instantiations should be removed from the conflict set.

2. Conflict-Resolution: This step chooses a single rule that is eligible to fire when more than one instantiation is in the conflict set.

3. Act: In this step, actions on the right hand side of the rule instantiation chosen during conflict-resolution are fired.

The match cycle is the most computationally intensive of these three cycles for most applications. Each time a change is made to working memory because of a right hand side action, the match cycle is performed. Depending on the characteristics of the rules and the size of working memory, each match cycle can be quite time consuming.

Description of the Rete Matching Algorithm

The algorithm used in OPS5 and many other knowledge-based tools to perform the match cycle is the Rete matching algorithm. Other matching algorithms exist, but the Rete algorithm is used by most commercially available knowledge-based tools. Therefore, the performance indicators noted for the set of benchmark programs largely reflect their effect on the performance of the Rete matching algorithm.

When performing the match cycle, the Rete network utilizes a network of nodes compiled from the left-hand sides of rules. The network is made up of different types of nodes depending on the types of tests being performed. Changes to working memory are filtered through the network of nodes in the form of tokens. The four major types of nodes in the Rete network are as follows:

1. Constant-test nodes: These nodes test whether a constant value test for an attribute in a condition element has been met.

2. Memory nodes: The function of these nodes is to maintain partial match

information to reduce the processing required in the match cycle.

3. Two-input nodes: Nodes of this type check for consistent matching between a condition element and a previous condition element within a rule.

4. Terminal nodes: There is one terminal node for each rule in the Rete network. A token that reaches a terminal node in the Rete network is entered into the conflict set and is eligible to fire.

Performance Characteristics of the Rete Matching Algorithm Studies have shown that current implementations utilizing the Rete matching algorithm spend approximately 90 percent of their processing time in the match cycle [1]. Therefore, for a large knowledge-based system in which fast run-time performance is essential, it is important to examine the performance of the match cycle under diverse conditions. The effect of the matching algorithm with respect to processing time can only be seen by running a benchmark program computationally large enough to stress the Rete matching algorithm (by frequency of invocation or complexity of matching).

Characteristics of Benchmark Programs

The benchmark programs we used were chosen for their variance in effect on the Rete matching algorithm. A summary of the code characteristics we have analyzed and their effect on the performance of the Rete matching algorithm are shown in Table 2-2. Areas considered in this and the next two sections were presented in a similar style for a set of benchmark programs used in Anoop Gupta's PhD thesis [1]. Code characteristics are drawn from an analysis of the source code of the application program. Examining characteristics of the source code can lend insight into the expected run-time performance of the knowledge-based tool for different application programs.

Table 2-2. Code Characteristics of Benchmark Programs.

CHARAC-TERISTIC	EFFECT ON KNOWLEDGE BASE PERFORMANCE
Condition Elements Per Production	Predicts likelihood of an instantiation of a rule entering the conflict set
Actions Per Production	Indicates that working memory could be changing rapidly
Attributes Per Condition Element	Number of tests which must be passed in order to satisfy a condition element
Variables Per Condition Element	When this number is large compared to the number of attributes per condition element, it indicates a large number of partial matches
Action Types	Comparing this number with the actions per production gives an indication of the rate at which working memory is changing

Condition Elements Per Rule The number of condition elements per rule for the set of five benchmark programs is shown in Figure 2-1. Both positive and negative condition elements were included. The number of condition elements in a rule reflects the set of situations for which the rule is likely to enter the conflict set.

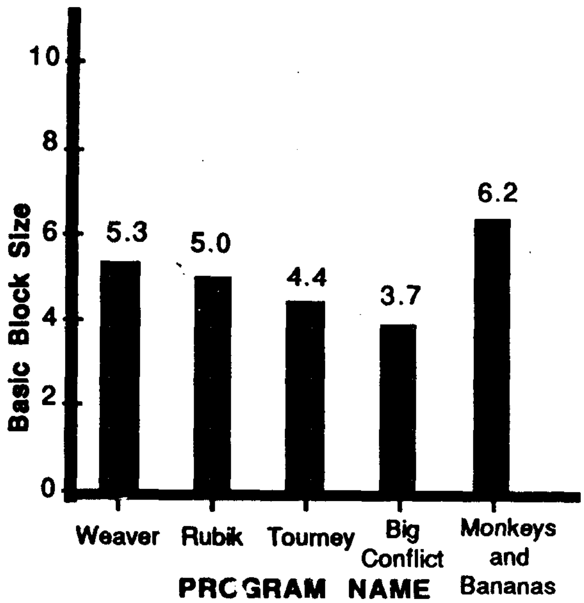

Figure 2-1. Condition Elements Per Rule.

