DAA/HQ

[redacted]

p. 353

Proceedings: Computer Science and Data Systems Technical Symposium Volume 1

NASA
ANALYTIC (1+19)
IN-CAT. 61 5 TM
DATE
OVERRIDE

Date for general release August 1987

Proceedings of a symposium held at the
Xerox Training Center
Leesburg, Virginia
April 16-18, 1985

(NASA-TM-89285) PROCEEDINGS: COMPUTER SCIENCE AND DATA SYSTEMS TECHNICAL SYMPOSIUM, VOLUME 1 (NASA) 353 p Avail: NTIS HC A16/MF A01 CSCL 09B G3/61

N87-29124
--THRU--
N87-29143
Unclas
0093257

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Proceedings: Computer Science and Data Systems Technical Symposium, Volume 1		5. Report Date August 1985 6. Performing Organization Code RC
7. Author(s) Ronald L. Larsen and Kenneth Wallgren, Co-Chairmen		8. Performing Organization Report No.
9. Performing Organization Name and Address Information Sciences and Human Factors Division NASA Office of Aeronautics and Space Technology Washington, D.C. 20546		10. Work Unit No. 11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		13. Type of Report and Period Covered Conference Publication 14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Progress reports and technical updates of programs being performed by NASA Centers are covered. Presentations in viewgraph form, along with abstracts, are included for topics in three categories: computer science, data systems, and space station applications. The Symposium was held at the Xerox Training Center, Leesburg, Virginia, from April 16 to 18, 1985. The Symposium schedule and a list of participants are included.

ORIGINAL PAGE IS OF POOR QUALITY

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Computer Science Data Systems Space Station	[redacted] Subject Category 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	366	

N-305

Computer Science and Data Systems Technical Symposium

Table of Contents

Volume 1

Volume 2

Computer Science and Data Systems Technical Symposium

Table of Contents (Cont.)

Volume 2 (Cont.)

Section 2: Data Systems Program (Cont.)

INTRODUCTION

The Computer Sciences and Data Systems Technical Symposium was held to respond to the communications challenges posed by the rapidly advancing technical arena surrounding NASA personnel. This was the second meeting in what will be periodic gatherings. The intended purpose of these meetings is to bring NASA people together to present their progress, to air their thinking and, in general, to discuss the nature and results of their work within the agency on a wholly technical level. These meetings are not intended as a forum for program reviews, budget presentations or advocacy hearings.

NASA personnel have long been recognized as prolific contributors to the journals of technical societies and organizations within the aerospace community. Symposiums such as this one, organized to improve the interchange of technical information and understanding within NASA, have resulted in valuable connections. These meetings will be continued. The Proceedings of the April 1985 Computer Sciences and Data Systems Technical Symposium are presented to provide a legacy for the latest gathering and a springboard to the next.

N87-29125

p.20

TOWARDS AN ASSESSMENT OF FAULT-TOLERANT DESIGN PRINCIPLES FOR SOFTWARE

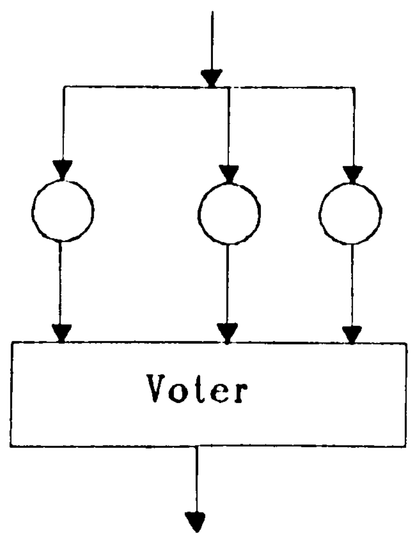
Dave E. Eckhardt, Jr.
NASA Langley Research Center

NASA
Computer Science / Data Systems
Technical Symposium

April 16-18, 1985

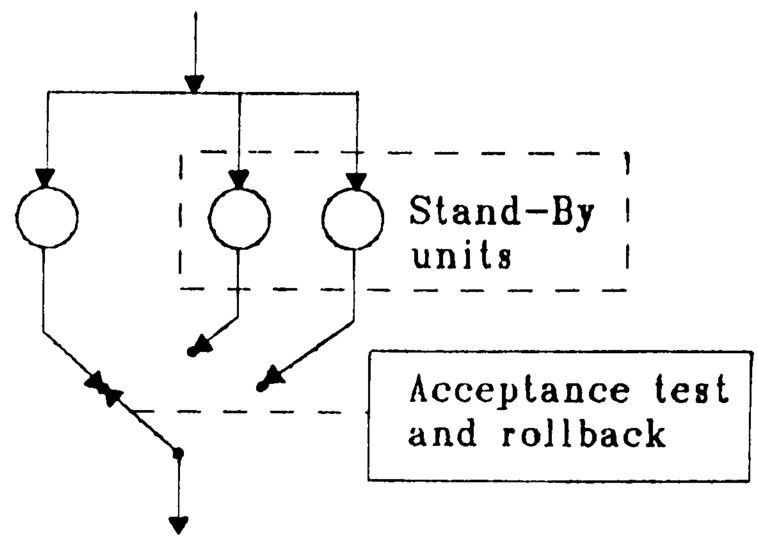
FAULT–TOLERANT SOFTWARE

Dissimilar, redundant software structured to reduce the probability of system failure due to software faults. Techniques address error detection and isolation, system recovery, and continued service.



HARDWARE: N–Modular Redundancy

SOFTWARE: N–Version Programming



Stand–By Sparing

Recovery Block

FAULT-TOLERANT SYSTEMS BRANCH: SOFTWARE RESEARCH SUPPORTED BY THE COMPUTER SCIENCE PROGRAM

DESIGN:

OPERATING SYSTEMS – Investigation of fault-tolerant software for SIFT operating system (Brunelle, NASA LaRC)

SYSTEM THEORY – Integration of control system-theoretic FDI Techniques (Caglayan, Charles River Analytics)

NEW TECHNOLOGY – Automatic generation of FDI software (Wild, ODU)

RELIABILITY ASSESSMENT:

EXPERIMENTAL
- Investigation of independence assumption (Knight, UVA & Leveson, UCI)
- Investigation of test procedures for multi-version software (McAllister, NC State)

MODELS
- Assessment of generic software reliability models (Migneault, NASA LaRC)
- Development of basis for analysing strategy of software redundancy (Eckhardt, Lee, NASA LaRC)

THE BASIS FOR FAULT–TOLERANT SOFTWARE

- Software faults are assumed to be "independent" so that errors will be randomly distributed among replicate codes.
- Currently, there are research efforts to analyse this fundamental assumption.

HOWEVER

- Independence is not strictly needed for fault–tolerant software to be effective at reducing failure probability.
- It is a mathematically convenient assumption used to project the reliability of software fault–tolerant structures.

ASSESSING FAULT–TOLERANT SOFTWARE

If software errors are not randomly distributed, what is the impact on reliability?

Current state of the art does not provide answers.

Consider:

(1) Is an N–Version system of highly reliable components always more effective at reducing failure probability than a single version of software (on average)? If not, what causes this?

(2) What are the effects of different intensities of coincident errors on software redundancy?

(3) What is the effect of increasing N? Is there a limit to the effectiveness of software redundancy?
Might an optimum value of N exist?

(4) Does the independence model provide a valid estimate of the failure probability of an N–Version system?

(5) Under what condition does independence hold?

GOAL OF CURRENT RESEARCH

Assess strategy of software redundancy

- population concepts
- sampling, sample size
- inference

AS OPPOSED TO:

Assessing an instance of fault-tolerance

- decide number of versions
- develop
- measure

QUANTITIES DESCRIBING COINCIDENT ERRORS MODEL

INPUT SPACE

COMPONENT POPULATION

	$\{ x_1$	x_2	x_3	$\ldots$	$x_k \}$	
C_1	0	1	0	$\ldots$	0	$\xrightarrow{v_1(x)}$
C_2	1	0	0	$\ldots$	1	
$\vdots$						
C_l	0	1	0	$\ldots$	0	
$\vdots$	$\downarrow$					
	$\theta(x_1)$					

CONDITIONAL FAILURE PROBABILITY

$$Q(F) = \int v_1(x)\, dQ$$

$$= \Pr\{ C_l \text{ fails} \}$$

Q = USAGE DISTRIBUTION

AVERAGE FAILURE PROBABILITY

$$E\left[\int V(x)\, dQ\right] = \int \theta(x)\, dQ$$

INTENSITY FUNCTION

$$\theta(x) = \Pr\{ V(x) = 1 \}$$

$$E[\, V(x)\,] = \theta(x)$$

INTENSITY DISTRIBUTION

$$G(y) = \int_{\{ x \,:\, \theta(x) \leq y \}} dQ$$

N-VERSION WITH MAJORITY VOTE

		IID INPUT			
RANDOM SAMPLE	$v_1(x)$	0	0	1	...
	$v_2(x)$	0	1	0	...
	$\vdots$				
	$v_N(x)$	0	0	0	...

SCORE FUNCTION

$$v(x) = \sum_{l=m}^{N} \sum_{\substack{\text{all} \\ \text{permutations}, i}} v_{i(1)}(x) \ldots v_{i(l)}(x) \, [\, 1 - v_{i(l+1)}(x) \,] \ldots [\, 1 - v_{i(N)}(x) \,]$$

$$P_N = E[\int V(x) \, dQ \,]$$

COINCIDENT ERRORS MODEL

Under the conditions that:

(1) components are selected from a random sample

(2) inputs are selected from a common distribution

$$p_N = \int \sum_{\ell=m}^{N} \binom{N}{\ell} \theta(x)^{\ell} \left[1 - \theta(x) \right]^{N-\ell} dQ$$

$\theta(x)$ = Intensity Function

Q = Usage Distribution

$$= \int h(y;N)\, dG$$

$$h(y;N) = \sum_{\ell=m}^{N} \binom{N}{\ell} y^{\ell} [1-y]^{N-\ell}$$

$$G(y) = \int_{\{x \,:\, \theta(x) \leq y\}} dQ$$ = Intensity Distribution

A DISCRETE INTENSITY DISTRIBUTION

Suppose $\theta(x) = \theta_i$ for $x \in A_i$

Where $A_1, A_2, \ldots, A_r$ is a partition of Ω

Indexing so that

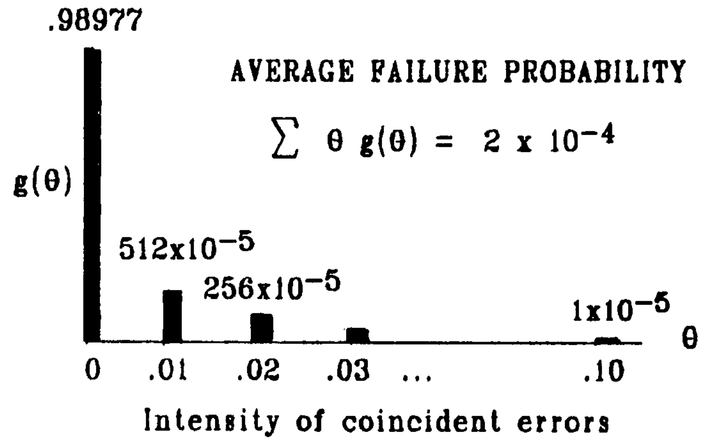
$$0 = \theta_1 < \theta_2 < \ldots < \theta_r < 1$$

$$G(y) = \sum_{\{ i \,:\, \theta_i \leq y \}} Q(A_i) \qquad -\infty \leq y \leq \infty$$

e.g

AVERAGE FAILURE PROBABILITY

$$\sum \theta \, g(\theta) = 2 \times 10^{-4}$$



(e.g. on .001% of inputs, expect 10% of population to produce error)

EFFECTS OF COINCIDENT ERRORS

Under what condition does independence hold?

Under the assumption of a constant intensity the Coincident Errors Model implies the Independent Errors Model.

This constant is the average component failure probability (also the mean of the Intensity Distribution).

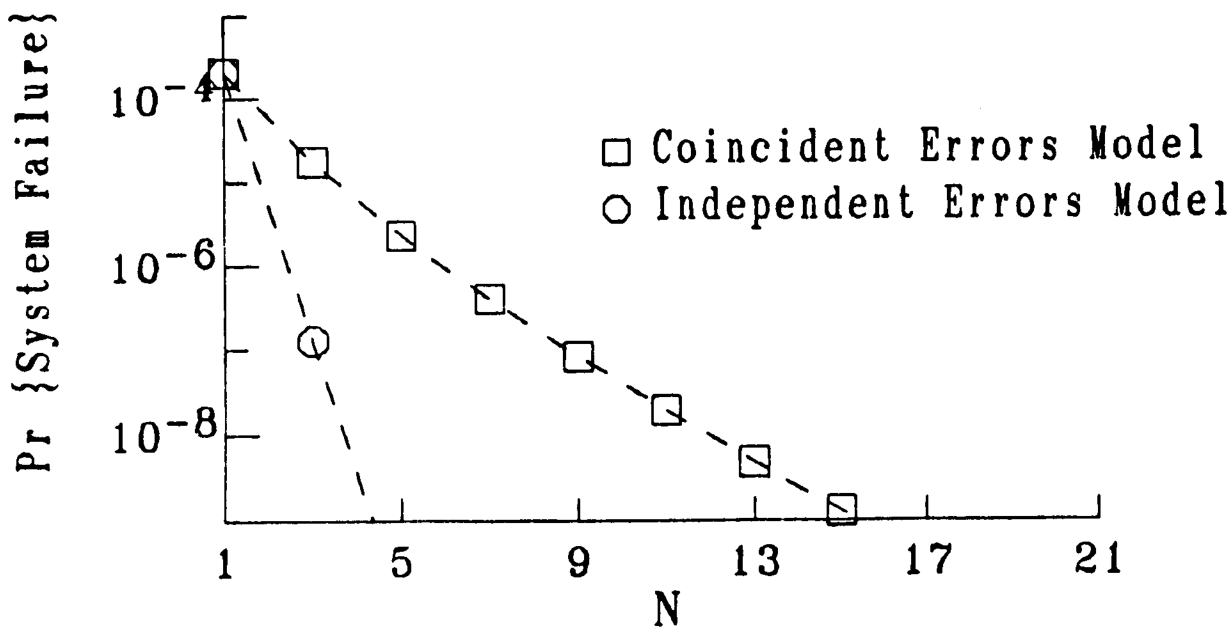
$$P = \int \Theta(x)\, dQ = \int y\, dG(y).$$

EFFECTS OF COINCIDENT ERRORS

Does the Independence Model give a valid estimate of failure probability?

e.g

θ	$g(\theta)$
0	.98977
.01	.00512
.02	.00256
⋮	⋮
.10	.00001

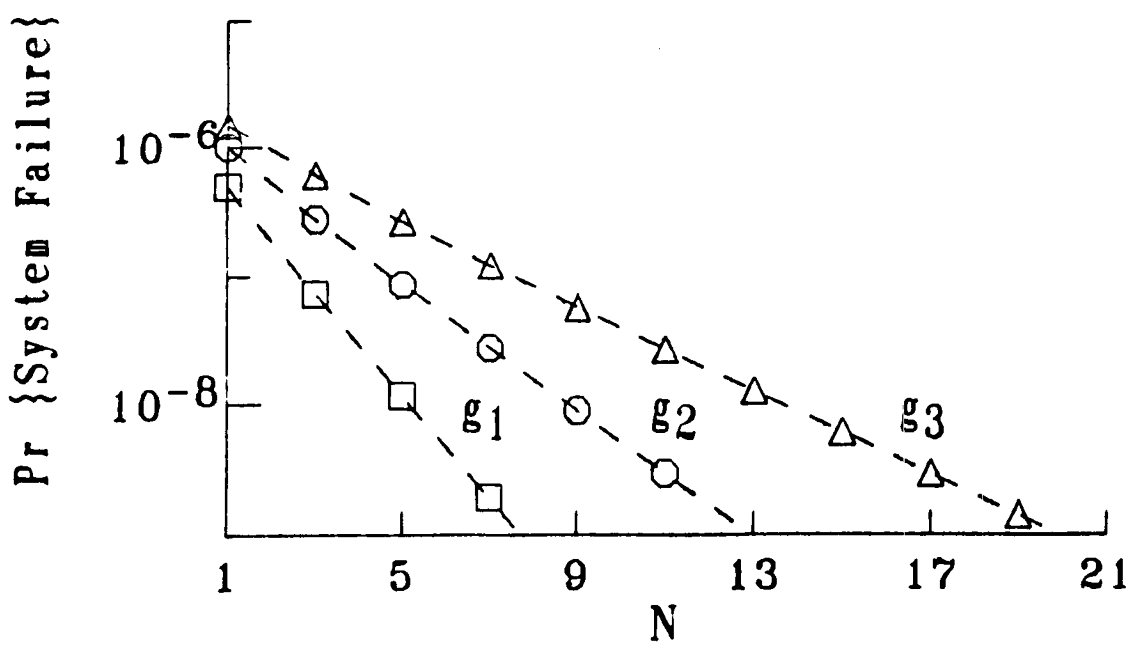


EFFECTS OF COINCIDENT ERRORS

What is the effect of shifting intensity mass probability to the right?

e.g.

θ	$g_1(\theta)$	$g_2(\theta)$	$g_3(\theta)$
0	.99999	.99999	.99999
.05	.00001		
.10		.00001	
.15			.00001

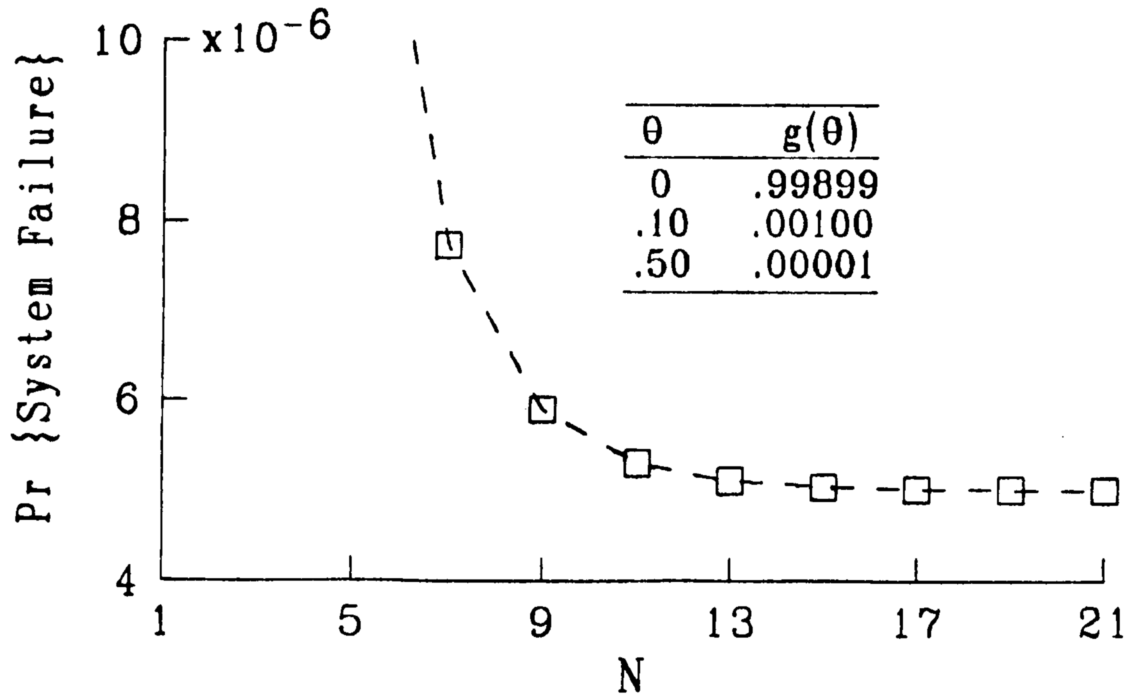


EFFECTS OF COINCIDENT ERRORS

Is there a limit on the effectiveness of redundancy?

Effectiveness limited by mass distributed along inteval $.5 \leq \theta \leq 1$

$$\lim_{N \to \infty} P_N = .5 \left[G(.5^+) - G(.5^-) \right] + \int_{.5^+}^{1} dG(\theta)$$



θ	g(θ)
0	.99899
.10	.00100
.50	.00001

EFFECTS OF COINCIDENT ERRORS

Under what condition is an N–Version strategy better than a single version choosen at random?

We say an N–Version strategy is better if $P_N < P$

Where

$$P_N = \int \sum_{l=m}^{N} \binom{N}{l} \theta(x)^{l} \, (1 - \theta(x))^{N-l} dQ$$

$$= \int h(y;N) \, dG(y)$$

and

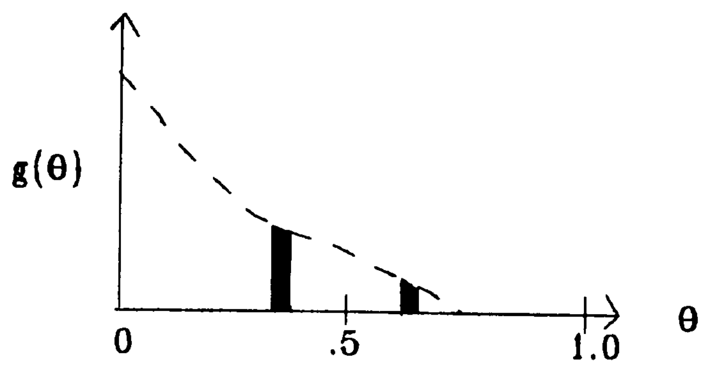
$$P = \int \theta(x) \, dQ = \int y \, dG(y)$$

EFFECTS OF COINCIDENT ERRORS

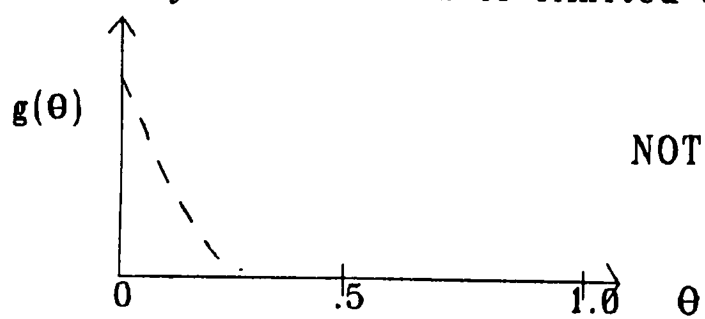
If the asymetry condition,

$$\int_{(y,y+\Delta]} dG \geq \int_{[1-y-\Delta,1-y)} dG,$$

holds for y, $y+\Delta < .5$, then $P_N < P$.

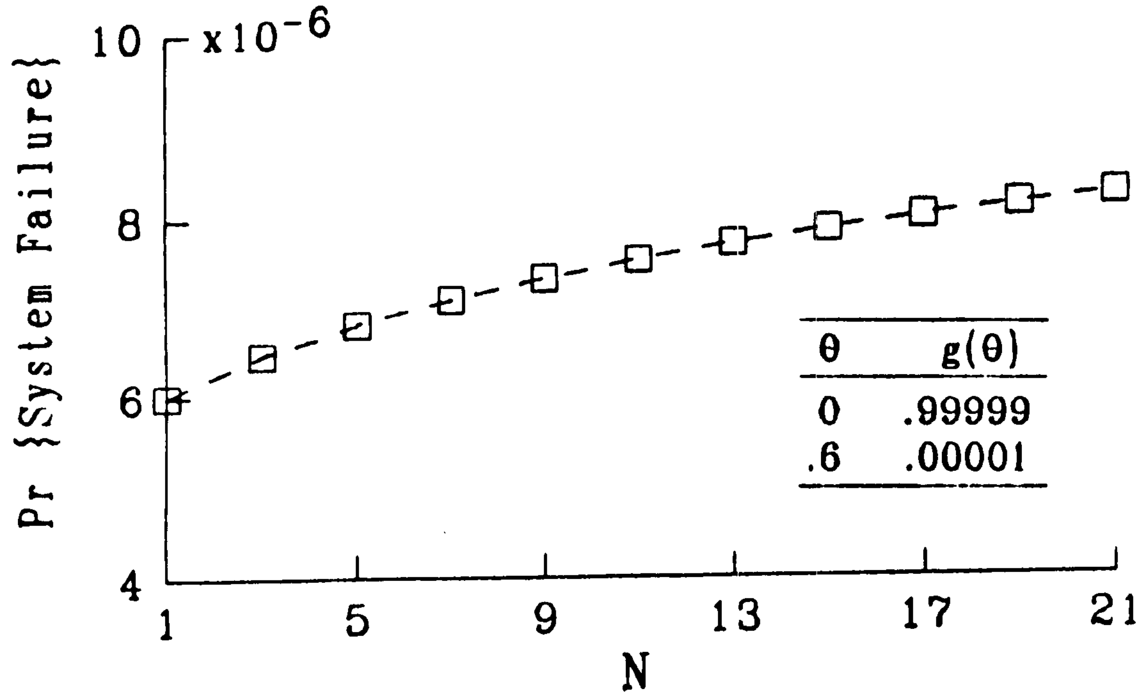


In particular, the asymmetry condition holds whenever the Intensity Distribution is limited above by .5.

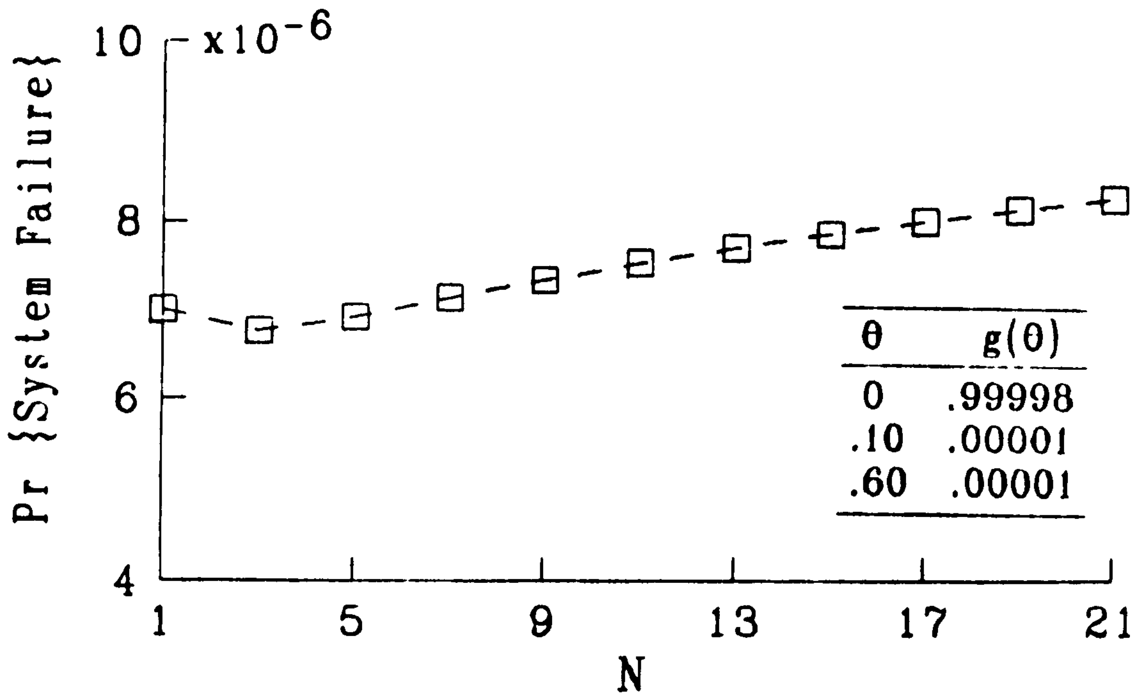


NOTE: $P < .5 \not\Longrightarrow P_N < P$

EFFECTS OF COINCIDENT ERRORS



EFFECTS OF COINCIDENT ERRORS



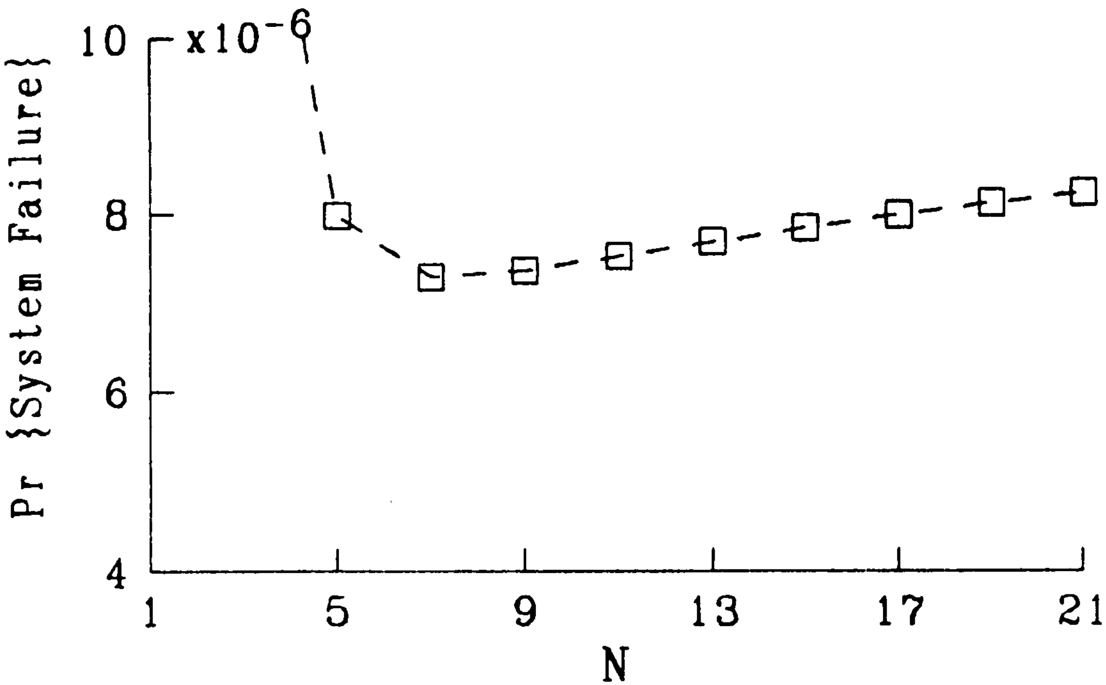
EFFECTS OF COINCIDENT ERRORS

Might an optimum value of N exist?

A necessary condition for system degradation in the limit is a violation of asymetry condition.
e.g.

θ	$g(\theta)$
0	.99899
.05	.00100
.60	.00001

Note: asymetry condition not necessary for $P_N < P$



RESULTS

- Provides a probabilistic framework for assessing strategy of redundant software.
- Provides foundation for experimental study of coincident errors.
- Permits an analytical study to increase understanding of impact of coincident errors.

SOFTWARE ERROR EXPERIMENT

COMPUTER SCIENCE DATA SYSTEMS TECHNICAL SYMPOSIUM

APRIL 16, 1985

N87-29126

LANGLEY RESEARCH CENTER

H. MILTON HOLT

P.7

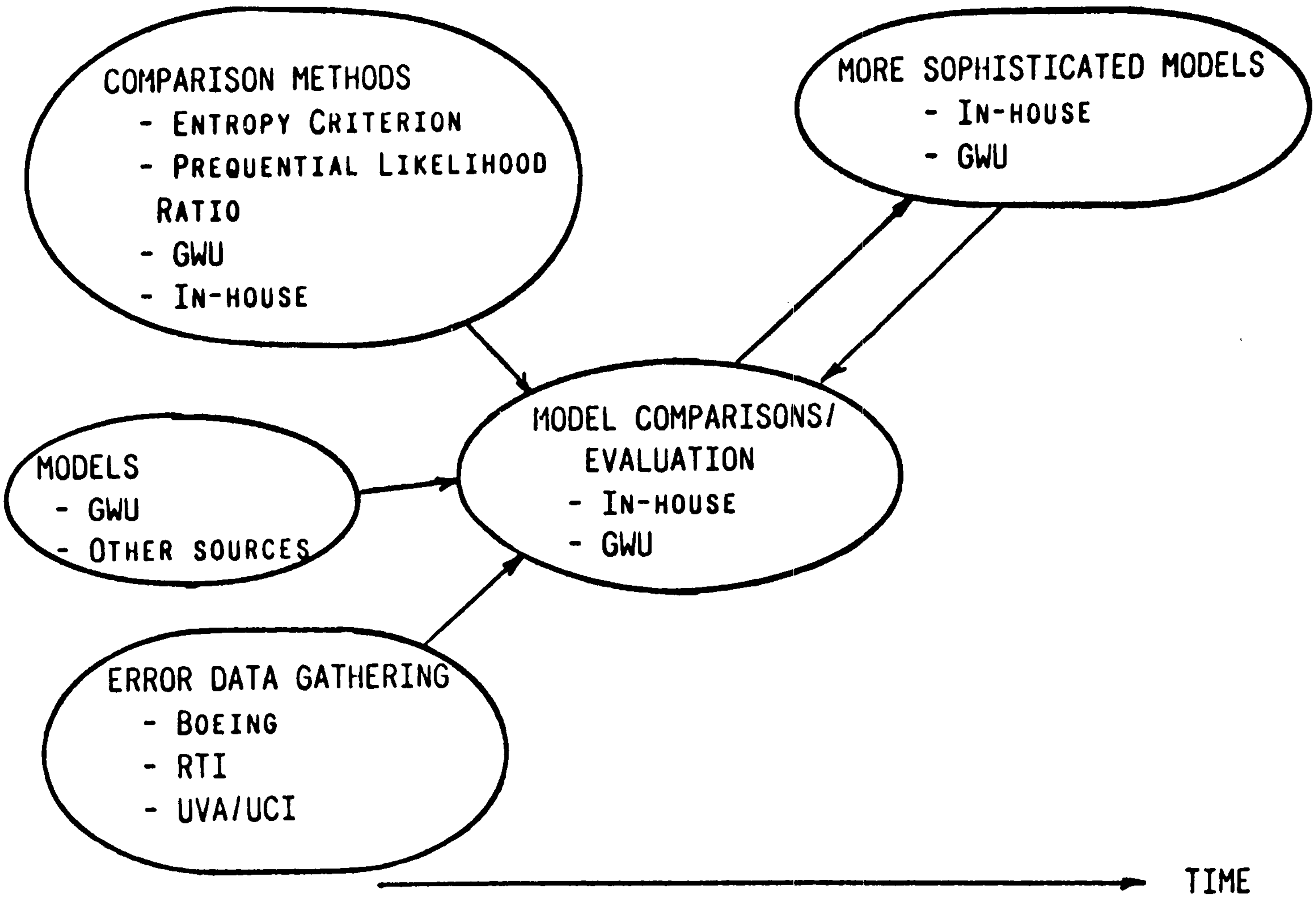
SOFTWARE ERROR EXPERIMENT

SOFTWARE RELIABILITY STUDIES

GOAL

TO DEVELOP ANALYTIC METHODS TO PROVE PERFORMANCE PROPERTIES AND MEASURE RELIABILITY PROPERTIES OF SOFTWARE

SOFTWARE RELIABILITY STUDIES



GENERAL STRATEGY

SOFTWARE ERROR EXPERIMENT

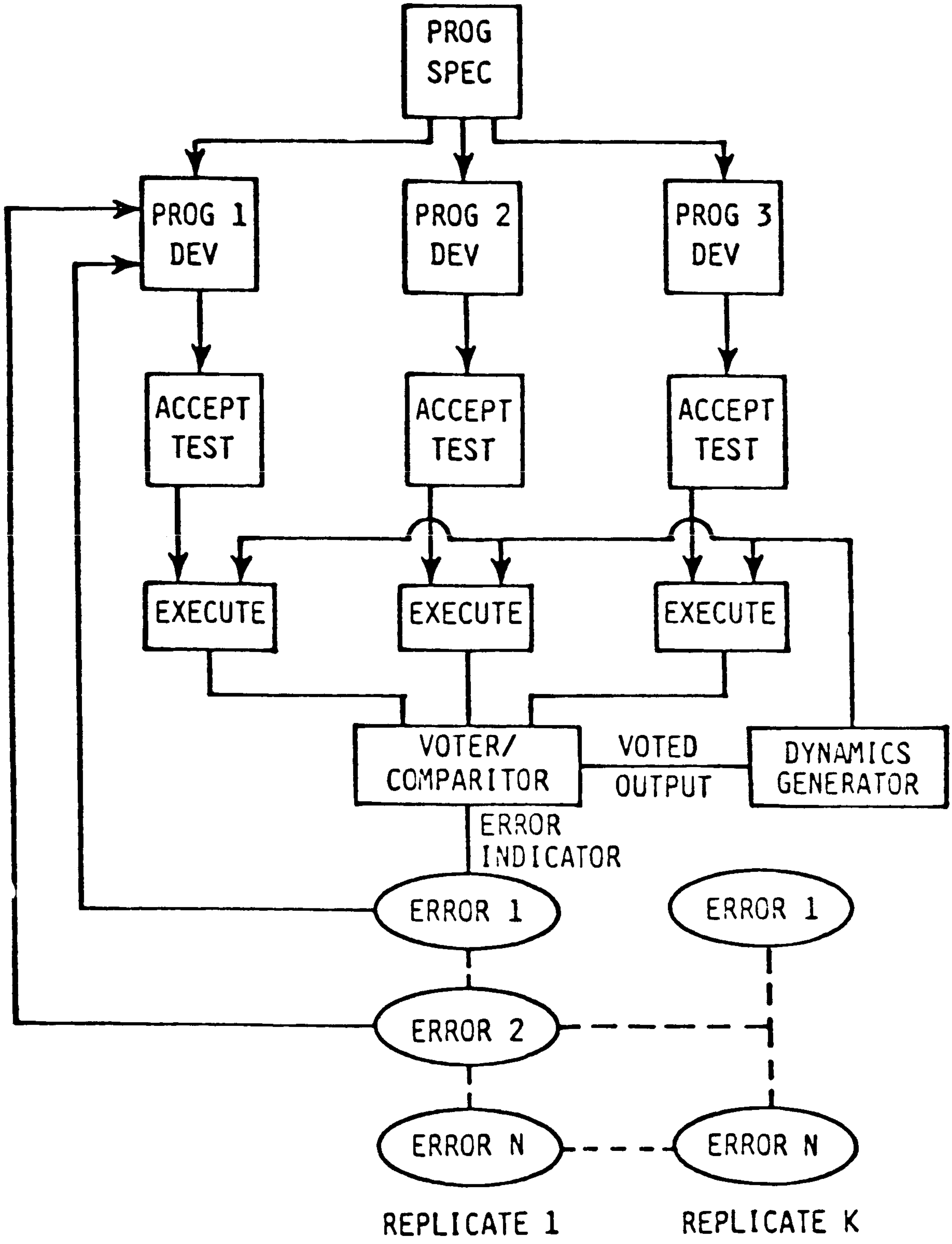
SOFTWARE RELIABILITY STUDIES

IMPORTANCE OF RESEARCH

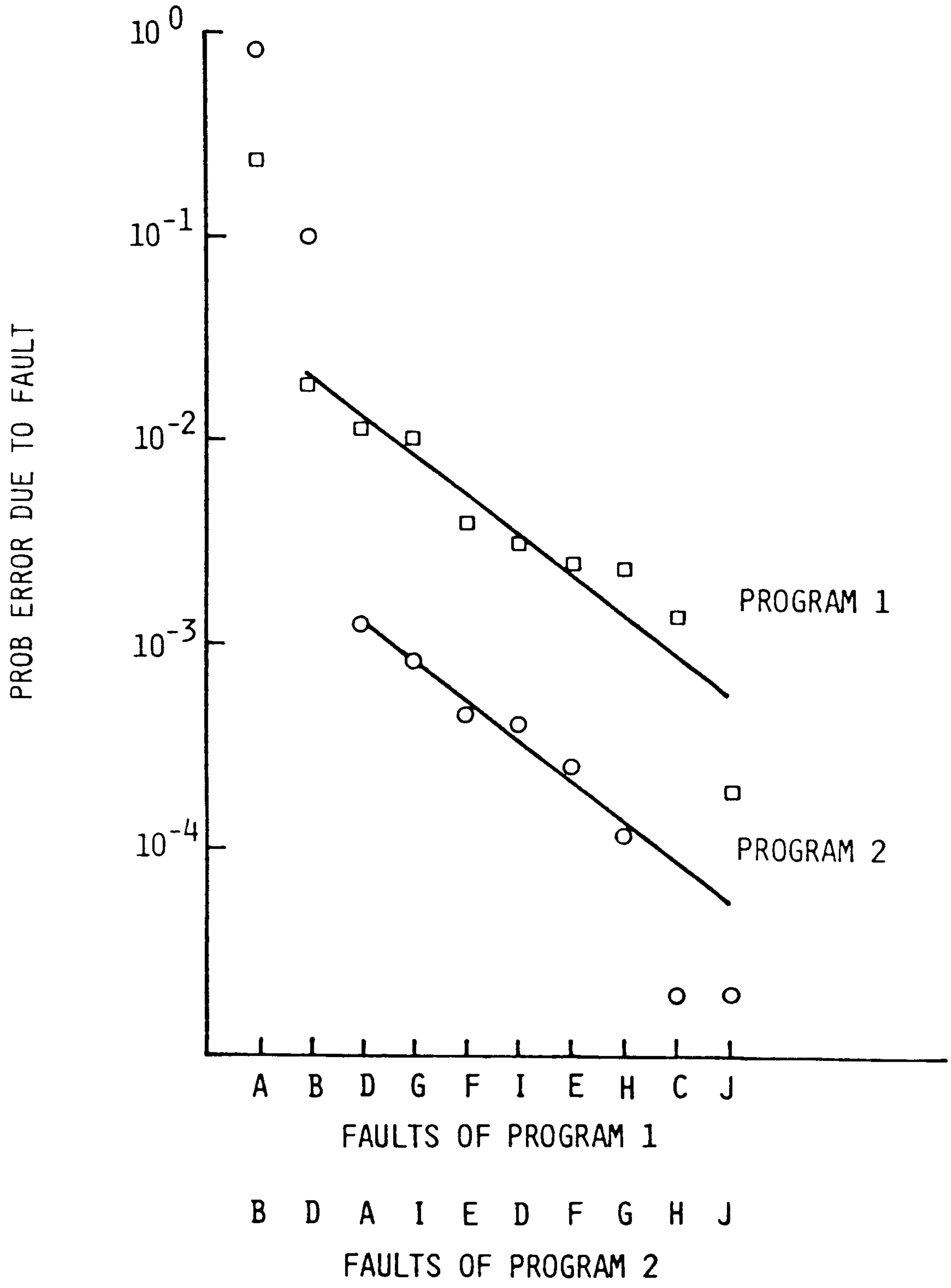
SOFTWARE IS ONE OF THE ELEMENTS WHOSE UNRELIABILITY CHARACTERISTICS NEED TO BE EVALUATED IF ESTIMATES OF AVIONICS SYSTEMS UNRELIABILITY ARE TO BE BELIEVABLE.

OF THE SOFTWARE RELIABILITY MODELS PROPOSED IN THE PAST DECADE, NONE HAS YET BEEN SHOWN TO BE ADEQUATE FOR PREDICTION/ESTIMATION PURPOSES IN THE CONTEXT OF HIGHLY RELIABLE SYSTEMS.

SOFTWARE ERROR EXPERIMENT



PRELIMINARY RESULTS OF THE SOFTWARE REPETITIVE RUN EXPERIMENT



SOFTWARE ERROR EXPERIMENTS

FUTURE PLANS

- ANALYZE DATA FROM ADDITIONAL VERSIONS
- GENERATE MULTIPLE VERSIONS OF CONTROL PROBLEM
- DEVELOP MORE SOPHISTICATED MODEL

N87-29127

7.10

DISTRIBUTED COMPUTER TAXONOMY BASED ON O/S STRUCTURE

Dr. Edwin C. Foudriat
NASA Langley Research Center
Hampton, VA 23665

Abstract

The taxonomy considers the resource structure at the operating system level. It compares a communication based taxonomy with the new taxonomy to illustrate how the latter does a better job when related to the client's view of the distributed computer. The results illustrate the fundamental features and what is required to construct fully distributed processing systems (network computer, "cooperative" autonomy, and decentralized computers).

The talk then discusses the problem of network computers for space station noting that the evolution from computer network operating systems to network computer operating systems is not practical (almost infeasible). The research direction is then discussed with the NASA research into network computers being listed.

DISTRIBUTED COMPUTER TAXONOMY
BASED ON O/S STRUCTURE

- O COMPARE SIMILAR TAXONOMIES

  COMMUNICATIONS VS. O/S

- O PROBLEM WITH NETWORK COMPUTER O/S

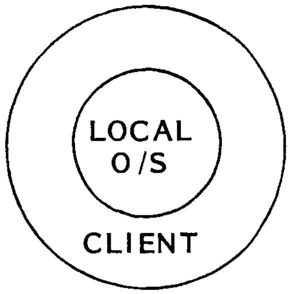
  APPLICATION TO SPACE STATION

- O RESEARCH SUPPORTIVE NETWORK COMPUTER O/S

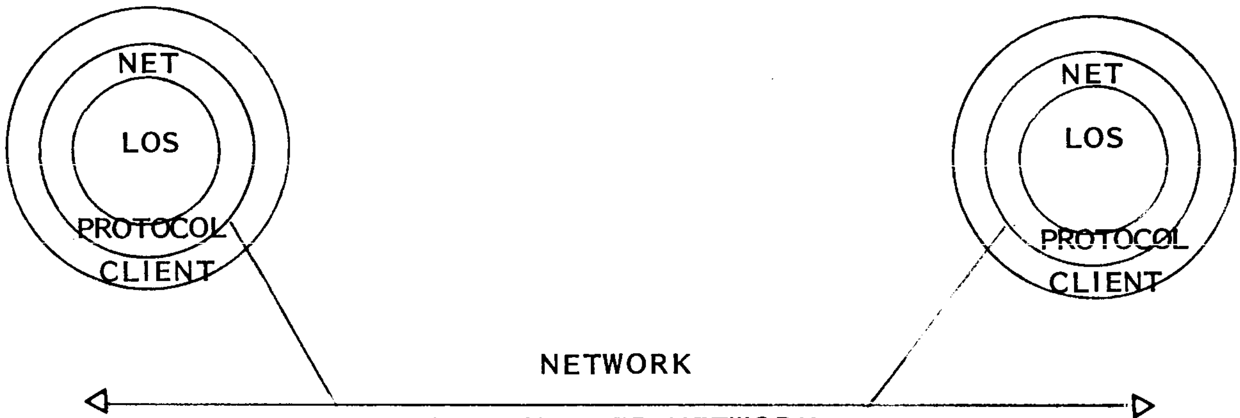
O/S OBJECTIVE: CREATE/CONTROL RESOURCES

EFFICIENTLY SHARE RESOURCES AMONG SET OF USERS

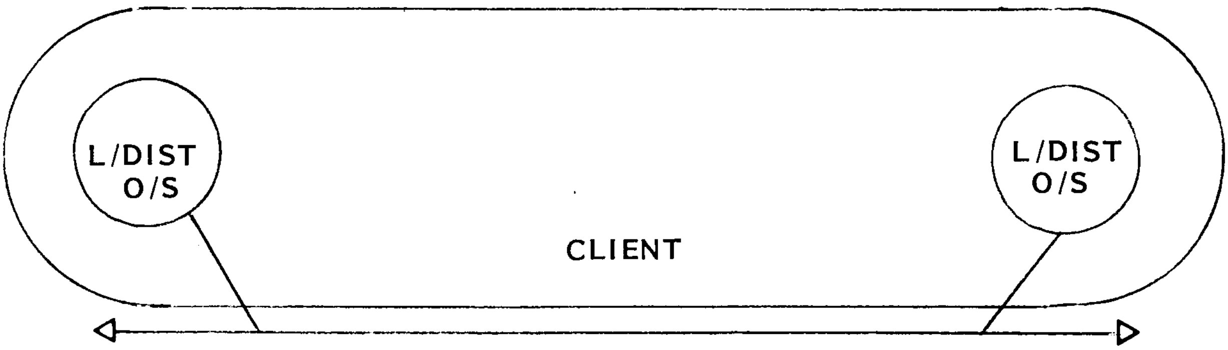
DISTRIBUTED COMPUTER DIVISIONS



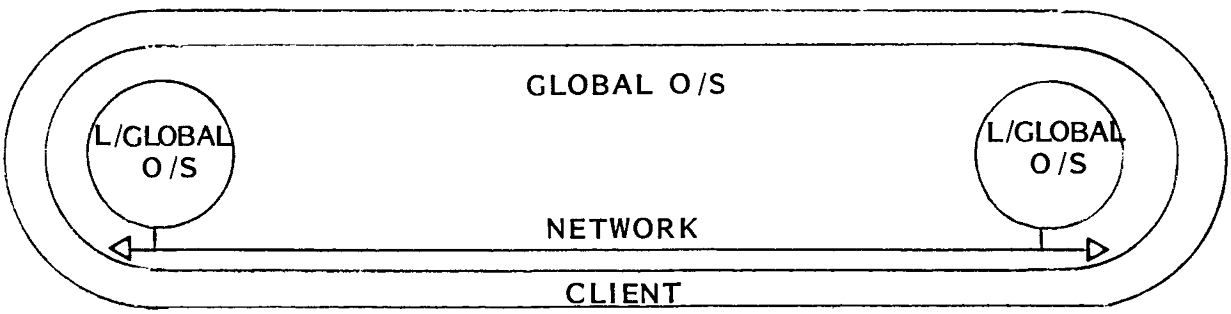
a) UNIPROCESSOR



b) COMPUTER NETWORK



c) NETWORK COMPUTER



d) MULTIPROCESSOR--ALGORITHM-DATA CONTROL PROCESSOR

NEW TAXONOMY FEATURES

O BETTER REPRESENTATION OF CLIENT'S VIEW

O DISTRIBUTED SYSTEM DISTINCTION

COMPUTER NET. - LOCAL CONTROL EXCEPT COMM.
COMM. RESOURCE VALUE-ADDED

NETWORK COMP. - INTEGRATED LOCAL - NETWORK -
DIST. CONTROL IN KERNEL

O ADMIT. SEPARATE CLIENT BUILT RESOURCE

OK, BUT LIMITED

DIFFICULT SHARING - INTEGRATING IN O/S

O INDICATES CRITICAL RESEARCH FOR NETWORK O/S

BETTER HARDWARE - FIRMWARE - O/S KERNEL

DISTRIBUTED CONTROL SUPPORT

DISTILLING THE QUOTATIONS

- CAREFUL ~~MAY NOT~~ BE A ROAD FROM COMP. NETWORK (LOS+NET.)
  TO NETWORK COMPUTER (LDOS)

  NO ROAD - (NEW TAXONOMY)

- NEW FEATURES NEED NEW METHODS

- EMBEDDED O/S COMPLEX--SELECT WITH CARE - FOUNDATION OF SYSTEM

  O/S NASTY HABIT OF GETTING FIXED IN CONCRETE

- COMPUTER NETWORK MODEL NOT APPROPRIATE FOR SPACE STATION

TAXONOMY - COMMUNICATIONS

(WITTIE & VAN TILBORG)

SEPARATION	FEATURE - USE		EXAMPLE
	COMPUTER NETWORK		
LARGE-MED. LOOSELY COUPLED	LOCAL-NODE AUTONOMY SIMPLE MESSAGES	REMOTE TERMINAL ACCESS RESOURCES	MULTI-PORT ARPANET, GRAPE-VINE, ETC.
	NETWORK COMPUTER		
MID-SMALL (LAN) CLOSELY COUPLED	LOCAL-GLOBAL CONTROL COMPLEX MESSAGES	DIST. RESOURCES NET. TRANSPARENCY RELIABILITY	OFFICE AUTOMATION FDPS, ARCHONS, CLOUDS FUTURE COMPUTERS
	MULTIPROCESSOR		
SHARED MEM. TIGHTLY COUPLED	GLOBAL CONTROL MACHINE LEVEL MESSAGES	TRANS. RESOURCES DISTRIBUTED TASKING RELIABILITY	CM*, APOLLO TANDEM, SYNAPSE, SEQUOIA, ETC. FUTURE COMPUTERS

TAXONOMY - O/S CONTROL STRUCTURE

O/S CONTROL LOCATION	FEATURE - USE		EXAMPLE
	UNIPROCESSOR		
LOCAL TO CPU	COMPLEX AUTONOMY	MULTI-USER MULTI-TASKING	MANY APÒLLO, SEQUOIA, ETC.
	COMPUTER NETWORK		
LOCAL + NETWORK	LOCAL-NODE AUTONOMY VALUE-ADDED COMM.	REMOTE TERM ACCESSED RESOURCES DIST. DATABASE (CLIENT)	MULTI-PORT ARPANET, X. PARC. OFFICE AUTOMATION CM*-STAROS, ETC. ROE, TABS, LOCUS, ETC.
	NETWORK COMPUTER		
LOCAL-NETWORK-DISTRIBUTED	LOCAL OR LOCAL-GLOBAL OR GLOBAL CONTROL	SHARED RESOURCES NETWORK TRANS. RELIABILITY	NONE IN EXIST. FDPS (CLOUDS) DEC. COM. (ARCHONS) PROCESS CONTROL, SPACE STATION
	ALGORITHM - DATA CONTROL COMPUTER		
LOCAL-(NETWORK) + SPEC. OP. + SPEC. DATA CTRL.	LOCAL AUTONOMY SPEC. CONTROL	PARTICULAR PROB. OR EQUATION	STAR, FEM, MPP, MANY

THE NETWORK COMPUTER PROBLEM - SPACE STATION

"Evolution is generally appropriate as the primary mode of computer (and other) system development, but it should be performed with much careful thought. Almost all work on "distributed" systems in general and "distributed"/network operating systems in particular has been evolutionary to an extreme--most of the resource management concepts have been simple adaptations of centralized ones, burdened by inappropriate and even counter-productive artifacts. The ineffectiveness of constructing airplanes which fly by flapping their wings was recognized early; but corresponding realizations about distributed systems have largely not taken place yet, as we have argued for several years." (Jensen - CMU - 84).

"Many claims have been made for distributed systems. Among them are improved reliability, increased processing power, and more flexible user environment. It is not clear, however, that current technology is able to realize these advantages. Without advances in methodologies for constructing distributed systems, we are faced with a situation in which we are likely to see less, not more, improvements in these areas, ... difficulties ... also results from increased complexity of managing the distributed environment." (McKendry - Georgia Tech - 83).

"Support for coordinated distributed computing, as exemplified by decentralized computing, network computers, or "cooperative" autonomy is critical to the development and use of embedded system when implemented over a network of computers." - "Future aerospace vehicles, like the Space Station, which must support autonomous, real-time subsystems, coordinated experiments including robotics and AI, and extensibility as new capability is added, will demand the best in operating system methodology. It is clear that without advances in distributed operating system methodologies" - this support is - "not going to be realized." (Foudriat - LaRC - 84).

To envision the Space Station Computer System Network as a data management problem with the addition of some "standard" networking protocols is a serious error. To concern one's self with bandwidth, network protocol and transfer of uninterpreted data, etc. is naive, at best. Once this computer system is conceived as an extremely complex resource management and sharing problem, progress on its development will have begun. (Foudriat - LaRC - since 1981).

OBJECT BASED O/S

OBJECT - ENCAPSULATION OF INFORMATION
SPECIFIC MECHANISMS FOR USE (ACCESS, ETC.)

SYNONYMS - PACKAGE (ADA), MODULE (MODULA - 2)
GUARDIAN (CLU), OBJECT (PATH PASCAL)

- O/S PARADIGM OF 80s-90s - LIKE VMS-DOS (70s)
  UNIX FILE, C (80s)
- OBJECT (NESTED) FOR EACH RESOURCE
- FEASIBLE FOR EMBEDDED SYSTEMS (FOUDRIAT - 84-85)

RESEARCH

- STRUCTURE FOR DISTRIBUTED CONTROL
- UNDERLYING COMPUTER SUPPORT
  (HARDWARE-FIRMWARE-O/S KERNEL-O/S LANGUAGE)
- PERFORMANCE TRADES
- DEBUG SUPPORT FOR DEVELOPMENT & TEST

NETWORK COMPUTER RESEARCH

- O GREAT DEAL OF RELATED DISTRIBUTED DATABASE WORK
- O ARCHONS - CMU $\approx$\$2-3 M/YEAR AF, NAVY, IBM, OTHERS
- O NASA (505-37-03)

INSTALLATION	NETWORK	LANGUAGE	OBJECTIVE
SUNY-STONY BROOK	MICROJET	MODULA-2	LARGE (100-10^3) DEBUG TECH.
GA. TECH.	CLOUDS	C	TRANSACTION IN KERNEL O/S LANG.
UNIV. OF ILL.	EMBEDDED O/S	PATH PASCAL, C, UNIX/ UNITED	EMBEDDED PERFORMANCE
UNIV. OF SO. FLA.	RDML	CONC. PASCAL & OTHERS	RELIABLE DISTRIBUTION
LARC	SDL-NET	PATH PASCAL	PERF. NET TOPOLOGY

MANY (AEROSPACE, IND, UN. GOVERN.) C.S. PROFESSIONALS (& LAITY) DO NOT UNDERSTAND THE SUBTLE NATURE OF THE DIST. O/S PROBLEM.

N87-29128

P-11

A WORKSTATION ENVIRONMENT FOR SOFTWARE ENGINEERING

SUSAN J. VOIGT
COMPUTER SCIENCE AND APPLICATIONS BRANCH
ANALYSIS AND COMPUTATION DIVISION
NASA LANGLEY RESEARCH CENTER

PRESENTED AT NASA COMPUTER SCIENCE/DATA SYSTEMS TECHNICAL SYMPOSIUM
LEESBURG, VA
APRIL 16, 1985

THE SOFTWARE PROBLEM

THERE ARE THREE FUNDAMENTAL PROBLEMS WITH SOFTWARE:

1. IT IS FREQUENTLY NOT SATISFACTORY TO THOSE WHO HAVE TO USE IT.
2. IT IS GENERALLY TOO COSTLY IN DEVELOPMENT AND OPERATION.
3. IT IS TOO OFTEN NOT MAINTAINABLE, NOT PORTABLE, AND NOT REUSABLE.

LANGLEY SOFTWARE ENGINEERING GROUP

PAST EXPERIENCE WITH

- SOFTWARE DEVELOPMENT
- SOFTWARE MAINTENANCE
- SOFTWARE PROCUREMENT
- TOOL EVALUATION

SPECIFIC TOOL DEVELOPMENTS

- IVTS
- MYSTRO
- SYNTAX-DIRECTED EDITORS
- NOTES FILE

EIGHTEEN MONTHS EXPERIENCE WITH UNIX-BASED WORKSTATION

OUR CONCLUSION

GOOD SOFTWARE ENGINEERING

IS THE

ANSWER

AND

A PERSONAL, UNIX-BASED WORKSTATION

IS THE

VEHICLE

WHY DO WE NEED SOFTWARE ENGINEERING?

- FOR CONSISTENT AND SUBSTANTIAL IMPROVEMENT IN SOFTWARE QUALITY.
- FOR CONSISTENT AND SUBSTANTIAL REDUCTION IN SOFTWARE DEVELOPMENT AND LIFE-CYCLE COSTS
  - o SOFTWARE TESTING COSTS
  - o SOFTWARE MAINTENANCE COSTS
- GOAL: CONSISTENT DEVELOPMENT OF "MAINTENANCE FREE SOFTWARE"

WHY CHOOSE UNIX?

"SMALL IS BEAUTIFUL" PHILOSOPHY

- O VAST COLLECTION OF UTILITIES AND TOOLS THAT CAN BE USED TO BUILD COMPLEX SOFTWARE FUNCTIONS
- O POWERFUL, PROGRAMABLE "SHELL" COMMAND LANGUAGES (FOREGROUND, BACKGROUND, CONTROL STRUCTURES)
- O LANGUAGE FLEXIBILITY (C, FORTRAN 77, PASCAL) (LINK OBJECT FILES FROM DIFFERENT COMPILERS, COMMON DEBUGGERS)

ADDITIONAL KEY FEATURES OF UNIX

- O POPULAR, MACHINE INDEPENDENT OPERATING SYSTEM, WRITTEN IN C (MICROS TO SUPERCOMPUTERS)
- O SIMPLE HIERARCHICAL FILE SYSTEM
- O COMPATIBLE I/O FOR FILES, DEVICES, AND PROCESSES
- O NETWORKING CAPABILITIES (INTER-PROCESS AND INTER-MACHINE)

UNIX TOOLS THAT HELP SOFTWARE ENGINEERS

- LINE AND SCREEN EDITORS (ED, EX, VI, SED)
- PIPES AND FILTERS (SHARED DATA BETWEEN PROCESSES)
- UTILITIES (LEARN, MAN, SPELL, DIFF, GREP, . . .)
- DOCUMENTATION: TEXT FORMATTING AND TYPESETTING (NROFF, TROFF, EQN, TBL, . . .)
- COMMUNICATION (MAIL, NEWS, NETWORK ACCESS, . . .)
- SOFTWARE DEVELOPMENT SUPPORT TOOLS (ADB, SDB, SCCS, MAKE, LEX, YACC, . . .)

SOFTWARE LIFE-CYCLE ENGINEERING TECHNOLOGY

OBJECTIVE: TO DEFINE, ESTABLISH, AND DEMONSTRATE A PROTOTYPE ENVIRONMENT TO SUPPORT THE SOFTWARE ENGINEERING LIFE CYCLE

RTOP: 505-37-13

E. H. SENN, K. A. SMITH, S. J. VOIGT

SOFTWARE ENGINEERING WORKSTATION ENVIRONMENT

- O PRESENT SYSTEM
  - CALLAN UNISTAR 200 DESKTOP MICROPROCESSOR M68000 CPU, 1 MB RAM, 43 MB DISK, FLOPPY DISK (FOR BACKUP), MULTI-USER
  - UNIX V7 (BERKELEY ENHANCEMENTS) WRITTEN IN "C"
  - LANGUAGES: C, FORTRAN, PASCAL, ADA SUBSET
  - SDDL/SOFTWARE DESIGN AND DOCUMENTATION LANGUAGE
  - SCMS/PROTOTYPE COMMAND LANGUAGE INTERPRETER

- O FUTURE SYSTEM
  - FILE SERVER AND WORKSTATION, EACH WITH M68010 CPU AND 2MB RAM, 130 MB CAPACITY DISK, CARTRIDGE TAPE
  - UNIX 4.2 BSD SOURCE CODE (LaRC MICRO/M680XX SUPPORT)
  - SAGA SOFTWARE ENVIRONMENT SUPPORT
    - o BASED ON SYNTAX-DIRECTED (LANGUAGE) EDITORS
    - o SOURCE CODE/VERSION CONTROL FACILITIES
    - o SOFTWARE PROOF MANAGEMENT SUPPORT

FUTURE PLANS

- ACQUIRE ADDITIONAL UNIX WORKSTATIONS TO SUPPORT SOFTWARE ENGINEERING RESEARCH AND LaRC UNIX CONSULTANTS
- FOSTER USE OF UNIX AT LANGLEY
- CONTINUE RESEARCH ACTIVITIES ON UNIX-BASED SOFTWARE SUPPORT TOOLS
- INFLUENCE AND SUPPORT SPACE STATION SOFTWARE PLANS (E.G., OPEN FORUM ON SPACE STATION SOFTWARE ISSUES APRIL 24-25, 1985 AT MSFC WILL BE DOCUMENTED IN NASA CP AND IEEE SOFTWARE)

N87-29129

p.5

ENGINEERING GRAPHICS AND IMAGE PROCESSING AT LANGLEY RESEARCH CENTER

SUSAN J. VOIGT
ANALYSIS AND COMPUTATION DIVISION

PRESENTED AT
NASA COMPUTER SCIENCE/DATA SYSTEMS TECHNICAL SYMPOSIUM
LEESBURG, VA
APRIL 16, 1985

PRECEDING PAGE BLANK NOT FILMED

PAGE 1-50 INTENTIONALLY BLANK

ENGINEERING GRAPHICS AND IMAGE PROCESSING

OBJECTIVE: TO MAKE RASTER GRAPHICS AND IMAGE PROCESSING TECHNIQUES READILY AVAILABLE FOR THE ANALYSIS AND DISPLAY OF ENGINEERING AND SCIENTIFIC DATA

RTOP: 505-37-23

KEY PERSONNEL: DR. STEPHEN K. PARK
DONALD L. LANSING

ENGINEERING GRAPHICS AND IMAGE PROCESSING

APPROACH: DEVELOP AND ACQUIRE TOOLS AND SKILLS WHICH ARE APPLIED TO SUPPORT LaRC RESEARCH ACTIVITIES IN SUCH DISCIPLINES AS AERONAUTICS AND STRUCTURES

0 SOLID GEOMETRY MODELING	SPACE STATION
0 MOVIE BYU	SPACE STATION, FLUID FLOW
0 RASLIB	FLUID FLOW
0 DI-3000	STRUCTURAL VIBRATIONS
0 IMAGE ANALYSIS	FLUID FLOW
0 SMOOTH SURFACING	PRESSURE DATA INTERPOLATION

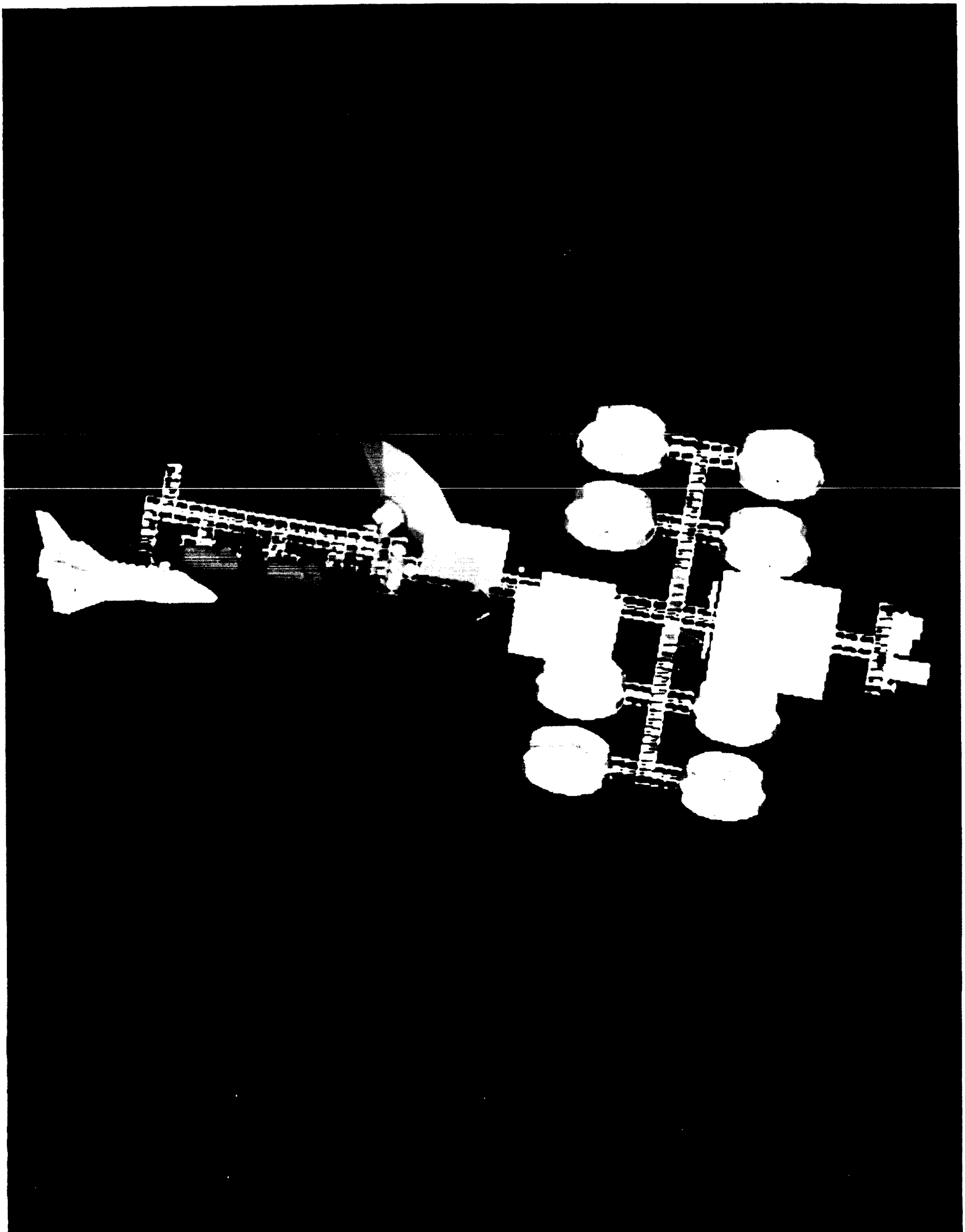
GRANTS

GEORGE WASHINGTON UNIVERSITY
DR. JAMES D. FOLEY
HIGH-LEVEL GRAPHICS PROGRAMMING LANGUAGE

BRIGHAM YOUNG UNIVERSITY
DR. MICHAEL B. STEPHENSON
SUBROUTINE LIBRARY FOR SHADED IMAGES

NORTH CAROLINA STATE UNIVERSITY
DR. DAVID F. McALLISTER
SURFACING TECHNIQUES WITH QUADRATIC SPLINES

COLLEGE OF WILLIAM AND MARY
DR. KEITH MILLER
DATA TYPES FOR IMAGE ENHANCEMENT



N87-29130

p.8

505-37-03-02

ICASE COMPUTER SCIENCE PROGRAM

- NASA AND ICASE HAVE LARGE INVESTMENT IN SCIENTIFIC COMPUTING
- FUTURE OF SCIENTIFIC COMPUTING DEPENDENT UPON EFFECTIVE UTILIZATION OF PARALLEL COMPUTERS
- PARALLEL COMPUTING IS SYSTEM ISSUE REQUIRING INTERACTION OF:

  PROBLEM DECOMPOSITION
  ALGORITHM DEVELOPMENT
  PROGRAMMING ENVIRONMENTS
  COMMUNICATION & SYNCHRONIZATION
  ARCHITECTURE

11/84

PROBLEM DECOMPOSITION

MAP ALGORITHM ONTO AN ARRAY OF PROCESSORS

- BALANCE COMPUTATIONAL LOAD
- REDUCE COMMUNICATION
- REDUCE SYNCHRONIZATION

DEVELOPING MAPPING HEURISTICS FOR CLASSES OF COMPUTATIONAL GRAPHS

<u>EXAMPLE</u>: MAP TREE ONTO MESH

ALGORITHM DEVELOPMENT

EXTENSIVE REVIEW OF NUMERICAL METHODS FOR PDEs

ASYNCHRONOUS ALGORITHMS

- ITERATIVE METHODS ON SIMULATOR AND FLEX
- NEW ALGORITHM FOR TIME DEPENDENT PROBLEMS

ADAPTIVE METHOD FOR TIME DEPENDENT PROBLEMS

- BALANCE PROCESSOR LOAD
- REDUCE COMMUNICATION

PROGRAMMING ENVIRONMENTS

- PISCES - PARALLEL IMPLEMENTATION OF SCIENTIFIC PROGRAMMING ENVIRONMENTS
  - FORTRAN - UNIX BASED
  - PROVIDE USERS WITH DIFFERENT VIEW OF SYSTEM DEPENDING ON INTERESTS AND NEEDS
  - ENHANCE PORTABILITY OF APPLICATIONS PROGRAMS
- BLAZE
  - NEW PARALLEL SCIENTIFIC PROGRAMMING LANGUAGE
  - OFFERS ADVANCED SUPPORT FOR ARRAY COMPUTATIONS
  - SIMPLIFIES EXTRACTION OF PARALLELISM BY COMPILER

COMMUNICATION AND SYNCHRONIZATION (C. & S.)

- MAJOR OVERHEAD ASSOCIATED WITH PARALLEL COMPUTATION
- LACK FORMAL METHODOLOGY FOR STUDY SUCH AS COMPUTATIONAL COMPLEXITY FOR NUMERICAL ALGORITHMS
- DEVELOPED TWO MODELS THAT RELATE C. & S. COSTS TO:
  - ALGORITHM CHARACTERISTICS
  - PROBLEM SIZE
- SEEK GENERIC CHARACTERISTICS OF C. & S. THAT IMPACT:
  - PROBLEM DECOMPOSITION
  - ALGORITHM DEVELOPMENT
  - ARCHITECTURE
- VALIDATE WITH ACTUAL COSTS ON REAL PARALLEL SYSTEMS

PARALLEL ARCHITECTURES

EVALUATION OF ARCHITECTURAL FEATURES IN VIEW OF SPECIFIC ALGORITHMS

- BIN PACKING PROBLEM
- MULTIGRID ALGORITHM
- SPARSE MATRIX ITERATION
- ADAPTIVE ALGORITHM

TOOLS

- SIMON SIMULATOR
- FLEX WITH DIFFERENT LOCAL/GLOBAL MEMORY CONFIGURATIONS
- HEP
- HYPERCUBE (PLANNED)

PROBLEM

SOFTWARE LAYERS MAKE A "VIRTUAL MACHINE" OUT OF ANY PARALLEL HARDWARE

IMPLICATIONS FOR:

- PERFORMANCE
- PROGRAMMING
- CODE PORTABILITY

PISCES PROJECT

WHAT SORT OF VIRTUAL MACHINE SHOULD THE SOFTWARE PROVIDE?

- EFFECTIVE PERFORMANCE ACROSS A RANGE OF ARCHITECTURES
- EASE OF PROGRAMMING
- PORTABLE APPLICATIONS CODES

PISCES DESIGN APPROACH

- SUPPORT RESEARCH IN PARALLEL SCIENTIFIC ALGS.
- BASE SEQUENTIAL LANGUAGE
- SIMPLE SET OF EXTENSIONS
- VIRTUAL MACHINE - PRECISELY DEFINED
- SEVERAL GRANULARITIES OF PARALLELISM
- DISTRIBUTED ACCESS TO DATA
- PORTABLE TO A VARIETY OF PARALLEL MACHINES
- COMPLETE VIRTUAL MACHINE ENVIRONMENT
  - SECONDARY STORAGE ACCESS
  - MULTIPLE USERS
  - ERROR HANDLING
  - RELIABILITY

STATUS
(4/85)

- FTN 77/UNIX/VAX UNIPROCESSOR VERSION - IN USE
  - FORTRAN EXTENSIONS
  - USER INTERFACE
  - ANALYSIS PACKAGE

- APPLICATIONS - RUNNING
  - ITERATIVE SPARSE MATRIX SOLVER

    PIPELINED DATA FLOW
    ASYNCHRONOUS OR SYNCHRONOUS
    BROADCAST OR POINT-TO-POINT
  - IMAGE UNDERSTANDING/AI PROGRAM

- APOLLO WORKSTATION NETWORK IMPLEM. - RUNNING
  - 10 WORKSTATIONS, 3 DISKS
  - SAME VIRTUAL MACHINE FOR PISCES USER
  - PERFORMANCE STUDIES

- FLEX/32 IMPLEMENTATION - IN DESIGN
  - 17 PES, SEVERAL DISKS
  - SHARED MEMORY - 3 PERFORMANCE LEVELS

- NEW FORMAL MODEL OF PARALLEL COMPUTATION

N87-29131

p.13

HETEROGENEOUS DISTRIBUTED QUERY PROCESSING--

THE DAVID SYSTEM

Barry E. Jacobs
Senior Research Scientist
Goddard Space Flight Center
NASA
Greenbelt, Maryland 20771

ABSTRACT

The objective of the Distributed Access View Integrated Database (DAVID) project is the development of an easy-to-use computer system with which NASA scientists, engineers and administrators can uniformly access distributed heterogeneous databases. Basically, DAVID will be a database management system that sits alongside already existing database and file management systems. Its function is to enable users to access the data in the other database and file systems without having to learn the different data manipulation languages.

The scope of this talk will concentrate on the DAVID system and several related issues. First, we describe the problems caused by the diversity of database types and implementations. Second, we discuss two solutions to the problem-- standardization and uniformization. Third, we consider the benefits of the recently developed uniform called "database logic." Fourth, we describe the DAVID system which uses database logic as its framework. Fifth, we outline the status of the development of the DAVID system.

HETEROGENEOUS DISTRIBUTED QUERY PROCESSING--

THE DAVID SYSTEM

Barry E. Jacobs
Senior Research Scientist
Goddard Space Flight Center
NASA
Greenbelt, Maryland 20771

THE PROBLEM

THERE ARE MANY WAYS FOR A SCIENTIST TO STORE DATA IN A DATABASE:

file approach --- VMS, DOS, UNIX, ...
relational approach --- Oracle, Ingres, dBase II, ...
hierarchical approach --- IMS, RAMIS, ...
network approach --- DEC-CODASYL, DMS/1100, ...

THIS ABUNDANCE HAS RESULTED IN THE FOLLOWING:

- A NASA SCIENTIST HAS TO LEARN MANY DIFFERENT ACCESS METHODS IN ORDER TO OBTAIN DATA.
- THE DIVERSITY OF TYPES OF DATABASES RESULTS IN AN EXPLOSIVE REPETITION OF DATA AND DATABASES.
- THE DATABASE PROFESSIONAL HAS TO LEARN MANY DIFFERENT TERMINOLOGIES IN ORDER TO READ THE LITERATURE.

OUTLINE OF TALK

1. THE PROBLEM CAUSED BY THE DIVERSITY OF DATABASE TYPES.
2. STANDARDIZATION VERSUS UNIFORMIZATION
3. DATABASE LOGIC AS A UNIFORM
4. THE DAVID SYSTEM
5. DAVID SYSTEM DEVELOPMENT

STANDARDIZATION VS. UNIFORMIZATION

STANDARDIZATION

- IS AN "A PRIORI" MEANS USED TO ESTABLISH COMMONALITY.
- NEW DATABASES ARE SUBSEQUENTLY BUILT ACCORDING TO THESE "STANDARDS".
- CANNOT HELP IN THE CASE OF ALREADY EXISTING DATABASES NOT ADHEARING TO THE STANDARD.

UNIFORMIZATION

- IS AN "A POSTERIORI" MEANS TO ESTABLISH COMMONALITY.
- PERMITS DATABASE CONSTRUCTION WITHOUT CONSTRAINT ON THE USER.
- CAN HELP IN THE CASE OF ALREADY EXISTING DATABASES.

THE RELATIONAL APPROACH

DATABASES

TAPEINFO				
NOTAPE	TAPETYPE	TITLE1	TITLE2	TITLE3
1003	SDT	NIMBUS6...	BY...	DATA...
1004	SDT	NIMBUS6...	BY...	DATA...

FILEINFO		
PB	FILE	NOTAPE
17400.0	2	1003
17401.0	3	1003
17457.0	2	1004
17459.0	3	1004

RECINFO											
DATE	TIME	LON	LAT	ALT	ZEN	PB	QUALITY	ELECTR	ILLUMIN	CALIB	SCAN
790103	124549	158.57	-1.00	1112.50	153.63	17400.0	0	ON	NIGHT	NO	OFF
790105	145629	25.90	-0.96	1112.30	153.97	17459.0	0	ON	NIGHT	NO	OFF
790117	110204	2.75	10.43	1105.00	34.11	17457.0	0	ON	NIGHT	NO	OFF
790107	105349	4.82	10.84	1103.00	35.59	17401.0	8	OFF	DAY	YES	ON

QUERIES ON DATABASES

```
select (t.tapetype, t.title1, t.title2, t.title3) as result
from tapeinfo t,
     fileinfo f,
     recinfo r
where t.notape = f.notape
  and f.pb = r.pb
  and r.zen = 153.63
  and r.calib = 'NO'
```

THE DATABASE LOGIC APPROACH

DATABASES

<table>
<tr><th colspan="10">RTAPE</th></tr>
<tr><th rowspan="3">NOTAPE</th><th rowspan="3">TAPETYPE</th><th colspan="7">PLAYBACK</th><th>RTITLE</th></tr>
<tr><th rowspan="2">PB</th><th rowspan="2">NOFILE</th><th colspan="5">CATALOG</th><th rowspan="2">TITLE</th></tr>
<tr><th>LON
QUALITY</th><th>LAT
ELECTR</th><th>ALT
ILLUMIN</th><th>ZEN
CALIB</th><th>TIME1
SCAN</th></tr>
<tr><td rowspan="4">1010</td><td rowspan="4">SDT</td><td colspan="7">e(PLAYBACK):1</td><td>e(RTITLE):1</td></tr>
<tr><td rowspan="3">81261.4</td><td rowspan="3">4</td><td colspan="5">e(CATALOG):1</td><td>NIMBUS6...</td></tr>
<tr><td>-133.69
1</td><td>-55.62
ON</td><td>1123.70
TWILIGHT</td><td>98.71
NO</td><td>012076
OFF</td><td>BY...</td></tr>
<tr><td>-134.25
1</td><td>-56.47
ON</td><td>1123.90
TWILIGHT</td><td>97.82
NO</td><td>012109
OFF</td><td>DATA...</td></tr>
<tr><td rowspan="4">2010</td><td rowspan="4">SDT</td><td colspan="7">e(PLAYBACK):2</td><td>e(RTITLE):2</td></tr>
<tr><td rowspan="3">81261.0</td><td rowspan="3">3</td><td colspan="5">e(CATALOG):2</td><td>NIMBUS7...</td></tr>
<tr><td rowspan="2">155.04
0</td><td rowspan="2">-80.07
ON</td><td rowspan="2">1126.80
DAY</td><td rowspan="2">66.65
NO</td><td rowspan="2">021349
OFF</td><td>BY...</td></tr>
<tr><td>DATA...</td></tr>
</table>

FUNCTIONS

NOTAPE	TITLE1	TITLE2	TITLE3
1010	NIMBUS6...	BY...	DATA...
2010	NIMBUS7...	BY...	DATA...

QUERIES ON DATABASES

```
select (rpc.tapetype, f.title1, f.title2, f.title3) as result
from rtape-playback-catalogue rpc,
     functions f
where rpc.notape = f.notape
  and f.pb = r.pb
  and rpc.zencalib = '153.63'||'NO'
```

DATABASE LOGIC AS A UNIFORM

1. DATABASE LOGIC IS A THEORETICAL FOUNDATION FOR THE STUDY OF DATABASE ISSUES:

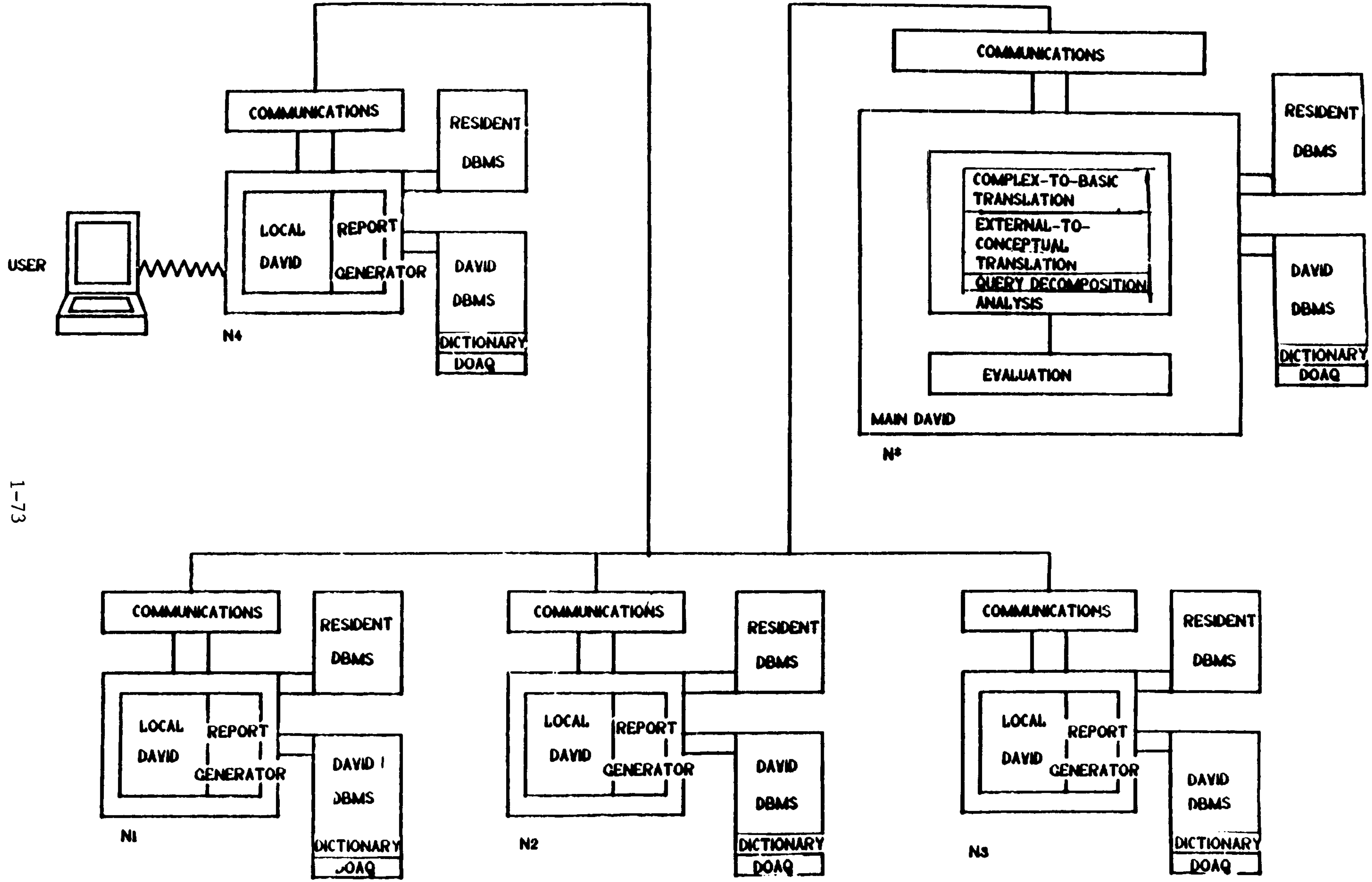
   First–Order logic is to the Relational Case
   as
   Database logic is to the Heterogeneous Case
   (Relational, Hierarchical and Network)

2. DATABASE LOGIC HAS BEEN APPLIED TO A NUMBER OF FUNDAMENTAL DATABASE ISSUES:

   - external–to–conceptual mapping,
   - view update,
   - view integration,
   - database conversion,
   - automatic program conversion, and
   - external axiomatization.

3. DATABASE LOGIC SERVES AS THE FOUNDATION FOR THE DEVELOPMENT OF A SYSTEM FOR ACCESSING HETEROGENEOUS DISTRIBUTED DATABASES.

4. DATABASE LOGIC SERVES AS THE BASIS FOR BUILDING EXPERT SYSTEMS ON TOP OF ALREADY EXISTING DATABASES AND EXPERT SYSTEMS.



THE DISTRIBUTED ACCESS
VIEW INTEGRATED DATABASE
(DAVID)
SYSTEM

DAVID DEVELOPMENT:

MILESTONES

MILESTONE A. (Summer 1985)

Build a demonstration system using the following environment:

VAX under VMS using an ORACLE DBMS
SUN under UNIX using a DAVID network DBMS
PC under DOS 2.0 using a dBASE II DBMS
MACINTOSH under Mac OS

MILESTONE B. (Summer 1987)

Build a demonstration system using several of NASA's real systems at:

GSFC, ARC, GISS, JPL, NSTL

DAVID DEVELOPMENT:

THE LOCAL DAVID.

Ashok Agrawala, University of Maryland
Michael Anshel, City University of New York
Joseph Aulino, USAF
Kenneth Baum, USAF
Jeanne Behnke, Johns Hopkins University
Nancy Broderick, University of Maryland
Isadore Brodsky, University of Puerto Rico
Dehe Cao, Beijing Polytechnic University
Upen S. Chakravarthy, University of Maryland
Liang Fang, University of Maryland
Adel Gharib, University of Maryland
John Grant, Towson State University
Kim Haynes, TRW Inc.
Hsiao–Fang Hu, University of Maryland
Kyu–Hyun Hwang, SAR Corp.
Barry E. Jacobs, NASA
Thomas E. Jacobs, University of Maryland
Elizabeth Nichols, Digital Analysis Corporation
Joseph Nichols, Digital Analysis Corporation
Surrendra Ray, SAR Corp.
Ira Sack, Stevens Institute of Technology
Michael Shapiro, Bell Labs
Duc Tran, Digital Analysis Corporation
Satish Trapathi, University of Maryland
Cynthia A. Walczak, NIH
Jack Welch, The Catholic University of America

DAVID DEVELOPMENT:

PROPOSED DAVID INTERFACES

CENTER	PROPOSED TESTBED	UNIVERSITY
ARC	PLDS, LAS	C.U.N.Y.
GSFC	PCDMS, PLDS, LAS	U of Maryland Catholic University
GISS	ISCCP	C.U.N.Y.
JPL	PODS, PLDS, PPDS, LAS	U of Southern California
NSTL	PLDS, EOS	Louisiana State University
JSC	TBD	U of Houston Louisiana State University
LeRC	TBD	U of Toledo
MSFC	TBD	U of Alabama
LaRC	TBD	Old Dominion University

SUMMARY

1. THE PROBLEM CAUSED BY THE DIVERSITY OF DATABASE TYPES.
2. STANDARDIZATION VERSUS UNIFORMIZATION
3. DATABASE LOGIC AS A UNIFORM
4. THE DAVID SYSTEM
5. DAVID SYSTEM DEVELOPMENT

N87-29132

p.14

INTELLIGENT DATA MANAGEMENT

WILLIAM J. CAMPBELL
SPACE DATA AND COMPUTING DIVISION
DATA MANAGEMENT SYSTEM FACILITY
NASA/GSFC

ABSTRACT

INTELLIGENT DATA MANAGEMENT IS THE CONCEPT OF INTERFACING A USER TO A DATABASE MANAGEMENT SYSTEM WITH A VALUE ADDED SERVICE THAT WILL ALLOW A FULL RANGE OF DATA MANAGEMENT OPERATIONS AT A HIGH LEVEL OF ABSTRACTION USING HUMAN WRITTEN LANGUAGE. THE DEVELOPMENT OF SUCH A SYSTEM WILL BE BASED ON EXPERT SYSTEMS AND RELATED ARTIFICIAL INTELLIGENCE TECHNOLOGIES, AND WILL ALLOW THE CAPTURING OF PROCEDURAL AND RELATIONAL KNOWLEDGE ABOUT DATA MANAGEMENT OPERATIONS AND THE SUPPORT OF THE USER WITH SUCH KNOWLEDGE IN AN ON-LINE, INTERACTIVE MANNER. SUCH A SYSTEM WILL HAVE THE FOLLOWING CAPABILITIES:

- AN UNDERSTANDING BETWEEN SCIENCE APPLICATIONS AND STORED DATA
- THE ABILITY TO CONSTRUCT A MODEL OF THE USERS VIEW OF THE DATABASE, BASED ON THE QUERY SYNTAX
- THE ABILITY TO TRANSFORM ENGLISH QUERIES AND COMMANDS INTO DATABASE INSTRUCTIONS AND PROCESSES
- THE USE OF HEURISTIC KNOWLEDGE TO RAPIDLY PRUNE THE DATA SPACE IN SEARCH PROCESSES
- AN ON-LINE EXPLANATION SYSTEM THAT WILL ALLOW THE USER TO UNDERSTAND WHAT THE SYSTEM IS DOING AND WHY IT IS DOING IT

SUCH A SYSTEM WILL BE GOAL ORIENTED RATHER THAN PROCEDURE ORIENTED SUPPORTING VARYING LEVELS OF ABSTRACTION OF THE DATA OF INTEREST TO THE USER.

INTELLIGENT DATA MANAGEMENT

Presentation at the
Computer Science/Data Systems Technical Symposium
Leesburg, Virginia

April 16-18, 1985

William J. Campbell
Data Management Systems Facility
NASA/Goddard Space Flight Center

THE INFORMATION EXPLOSION
PROBLEM/SOLUTION

PROBLEM

- A PRIMARY PROBLEM IN EARTH SCIENCE RESEARCH IS COPING WITH THE MASSIVE AMOUNTS OF REALTIME AND ANALYTICAL DATA BEING GENERATED
- THE AMOUNT OF AVAILABLE INFORMATION AND DATA ALREADY EXCEEDS THE SCIENTIST'S ABILITY TO MANAGE AND USE IT
- USING PRESENT TECHNOLOGIES ONLY A SMALL PERCENTAGE OF THE DATA WILL EVER BE UTILIZED

SOLUTION

- THE APPLICATION OF ARTIFICIAL INTELLIGENCE (AI) TO EARTH SCIENCE INFORMATION MANAGEMENT OFFERS THE ONLY REAL POSSIBILITY OF REVERSING THIS TREND

WHAT IS ARTIFICIAL INTELLIGENCE (AI)

FROM THE EARTH SCIENCE VIEW, WE DEFINE AI AS:

- A VALUE ADDED SERVICE THAT EXTENDS THE OPERATIONAL CAPABILITIES OF A USER BY PROVIDING A SYSTEM THAT CAPTURES HUMAN KNOWLEDGE FOR SUPPORTING PROCEDURAL, OPERATIONAL AND RESEARCH TASKS IN THE AREA OF DATA MANAGEMENT FOR EARTH SCIENCE RESEARCH

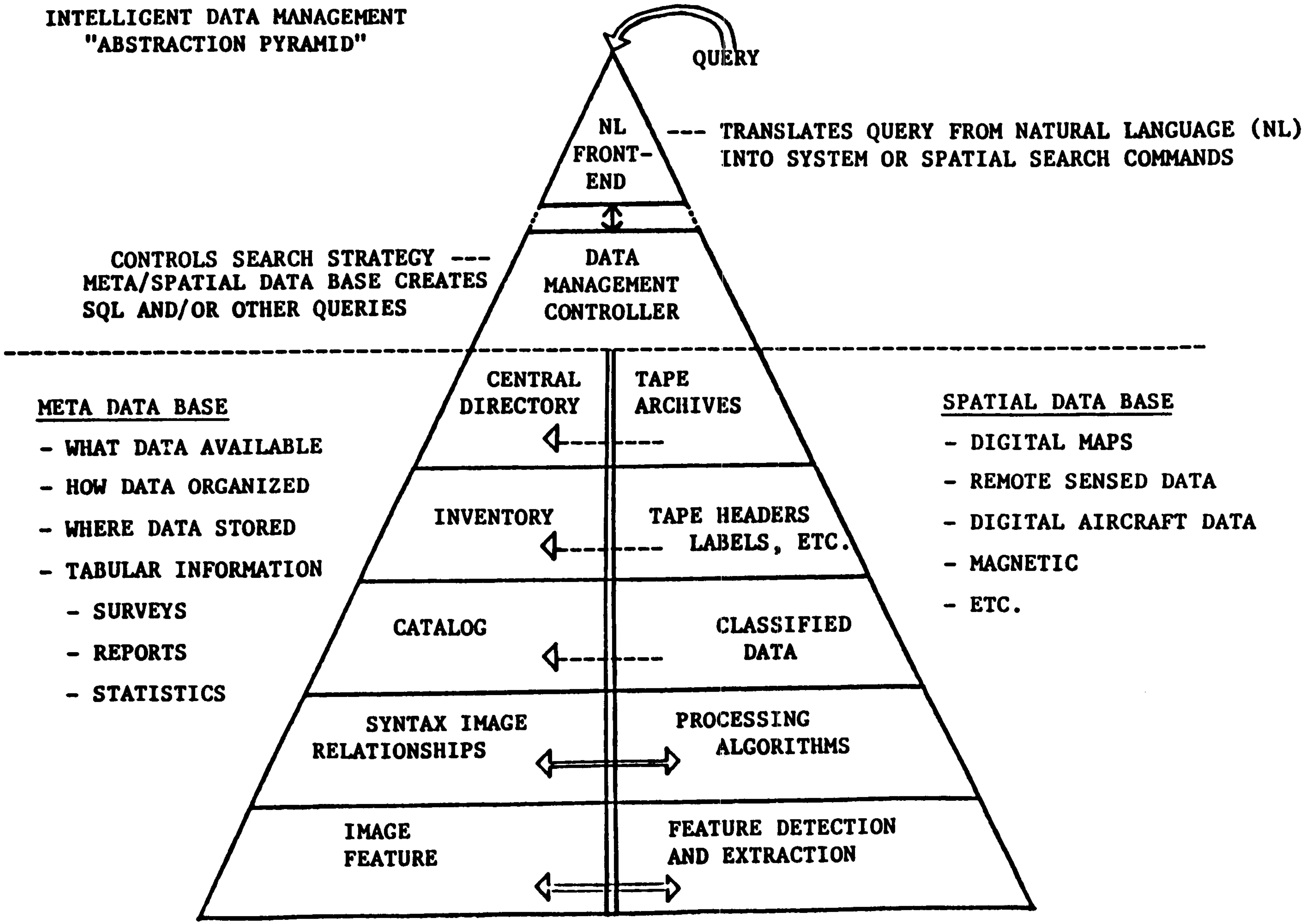
HOW INTELLIGENT DATA MANAGEMENT CAN SUPPORT EARTH SCIENCE

- PROVIDE VALUE-ADDED SERVICES TO SYSTEMS THAT SUPPORT EARTH SCIENCE RESEARCH
- PROVIDE THE ABILITY TO MINIMIZE THE INFORMATION PROCESSING AND MANAGEMENT TIME, WHILE MAXIMIZING THE RESEARCH AND APPLICATION TIME
- THE MAJOR CONTRIBUTIONS ARE EXPECTED TO BE:
  - THE UNBURDENING OF THE SCIENTIST FROM THE MORE MUNDANE TASKS OF DATA IDENTIFICATION, SELECTION AND MODIFICATION
  - THE EXTENSION OF HUMAN COGNITIVE PROCESSES TO OPERATIONS THAT SUPPORT THE MANAGEMENT, ANALYSIS, UNDERSTANDING AND APPLICATION OF EARTH RELATED DATA
  - THE CAPTURING OF HUMAN KNOWLEDGE INTO SYSTEMS THAT CAN BE USED BY NON-EXPERTS

KNOWLEDGE-BASED EXPERT SYSTEMS

- WOULD SUPPORT THE USER WITH EXPERT KNOWLEDGE IN DATA SEARCH AND FEATURE IDENTIFICATION
- WOULD PROVIDE PROCEDURAL KNOWLEDGE TO SUPPORT DATA MANAGEMENT SYSTEM OPERATION AND EXPLANATIONS
- WOULD PROVIDE SELF CHECKING AND CORRECTION PROCESSES
- SUPPORT THE COLLECTION OF HEURISTIC KNOWLEDGE ON A REALTIME ON-GOING BASIS
- WOULD SUPPORT APPROXIMATE REASONING TO INFER CONCLUSIONS THAT ARE NOT EXPLICITLY STATED BY THE USER (I.E. FUZZY QUERIES)
- WOULD USE HEURISTIC SEARCH PROCESSES THAT LEAD TO SHORTCUTS BY EARLY PRUNING OF LARGE PORTIONS OF UNWANTED DATA SPACE

INTELLIGENT DATA MANAGEMENT
"ABSTRACTION PYRAMID"



NATURAL LANGUAGE

- NATURAL LANGUAGE MEANS AUTOMATED HUMAN LANGUAGE UNDERSTANDING
- NATURAL LANGUAGE PROCESSING FOR SUPPORTING EARTH SCIENCE INVOLVES THE DEVELOPMENT OF COMPUTER PROGRAMS WHICH CAN ANALYZE HUMAN LANGUAGE AND PERFORM THE APPROPRIATE ACTION ON THE INFORMATION CONTAINED IN THE TEXT OR UTTERANCE
- PRESENT NATURAL LANGUAGE APPLICATIONS ARE LIMITED TO AREAS THAT REQUIRE A LIMITED DICTIONARY (E.G. DATABASE QUERIES)

THE RELATIONAL APPROACH

DATABASES

TAPEINFO				
NOTAPE	TAPETYPE	TITLE1	TITLE2	TITLE3
1003	SDT	NIMBUS6...	BY...	DATA...
1004	SDT	NIMBUS6...	BY...	DATA...

FILEINFO		
PB	FILE	NOTAPE
17400.0	2	1003
17401.0	3	1003
17457.0	2	1004
17459.0	3	1004

RECINFO											
DATE	TIME	LON	LAT	ALT	ZEN	PB	QUALITY	ELECTR	ILLUMIN	CALIB	SCAN
790103	124549	158.57	-1.00	1112.50	153.63	17400.0	0	ON	NIGHT	NO	OFF
790105	145629	25.90	-0.96	1112.30	153.97	17459.0	0	ON	NIGHT	NO	OFF
790117	110204	2.75	10.43	1105.00	34.11	17457.0	0	ON	NIGHT	NO	OFF
790107	105349	4.82	10.84	1103.00	35.59	17401.0	8	OFF	DAY	YES	ON

QUERIES ON DATABASES

```
select (t.tapetype, t.title1, t.title2, t.title3) as result
from tapeinfo t,
     fileinfo f,
     recinfo r
where t.notape = f.notape
  and f.pb = r.pb
  and r.zen = 153.63
  and r.calib = 'NO'
```

NATURAL LANGUAGE INTERACTION

USER	SYSTEM
1. WHERE IS THE HIGHEST SOIL EROSION POTENTIAL IN THE CHESAPEAKE BAY WATERSHED?	1. WHAT DOES EROSION MEAN?
2. EROSION MEANS	2. WHAT TEMPORAL RANGE DO YOU WISH?
3. PAST SIX YEARS	3. DO YOU WISH ME TO CONSIDER ALL RELEVANT DATA TYPES?
4. YES, BUT PRIORITIZE WITH THE MOST CURRENT	4. SYSTEM NOW TRANSLATES AND PASSES QUERY TO DATA MANAGEMENT CONTROLLER

DATA MANAGEMENT CONTROLLER

1. REVIEW QUERY AND IDENTIFIES WHAT INFORMATION IS NEEDED, IF IT IS AVAILABLE AND WHERE IT IS LOCATED

2. INTERACTS WITH APPROPRIATE DATA BASE AND SEARCH STRATEGY TO DETERMINE IF QUERY IS SATISFIED

3. IF ANSWER TO QUERY IS NOT RESIDENT AT TOP OF "ABSTRACTION PYRAMID" CONTROLLER THEN COMPARES DATA SELECTED AGAINST DATA MISSING OR INSUFFICIENT

4. CONTROLLER THEN POSTULATES ALTERNATIVES AND SUGGESTS THESE TO USER - IF CONCURRENCE, SYSTEM CONTINUES TO SATISFY QUERY. (THIS IS A REVIEW OF CENTRAL DIRECTORY, CATALOG AND INVENTORY)

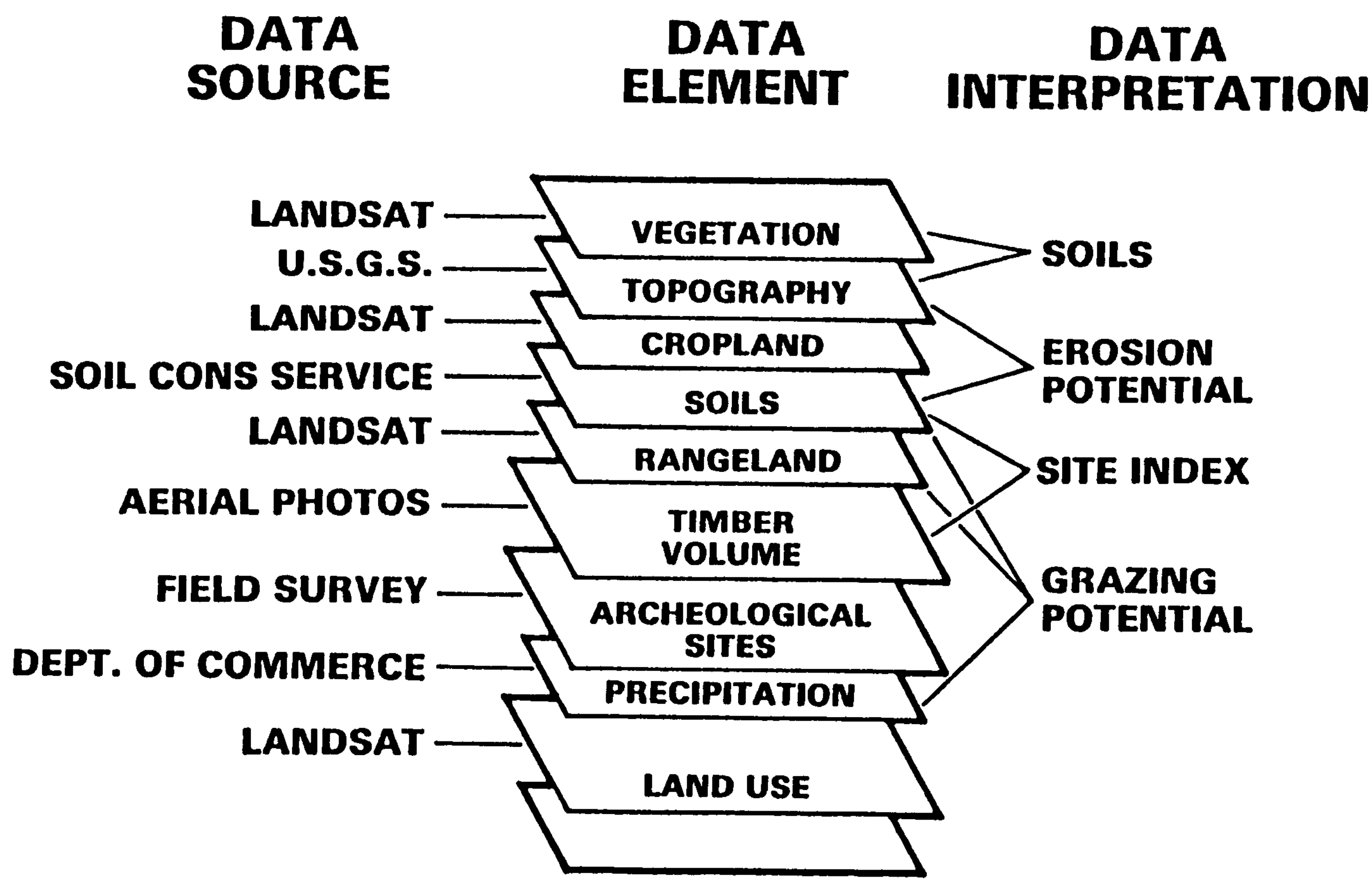
5. CONTROLLER REVIEWS EXISTING DATA AND SETS UP OPERATIONAL PROCEDURES FOR DATA THAT NEEDS PROCESSING FROM "RAW" FORM TO "CLASSIFIED" FORM

   THIS INCLUDES:

   -- REGISTRATION
   -- CORRELATION (GEOLOGY WITH HYDROLOGY)
   -- PLOTTING VARIANCE AMONG DATA SETS
   -- SPATIAL MODELING
   -- IMAGE GENERATION

   RESULTS WILL BE RETURNED AND DISPLAYED AT THE USER'S WORKSTATION FOR CONCURRENCE OR FURTHER ITERATION

6. THE ABOVE OPERATIONS WILL BE A CONTINUOUS SERIAL DIALOG BETWEEN USER, "CONTROLLER" AND THE RULES CONTAINED IN EACH DATA BASE AND LEVELS OF "ABSTRACTION." CERTAIN PROCESSES MAY BE PERFORMED ON SPECIALIZED MACHINES (AP, SYMBOLIC, MPP, ETC.) AS APPROPRIATE. IT WILL ALSO BE THE JOB OF THE CONTROLLER USING "PROCEDURAL KNOWLEDGE" TO DETERMINE EACH PROCESS


DATA SOURCE
DATA ELEMENT
DATA INTERPRETATION
LANDSAT
U.S.G.S.
LANDSAT
SOIL CONS SERVICE
LANDSAT
AERIAL PHOTOS
FIELD SURVEY
DEPT. OF COMMERCE
LANDSAT
VEGETATION
TOPOGRAPHY
CROPLAND
SOILS
RANGELAND
TIMBER VOLUME
ARCHEOLOGICAL SITES
PRECIPITATION
LAND USE
SOILS
EROSION POTENTIAL
SITE INDEX
GRAZING POTENTIAL

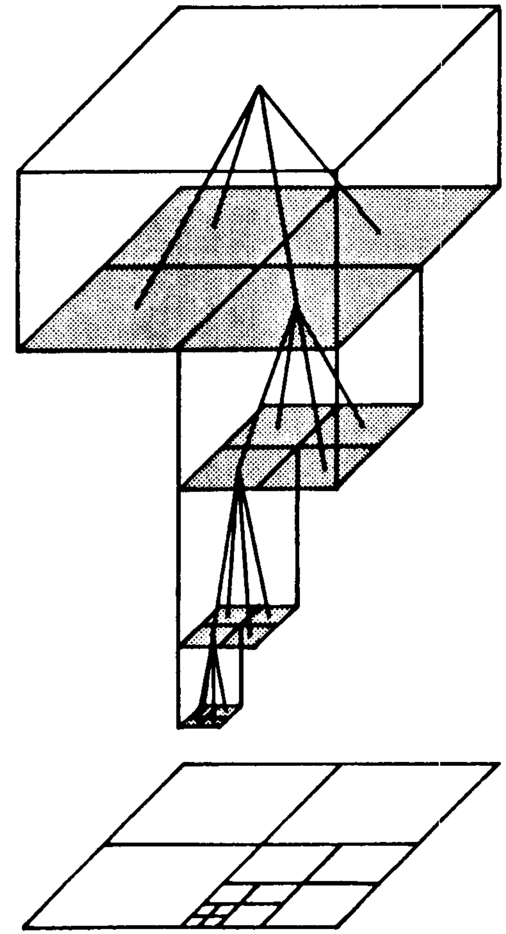


FIGURE 4. QUADTREE STRUCTURE (FROM HUNTER AND STEIGLITZ, 1979)

C-2

USER	INTELLIGENT INTERFACE	DBMS
1. SELECT SPATIAL DATA FOR THE CBWS FOR 1983, 84, 85	1A. NATURAL LANGUAGE PROCESSOR (NLP) TRANSLATES THE QUERY INTO SQL AND LAUNCHES QUERY TO TEST FOR VALID SYNTAX ONLY	2B. DATA MANAGEMENT SYSTEM ACCEPTS QUERY AND FAILS TO FIND CHESAPEAKE BAY LOCATION
	. DATA MANAGEMENT CONTROLLER LOOKS FOR RESPONSE AND DETERMINES THAT QUERY FAILED. . NLP RESPONDS THAT IT DOES NOT UNDERSTAND THE LOCATION (CBWS). . DATA MANAGEMENT CONTROLLER AMENDS RESPONSE TO INCLUDE APPLICATION OF QUERY AND MINIMAL DATA NECESSARY	
2. USER PROVIDES LAT/LONG INDICATES IT IS FOR SOIL EROSION AND REQUIRE 10 IMAGE MIN.	2A. DATA MANAGEMENT CONTROLLER REVIEWS RESPONSE AND BASED ON KNOWLEDGE BASE (KB) ASKS USER MINIMUM TIME BETWEEN IMAGES	
3. USER RESPONDS 3 MONTHS	3A. DATA MANAGEMENT CONTROLLER CREATES AN ENGLISH QUERY AS FOLLOWS SELECT DATA SETS FROM LANDSAT CREATED FROM THE TM SENSOR, IN BAND 2, 3, 4 FOR QUARTERS 1, 2, 3, 4 FOR YEARS 1983, 84, 85 FOR LAND AREA LAT/LONG QUERY TRANSLATED BY NLP	3B. DATA MANAGEMENT SYSTEM SEARCHES AND FINDS DATA FOR USER

RULES FOR 3A

1. IF QUERY SUPPORTS SOIL EROSION ANALYSIS THEN THE TYPE OF SENSOR THAT SHOULD BE USED IS INFRARED IMAGING

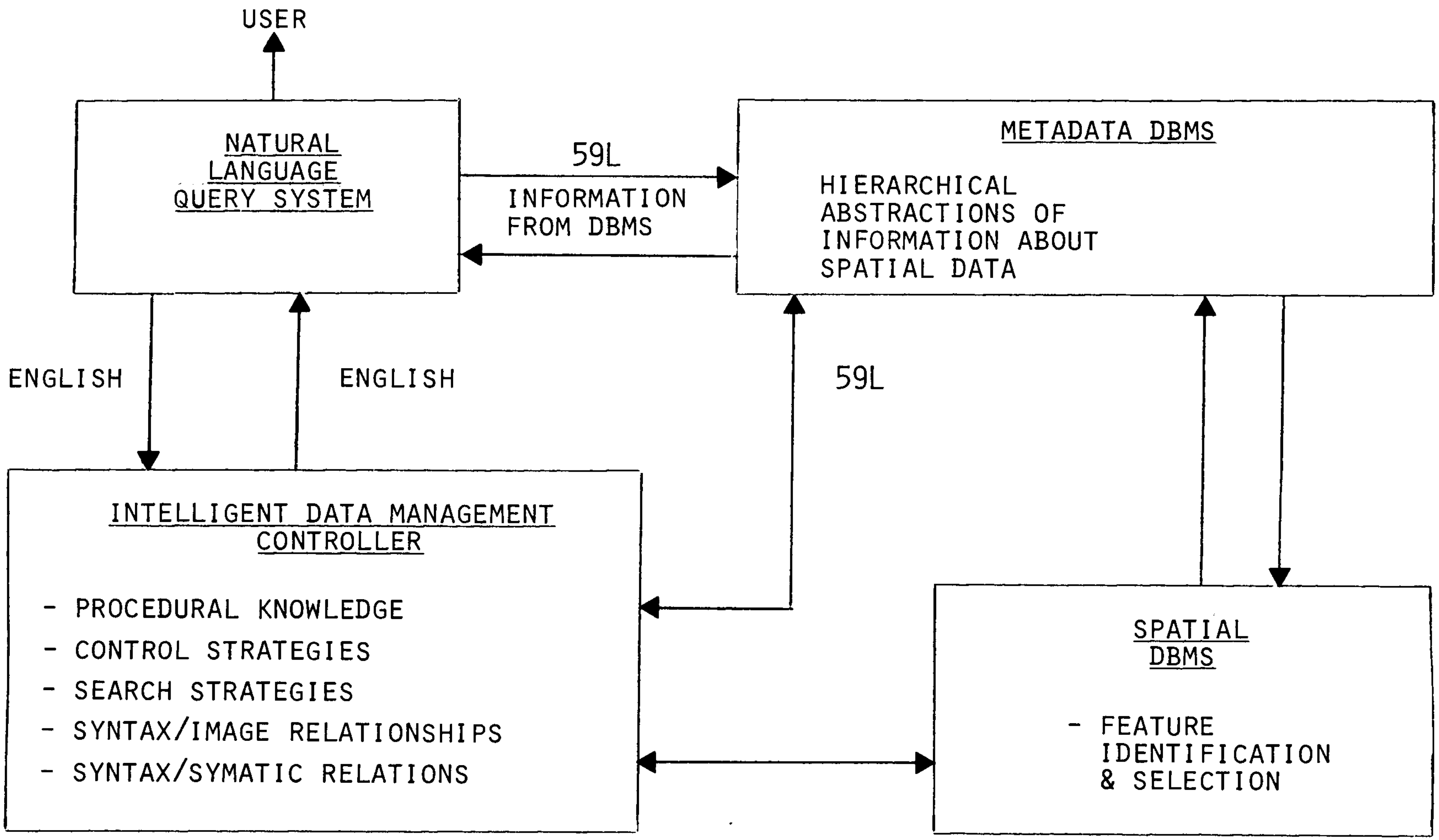
2. IF YEARS FOR QUERY IS AFTER 1983 THEN THE IMAGE DATA SHOULD BE THE THEMATIC MAPPER OF LANDSAT

3. IF THE APPLICATION IS FOR SOIL EROSION AND THE DATA DESIRED IS IN THE INFRARED THEN THE INFORMATION DESIRED IS BANDS 2, 3, 4

4. IF THE INFORMATION DESIRED IS BY YEAR THEN THE NUMBER OF IMAGES MUST BE SELECTED BASED ON MINIMUM DURATION BETWEEN IMAGES

5. IF THE APPLICATION IS FOR SOIL EROSION THEN CLOUD COVER MUST BE LESS THAN 30%

INTELLIGENT DATA MANAGEMENT CONCEPT



Software Management Environment

Frank McGarry - Goddard Space Flight Center

There exists an overall RTOP effort, supported by Code R, which is attempting to define, assess and integrate software measures and tools into an environment that will aid the management process for software development. During this briefing, examples of three specific areas of work are discussed. These areas are:

1. Research into the development of Software Design measures
2. Research into the development of Software Specification measures
3. Attempts at integrating identified measures and models into a 'Dynamic Management Information Tool'(DYNAMITE)

A. Design Measures

By closely studying and monitoring numerous software development projects at Goddard, several promising approaches to assessing design characteristics have been developed. One is a mechanism by which the classical measures of 'strength' and 'coupling' are extracted at design time and are used to predict reliability and overall quality of the software product. Early indications imply a high correlation with overall reliability and productivity of the end product.

A second design measure being investigated is one that tracks the evolution of the general architecture of a software system during the design process. Early results imply that projects with apparent disparity between the evolution of 'data structure' and 'control structure' will be more likely to have reliability problems later.

B. Specifications Measures

Attempts were carried out at developing measures by which software specifications could be used to predict complexity, reliability and other characteristics of the software product. One approach was utilized where objective measures (or counts) of such items as number of external file requirements, size of specifications, number of processes, etc. resulted in a dead end; no reliable measures were identified.

A second approach, called the Composite Specification Model (CSM), was applied to an ongoing project at Goddard and the early indicators show promise for this approach as a means for truly analyzing software complexity during the requirements phase.

C. Dynamic Management Information Tool (DYNAMITE)

A software tool has been designed which attempts to support the user in <u>assessing</u> software quality as well as <u>predicting</u> future events such as schedules and cost. The tool has at its disposal a set of historical data on completed projects, a set of measures and models developed by this research effort, as well as an 'expert system' which contains a set of software development 'rules', also developed by this and other research projects.

This tool exists in the prototype stage and currently contains about 100 rules. It utilizes the KMS inference engine and currently has the capability of performing some very basic predictions and assessments of active projects where some very basic development information is made known to it.

N87-29133

p.26

SOFTWARE MANAGEMENT ENVIRONMENT FOR NASA

Frank E. McGarry

NASA GSFC

SOFTWARE MANAGEMENT ENVIRONMENT FOR NASA

OBJECTIVE

Develop. assess and implement software management aids
(tools, measures, techniques)
leading to an environment producing software of 'increased quality'
(reliability and life cycle cost)

AREAS OF INVESTIGATION

Design and specification measures
Management tools (including rapid prototyping aids)

NASA

SOFTWARE MANAGEMENT ENVIRONMENT

AREAS OF CONSIDERATION

DESIGN/SPECIFICATION MEASURES

- Can we determine 'quality' of design (or specs)?
- What is 'quality' for design?
- How do we determine trade-offs for various design approaches?
- Can I determine early what part of system is 'easy' or 'hard'?

MANAGEMENT TOOLS

- Given existing development information, PREDICT-ASSESS-SELECT-CONTROL
- Automatically determine quality of design
- Automatically determine 'improved' design
- Evaluate specs

SOFTWARE ENGINEERING LABORATORY DATA STUDIED

TYPE OF SOFTWARE: SCIENTIFIC, GROUND-BASED, INTERACTIVE GRAPHIC, MODERATE RELIABILITY AND RESPONSE REQUIREMENTS

LANGUAGES: 85% FORTRAN, 15% ASSEMBLER MACROS

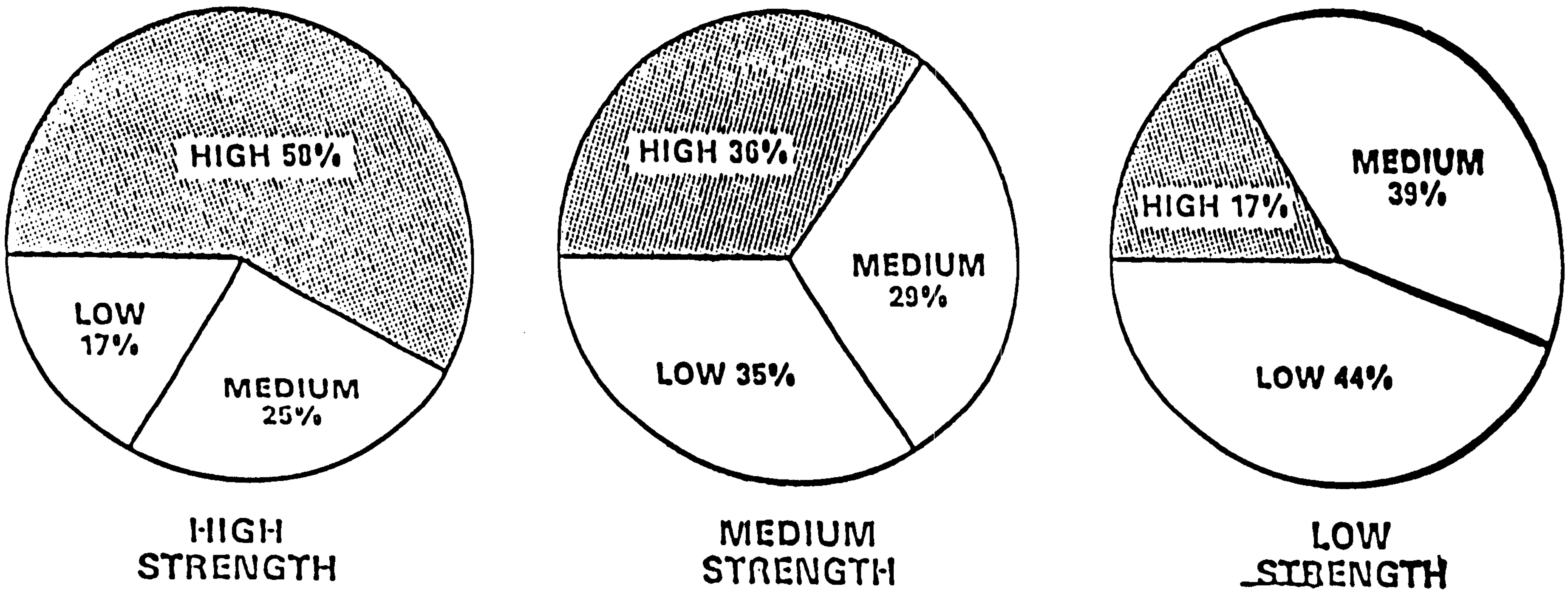
MACHINES: IBM S/360 AND 4341, BATCH WITH TSO

PROJECT CHARACTERISTICS:	AVERAGE	HIGH	LOW
DURATION (MONTHS)	15.6	20.5	12.9
EFFORT (STAFF-YEARS)	8.0	11.5	2.4
SIZE (1000 LOC)			
DEVELOPED	57.0	111.3	21.5
DELIVERED	62.0	112.0	32.8
STAFF (FULL-TIME EQUIV.)			
AVERAGE	5.4	6.0	1.9
PEAK	10.0	13.9	3.8
INDIVIDUALS	14	17	7
APPLICATION EXPERIENCE			
MANAGERS	5.8	6.5	5.0
TECHNICAL STAFF	4.0	5.0	2.9
OVERALL EXPERIENCE			
MANAGERS	10.0	14.0	8.4
TECHNICAL STAFF	8.5	11.0	7.0

SAMPLE: 22 SYSTEMS USING A VARIETY OF TECHNOLOGIES

438-CAR-(33)-8

STRENGTH AS A DESIGN MEASURE



HIGH STRENGTH IMPLIES HIGH RELIABILITY

BASED ON: * 480 Modules
* 3 Projects

RELIABILITY:

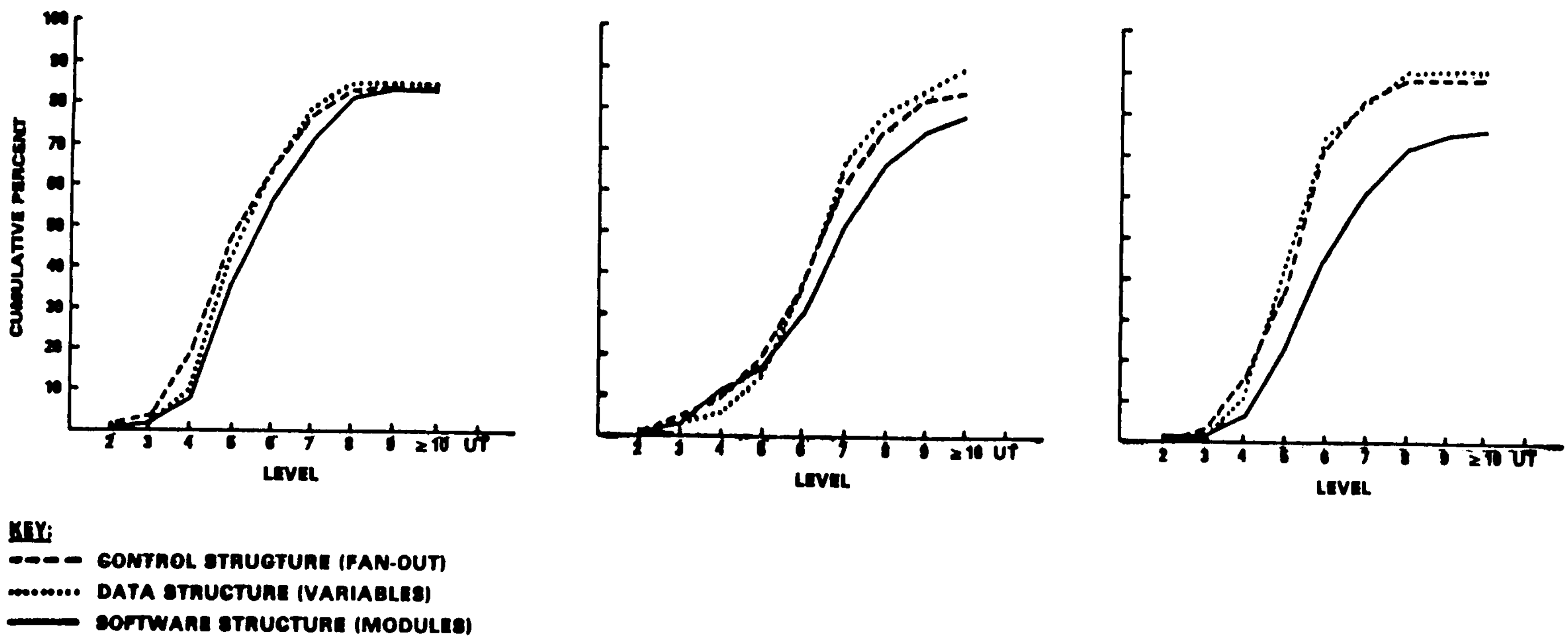
High Error=0/1000 L.O.C.