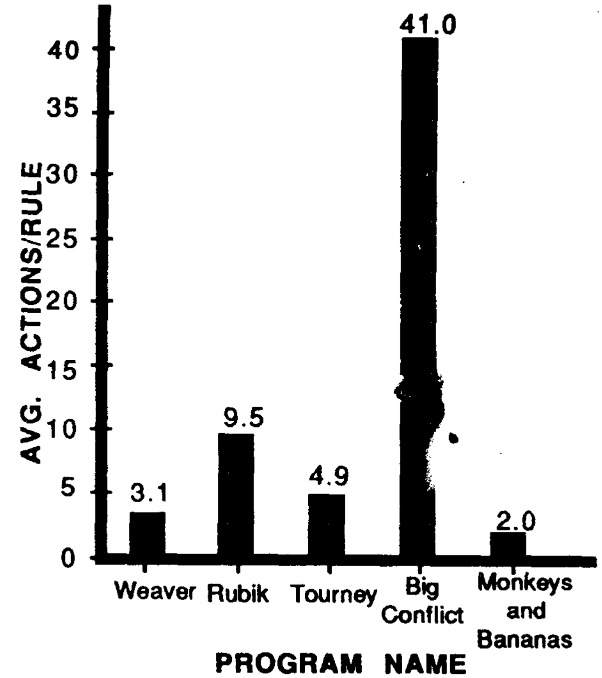

Figure 2-2. Actions Per Rule.

Actions Per Rule Figure 2-2 shows the average number of actions per rule. When the number of actions per rule is large, it reflects the fact that the contents of working memory are changing quite rapidly. A large number of working memory changes implies a correspondingly large number of match cycles.

Attributes Per Condition Element Figure 2-3 shows the number of attributes per condition element. The number of attributes reflects the number of tests (constant and variable) that a new working memory element must pass in order to satisfy the condition element.

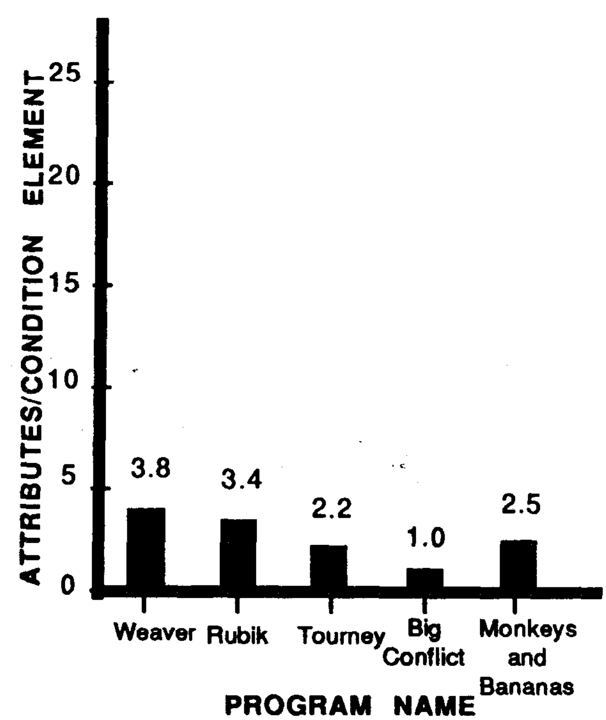

Figure 2-3. Attributes Per Condition Element.

Variables Per Condition Element The number of variables per condition element is shown in Figure 2-4. When the number of variables per condition element is large with respect to the number of attributes per condition element