Med Error.LE.2/1000 L.O.C.

Low Error.GT.2/1000 L.O.C.

DESIGN IS A PARTITIONING OF STRUCTURE

PROJECT 1: GOOD | PROJECT 2: MEDIUM | PROJECT 3: POOR



Developing 'Specification' Measures

OUR APPROACH

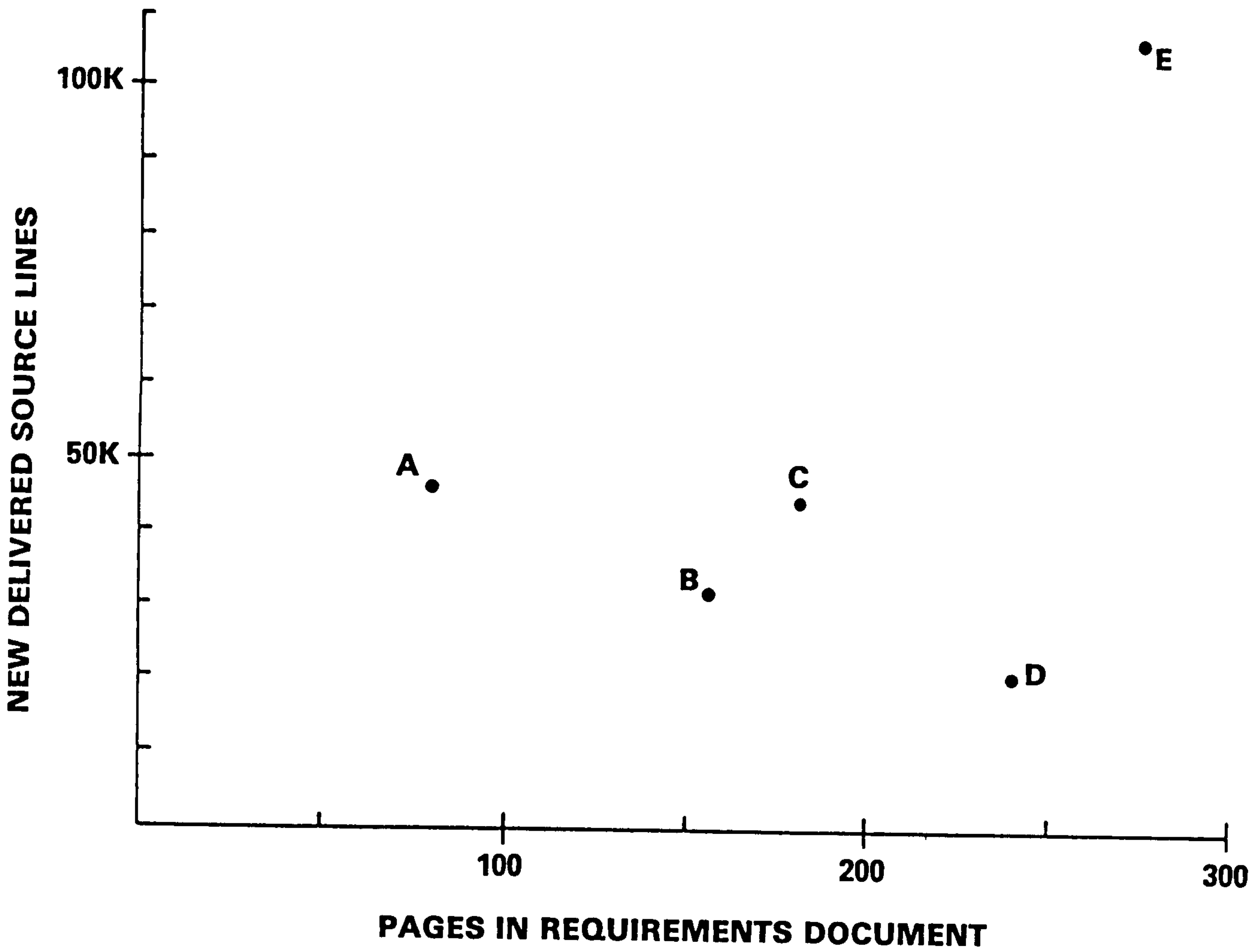
FOCUS: OBJECTIVE MEASURES

PROCEDURE: DEFINED 29 EXPLICIT MEASURES BASED ON EXISTING REQUIREMENTS SPECIFICATIONS

NUMBER OF PAGES
NUMBER OF CONSTRAINTS
NUMBER OF I/O REQUIREMENTS
o
o
o

RESULT: MEASURES WERE EXTRACTABLE BUT NOT USEFUL

FIVE FLIGHT DYNAMICS SOFTWARE PROJECTS NEW SOURCE LINES VS. PAGES OF REQUIREMENTS



LESSON: TO DEVELOP OBJECTIVE SPECIFICATION MEASURES, REPRESENTATION IS EVERYTHING!

OUR REVISED APPROACH

STEP 1: PROPOSE A NEW REPRESENTATION

STEP 2: DEFINE SPECIFICATION MEASURES BASED ON IT

STEP 3: APPLY IT TO A REAL SYSTEM

STEP 4: EXTRACT THE MEASURES

STEP 5: ASSESS THE PROCESS AND THE RESULTING MEASURES

STEP 1: PROPOSE A NEW REPRESENTATION

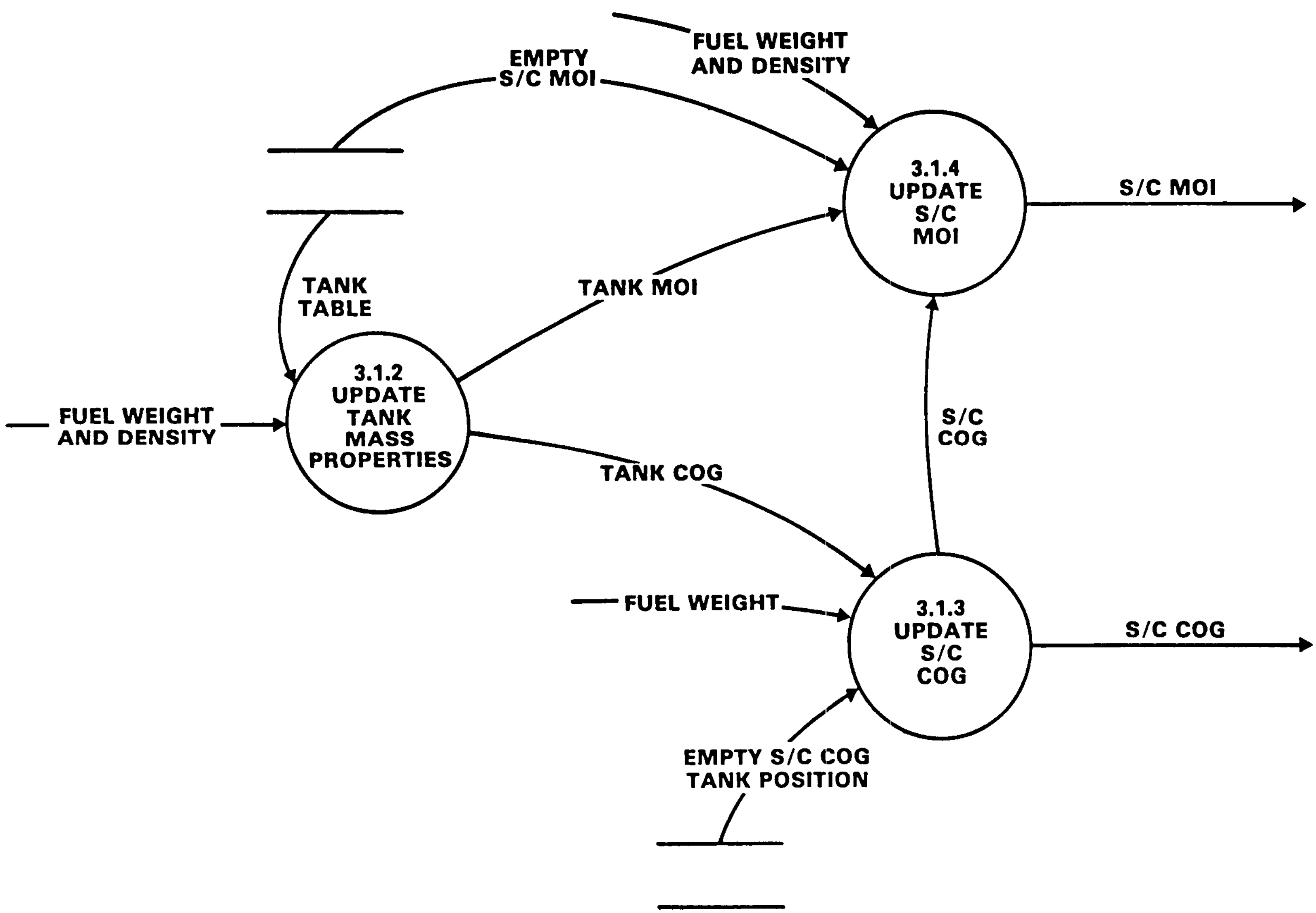
COMPOSITE SPECIFICATION MODEL (CSM)

RATIONALE: REQUIREMENTS FOR COMPLEX SOFTWARE NEED TO BE SPECIFIED FROM MULTIPLE VIEWPOINTS

VIEWPOINT	NOTATION
• FUNCTIONAL	• DATA FLOW
• CONTEXTUAL	• ENTITY/RELATIONSHIP
• DYNAMIC	• STATE/TRANSITION

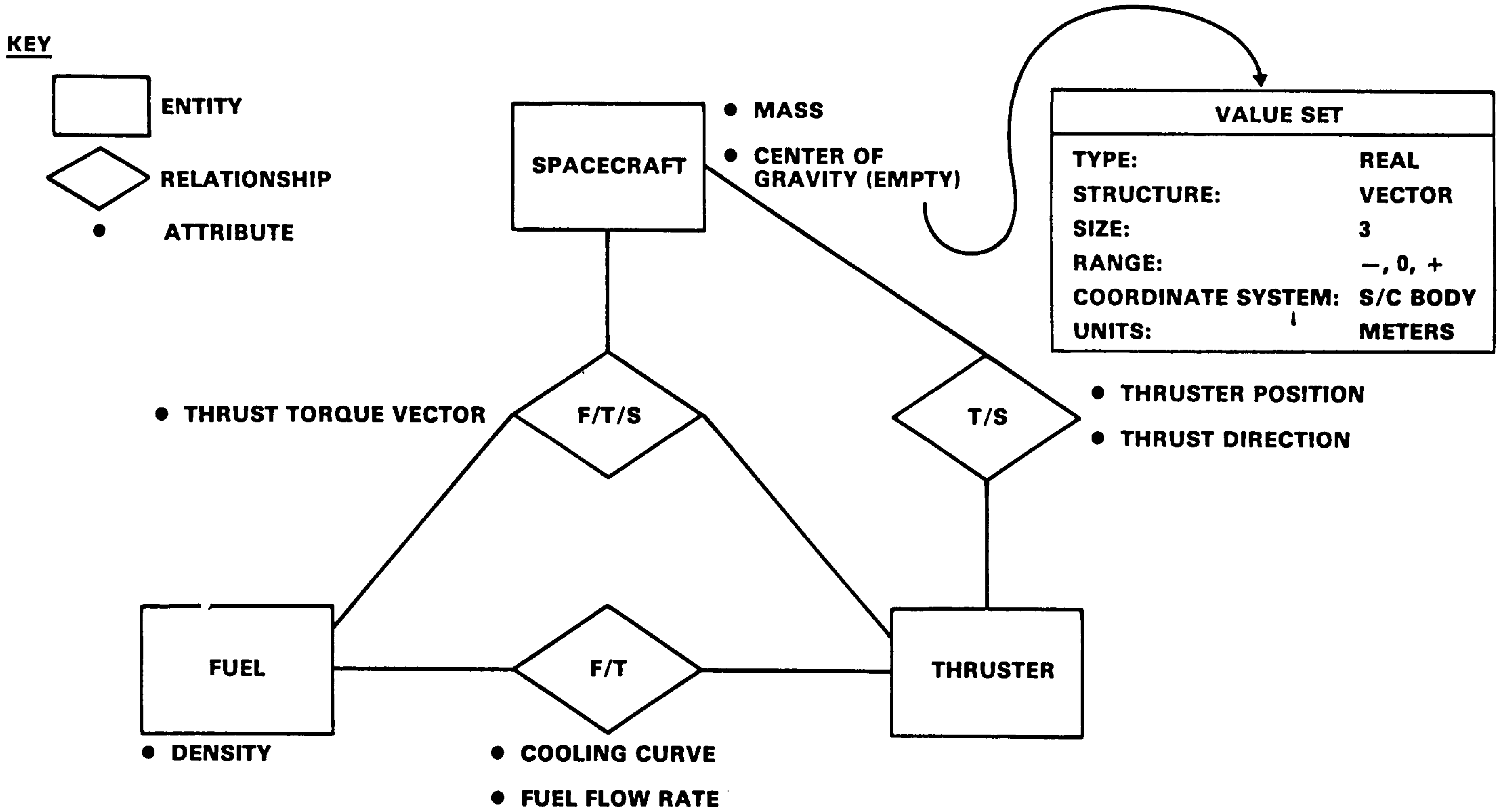
714-AGR-(89c)

EXAMPLE OF FUNCTIONAL VIEW



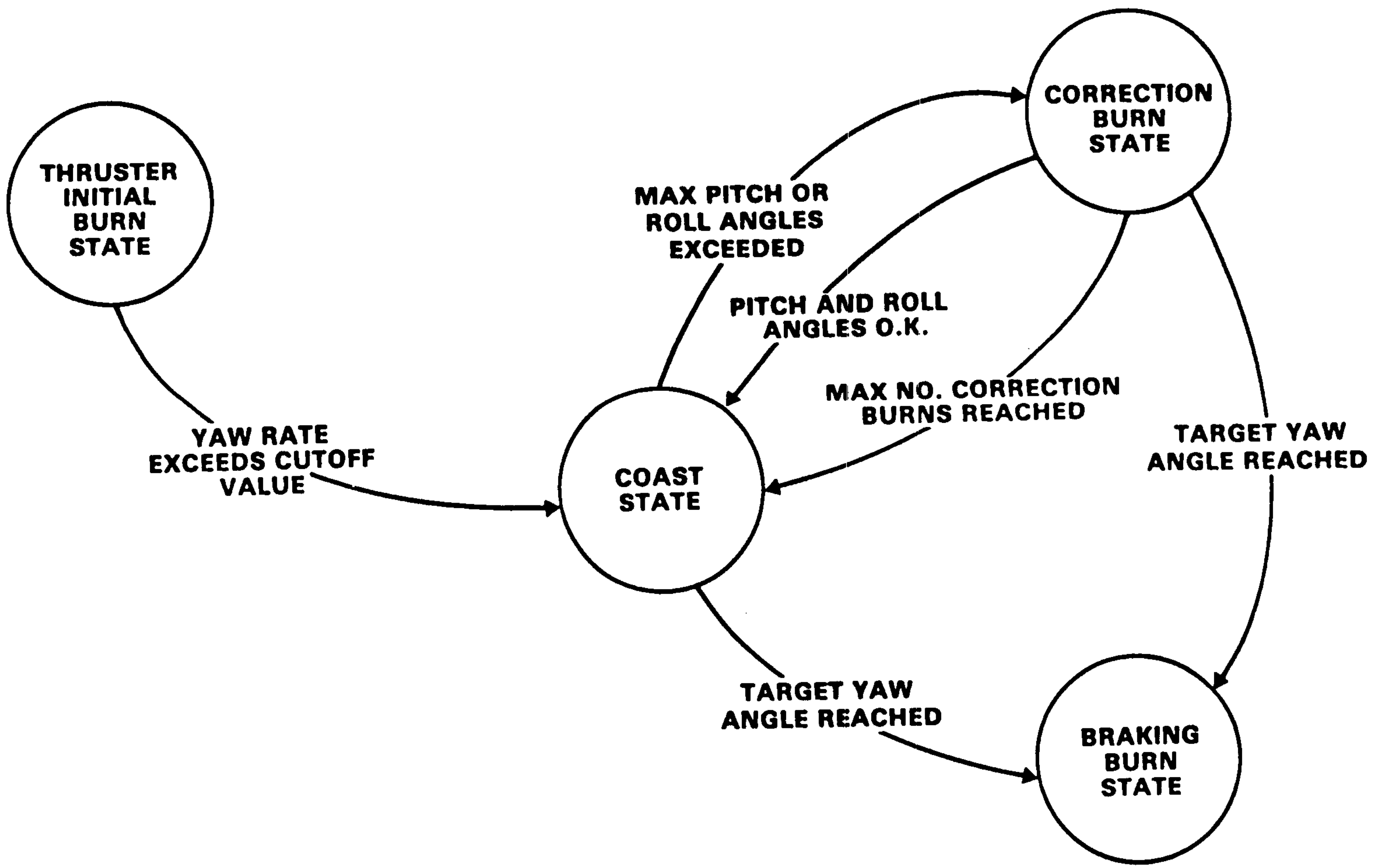
714-AGR-(89c)

EXAMPLE OF CONTEXTUAL VIEW



714-AGR-(89c)

EXAMPLE OF DYNAMIC VIEW (STATES AND TRANSITIONS)



714-AGR-(98a)

STEP 2: DEFINE MEASURES BASED ON THE COMPOSITE SPECIFICATION MODEL

58 MEASURES DEFINED

EXPLICIT	ANALYTIC
NUMBER OF FUNCTIONAL PRIMITIVES	WEIGHTED FUNCTION
NUMBER OF DATA ITEMS	RELATION DENSITY
NUMBER OF STATES	ARC WEIGHT
• • •	• • •

714-AGR-(89)

STEP 3: APPLY THE COMPOSITE SPECIFICATION MODEL TO A REAL SYSTEM

- YAW MANEUVER CONTROL UTILITY OF EARTH RADIATION BUDGET SATELLITE (ERBS)
- FORTRAN
- 11,200 DELIVERED SOURCE LINES
- 85 MODULES

714-AGR-(89a)

STEP 4 EXTRACT THE MEASURES

MEASURE	VALUE
FUNCTIONAL VIEW	
• FUNCTIONAL PRIMITIVES	39
• INTERFACE COUNT	3
• INTERNAL ARCS	60
• INTERNAL DATA ITEMS	42
• SYSTEM IN/OUT DATA ITEMS	67
• FILE IN/OUT DATA ITEMS	74
• WEIGHTED FUNCTION	688
CONTEXTUAL VIEW	
• ENTITIES	11
• EVENTS	14
• RELATIONS	19
• ATTRIBUTES	91
• VALUE SETS	29
DYNAMIC VIEW	
• STATES	7
• TRANSITIONS	11

714-AGR-(89a)

STEP 5: ASSESS THE PROCESS AND RESULTING MEASURES

PROCESS

- EFFORT REQUIRED FOR CSM MAY REDUCE EFFORT IN LATER PHASES
  - 2.1 STAFF MONTHS FOR TRADITIONAL REQUIREMENTS ANALYSIS
  - 1.7 STAFF MONTHS FOR BUILDING CSM

RESULTING MEASURES

- HUMAN JUDGMENT STILL IS A FACTOR
- NEED TO MEASURE MORE PROJECTS

714-AGR-(90f)

CONCLUSIONS

- OBJECTIVE SPECIFICATION MEASURES NEED DISCIPLINED REPRESENTATION OF REQUIREMENTS

- BUILDING THE CSM IS FEASIBLE
  - YIELDS OBJECTIVE SPECIFICATION MEASURES
  - MULTIPLE VIEWS ARE MORE REVEALING
  - MORE EFFECTIVE REPRESENTATION TO BEGIN DESIGN

- CAPTURING THE CONTEXT OF A SYSTEM IS BENEFICIAL
  - SOURCE OF CHANGES TO THE SYSTEM
  - LOGICAL PREDECESSOR OF OBJECT-ORIENTED DESIGN

714-AGR-(89f)

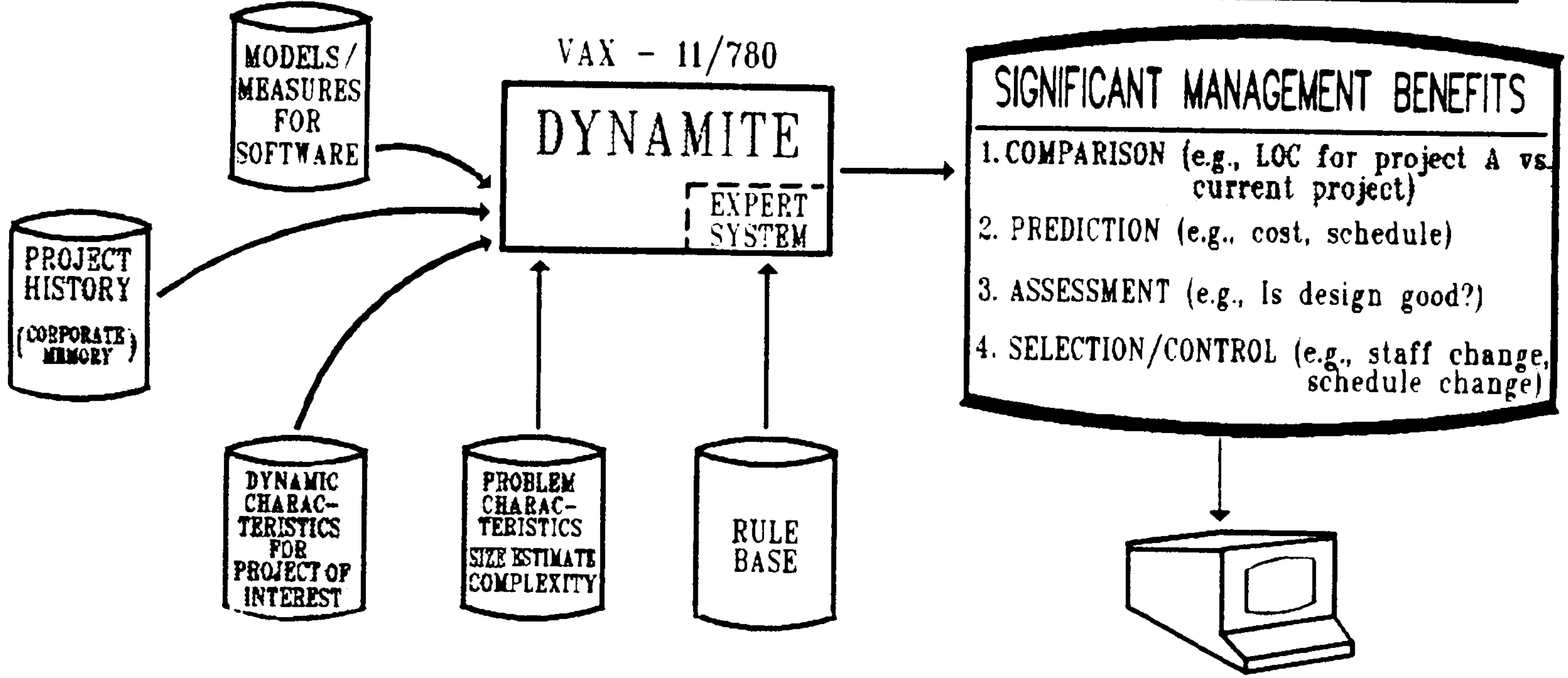
DYNAmic Management Information Tool

The Idea

INPUT	OUTPUT
1. Verified Measures/models for Development (e.g. 40-20-40 Rule or Rayleigh Curve)	1. PREDICT (e.g. When will project be complete?)
2. Past Project Histories (e.g. Staffing Profiles)	2. ASSESS (e.g. Tesing procedures are bad)
3. Verified 'RULES' of Software Development (e.g. If excessive ECR's then specs are of poor quality)	3. COMPARE (e.g. Relative to past projects, the code development rate is very low.)
4. Current Project Development Data (e.g. Staffing, Changes, Resource Consumption)	4. SELECT/CONTROL (e.g. Use tighter testing standards for this project.)

SOFTWARE MANAGEMENT ENVIRONMENT

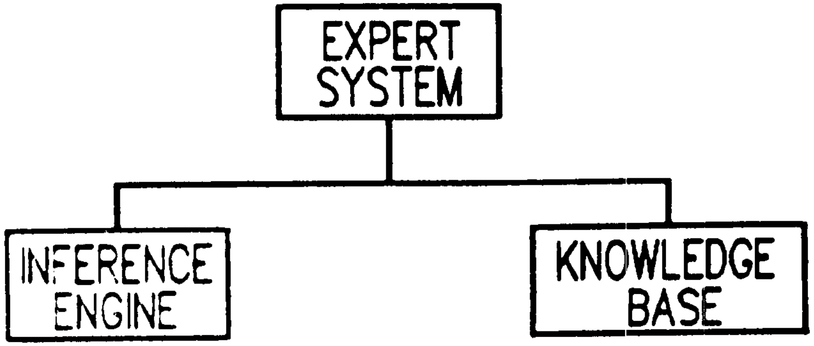
DYNAMIC MANAGEMENT INFORMATION TOOL (DYNAMITE)



* SHADED = FY84 capabilities
* OPEN = Planned Work

NASA

DYNAMITE EXPERT SYSTEM

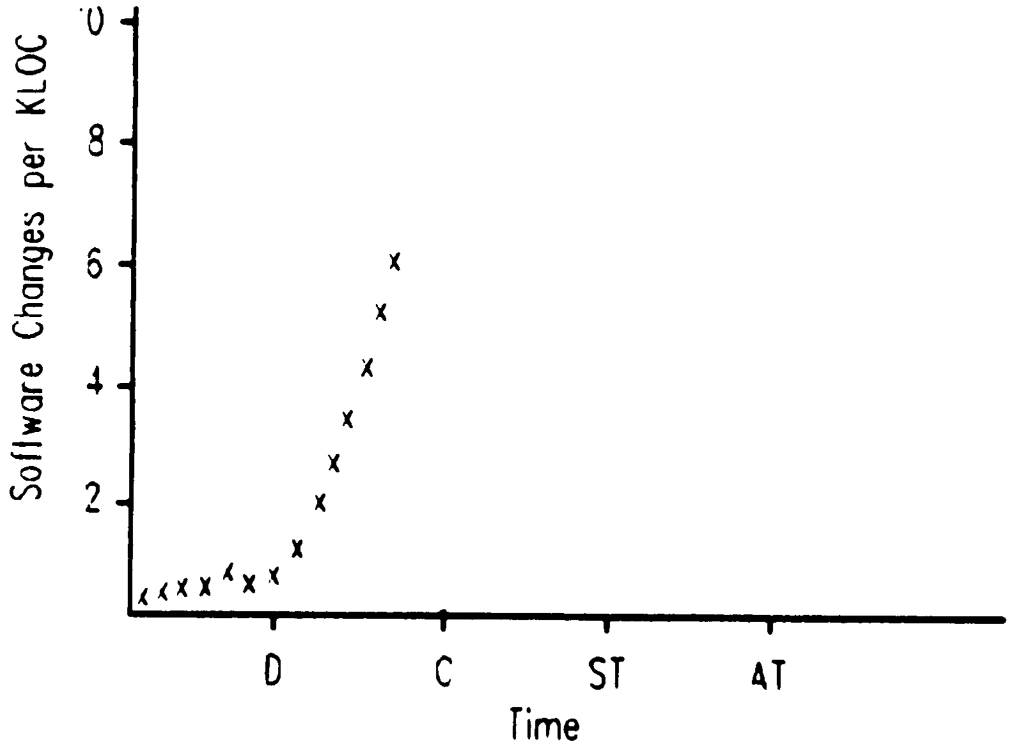


University of
Maryland Tool
KMS - Knowledge
Management System

3 representations of knowledge
- Rule based
  approximately 100 rules
- Frame based
- Hypothesize and Test

DYNAMITE SCENARIO UTILIZING EXPERT SYSTEM

STEP 1
Retrieve data from <u>Dynamic project file</u>



SAMPLE RULES

RULE 1: If computer run per line of source code is above normal
and in early code phase then
interpretation is

- lots of testing 75%
- error-prone code 75%
- high complexity or tough problem 50%
- low productivity 25%
- removal of code by testing or transporting 25%

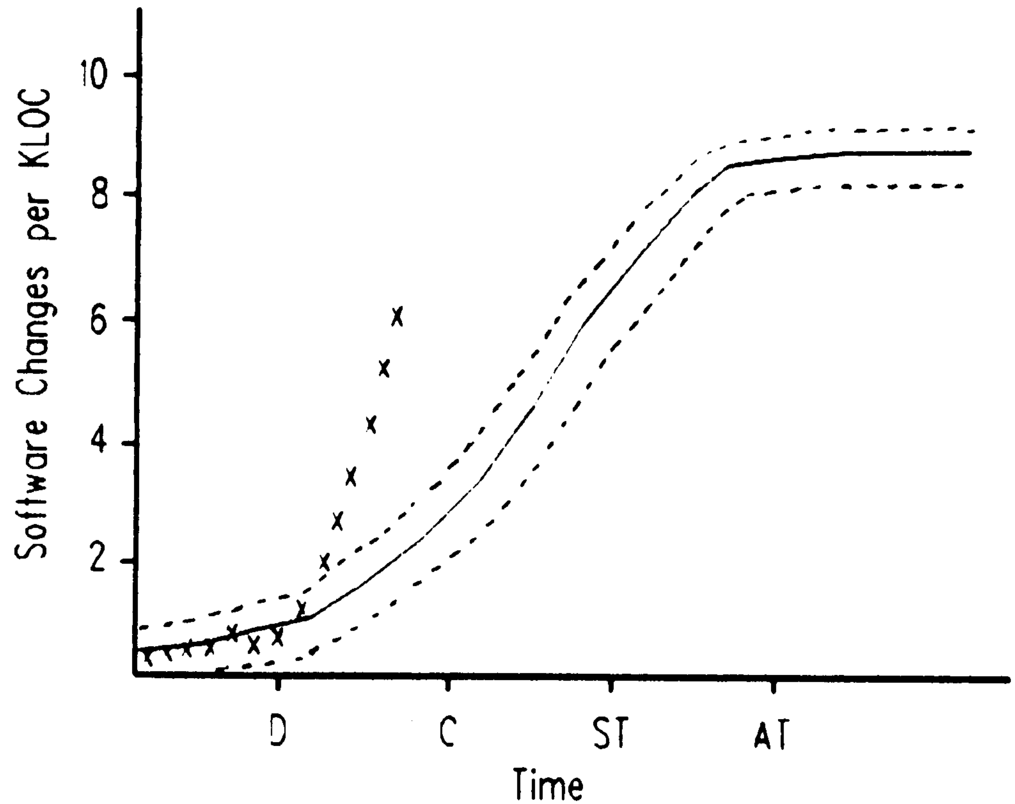
RULE 2: If software changes per line of source code is above normal
and in system test phase then
interpretation is

- error-prone code 75%
- unstable specification 75%
- loose configuration management or unstructured development 75%
- good testing or good test plan 25%
- removal of code by testing or transporting 25%
- near build or milestone date 25%

DYNAMITE SCENARIO UTILIZING EXPERT SYSTEM

STEP 1
Retrieve data
from <u>Dynamic</u>
<u>project file</u>

STEP 2
Retrieve Mode
of SEL Experience

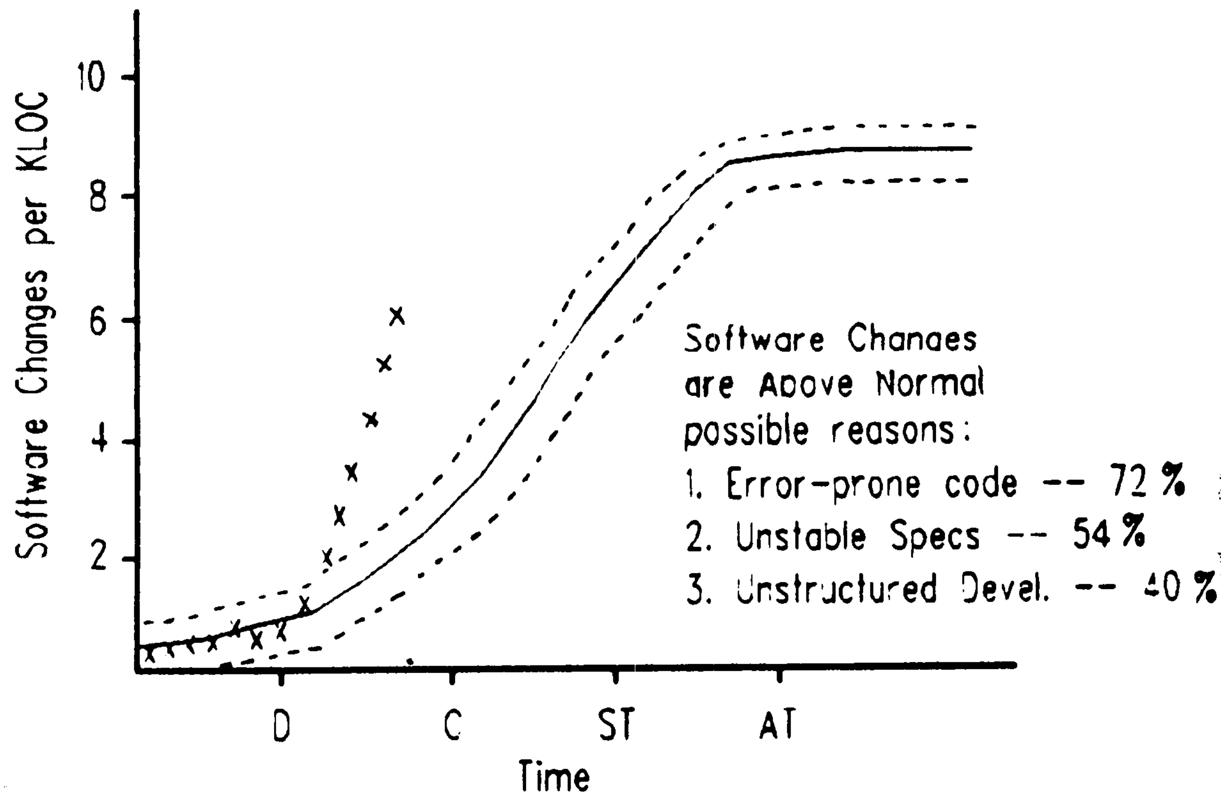


DYNAMITE SCENARIO UTILIZING EXPERT SYSTEM

STEP 1
Retrieve data
from <u>Dynamic</u>
<u>project file</u>

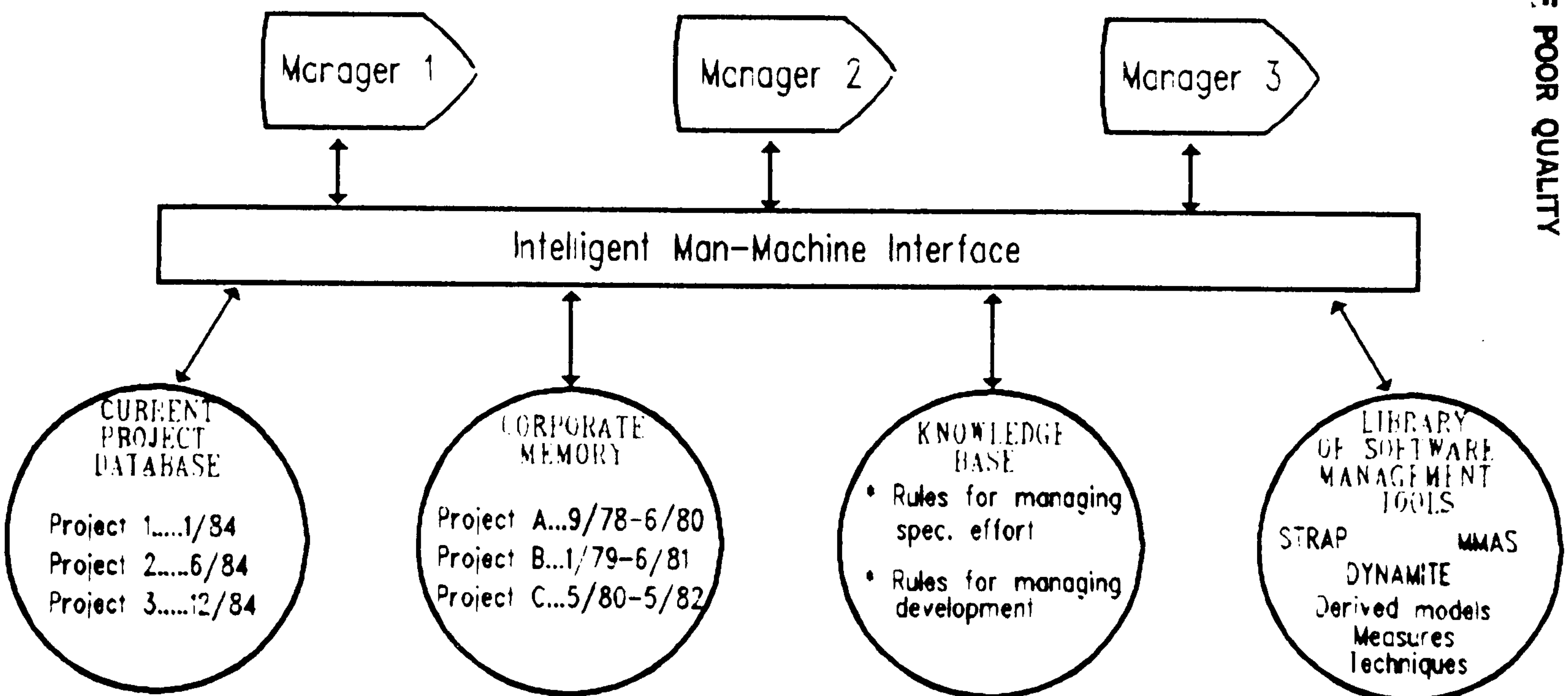
STEP 2
Retrieve Mode
of SEL Experience

STEP 3
Assess meaning
of Comparison



SOFTWARE MANAGEMENT ENVIRONMENT

Functional Diagram



ORIGINAL PAGE IS OF POOR QUALITY

NASA

N87-29134

P.21

MASSIVELY PARALLEL PROCESSOR

The Massively Parallel Processor (MPP) was delivered to NASA Goddard in May 1983, by Goodyear Aerospace Corporation following four years of development. MPP is the product of a research and technology program designed to evaluate the application of a computer architecture containing thousands of processing elements (PE's), all operating concurrently, to the computational requirements of the sensors of the next decade.

Major applications of the MPP are in the area of image processing (where operands are often small integers) from very high spatial resolution passive image sensors, signal processing of RADAR data, and numerical modeling simulations of climate.

At the heart of the MPP is a custom integrated circuit chip containing 8 PE's. 2112 of these chips have been combined with commercial memory and control chips to pack into 18 square feet of floor space the capability to perform 400 million floating point operations per second and 6 billion fixed point operations per second. The system can be programmed in assembly language or a high level language, Parallel Pascal, which is an extension of standard Pascal. Research is underway to develop techniques and programming tools to better expose the power of the massive parallelism.

Because the MPP is a one-of-a-kind system and is not a commercial product supported by the field engineering wing of the manufacturer's organization, NASA has assumed responsibility for providing all spare printed circuit boards, spare component parts, diagnostic software, and an on-site maintenance engineer. Spares exist for only 11% of the printed circuit board assemblies so hardware failures must often be traced to the failed component while users wait. This situation has proved workable, though ocassionally tenuous when several failures occur close together in time.

The MPP is being developed as a national resource around which will grow a diverse community of science and applications users requiring its unique parallel processing capabilities. Their work will help determine the practical computational limits of the MPP's parallel architecture. A Space Science and Applications Notice (AN) titled "Computational Investigations Utilizing the Massively Parallel Processor" was issued in December 1984. It announced an ongoing opportunity to carry out computational investigations exploiting the unique characteristics of the MPP. Despite the fact that no funding was offered, forty proposals were received. Their topics were spread almost evenly across the categories of signal and image processing, earth sciences, physics, and computer science. Those investigators whose proposals are accepted will form the first MPP working group. Their experiences and recommendations will play a large factor in motivating future enhancements to the current system and in justifying future NASA efforts in parallel processor development.

THE MASSIVELY PARALLEL PROCESSOR (MPP)

506-58-16 DATA SYSTEMS

506-54-56 COMPUTER SCIENCE

COMPUTER SCIENCE / DATA SYSTEMS
TECHNICAL SYMPOSIUM

LEESBURG, VA

APRIL 17, 1985

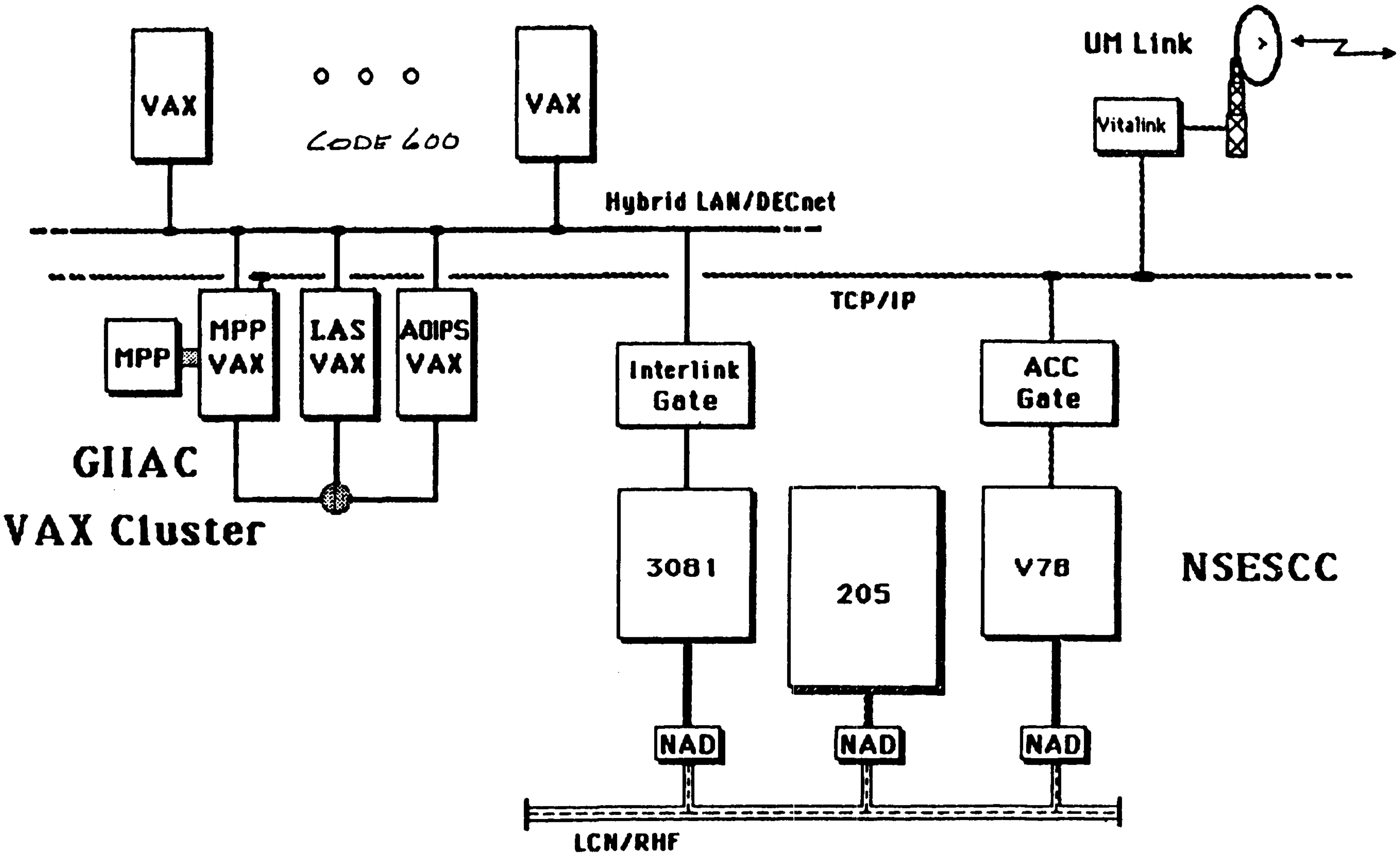
JIM FISCHER NASA/GSFC CODE 635

OVERVIEW

- BACKGROUND AND STATUS
- ARCHITECTURE AND PROGRAMMING
- HARDWARE RELIABILITY
- APPLICATIONS - TODAY
- DEVELOPMENT AS A NATIONAL RESOURCE FOR PARALLEL ALGORITHM RESEARCH

BACKGROUND

- THE MPP IS A 2-DIMENSIONAL ARRAY CONSISTING OF 16,384 (128 X 128) SIMPLE PROCESSORS.
- THE MPP IS THE RESULT OF A NASA R & D PROGRAM TO DEVELOP A HIGH SPEED IMAGE PROCESSING COMPUTER.
- THE INITIAL CONCEPT AND DESIGN OF THE MPP WAS DEVELOPED AT THE GODDARD SPACE FLIGHT CENTER.
- GOODYEAR AEROSPACE CORPORATION DEVELOPED THE MPP SYSTEM UNDER CONTRACT.



FUTURE NETWORK

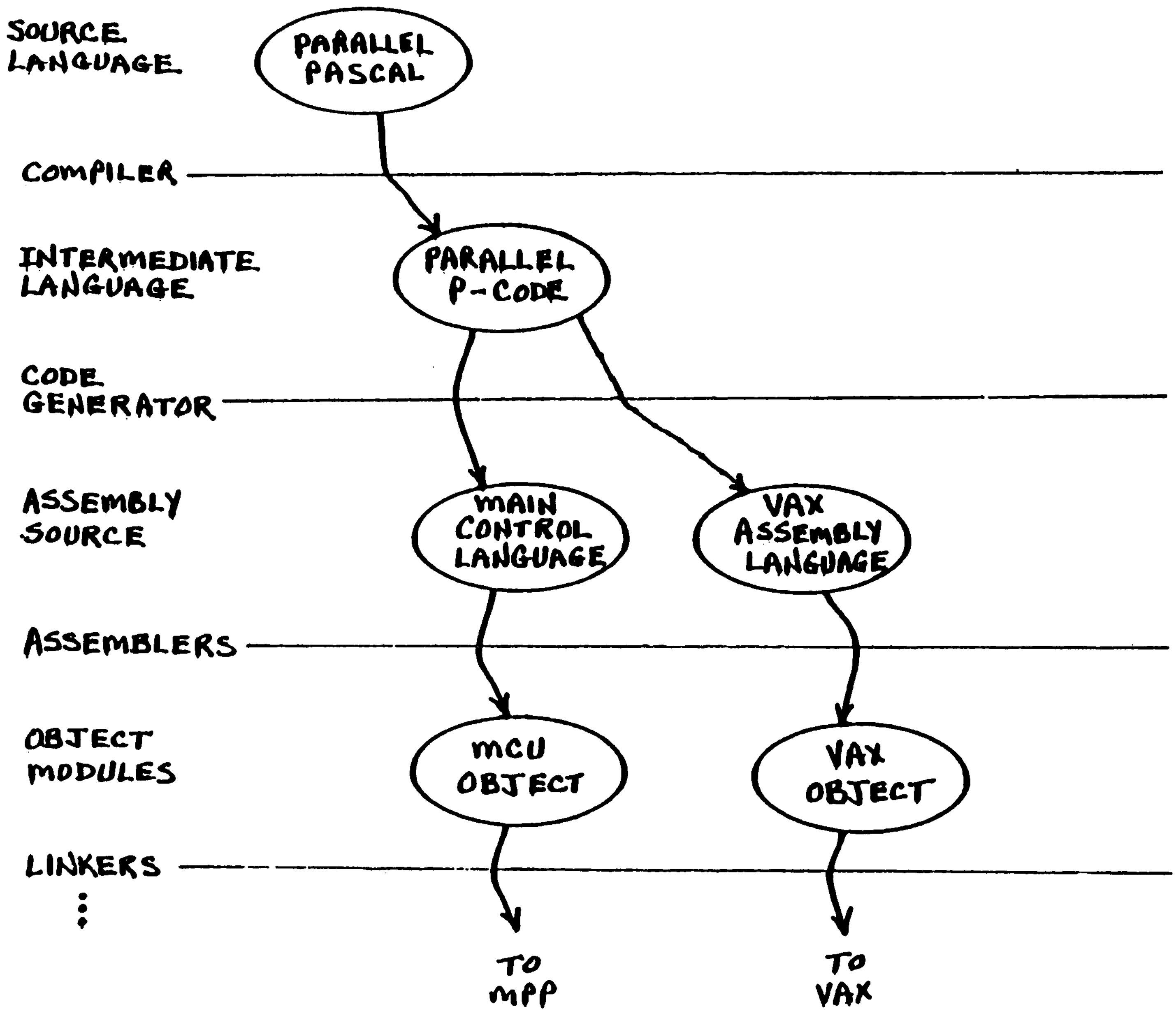
Access to Supercomputers
via Hybrid LAN

PROGRAMMING THE MPP

FOR TIGHT CONTROL OF HARDWARE	-	USE ASSEMBLY LANGUAGE
FOR PROGRAM DEVELOPMENT	-	USE HIGH LEVEL LANGUAGE PARALLEL PASCAL
FOR TRANSPARENT USE	-	USE PRE-PACKAGED ROUTINES REQUEST VIA MENUS
FOR BEST ALGORITHM MAPPING TO HARDWARE	-	(ACTIVE AREA OF RESEARCH)

PARALLEL PASCAL (AN EXTENSION OF STANDARD PASCAL)

```
VAR
  A     : PARALLEL ARRAY [ Ø..127, Ø..127 ] OF Ø..511;
  B, C  : PARALLEL ARRAY [ Ø..127, Ø..127 ] OF Ø..255;
  M     : PARALLEL ARRAY [ Ø..127, Ø..127 ] OF BOOLEAN;

BEGIN
  WHERE M = 1 DO
    A := B + C;          { UP TO 16384 ARRAY ELEMENTS
                           PROCESSED SIMULTANEOUSLY }
END.
```


SOURCE LANGUAGE
PARALLEL PASCAL
COMPILER
INTERMEDIATE LANGUAGE
PARALLEL P-CODE
CODE GENERATOR
ASSEMBLY SOURCE
MAIN CONTROL LANGUAGE
VAX ASSEMBLY LANGUAGE
ASSEMBLERS
OBJECT MODULES
MCU OBJECT
VAX OBJECT
LINKERS
TO MPP
TO VAX

MPP HARDWARE RELIABILITY SINCE NOVEMBER 15, 1984

	UP AVAILABLE	*DEGRADED BUT AVAILABLE	DOWN
NOVEMBER (16 DAYS)	79	-	21
DECEMBER	82	-	18
JANUARY	60	34	6
FEBRUARY	55	15	30
MARCH	96	-	4

ALL NUMBERS REFLECT PERCENTAGE OF NORMALLY SCHEDULED HOURS OF OPERATION:

(3 SHIFTS/DAY 6 DAYS/WEEK)

* THE GODDARD INVENTORY OF A CRITICAL COMMERCIAL CHIP WHICH FAILED OFTEN, HAD NOT BEEN MANUFACTURED IN FOUR YEARS, AND WAS UNPURCHASABLE, WENT TO ZERO. COMPATIBLE 'REJECT' CHIPS WERE EVENTUALLY LOCATED AND THE PROBLEM WAS RELIEVED.

MPP APPLICATIONS - TODAY

TYPE OF WORK	INSTITUTION	STATUS
ISODATA CLUSTERING ALG	GSFC/630	WORKING
SINGLE LAYER FLUID MODEL	GSFC/630/611	WORKING
MAXIMUM LIKELIHOOD CLASSIFIER	GSFC/630/USDA	WORKING
LARGE IMAGE ROTATION & WARPING	GSFC/680/PENN ST	WORKING
CONTEXTUAL CLASSIFIER	GSFC/630	WORKING
CONNECTED COMPONENTS LABELING	GSFC/630	WORKING
TEXTURAL FEATURE EXTRACTION	GSFC/630	WORKING
SIR-B SAR SIGNAL PROCESSING	GSFC/630	ADV DEBUG
NUMBER FACTORING	DOD/GOODYEAR	ADV DEBUG
COMPUTATIONAL ENGINE RESEARCH	GSFC/630	ADV DEBUG

MPP APPLICATIONS - TODAY (CONTINUED)

TYPE OF WORK	INSTITUTION	STATUS
ASSOCIATIVE QUERY PROCESSING	GSFC/520	DEBUG
CONVOLUTION (IMAGE FILTERING)	GSFC/CORNELL	DEBUG
MEDIAN FILTERING OF IMAGES	GSFC/CORNELL	DEBUG
TOPOGRAPHIC DATA FROM SIR-B STEREO PAIRS	GSFC/630	DEBUG
TWO LAYER FLUID MODEL	GSFC/611	DEBUG
CLASSY CLUSTERING ALG	GSFC/630	DEBUG
THEMATIC MAPPER GEOMETRIC CORRECTION	GSFC/GOODYEAR	DEBUG
IMAGE DEBLURRING	GSFC/681	DESIGN
HILLSLOPE HYDROLOGICAL MODEL	GSFC/620	DESIGN

TIMING SUMMARY

	VAX	VAX WITH AP180V ARRAY PROCESSOR	MPP
ISODATA CLUSTERING		3 HRS	20 SEC
MAXIMUM LIKELIHOOD CLASSIFICATION	15 MIN		0.5 SEC
'CLASSY' CLUSTERING (128 X 128 ESTIMATE)		2 - 3 HRS	60 SEC
CONTEXTUAL CLASSIFIER		2 - 3 HRS	18 SEC
SYNTHETIC APERTURE RADAR 'SEASAT' IMAGE GENERATION		2 - 3 HRS	3 - 5 MIN

MPP
A NATIONAL
RESOURCE

MPP GOAL

ADVANCE STATE-OF-THE-ART
IN CONCURRENT PROCESSING FOR

- IMAGE ANALYSIS & INFORMATION EXTRACTION
- SIGNAL PROCESSING & KALMAN FILTERING
- ATMOSPHERIC & OCEANOGRAPHIC MODELING
- BASIC PHYSICAL, MATHEMATICAL, & COMPUTER SCIENCES RESEARCH

GSFC

A NATIONAL RESOURCE

MISSION OBJECTIVES FOR THE MPP

- —DEMONSTRATE THE MPP'S UNIQUE SCIENTIFIC APPLICATIONS CAPABILITIES
- —FACILITATE RESEARCH PROJECTS THAT ARE REASONABLE ONLY BY USING MPP
- —EVALUATE THE MPP SYSTEM HARDWARE & SOFTWARE FOR GENERAL USER AVAILABILITY
- —RECOMMEND FUTURE ENHANCEMENTS (SOFTWARE & HARDWARE) NEEDED FOR GENERAL SCIENTIFIC USE

GSFC

BACKGROUND OF THE APPLICATIONS NOTICE

- ANNOUNCED AN ONGOING OPPORTUNITY TO CARRY OUT COMPUTATIONAL INVESTIGATONS EXPLOITING THE UNIQUE CHARACTERISTICS OF THE MPP
- SIGNED DECEMBER 20, 1984, BY DR. EDELSON
- MORE THAN 2000 DISTRIBUTED NATIONALLY

OVERVIEW OF PROPOSALS RECEIVED

- 4Ø PROPOSALS RECEIVED

- PRINCIPLE CATEGORIES:
  - 7 - SIGNAL/IMAGE PROCESSING
  - 8 - EARTH SCIENCES
  - 1Ø - PHYSICS
  - 15 - COMPUTER SCIENCE

SIGNAL/IMAGE PROCESSING

SYNTHETIC APERTURE RADAR PROCESSING IMPROVEMENTS	MANGO NRL
RECONSTRUCTION OF CODED-APERTURE X-RAY IMAGES	YIN GSFC/682
COMET HALLEY LARGE-SCALE IMAGE ANALYSIS	KLINGLESMITH GSFC/684
AUTO DETECT AND CLASSIFY GALAXIES ON DEEP-SKY PICTURES	HEAP GSFC/681
FIXED POINT OPTIMAL NONLINEAR PHASE DEMODULATION	BUCY USC
USE SPATIAL INFO FOR ACCURATE INFORMATION EXTRACTION	TILTON GSFC/636

EARTH SCIENCES

KALMAN FILTERING AND BOOLEAN DELAY EQUATIONS	GHIL UCLA
COMPARE W/OTHER SUPERCOMPUTERS FOR LANDSAT DATA PROC	OZGA USDA
TROPOSPHERIC TRACE GAS MODELING	CARMICHAEL IOWA
NUMERICAL MODELING WIND-DRIVEN INDIAN OCEAN CIRC.	O'BRIEN FLORIDA STATE
MAGNETOSPHERIC INTERACTIVE MODEL USING CURRENT SHEETS	WHIPPLE UCSD
AUTO TECHNIQUES TO DETECT GEOLOGICAL FRACTURE PATTERNS	RAMAPRIYAN GSFC/636
NEAR-REAL-TIME PROCESSING OF GLOBAL POSITIONING SATELLITE DATA FOR PRECISION ORBIT DETERMINATION	MADRID JPL

PHYSICS

PARTICLE SIMULATION OF PLASMAS	STOREY STANFORD
PROBLEMS IN CONDENSED MATTER PHYSICS AND CHEMISTRY	SULLIVAN NBS
SIMULATIONS OF BEAM PLASMA INTERACTIONS	LIN SW RESEARCH INST
DYNAMICS OF COLLISIONLESS STELLAR SYSTEMS	WHITE SPACE TELESCOPE INST
WAVE SCATTERING BY ARBITRARILY SHAPED TARGETS	TOBOCMAN CASE WESTERN RES U
ADAPTING A NAVIER-STOKES CODE	GROSCH ICASE
FREE-ELECTRON LASER DESIGN STUDIES	VON LAVEN KMS FUSION
NUMERICAL CALCULATIONS OF CHARGED PARTICLE TRANSPORT	EARL MARYLAND

COMPUTER SCIENCE

GRAPHICS APPLICATIONS	DAVIS NCSU
SOLUTION OF COMPLEX, LINEAR SYSTEMS OF EQUATIONS	IDA U AKRON
SIMULATE APPLICATIVE PROGRAMMING STORAGE ARCHITECTURE	O'DONNELL INDIANA
SORTING AND SIGNAL PROCESSING ALGORITHMS	DEMUTH U TULSA
STOCHASTIC AND REACTION-DIFFUSION CELLULAR AUTOMATA	HASTINGS HOFSTRA
FORTH, AN INTERACTIVE LANGUAGE FOR CONTROLLING THE MPP	KLINGLESMITH GSFC/684
DIAGRAMMATIC INFORMATION PROCESSING IN NEURAL ARRAYS	BARDEN INDIANA
SPACE PLASMA GRAPHICS ANIMATION	GREENSTADT TRW
GENERATE TOPOGRAPHIC MAPS FROM SPACECRAFT IMAGERY	STRONG GSFC/636
ANIMATED MODELS OF SPACE & EARTH SCIENCES DATA	TREINISH GSFC/634

IMPACTS OF THE MPP ON PROBLEM SOLVING

- DRAMATICALLY IMPROVE MACHINE RESPONSE TIME
- MAKE MANY MORE COMPUTATIONALLY INTENSIVE PROBLEMS REASONABLE TO PERFORM
- REDUCE FUTURE SYSTEM SIZE AND COST

N87-29135

2.2

Supercomputing on Massively Parallel Bit-Serial Architectures

Consider the idea that supercomputing is a synergy of generic algorithms, languages and architectures and that real breakthroughs in parallel computing will be achieved by considering all three together in a simulated software environment. Engineering tradeoffs could be made between performance, machine transparency, standardization and program portability before any new machines are actually built. Standardized languages could be developed for generic subclasses of parallel machines; languages that really give high peformance and encourage free parallel expression and "thinking in parallel".

My own research on the Goodyear MPP (Massively Parallel Processor), suggests that high-level parallel languages are practical and can be designed with powerful new semantics that allow algorithms to be efficiently mapped to the real machines. For the MPP these semantics include parallel/associative array selection for both dense and sparse matrices, variable precision arithmetic to trade accuracy for speed, micro-pipelined "train" broadcast, and conditional branching at the PE control unit level.

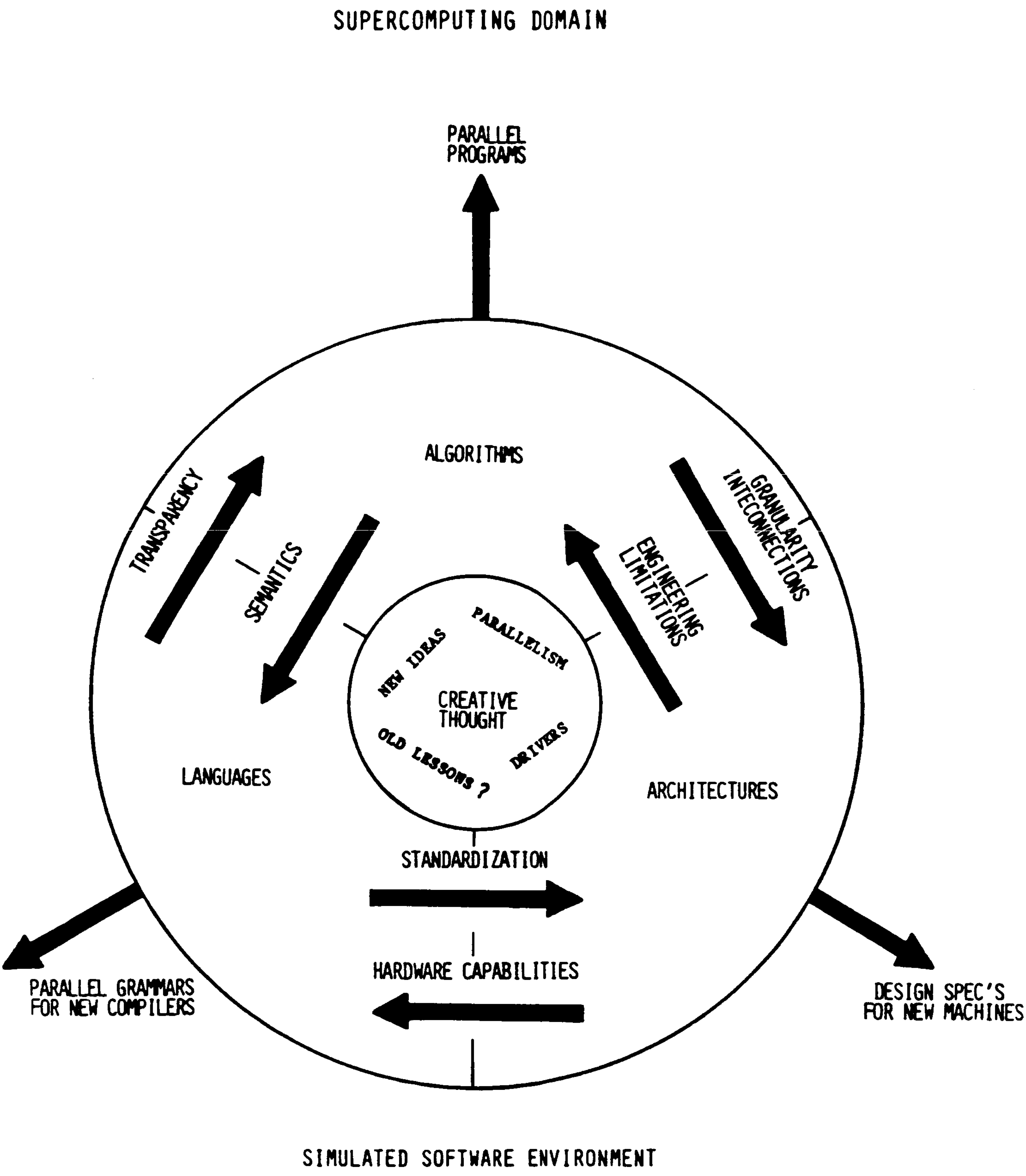
The preliminary design of a FORTRAN-like parallel language for the MPP has been completed and is being used to write programs to perform sparse matrix array selection, min/max search, matrix multiplication, Gaussian elimination on single bit arrays and other generic algorithms. The MPP timing estimate for Gaussian elimination of a 4K by 4K single bit matrix is under one second -- the equivalent of approximately 64 billion scalar operations. Parallel Gauss-Jordan matrix inversion is also being investigated. The estimated time to invert a 128 X 128, 32 bit real matrix using full pivoting on the MPP is 50 msec. This is roughly equivalent to a 100 MFLOP scalar rate.

The MPP is a SIMD machine of 16384 single bit processors arranged in a 128 X 128 array. Individual PE's are interconnected with their four nearest neighbors. Each PE can address 1024 bits of its own local memory. A 32 bit shift register in each PE allows for micro-pipelining of long words and faster partial sum accumulation for multiplication. The machine can execute 160 billion micro-instructions per second which translates to 800 GOPS for some instructions. Operations include single bit logical, shift, and add as well as column I/O and one or two dimensional routing in a spiral, cyclinder, or torus. All operations can be directly or indirectly masked. The logical "or" of one bit per PE (SUMOR) can be used to pass array information back to the PE control unit for broadcast to other PE's, scalar I/O or conditional branching. If a second MPP were ever built, it might look considerably different than the current MPP. For example, it would certainly have greater memory depth -- at least 64K bits per PE. It might also have a reconfigurable bit/byte serial ALU, staged PE's for table lookup arithmetic, and pipelined SUMOR logic.

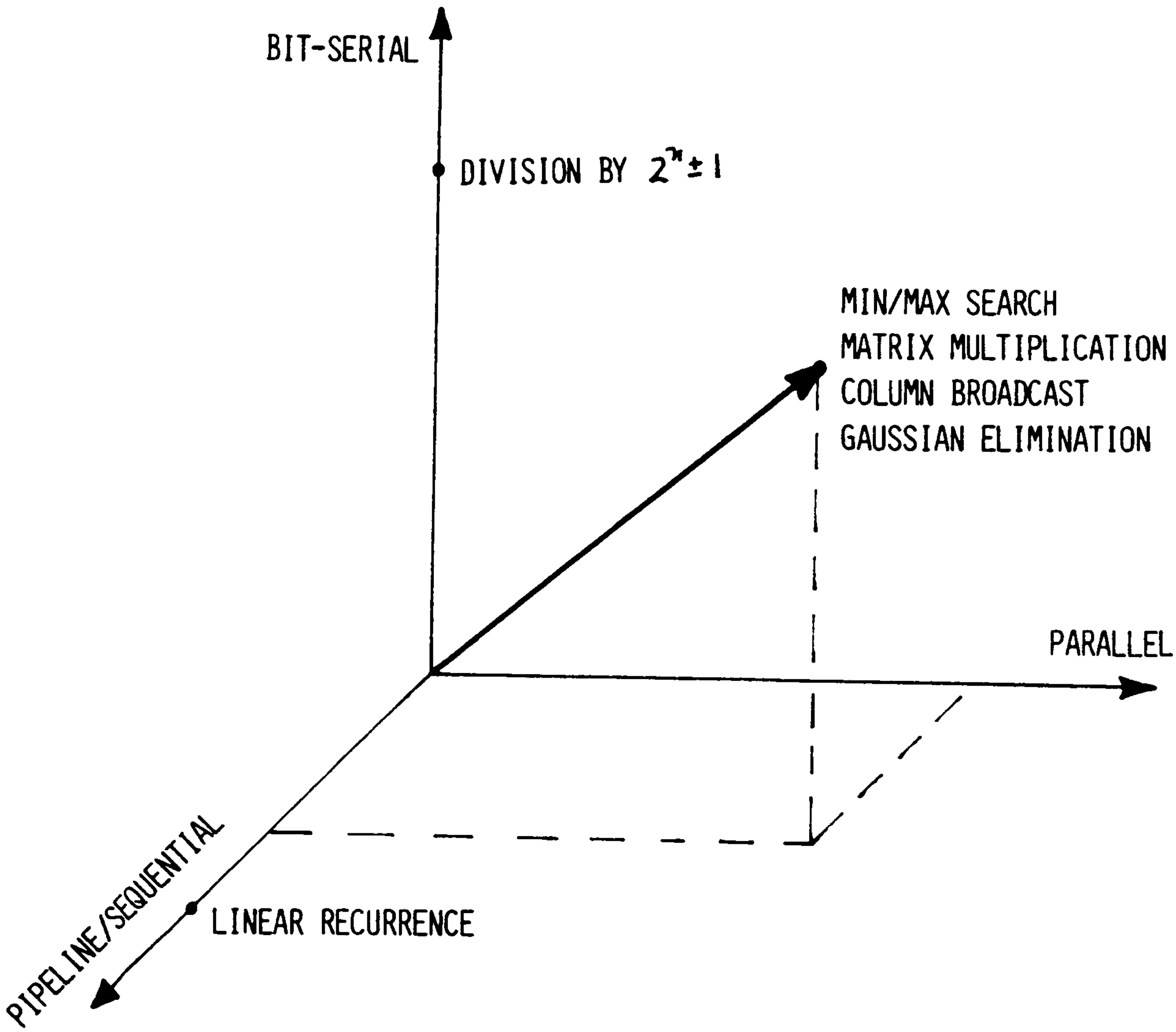
Ken Iobst
4/15/85

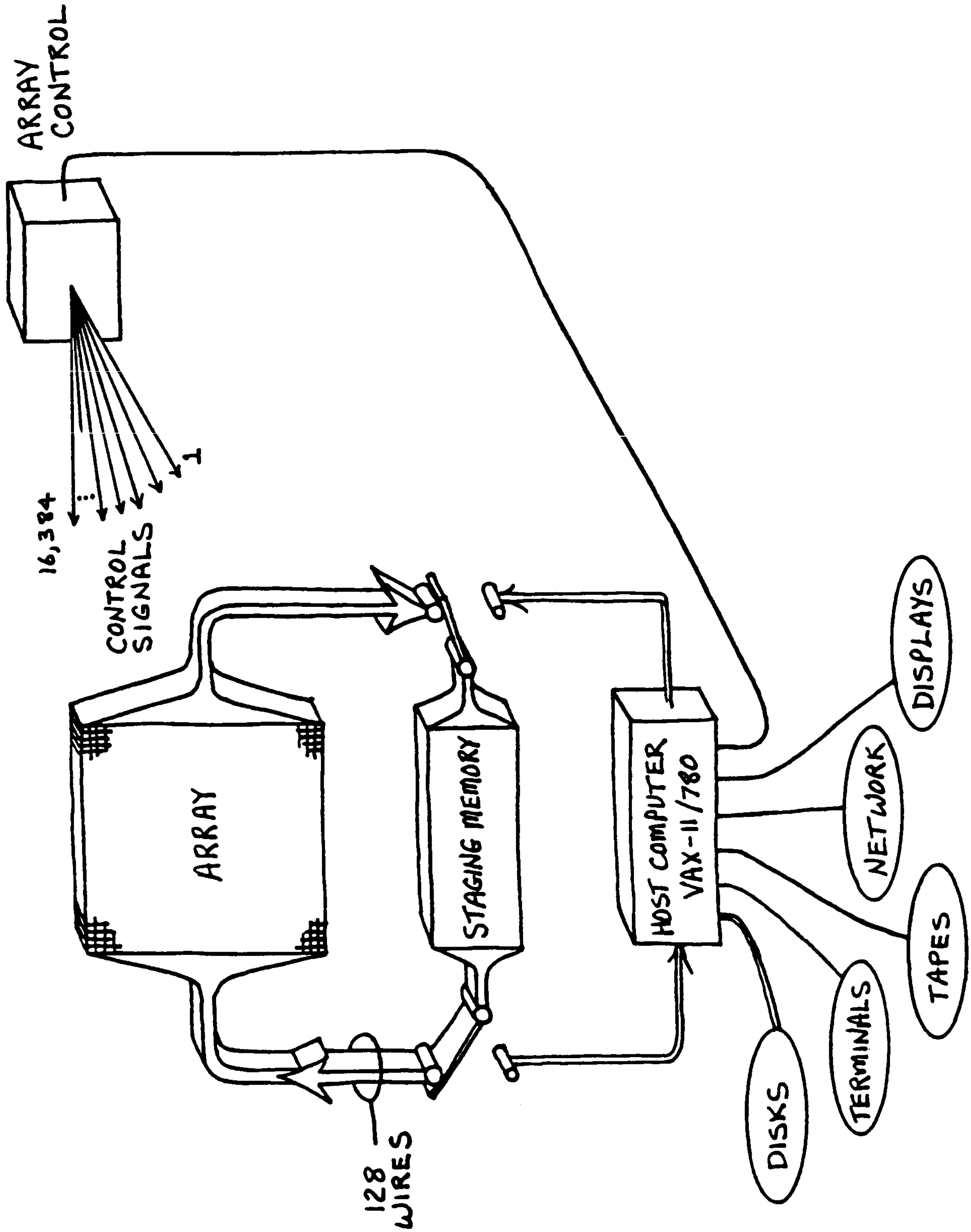
SUPERCOMPUTING ON MASSIVELY PARALLEL BIT-SERIAL ARCHITECTURES

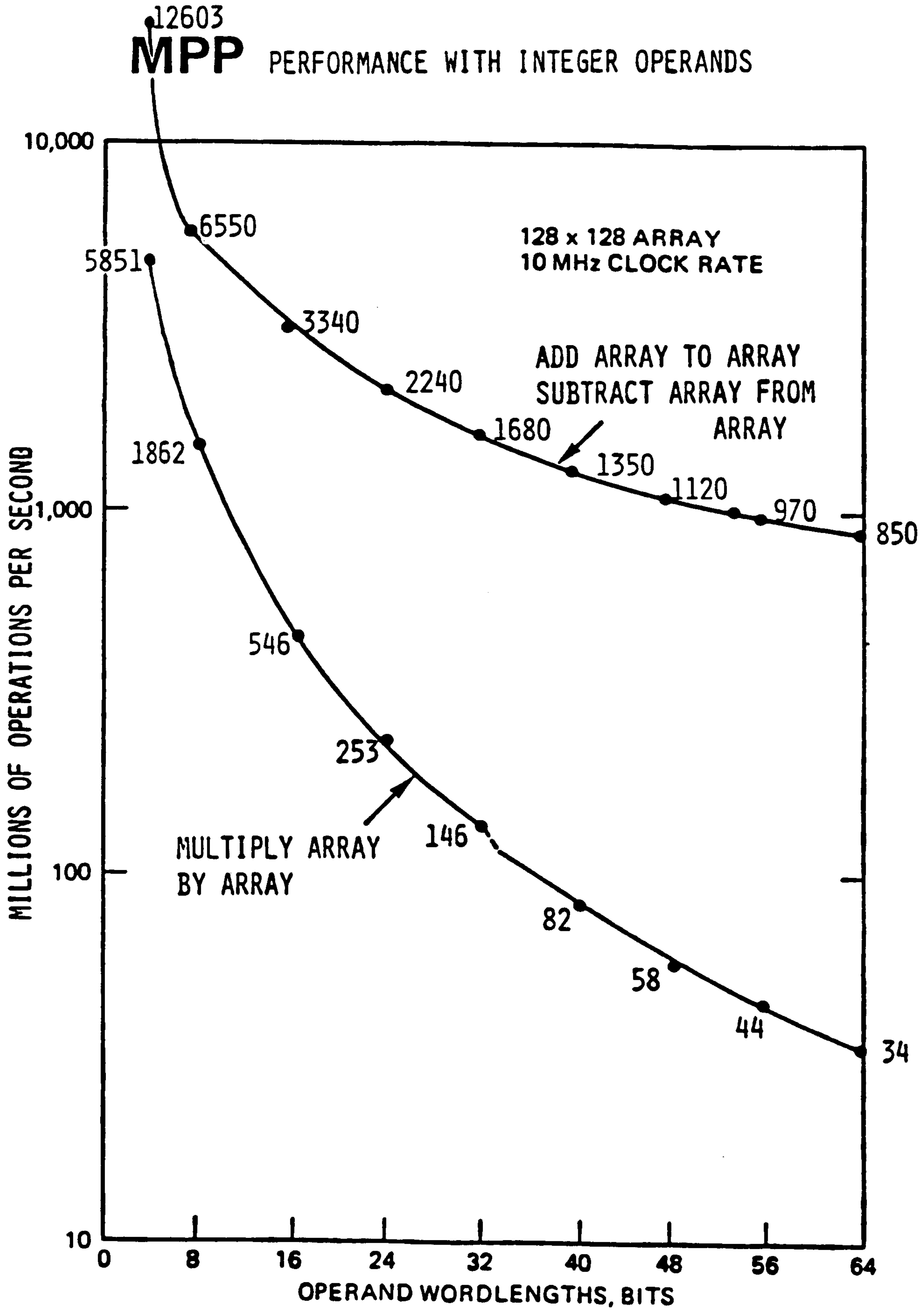
- SUPERCOMPUTING DOMAIN
- NEW DIMENSIONS IN PARALLEL COMPUTING
- SOME GENERIC ALGORITHMS
- THE GOODYEAR MPP
- SOME MPP SPECIFIC ALGORITHMS CODED IN A FORTRAN-LIKE BIT-SERIAL PROGRAMMING LANGUAGE
- WHAT MIGHT A SECOND GENERATION MPP LOOK LIKE?


SUPERCOMPUTING DOMAIN
PARALLEL PROGRAMS
ALGORITHMS
TRANSPARENCY
SEMANTICS
GRANULARITY
INTECONNECTIONS
ENGINEERING LIMITATIONS
NEW IDEAS
PARALLELISM
CREATIVE THOUGHT
OLD LESSONS ?
DRIVERS
LANGUAGES
ARCHITECTURES
STANDARDIZATION
HARDWARE CAPABILITIES
PARALLEL GRAMMARS FOR NEW COMPILERS
DESIGN SPEC'S FOR NEW MACHINES
SIMULATED SOFTWARE ENVIRONMENT

NEW DIMENSIONS IN PARALLEL COMPUTING




ARRAY CONTROL
16,384
1
CONTROL SIGNALS
ARRAY
STAGING MEMORY
HOST COMPUTER
VAX-11/780
128 WIRES
DISKS
TERMINALS
TAPES
NETWORK
DISPLAYS



12603
MPP PERFORMANCE WITH INTEGER OPERANDS
10,000
1,000
100
10
MILLIONS OF OPERATIONS PER SECOND
128 x 128 ARRAY
10 MHz CLOCK RATE
6550
3340
2240
1680
1350
1120
970
850
ADD ARRAY TO ARRAY
SUBTRACT ARRAY FROM ARRAY
5851
1862
546
253
146
82
58
44
34
MULTIPLY ARRAY BY ARRAY
0
8
16
24
32
40
48
56
64
OPERAND WORDLENGTHS, BITS

DIVISION BY $2^n \pm 1$ EXAMPLE

FROM THE BINOMIAL THEOREM,

$$\frac{1}{1 \pm x} = 1 \mp x + x^2 \mp x^3 + \cdots \quad (x^2 < 1)$$

BY A CHANGE OF VARIABLE $y = \frac{1}{x}$ THEN

$$\frac{1}{y \pm 1} = \frac{1}{y} \mp \frac{1}{y^2} + \frac{1}{y^3} \mp \cdots \quad (y^2 > 1)$$

NOW LET $y = 2^n$ AND DIVISION BY $2^n \pm 1$
REDUCES TO A SHORT SEQUENCE OF BINARY
SHIFTS AND ADDS (AND/OR SUBTRACTS),

$$\frac{v}{2^n \pm 1} = \frac{v}{2^n} \mp \frac{v}{2^{2n}} + \frac{v}{2^{3n}} \mp \cdots$$

FOR EXAMPLE, LET V = 237658 AND N = 10
THEN

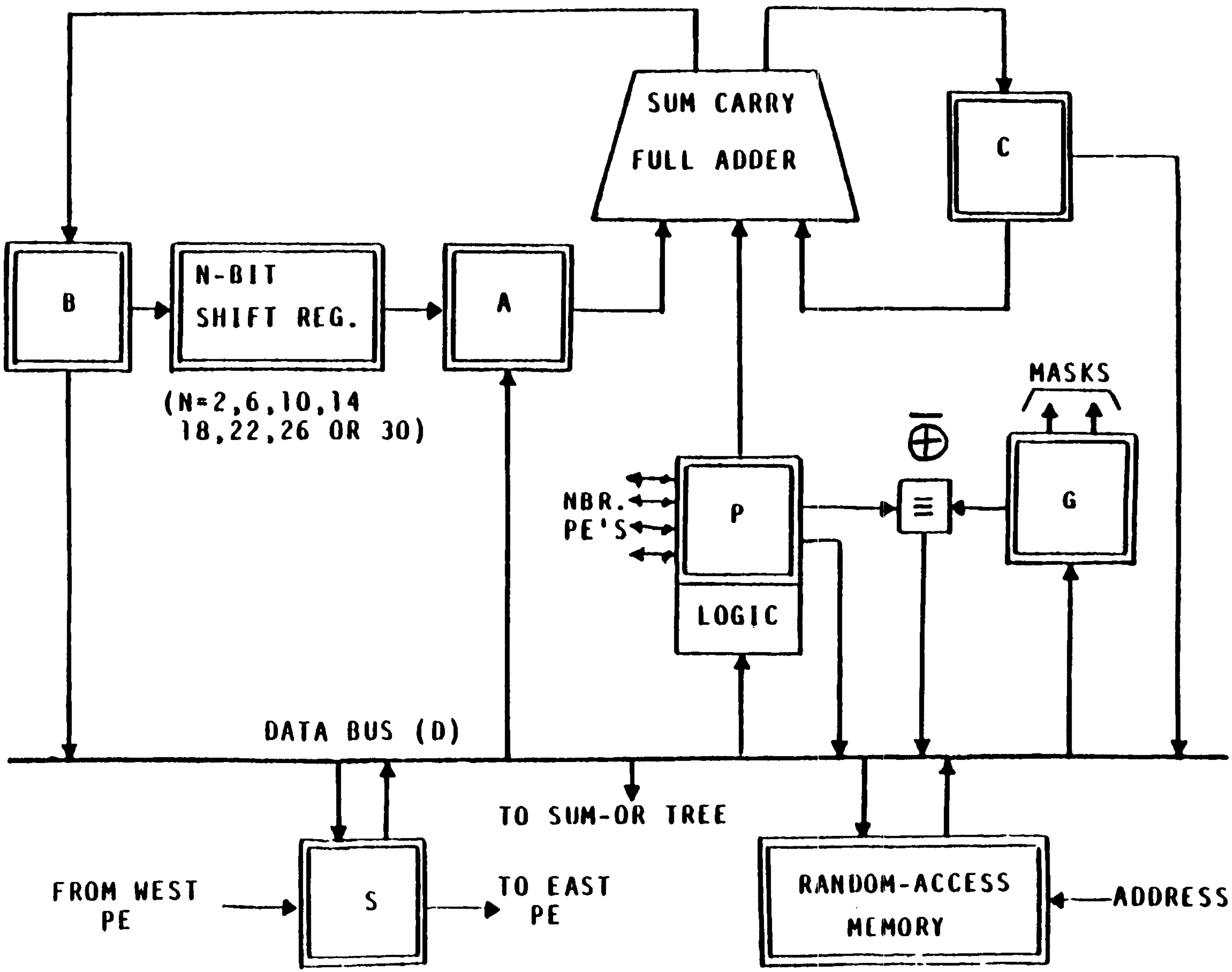
$$\frac{v}{2^n - 1} = \frac{237658}{1023} = 232.315$$

AFTER 3 SHIFTS AND 2 ADDS

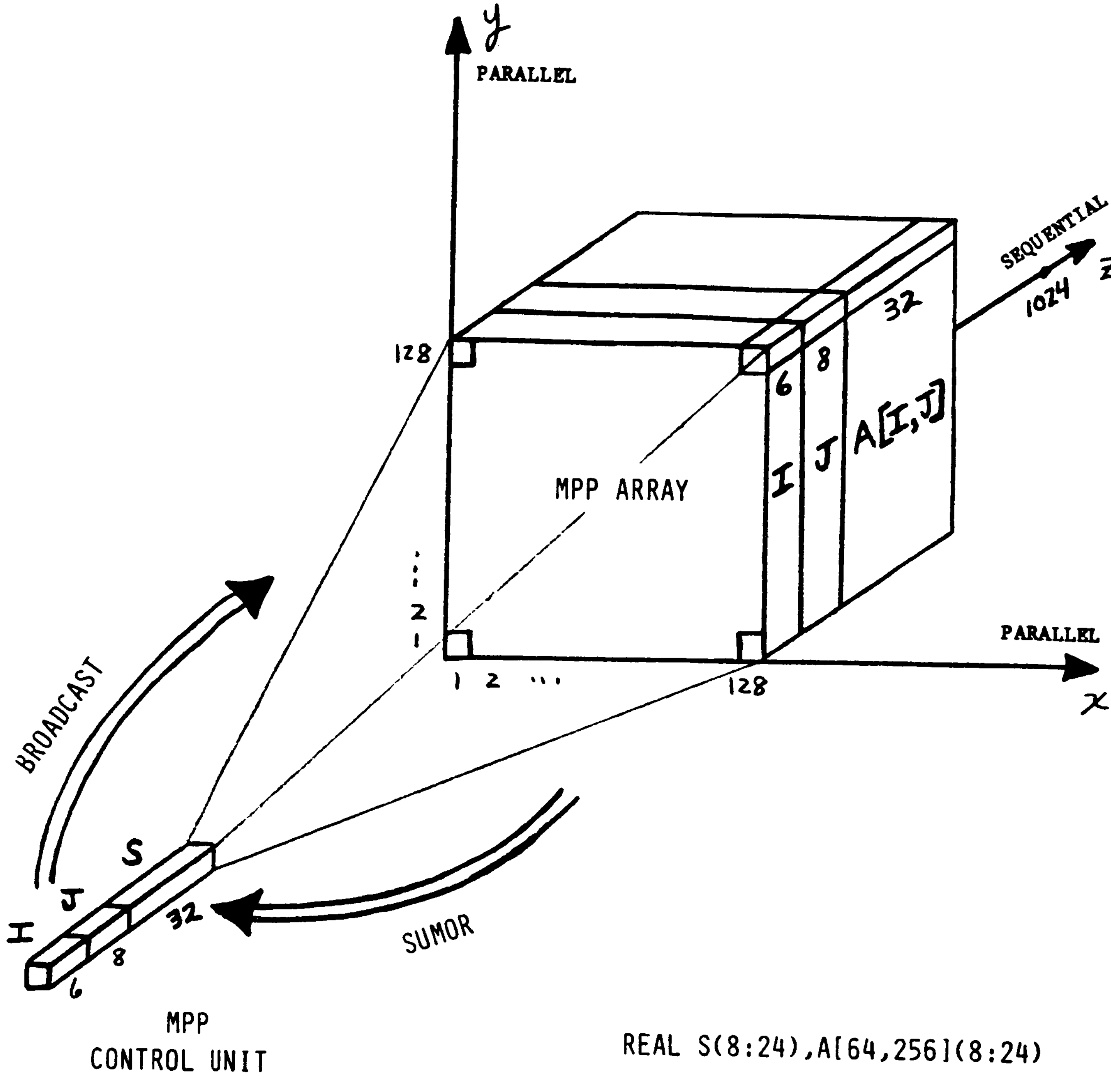
THE GOODYEAR MPP

- SIMD MACHINE OF 16384 SINGLE BIT PROCESSORS ARRANGED IN A 128 X 128 ARRAY
- NEAREST NEIGHBOR INTERCONNECTIVITY
- 1024 BITS OF MEMORY PER PE
- 32 BIT SHIFT REGISTER ALLOWS FOR MICRO-PIPELINING AND FASTER MULTIPLICATION
- EXECUTION SPEED OF 160 BILLION MICRO-INSTRUCTIONS PER SECOND WHICH TRANSLATES TO 800 GOPS FOR SOME INSTRUCTIONS
- OPERATIONS INCLUDE SINGLE BIT LOGICAL, SHIFT, AND ADD AS WELL AS COLUMN I/O AND ONE OR TWO DIMENSIONAL ROUTING IN A SPIRAL, CYLINDER, OR TORUS
- ALL OPERATIONS CAN BE DIRECTLY OR INDIRECTLY MASKED
- THE LOGICAL "OR" OF ONE BIT PER PE (SUMOR) CAN BE USED TO PASS ARRAY INFORMATION BACK TO THE PE CONTROL UNIT FOR BROADCAST, SCALAR I/O, OR CONDITIONAL BRANCHING

ONE OF 16384 MPP PROCESSING ELEMENTS (PE'S)



PARALLEL/ASSOCIATIVE ARRAY SELECTION



REAL S(8:24),A[64,256](8:24)

S=SUMOR(A[64,256])

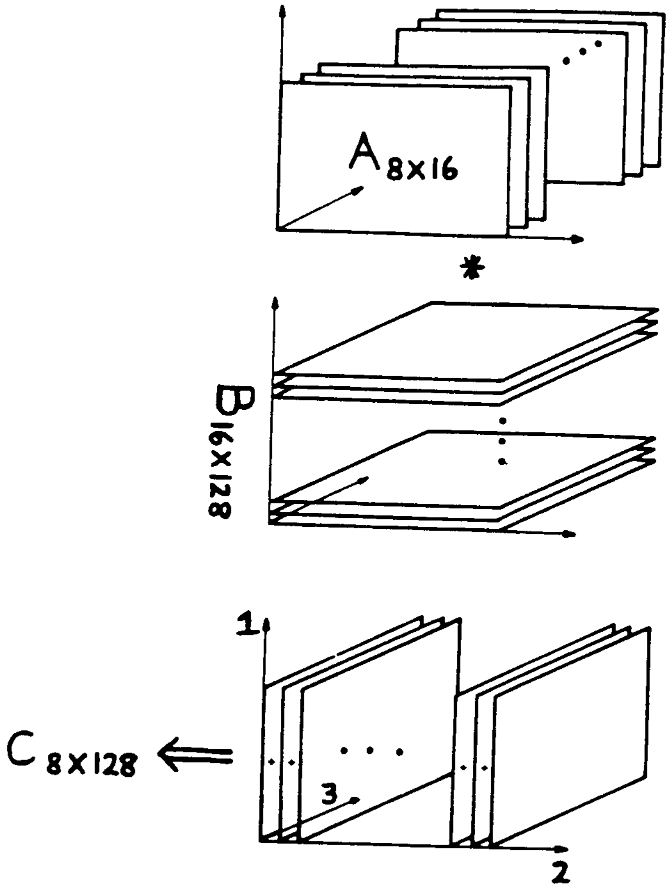
MAXIMUM OF 32 BIT INTEGER ARRAY (OF UNIQUE VALUES)

```
  BIT MAX[ ]                             ; DECLARE MAX AS BIT MASK
                                           OVER ALL PE'S
  INTEGER A[128,128](0:32)               ; DECLARE A AS A 128 X 128
                                           UNSIGNED INTEGER ARRAY
  MAX=1                                  ; INITIALIZE MAX TO 1 OVER
                                           ALL PE'S
  DO 1 I=1,32                            ; SCAN BITS IN A FROM MOST
                                           TO LEAST SIGNIFICANT BITS
  IF (SUMOR(A[MAX](I))) MAX=A[MAX](I)    ; REPLACE MAX WITH A NEW
                                           SUBSET OF MAXIMUM VALUES
                                           FOR EACH NON ZERO BIT
                                           PLANE OF A
1 CONTINUE
```

MAXIMUM OF 32 BIT INTEGER ARRAY (GENERAL CASE)

```
  BIT MAX[ ],T[ ](46),INDEX[ ](14)
  INTEGER A[128,128](0:32)
  COMMON /INIT/ INDEX                    ; SAME ALGORITHM AS BEFORE
                                           EXCEPT A ARRAY IS FIRST
                                           CONCATENATED WITH THE
                                           PE ADDRESS FIELD TO INSURE
                                           UNIQUENESS OF RESULT
  MAX=1
  T=A.CON.INDEX
  DO 1 I=1,46
  IF (SUMOR(T[MAX](I))) MAX=T[MAX](I)
1 CONTINUE
```

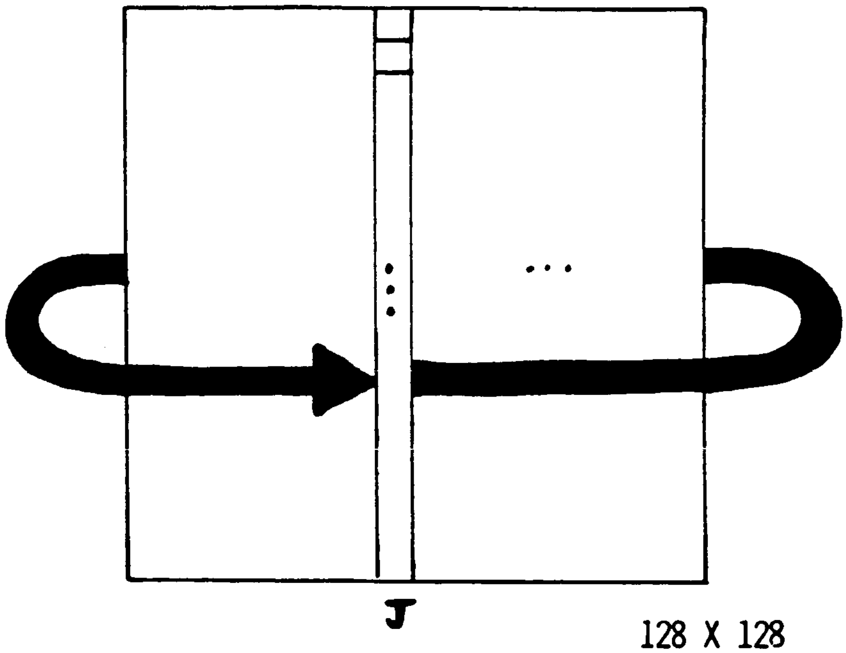
MATRIX MULTIPLICATION EXAMPLE




```
 REAL  A[8,16,128](8:32),B[8,16,128](8:32),
&      C[8,16,128](8:32),T[8,16,128](8:32)

       READ A[,,1],B[1,,]
       T=A[,,1...]*B[1...,,]
       C=T[,+,]
       PRINT C[,1,]
```

COLUMN BROADCAST EXAMPLE

$A_{ij} =$



128 X 128

$A_{ij} = A_{iJ}$

```
REAL A[128,128](8:32)
A=A[,J...]
```

OR

```
REAL A[128,128](8:32)
BIT M[ ]
M=[128,128;,J]
A=A[.NOT.M][,128→]
```

COLUMN BROADCAST EXAMPLE

PROBLEM: TO BROADCAST A COLUMN OF FLOATING POINT NUMBERS ACROSS THE MPP ARRAY

SOLUTION #1: WITH PE'S INTERCONNECTED IN AN E/W CYLINDER; LOAD, SHIFT AND STORE THE 32 BIT VALUES ACROSS THE ARRAY. THIS TAKES APPROXIMATELY 3 X 32 X 128 = 12288 CYCLES.

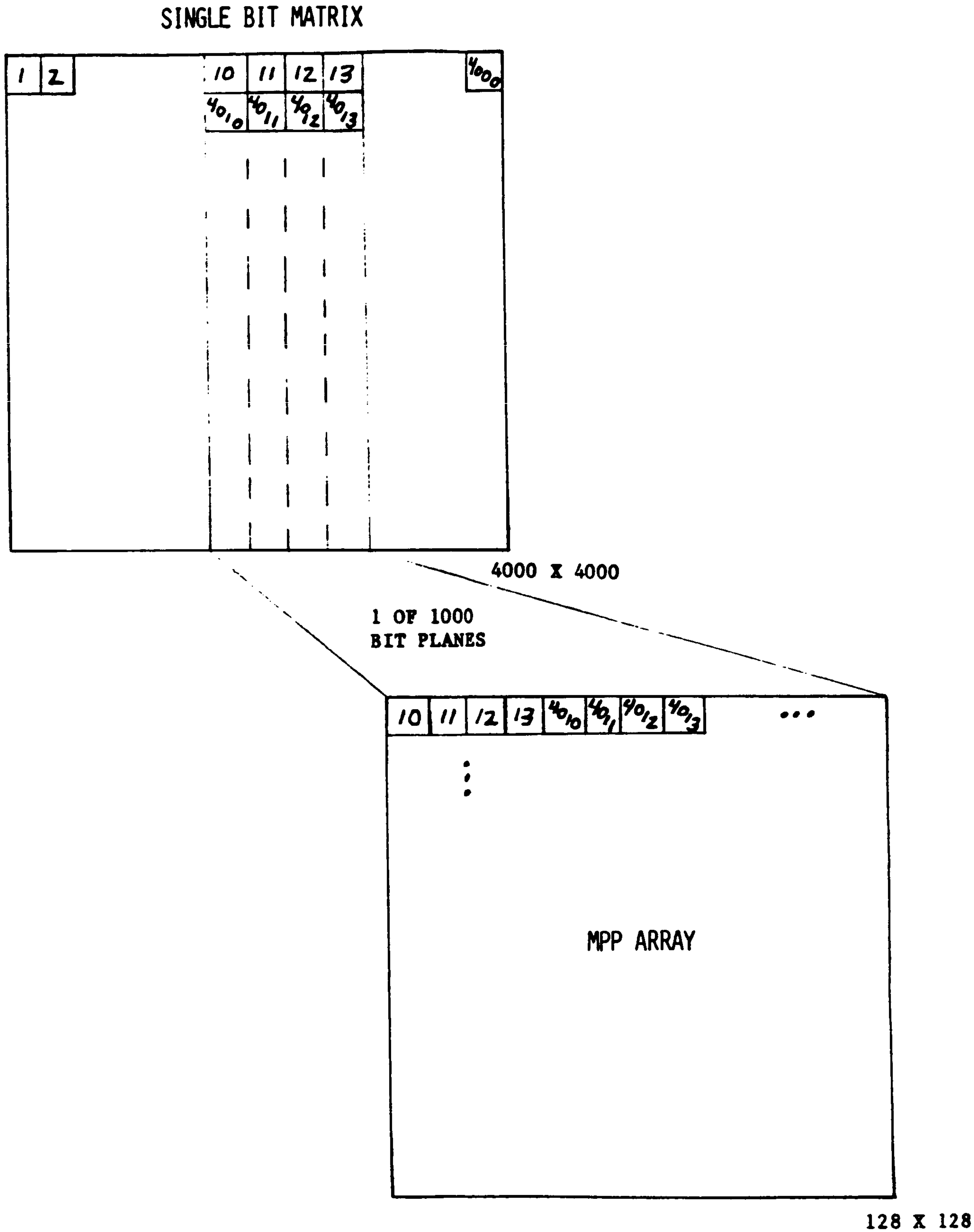
SOLUTION #2: WITH PE'S INTERCONNECTED IN AN E/W CYLINDER; "TRAIN" BROADCAST THE 32 BIT VALUES ACROSS THE ARRAY. THIS CAN BE VIEWED AS A MICRO-PIPELINING OPERATION AND TAKES ONLY 207 CYCLES. THE ALGORITHM IS AS FOLLOWS:

- GET "TRAIN" OF 1 STOP BIT + 32 BIT VALUES OUT ONTO THE E/W PE CHANNEL (≈ 33 CYCLES)

- CIRCULATE "TRAIN" ONCE AROUND (≈ 128 CYCLES). DURING THIS PROCESS INDIVIDUAL PE'S WILL STORE THE "TRAIN" IN THEIR SHIFT REGISTERS. SHIFTING STOPS WHEN THE STOP BIT ENTERS THE CONDITIONAL MASK REGISTER OF EACH PE.

- STORE ALL SHIFT REGISTERS (≈ 32 CYCLES).

GAUSSIAN ELIMINATION EXAMPLE



GAUSSIAN ELIMINATION EXAMPLE

```
      BIT A[4000,4](1000),M[4000,4],USED(4000)
      INTEGER PIVOT(4000,0:14),J1(0:2),J2(0:12),J(0:14)
      EQUIVALENCE (J1,J(1)),(J2,J(3))

      READ A                              ; READ IN ARRAY
      DO 1 I=1,4000                       ; INITIALIZE HISTORY MATRIX
      USED(I)=0
    1 CONTINUE
      DO 7 I=1,4000
      DO 2 J2=1,1000                      ; SEARCH FOR A 1 IN ROW I
                                            IN STEPS OF 4 COLUMNS
      IF (SUMOR(A[I,](J2))) GO TO 3
    2 CONTINUE
      GO TO 8                             ; ROW OF ALL 0'S - EXIT
    3 CONTINUE
      DO 4 J1=1,4
      IF (SUMOR(A[I,J1](J2))) GO TO 5     ; FIND WHICH COLUMN OF 4
    4 CONTINUE
    5 CONTINUE
      PIVOT(I)=J                          ; SAVE HISTORY INFORMATION
      USED(J)=1
      M=A[ ](J2).AND..NOT.[4000,4;I,J1]   ; SAVE PIVOT COLUMN IN NEW
                                            MATRIX M, ZEROING THE PIVOT
                                            ROW VALUE
      DO 6 J2=1,1000
      A[ ](J2)=A[ ](J2).XOR.M[,J1...]     ; ELIMINATE 4 COLUMNS AT A TIME
                                            BY BROADCASTING THE PIVOT
                                            COLUMN ACROSS THE M ARRAY
    6 CONTINUE
    7 CONTINUE
    8 CONTINUE
```

•
•
•

GAUSS-JORDAN MATRIX INVERSION

WITH FULL PIVOTING

PARALLEL DATA STRUCTURES

REAL ARRAYS

$U = [\ A : I\ ]$ AUGMENTED MATRIX

$V = [\quad : \quad]$ WORKING ARRAY

$W = [\quad : \quad]$ WORKING ARRAY

BIT MASKS

$X = [\ \bar{1} : \bar{0}\ ]$ PIVOTED ROW/COLUMNS

$Y = [\ \bar{1} : \bar{1}\ ]$ PIVOT ROW

WHERE I IS THE IDENTITY MATRIX

$\bar{1}$ IS THE UNITY MATRIX

$\bar{0}$ IS THE ZERO MATRIX

OTHER DATA STRUCTURES

SCALARS

DET = 1   PIVOT

PARALLEL APPROACH TO MATRIX INVERSION

REPEAT FOLLOWING STEPS N TIMES

- FIND NEXT PIVOT
- UPDATE DETERMINATE (OPTIONAL)
- ZERO PIVOT ROW AND COLUMN IN X
- ZERO PIVOT ROW IN Y
- NORMALIZE PIVOT ROW IN U
- BROADCAST PIVOT ROW N TIMES INTO V
- BROADCAST PIVOT COLUMN 2N TIMES INTO W
- PERFORM PARALLEL ROW OPERATIONS FOR A SINGLE PIVOT
- RESET PIVOT ROW IN Y

THEN REORDER ROWS IN U TO FORM

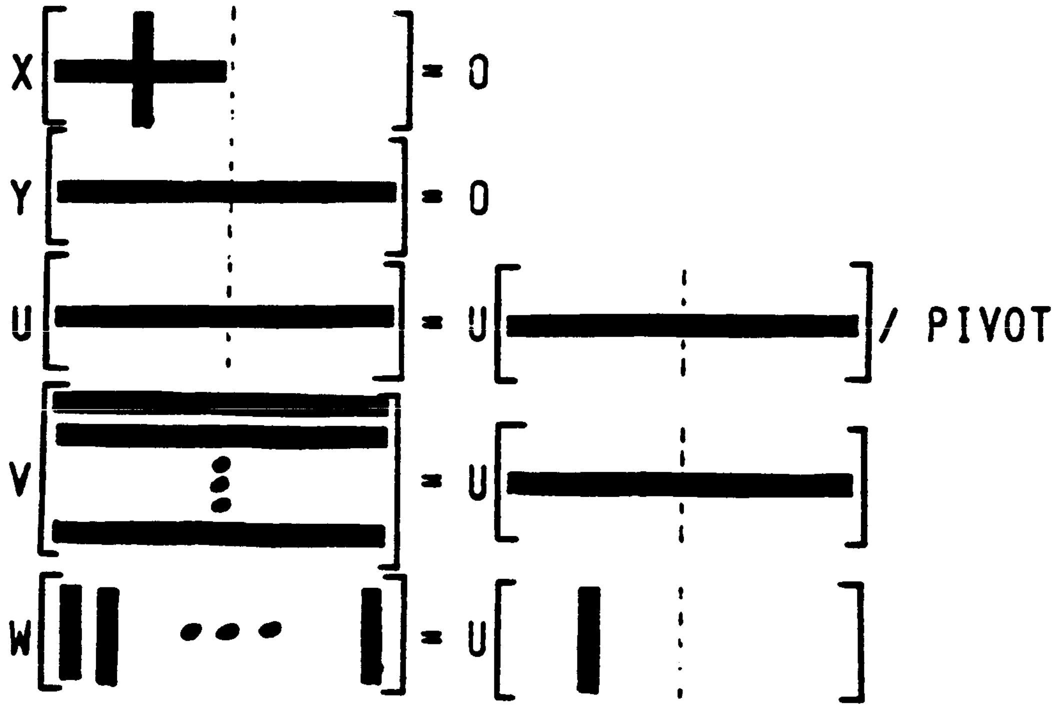
$$U = [\ I : A^{-1}\ ]$$

PARALLEL MATRIX INVERSION ALGORITHM

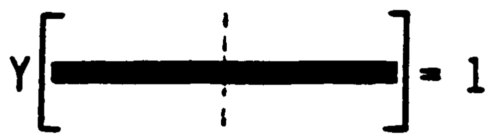
FOR I = 1 TO N

PIVOT = MAX|U| PER X

DET = DET * PIVOT



U = U - V * W PER Y



END I

FOR J = 1 TO N

FOR I = 1 TO N

IF U[I,J] = 1 THEN V[J,*] = U[I,*]

END I

END J

U = V

MPP II:
WHAT MIGHT IT LOOK LIKE?

- MUCH GREATER MEMORY DEPTH: AT LEAST 64K BITS PER PE, WITH AT LEAST ONE LEVEL OF INDIRECT ADDRESSING.

- RECONFIGURABLE BIT/NIBBLE/BYTE SERIAL ALU

- STAGED PE'S FOR TABLE LOOKUP ARITHMETIC. HOW MANY TABLES? WHAT SIZE? RAM OR ROM?

- PIPELINED SUMOR LOGIC

N87-29136

PROTOTYPE FAULT ISOLATION EXPERT SYSTEM FOR SPACECRAFT CONTROL

FY84 RTOP 506-54-66

WALT TRUSZKOWSKI

OBJECTIVE:

1. IDENTIFY SPACECRAFT OPERATIONS FUNCTIONS AT GSFC THAT COULD POTENTIALLY BE AUTOMATED USING AI TECHNIQUES.

2. DEVELOP A PROTOTYPE SYSTEM TO:

   - DEMONSTRATE AND VALIDATE EXPERT SYSTEM OPERATION FOR CONTROL CENTER SYSTEM DEVELOPERS AND OPERATORS.

   - PROVIDE EXPERIENCE FOR OPERATIONAL SYSTEM DEVELOPMENT.

PROTOTYPE FAULT ISOLATION EXPERT SYSTEM FOR SPACECRAFT CONTROL

W. TRUSZKOWSKI

ABSTRACT

Over the past year, work has been ongoing to identify areas of spacecraft command/control which could benefit from automation using artificial intelligence (AI), especially expert systems technology. A major part of this work has been the development of a demonstration expert system to help illustrate system attributes and development methodology.

An output of this activity is a program which illustrates how an expert system might perform fault handling for a satellite propulsion subsystem. The program runs on a VAX 11/780 under VMS with a VT100 interface and is written in FRANZ/LISP. The Program presents a series of menus to the user, requesting the selection of:

- The level of explanation which is presented to the user
- The displaying of assertions (fault symptoms passed to the rule base)
- The displaying of rules which "fire" during the running of the program
- The choice of Hydrazine Propulsion Subsystem (HPS) configuration (currently either ISEE or IUE configurations are selectable)
- The choice of fault type and fault location

Once the user make these selections, the program simulates the behavior of the HPS under the fault condition, detects the presence of the fault, isolates the exact nature and location of the fault, and then determines what set of actions to perform in implementing the fault workaround.

During the running of the program, a graphical depiction of the chosen HPS configuration is displayed, occupying all but the bottom five lines of the screen. The remaining portion of the screen is used to display explanations of faults and workaround actions. The display of the HPS includes the telemetry values for the sensors present in the system, which are updated during the running of the program.

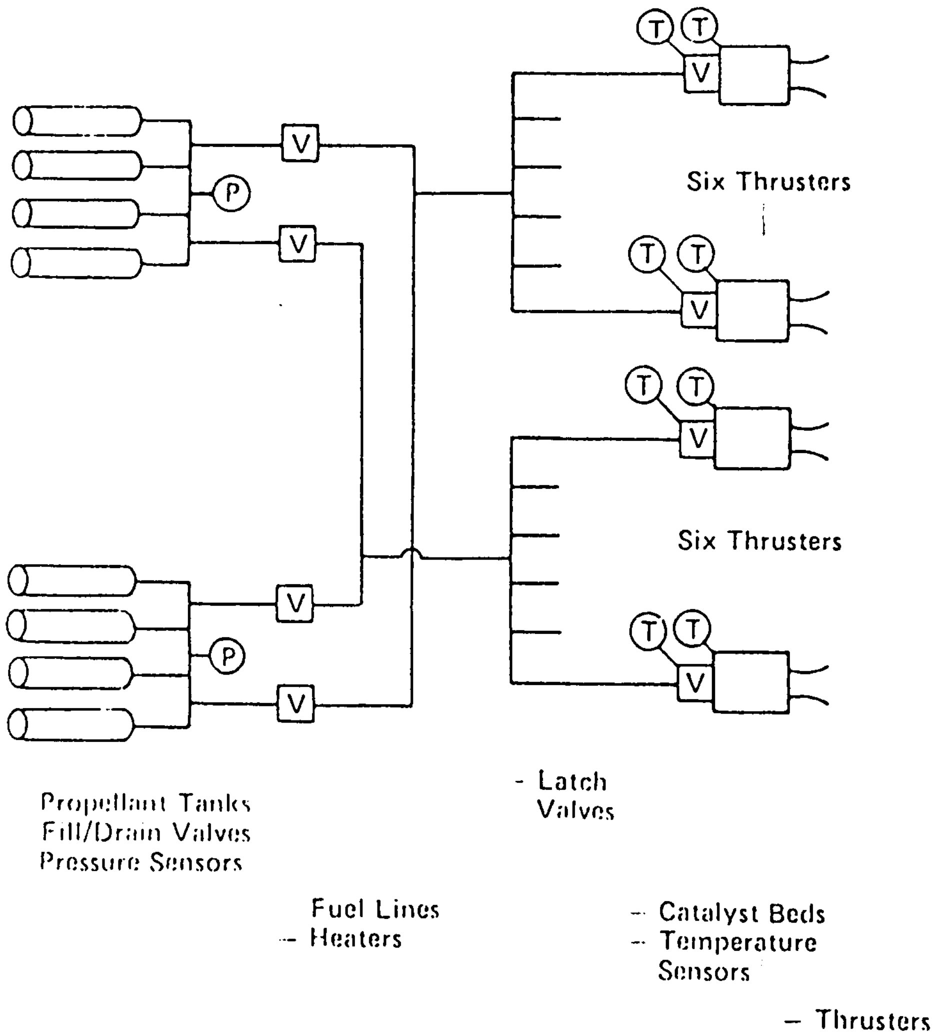
This activity has provided insight into the use of expert system technology in the spacecraft ground command/control environment. This prototype system will now be used to demonstrate and evaluate the use of expert systems in a control center environment. The propulsion system simulation and display approach should allow control center operators to experiment with and understand the operation of expert systems and gain confidence in their operational use.

PROTOTYPE FAULT ISOLATION EXPERT SYSTEM FOR SPACECRAFT CONTROL

PROTOTYPE SYSTEM

- DEVELOPED BY MARTIN-MARIETTA/DENVER
- EXPERT SYSTEM FOR FAULT ISOLATION/RECOVERY IN HYDRAZINE PROPULSION SYSTEM OPERATION
- MODELS AND GRAPHICALLY DISPLAYS OPERATION OF ISEE-1 AND IUE PROPULSION SYSTEMS
  - PHYSICAL MODEL (IDEALIZED)
  - TELEMETRY SIMULATOR
- ALLOWS USER INTRODUCTION OF MULTIPLE FAULTS
- DISPLAYS RULES AND WORK AROUND PROCEDURE
- IMPLEMENTATION: FRANZ/LISP AND MRS

ISEE Hydrazine Propulsion System



Fault Detection
- Exposure module performs "unmasking" activity.
- Telemetry generator produces telemetry data & reports differences from expected valves.

Fault Isolation
- Fault handler acts on telemetry generator data.
- Fault handler generates hypotheses and tests by commanding the S/C.

Fault Workaround
- Isolates failed component.
- Allows limited operation.
- Maintains workaround during second fault analysis.

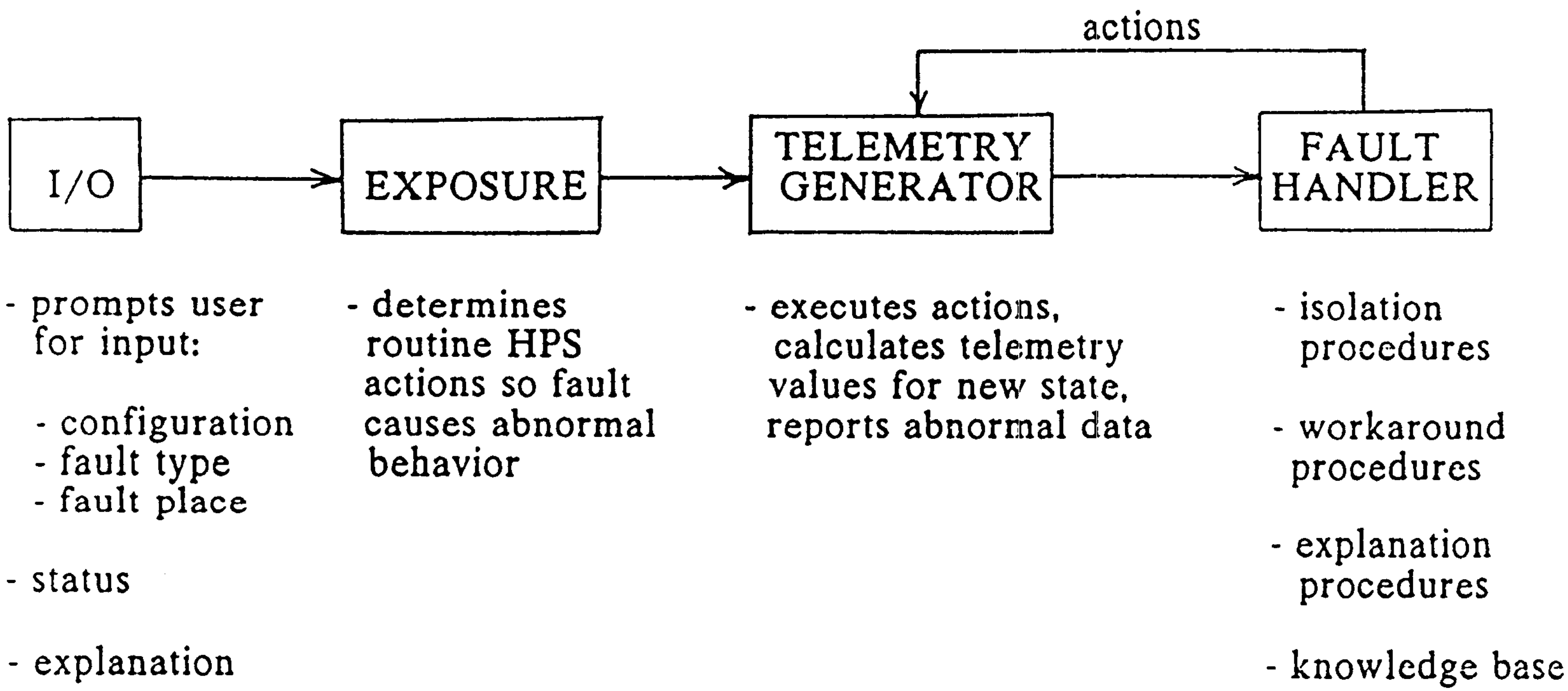
Note:

1. Fault handling module does not have an explicit model of the HPS.
2. Fault handler input comes exclusively from the telemetry generator (via the executive).

- *HPS Demonstration System provides high degree of visibility into Expert System activity:*
  - *symptom generated assertions*
  - *binding variables to LHS of rules*
  - *RHS procedures*
  - *rule generated assertions*
  - *forward chaining process*
- *HPS Demonstration System provides insight into Expert System application potential for:*
  - *fault handling/troubleshooting*
  - *executive controller for automated center*
  - *scheduling*
  - *mission workarounds*

FAULTS TO BE CONSIDERED:

- *Leak in the hydrazine storage tank*
- *Leak in a section of the line*
- *Failure of a latch valve to open*
- *Failure of a latch valve to close*
- *Failure of an engine valve to open*
- *Failure of an engine valve to close*
- *Failure of a heater to turn off*
- *Failure of a heater to turn on*
- *Failure of a catalyst bed*
- *Sensor failures*
  (engine and latch valves [open/closed], fuel pressure sensors, fuel line and catalyst bed sensors [on/off], fuel line and catalyst bed temperature sensors)



actions
I/O
EXPOSURE
TELEMETRY GENERATOR
FAULT HANDLER
- prompts user for input:
- configuration
- fault type
- fault place
- status
- explanation
- determines routine HPS actions so fault causes abnormal behavior
- executes actions, calculates telemetry values for new state, reports abnormal data
- isolation procedures
- workaround procedures
- explanation procedures
- knowledge base

CONCEPTUAL FLOW

- *USER SPECIFIES TYPE AND LOCATION OF FAULT*
- *SYSTEM MODIFIES INTERNAL MODEL ACCORDINGLY*
- *TELEMETRY GENERATION COMPONENT PRODUCES APPROPRIATE TELEMETRY DATA*
- *FAULT HANDLING COMPONENT EXAMINES TELEMETRY DATA AND USES ITS RULE BASE AND KNOWLEDGE OF NOMINAL STRUCTURE AND FUNCTION TO LOCATE FAULT (MAY REQUIRE "COMMANDING" THE MODEL)*
- *FAULT HANDLING MODULE USES RULE BASE TO IDENTIFY WORKAROUND*
- *MODEL IS MODIFIED AND TELEMETRY NORMALIZES*

KNOWLEDGE BASE ARCHITECTURE

- STRUCTURE OF TYPICAL RULES

```
(if (and ( <find-assertion> )
              ...
         ( <find-assertion> )
         (unknown ( <find-assertion> ))
              ...
         (unknown ( <find-assertion> )))
    (runnable (assert-and (and ( <lisp-procedure> )
                                      ...
                               ( <lisp-procedure> )
                               ( <make-assertion> )
                                      ...
                               ( <make-assertion> )))))
```

- KNOWLEDGE BASE IS PARTITIONED
    - GLOBAL AND SPECIFIED "THEORIES"

RULE FORMAT:

Telemetry Generator produces a list of values that are fault symptoms in a general format:

```
(|component| |number| |qualitative value| |telemetry| |expected|)
```

as variables:

```
(|$c| |$n| |$q| |$t| |$e|)
```

which are matched to rules:

```
(IF (and (assertion is true)
         (constraint is not true))

(THEN (and (LISP procedure to print rule)
           (LISP procedure to print explanation)
           (place data/result in working memory))))
```

PROTOTYPE FAULT ISOLATION EXPERT SYSTEM FOR SPACECRAFT CONTROL

STATUS

- o PROTOTYPE INSTALLED AT GSFC
- o INITIAL DEMONSTRATIONS TO CONTROL CENTER OPERATIONS MANAGERS

NEXT STEP

- o EVALUATION BY SPACECRAFT CONTROLLERS
- o EVALUATE SYSTEM FOR EXTENSION TO OTHER SPACECRAFT AND/OR SUBSYSTEMS

N87-29137

P.22

Performance Study of a Data Flow Architecture

George Adams

Research Institute for Advanced Computer Science

RIACS

Outline

- Motivation
- Methodology
- Data Flow
- Results

PRECEDING PAGE BLANK NOT FILMED

PAGE 1-178 INTENTIONALLY BLANK

Extended Abstract

In September 1984 RIACS conducted a two-week study of the proposed MIT static data flow machine for applications of interest to NASA Ames and DARPA. NASA and RIACS scientists formed seven one- or two-person teams to study data flow concepts, the static data flow machine architecture, and the VAL language. Each team mapped its application onto the machine and coded it in VAL.

The application areas were computational fluid dynamics, computational chemistry, galactic simulation, linear systems, queueing network models, and artificial intelligence. The considerations for mapping these applications onto the machine were primarily architectural: the number of individual processing elements (PE), the size of the instruction memory in each PE, the speed of the PEs, the instruction issue rate, the size of the routing network among the PEs, and the size and speed of the array memory. The goal in mapping was to maximize the number of busy PEs and to minimize the traffic on the routing network. The target machine contained 256 PEs and was capable of an aggregate rate of 1.28 GFLOPS.

The principal findings of the study were:

1. Five of the seven applications used the full power of the target machine – they sustained rates of 1.28 GFLOPS. The galactic simulation and multigrid fluid flow teams found that a significantly smaller version of the machine (16 PEs) would suffice.
2. A number of machine design parameters including PE function unit numbers, array memory size and bandwidth, and routing network capability were found to be crucial for optimal machine performance. Thus, studies of this type can provide valuable feedback to machine architects.
3. The study participants readily acquired VAL programming skills. A very high level programming environment is essential to make the data flow machine usable by most programming scientists, however, because of the complexity of the machine architecture. For example, tools to aid debugging and mapping VAL programs onto the architecture are required.
4. We learned that application-based performance evaluation is a sound method of evaluating new computer architectures, even those that are not fully specified. During the course of the study we developed models for using computers to solve numerical problems and for evaluating new architectures. We feel these models form a fundamental basis for future evaluation studies.

RIACS

Institute of the Universities Space Research Association (USRA)

- Began operation June 1983

Purpose

- Strengthen CS at Ames Research Center
- Strengthen ties to University and Industry

Technical Program

- Independent research (core)
- Joint projects (tasks)

Central Theme –

Integration of concurrent processing and artificial intelligence into every aspect of scientific investigation

Study of MIT Static Data Flow Machine and VAL Language

- Jointly funded by NASA and DARPA
- Focussed on algorithms for several applications

Goals

- Gain experience with VAL, pipe–structured programming, machine architecture
- Estimate machine performance
- Evaluate system usability
- Assess study value and methodology

Motivation

- Progress in concurrent processing research limited by lack of understanding of interactions between
  - computational models
  - system architecture
  - classes of algorithms
- Goal is **usable systems**
- Model of computation can be an experiment focus

Nature of Study

- Six application areas studied

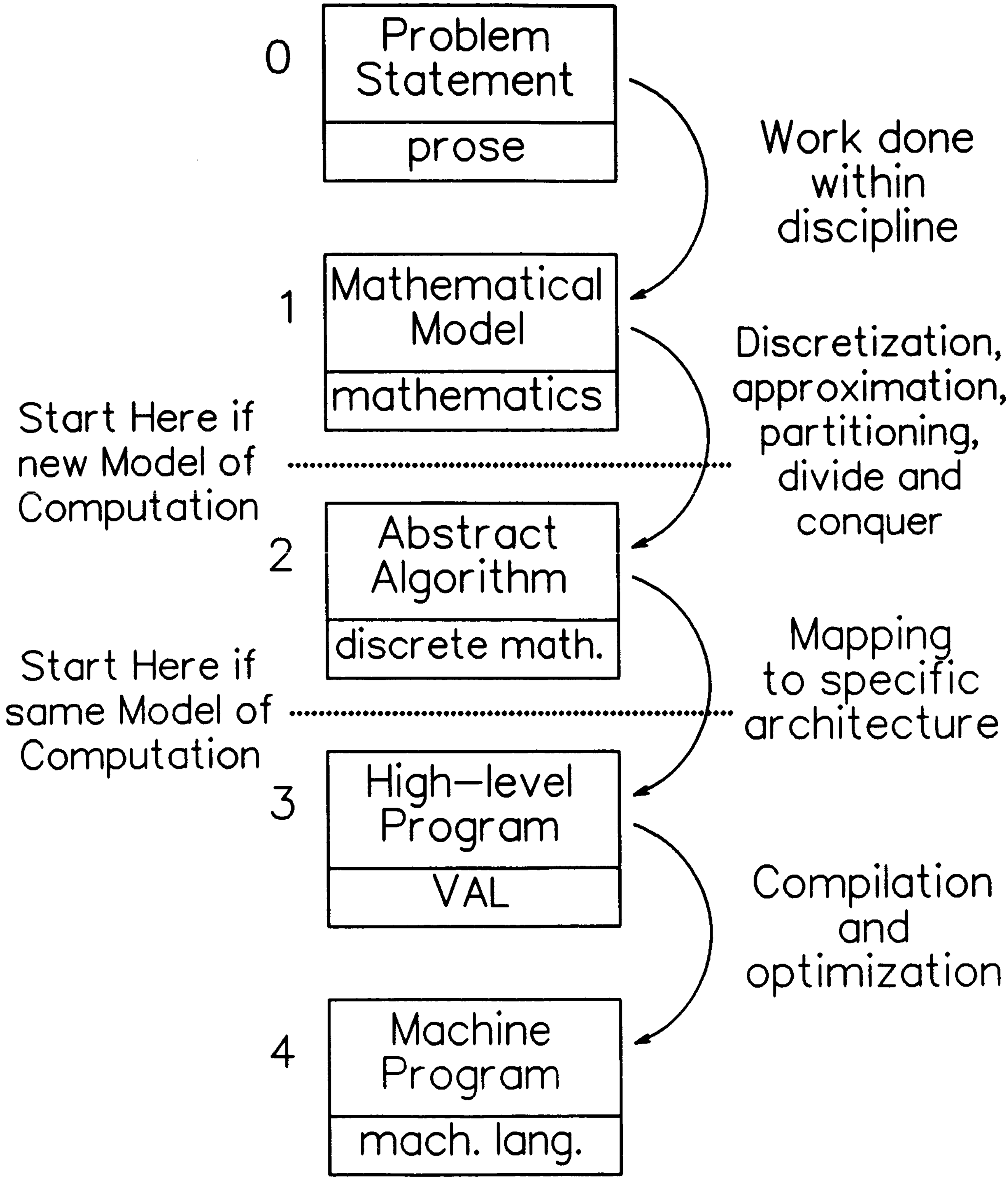
  Computational fluid dynamics
  Computational chemistry
  Galactic simulation
  Linear systems
  Natural language processing
  Queueing network models

- Eleven team members — expert in discipline, novice to data flow

- Two weeks release time from normal duties

- Study held away from daily distractions

- Familiar computer environment ported to RIACS facility for each participant; no time spent learning new system



Steps in Problem Solving

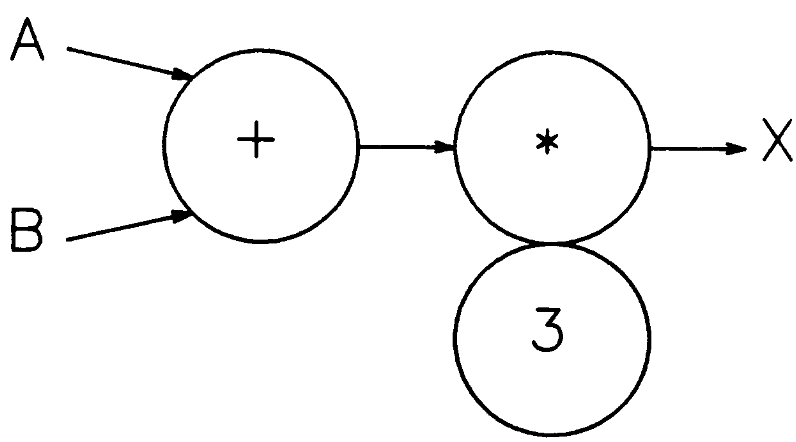
C-3

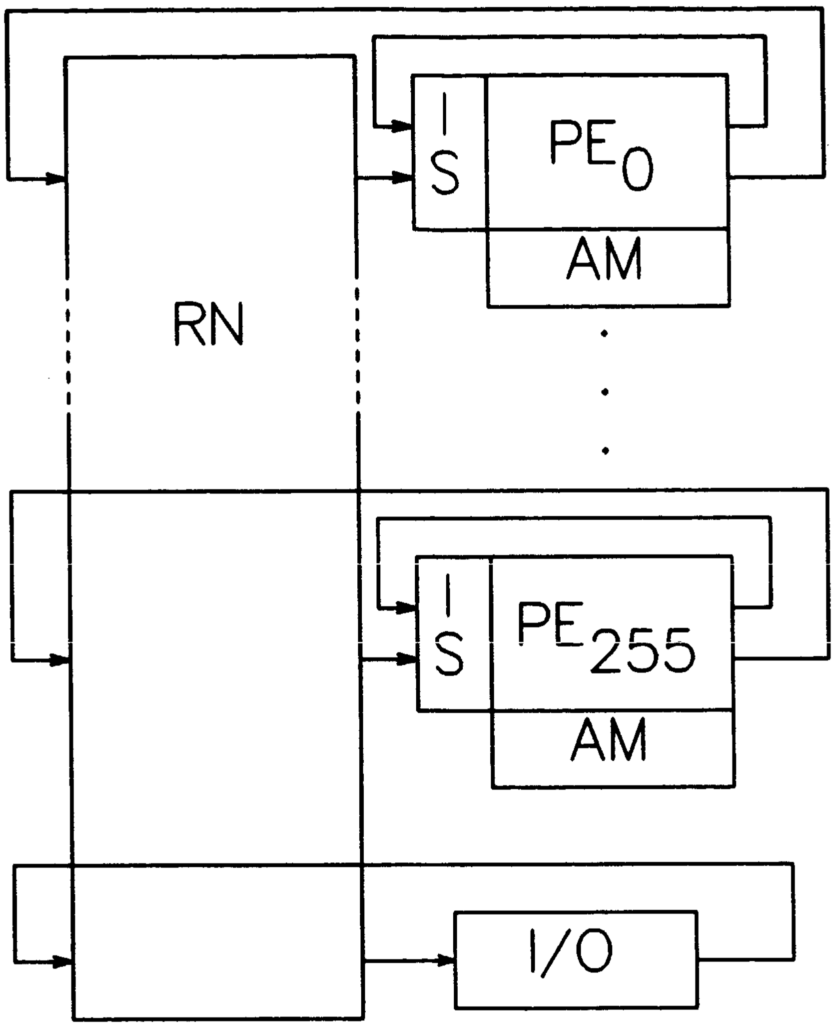
Static Data Flow Model of Computation

- Program is a directed graph
  - graph nodes represent instructions
  - operands travel on edges
- Instruction **enabled** by presence of operands and permission to send result

Example:

$$X = 3*(A+B)$$





RN — Routing Network.
PE — Processing Elements.
IS — Instruction Store.
AM — Array Memory.
I/O — Input/Output.

Static Data Flow Machine Architecture

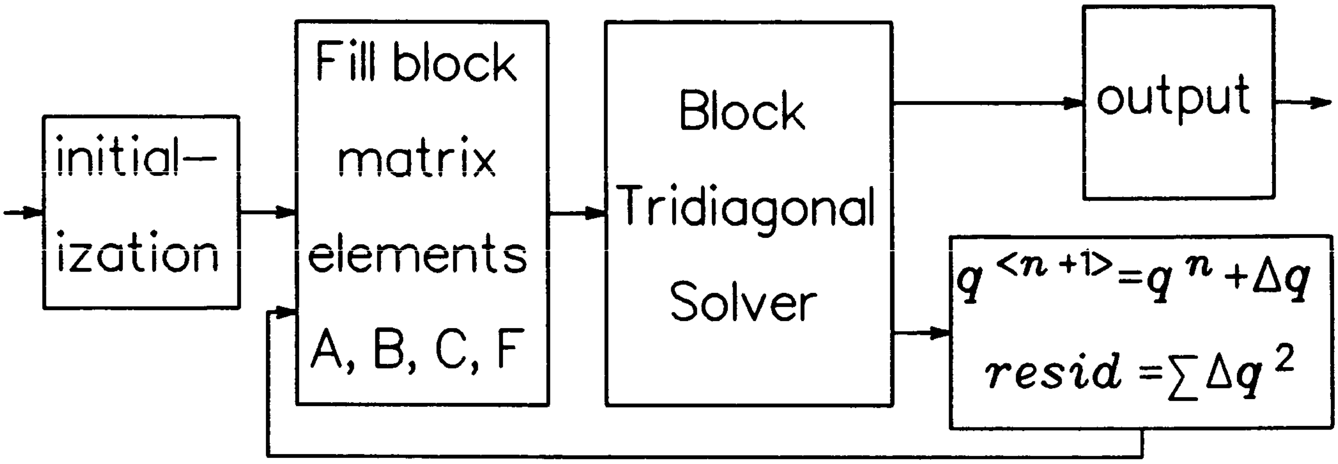
VAL – Value–Oriented Algorithmic Language

- Functional
- Single assignment
- Side–effect free
- Modular – external and internal functions
- Data treated as values not objects

```
A, B, C :=
    forall J in [1,N]
        X: real := square–root (real(J));
        construct J, X, X/2
    endall
```

Computational Fluid Dynamics

CSM



Structure of CSCM Data Flow Program

- 3–D with 10,000 grid points per 4–PE "cluster" to achieve peak speed
- PE instruction store probably too small
- Recomputation done to save memory

Computational Fluid Dynamics

Euler Method

- Method routinely used by industry
- Describes inviscid, compressible fluid flow

Findings

- Handle boundary conditions better than vector processors
- Limited parallelism; machine with 16 PEs may be good choice
- Tiny scalar speed of single PE may be problem

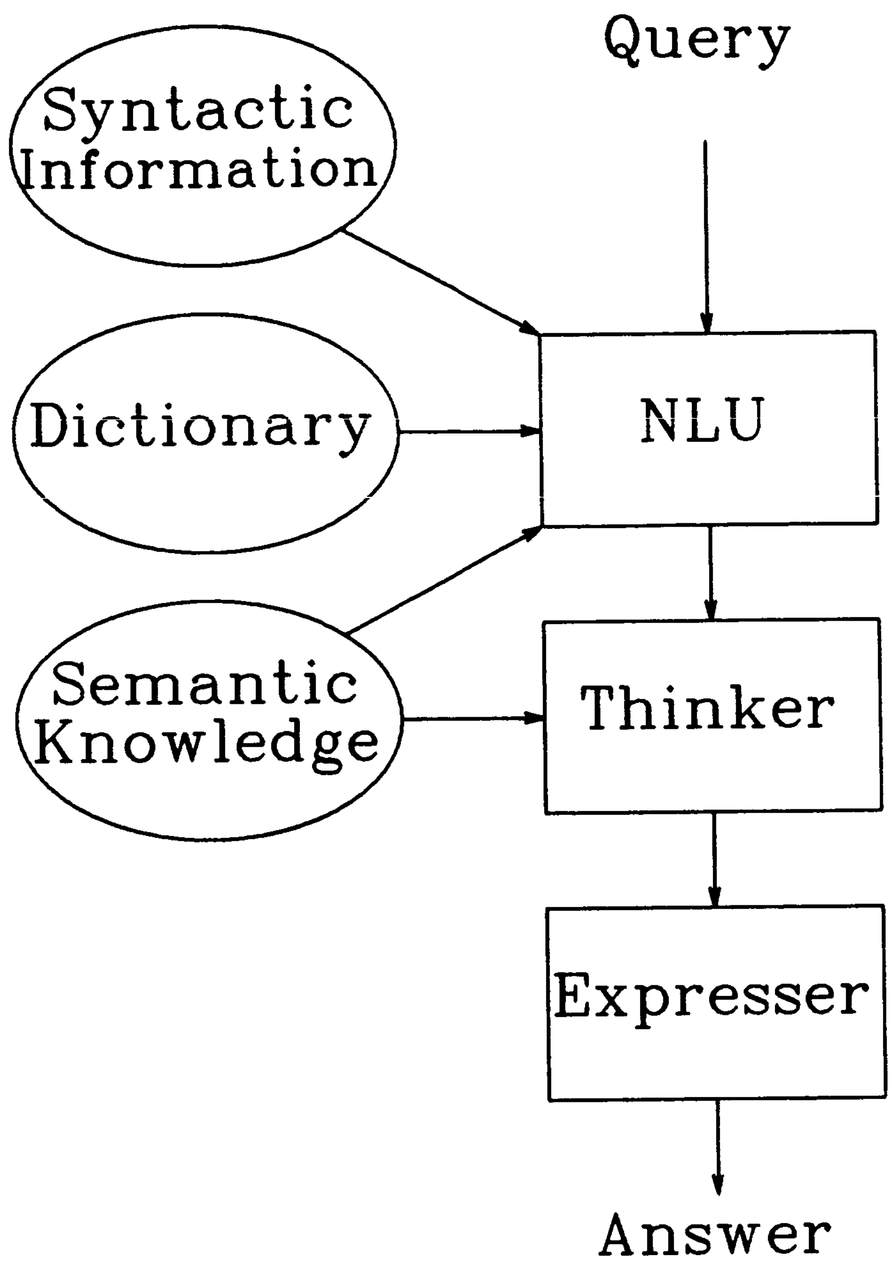
Computational Chemistry

- Calculate material properties from fundamental physics
- Complements and supplements experiment
- Problem sizes often
  - 40,000×40,000 random sparse arrays
  - 400 billion floating point operations
  - 64 Mword files
- Currently studying Kapton to understand degradation experienced on Shuttle missions

Findings

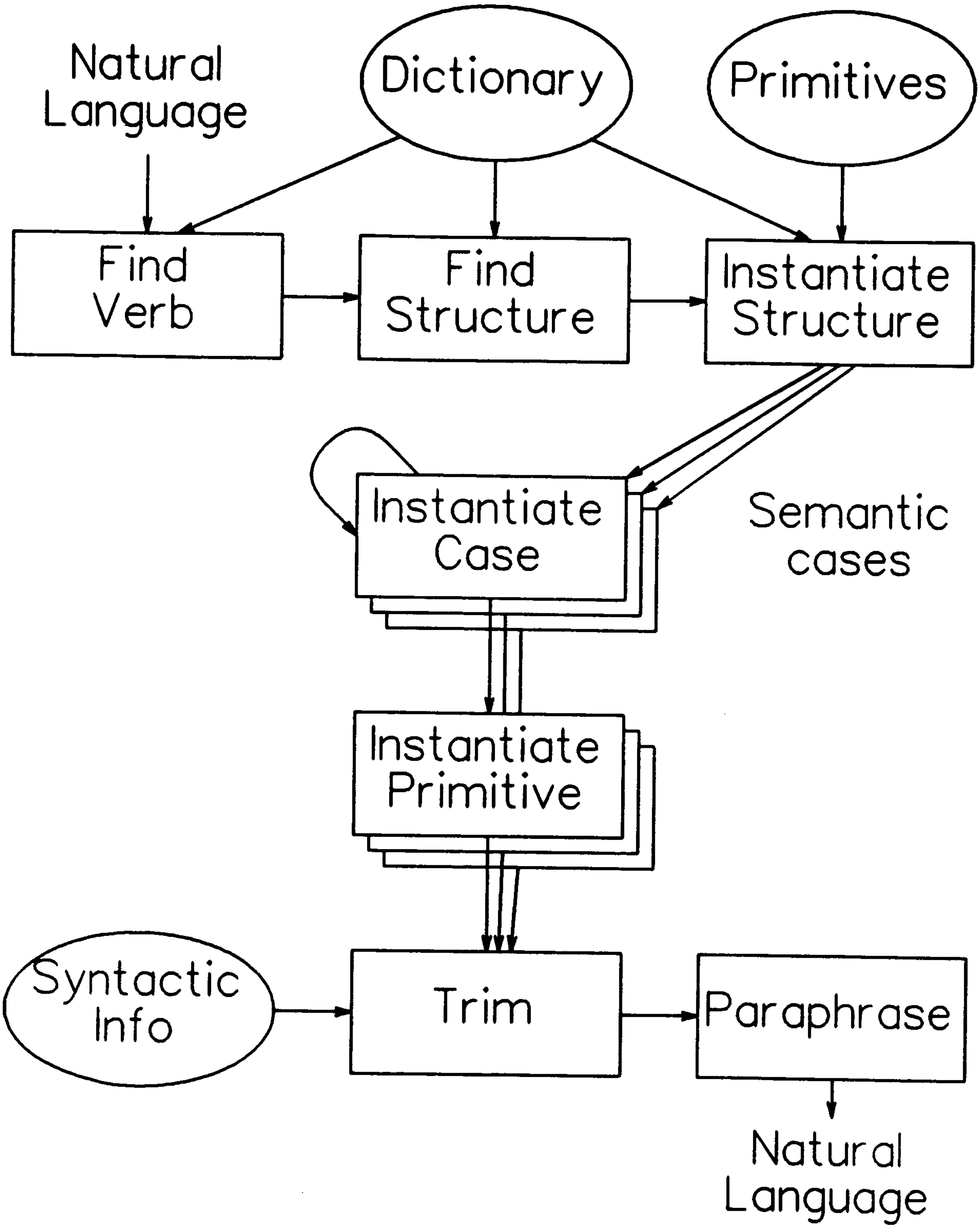
- Algorithms adaptable
  - Diatomic Integral Code (DERIC) may operate at near 1.2 GFLOP rate
- Major problem was I/O
  - inadequate bandwidth
  - data set partitioning
- 2K instruction cell memory too small
- VAL compiler directives may be difficult to write
- Since machine is single user I/O wait time cannot go to another job
- Fewer, higher–performance PEs recommended

Natural Language Processing



NLU – Natural Language Understander

Structure of Natural Language Processor



Structure of the NLU VAL Program

Findings

- Lack of global databases a significant problem
- Lack of high-level support for I/O
- New algorithmic approaches must be devised that readily exhibit parallelism

Queueing Network Models

- A tool for deriving performance estimates for computer systems and communication networks
- Compute performance measures such as throughput, response time, utilization, average queue length

Findings

- Practical considerations limit useful problem sizes to those that can be solved interactively on sequential computers
- Approximate techniques suffice for very large problems
- The most parallel algorithm did not lead to efficient parallel code due to fan–in/fan–out restrictions

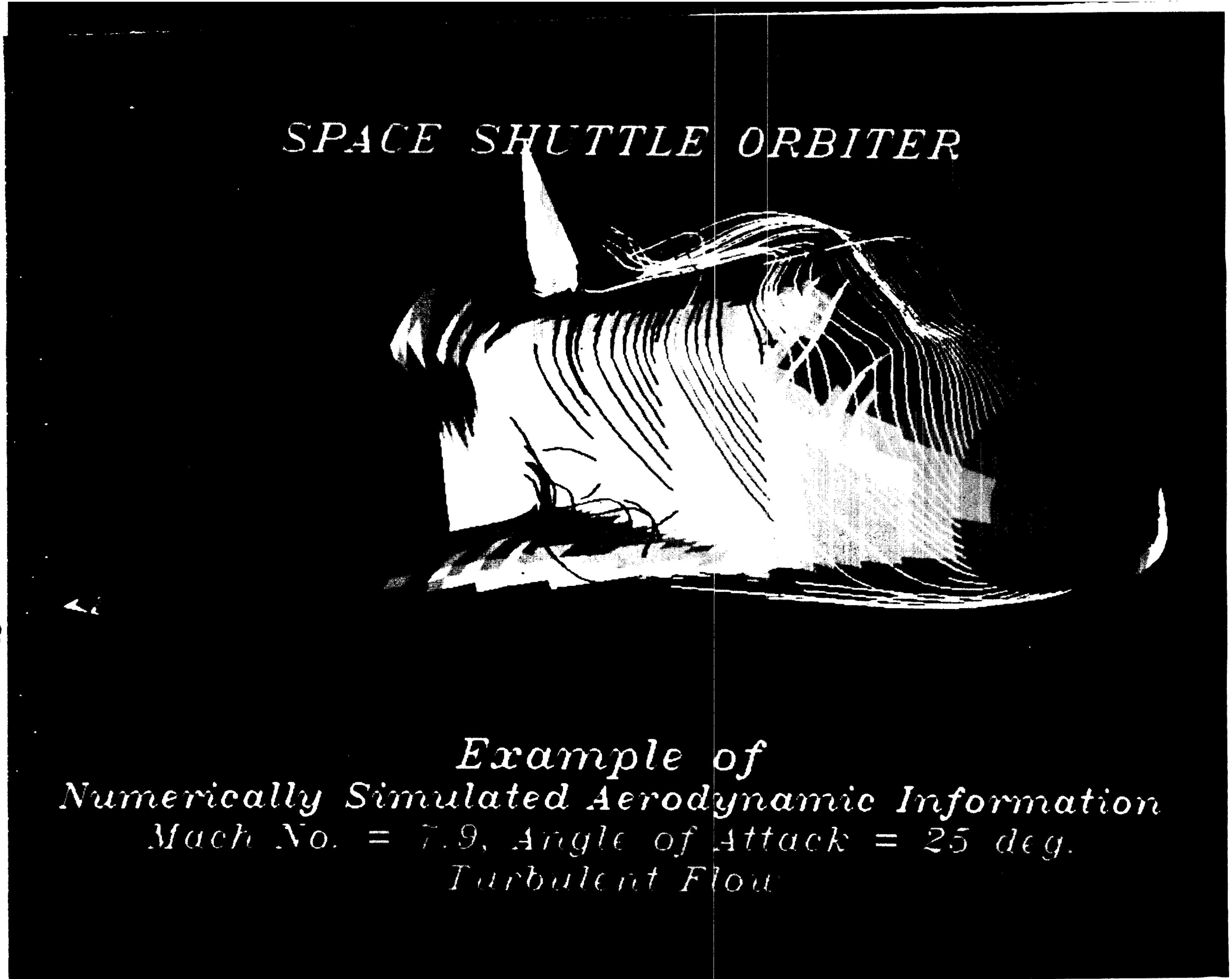
What We Learned

- VAL readily learned
- Data flow model of computation readily learned
- Most applications could use all parallelism available
- Smaller machines may be quite cost effective

- VAL programming environment must improve; debugging support
- Efficient sparse matrix handling may require hardware support
- Array memory bandwidth should be increased
- Array memory size should be increased
- Bandwidth for disk I/O should be increased
- Appropriate number of PE logic units difficult to assess

Benefits of Study

- Broken some of the resistance of users to look at new computer architectures
  - strong effort made by team members
- Computer scientists working with users
  - machine and language design changes
- Knowledge on how to conduct future studies
  - other combinations of model/machine/applications


SPACE SHUTTLE ORBITER
Example of
Numerically Simulated Aerodynamic Information
Mach No. = 7.9, Angle of Attack = 25 deg.
Turbulent Flow

References

1. G. B. Adams, R. L. Brown, and P. J. Denning, *Report on an Evaluation Study of Data Flow Computation*, RIACS Technical Report, TR 85.2, April 1985.

2. G. B. Adams, R. L. Brown, and P. J. Denning, *On Evaluating Concurrent Architectures*, RIACS Technical Report, May 1985.

3. G. B. Adams and P. J. Denning, *A Ten Year Plan for Concurrent Processing Research*, RIACS Technical Report, March 1984.

4. W. B. Ackerman and J. B. Dennis, *VAL -- A Value-Oriented Algorithmic Language: Preliminary Reference Manual*, MIT/LCS/TR-218, Massachusetts Institute for Technology, June 1979.

5. J. B. Dennis, "Data Flow Ideas for Supercomputers," *Proc. COMPCON84*, February 1984.

6. J. B. Dennis, G. R. Gao, and K. W. Todd, "Modeling the Weather with a Data Flow Computer," *IEEE Transactions on Computers*, vol. C-33, July 1984.

7. P. J. Denning and K. C. Sevcik, "Execution of Queueing Network Analysis Algorithms on a Data Flow Computer Architecture," RIACS Technical Report, 1985.

8. D. A. Reed and M. L. Patrick, "Iterative Solution of Large, Sparse Linear Systems on a Static Data Flow Architecture: Performance Studies," RIACS Technical Report, February 1985.

9. M. L. Merriam, "Application of Data Flow Concepts to a Multigrid Solver for the Euler Equations," *Proc. 2nd Copper Mountain Conf. on Multigrid Methods*, April 1985.

10. E. Levin, *Performance Evaluation of a Static Data Flow Processor for Transformations of Large Arrays*, RIACS Technical Report, TR 85.1, Jan. 1985.

11. D. S. Eberhardt and K. Rowley, *An Analysis of Static Data Flow to a Sample CFD Algorithm*, RIACS Technical Report, Mar. 1985.

N87-29138

p.22

Computer Architectures for Computational Physics

work done by

Computational Research and Technology Branch

and

Advanced Computational Concepts Group

Ames Research Center

The following slides describe the importance of having high performance number crunching and graphics capability. They also indicate the types of research and development underway at Ames Research Center to ensure that, in the near-term, Ames is a smart buyer and user, and in the long-term we know what the best possible solutions are for our number crunching and graphics needs.

The drivers for this research are real computational physics applications of interest to Ames and NASA. We are concerned with how to map the applications, how to develop the optimal system software and system architecture, and how to maximize the physics learned from the results of the calculations (which at the present time means graphics). We are utilizing a group of DEC and CRAY manufactured MIMD architectures, various simulation tools for larger MIMD architectures, and also plan to utilize various versions of the Hypercube architecture. To control flow we are looking at simulations and prototypes for the study of data flow and systolic architectures. At present, it is a competition between the three architectures to determine which one will hold the most promise for the early 1990s. Once we have discovered which one (or two) hold the promise we will concentrate our computer science R&D in that area.

The computer graphics R&D activities are directed at getting maximum information from our three-dimensional calculations by utilizing the real time manipulation of three-dimensional data on the Silicon Graphics IRIS Workstation. We are also working on new algorithms which will permit the display of experimental results, which are sparse and random, the same way we display computed results, which are dense and regular. This would permit the synergistic coupling of computational and experimental techniques.

Computer Architectures for Computational Physics

by

Computational Research and Technology

and

Advanced Computational Concepts

presented by

K. G. Stevens, Jr.

Ames Research Center

Related Research and Development

More than 50 academic projects to build prototype systems

Many start—up and established companies developing SIMD, MIMD, and Systolic architectures

Several Government Agencies including DARPA, DoE, and NSA are into architecture studies

Rapid growth in computer graphics hardware by start—ups and established companies

How the Research at Ames is Different

Directed towards the computational physics applications of interest to NASA and Ames

Total system approach including hardware, software, applications, peripherals, and the user interface

Complete application programs are the target

Existing, Emerging, and Future designs are studied

OBJECTIVE

Conduct Research Which Will Have Benefit to Computational and Experimental Physics Research

Computer Architecture

Short—term: How do we use what we have and what should we buy?

Long—term: What are the best architectures possible?

Computer Graphics

Develop new algorithms and software to exploit computer graphics for experimental and computational physics

Technical Approach

Start with "real" complete applications

Map them onto architectures of interest

Predict performance via analysis, simulation, emulation and/or execution

Compare with other architectures and consider performance improving modifications

Determine the user interface implications —— programming languages, debuggers, environments, graphics packages, etc.

Areas of emphasis

- Architectures for "Number Crunching"
  - SIMD
  - MIMD
  - Data Flow
  - Systolic Arrays
- Computer Graphics

Algorithms of Interest

TWING

Conservative Full Potential Equation

(Implicit, Approximate Factorization Algorithm)

AIR3D

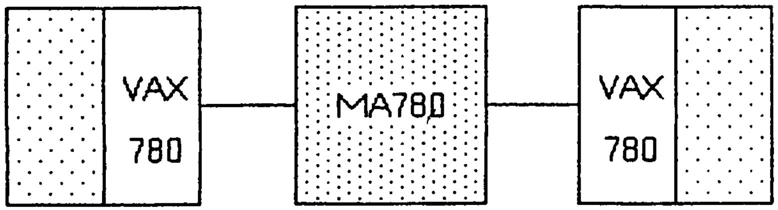
Reynolds—Averaged Navier Stokes

(Implicit, Approximate Factorization Algorithm)

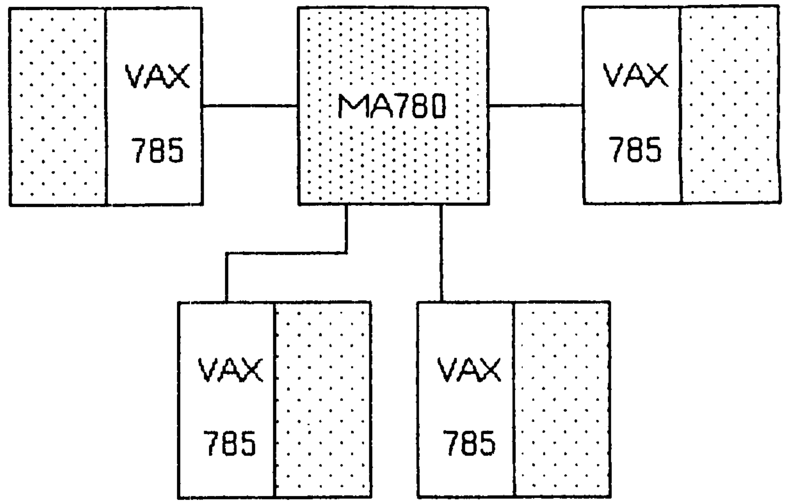
LES

Large Eddy Simulation Utilizing Spectral Methods

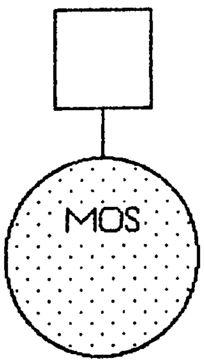
Architectures



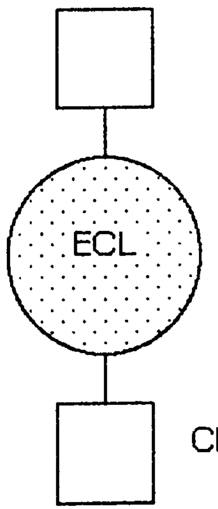
Dual VAX 11/780



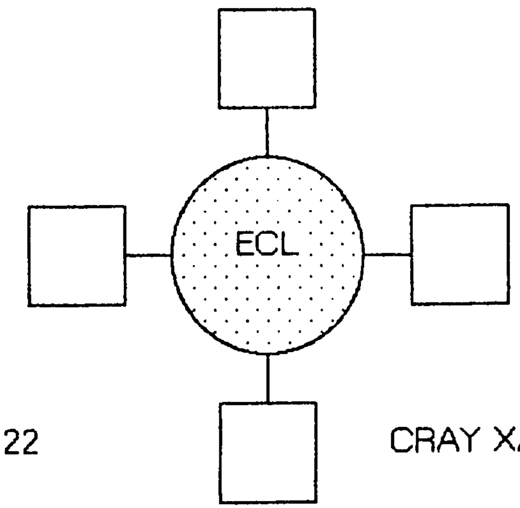
Quad VAX 11/785



CRAY X/MP–12



CRAY X/MP–22



CRAY X/MP–48

Performance of Multitasking on the CRAY X/MP

LES with 100 iterations on a 32^3 Mesh

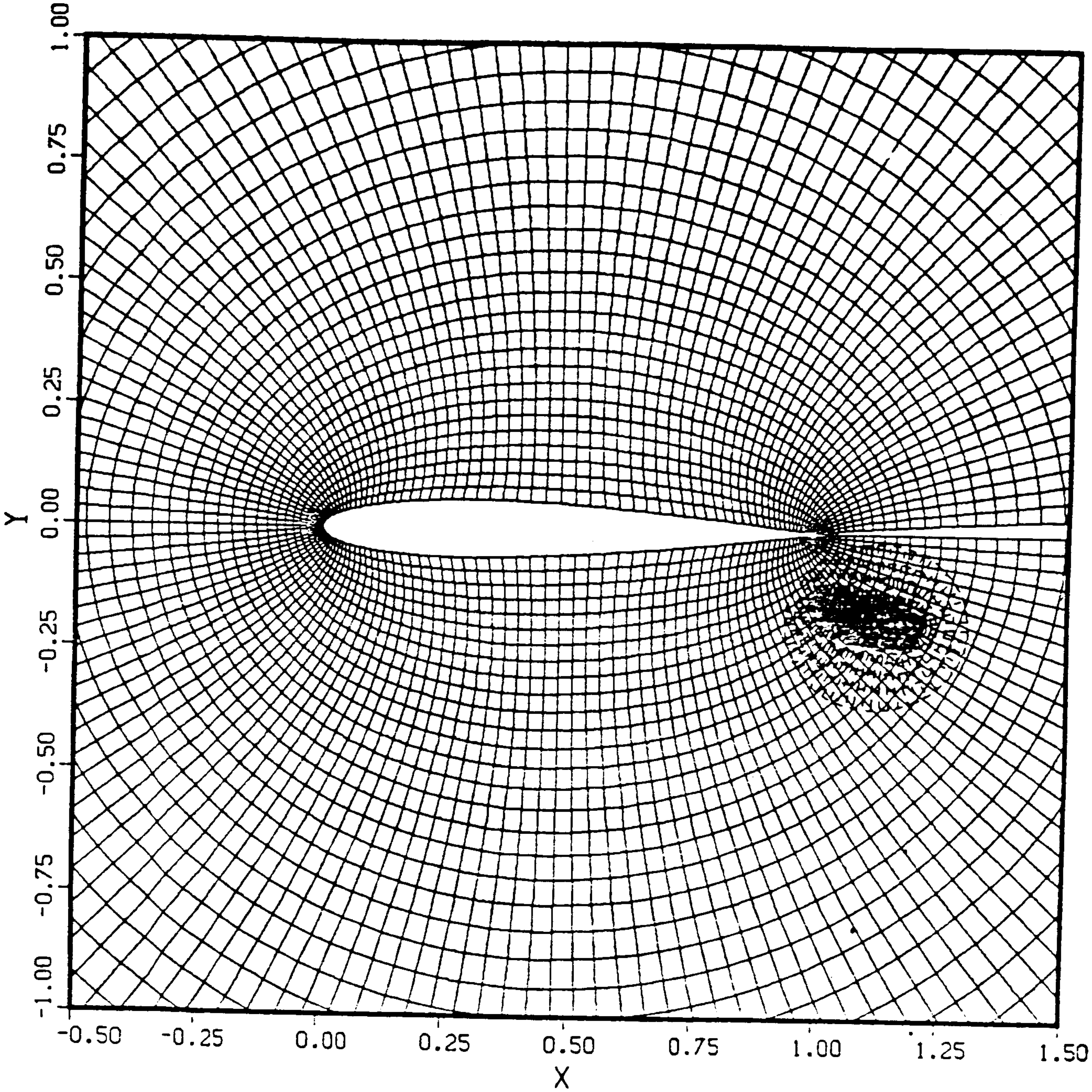
CPU Times in Seconds/Speed up

Mode	Loop	Bombard	LES	None
Static	31.1 / 1.85	35.9 / 2.13	33.2 / 1.98	* / *
Stack	31.1 / 1.85	36.2 / 2.15	33.5 / 1.99	* / *
Mtsk−1	31.1 / 1.86	36.3 / 2.16	33.6 / 2.00	* / *
Mtsk−2	* / *	* / *	* / *	33.0 / 1.96

Time in Seconds	
Task 1	16.8
Task 2	16.2
Total	33.0

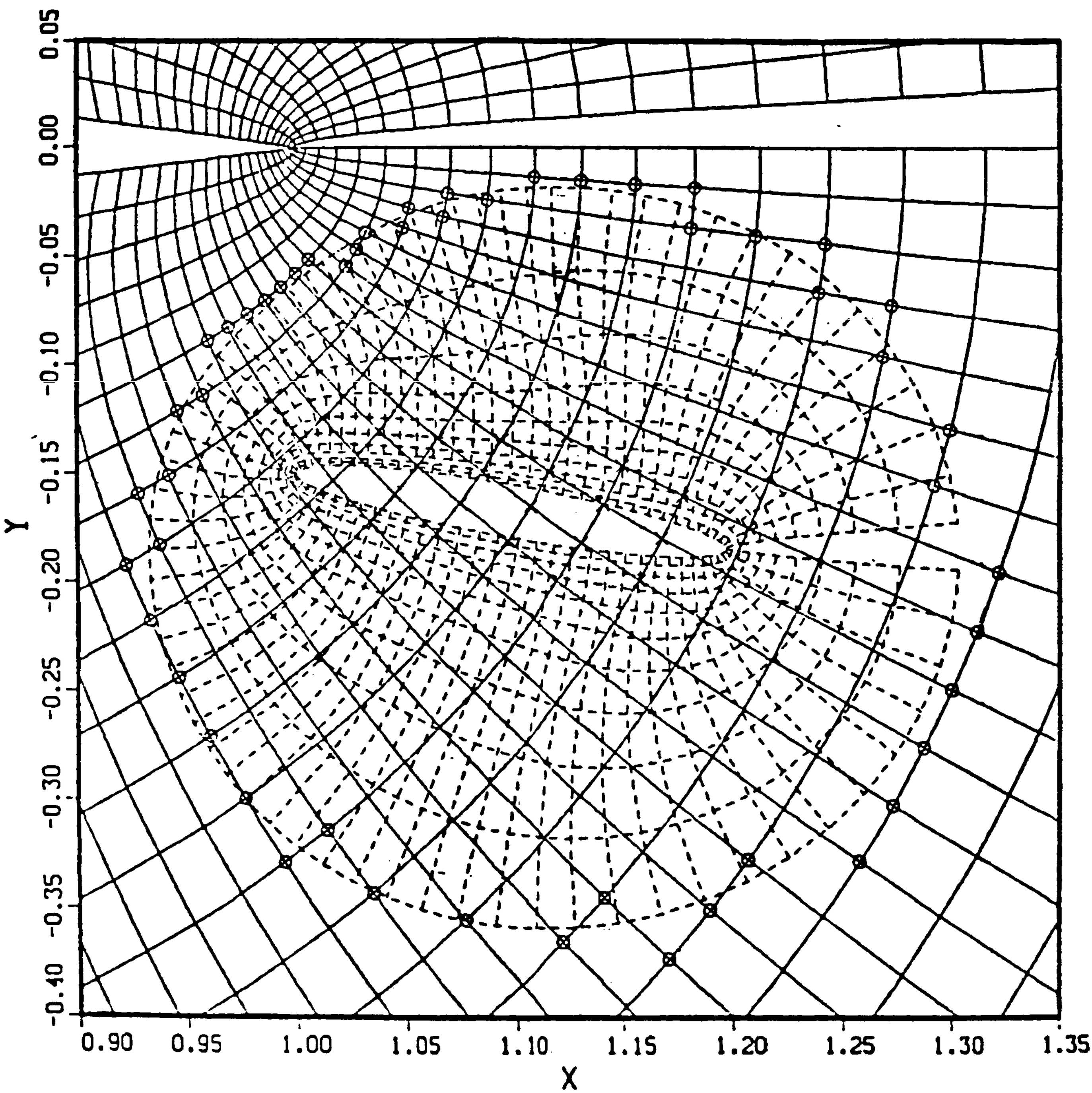
Performance of Multitasking on the Dual VAX 11/780

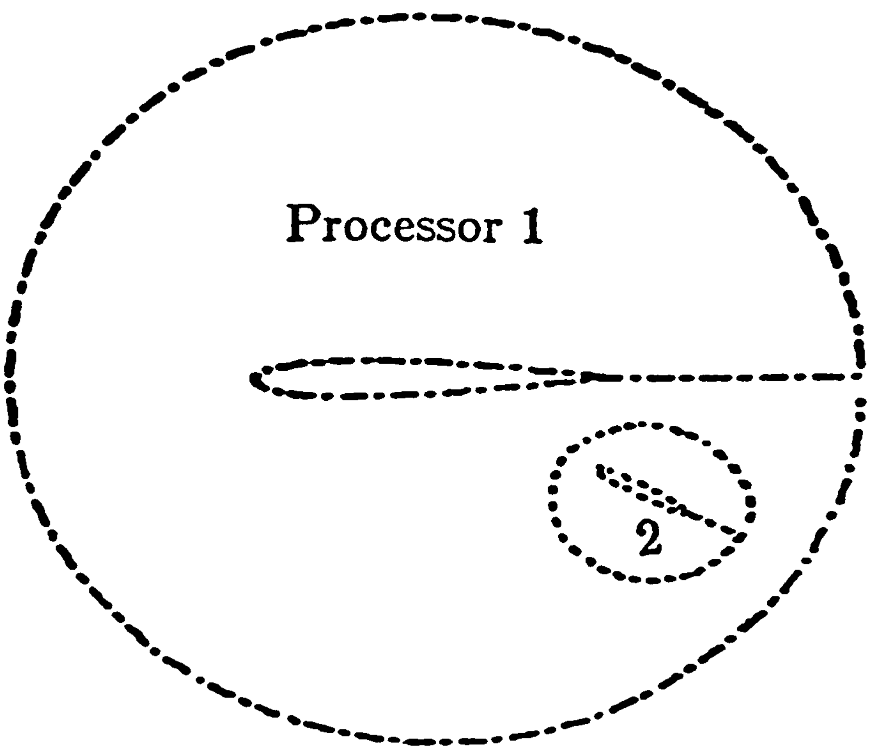
Code	Speedup
Twing	1.55
AIR3D	1.85
LES	1.98



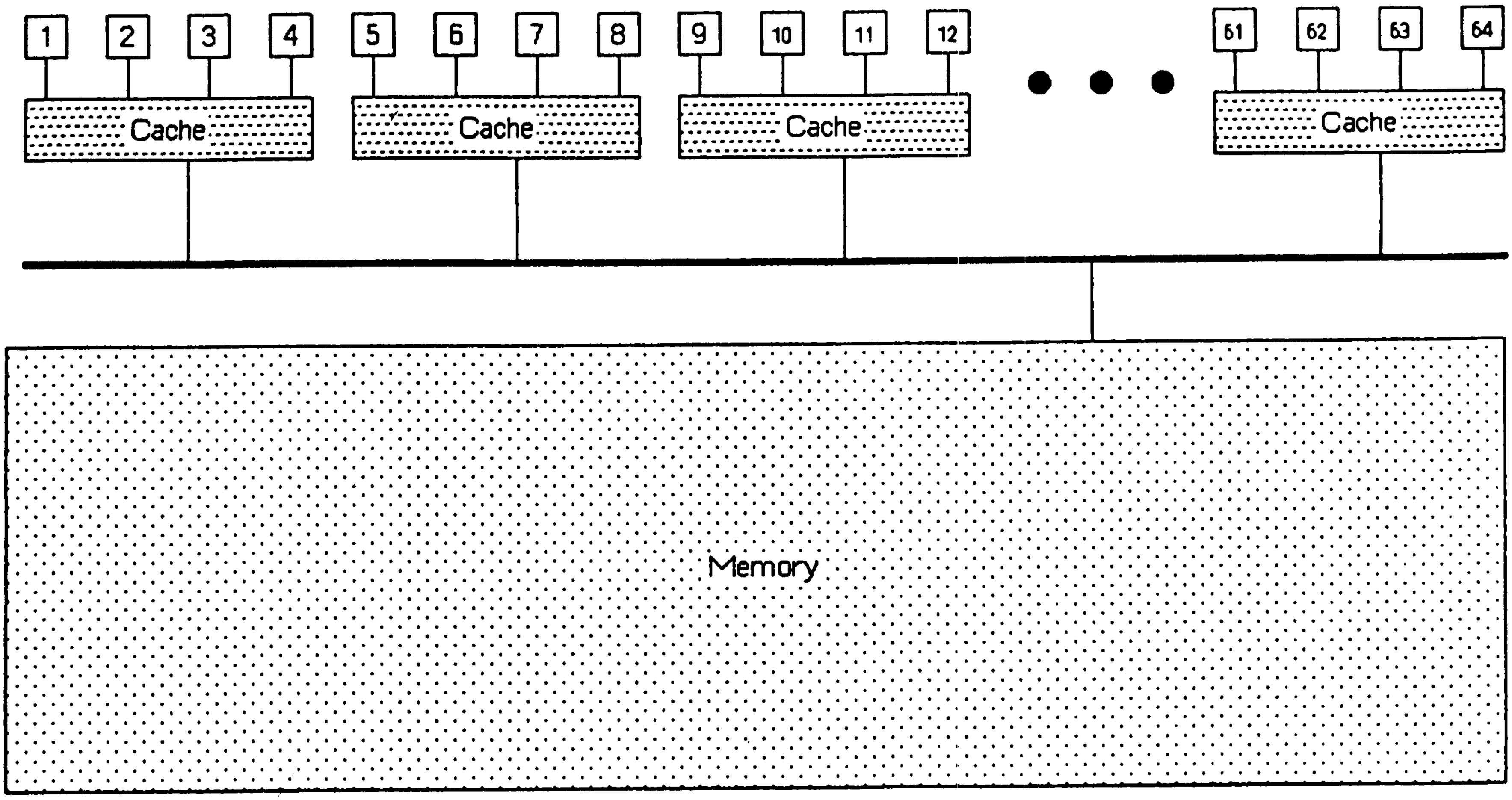
1.00
0.75
0.50
0.25
0.00
Y
-0.25
-0.50
-0.75
-1.00
-0.50
-0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
X

CHIMERA GRID




Processor 1
2

PPA Architecture



Circuit-switched Network Simulation

Motivation and Objectives

- Understand performance of networks which could be used to build high-performance parallel architectures
- Use a real application (LES) from Ames to generate data for this study
- Understand how a real CFD problem could map onto a large MIMD architecture

The Model

- A circuit switched Omega network serving multiple processors connected to multiple modules of a shared memory
- Queues of requests exist at each processor port and are served one at a time

Construction of the Simulator

- Discrete event simluation facility of SLAM driven by FORTRAN subroutines
- Statistics collected on service times

Bandwidth of Network for Various Cases

Three cases:

- Real data from a CFD code (LES)
- Random data
- Infinite vectors with p=1

Total Bandwidth in MW/sec.				
n	MAX	Random	Vectors	Actual
8	36	12.5	5.52	5.75
16	67	12.2	5.60	5.62
32	123	5.12	5.12	5.24
64	229	5.76	4.16	4.36

For comparison look at Crays:

Maximum Bandwidth in MW/sec.	
Machine	Bandwidth
Cray 1	80
Cray X-MP	631
512*512	1500

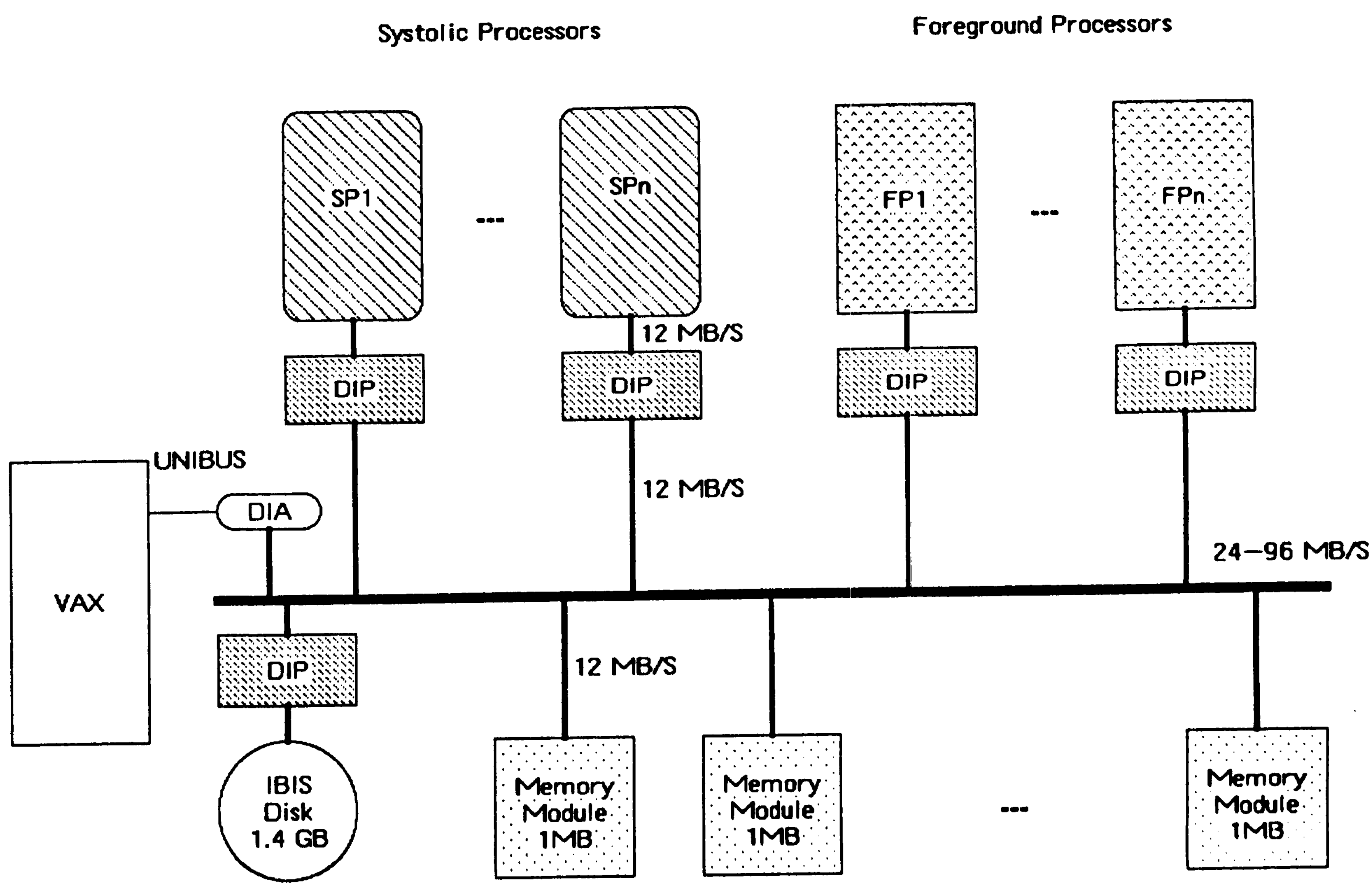
Conclusions from the Network Simulation

Modelling network traffic with streams of random data can be very misleading since actual codes exhibit a very different behavior

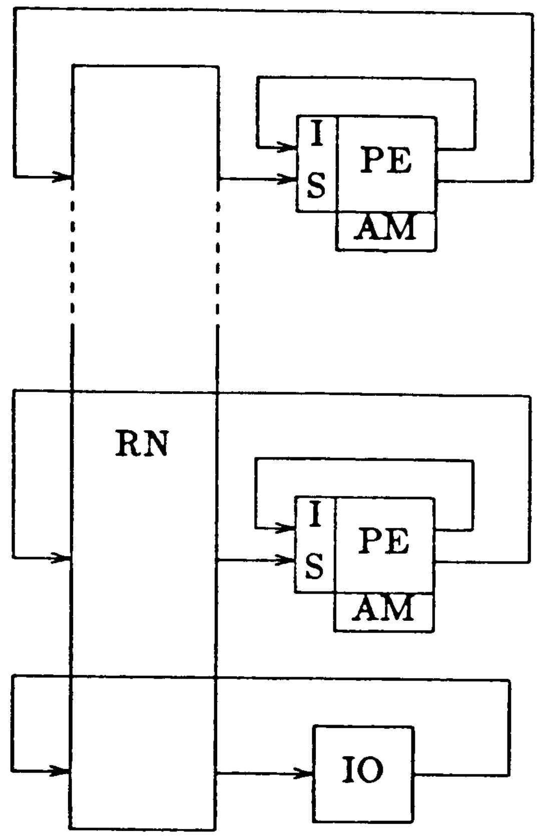
The bandwidth of the network does not increase linearly with the number of ports

A circuit-switched network such as this is far too slow to be useful for building high-performance MIMD architectures

System Architecture of a Systolic Attached Processor



Static Data Flow Machine Architecture



RN Routing Network. 512 by 512, 16 bit data paths, operates at > 5MHz, average rate of transmitting FP packets 0.25 MHz from a single PE to another.

PE Processing Elements. 5 to 8 MFLOPS with two 1.25 to 2 MFLOP multipliers. 256 PE's in the system.

IS Instruction Store. 1024 cells for FP instructions, 1024 for others.

AM Array Memory. Size not fully determined. At least 256K 64 bit words per PE.

IO Input Output. Includes mass memory, host processor, and display systems. 256 paths through the RN are reserved for IO.

Status of Data Flow Simulator

Design of simulator complete.

Coding of simulator begun.

Coding being done in PASCAL, and problems encountered with CRAY compiler

Input Codes are being developed

Questions to be Answered by the Simulator

Are previous performance predictions realistic?

What is the load on the routing network? Can the network handle it?

How much instruction memory and array memory is needed?

What is the effect of adding more processors?

What is the best way to distribute instructions across the processming elements?

GRAPHICS RESEARCH AND DEVELOPMENT

PURPOSE: PROVIDE FOR GREATER USER PRODUCTIVITY BY ENABLING VISUALIZATION OF 3 DIMENSIONAL EXPERIMENTS AND SIMULATIONS (EG. VISUALIZATION OF FLUID FLOW IN THREE DIMENSIONS)

RESULTS:

Established consortium agreement with Robert Barnhill (Utah) to develop algorithms for generating smooth contours from sparse random data such as those from wind tunnel tests

Developed State-of-the-Art Three-Dimensional graphics program for the Silicon Graphics IRIS terminals and demonstrated its use for several computational physics applications

Organization of Data Flow Simulator

DRIVER: Defines Characteristics of the architecture to be simulated, e.g. network characteristics number of processing elements, number and type of functional units in processing elements, etc.

TRANSLATOR: Takes code written in intermediate Data Flow Language (IF1) and translates it to input for the simulator (LLNL supplying SISAL to IF1 front end)

SIMULATOR: Perfroms actual simulation

DRIVER will run on VAX to allow interactive use. TRANSLATOR and SIMULATOR will run on Cray because of length of run.

PROGRESS IN KNOWLEDGE REPRESENTATION RESEARCH

COMPUTER SCIENCES AND DATA SYSTEMS TECHNICAL SYMPOSIUM

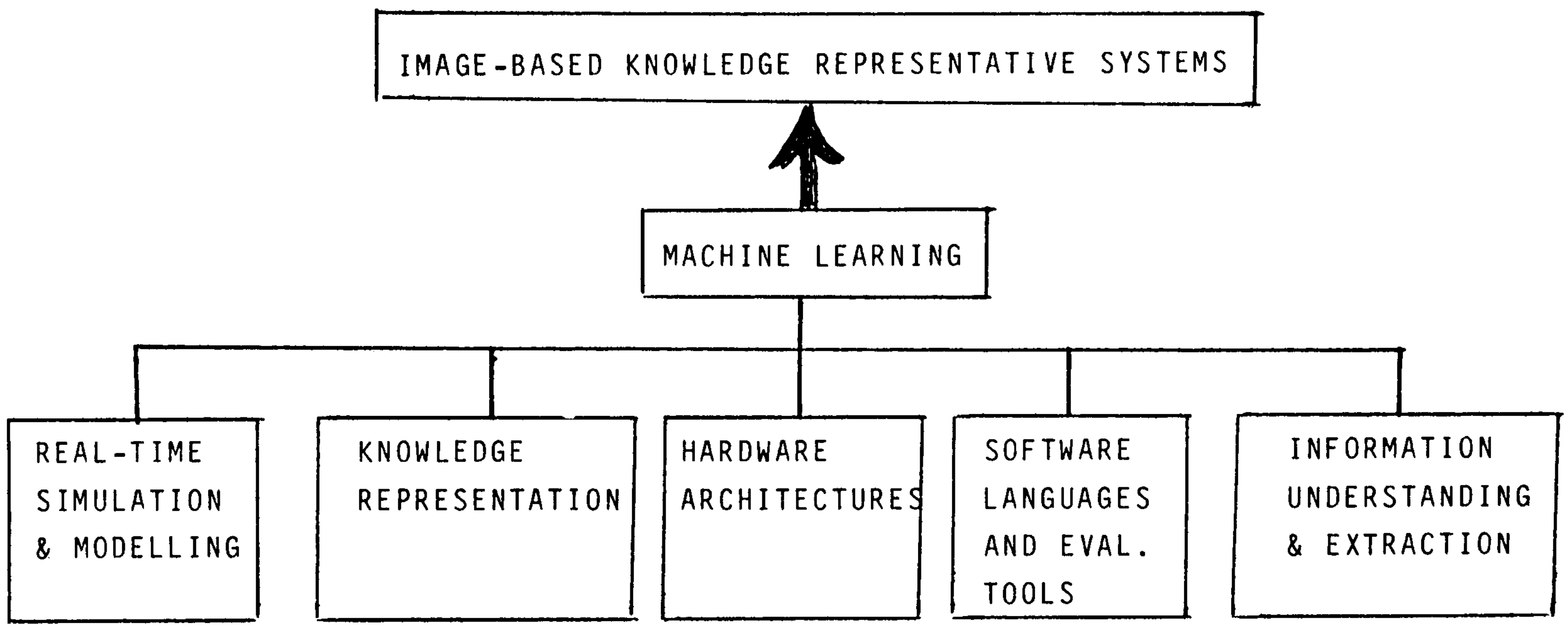
APRIL 16, 1985

DR. HENRY LUM
NASA AMES RESEARCH CENTER

KNOWLEDGE REPRESENTATION RESEARCH

GOAL: RESEARCH LEADING TO IMAGE-BASED KNOWLEDGE REPRESENTATION SYSTEMS

OBJECTIVES AND TECHNICAL APPROACH



REAL-TIME SIMULATION AND MODELLING

PRINCIPAL INVESTIGATORS: AMES RESEARCH CENTER
MIT

OBJECTIVE: DYNAMIC SIMULATION AND MODELLING OF PLANNING SYSTEMS WITH REAL-TIME SENSOR INPUTS - DERIVATION OF "ERROR SIGNAL" TO REPLAN NEW ACTIVITY

CURRENT STATUS:

- O LONG-RANGE RESEARCH PROGRAM CURRENTLY BEING REDEFINED
- O EXISTING SIMULATION FACILITIES BEING REDIRECTED TO SUPPORT ACTIVITY
- O EVANS & SUTHERLAND PS-350 SYSTEM BEING INSTALLED IN AUTOMATION SCIENCES LABORATORY TO VALIDATE/EVALUATE TECHNOLOGY PLAN - OPERATIONAL MAY 1985
- O DEMONSTRATION OF AUTOMATED CRYOGENIC TRANSFER TO BE USED TO VALIDATE SIMULATION/MODELLING SYSTEM - BEING DONE IN CONJUCTION WITH JSC

ISSUES: INTERACTION OF REAL-TIME SENSOR SYSTEMS WITH PLANNING SYSTEMS

- O IMPACT ON HARDWARE AND SOFTWARE ARCHITECTURE
- O IMPACT ON PLANNING SYSTEMS
- O ROLE OF HUMAN IN THE AUTOMATED PLANNING SYSTEM

REAL-TIME SIMULATION AND MODELING

The interaction between task planning and execution systems with the "external world" and the operator is not widely understood at the present time. Of particular interest is the derivation of the "error signal" (difference signal in symbolic notation between the expected action and the actual action) and its interaction with the planner in real-time. Issues being investigated are the role of the operator in the planning cycle, the interpretation and translation of the symbolic "error signal" into a new planning requirement and a resulting analog correction signal, the interpretation (understanding) of the sensor data, and the display/explanation of the data to the operator.

To understand the above issues, a feasibility study is being undertaken to automate the transfer of cryogens between a helium dewar and an instrument. Coupling and uncoupling of cryogenic fill values are also included in the study. This effort is being done in conjunction with NASA JSC, which has provided the procedures used by the astronauts in a manual transfer feasibility study and which will provide a sample valve. Target mission is the Space Infrared Telescope Facility (SIRTF).

KNOWLEDGE REPRESENTATION

PRINCIPAL INVESTIGATORS: AMES RESEARCH CENTER
STANFORD UNIVERSITY
UNIVERSITY OF CALIFORNIA, BERKELEY
SRI, INTERNATIONAL

OBJECTIVE: DEVELOPMENT OF DOMAIN-INDEPENDENT KNOWLEDGE REPRESENTATION TOOLS WHICH CAN BE USED BY THE AGENCY IN THE DEVELOPMENT OF APPLICATION-SPECIFIC EXPERT AND PLANNING SYSTEMS. INTEGRATION OF KNOWLEDGE-REPRESENTATION TECHNIQUES TO MACHINE LEARNING ALGORITHMS.

CURRENT STATUS:

- O RESEARCH IN KNOWLEDGE REPRESENTATION TECHNIQUES APPLICABLE TO EXPERT SYSTEMS, PLANNING SYSTEMS, AND FAULT DIAGNOSTICS SYSTEMS IN THIRD YEAR OF STUDY - PRELIMINARY TOOLS EXPECTED TO BE DELIVERED FOR EVALUATION LATE CY 1986
- O PRELIMINARY APPLICATIONS BEING DEVELOPED TO EVALUATE TOOLS FOR MULTI-USER/DISCIPLINE ENVIRONMENT AND EXPLANATION CAPABILITIES
- O INTEGRATION OF FUZZY SET THEORY INTO KNOWLEDGE REPRESENTATION ALGORITHMS BEING EVALUATED - PRELIMINARY DATA INDICATES THAT THEORY HAS PRACTICAL APPLICATIONS TO DOMAINS WHERE KNOWLEDGE IS UNCERTAIN OR UNRELIABLE. RELIABILITY OF DECISIONS COULD BE AS HIGH AS 70 PERCENT.

ISSUES:

- O COMMERCIAL KNOWLEDGE REPRESENTATION SYSTEMS INADEQUATE FOR DEEP AND COMPLEX SYSTEMS. REQUIRES SKILLED KNOWLEDGE ENGINEER. NOT EFFICIENT FOR ALL TECHNICAL DOMAINS.
- O COMMON PROGRAMMING LANGUAGE AND SYSTEMS REQUIRED FOR TRANSFER OF RESEARCH TOOLS - PROGRESS BEING MADE TO ESTABLISH "STANDARD ENVIRONMENT"
- O "GLAMOUR PROBLEM" - CONCERN EXISTS THAT TOO MUCH IS BEING PROMISED WITHOUT IN-DEPTH UNDERSTANDING OF THE UNDERLYING PROBLEMS - COULD ENDANGER EFFORT

KNOWLEDGE REPRESENTATION

Knowledge representation research is underway to correct the present deficiencies of commercial expert building tools, i.e., capability to reason, represent knowledge, and acquire knowledge from input data; verify and explain the heuristic rules used in the manipulation of the knowledge and the execution of the decision; ability to interact with multiple knowledge domains and coorelate the knowledge represented in each domain; and the ability to learn as more knowledge is acquired. The current research involves areas of planning, representation of knowledge, fault diagnostics, and uncertainty (where knowledge is unknown or unreliable). The research tools are being evaluated in limited applications to validate their potential relative to the above stated objectives and to understand their impact on the need for skilled vs. unskilled knowledge engineers and on the performance of the hardware and software architectures. A limited set of the tools will be available for Agency evaluation during late CY 1986.

The integration of the knowledge representation research will hopefully lead to machine learning algorithms and executive controllers for multiple expert systems/planners/diagnostic systems. Research is this field is very complex and breakthroughs/technology readiness for user applications cannot be forecasted with any degree of certainty.

HARDWARE ARCHITECTURES

PRINCIPAL INVESTIGATORS: AMES RESEARCH CENTER
STANFORD
SYMBOLICS, INC.
TRW, INC. (ROLE BEING DISCUSSED)
UNIVERSITY OF ALABAMA (SDI CENTER OF EXCELLENCE IN OPTICAL COMPUTING)

OBJECTIVE: DEVELOPMENT OF A SPACE-BORNE VHSIC SYMBOLIC PROCESSOR WITH 15 TIMES THE PERFORMANCE OF THE CURRENT JAPANESE 5TH GENERATION MACHINE; DEVELOPMENT OF INTERFACES TO PROVIDE FRONT-END OPTICAL PROCESSING CAPABILITY AND OPTICAL READ-WRITE DISK STORAGE; DEVELOPMENT OF AN OPTICAL PROCESSOR WHICH WILL PROVIDE A FACTOR OF 10 IMPROVEMENT IN PERFORMANCE TO THE VHSIC SYMBOLIC PROCESSOR.

CURRENT STATUS:

- O DETAILED DEFINITION OF VHSIC SYMBOLIC PROCESSOR ARCHITECTURE UNDERWAY - TARGET DATE FOR CRITICAL DESIGN REVIEW 3RD QUARTER, FY 1986. CURRENT STUDIES SHOW APPROXIMATELY TWO TIMES PERFORMANCE OF THE CURRENT JAPANESE 5TH GENERATION MACHINE (PSI) USING THE IDENTICAL TEST CASES RUN IN PROLOG.
- O VHSIC CHIPS FOUND APPLICABLE TO THE PROCESSOR - TRW'S CONTENT ADDRESSABLE MEMORY CHIP COULD BE CRITICAL ELEMENT IN PROCESSOR DESIGN

- O WITH OPTICAL PROCESSOR FRONT-END AND OPTICAL READ-WRITE DISK STORAGE, VHSIC SYMBOLIC PROCESSOR COULD POTENTIALLY MEET THE NEEDS OF ALL PROJECTED SPACE STATION AUTOMATED SYSTEMS AND SCIENTIFIC EXPERIMENTS.
- O LABORATORY TESTS TO INVESTIGATE TECHNIQUES REQUIRED TO INCREASE THE SPEED OF THE PROGRAMMABLE LCD MASKS - CRITICAL ELEMENT IN OPTICAL PROCESSOR. CRITICAL DEMONSTRATION SET FOR LATE CY 1986 TO DETERMINE FEASIBILITY OF OPTICAL PROCESSOR FOR SPACE-BORNE APPLICATIONS.
- O OPTICAL READ-WRITE DISKS APPEAR TO BE VIABLE SOLUTION FOR DATA NEEDS OF SYMBOLIC PROCESSOR - REQUIREMENTS FOR CONTROLLER UNKNOWN AT THE PRESENT TIME. DATA TRANSFER SPEEDS COULD BE LIMITING FACTOR.

ISSUES:

- O TRANSLATION OF OPTICAL INFORMATION INTO SYMBOLIC REPRESENTATION
- O APPROACH TO ACCELERATE DATA TRANSFER RATES BETWEEN DATA STORAGE SYSTEM AND SYMBOLIC PROCESSOR
- O LIMITED RESOLUTION OF OPTICAL PROCESSORS RELATIVE TO DIGITAL PROCESSORS

HARDWARE ARCHITECTURES

PRINCIPAL INVESTIGATORS: AMES RESEARCH CENTER
STANFORD
SYMBOLICS, INC.
TRW, INC. (ROLE BEING DISCUSSED)
UNIVERSITY OF ALABAMA (SDI CENTER OF EXCELLENCE IN OPTICAL COMPUTING)

OBJECTIVE: DEVELOPMENT OF A SPACE-BORNE VHSIC SYMBOLIC PROCESSOR WITH 15 TIMES THE PERFORMANCE OF THE CURRENT JAPANESE 5TH GENERATION MACHINE; DEVELOPMENT OF INTERFACES TO PROVIDE FRONT-END OPTICAL PROCESSING CAPABILITY AND OPTICAL READ-WRITE DISK STORAGE; DEVELOPMENT OF AN OPTICAL PROCESSOR WHICH WILL PROVIDE A FACTOR OF 10 IMPROVEMENT IN PERFORMANCE TO THE VHSIC SYMBOLIC PROCESSOR.

CURRENT STATUS:

- DETAILED DEFINITION OF VHSIC SYMBOLIC PROCESSOR ARCHITECTURE UNDERWAY - TARGET DATE FOR CRITICAL DESIGN REVIEW 3RD QUARTER, FY 1986. CURRENT STUDIES SHOW APPROXIMATELY TWO TIMES PERFORMANCE OF THE CURRENT JAPANESE 5TH GENERATION MACHINE (PSI) USING THE IDENTICAL TEST CASES RUN IN PROLOG.
- VHSIC CHIPS FOUND APPLICABLE TO THE PROCESSOR - TRW'S CONTENT ADDRESSABLE MEMORY CHIP COULD BE CRITICAL ELEMENT IN PROCESSOR DESIGN

An efficient method for translating optical information into symbolic representation is not known at this time. This issue will need to be resolved before the symbolic processor can interact in real-time with the symbolic processor. In addition, the impact of the resolution of optical processors for real-time applications will need to be investigated.

SOFTWARE LANGUAGES AND EVALUATION TOOLS

PRINCIPAL INVESTIGATORS: AMES RESEARCH CENTER
STANFORD UNIVERSITY
LUCID (GABRIEL) - BEING DISCUSSED

OBJECTIVE:

- O DEVELOPMENT OF A "STANDARIZED" PROGRAMMING ENVIRONMENT
- O DEVELOPMENT OF AN EXPERT PROGRAMMING ENVIRONMENT FOR TRANSPARENCY OF LISP, PROLOG, AND ADA
- O DEVELOPMENT OF STANDARD BENCHMARKS TO EVALUATE POTENTIAL SYMBOLIC ARCHITECTURES

CURRENT STATUS:

- O 1500-RULE BENCH MARK CASE IN PROGRESS - EXPECTED TO BE COMPLETED MID-CY 1986
- O TEST CASES FOR EVALUATING NUMERIC AND SYMBOLIC PROCESSORS COMBINING NUMERIC AND SYMBOLIC ALGORITHMS BEING TESTED - REPORT EXPECTED LATE CY 1985; WILL PROVIDE EFFICIENT METHOD FOR EVALUATING WORK STATIONS, PERSONAL COMPUTERS, AND LARGE FRAME ARCHITECTURES FOR BOTH NUMERIC AND SYMBOLIC APPLICATIONS
- O RESEARCH IN THE USE OF FIRMWARE FOR AN EXPERT PROGRAMMING ENVIRONMENT IN 2ND YEAR OF EFFORT - FEASIBILITY UNCERTAIN AT THIS TIME

ISSUES: NONE

SOFTWARE LANGUAGES AND EVALUATION TOOLS

Research in this area is directed at the establishment of a "standarized" programming environment for the development of expert, fault diagnostic, and procedural planning systems. In addition, progress has been made in the "standarization" of ground-based symbolic processors (Symbolics 3600, 3670, and 3640) which will significantly aid in the transfer of software between Centers and NASA-sponsored university research. On-going efforts between Ames, GSFC, and JSC have demonstrated the viability and productivity of such a concept. Expert systems will be applied in many technical disciplines; as a result, there is a need to provide a programming environment based in the use of LISP-like languages (Common LISP is a potential "standard"), Prolog, and ADA since each language is extremely efficient in specific domains. A potential solution is an expert programming system (or automated programmer) which will select the optimum language based on the supplied user inputs and requirements and allow the user to develop the software code in a natural language context. Research in both the use of firmware and software algorithms to accomplish this is currently underway.

The state-of-the-art in architectures (work stations, personal computers, and large-frame machines) are rapidly advancing. There is no current way to efficiently evaluate these architectures for overall performance relative to specific applications and codes; test cases involving numeric and symbolic computations and large (1500 rules minimum) rule-based systems are currently in development and/or evaluation. The use of "standarized" bench marks will allow correlation of performance data between different machines and permit identification of architectures which are most efficient for specific applications. Tradeoffs between architectures can also evaluated using a common baseline.

INFORMATION UNDERSTANDING AND EXTRACTION

PRINCIPAL INVESTIGATORS: AMES RESEARCH CENTER
MIT
UNIVERSITY OF MICHIGAN
UNIVERSITY OF TEXAS
MACHINE INTELLIGENCE CORPORATION

OBJECTIVE: DERIVE MAXIMUM INFORMATION CONTENT/UNDERSTANDING FROM INCOMING IMAGES

CURRENT STATUS:

- PROTOTYPE SYSTEM WILL BE COMPLETED IN LATE CY 1985 TO EVALUATE TRADEOFFS BETWEEN AN IMAGE-BASED EXPERT SYSTEM AND AN EXPERT IN THE AREA OF UPPER ATMOSPHERIC RESEARCH (AEROSOL PARTICLES)
- USE OF COLOR/IR VISION BEING INVESTIGATED FOR MAXIMUM INFORMATION CONTENT VERSUS GRAY-SCALE APPROACH
- INTEGRATION OF SENSOR INFORMATION AND ITS IMPACT ON MACHINE LEARNING ALGORITHMS UNDERWAY - 1ST YEAR OF RESEARCH

ISSUES: NONE

INFORMATION UNDERSTANDING AND EXTRACTION

With the approaching advent of "smart" sensors and complex scientific experiments/facilities, there is an ever increasing need for on-board data compression and preprocessing prior to the data being transmitted back down to the ground. Even then, it is likely that the channel capacity will be severely taxed and will not be able to accommodate all of the sensory data. As a result, there is a need to derive the maximum information content and understanding from images, patterns, and sensors. By understanding the minimum quantity of information required by an expert to make an intelligent decision, one can reduce the data required for transmission to the ground. The research in this area is focused on the above goad. Objectives are to understand the tradeoffs between human understanding and perception and machine processing/intelligence. It is felt that the use of color and infrared vision can increase the information content over that of gray-scale systems by at least a factor of 15; however, the tradeoffs involving the complexities of the hardware and the processing are not well understood at this time. Research is this area is directed towards a better understanding of the pertinent issues involved and the definition of guidelines which can be used to specify the performance characteristics of such a system relative to real-time image-based applications.

RESEARCH "TEST BEDS"

- O KUIPER AIRBORNE OBSERVATORY (KAO - C141A) - ASTROPHYSICS APPLICATIONS
  - O EXPERT SYSTEMS
  - O PLANNING SYSTEMS
  - O DIAGNOSTICS SYSTEMS
- O EARTH RESOURCES SURVEY AIRCRAFT (U-2 AND ER-2) - UPPER ATMOSPHERIC RESEARCH
  - O IMAGE-BASED EXPERT SYSTEMS
- O AIRBORNE RESEARCH LABORATORY (CV-990) - SPACE SCIENCE RESEARCH
  - O EXPERT SYSTEMS
  - O FAULT DIAGNOSTICS SYSTEMS
- O ARTIFICIAL INTELLIGENCE RESEARCH LABORATORY, INFORMATION SCIENCES OFFICE
- O AUTOMATION SCIENCES RESEARCH LABORATORY, INFORMATION SCIENCES OFFICE
- O HUMAN FACTORS RESEARCH LABORATORY (JOINT RESEARCH ROLE BEING DEFINED)
- O SIMULATION RESEARCH LABORATORY (JOINT RESEARCH ROLE RECENTLY DEFINED)

ORIGINAL PAGE IS
OF POOR QUALITY

CURRENT PROBLEMS WITH EXPERT SYSTEMS

- O MAJOR PROBLEM IS DEVELOPMENT TIME AND COST - WITH THE SPACE STATION, ADDITIONAL PROBLEM IS THE HIGH RELIABILITY REQUIRED
- O LACK OF TRAINED PEOPLE - SEVERAL UNTRAINED "EXPERTS" CURRENTLY IN EXISTENCE
- O DEVELOPMENT OF SUITABLE REPRESENTATIONS FOR EACH DOMAIN
  - O BETTER HARDWARE WILL NOT HELP MUCH
  - O TRAINED AI PEOPLE REQUIRED
  - O EXPERT SYSTEMS FOR COMPLEX DOMAINS AT LEAST 15 YEARS AWAY
- O LONG-TERM PERFORMANCE LIMITATIONS
  - O LACK OF ABILITY TO LEARN
  - O DIFFICULTY OF DOMAIN KNOWLEDGE REPRESENTATION
  - O TEMPORAL AND GEOMETRICAL REASONING ABILITY
- O INADEQUATE LONG-TERM RESEARCH PROGRAM/FUNDING AND TRAINED KNOWLEDGE ENGINEERS
  - O PROVIDE INCREASED FUNDING FOR STUDENTS AT MAJOR AI UNIVERSITIES
  - O PROVIDE COOPERATIVE RESEARCH ENVIRONMENT FOR TRAINING OF IN-HOUSE PERSONNEL
  - O PROVIDE FUNDS FOR RESEARCH IN EXPERT SYSTEMS
  - O PROVIDE SUPPORT FOR PROVEN EXPERT SYSTEM DEVELOPMENT TEAMS

PRESENT LIMITATIONS OF EXPERT SYSTEMS

Knowledge representation.

Reasoning.

Knowledge acquisition facilities.

Verification.

Explanation capabilities.

Metaknowledge.

Learning Capability.

3

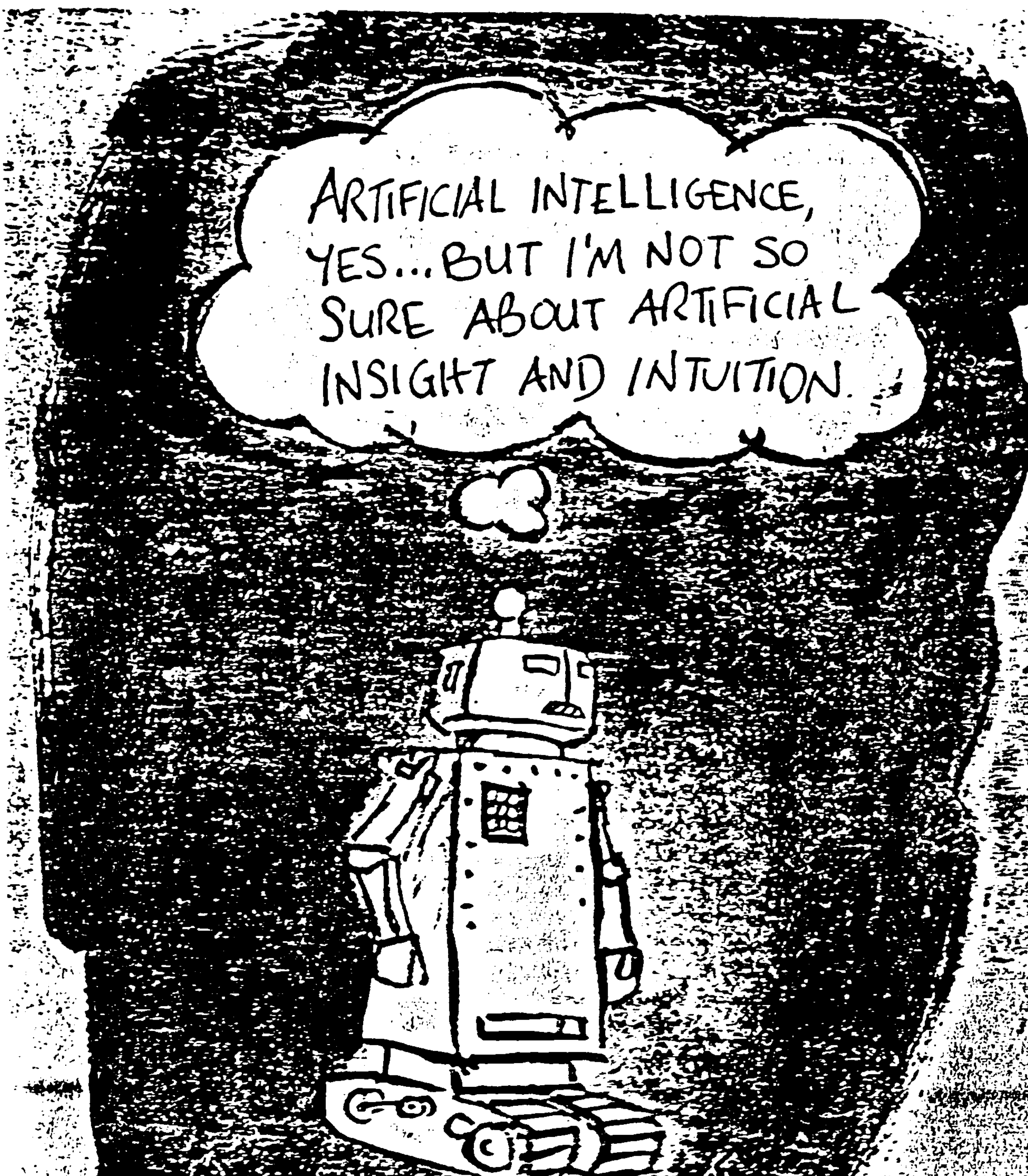
LONG RANGE TECHNOLOGY CHALLENGES

- DOMAIN-INDEPENDENT KNOWLEDGE REPRESENTATION AND MODELING
- SENSING AND INFORMATION EXTRACTION AND INTERPRETATION
- MACHINE LEARNING (INTELLIGENCE AND DECISION MAKING)
- NATURAL LANGUAGE INTERFACE
- AI PROGRAMMING LANGUAGES
- INTEGRATION AND APPLICATIONS
  - REMOTE
  - IN-SITU

ORIGINAL PAGE IS
OF POOR QUALITY

SYSTEM OPERATIONS

	1986	1987	1988	1989	1990	1991	1992	1993
	All Systems must be Extensible, Opportunistic, Explanatory	Single, Isolated Subsystems Interactive, Standard Maintenance Procedure Problems (approx 80% of problems)	Multiple Subsystems Standard Maintenance Procedures Multiple Faults (approx 80% of problems)	Real time, fault tolerant systems, failed actions, handle Inaccurate Information, Errors	Single Subsystem Almost fully automatic (95% of problems) Advanced Explanations Errorful data		Multiple Subsystems Cooperation Synchronization Fully Automatic (99%) Errorful data Advanced Explanation	
DEMOS:		RCS Life Support Power →	2 Subsystems (Interaction Simulated) →	2 Subsystems →		2 Subsystems →		2 Subsystems (Combined)
RESEARCH:				Probabilistic Reasoning				
	Reasoning about Action and Process		Representation of Procedures			Reasoning about Deadlock/Cooperation/Communication		
		Interaction Concurrency	Synchronization Scheduling			Integration with Quantitative Reasoning Techniques		
			Qualitative Reasoning (Physics, Mechanics)					
		Distributed Reasoning						
	Frame Problem/Effects of Actions					3D Space and Time		
	Single Agent Planning	Multi-Agent Planning		Planning from Knowledge of Structure and Mechanism				
		Truth Maintenance						
		Casualty, Other Events						
INTERFACE:		Graphics, Video, Natural Language, Speech		e.g., Graphical display of procedures, systems (multi-level); explanation at various levels; causal explanations; speech				
		Fast Data-Base Access and Update						
SUPPORTING:		Fast Theorem Provers						
			Highly Parallel Systems					
	CAD/CAM Documentation Techniques							
		Manual Verification Techniques		Automatic Verification Techniques				
	Knowledge-Acquisition Techniques							
		Space-Rated Symbolic Processors						


ARTIFICIAL INTELLIGENCE, YES... BUT I'M NOT SO SURE ABOUT ARTIFICIAL INSIGHT AND INTUITION.

25.

Project: Knowledge Representation and Knowledge Acquisition

Participants: Henry Lum, SI
Claire Wolfe, SI
Bruce G. Buchanan, Stanford University

Status: Work in progress
Grant

Project Description:

By working on PROTEAN, an expert system which determines the three-dimensional molecular structure of a substance from NMR (nuclear magnetic resonance) data, the blackboard model BB1, which is a framework for knowledge representation and control, is examined.

Reference:

Clancey, W.J.: "Acquiring, Representing, and Evaluating a Competency Model of Diagnostic Strategy", Stanford HPP Report 84-2, February 1984.

26.

Project: Reasoning With Uncertainty

Participants: Henry Lum, SI (grant monitor)
Lotfi Zadeh, U.C. Berkeley

Status: Work in progress
Grant

Project Description:

"Fuzzy Logic" is one way to handle uncertainty in reasoning. This project focuses on the further development of fuzzy logic.

References:

Zadeh, L.A.: "A Computational Theory of Dispositions", Proceedings of the 1984 International Conference of the Association for Computational Linguistics.

Zadeh, L.A.: "Test-Score Semantics as a Basis for a Computational Approach to the Representation of Meaning", U.C. Berkeley Memorandum No. UCB/ERL M84/8, January 1984.

27.

Project: Fuzzy Rule-Making for Failure Detection and Expert Systems

Participants: Henry Lum, SI (grant monitor)
Tom Sheridan, MIT

Status: Work in progress
Grant

Project Description:

This project will investigate the use of fuzzy logic in diagnosis of failures in complex space systems.

28.

Project: System Procedural Knowledge Engineering Tools

Participants: Henry Lum, SI
Mike Georgeff, SRI

Status: Work in progress
Contract

Project Description:

"Active intelligent systems need to be able to represent and reason about actions and how those actions can be combined to achieve given goals. This knowledge is often in the form of sequences of actions or procedures for achieving given goals or reacting to certain situations.

[In the tools being developed] the knowledge representation has a declarative semantics that provides for incremental changes to the system, rich explanatory capabilities, and verifiability. The scheme also provides a mechanism for reasoning about the use of this knowledge, thus enabling the system to choose effectively between alternative courses of action."

Reference:

Georgeff, M.: "Development of an Expert System for Representing Procedural Knowledge", contract report, December 1984.

29.

Project: Information Understanding

Participants: Henry Lum, SI (grant monitor)
Richard Volz, U. of Michigan

Status: Work in progress
Grant

Project Description:

Under a grant to the University of Michigan, Richard Volz is investigating the integration and fusion of sensor information for use by expert systems. It is hoped that this research will contribute to space-borne robotics applications.

3Ø.

Project: Symbolic Processor Architectures

Participants: Henry Lum, SI (grant monitor)
Edward A. Feigenbaum, Stanford University

Status: Work in progress
Grant

Project Description:

This long-term project undertakes to develop a symbolic processor architecture which can equal or surpass a fifth generation computer in performance.

References:

Long, C.: "Framework for Circuit Design", Stanford HPP Report 83-45, December 1983.

Dietterich, T.G.: "Learning About Systems that Contain State Variables", Stanford HPP Report 84-1Ø, May 1984.

ORIGINAL PAGE IS OF POOR QUALITY

31.

Project: Inference Engine Evaluation

Participants: Claire Wolfe, SI
Rafael Villegas, SI
Vivian Frederick, De Anza College

Status: Work in progress
In-house

Project Description:

"The strengths and weaknesses of several existing inference engines (such as MRS, EMYCIN, AGE, OPS5) are being evaluated for various types of applications and appropriate inference engines are being maintained for use throughout NASA. An intra-agency class will be conducted this summer (1985) to familiarize people with the Symbolics Lisp Machine and available expert system building tools."

32.

Project: Robotics Perception Laboratory

Participants: Scott Starks, University of Texas
Veena Bhatia, SI
Harold Fujii, SI
Rajiv Mehta, FSN

Status: Work in progress
In-house

Project Description:

An experimental robotics laboratory is being established to test various sensor hardware and software for robot perception systems. Laboratory and educational robots are being acquired to equip the lab. This work will contribute information needed for determining requirements and specifications for the SMART program (see applications project 9).

33.

Project: Concept Design of Intelligent Iconic Processors

Participants: Wun Chiou, SI
Carolyn Banda, SI (Informatics)

Status: Planning stage
In-house

Project Description:

The goal of this project is to develop a translation between symbolic representation of information and representation by icons. This work will be done on a Symbolics, and will make use of Easy Graph, a Lisp tool developed in-house to draw polygons and various other geometric figures.

34.

Project: Knowledge Representation of an Executive Expert System Controller

Participants: Wun Chiou, SI
Bruce G. Buchanan, Stanford University

Status: Planning stage
Grant, in-house

Project Description:

This project will undertake an in-depth look at the knowledge representation involved in the control of the space station expert sub-systems.

STATE OF THE TECHNOLOGY FOR INTELLIGENT AIDING IN THE COCKPIT
(CONTINUED)

TECHNICAL AREA	DESIRED/REQUIRED CAPABILITIES	CURRENT CAPABILITIES
O ARTIFICIAL INTELLIGENCE		
– PROBLEM SOLVING/ PLANNING	DYNAMICALLY CHANGING GOALS, CONDITIONS, OBJECTS, AND PROPERTIES	WELL–DEFINED, FIXED GOALS, CONDITIONS, OBJECTS, AND PROPERTIES
	MULTIPLE AGENTS	SINGLE AGENT
	SIMULTANEOUS AND OVERLAPPING EVENTS	NON–OVERLAPPING EVENTS
	TEMPORAL RELATIONS	
	PLAN EXECUTION MONITORS	
	INCREMENTAL PLANNERS	

STATE OF THE TECHNOLOGY FOR INTELLIGENT AIDING IN THE COCKPIT
(CONTINUED)

TECHNICAL AREA	DESIRED/REQUIRED CAPABILITIES	CURRENT CAPABILITIES
O ARTIFICIAL INTELLIGENCE		
– KNOWLEDGE REPRESENTATION	REPRESENTATIONS FOR TEMPORAL, SPATIAL, QUALITATIVE, DEFAULT, FUNCTIONAL, STRUCTURAL, AND ANALOGICAL KNOWLEDGE	LIMITED EXPRESSIBILITY

STATE OF THE TECHNOLOGY FOR INTELLIGENT AIDING IN THE COCKPIT

Reference: LaRC, 4/16/85

TECHNICAL AREA	DESIRED/REQUIRED CAPABILITIES	CURRENT CAPABILITIES
O ARTIFICIAL INTELLIGENCE		
– EXPERT SYSTEMS	REAL TIME REASONING ABOUT DYNAMICALLY CHANGING ENVIRONMENT AND TIME–BASED INFORMATION	NON–REAL TIME REASONING ABOUT STATIC SITUATION
	RIGOROUS METHODS FOR DEALING WITH UNCERTAINTY	LIMITED CAPABILITY FOR DEALING WITH UNCERTAIN, INCOMPLETE, OR INCONSISTENT INFORMATION
	MODEL–BASED SYSTEM +	RULE–BASED SYSTEMS
	EFFICIENT CONTROL STRUCTURES FOR DEALING WITH MULTIPLE REPRESENTATIONS	LIMITED CONTROL STRUCTURES
	EXTENSIVE EXPLANATION CAPABILITY AS NEEDED	LIMITED EXPLANATION CAPABILITY
	HYBRID REASONING ABOUT SYMBOLIC AND NUMERIC INFORMATION	SYMBOLIC COMPUTATION
	MULTIPLE COOPERATING INTELLIGENT SYSTEMS	SINGLE EXPERT

STATE OF THE TECHNOLOGY FOR INTELLIGENT AIDING IN THE COCKPIT

(CONCLUDED)

TECHNICAL AREA	DESIRED/REQUIRED CAPABILITIES	CURRENT CAPABILITIES
O SPEECH UNDERSTANDING	1000 WORD VOCABULARY, CONNECTED SPEECH, NATURAL LANGUAGE	100 WORDS, RESTRICTED SPEECH
O COMPUTER HARDWARE	PARALLEL OPERATIONS	SEQUENTIAL OPERATIONS
O CREW INTERFACE	CREW INFORMATION REQUIREMENTS BY FUNCTION	NON-FLEXIBLE & LIMITED CAPABILITY
	ADAPTIVE AIDING	
	MULTIPLE INTERFACE MEDIA	
	NATURAL HUMAN-LIKE COMMUNICATION	

N87-29140

p. 21

AI at AMES

Artificial Intelligence Research and Application
at NASA Ames Research Center
Moffett Field, California
February 1985

edited by Alison E. Andrews
Applied Computational Aerodynamics Branch
STA

AI AT AMES

Rumors of artificial intelligence (AI) activity at NASA Ames have been floating around for some time. To the relief of those of us here at Ames attempting to work in AI, those rumors have not been greatly exaggerated, and, in fact, the discovery of such a quantity and diversity of planned and on-going projects exceeded all expectations. To promote cooperation among scientists and better-informed planning in management, this document was conceived and prepared. Several months were required to gather and organize the project descriptions contained herein. It is hoped that the effort expended initially will make future updates easier to prepare.

This document contains three principal types of information, in this order:

1. charts of function versus domain for AI applications and interests, and research area versus project number for AI research

2. a list of project titles with associated project numbers and page numbers

3. project descriptions, including title, participants, and status

In particular, the chart for AI applications and interests is an attempt to present information in such a way that

1. function commonality across domains is shown

2. the amount and type of work in a particular domain is seen at a glance

3. blank regions are easily detected and evaluated for possible future work

Feedback on this presentation is requested and will be appreciated.

Many thanks to all of the scientists and engineers at Ames who contributed project descriptions. Their work made this document necessary - their patience, help, and encouragement made it possible.

Alison Andrews

ORIGINAL PAGE IS
OF POOR QUALITY

AI AT AMES

Domain / Function*	AI Applications								AI Interests
	Life Science		Space Science		Earth Resources		Aeronautics		
	Human Factors	Life Sup. /Medicine	Space Station	Astro- nomy	Atmosph. Research	Remote Sensing	Aircraft Systems	CFD	
Prediction	1, 2, 3						18		43
Interpretation				10	14	15,16	18,19, 20		
Diagnosis	5, 6	8					19,22		44
Design							17,18	23,24	45
Planning				11,12, 13			18,19		
Monitoring	1, 2, 5						18,19, 20		
Control	3, 5	7	9				18,19, 20,21		
Debugging	5, 6						22		
Repair									
Instruction	4								46
Advice	5, 6								
Natural Language	2						18		
Machine Vision		7	9		14	16	18		
Expert System Tools									
Robotics		7	9						44,46, 47
Learning	3, 4								46

* For function descriptions, see reverse side

The following function categories and their descriptions are from Table 1.1, p.14 of Building Expert Systems, edited by F. Hayes-Roth, D. Waterman, and D. Lenat.

- PREDICTION - Inferring likely consequences of given situations (eg. military forecasting, traffic prediction, weather)
- INTERPRETATION - Inferring situation descriptions from sensor data (eg. speech understanding, image analysis, signal interpretation)
- DIAGNOSIS - Inferring system malfunctions from observables (eg. medical, electronic, mechanical, software)
- DESIGN - Configuring objects under constraints (eg. circuit layout, building, budgeting)
- PLANNING - Designing actions (eg. robot, route, communication, military, project)
- MONITORING - Comparing observations to plan vulnerabilities (eg. nuclear power plant, air traffic, disease)
- CONTROL - Interpreting, predicting, repairing, and monitoring system behaviors (eg. air traffic control, battle management, mission control)
- DEBUGGING - Prescribing remedies for malfunctions [planning, design, prediction] (eg. intelligent knowledge base and text editors)
- REPAIR - Executing a plan to administer a prescribed remedy (eg. automotive, avionic, and computer maintenance)
- INSTRUCTION - Diagnosing, debugging, and repairing student behavior (eg. intelligent computer-aided instruction)

ORIGINAL PAGE IS
OF POOR QUALITY

AI AT AMES

AI Research																				
Research Area	Projects																			
	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40				
Human Perception, Cognition											X	X		X						
Human-Machine Interaction									X			X	X	X						
Robotics								X												
Machine Vision, Perception								X												
Machine Architectures						X														
Info. Extraction, Understanding					X															
Reasoning		X	X												X					
Knowledge Representation	X	X	X	X					X	X					X	X				
Planning and Problem Solving														X						
Learning																				
Natural Language														X		X				
Automatic Programming																				
Expert System Building Tools	X			X			X								X					

Category	AI Books
General AI	41
Expert Systems	42

AI AT AMES

AI AT AMES

AI APPLICATIONS

****************************** LIFE SCIENCE ******************************

HUMAN FACTORS

1.

Project: Automatic Detection of Procedural Errors

Participants: Everett Palmer, LH (grant monitor)
John Hammer, Georgia Tech

Status: Work in progress
Grant

Project Description:

"Under a grant to Georgia Tech, Dr. John Hammer has developed a computer program (PLANE) which can automatically detect procedural errors made by an aircraft crew. The program is based on Schank's [Roger Schank of Yale] concept of scripts. The flight is structured as a four-level tree consisting of mission, flight phases, procedures and actions. Each node in the tree can be in one of four states (unstarted, inprogress, done, or aborted). As actions occur during the mission, the state of the tree is updated and erroneous actions are called out. During the coming year this program will be applied to data from a full mission simulation in the B727 simulator at Ames."

Reference:

Hammer, John : "An Intelligent Flight Management Aid for Procedure Execution", IEEE Transactions on Systems, Man and Cybernetics, in press.

2.

Project: SMART CHECKLIST for Aircraft and Spacecraft

Participants: Everett Palmer, LH
Ed Lee, LH (Informatics)

Status: Work in progress
In-house

Project Description:

"At Ames we are incorporating concepts from John Hammer's program into the design of a SMART CHECKLIST program. The goal of the SMART CHECKLIST is to aid the operator of aircraft and space craft systems in executing procedures. The program will also monitor for procedural errors made by the operator. This project will combine concepts from rule based models of human behavior, AI theories of story understanding, and syntax directed editors used in software development environments. The space shuttle payload that we have chosen to investigate is the Orbiting Refueling System (ORS)."

3.

Project: Army Aircrew-Aircraft Integration Program (A^3I)

Participants: Irving Statler, Y
David Nagel, LH
Earl J. Hartzell, LHAC

Status: Work in progress
In-house, contract

Project Description:

"[This] activity is a new one that the Aeromechanics Laboratory has undertaken jointly with NASA's Aerospace Human Factors Research Division. This ambitious endeavor, called the Army Aircrew-Aircraft Integration (A^3I) Program, will attempt to develop a validated methodology for predicting the pilot behavior in a single-seat scout/attack helicopter. The model is intended to be used for helping engineers account for the design of cockpits and training systems. We have initiated some discussions with representatives from Stanford University about the possibility of developing an 'expert system' that might be useful as a checkpoint in the process of validating the human model. In the meantime, we may also be developing some expert systems for controlling opponent aircraft in simulations of air-to-air combat."

4.

Project: Expert System for Pilot Training (A^3I)

Participants: Wun Chiou, SI
William Gevarter, SI
Bruce G. Buchanan, Stanford University

Status: Work in progress
In-house, grant

Project Description:

This project involves the development of an expert system which is capable of transferring an instructor pilot's knowledge to a student pilot. The system must be able to learn in some sense, so that a model of the student's knowledge can be formed.

5.

Project: Plant Control and Failure Diagnosis

Participants: Everett Palmer, LH (grant monitor)
Bill Rouse, Georgia Tech
Annette Knaeuper, Georgia Tech

Status: Work in progress
Grant

Project Description:

"Under a research grant to Georgia Tech, Dr. Bill Rouse and Annette Knaeuper have developed a rule based model to describe the behavior of the operator of a generic process control plant. The model, named KARL, controls the plant and diagnoses plant failures. The design goal of KARL was to develop a model that could balance the competing demands of controlling the plant with needs to cope with failures. During the past year they have modified the model so that it provides advice to the plant operator. During the coming year this model will be extended to the description of operators controlling space shuttle payloads."

6.

Project: ORS Operator Advisor

Participants: Guy Boy, LH (ONERA-CERT-DERA)
Will Taylor, LH (Informatics)

Status: Work in progress
In-house

Project Description:

"Dr. Guy Boy from ONERA-CERT-DERA in Toulouse, France is visiting our group [LH] for a year. He has developed rule based models to model crew performance in aircraft cockpits. His project at Ames will be to develop an expert system (HORSES) [Human-Orbital Refueling System Expert System] to advise the operator of the Orbiting Refueling System during system malfunctions. He is basing this system on a system he developed in France to aid the operator of a geosynchronous satellite. He has a first version of the HORSES system operating on our [LH] VAX 750. It is programmed in Franz Lisp."

LIFE SUPPORT / MEDICINE

7.

Project: Automatic Control of Closed Environment Life Support Systems (CELSS)

Participants: Silvano Colombano, LXL

Status: Planning stage
Some in-house, most probably contract

Project Description:

"A typical operational CELSS module, as presently envisioned, will provide life support, at least in part, to a crew of astronauts, in the form of breathable air and food. It is not envisioned, however, that the crew should spend a substantial amount of their time tending the system. The ideal CELSS will be largely self-sufficient, will recognize and maintain the state or sequence of states that are necessary to fulfill its function. It will detect problems, attempt to provide a diagnosis, select a course of action for self-repair, or at least suggest one. The reason for this is that the system will certainly be extremely complex, and the crew cannot be expected to have mastered its intricacies.

For self-regulation this system will need software that integrates aspects of AI, simulation and process control. It will have to be completely reliable and run on equally reliable and compact hardware.

While still keeping the need for increasing integration in mind, separate areas of software development have been listed: Simulation, Expert Systems, Computer Vision, Robotics, Computer Aided Design and Computer Aided Process Control."

8.

Project: Aerospace Medicine

Participants: James Stevenson, LH

Status: Planning stage
In-house

Project Description:

With a background in experimental psychology and biostatistics, Dr. Stevenson is exploring the possible use of expert systems in aerospace medicine.

SPACE STATION

ORIGINAL PAGE IS
OF POOR QUALITY

9.

Project: SMART Program (Space Missions for Automation and Robotics Technologies)

Participants: Henry Lum, SI

Status: Planning stage
In-house

Project Description:

"The SMART Program is proposed as a multi-flight Shuttle-based Automation and Robotics Test Facility for the validation of advanced robotics, automation, and telepresence technologies and real-time operational concepts. The Facility will offer the potential for evaluating 'standard' operational environments for robotics and automation applications. Technology and the over-all system capability will be upgraded periodically by the Ames Cooperative Research Team consisting of researchers from Ames, industry, and academia.

The necessity of a space flight is dictated by the constraints imposed by space such as remoteness, zero-G, and 'hostile' environment. It is planned that all Facility experiments will be evaluated prior to flight in NASA's ground test facilities such as MSFC's Neutral Buoyancy Facility. This procedure will maximize the potential for a successful flight. Complexity of the flight experiments will progress from RMS-attached robots to supervisory tethered robots to free-flying multiple robots working in a controlled distributed environment towards a common work objective. Telepresence mode is feasible for the latter."

ASTRONOMY

10.

Project: Infrared Data Examination and Analysis Expert System

Participants: Paul Swan, SST
Jeffrey Scargle, SST
William Likens, LXR
Robert Mecklenburg, SST (Informatics)
Gary Villere, SST (Informatics)
Mike Werner, SSA

Status: Work in progress
In-house

Project Description:

"The primary effort of this project has been in the conceptual design of an Infrared Data Expert Assistant (IDEA) computer program which would utilize expert system techniques to assist the infra-red astronomer in the examination and analysis of the very large set of data which is now being made available from the IRAS program."

The program will incorporate:

- Command Language Interface ("being developed by using the LEX and

YACC programs provided as part of the UNIX operating system on the VAX")
- Data Base Management System ("...has been implemented by buying a commercial data base management system available for the VAX computer, and entering into it the data from IRAS, as well as astronomical catalogs obtained on magnetic tape from the Goddard Space Flight Center.")
- Expert Infrared Astronomy Knowledge ("...a set of rules of the form IF...THEN... concerning interpretation of the IRAS data, as well as two specific search procedures for solving the problem of identification and classification of the IRAS point sources.")

11.

Project:	Kuiper Airborne Observatory Flight Planner
Participants:	Philip Nachtsheim, SI John Stutz, SI (Informatics) Carolyn Banda, SI (Informatics) William Gevarter, SI
Status:	Work nearly completed In-house

Project Description:

A computer program named KAOS (Kuiper Airborne Observatory Scheduler) has been developed to perform flight planning. "The basic mode of operation of KAOS is generate and test, that is, a leg is calculated for the observation of an object and then tested. Typically an observation is rejected if (1) the object is outside of the window, (2) the leg overflies a restricted zone, (3) the leg overflies a warning zone, or (4) the completion of a leg leads to a point out of range. These rules can be relaxed by the user, and other rules can be added. In a sense, the system can be 'trained'."

"The essential features of the expert system are the data base, rule base, and inference engine:
- data base: catalog of objects and ephemeris data
  geographical locations of warning and restricted zones
  flight departure points.
- rule base: if...then... (or situation-action) rules
- inference engine: MRS, or Meta-level Reasoning System, written in LISP and developed at Stanford University"

Reference:

Banda, C., Stutz, J., Nachtsheim, P., Gevarter, W.: "User's Guide for KAOS", November 1984.

12.

Project:	Halley's Comet Observation Scheduler
Participants:	John Stutz, SI (Informatics) Wun Chiou, SI
Status:	Work nearly completed In-house

Project Description:

This expert system schedules observations in the KAO for the fly-by of Halley's Comet. The system is an extension of the KAOS system (described previously). The similarity of the application allows the use of the KAOS structure with only minor modifications. The knowledge base has been modified for the new application.

13.

ORIGINAL PAGE IS
OF POOR QUALITY

Project: SIRTF Scheduler

Participants: Philip Nachtsheim, SI

Status: Planning stage
In-house

Project Description:

This project will develop an expert system to schedule the observations for the Shuttle Infrared Telescope Facility. The KAOS system will serve as the foundation for this system.

****************************** EARTH RESOURCES ******************************

ATMOSPHERIC RESEARCH

14.

Project: Aerosol Particle Image Classifier

Participants: Libby Netland, SI
Scott Starks, University of Texas

Status: Work in progress
In-house

Project Description:

"[The Aerosol Particle System (APS) is] a developmental system which will use expert system building techniques to automate the classification of sulfuric acid particles. This will function as an aid to stratospheric aerosol research. Each month, a U-2 aircraft collects sulfuric acid samples by exposing palladium wires to the stratosphere at 70,000 ft. elevation. These wires are then photographed under a Scanning Electron Microscope. The photographs are examined by one of the aerosol researchers, and the sulfuric acid particles are counted and sized. Since the photographs are taken at a high magnification, flaws and dark spots appear which must be discriminated from the sulfuric acid particles. For this reason, traditional particle counting systems are inadequate.

The APS software uses similar isolation techniques and numerical feature measurements as those used in traditional Image Classifiers. However, once the feature measurements are known, they are related symbolically rather than numerically to choose the correct classification for the unknown object. The APS data base contains rules about predefined objects instead of numerical statistics about the objects. One advantage of having a symbolic rather than numeric representation is that the system can be easily modified by adding rules."

Reference:

Starks, S., Netland, E., and Elizandro, D.: "Analysis of Aerosol Particles by Digital Image Processing Techniques", 1984 ASEE Annual Conference Proceedings, p. 1335.

15.

Project: Remote Sensing Image Classification

Participants: William Likens, LXR
William Erickson, SI
Steve Engle, LXR (Informatics)

Status: Work in progress
In-house

Project Description:

"The problem selected for this demonstration is that of automated labeling by land cover of spectral clusters developed through unsupervised clustering of Landsat Multi-Spectral Scanner (MSS) imagery. The prototype is to label spectral clusters for Level I information; namely, urban, forest, range, water, barren, and 'other'. This problem is well suited for expert system analysis in that, 1) the set of possible solutions is clearly defined, and 2) the problem is well understood and routinely solved by image analysis professionals.

Inputs to the prototype system will consist not only of MSS spectral statistics, but also collateral data. These collateral data may include digital terrain, zoning, prior land cover maps, and other data. Data are to be statistically summarized when input into the system in order to reduce their volume and complexity.

The concept for the mature system emphasizes minimal inputs from the user with an emphasis upon use of a dedicated expert system processor, such as a Xerox 11Ø8, linked to image analysis functions on a larger general purppose computer, such as a DEC VAX 11/78Ø."

Reference:

Erickson, W.K., and Likens, W.C.: "An Application of Expert Systems Technology to Remotely Sensed Image Analysis", Proceedings of the Pecora 9 Symposium, Spatial Information Technologies for Remote Sensing Today and Tomorrow. IEEE, October 1984.

16.

Project: Image-Based Geological Exploration

Participants: Wun Chiou, SI

Status: Work completed
In-house

Project Description:

An image-based expert system for geological exploration was developed by Wun Chiou while he was at JPL.

Reference:

Chiou, W.: "System Architecture of a Remote Sensing Expert System", Proceedings of the IEEE Computer Society Workshop on Computer Architectures for Pattern Analysis and Image Data Base Management, p243-25Ø.

AIRCRAFT SYSTEMS

17.

Project: Conceptual/Preliminary Design of Aircraft

Participants: George Kidwell, FHS
Megan Eskey, FHP

Status: Work in progress
In-house, possible contracts

Project Description:

"The ultimate goal of this project is the development of an integrated system capable of greatly assisting the advanced aircraft designer in achieving the optimal configuration based on requirements, constraints, and available technologies. Such a system will necessarily be a hybrid using many computer programming and analytical techniques, such as knowledge-based expert systems, numerical optimization, parametric approximation, design synthesis, and several levels of analysis algorithms. The driving rationale behind this project is based on the increasingly divergent design goals of high performance (speed, maneuverability, revenue, quietness, etc.) and economy (cost, maintenance, size, noise, etc.). In other words, a globally-optimal solution is required, but it is subject to many diverse constraints and must be found in a very large design space.

The general concept is to employ a knowledge-based expert system to act as the design expert and executive for the network of methods. At the next level would be a numerical optimization program to deal with those variables that are continuous and finite. Beneath this would be a heirarchy of synthesis and analysis programs that would be selected at appropriate times by the expert system. A database manager is necessary to interface with the user and the many diverse elements of the system. Finally, such a system must be able to network automatically (or semi-automatically) among the most appropriate computers for the analysis currently being used."

18.

Project: Single Crew Scout/Attack Helicopter

Participants: Irving Statler, Y
David Key, YC
Loran Haworth, YC
Edwin Aiken, FSDC
Earl J. Hartzell, LHAC
James Voorhees, LHA

Status: Planning stage
In-house

Project Description:

"The Aircrew-Aircraft Systems Division of the Army Aeromechanics laboratory is interested in applications of artificial intelligence which would make feasible the fielding of a single-crew combat helicopter. Applications of expert systems, natural language processing, computer vision, and problem solving and planning to both the flight path control and mission management crew functions are required to reduce the single pilot's workload to a reasonable level in the combat environment. The Aircrew-Aircraft Systems Division includes 'experts' in the fields of flight control/handling qualities and engineering psychology/human factors."

ORIGINAL PAGE IS
OF POOR QUALITY

19.

Project: Real-Time Applications of Expert Systems

Participants: Lee Duke, OFDC
Victoria Regenie, OFDC

Status: Work in progress
In-house, contract, university grant

Project Description:

"The cockpit of a modern high performance aircraft is an exceedingly complex interface between a human pilot with limited resources and an avionics system and aircraft of almost limitless complexity. The pilot is often required to perform multi-item tasks, using mutiple controllers in conditions of high stress and extreme danger while monitoring the health and status of an avionics system consisting of numerous sensors, weapons, and computers. The pilot is virtually saturated visually and manually. The problem facing the systems engineer is either to simplify the system or simplify the interface between the pilot and the vehicle.

The ultimate goal of the Ames Dryden research and development in expert systems is the intent driven cockpit. The research and development leading to this goal has been partitioned into several areas to allow an orderly development process with intermediate goals and useful products.

- Flight System Monitoring Functions
- Expert Control Systems
- Pilot's Associate

The Ames Dryden research and development in real-time applications of expert systems is intended to concentrate on applications rather than basic research in expert systems. However, the goal of this activity is both to develop useful expert systems and to focus on fundamental issues that can be pursued by others involved in basic research.

Two sub-activities are being pursued, both under the auspices of the Pilot's Associate segment of DARPA's strategic computing initiative. The first is an expert flight systems monitor for the X-29A Forward Swept Wing Aircraft. The second is a closed-loop, goal seeking system."

Reference:

Regenie, V.A., and Duke, E.L.: "Expert Systems Applications to Failure Modes and Affects Analysis, Testing and Real-Time Monitoring", NASA Ames Dryden Flight Research Facility X-29A Internal Document X-84-ØØ7, June 1984.

2Ø.

Project: Advanced Integrated Flight Control System Study

Participants: Dale Mackall, OFM (contract monitor)
Boeing Military Airplane Company

Status: Work in progress
Contract

Project Description:

"The study is to provide a preliminary design for an advanced aircraft by integrating previously independent aircraft functions, such as the aerodynamics, flight control, propulsion, structures, and the pilot. The study is to examine the tools and methods available to accomplish the design and identify any deficiencies. The study will be assessing the tools and methods to be used for the design of the pilot-vehicle interface. Boeing is applying their work from the 'Avionics Expert Systems Definition Study' to this task.

A small in-house effort was accomplished to obtain a basic understanding of how AI techniques can be applied to the pilot interface. The emergency procedures for the electrical and flight control system of a fighter aircraft were implemented into an 'expert pilot assistant' demonstration using an IBM PC. A NASA TM is in editorial, and a demonstration disk is available."

21.

ORIGINAL PAGE IS
OF POOR QUALITY

Project: Advanced Guidance Systems for Aircraft

Participants: Heinz Erzberger, FSN

Status: Work in progress
In-house

Project Description:

"In the Aircraft Guidance and Navigation Branch we are applying AI techniques to the design of advanced guidance systems for aircraft. Current projects involve developing expert based systems for automated air traffic control and air-to-air combat."

22.

Project: Verification Testing for Digital Flight Control Systems

Participants: Douglas Doane, FSN
Jeremy Saito, FSN
N. Rajan, FSN (Stanford)

Status: Planning stage
In-house

Project Description:

"The Advanced Verification and Validation Project (Digital Flight Control System Verification Laboratory) is considering the use of AI concepts to develop a verification testing methodology for digital flight control systems. Initial planning indicates that an 'expert system' approach may provide an effective mechanism."

COMPUTATIONAL FLUID DYNAMICS (CFD)

23.

Project: Automatic Zonal Grid Generation

Participants: Alison Andrews, STA
Kristin Hessenius, STA
Man Mohan Rai, STA (Informatics)

Status: Work in progress
In-house

Project Description:

"Because it is usually difficult or impossible to generate a reasonable single grid about a general three-dimensional aircraft configuration, a computational flow field is often segmented into simpler regions or zones, which are then discretized separately. Intelligent zonal grid generation requires knowledge about zone criteria, grid generation capabilities, flow solver behavior, and expected flow field features. Obtaining a good zoning is important to solution accuracy, efficiency, and ease of grid generation. With some guidelines, a CFD expert familiar with the concept of zonal methods can come up with a reasonable zoning in two dimensions by 'just looking at' the problem's configuration, qualitatively noting shapes, orientations, abrupt transitions, relative positions, and other features. However, such expertise is not widespread, and it is difficult to express. Furthermore, complicated three-dimensional problems present even experts with difficulties in the visualization and specification of zonal boundary surfaces. Hence there is a need to systematize and automate the process of obtaining and evaluating a flow field zoning.

An expert system approach is a promising one for zoning because it offers a way to encode zoning knowledge and manipulate it within a highly developmental and extensible framework. A rule-based expert system is being built using MRS, a system building tool (written in Lisp) developed at Stanford University. Work is being done on a VAX 11/780."

ORIGINAL PAGE IS
OF POOR QUALITY

24.

Project: CFD Expert System

Participants: Scott Eberhardt, STC
Rick Briggs, RIACS

Status: Planning stage
In-house

Project Description:

"The goal of this project is to implement a deep expert system which generates the grid and all necessary parameters for ARC2D (a two-dimensional Euler/Navier-Stokes flow solver) given only the geometry, along with Mach and Reynolds Numbers. The geometry will be interpreted as a combination of 'shapes' which will be abstracted from the raw data. The expert system will then work symbolically using only the shapes, which will be matched with a hierarchy of shape-frames. After the initial phase is completed, we will be examining the possibility of the expert system running autonomously in special cases, without resorting to ARC2D. If a significant percentage of problems can be handled entirely symbolically (without using a numerical code) this will open up an entirely new area of research."

25.

Project: Knowledge Representation and Knowledge Acquisition

Participants: Henry Lum, SI
Claire Wolfe, SI
Bruce G. Buchanan, Stanford University

Status: Work in progress
Grant

Project Description:

By working on PROTEAN, an expert system which determines the three-dimensional molecular structure of a substance from NMR (nuclear magnetic resonance) data, the blackboard model BB1, which is a framework for knowledge representation and control, is examined.

Reference:

Clancey, W.J.: "Acquiring, Representing, and Evaluating a Competency Model of Diagnostic Strategy", Stanford HPP Report 84-2, February 1984.

26.

Project: Reasoning With Uncertainty

Participants: Henry Lum, SI (grant monitor)
Lotfi Zadeh, U.C. Berkeley

Status: Work in progress
Grant

Project Description:

"Fuzzy Logic" is one way to handle uncertainty in reasoning. This project focuses on the further development of fuzzy logic.

References:

Zadeh, L.A.: "A Computational Theory of Dispositions", Proceedings of the 1984 International Conference of the Association for Computational Linguistics.

Zadeh, L.A.: "Test-Score Semantics as a Basis for a Computational Approach to the Representation of Meaning", U.C. Berkeley Memorandum No. UCB/ERL M84/8, January 1984.

27.

Project: Fuzzy Rule-Making for Failure Detection and Expert Systems

Participants: Henry Lum, SI (grant monitor)
Tom Sheridan, MIT

Status: Work in progress
Grant

Project Description:

This project will investigate the use of fuzzy logic in diagnosis of failures in complex space systems.

28.

Project: System Procedural Knowledge Engineering Tools

Participants: Henry Lum, SI
Mike Georgeff, SRI

Status: Work in progress
Contract

Project Description:

"Active intelligent systems need to be able to represent and reason about actions and how those actions can be combined to achieve given goals. This knowledge is often in the form of sequences of actions or procedures for achieving given goals or reacting to certain situations.

[In the tools being developed] the knowledge representation has a declarative semantics that provides for incremental changes to the system, rich explanatory capabilities, and verifiability. The scheme also provides a mechanism for reasoning about the use of this knowledge, thus enabling the system to choose effectively between alternative courses of action."

Reference:

Georgeff, M.: "Development of an Expert System for Representing Procedural Knowledge", contract report, December 1984.

29.

Project: Information Understanding

Participants: Henry Lum, SI (grant monitor)
Richard Volz, U. of Michigan

Status: Work in progress
Grant

Project Description:

Under a grant to the University of Michigan, Richard Volz is investigating the integration and fusion of sensor information for use by expert systems. It is hoped that this research will contribute to space-borne robotics applications.

30.

Project: Symbolic Processor Architectures

Participants: Henry Lum, SI (grant monitor)
Edward A. Feigenbaum, Stanford University

Status: Work in progress
Grant

Project Description:

This long-term project undertakes to develop a symbolic processor architecture which can equal or surpass a fifth generation computer in performance.

References:

Long, C.: "Framework for Circuit Design", Stanford HPP Report 83-45, December 1983.

Dietterich, T.G.: "Learning About Systems that Contain State Variables", Stanford HPP Report 84-10, May 1984.

31.

Project: Inference Engine Evaluation

Participants: Claire Wolfe, SI
Rafael Villegas, SI
Vivian Frederick, De Anza College

Status: Work in progress
In-house

Project Description:

"The strengths and weaknesses of several existing inference engines (such as MRS, EMYCIN, AGE, OPS5) are being evaluated for various types of applications and appropriate inference engines are being maintained for use throughout NASA. An intra-agency class will be conducted this summer (1985) to familiarize people with the Symbolics Lisp Machine and available expert system building tools."

32.

Project: Robotics Perception Laboratory

Participants: Scott Starks, University of Texas
Veena Bhatia, SI
Harold Fujii, SI
Rajiv Mehta, FSN

Status: Work in progress
In-house

Project Description:

An experimental robotics laboratory is being established to test various sensor hardware and software for robot perception systems. Laboratory and educational robots are being acquired to equip the lab. This work will contribute information needed for determining requirements and specifications for the SMART program (see applications project 9).

33.

Project: Concept Design of Intelligent Iconic Processors

Participants: Wun Chiou, SI
Carolyn Banda, SI (Informatics)

Status: Planning stage
In-house

Project Description:

The goal of this project is to develop a translation between symbolic representation of information and representation by icons. This work will be done on a Symbolics, and will make use of Easy Graph, a Lisp tool developed in-house to draw polygons and various other geometric figures.

34.

Project: Knowledge Representation of an Executive Expert System Controller

Participants: Wun Chiou, SI
Bruce G. Buchanan, Stanford University

Status: Planning stage
Grant, in-house

Project Description:

This project will undertake an in-depth look at the knowledge representation involved in the control of the space station expert sub-systems.

35.

Project: Computational Models of Human Vision

Participants: David Nagel, LH
Albert Ahumada, LH
Ken Nollsen, LH
Andrew Watson, LH
Larry Maloney, NRC
John Perrone, NRC
Amyjo Bilson, U. of Washington
Brian Wandell, Stanford
Jack Yellott, UC Irvine

Status: Work in progress
In-house, contract

Project Description:

"Biological vision is the only working example of a system capable of recognising objects under varying conditions. Vision is also the principal interface between the human and complex devices such as computers, cockpits, and space stations. We are constructing a computational model of the early stages in the human visual process. At present, we have working modules for processing temporal, spatial, and motion components of the visual input. Future modules will deal with stereo and color. The model is being validated by psychophysical experiments in our facilities. This model will improve our understanding of human and machine vision, and will guide the design of future displays, interfaces, and image management systems."

36.

Project: Pilot-Cockpit Interface Design

Participants: Everett Palmer, LH (grant monitor)
Ken Papp, New Mexico State
Renate Roske-Hofstrand, New Mexico State

Status: Work in progress
Grant

Project Description:

"Under a grant to New Mexico State, Dr. Ken Papp and Ms. Renate Roske-Hofstrand are investigating features of interface design which will maximize the compatibility between the cognitive structures of pilots and the way information is organized for retrieval in the computer. They have developed a methodology based on link weighted networks which derives a network representation of how a pilot's knowledge about a specific subject domain is organized. They are applying this approach to the Flight Management System's Control Display Unit in the Advanced Concepts Flight Simulator at Ames."

37.

Project: Information Processing Models

Participants: Everett Palmer, LH (grant monitor)
Misha Pavel, Stanford University
Stuart Card, Stanford University

Status: Work in progress
Grant

Project Description:

"We also have a grant with Drs. Misha Pavel and Stuart Card in the Psychology department at Stanford. The objective of their grant is to develop design principles for the use of displays with multiple windows. They have used production system like models to describe the information processing operators do when working with a computer. They are using an IRIS graphics system connected to a XEROX LISP machine."

ORIGINAL PAGE IS
OF POOR QUALITY

38.

Project: Aircrew Cockpit Communications

Participants: Miles Murphy, LH (contract monitor)
Joseph Goguen, Structural Semantics
Charlotte Linde, Structural Semantics

Status: Work in progress
Contract

Project Description:

"Under a contract with Structural Semantics, Drs. Joseph Goguen and Charlotte Linde have analyzed aircrew cockpit communications as linguistic phenomena and developed formalisms showing 'planning' and 'explaining' to be structured discourse types, described by formal rules. These formalisms are integrated with those describing the other most important discourse type within the cockpit: the command and control speech act chain. Command and control discourse is described as a sequence of speech acts for making requests (including orders and suggestions), for making reports, for supporting or challenging statements, and for acknowledging previous speech acts.

This work can have application to expert system interface, goal, and rule design - whether to ensure natural crew interaction (for near-term systems), or possible system understanding of crew communications (for very advanced systems). The formalisms are being further refined on data from in-house full mission simulations."

References:

Goguen, J., Linde, C., and Murphy, M.: "Crew Communication as a Factor in Aviation Accidents". In Hartzell, E.J., and Hart, S. (eds) "Papers from the 20th Annual Conference on Manual Control", NASA Ames Research Center, 1984.

Goguen, J. and Linde, C.: "Linguistic Methodology for the Analysis of Aviation Accidents", NASA CR 3741, 1983.

39.

Project: Expert Systems Research

Participants: Rick Briggs, RIACS

Status: Work in progress
In-house

Project Description:

"A new model of expert systems is being developed which moves away from current rule-based approaches. The rule-based system is viewed as incapable of encoding higher level cognitive processes of experts. The deep expert system has been proposed, in which the lowest level is an EMYCIN-like rule-based system, but which has in addition a series of levels of abstraction which include analogical methods and deep abstract methodologies which solve from 'first principles' rather than doing manipulation of rules on the basis of the syntax of the rules. Rules are considered compiled pieces of knowledge which are used for convenience. A system of semantic primitives has been developed which is used at the 'deepest' level of the system. A deep expert system is currently being developed in the domain of CFD."

40.

ORIGINAL PAGE IS OF POOR QUALITY

Project: Natural Language Research

Participants: Rick Briggs, RIACS

Status: Work in progress
In-house

Project Description:

"A program is being written to 'understand' (i.e. represent at a level equivalently deep to the representation which human beings create when they hear or read natural language) simple natural language sentences. Sentences are broken down into very deep representations expressed entirely in a set of semantic primitives, and a method of inferencing is used which allows the program to expect further types of sentences in further discourse."

Also being studied is Sastric Sanskrit, "an obscure untranslated branch which functions as an unambiguous automatic inference-generating knowledge representation language." It is a "remarkable natural language which is ambiguity and syntax free. The equivalence of the language to semantic nets has been demonstrated in 'Knowledge and Representation in Sanskrit and Artificial Intelligence', by Rick Briggs, to appear Spring 85 in AI Magazine."

41.

William B. Gevarter, SI

"Intelligent Machines: An Introductory Perspective of Artificial Intelligence and Robotics"

1985, Prentice-Hall, Inc., Englewood Cliffs, N.J.

"This book provides an easy-to-understand, integrated view of the many diverse aspects of the fields of artificial intelligence (AI) and robotics. It incorporates a summary of the basic concepts utilized in each of the many technical areas; a review of the state-of-the-art; research developments and needs; an indication of the organizations involved; applications; and a 5-10 year forecast of emerging technology."

42.

Unmeel B. Mehta, STT

Contribution of the chapter "Knowledge-Based Systems for Computational Aerodynamics" in the book "Knowledge-Based Problem Solving", edited by J.S. Kowalik

1985, Prentice-Hall, Inc., Englewood Cliffs, N.J.

43.

George Tucker, OAF
Background: - Research pilot, helicopter operations
Interests: - Expert systems applied to aiding pilots
- may work on "pilot prompter" expert system as a starter project

44.

James T. Wong, XA
Interests: - Mathematical characterization of functional systems
- Optimal diagnostic techniques for fault isolation
- Search techniques, knowledge representation, pattern recognition, and knowledge-based systems

45.

Hiro Miura, FHS
Interests: - System optimization with mathematical programming methods, more generally passive and adaptive optimization of the decision making process
- Integration of AI type decision tools with numerical optimization methods

Application areas considered: - Structural design
- Rotorcraft system
- Flight plan (military, civilian)
- Aerodynamics of wings and propellers

46.

Carolyn Banda, SI (Informatics)
Interests: - Building expert systems
- Tools for building expert systems
- AI languages
- Programs that learn
- Programs that help people learn; CAI using AI techniques, such as constructing a model of the learner

47.

Mo Aidi, ECS
Interests: - AI languages
- Operating systems
- Expert systems

JPL HYPERSPECTRAL IMAGE ANALYSIS PROGRAM

- BACKGROUND - IMAGING SPECTROMETRY IS BOTH QUANTITATIVELY AND QUALITATIVELY DIFFERENT FROM TRADITIONAL MULTISPECTRAL REMOTE SENSING IMAGERY

- IMPACTS ON SCIENCE USERS:
    - VISUAL INTERACTION WITH IMAGE DATA
    - VISUAL INTERACTION WITH SPECTRAL INFORMATION
    - TRADITIONAL STATISTICAL ANALYSIS METHODS BECOME SEVERELY COMPUTE BOUND
    - DATA MANAGEMENT AND SELECTION OF SPECTRAL COVERAGE FOR SPECIFIC ANALYSES
    - PHYSICAL MODELS TO ADEQUATELY DESCRIBE THE DATA

P-13

C-4

JPL HYPERSPECTRAL IMAGE ANALYSIS PROGRAM

- PROGRAM OBJECTIVES
  - DEVELOPMENT OF EFFICIENT INTERACTIVE EXPLORATORY ANALYSIS TOOLS
  - DEVELOPMENT OF EFFICIENT QUANTITATIVE INFORMATION EXTRACTION ALGORITHMS FOR IMAGING SPECTROMETRY
  - APPLICATION OF EXPERT SYSTEM METHODS TO BUILDING AN INTEGRATED DATA MANAGEMENT/ANALYSIS SYSTEM
  - IDENTIFICATION AND UTILIZATION OF EMERGING HARDWARE TECHNOLOGY WHICH PROVIDE COST EFFECTIVE SOLUTIONS TO STORAGE AND ANALYSIS

JPL EXPERT SYSTEM FOR IMAGING SPECTROMETRY

- WHY AN EXPERT SYSTEM?
  - NEED INTEGRATED DATA MANAGEMENT AND ANALYSIS
  - ANALYSIS OF DATA REQUIRES INTELLIGENT GUIDANCE OF INTERACTIVE SESSION (CONSULTANT ROLE)
  - SPECTRAL INTERPRETATION CAN UTILIZE WELL-DEVELOPED INFERENCE RULES AND HEURISTICS
  - INTELLIGENT MANAGEMENT OF AVAILABLE COMPUTATIONAL RESOURCES
  - SYSTEM CAN BE TAILORED (DYNAMICALLY) TO FIT THE NEEDS OF SPECIFIC USERS

JPL EXPERT SYSTEM FOR IMAGING SPECTROMETRY

- FUNCTIONAL CAPABILITIES
  - INTERACTIVE WITHOUT EXHAUSTIVE INTERROGATION OF THE USER - CRISP "FRIENDLY" DIALOGUE
  - INTEGRATED USE OF GRAPHICS AND IMAGE DISPLAY
  - MULTI-MODE USAGE:
    - "DUM B"
    - SMART ADVISOR
    - AUTONOMOUS EXPERT
  - DYNAMICALLY MODIFIABLE KNOWLEDGE BASE
    - USER MODE
    - "MASTER" MODE
  - CAN BE RUN IN GENERAL COMPUTING ENVIRONMENT, I.E., NON-LISP HARDWARE

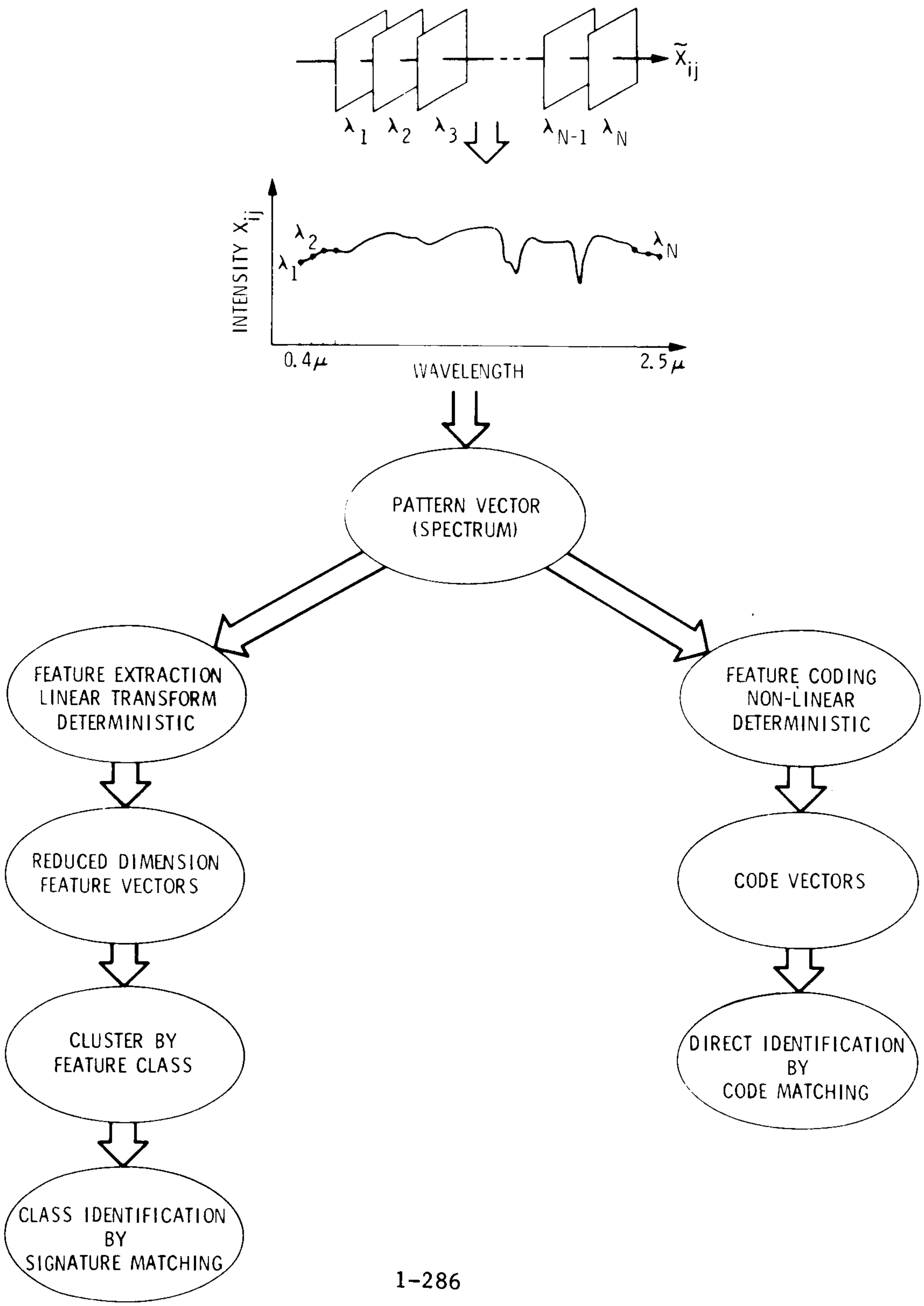
JPL EXPERT SYSTEM FOR IMAGING SPECTROMETRY

- TECHNICAL APPROACH
  - IMPLEMENTATION OF C-BASED EXPERT SYSTEM DESIGN TOOL
  - UTILIZE SEMANTIC NETWORK STRUCTURE FOR KNOWLEDGE REPRESENTATION
  - COMBINE SYMBOLIC/NUMERICAL COMPUTING TECHNIQUES
  - ORIENT SYSTEM AROUND VISUAL DISPLAY OF INFORMATION
  - CONCENTRATE DESIGN ON GEOLOGY APPLICATIONS
    - SPECTRAL KNOWLEDGE BASE MATURE
    - WELL DEVELOPED INFERENCE RULES AND HEURISTICS
    - LOCALLY AVAILABLE "EXPERTS"

JPL

IMAGING SPECTROMETER DATA ANALYSIS METHODS

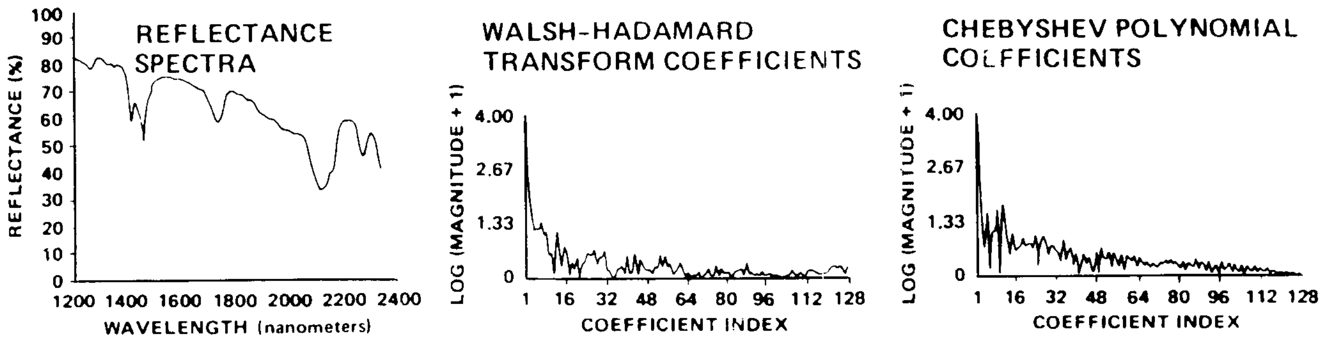
N-DIMENSIONAL PATTERN SPACE



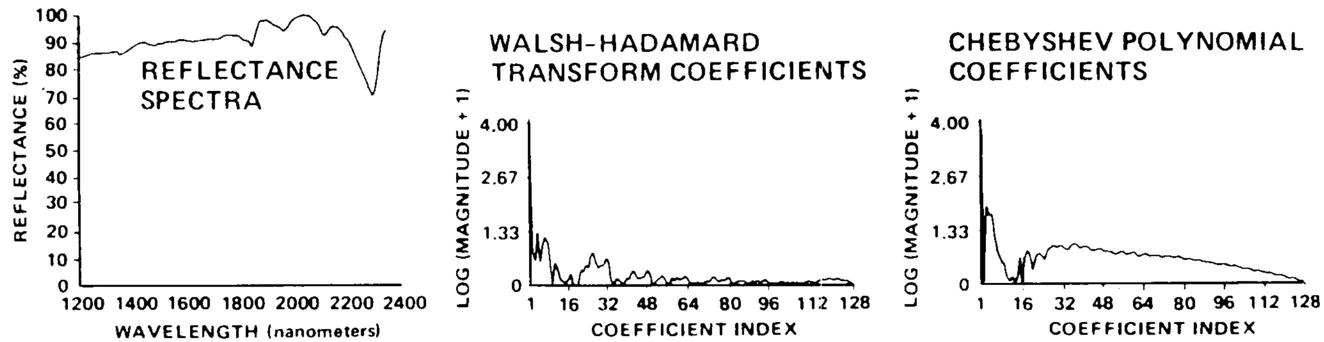
JPL

IMAGING SPECTROMETER DATA ANALYSIS METHODS

- DETERMINISTIC LINEAR TRANSFORM AND CURVE-FITTING METHODS



ALUNITE MINERAL SAMPLE



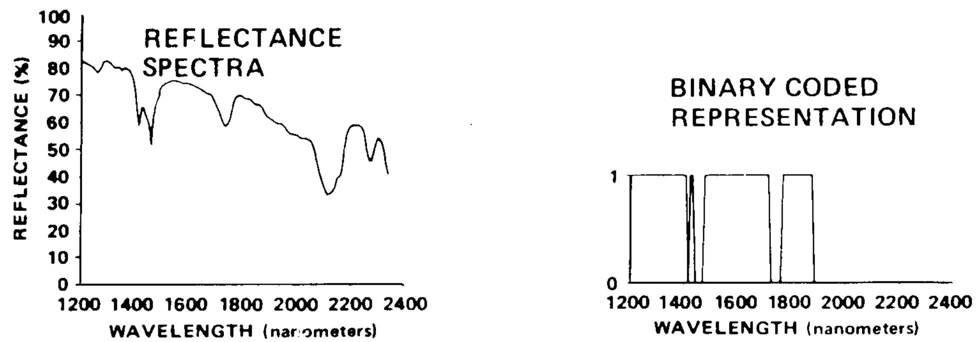
CALCITE MINERAL SAMPLE

- SPECTRAL DIMENSIONALITY CAN BE REDUCED WITHOUT LOSING SIGNATURE UNIQUENESS

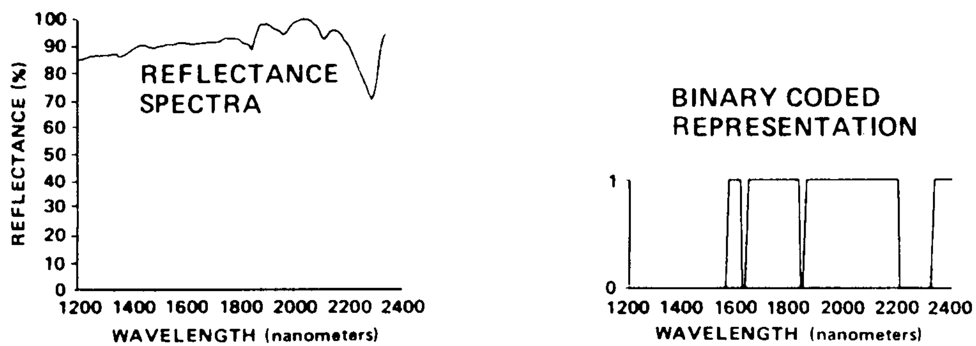
JPL

IMAGING SPECTROMETER DATA ANALYSIS METHODS

- NON-LINEAR FEATURE CODING METHODS



ALUNITE MINERAL SAMPLE



CALCITE MINERAL SAMPLE

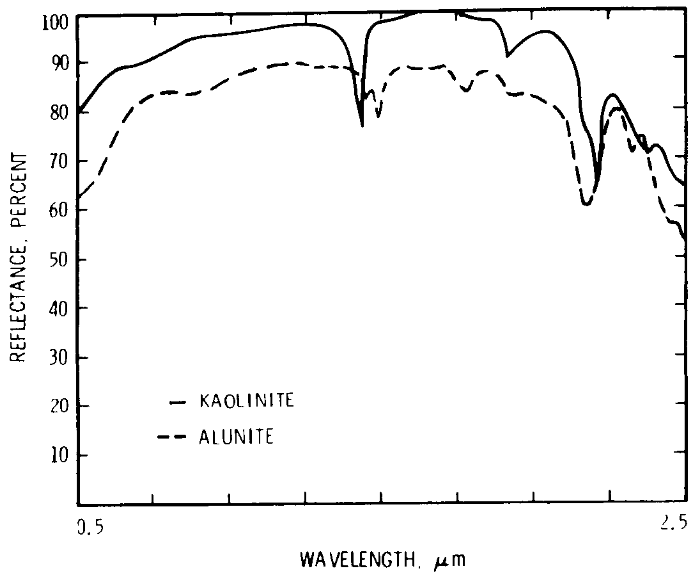
DIRECT BINARY ENCODING

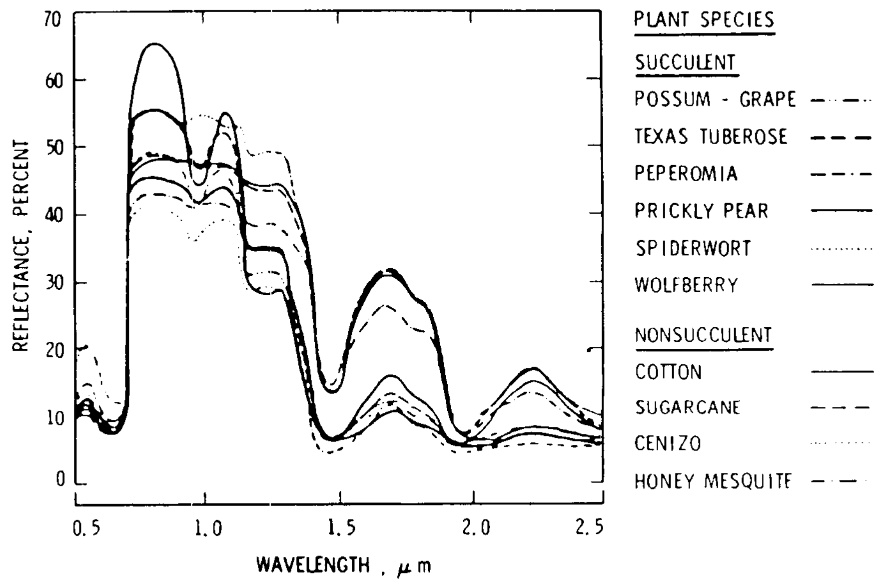
- PRESERVES SIGNATURE UNIQUENESS
- EXTREMELY FAST CROSS-MATCHING CAPABILITIES

JPL

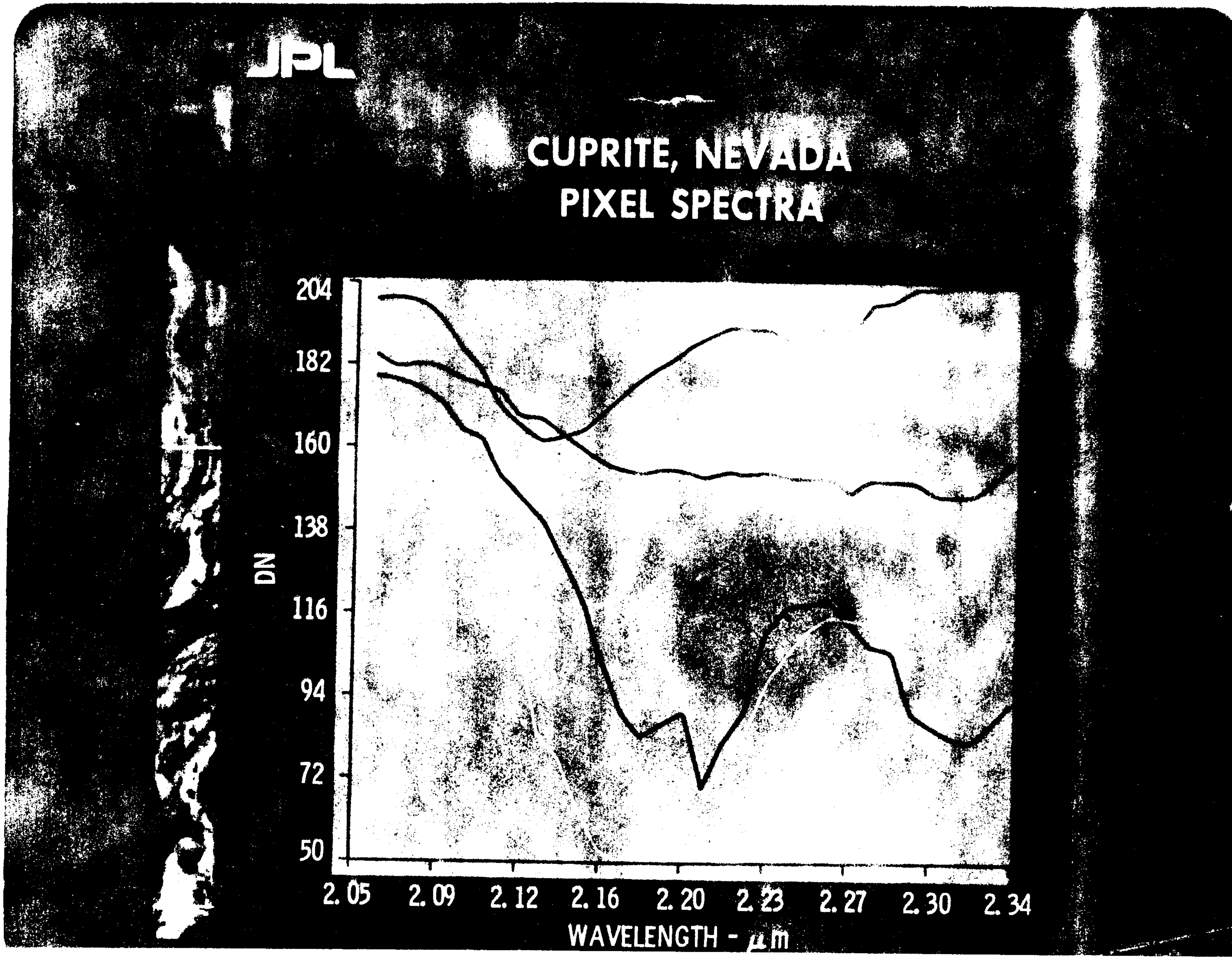
IMAGING SPECTROMETER DATA ANALYSIS METHODS

- SUITABILITY OF APPROACHES - MINERALS vs. VEGETATION





- ROCK/MINERAL SPECTRA SUITABLE FOR FEATURE CODING AND DIRECT IDENTIFICATION
  NON-LINEAR APPROACH
- VEGETATION SPECTRA REQUIRE DETECTION OF MORE SUBTLE EFFECTS
  LINEAR TRANSFORM METHODS

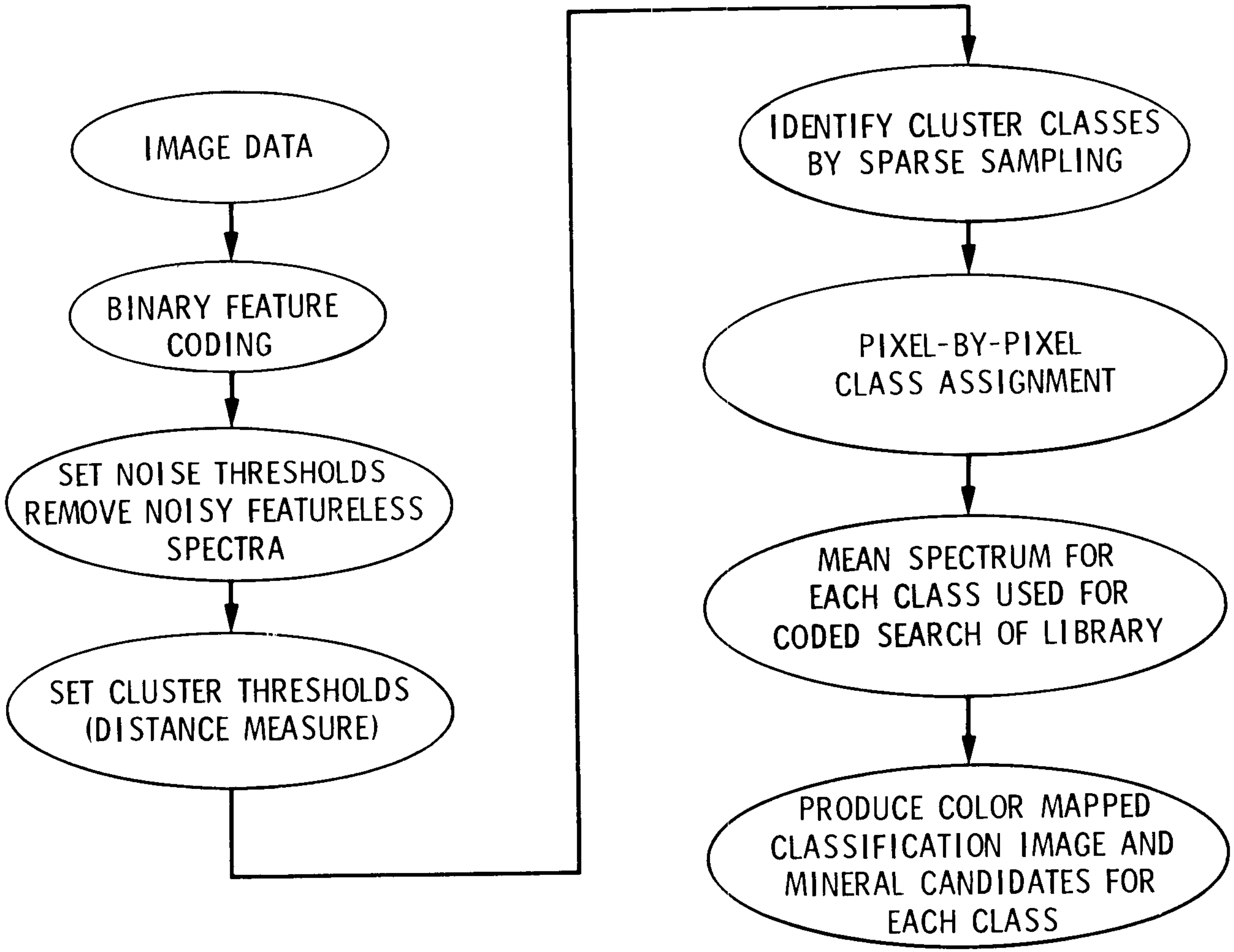


ORIGINAL PAGE IS
OF POOR QUALITY

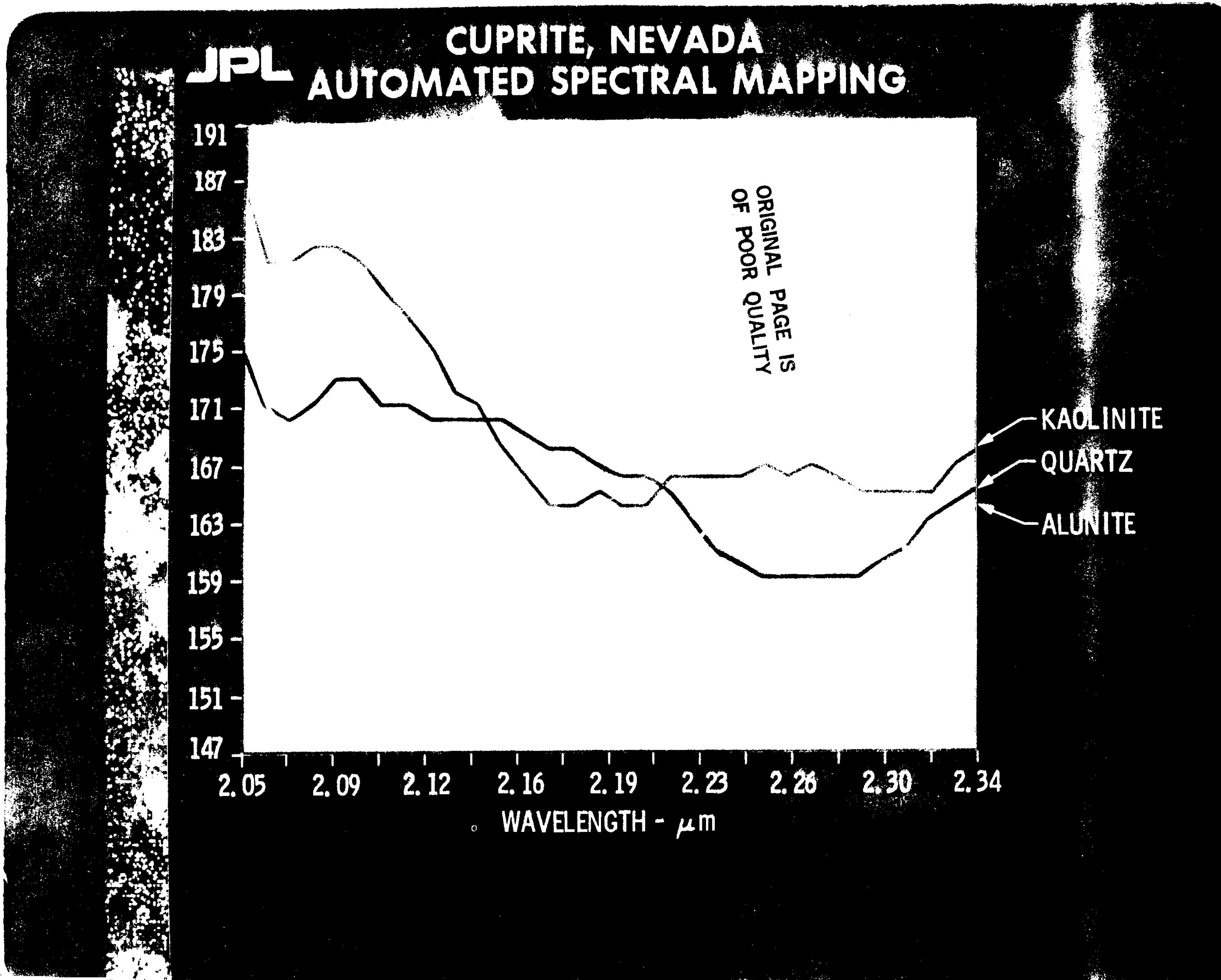
JPL

IMAGING SPECTROMETER DATA ANALYSIS METHODS

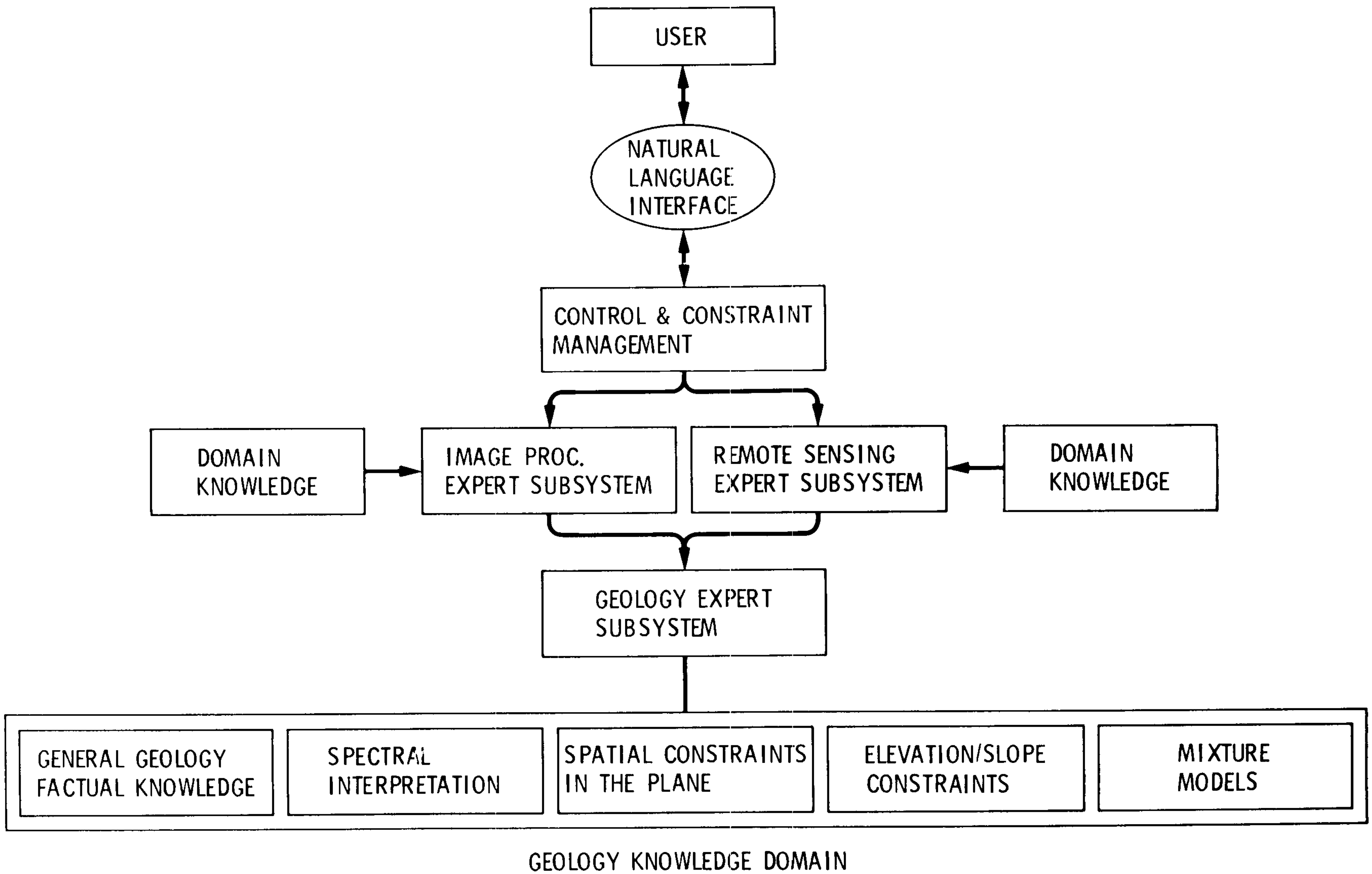
- AUTOMATED MATERIALS IDENTIFICATION USING SPECTRAL LIBRARY



- THIS PROCESS REQUIRES ABOUT 2 MINUTES ON A SUN WORKSTATION FOR A 512 LINE IMAGE


JPL
CUPRITE, NEVADA
AUTOMATED SPECTRAL MAPPING
191
187
183
179
175
171
167
163
159
155
151
147
2.05
2.09
2.12
2.16
2.19
2.23
2.26
2.30
2.34
WAVELENGTH - μm
KAOLINITE
QUARTZ
ALUNITE

JPL EXPERT SYSTEM FOR IMAGING SPECTROMETRY



N87-29142

P-19

EXPERT SYSTEM FOR IMAGING SPECTROMETER ANALYSIS RESULTS

JPL

Gary C. Borchardt

IMAGE PROCESSING APPLICATIONS
& DEVELOPMENT SECTION

JET PROPULSION LABORATORY

JPL

IMPLEMENTATION REQUIREMENTS

- INTEGRATION OF SYMBOLIC AND NUMERICAL TECHNIQUES
- PORTABILITY AND SIZE
- EFFICIENCY OF OPERATION

JPL

STAR
(SIMPLE TOOL FOR AUTOMATED REASONING)

- A LISP-LIKE SOFTWARE ENVIRONMENT FOR THE DEVELOPMENT AND OPERATION OF RULE-BASED EXPERT SYSTEMS

- IMPLEMENTED IN "C." APPROXIMATELY 7000 LINES SOURCE CODE

- SEMANTIC NETWORK REPRESENTATION OF FACTS AND RULES

- FACILITIES FOR INTERACTION WITH PROCEDURES AND DATA STRUCTURES CODED IN C

JPL

STAR DATA STRUCTURES (UNITS)

	UNIT TYPE	EXAMPLES
(1)	NUMBER	100.4 -3.72
(2)	TOKEN	GRANITE FELDSPAR
(3)	STRING	"CLASTIC SEDIMENTARY ROCK"
(4)	LIST	[3.6 -4.88 5.0]
(5)	RECORD	{name -> CHLORITE member_of -> material_classification variety_of -> mica plots -> [^PLOT ^PLOT ^PLOT] }
(6)	EXPRESSION	add(10 20) compare(^PLOT calcite)
(7)	CONNECTION	^ROUTINE ^PLOT

JPL

```
{name -> MENU
 member_of -> function
 arguments -> []
 algorithm  ->
      [initialize()
       repeat(
         [set(current_time +(.current_time 1))
          determine_menu()
          repeat(
            [fetch_user_input()
             through(.current_menu menu_i
               [if(match_menu_item()
                  'break(break(apply(get(.menu_i to_call)[]))))
                 ])
             output_menu()
             ])
          ])
       ]
 }
```

JPL

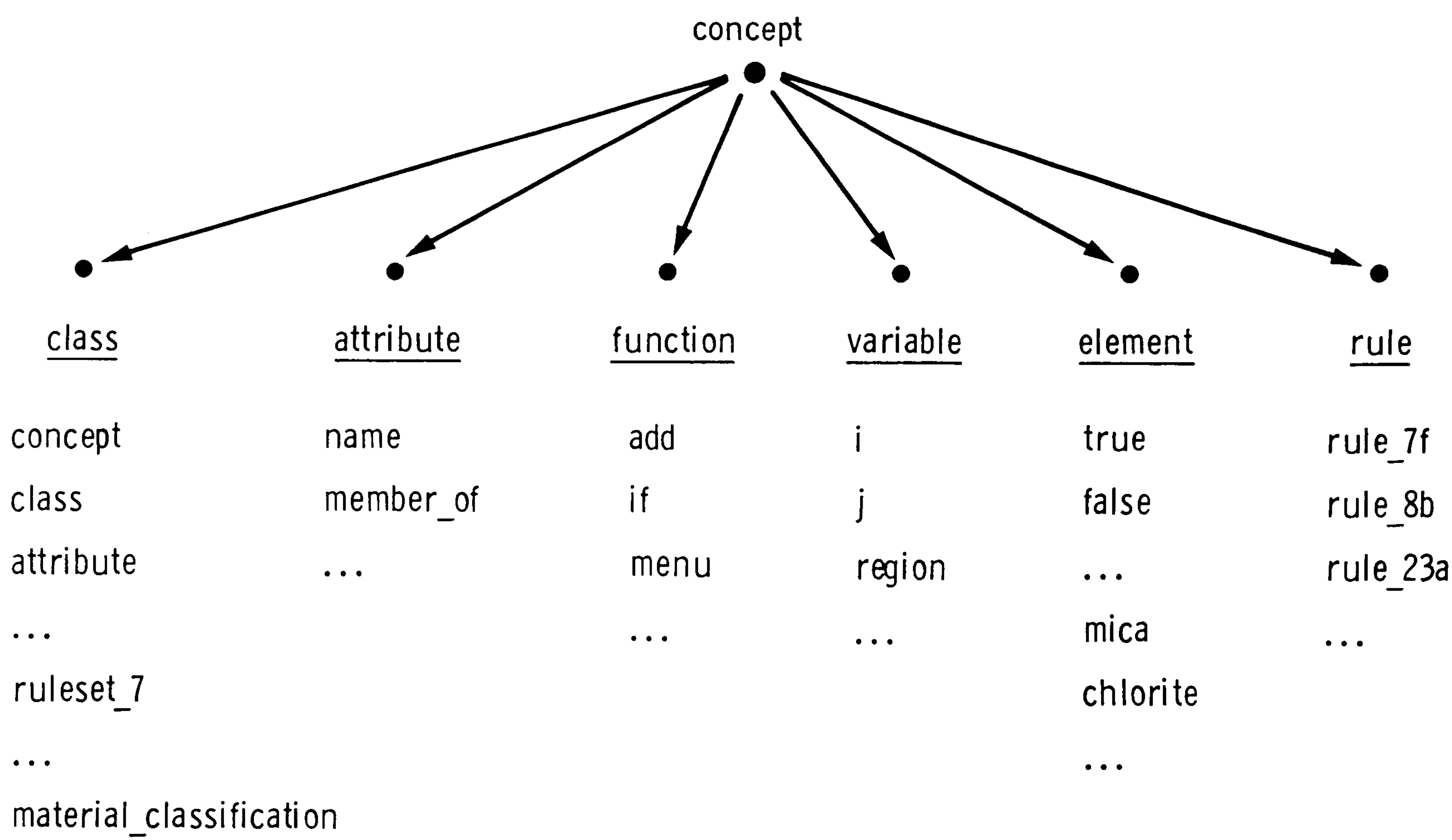
```
{name -> RULE_23A
 member_of -> ruleset_23
 mode -> single_test
 condition -> >(natural_abundance(.i) natural_abundance(.j))
 action -> [discourage(.j .region)]
}

{name -> RULE_7F
 member_of -> ruleset_7
 mode -> single_application
 condition -> exists(.neighboring_minerals j likely_association(.i .j))
 action -> [encourage(.i .region)]
}

{name -> RULE_8B
 member_of -> ruleset_8
 mode -> multiple_application
 condition -> >(size(.possibilities) 20)
 action -> [set(thresh -(.thresh 2)) eliminate_possibilities()]
}
```

```
{name -> MICA
 member_of -> material_classification
 variety_of -> silicate
 varieties -> [biotite chlorite muscovite]
}
```

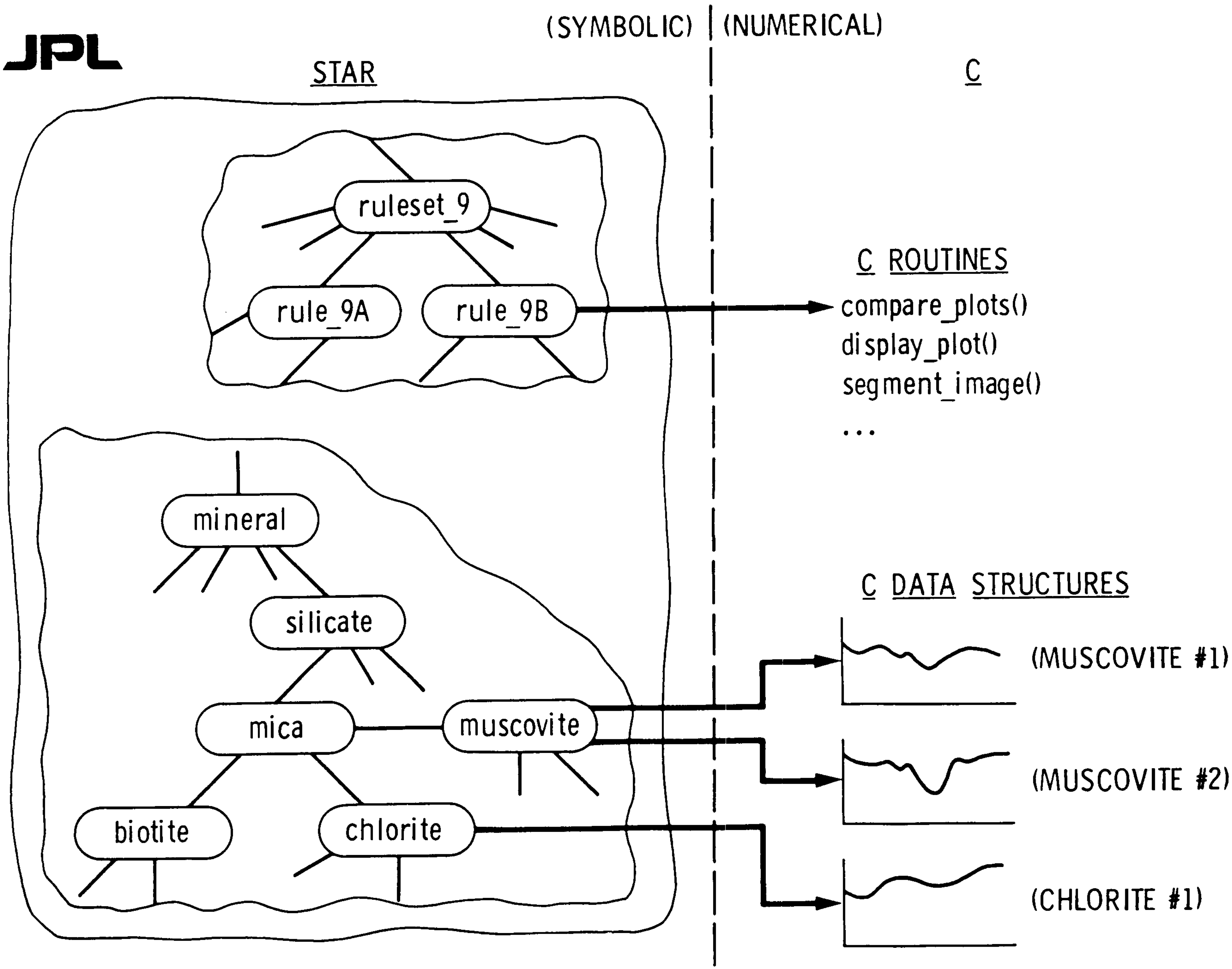
```
{name -> BIOTITE
 member_of -> material_classification
 variety_of -> mica
 plots -> [^PLOT]
}
```

```
{name -> CHLORITE
 member_of -> material_classification
 variety_of -> mica
 plots -> [^PLOT ^PLOT ^PLOT]
}
```

```
{name -> MUSCOVITE
 member_of -> material_classification
 variety_of -> mica
 plots -> [^PLOT ^PLOT]
}
```


concept
class
attribute
function
variable
element
rule
concept
class
attribute
...
ruleset_7
...
material_classification
name
member_of
...
add
if
menu
...
i
j
region
...
true
false
...
mica
chlorite
...
rule_7f
rule_8b
rule_23a
...

JPL


(SYMBOLIC)
(NUMERICAL)
STAR
C
ruleset_9
rule_9A
rule_9B
C ROUTINES
compare_plots()
display_plot()
segment_image()
...
mineral
silicate
mica
muscovite
biotite
chlorite
C DATA STRUCTURES
(MUSCOVITE #1)
(MUSCOVITE #2)
(CHLORITE #1)

JPL

A PROTOTYPE EXPERT SYSTEM

- COMPARING TRADEOFFS BETWEEN SYMBOLIC AND NUMERICAL COMPONENTS
- PROVIDING A BASIS FOR AN EVOLVING SYSTEM
- PROVIDING A FOCUS FOR INTERACTION WITH EXPERTS AND POTENTIAL USERS

JPL

PROTOTYPE EXPERT SYSTEM FOR IMAGING SPECTROMETER ANALYSIS

(SYMBOLIC)

MENU DRIVEN. MENU CHANGES DYNAMICALLY AS SESSION PROGRESSES

OPTIONS
- LOAD IMAGE DATA
- DESCRIBE THE SCENE
- CALIBRATE IMAGE DATA
- SEGMENT THE IMAGE
- IDENTIFY SEGMENTED REGIONS
- DISPLAY A QUANTITY
- EXIT

RULE-BASED IDENTIFICATION OF SURFACE MATERIALS BASED ON

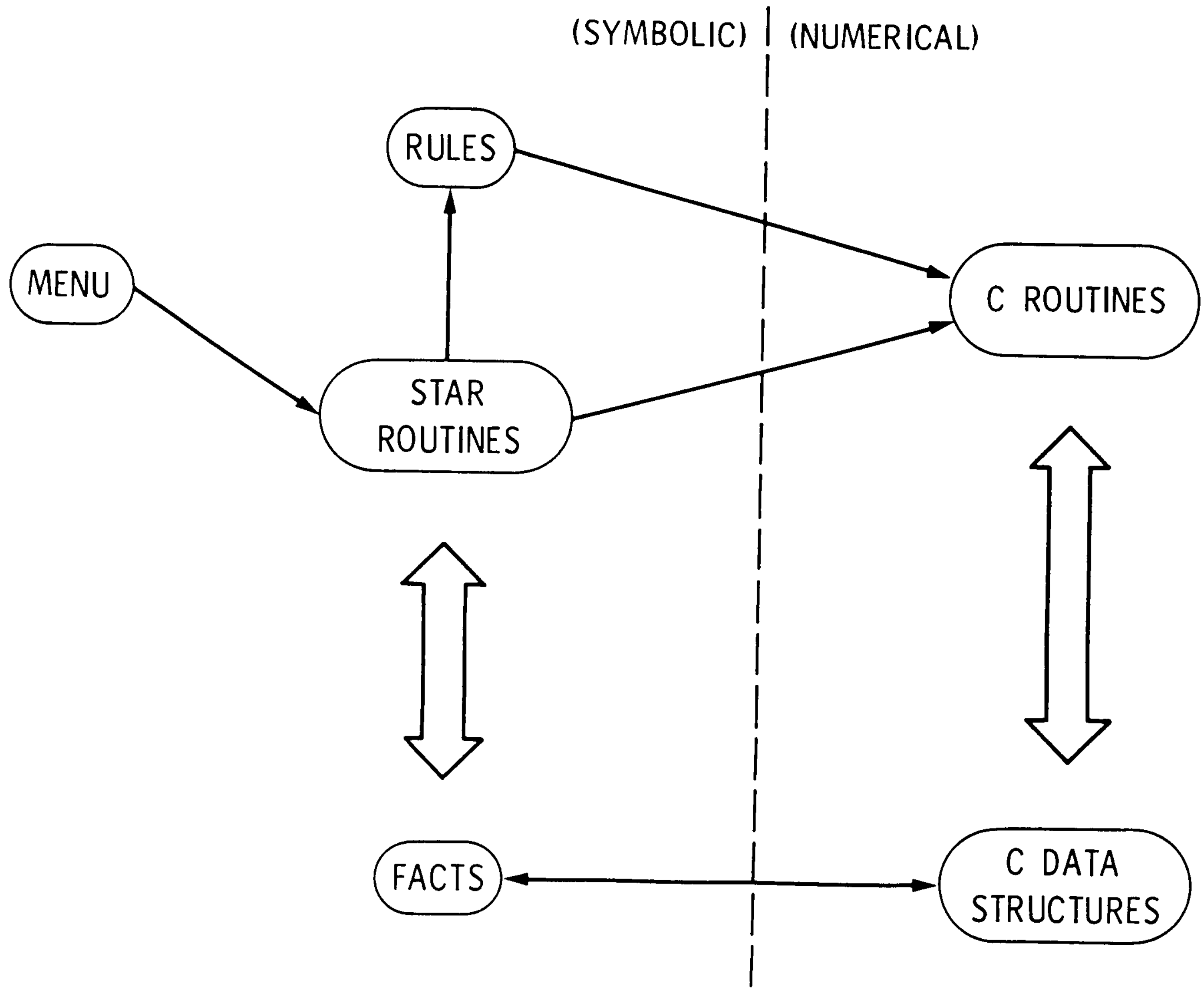
- VARIOUS COMPARISONS WITH PLOTS IN SPECTRAL LIBRARY
- SPATIAL ASSOCIATIONS OF SEGMENTED REGIONS
- RELATIVE NATURAL ABUNDANCE OF MATERIALS
- EXPECTATIONS OF THE USER

(NUMERICAL)

C ROUTINES FOR

- DISPLAY OF QUANTITIES
- IMAGE TRANSFORMATION
- DATA COMPRESSION
- COMPARISON OF SPECTRA
- SEGMENTATION OF THE IMAGE
- MIXTURE COMPONENT ANALYSIS

JPL


(SYMBOLIC)
(NUMERICAL)
RULES
MENU
STAR ROUTINES
C ROUTINES
FACTS
C DATA STRUCTURES

JPL

KEEPING TRACK OF MENU OPTIONS

A STEP IS ...	IF ...
ENABLED	IT MAY BE TAKEN AT THAT POINT
COMPOUNDED	TAKING IT WOULD INVALIDATE A STEP
INVALIDATED	ITS RESULTS ARE NO LONGER VALID

EXAMPLE

```
{name -> CALIBRATION_STEP
 member_of -> step
 description -> "calibrate/transform the image"
 enabled_if -> con(image_loaded master_library_loaded)
 compounded_if -> dis(image_recalibrated segmentation_completed)
 invalidated_if -> aft(image_loaded image_recalibrated)
 to_call -> calibration_step_function
 }
```

JPL

A SAMPLE SESSION

At each prompt ("<>"), either:

1.) hit "return" to see a menu of currently available steps.

2.) enter the first word of a step as it appears in the menu.

<>

Possible steps at this point are:

-> load image and library data from files.

-> exit the session.

<> load

(... load step ...)

<>

Possible steps at this point are:

-> describe the scene and the intended analysis.

-> calibrate/transform the image.

-> segment the image into regions of similarity.

-> display a quantity.

-> backtrack to a previous step.

-> exit the session.

JPL

```
<> segment
     (... segmentation step ... )

<>

Possible steps at this point are:
     -> describe the scene and the intended analysis.
     -> identify segmented regions of the image.
     -> display a quantity.
     -> backtrack to a previous step.
     -> exit the session.

<> backtrack

Backtrack possibilities ("no" if not desired):
     -> load image and library data from files.
     -> calibrate/transform the image.
     -> segment the image into regions of similarity.
     -> no backtrack: return to main menu.

<> calibrate
     (... calibration step ... )
```

JPL

<>

As a consequence of the backtracking just performed, the results of the following steps are currently invalid and have been removed from the system. These steps may be redone if you desire:

-> segment the image into regions of similarity.

<>

Possible steps at this point are:

-> describe the scene and the intended analysis.
-> segment the image into regions of similarity.
-> display a quantity.
-> backtrack to a previous step.
-> exit the session.

<> segment

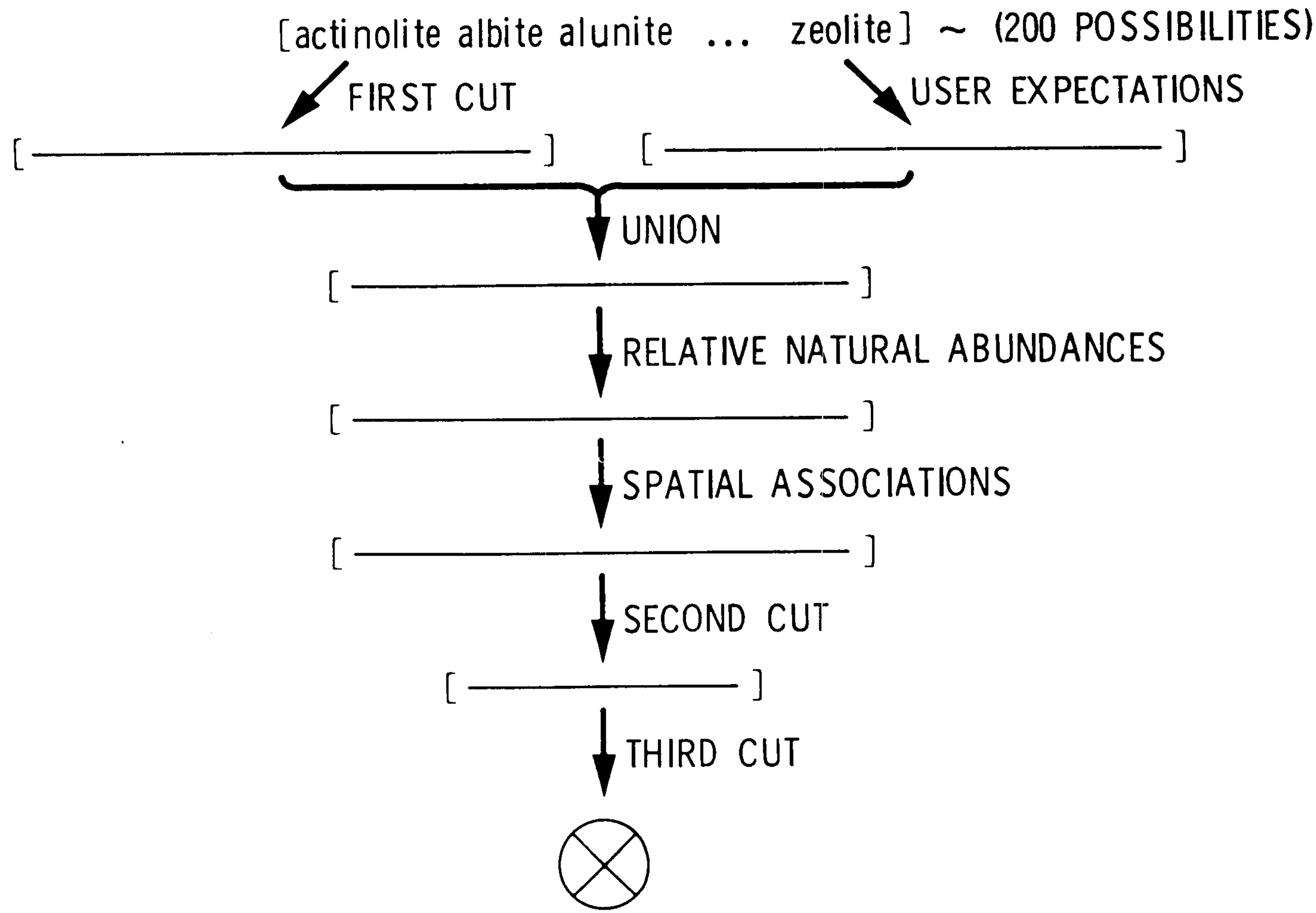
(... segmentation step...)

<>

...

JPL

RULE-BASED IDENTIFICATION OF SURFACE MATERIALS



JPL

EXPLANATIONS

SUPPORT FOR DOLOMITE IN REGION 4:

1) DOLOMITE WAS SELECTED AS ONE OF 63 MATERIALS HAVING ROUGHLY SIMILAR SPECTRAL CHARACTERISTICS TO THAT OF REGION 4.

2) DOLOMITE IS A RELATIVELY COMMON MINERAL.

3) NEIGHBORING REGION 5 APPEARS TO BE COMPOSED OF CALCITE WHICH IS COMMONLY FOUND IN ASSOCIATION WITH DOLOMITE.

4) A SECOND COMPARISON OPERATION DETERMINED DOLOMITE TO BE ONE OF 23 MATERIALS HAVING SIMILAR SPECTRAL CHARACTERISTICS TO THAT OF REGION 4.

5) A FINAL COMPARISON OPERATION SELECTED DOLOMITE AS ONE OF 3 MATERIALS HAVING SIMILAR SPECTRAL CHARACTERISTICS TO THAT OF REGION 4.

JPL

- STAR EXPERT SYSTEMS TOOL COMPLETE AT THIS POINT
- MENU COORDINATION AND NUMERICAL PORTIONS OF PROTOTYPE EXPERT SYSTEM NEARLY COMPLETE
- PROJECTED OPERATIONAL STATUS OF PROTOTYPE EXPERT SYSTEM BEGINNING JUNE 85
- EXPERT SYSTEMS APPROACH IN THIS DOMAIN DOES APPEAR TO BE APPROPRIATE IN CONJUNCTION WITH TRADITIONAL NUMERICAL TECHNIQUES

SOFTWARE LIFE CYCLE DYNAMIC SIMULATION MODEL:
THE ORGANIZATIONAL PERFORMANCE SUBMODEL*
3/1/85

Robert C. Tausworthe
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109

ABSTRACT

This paper describes the submodel structure of a software life cycle dynamic simulation model. The software process is divided into seven phases, each with product, staff, and funding flows. The model is subdivided into an organizational response submodel, a management submodel, a management influence interface, and a model analyst interface. The paper concentrates on the organizational response model, which simulates the performance characteristics of a software development subject to internal and external influences. These influences emanate from two sources: the model analyst interface, which configures the model to simulate the response of an implementing organization subject to its own internal influences, and the management submodel that exerts external dynamic control over the production process.

The paper provides a complete characterization of the organizational response submodel in the form of parametrized differential equations governing product, staffing, and funding levels. The parameter values and functions are allocated to the two interfaces.

* This work reported in this document was performed at the Jet Propulsion Laboratory of the California Institute of Technology under a contract with the National Aeronautics and Space Administration.

SOFTWARE LIFE CYCLE DYNAMIC SIMULATION MODEL:
THE ORGANIZATIONAL PERFORMANCE SUBMODEL

Robert C. Tausworthe
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109

1. INTRODUCTION

In earlier papers [1, 2], the author, in collaboration with Chi Lin and Merle McKenzie, exposed structural design concepts for the construction of a dynamic simulation model of the software life cycle process. These works derived requirements on the form and granularity of the activity breakdown necessary for an accurate simulation. In subsequent works, Don Reifer [3, 4] produced a generic software life cycle work breakdown structure having the required level of detail, and studied the infrastructural dependencies among rates of production, utilization of staff and funding resources, product size characteristics, and situational and environmental factors.

This paper is an extension of these works, describing further structural details of the model, the organization of the overall model into submodels, and the description of one of these submodels in detail.

2. MODEL STRUCTURE

2.1. Core Unit Structure

The works cited above describe the software life cycle process as a dynamic cyclic architecture of project phases, each broken into its constituent unit-task-level activities and flows of products, personnel, funding, and other resources among the activities. Each activity is viewed as having a common structure, referred to as the 'core unit,' shown in Figure 1. The cylindrical 'tank' symbols in the figure refer to quantities, or 'levels,' that may flow within the model. Directed arrows denote paths of flow, and the oblong symbols in the flow paths denote rate controllers. The triangular symbol is a level duplicator, and the pentagonal symbol is a flow duplicator.

2.2. The Software Life Cycle Phase Structure

The software life cycle process treated here has one core unit activity for each of the following seven major phases in the process:

1. *system requirements definition and analysis
2. *system design and hardware/software allocation
3. software requirements analysis

4. software preliminary design
5. software detailed design, implementation, and test
6. *system integration and testing
7. *system maintenance

The four phases marked with asterisks (*) above are not exclusively software-oriented. Modeling of the activities in these phases is limited to the involvement of software personnel.

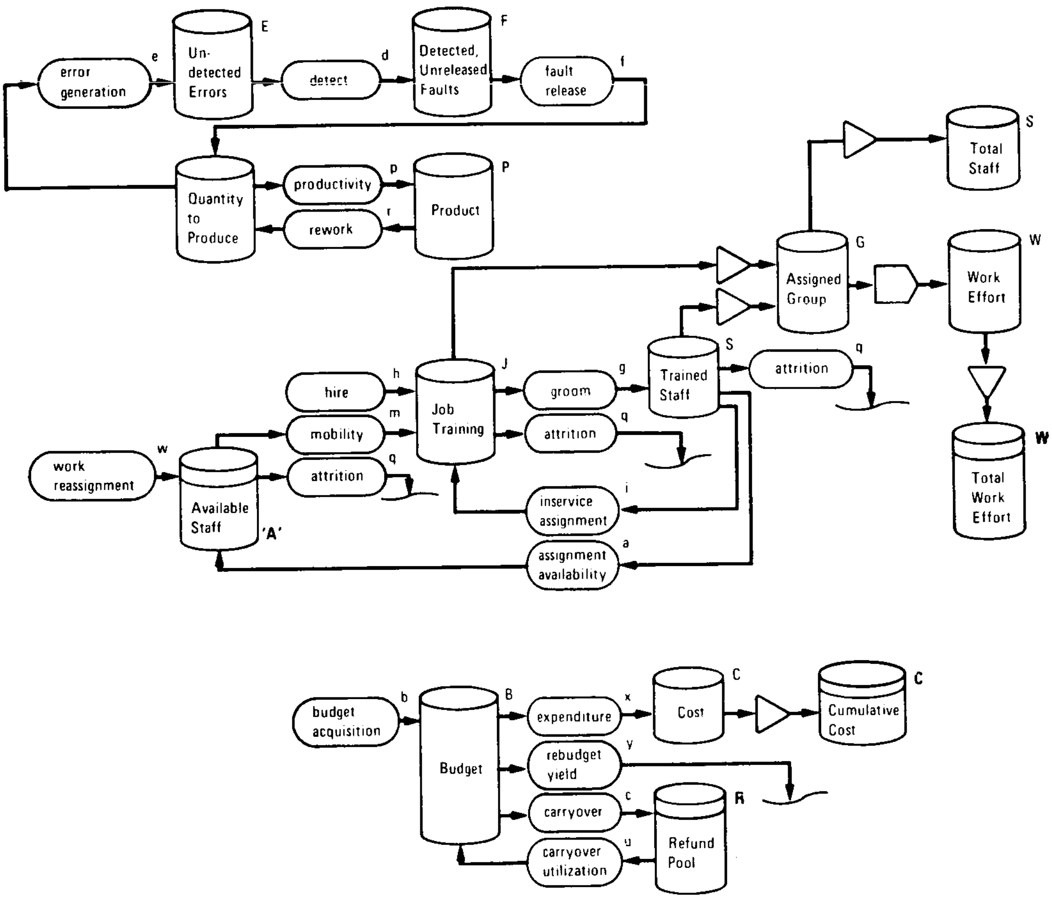


Fig. 1: Software Life Cycle Simulator Core Unit Structure

Each phase starts with a certain unknown volume of product that must be produced as precedent to the succeeding phases. A certain amount of the product will be produced correctly, some will be produced with as-yet-undetected faults in it, and some of the product will contain faults that have been discovered, but not yet repaired. Even portions produced correctly may have to be reworked when requirements change. The rates at which each of these portions of the product are generated (and, for errors, disposed of) are dynamic functions of both inherent and manageable parameters.

The production rate, or rate at which the quantity of product backlog is transformed into the finished product, is dependent, among other things, on the size and characteristics of the applied staff. For each phase, staff may be acquired from in-house resources or from the labor market. Each, upon entering into the new phase, may undergo a period of training (and a longer period of learning), perhaps administered by staff elements already on the job (inservice training), who take time out from their regular duties for this purpose. Staff may also be lost through attrition, or may be reassigned to activities in other phases or to other in-house tasks.

Staffing requires fiscal resources in order to exist. The allocation and acquisition of sufficient budget to sustain the staff is a prime requisite for doing work in a particular phase. On occasion, there may some funding for a phase left over after the phase product is complete that may be made available to another phase. In some cases, funds budgeted to a particular activity may have to be preempted to support another phase, or another project. In the latter case, the funding is lost.

2.3. Submodel Structure

The overall simulator is divided into four parts (Figure 2): an organizational performance submodel, a management submodel, a management influence interface, and a model analyst interface. The organizational performance submodel includes all of the core unit activities, and is described by a set of differential equations governing the levels and flows of products, personnel, and funding. Organizational performance is completely specified by the current state of the levels and the flowrate functions.

The management submodel contains a plan model, a visibility model, and an action model, each appropriately parametrized. Visibility into organizational performance is achieved via access to levels, flowrates, and flowrate parameters. Control is accomplished through manipulation of parameters within the management submodel interface.

The model analyst provides non-management performance parameter values that are inherent to the software process and to the performing organization, as derived from statistical or conceptual data.

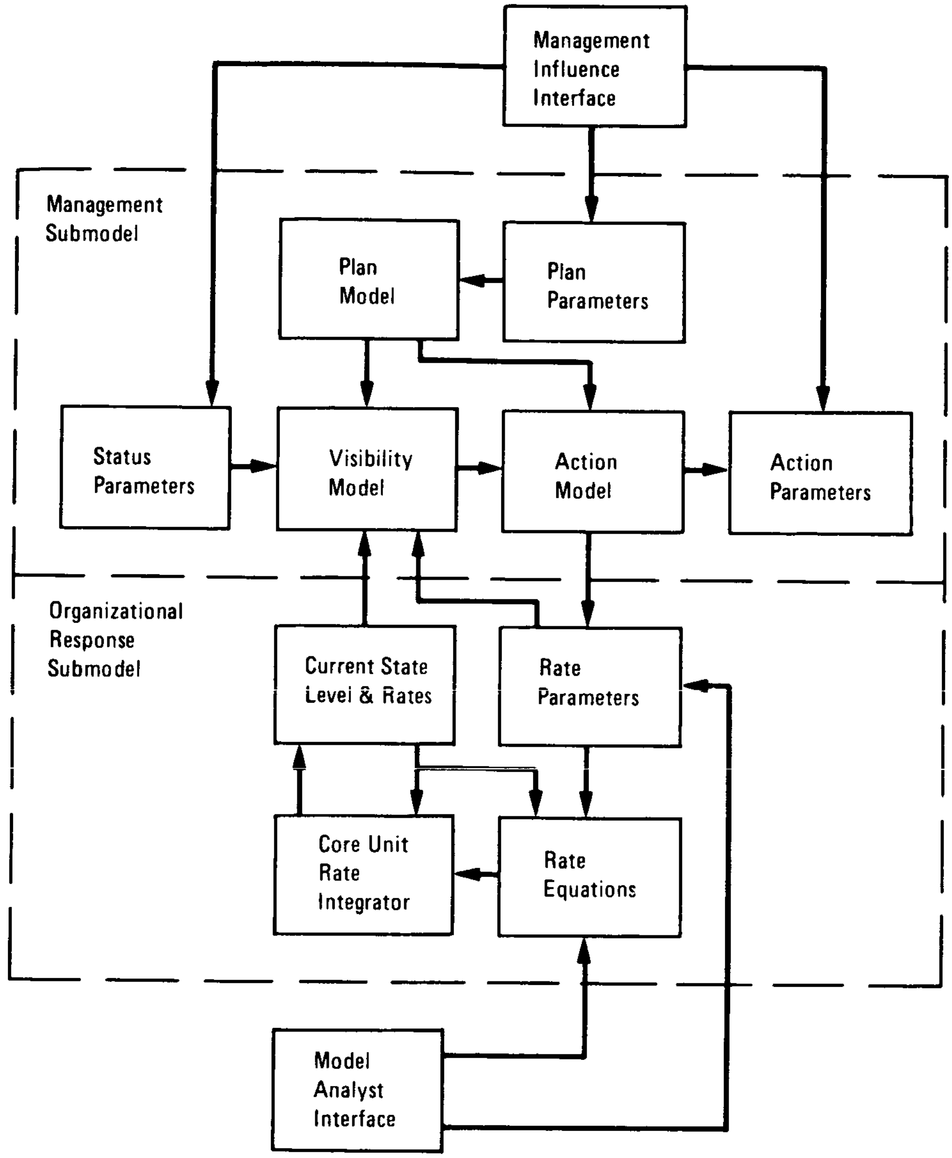


Fig. 2 : Software Life Cycle Simulator Submodel Structure

3. MODEL DESCRIPTION

3.1. Notation

In describing the equations governing the organizational performance model, the following notation is used:

Capital letters denote levels, lower-case letters denote flowrates and flowrate auxiliaries, and Greek characters denote flowrate parameters. Boldface capitals refer to cumulative levels across the entire submodel. Levels, rates, and parameters are subscripted by their phase indexes, 1 through 7, as above. The subscript ϕ refers to the phase under current consideration. When describing the relationships of quantities within a particular phase, this subscript is often suppressed.

Some parameters may have multiple subscripts, the first of which is always the phase number, perhaps suppressed. However, if one quantity in an equation describing the behavior of the model bears a phase subscript, then all parameters in the equation are subscripted by the appropriate phase. The phase number is never suppressed when all quantities in an equation do not refer to the same phase.

All quantities are, or potentially are, functions of time. The time dependency, however, is generally also suppressed for simplification of the formulas.

For each phase core unit, the levels are:

Q = Quantity of product yet to be produced
P = amount of Product produced so far
F = amount of Faulty product so far detected, this phase
E = product Error (fault) content, this phase
J = size of Job training staff pool
S = size of trained Staff
G = Group size, staff in this phase
S = total project Staff level
A = size of Available staff pool
W = Work effort, this phase
W = total Work effort, all phases
B = unspent Budget
C = Cost, so far this phase
C = total Cost, so far, all phases
R = carryover Refund pool

The flowrates among these levels are:

p = productivity
r = rework rate
d = fault detection rate
e = error generation rate
f = fault release rate
h = hire rate, labor market
w = worker reassignment rate
m = mobility of available staff
g = grooming (training) rate
i = inservice training assignment rate
q = quit (attrition) rate
a = available staff assignment availability rate
b = budget acquisition rate
x = budget expenditure rate
c = carryover rate
y = yank (budget reduction) rate
u = utilization rate of carryover pool

The function $U(x)$ is the so-called 'unit-step' function

$$U(x) = 1 \quad \text{if } x = 0 \text{ or } x > 0$$
$$= 0 \quad \text{otherwise}$$

3.2. Normalization

All levels and rates defined above are normalized quantities with respect to amount of product, effort, budget, and time duration. In the following discussion, a prime symbol (') will be affixed to unnormalized quantities, and unprimed symbols denote normalized quantities.

Time is normalized such that unit time corresponds to a certain fraction of each simulated project, no matter how long the actual schedule is. It also will be convenient to normalize level values so that unit product, total effort, and total funding levels are used, regardless of project size.

The same time normalization is used for all core units, viz., the planned schedule time to enter phase 7. The product levels and product flow rates within each core unit are normalized individually by the actual (unknown) volume of product for that unit. The remaining levels and rates are normalized by overall planned values. Effort levels and rates are normalized by the planned expected project effort, while funding levels and rates are normalized using the planned project cost.

3.2.1. Parametric Time Normalization

Let t' denote actual (real) time values, and t denote parametric time, related by

$$t' = T\,t$$

where T is the specified time-normalization parameter. Note that when parametric time $t = 1$, then real time $t' = T$.

Let $Z'(t')$ be a flow volume, or level, and $Z(t)$, its corresponding normalized equivalent under the relationship

$$Z'(t') = Z_0\, Z(t)$$

with an appropriately defined time-independent parameter Z_0. When normalized such that $Z_{max} = 1$, the value of Z_0 becomes

$$Z_0 = Z'_{max}$$

For a flowrate $z'(t')$, the corresponding equation is

$$z'(t') = z_0\, z(t)$$

The volume of flow due to z' over a period of time will be finite, because the model deals with finite resources. Let this accumulated value be Z', and the corresponding normalized value be Z. Because of this time normalization factor, the values Z and Z' are related by

$$Z' = T\, z_0\, Z$$

or

$$z_0 = Z'/(Z\ T) = Z_0/T$$

where Z_0 is the level normalization factor. Time normalization, therefore, leads to T factors in the normalizing coefficients of flowrates.

3.2.2. Product Normalization

Let P_0 denote the actual (unknown) final amount of product to be produced in a particular phase. Then the relationship between P' and P is

$$P'(t') = P_0\, P(t)$$

in which P is confined to the interval [0, 1]. The flowrates influencing the product are

$$p'(t') = (P_0/T)\, p(t)$$

and

$$r'(t') = (P_0/T)\, r(t)$$

The model thus concerns itself with unit products in each phase, so that the actual sizes of each unit within the normalized organizational submodel are immaterial. Should the unknown final amount of product P_0 expand to P_1 due to, for example, an increase in requirements, then the submodel would view this as if P_1 were the normalizing factor, whereupon the normalized P would show a decrease by an amount $(P_1 - P_0)/P_1$.

3.2.3. Fault Normalization

Both detected and undetected product fault levels are normalized with respect to the unknown final product size,

$$E'(t') = P_0\, E(t)$$
$$F'(t') = P_0\, F(t)$$
$$d'(t') = (P_0/T)\, d(t)$$
$$f'(t') = (P_0/T)\, f(t)$$

Therefore, E and F are measures of the relative error content in the product.

3.2.4. Staff Normalization

Let $\mathbf{W}_0$ be the real planned total effort for the project, and let W_0 be the planned effort for a core unit. Then

$$\mathbf{W}_0 = \mathbf{W}_{1,0} + \dots + \mathbf{W}_{7,0}$$

$$\mathbf{W}'(t') = \mathbf{W}_0 \, \mathbf{W}(t)$$

Let $\mathbf{S}_0$ be a staffing normalization value defined by

$$\mathbf{S}_0 = \mathbf{W}_0 \;/\; T$$

i. e., $\mathbf{S}_0$ is the average full-time-equivalent staff engaged in the project. The flowrate normalizations are

$$\begin{aligned}
h'(t') &= (\mathbf{S}_0/T)\ h(t)\\
m'(t') &= (\mathbf{S}_0/T)\ m(t)\\
q'(t') &= (\mathbf{S}_0/T)\ q(t)\\
g'(t') &= (\mathbf{S}_0/T)\ g(t)\\
i'(t') &= (\mathbf{S}_0/T)\ i(t)\\
a'(t') &= (\mathbf{S}_0/T)\ a(t)
\end{aligned}$$

and the levels are

$$\begin{aligned}
J'(t') &= \mathbf{S}_0\ J(t)\\
S'(t') &= \mathbf{S}_0\ S(t)\\
W'(t') &= \mathbf{W}_0\ W(t)
\end{aligned}$$

3.2.5. Budget Normalization

If $\mathbf{C}_0$ represents the real planned total cost of a project, and if C_0 is the cost allocation made to a particular phase, then

$$\mathbf{C}_0 = C_{1,0} + \dots + C_{7,0}$$

$$\mathbf{C}'(T') = \mathbf{C}_0\ \mathbf{C}(t)$$

The normalized cost rates are then

$$\begin{aligned}
b'(t') &= (\mathbf{C}_0/T)\ b(t)\\
x'(t') &= (\mathbf{C}_0/T)\ x(t)\\
y'(t') &= (\mathbf{C}_0/T)\ y(t)\\
c'(t') &= (\mathbf{C}_0/T)\ c(t)\\
u'(t') &= (\mathbf{C}_0/T)\ u(t)
\end{aligned}$$

and the levels are

$$\begin{aligned}
B'(t') &= \mathbf{C}_0\ B(t)\\
C'(t') &= \mathbf{C}_0\ C(t)\\
\mathbf{R}'(t') &= \mathbf{C}_0\ \mathbf{R}(t)
\end{aligned}$$

3.3. Approximation of Time Delays

The modeling of time delays in a process simulation model can be accomplished in one of two ways: all samples of the process may be stored in a queue during the delay time, or the samples may be put through a linear filter whose transfer characteristic approximates the desired delay. The former method requires a queue length equal to the process sampling rate times the delay time, and is appropriate whenever queue storage requirements are not extreme. The latter method suffers from amplitude and phase distortions when the degree of the delay filter is too small, but memory requirements are generally much more modest. Each of the methods may appear in the organizational performance submodel, as appropriate.

The linear filter transfer function of degree n corresponding to a 'maximally flat' unit-delay is given (in Laplace transform notation) by the equation [5]

$$D_n(s) = b_0 \,/\, (b_0 + b_1 s + \ldots + b_n s^n)$$

where b_k, $k = 0, \ldots, n$ represent the coefficients

$$b_k = \frac{(2n - k)!}{2^{n-k}\, k!\, (n - k)!}$$

The filters for n = 2 and n = 3 are, for example,

$$D_2(s) = \frac{3}{3 + 3s + s^2}$$

and

$$D_3(s) = \frac{15}{15 + 15s + 6s^2 + s^3}$$

The response of these filters may be expressed as nth-order linear differential equations. That is,

$$y(t) = D_n(s)\, x(t)$$

translates into the differential equation

$$b_n y^{(n)} + \ldots + b_1 \dot{y} + b_0 y = b_0 x$$

For ease in computer solution, this equation is usually rewritten in state-vector form, in which y_k denotes the kth derivative of y(t), as

$$\dot{y}_{n-1} = [b_0 x - (b_0 y_0 + \ldots + b_{n-1} y_{n-1})] \,/\, b_n$$

$$\dot{y}_{n-2} = y_{n-1}$$

$$\cdots$$

$$\dot{y}_0 = y_1$$

In this way, the vector of derivatives, $(\dot{y}_0, \ldots, \dot{y}_{n-1})$ can be determined from the state vector $(y_0, \ldots, y_{n-1})$ at time t, and then numerically integrated to give the state vector values at time $t + \Delta t$.

For delay τ, rather than unit delay, it is merely necessary to replace each b_k above by $b_k' = b_k \tau^k$. For example, the 3rd-order maximally-flat τ-delay filter equations are

$$\dot{y}_0 = y_1$$

$$\dot{y}_1 = y_2$$

$$\dot{y}_2 = (15/\tau^3)\,(x - y_0 - \tau y_1 - 0.4\tau^2 y_2)$$

4. LEVEL EQUATIONS

4.1. Level Equation Form

The normalized equations of flow for each core unit may now be completely specified in terms of flowrates and levels. As above, the overdot in the equations below denotes time-differentiation,

$$\dot{Z} = dZ/dt$$

Since all levels in the model are non-negative quantities, the flow equation for each level necessarily takes the form

$$\dot{Z} = z\ U(Z)$$

so that a negative flow rate never produces a negative level. In the submodel level equations to follow, the step-function factor is omitted for simplicity.

The level equations that follow are mere mathematical restatements of the flow structure depicted in Figure 1.

4.2. Quantity of Product

Each phase deals with a unit product, which, at any particular time, may be composed of an as-yet-unproduced Quantity (Q), an amount having as-yet-undiscovered Errors (E), an amount having discovered, unrepaired Faults (F), and an amount of finished, correct Product (P).

$$\dot{Q} = r + f - p - e$$

$$\dot{P} = p - r$$

$$\dot{E} = e - d$$

$$F = 1 - Q - P - E$$

4.3. Job Training Staff Pool

The job-training level is composed of incoming untrained (new) staff, J_n, and trained staff, J_t, brought in for inservice training. The overall level is described by

$$\dot{J} = h + m + i - g - q_J$$

and the individual untrained and trained levels are

$$\dot{J}_n = h + m - (J_n/J)(g + a_J + q_J)$$

$$\dot{J}_t = i - (J_t/J)(g + a_J + q_J)$$

where a_J is the staff assignment availability (not shown in Figure 1) applicable to the job training pool.

4.4. Trained Staff

The trained Staff, S, consists of personnel dedicated to production and QA activities within the current phase.

$$\dot{S} = g - i - a_S - q_S$$

Note that the staff assignment availability, a, in Figure 1 is, in reality, made up of two parts, a_J and a_S, respectively applicable to the job training pool and the trained staff pool. The total group staff, G, is thus described by

$$\dot{G} = h + m - a - q$$

where $a = a_J + a_S$ and $q = q_J + q_S$.

4.5. Work Effort

The Work effort level, W, is the cumulative group staff time spent so far in the current phase.

$$\dot{W} = J + S = G$$

4.6. Project Work Effort

The project cumulative work effort, $\mathbf{W}$, is the current value of total work effort in all phases.

$$\mathbf{W} = W_1 + \ldots + W_7$$

4.7. Available Staff

Available staff, $\mathbf{A}$, denotes a pool of personnel resources not currently engaged in the project, but available to do so. Staff completing a given phase are reassigned to the available staff level before being reassigned, as appropriate, to other phases. Staff external to the project may also be added to the pool by work reassignment.

$$\dot{\mathbf{A}} = w + (a_1 - m_1 - q_{1,A}) + \ldots + (a_7 - m_7 - q_{7,A})$$

4.8. Budget

The phase budget, B, is the current value of remaining dollar resources allocated to the current phase.

$$\dot{B} = b + u - x - c - y$$

4.9. Cost

The phase cost, C, is the current value of the phase expenditure.

$$\dot{C} = x$$

4.10. Total Cost

The project total current cost, $\mathbf{C}$, is the sum of all phase costs.

$$\mathbf{C} = C_1 + \ldots + C_7$$

4.11. Carryover Refund Pool

The carryover refund pool, $\dot{R}$, is the total of all funds given up by some phases and not yet utilized (obligated) to other phases.

$$\dot{R} = (c_1 - u_1) + \ldots + (c_7 - u_7)$$

5. RATE EQUATIONS

The formulation of the organizational performance model, as can be seen from the foregoing paragraphs, places the burden of achieving simulation accuracy in proper definition of rate equations and initial values for levels. This section parametrizes the flowrate quantities using simple, intuitive phenomenological models, as follows:

5.1. Productivity

The general form for the productivity, or production rate, equation is

$$p = p_0 \, p_J \, p_c \, p_q \, p_s \, p_1 \cdots p_n$$

where p_0 is a nominal time-independent productivity value for trained staff, p_J is an adjustment factor for staff in training, p_c is a compensation for communication overhead and other effects of overall project staff size, p_q adjusts for effort being used in error-detection and quality assurance (QA), p_s compensates for learning effects in the phase, and the other multipliers p_k are adjustments due to environmental, situational, organizational, experience, and other factors.

Each of the phase products is considered a separate, precedent milestone for doing correct work in the succeeding phases. Work may still be done without having 100% precedent in succeeding phases, but there will be a higher probability of making errors in that work that will later have to be corrected. No total overall product metric is defined.

5.1.1. Job Training Pool Effects

The effects of having a staff group G split between an untrained staff pool (J) and trained staff pool (S) is modeled by the productivity adjustment factor

$$p_J = S + \pi_J J$$

where π_J is a productivity ratio value for staff in the job-training pool (J), including personnel doing the training. It principally reflects the effects of time spent in training activities rather than in production. Learning-curve effects are treated separately, below. The parameter π_J is supplied by the management submodel.

5.1.2. Communications Overhead

One productivity adjustment is due to organizational communications overhead. This overhead is simulated using an overhead model [6] that postulates that the time increment spent in overhead activities is proportional to the staff increment and the non-overhead time remaining. The productivity adjustment factor in this case is given by

$$p_c = \exp[-(S - 1/S_0)\gamma_c]$$

for $S > 1/S_0$, and $p_c = 1$ otherwise.

The parameter γ_c is the communications relative time factor, supplied by the model analyst.

5.1.3. Staffing and Learning Curve Effects

It is generally accepted that productivity of personnel increases due to several kinds of learning, among which are general experience, organizational experience, and specific task familiarity. Each of these productivity effects is commonly described dynamically by a first-order linear differential equation having a 'learning time-constant' parameter.

The organizational response submodel approximates only the task familiarization effects, and relegates the other, longer-term experience adjustments to other p_k factors. Thus, the total staff productivity due to size and state of familiarity is taken to be the form,

$$p_s = \pi_0 + \pi_s \; / \; G$$

where π_s satisfies the equation

$$\tau_s \dot{\pi}_s + \pi_s = (1 - \pi_0) \; G$$

in which τ_s is the learning time-constant, G is the staffing function to which that kind of learning applies, and π_0 represents the untrained-staff/trained-staff productivity ratio. The trained-staff productivity is thus normalized to unity.

The value of p_s is the learning-state productivity adjustment for the staff group G as a function of time. If the staff G were applied all at once, p_s would rise from π_0 asymptotically to 1. However, since the staffing plan may not be a step-function, the differential equation form is used.

The learning time parameter τ_s is a function of the teacher/student ratio, ρ, and is longest when staff members learn on their own (i. e., at $\tau_{s,0}$, when $\rho = 0$).

The variation in learning time may be approximated by a cubic form

$$\tau_s = \tau_{s,0} + \tau_s' \rho (1 - \rho)^2 + \Delta\tau_s \rho^2 (2\rho - 3)$$

This particular form takes on the self-taught time value at $\rho = 0$, is minimum when the teacher/student ratio is unity, being reduced by a (positive) increment $\Delta\tau_s$ at this point, and has (negative) slope τ_s' at the origin. For $\rho > 1$, it was assumed that there would be too many teachers per student, so that the training time would actually take longer than 1-on-1 training. The form is subject to the restriction

$$\Delta\tau_s > - \tau_s' / 2 > 0$$

The parameters π_0, $\tau_{s,0}$, τ_s', and $\Delta\tau_s$ all are supplied by the model analyst interface.

5.1.4. Effect of Quality Assurance

If ξ_q represents the fraction of effort devoted to quality assurance (QA) pursuits (i. e., the 'extent' of QA) to discover errors or otherwise improve the quality of product, there is a corresponding reduction in productivity due to the reduced effort. Nevertheless, the overall correct-product rate may be improved, because faults may be discovered before they propigate into other phases.

The productivity adjustment factor for QA activity is thus of the linear type,

$$p_q = 1 - \xi_q$$

The QA fraction emanates from the management submodel.

5.1.5. Linear Extent Factors

Other productivity adjustment factors may take the linear form exhibited above,

$$p_k = \begin{cases} 1 + \pi_k\xi_k & \text{for } \pi_k\xi_k > -1 \\ 0 & \text{otherwise} \end{cases}$$

The extent factors ξ_k range in the intervals [0, 1] or [-1, 1]. In either case, the productivity adjustment factor ranges from its least to most beneficial value as ξ_k varies from the lower to the upper limit. In the former case, π_k is the total swing in productivity adjustment,

$$\pi_k = p_{k(max)} - p_{k(min)}$$

while in the latter case, it is only half this amount.

The π_k parameters are supplied by the model analyst for projects in general, while the ξ_k are supplied by the model user to relate the extent to which each factor under consideration is present in the project to be simulated.

5.1.6. Exponential Extent Factors

Several productivity adjustment factors take the form

$$p_k = (\pi_k)^{\xi}$$

where $\pi_k > 1$ is the maximum beneficial effect of project factor k, and $\xi = \xi_k$ is a value in the interval [-1, +1] that registers the extent to which factor k is present in the current project. When $\xi_k = -1$, there is minimum benefit of factor k, so the value π_k is seen to be the square of the max/min productivity ratio,

$$\pi_k^2 = p_{k(max)} \; / \; p_{k(min)}$$

The π_k parameters are supplied by the model analyst for projects in general, while the ξ_k is supplied by the model user to relate the extent to which the factor under consideration is present in the project to be simulated.

5.2. Error Generation Rate

The rate that undetected errors are introduced into the product of a given phase is assumed to be proportional to the rate at which the product is being produced. That is, all other things being equal, the error content per unit of product would be the same. Also, the error content is assumed to decrease as the staff comes up on the learning curve, by an amount proportionate to the staff's increase in productivity. Additionally, the error rate depends on the amount of products in precedent phases not yet produced, or produced in error. For example, if the software requirements generated as the product of phase 3 are incomplete or in error, yet phase 5 insists on doing implementation, then there will be a higher likelihood of work being erroneously done in phase 5. Parametrically, the error generation rate takes the form

$$e = (p \; / \; p_s) \; [\varepsilon_0 + (Q_1 + F_1)\varepsilon_1 + \ldots + (Q_{\emptyset-1} + F_{\emptyset-1})\varepsilon_{\emptyset-1} + E_1\eta_1 + \ldots + E_7\eta_7]$$

The quantity ε_0 represents the relative volume of errors that would be introduced in the current phase, even if all precedent work were completed. Each ε_k and η_k reflects an increase in error generation rate in the current phase due to incompleted products $(Q_k + F_k)$ and errors (E_k) of precedent phases. The contribution to error generation due to incompleted products is termed 'speculative error'. The error generation due to precedent errors is 'compounded error'.

This model of error creation presumes that the magnitudes of the product levels and detected fault levels of precedent phases are immediately transmitted to the current phase. While this may not be generally true, preliminary results are often made available at regular intervals. If the project behavior is sensitive to this assumption, each of the Q_k, F_k, and E_k in the error rate equation above may be delayed by a parametric amount, e. g.,

$$Q_k = Q_k(t - \tau_k)$$

The values ε_k and η_k are supplied to the organizational performance submodel by the model analyst, and τ_k comes from the management submodel.

5.3. Fault Detection Rate

The rate at which faults are detected is assumed to be a function of the productivity of the effort devoted to finding errors (QA), the number of errors yet undetected, and the detectability of those errors. The following linear form is postulated:

$$d_{\emptyset} = E_{\emptyset} \{p_{\emptyset} \, \delta_{\emptyset,\emptyset} \, [\xi_{\emptyset,q} \, / \, (1-\xi_{\emptyset,q})] \; + \; \ldots + p_7 \, \delta_{\emptyset,7} \, [\xi_{7,q} \, / \, (1-\xi_{7,q})]\}$$

The δ_k are related to the ease with which a (later) phase k detects an error created in the current phase, and are supplied to the organizational performance submodel by the model analyst.

5.4. Fault Release Rate

The rate at which work detected to be faulty is released back to the product backlog queue depends heavily on management policy and decision. In some instances, work is immediately released to be corrected. In other cases, work may be held for correction in a later software version update. Therefore, the release function, f, is supplied to the organizational performance submodel via the management submodel.

5.5. Rework Rate

Rework here is the process of returning portions of a phase product back to the product backlog queue. Such action is taken when improvements to a phase's products are in order, or when requirements change and a revision is necessary. Both of these situations are management driven, and thus the rework rate function, r, is supplied via the management submodel interface.

5.6. Outside Hire Rate

The hire rate from the external labor pool, h, will also be designated as a function supplied by the management submodel, since the strategy governing hires is a management prerogative.

5.7. Work Reassignment Rate

The rate at which personnel from outside the project are made available for use in the project is controlled by management. Hence, the work reassignment rate, w, emanates from the management submodel.

5.8. Staff Mobility Rate

As with outside hire rate, the mobility strategy, m, or use of available staff, is provided within the management interface.

5.9. Staff Attrition Rate

Staff attrition rates within the job-training pool, the trained staff pool, and available staff pool are assumed to all be the same proportion of the staff levels involved:

$$q_J = q_0 J$$

$$q_S = q_0 S$$

$$q_A = q_0 A$$

The attrition coefficients, q_0, are supplied by the management submodel.

5.10. Staff Inservice Training

The assignment of trained staff to perform inservice training is a management action, whose purpose is to shorten the training period for incoming untrained staff. The assignment is assumed to depend on the number to be trained, and the number available to train them. An inservice assignment rate of

$$i = (\rho J_n - J_t) / \tau_i$$

will bring (asymptotically) a trained-to-untrained personnel ratio (i.e., a teacher/student ratio) of $\rho = J_t / J_n$ into the training pool. The time-constant τ_i reflects the time required to break trained staff free and bring them into the training activity. Both the teacher/student ratio, ρ, and the time constant τ_i are defined within the management submodel.

5.11. Staff Grooming Rate

Personnel receiving job-training from others are assumed to spend a fixed, finite time at their studies, or in classrooms. Thereafter, they commence activities as trained staff. If τ_g represents the nominal time spent in such activities, then the 'grooming' rate is

$$g(t) = h(t - \tau_g) + m(t - \tau_g) + i(t - \tau_g) - q_0 J(t - \tau_g)$$

That is, the transfer of personnel from training activities into productive status is modeled as a mere time delay imposed on the inflow into the training pool. The training time parameter τ_g is defined by the management submodel.

5.12. Staff Assignment Availability

Staff assignment is a management prerogative, so the assignment availability functions, aJ and a_S, are provided by the management submodel.

5.13. Budget Acquisition Rate

Funding acquisition, distribution among phases, and time of activity initiation are management functions, so the budget acquisition rate, b, controlling these characteristics, is provided by the management submodel.

5.14. Budget Expenditure Rate

Budget expenditures in a given phase are primarily due to two factors: personnel and product costs.

$$x = \omega_G G + \omega_p(p + e)$$

The parameter ω_G is the average burdened wage of the group staff, and ω_p is the non-personnel-related production cost per unit product. The latter term includes costs to create errors in the product, as well as to generate correct product. In all phases, ω_p includes documentation. During the implementation and testing phases, it also includes computer utilization and other support costs. The management submodel supplies both ω_G and ω_p.

5.15. Budget Reduction Rate

Projects may, at times, increase or decrease a task's funding allotment. A decrease that is lost to the project is termed the 'budget reduction,' or 'yank' rate. The amounts of reductions and conditions for initiating such events are not part of the organizational response, although the effects within the project caused by such reductions are. The yank function, y, is, therefore, supplied by the management submodel.

5.16. Budget Carryover Rate

Budget carryover, or the release of funding from one phase for use elsewhere within the project, is a management prerogative. The carryover function, c, therefore emanates from the management submodel.

5.17. Budget Carryover Utilization Rate

Reutilization of carryover (and contingency) funds is a management control, and, therefore, the reutilization strategy, u, is supplied by the management submodel.

APPENDIX A
MANAGEMENT SUBMODEL INTERFACE

Parameters needed by the organizational response submodel supplied by the management submodel are contained in this appendix. The parameter subscript k refers to phase k, where k ranges from 1 through 7.

$\xi_{k,q}$	Fraction of effort applied to QA
$\pi_{k,J}$	Training productivity ratio
$\xi_{k,j}$	Extent of productivity factor $j = 1, \ldots, n_{k,p}$
$\tau_{k,j}$	Time delay with which product-information from phase j is known in phase k, for $j < k$
f_k	Fault release rate strategy
r_k	Product rework strategy
h_k	Outside-hire rate
w_k	Staff work reassignment rate from outside the project
m_k	Staff mobility rate
$q_{k,0}$	Staff attrition rate
ρ_k	Training ratio
$\tau_{k,i}$	Inservice instructor break-free time constant
$\tau_{k,g}$	Inservice training grooming time delay
$a_{k,J}$, a_S	Staff reassignment availability
b_k	Budget acquisition rate
$\omega_{k,G}$	Average staff burdened wage
$\omega_{k,p}$	Production costs per-unit of product
y_k	Rebudget yank rate
c_k	Budget carryover
u_k	Budget carryover reutilization

Parameters within the management submodel which interface with the user are:

$P_{k,0}$	Actual amount of product for phase k
W_0	Planned overall work effort, Staff-months
T	Planned project time duration, months
C_0	Planned project cost, constant dollars

APPENDIX B
MODEL ANALYST INTERFACE

Parameters supplied by the model analyst interface are contained in this appendix. The parameter subscript k refers to phase k, where k ranges from 1 through 7.

$p_{k,0}$	Nominal organizational productivity
$\gamma_{k,c}$	Communications overhead time factor
$\pi_{k,0}$	Untrained staff relative productivity
$\tau_{k,s,0}$	Staff self-taught learning time-constant
$\tau'_{k,s}$	Staff inservice learning time-constant slope
$\Delta\tau_{k,s}$	Maximum inservice learning time-constant reduction factor
$\pi_{k,j}$	Productivity adjustment coefficient, factor j, $j = 1, \ldots, n_{k,p}$
$n_{k,p}$	Number of productivity factors
$\varepsilon_{k,0}$	Phase k inherent error generation rate
$\varepsilon_{k,j}$	Speculative error generation rate in phase k due to uncompleted precedent work and uncorrected detected faults in phase j, $j < k$
$\eta_{k,j}$	Compounded error generation rate in phase k due to undetected faults in phase j, $j < k$
$\delta_{k,j}$	Detectability of faults created in phase k during phase j, $j >= k$

REFERENCES

1. Tausworthe, Robert C., McKenzie, Merle, and Lin, Chi Y., 'Structural Considerations for a Software Life Cycle Dynamic Simulation Model', **Proceedings of the AIA Computers in Aerospace IV Conference**, October, Hartford, CT, 1983.

2. Tausworthe, Robert C., McKenzie, Merle, and Lin, Chi Y., 'A Software Life Cycle Dynamic Simulation Model', **Proceedings of the Summer Computer Simulation Conference**, Boston, MA, July, 1983.

3. Reifer, Donald J., 'Interim Report, Software Life Cycle Modeling Investigations,' Report RCI-TR-029, Reifer Consultants, Inc., Hawthorne, CA, January 23, 1984.

4. Reifer, Donald J., 'Rate Equation Derivation and Relationship Portrayal,' Report RCI-TN-068, Reifer Consultants, Inc., Hawthorne, CA, April 23, 1984.

5. Balabanian, Norman, **Network Synthesis**, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1958.

6. Tausworthe, Robert C., 'Staffing Implications of Software Productivity Models', **TDA Progress Report 42-72**, Jet Propulsion Laboratory, Pasadena, CA, October—December1982, pp. 70-77. Also in **Proceedings of the Seventh Annual Software Engineering Workshop**, Goddard Space Flight Center, Greenbelt, MD, December 1982.

SOFTWARE LIFE-CYCLE DYNAMIC SIMULATOR

Robert Tausworthe
Merle McKenzie
Chi Lin

JPL

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

RCT-1
4-16-85

OMIT TO END

PRECEDING PAGE BLANK NOT FILMED

PAGE 1-340 INTENTIONALLY BLANK

JPL

OVERVIEW

- INTRODUCTION
- MODEL STRUCTURE
- THE ORGANIZATIONAL RESPONSE SUBMODEL
- MODEL PARAMETRIZATION
- MANAGEMENT SUBMODEL STUDIES
- FUTURE WORK

RCT-2
4-16-85

JPL

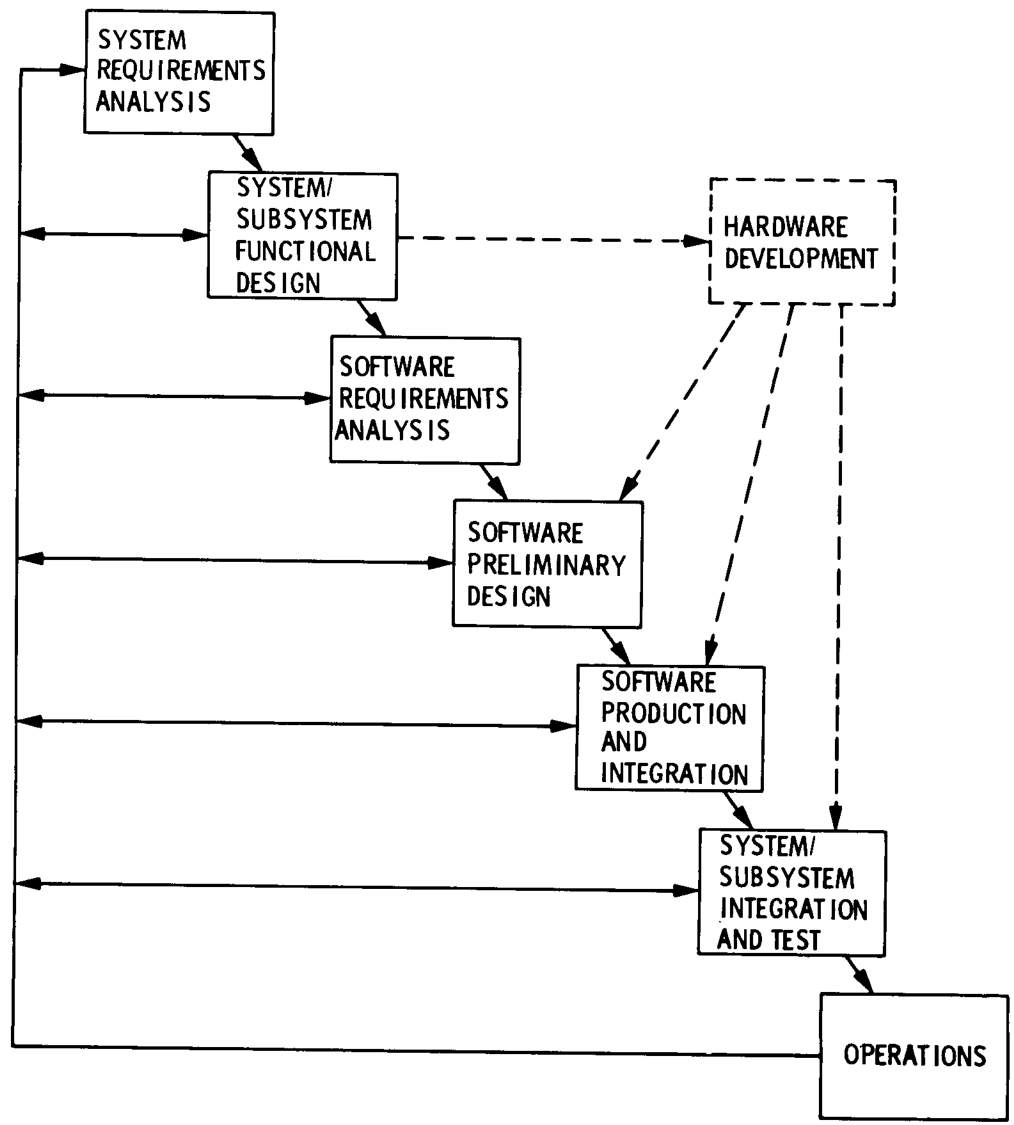
INTRODUCTION

- OBJECTIVE
  - DEVELOP AN ACCURATE DYNAMIC SIMULATION MODEL OF THE SOFTWARE LIFE CYCLE PROCESS
- GOALS
  - REDUCE SOFTWARE BUDGETARY AND SCHEDULE PLANNING UNCERTAINTIES
  - PROVIDE STRATEGIC SUPPORT FOR TECHNOLOGY DEVELOPMENT AND UTILIZATION, AS WELL AS FOR PRODUCTIVITY IMPROVEMENT
  - IMPROVE SOFTWARE TASK PLANNING AND MANAGEABILITY
- APPLICATIONS
  - INITIAL AND DETAILED PLANNING
  - IN-PROCESS REPLANNING
  - SOFTWARE MANAGEMENT TRAINING
  - LIFE CYCLE MONITORING AND CONTROLLING

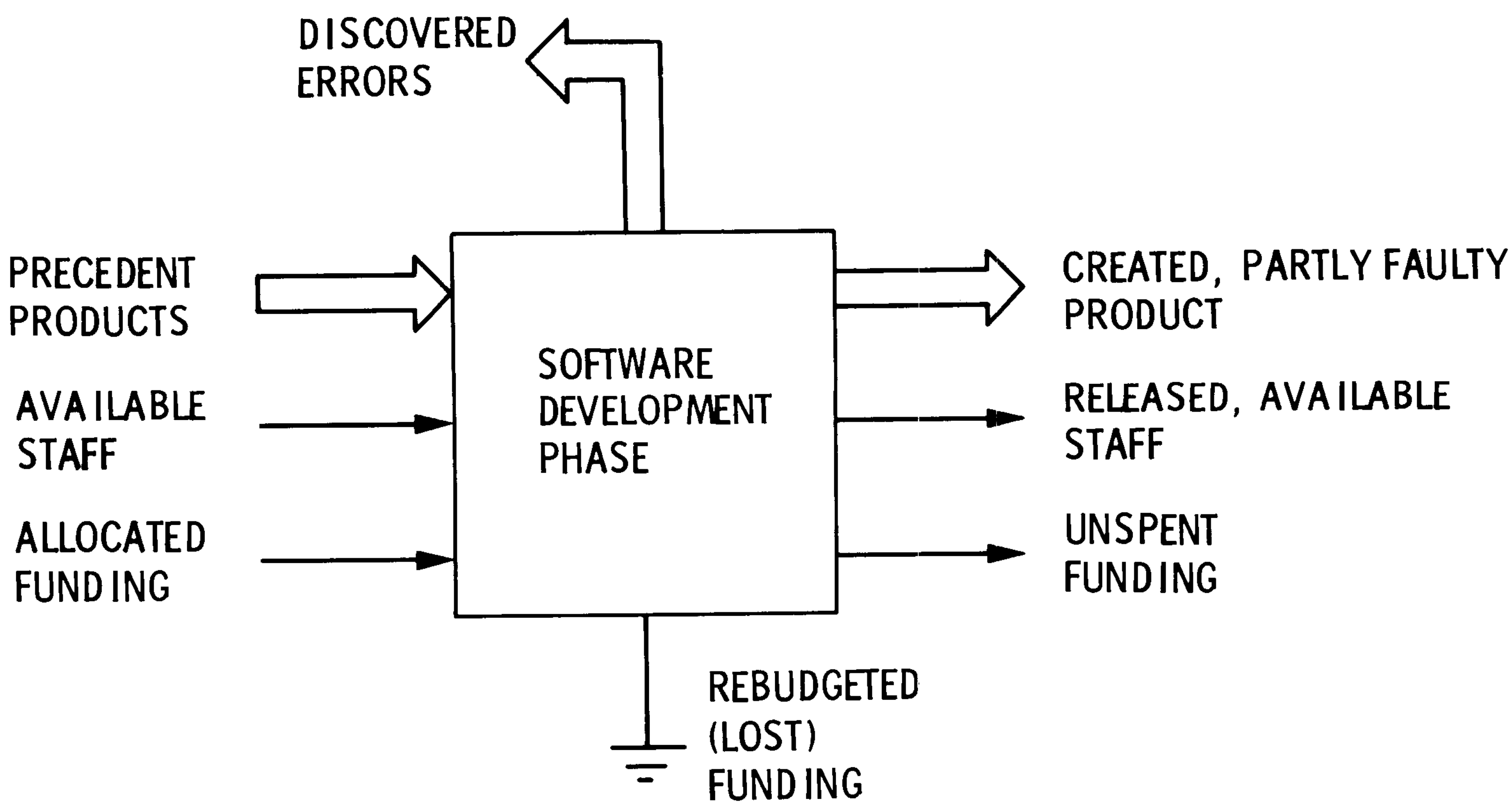
RCT-3
4-16-85

JPL

PHASED—DEVELOPMENT ARCHITECTURE



RCT-4
4-16-85


DISCOVERED ERRORS
PRECEDENT PRODUCTS
AVAILABLE STAFF
ALLOCATED FUNDING
SOFTWARE DEVELOPMENT PHASE
CREATED, PARTLY FAULTY PRODUCT
RELEASED, AVAILABLE STAFF
UNSPENT FUNDING
REBUDGETED (LOST) FUNDING

RCT-5
4-16-85

JPL

QUANTITY LEVEL, PER PHASE

- PRODUCT
  - AMOUNT CORRECT, TO DATE (NOT DIRECTLY PERCEIVABLE)
  - AMOUNT IN ERROR, AS-YET UNDETECTED
  - AMOUNT KNOWN IN ERROR, NOT YET BEING REPAIRED
- STAFF
  - AVAILABLE STAFF
  - STAFF IN/PERFORMING TRAINING
  - STAFF IN PRODUCTION
- FUNDING
  - ALLOCATED BUDGET
  - COST-TO-DATE
  - CARRYOVER/CONTINGENCY FUND

RCT-6
4-16-85

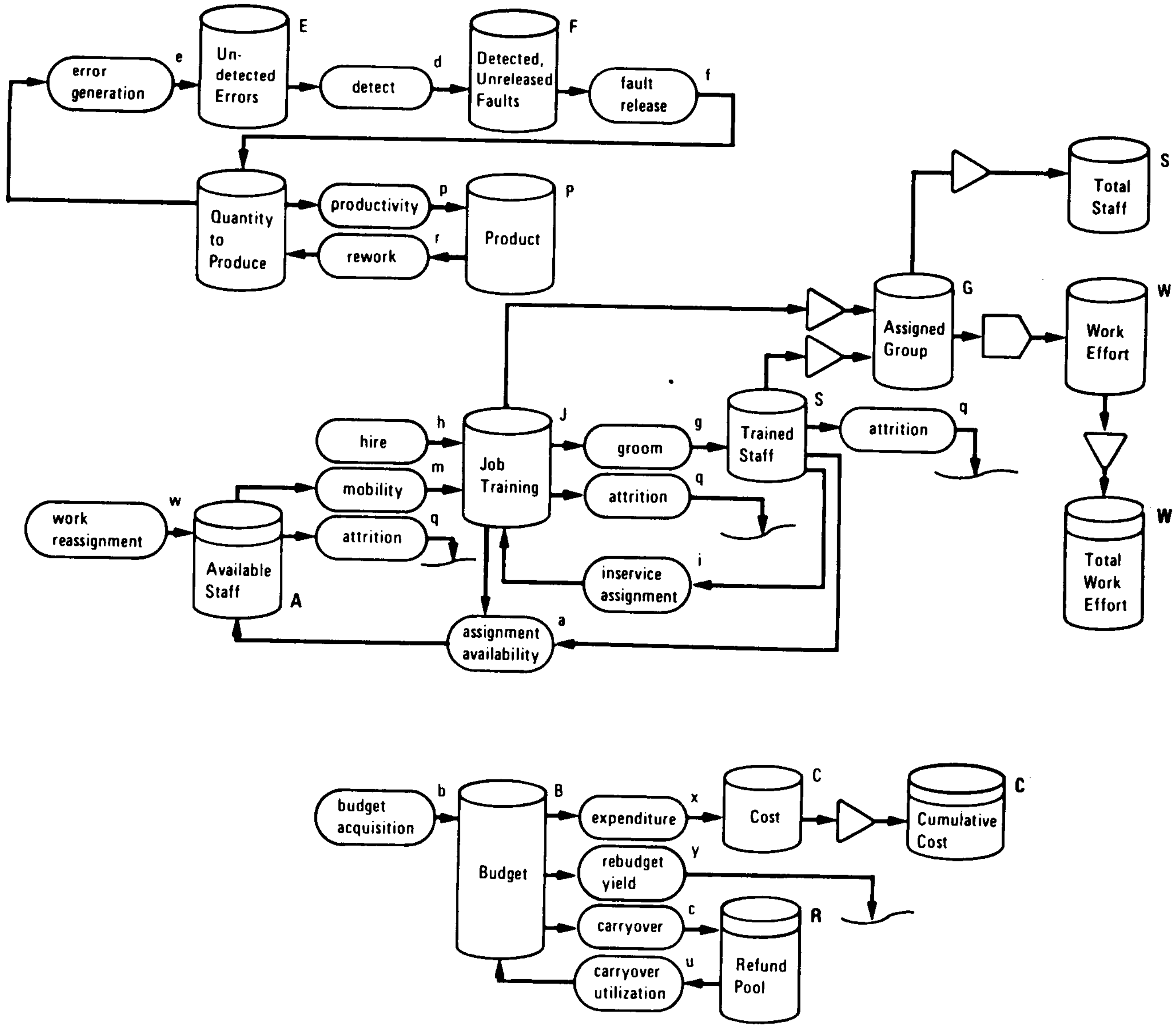
JPL

QUANTITY CHANGE RATES, PER PHASE

- PRODUCT
  - STAFF PRODUCTIVITY
  - ERROR GENERATION
  - ERROR DETECTION
  - FAULT RELEASE
  - PRODUCT REWORK
- STAFF
  - WORK REASSIGNMENT AVAILABILITY
  - OUTSIDE HIRE ACQUISITION
  - STAFF MOBILITY
  - ATTRITION
  - TRAINING (STUDENT) ASSIGNMENT
  - TRAINING (INSTRUCTOR) ASSIGNMENT
- FUNDING
  - BUDGET ACQUISITION
  - EXPENDITURE
  - REBUDGET (LOSS) RATE
  - CARRYOVER/CONTINGENCY ACQUISITION
  - UTILIZATION OF CARRYOVER/CONTINGENCY

RCT-7
4-16-85

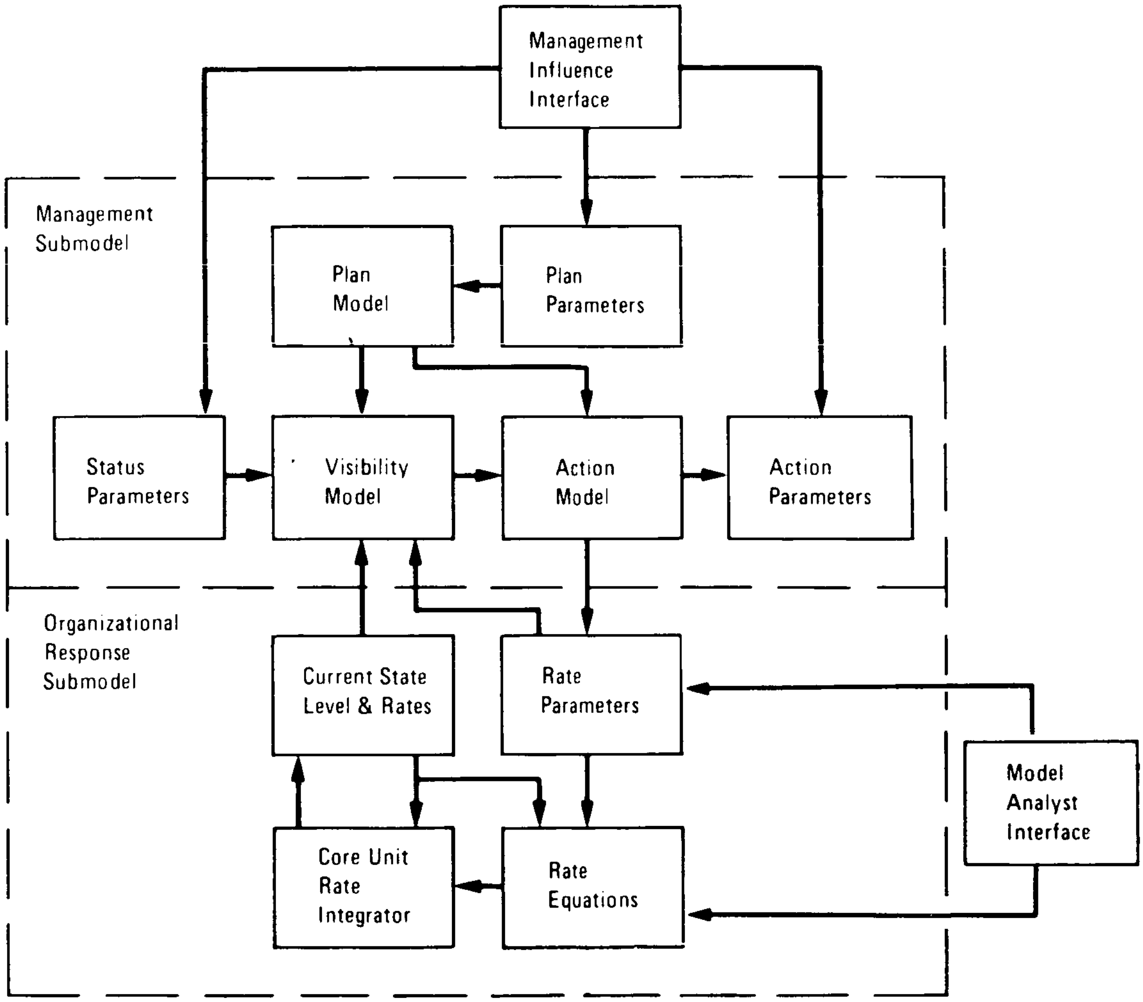
JPL



RCT-8
4-16-85

JPL

MODEL STRUCTURE



RCT-9
4-16-85

JPL

MODEL ANALYST INTERFACE, PER PHASE

- NOMINAL STAFF PRODUCTIVITY PARAMETER
- COMMUNICATIONS OVERHEAD EQUATION PARAMETER
- UNTRAINED/TRAINED STAFF PRODUCTIVITY RATIO
- STAFF SELF-TAUGHT LEARNING TIME CONSTANT
- RATE OF CHANGE IN LEARNING TIME CONSTANT WRT LEVEL-OF-INSTRUCTION
- MAXIMUM LEARNING TIME REDUCTION FACTOR
- MAXIMUM PRODUCTIVITY CHANGES FOR GIVEN ENVIRONMENT/SITUATION FACTORS
- INHERENT ERROR GENERATION RATE
- SPECULATIVE ERROR GENERATION RATE
- COMPOUNDED ERROR GENERATION RATE
- ERROR DETECTABILITY IN PRECEDENT AND CURRENT PRODUCTS

RCT-10
4-16-85

JPL

EXAMPLES

ERROR GENERATION RATE = (PRODUCTION RATE) / (LEARNING CURVE)

* {(INHERENT ERROR PROPENSITY)

$+\sum_{\text{PRECEDENT}}$ [(INCOMPLETED PRECEDENT) * (SPECULATIVE ERROR PROPENSITY)

\+ (PRECEDENT IN ERROR) * (COMPOUND ERROR PROPENSITY)

FAULT DETECTION RATE = (UNDETECTED - ERROR CONTENT)

$*\sum_{\text{SUCCESSOR}}$ (PRODUCTION RATE) * (FRACTION-IN-FAULT DETECTION)

* (FAULT DETECTION PROPENSITY)

RCT-11
4-16-85

JPL MANAGEMENT SUBMODEL INTERFACE, PER PHASE

- FRACTION OF EFFORT FOR TESTING, ERROR DETECTION, QA
- TRAINING/WORK-TIME RATIO
- EXTENT OF ENVIRONMENTAL/SITUATIONAL FACTORS
- TIME DELAY OF PHASE INFORMATION PROPAGATION
- FAULT RELEASE STRATEGY
- PRODUCT REWORK NECESSITY (E. G. REQUIREMENTS CHANGE)
- OUTSIDE-HIRE STRATEGY/PLAN
- STAFF ASSIGNMENT TO AVAILABLE STAFF
- MOBILIZATION OF AVAILABLE STAFF STRATEGY/PLAN
- ATTRITION RATE
- TRAINING INSTRUCTOR/STUDENT RATIO
- INSTRUCTOR BREAK-FREE TIME CONSTANT
- TRAINING TIME
- STAFF RELEASE TO AVAILABLE STAFF STRATEGY/PLAN

RCT-12
4-16-85

JPL

MANAGEMENT SUBMODEL INTERFACE (CONT'D)

- BUDGET ACQUISITION PLAN/STRATEGY
- AVERAGE STAFF BURDENED WAGE
- PRODUCTION COST PER UNIT PRODUCT
- REBUDGET (LOSS) NECESSITY (E.G. BUDGET CUT)
- BUDGET CARRYOVER STRATEGY/PLAN
- CARRYOVER/CONTINGENCY UTILIZATION STRATEGY/PLAN

RCT-13
4-16-85

JPL

MANAGEMENT MODEL DEVELOPMENT PROCESS

- DEVELOPED A BROAD-BASED QUESTIONNAIRE TO ELICIT HEURISTIC KNOWLEDGE FROM EXPERTS. ADDRESSED:
  - ALL PHASES OF DIFFERENT PROJECTS UNDER DIFFERENT MANAGERS
  - HISTORICAL OBSERVATIONS (DATA POINTS) AND UNDERSTANDING OF THE PROCESS, AS WELL AS PREDICITIVE HEURISTICS
  - ALTERNATE MANAGEMENT STYLES AND ALTERNATE CONSTRAINT SCENARIOS
- INTERVIEWED MANAGERS AND STAFF
  - PROGRAMMATIC MANAGERS
  - DIVISION-LEVEL MANAGERS
  - SECTION-LEVEL MANAGERS
  - TASK MANAGERS
  - COGNIZANT DEVELOPMENT ENGINEERS
  - PROGRAMMERS
- ANALYZED INFORMATION AND NOTED THAT
  - THERE WERE SURPRISINGLY MANY AREAS OF CONSISTENCY
  - DIFFERENCES TENDED TO INDICATE NEED FOR DIFFERENT SCENARIOS
  - STRONG INTEREST WAS EXPRESSED IN THE MODEL
- FOLDED KNOWLEDGE INTO A FUNCTIONING TWO-PHASE MODEL (S/W REQUIREMENTS, ARCHITECTURAL DESIGN) BASED ON SELECTED SCENARIOS
- TESTED MODEL STRUCTURE, PARAMETER VALUES, AND BEHAVIOR

CYL/MM-1

JPL MANAGEMENT SUBMODEL COMPONENTS

- PLAN MODEL AND PARAMETERS
  - STAFF PLAN
  - MILESTONE PLAN
  - BUDGET PLAN
- VISIBILITY (STATUS) MODEL AND PARAMETERS
  - EXISTING STAFF COMPARED TO PLANNED STAFF
  - MILESTONES COMPLETED COMPARED TO SCHEDULED MILESTONES
  - AMOUNT OF BUDGET SPENT COMPARED TO BUDGET PLANNED
  - WORK COMPLETED IN THE PREVIOUS PHASE
  - VALUES OF ALL LEVELS AND RATES, FORMING INFORMATION UPON WHICH DECISIONS ARE MADE

CYL/MM-2

JPL

MANAGEMENT SUBMODEL COMPONENTS (Con't)

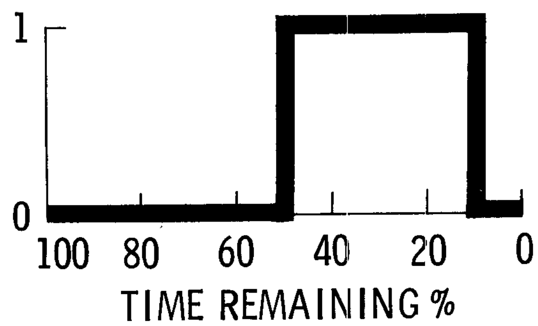
- DYNAMIC ACTION (STRATEGY AND CONTROL) MODEL AND PARAMETERS
  - MANAGEMENT STRATEGY ON STAFFING ALGORITHM (FUNCTION OF SCENARIO CHOSEN)
  - DEGREE TO WHICH REQUIREMENTS ARE ALLOWED TO CHANGE
  - TRAINING STRATEGY, AND INPUTS THERETO
  - STRATEGY FOR APPLYING QA, AND INPUTS THERETO
  - MANAGEMENT STRATEGY ON PHASE OVERLAP AND EFFECT ON ERROR GENERATIONS
  - ERROR RELEASE STRATEGY AS IMPACTED BY CRITICALITY OF ERRORS
  - STAFF SOURCE STRATEGY AND INPUTS THERETO
  - MANAGEMENT WILLINGNESS TO HIRE AS A FUNCTION OF TIME REMAINING
  - STAFF RELEASE STRATEGY: BASED ON STAFF PLAN, BASED ON SLACK TIME
  - MANAGEMENT POLICY ON WORKER ENVIRONMENT, AND INPUTS THERETO

CYL/MM-3

JPL

EXAMPLES OF MANAGEMENT CONTROL PROCESS

- ADDITIONAL HIRE RATE = DELAY {[(ACTUAL PRODUCT PRODUCED - SCHEDULED MILESTONES) * (NOMINAL INDV. PRODUCTIVITY * TIME REMAINING)] * MANAGER'S WILLINGNESS TO HIRE}



- ADDITIONAL BUDGET ACQUISITION RATE = DELAY {(COST-BUDGET) * f_1 (t) * DEGREE OF MATRIX MANAGEMENT * SCHEDULE PRESSURE FACTOR}

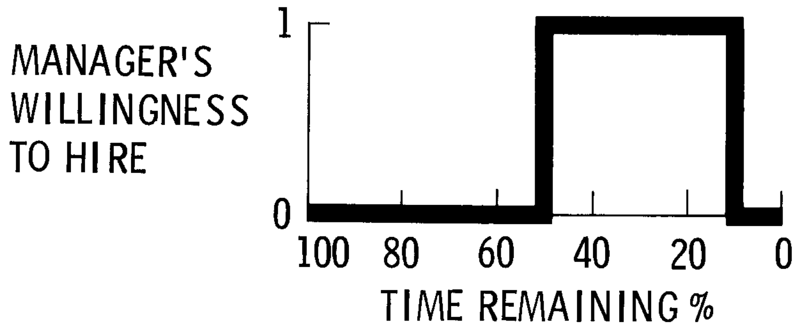
$$f_1(t) = \begin{cases} 1 & \text{WHEN AT START OR MIDDLE OF FISCAL YEAR} \\ 0 & \text{ELSE} \end{cases}$$

CYL/MM-4

JPL

EXAMPLES OF MANAGEMENT CONTROL PROCESS

- ADDITIONAL HIRE RATE = DELAY {[(SCHEDULED MILESTONES - ACTUAL PRODUCT PRODUCED) ÷ (NOMINAL INDV. PRODUCTIVITY * TIME REMAINING)] * MANAGER'S WILLINGNESS TO HIRE}



- ADDITIONAL BUDGET ACQUISITION RATE = DELAY {(COST-BUDGET) * f_1 (t), DEGREE OF MATRIX MANAGEMENT}

$$f_1(t) = \begin{cases} 1 & \text{WHEN AT START OR MIDDLE OF FISCAL YEAR} \\ 0 & \text{ELSE} \end{cases}$$

CYL/MM-4

JPL

FUTURE WORK

- DETERMINATION AND REFINEMENT OF MODEL ANALYST PARAMETER VALUES
- DEVELOPMENT OF PARAMETRIZED MANAGEMENT SUBMODEL
- DETERMINATION AND REFINEMENT OF MANAGEMENT SUBMODEL PARAMETER VALUES
- DEVELOPMENT OF MANAGER USER INTERFACE

RCT-17
4-16-85