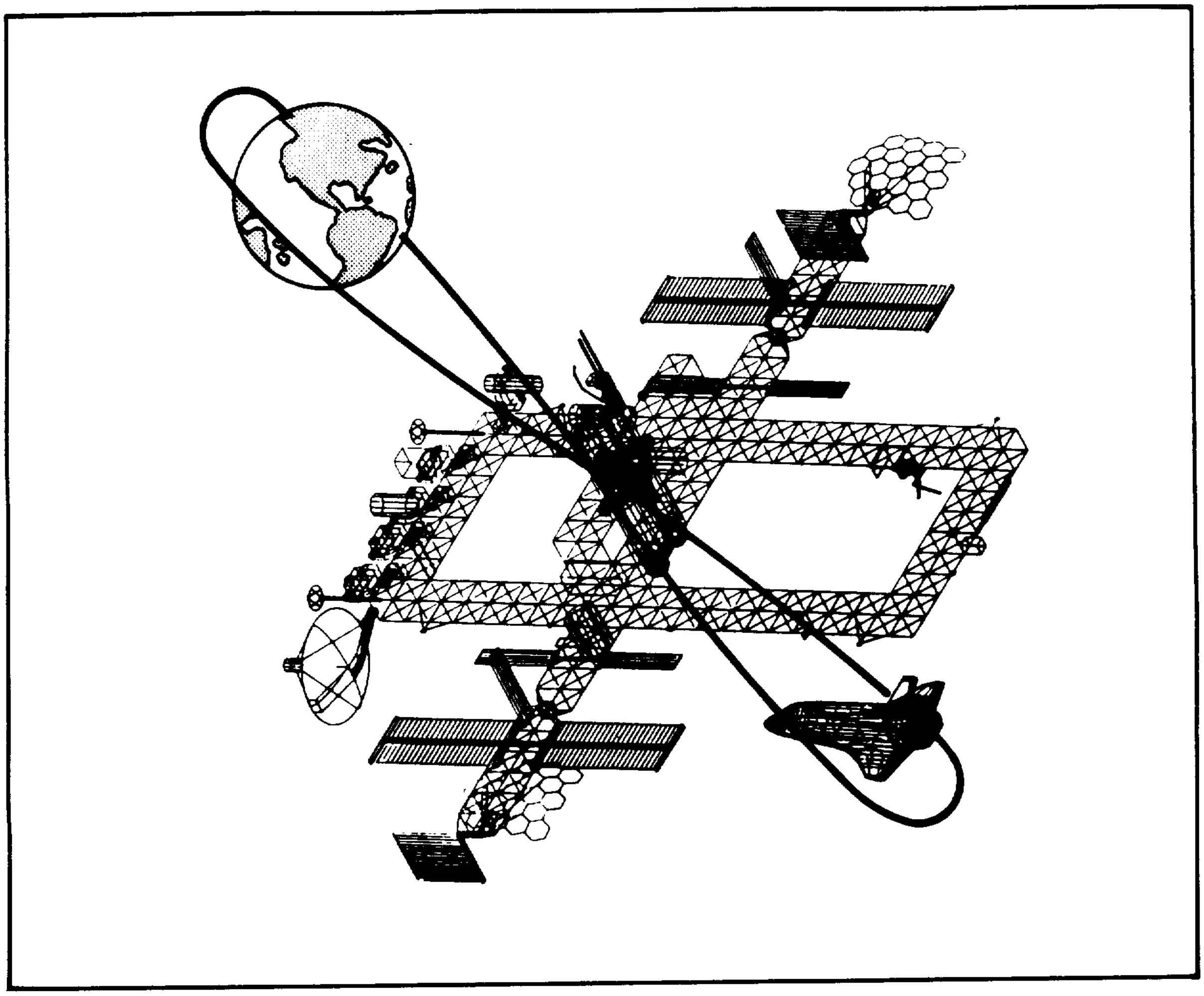


RICIS '87 SYMPOSIUM

(NASA-CR-187906) RICIS 1987 SYMPOSIUM. EXECUTIVE SUMMARY (Houston Univ.) 346 p CSCL 09B

N91-18616 --THRU-- N91-18623

Unclas

G3/61 0320439

Research Institute for Computing and Information Systems

Executive Summary of Symposium '87

A.G. Houston
editor

RESEARCH INSTITUTE FOR COMPUTING AND INFORMATION SYSTEMS

Executive Summary of Symposium '87

Houston, Texas
October 14-15, 1987

edited by
A. Glen Houston
Director, RICIS
University of Houston-Clear Lake

PREFACE

This document summarizes the proceedings of RICIS Symposium '87, which was held at the University of Houston - Clear Lake on October 14-15, 1987.

This symposium was sponsored by the Research Institute for Computing and Information Systems of the University of Houston - Clear Lake. RICIS was founded in 1986 to encourage the NASA Johnson Space Center and local industry to actively support research in the computing and information sciences. In May, 1986, UH - Clear Lake and JSC entered into a three-year cooperative agreement to jointly plan and execute such research through RICIS.

This symposium was held to present the first year's research activities conducted through RICIS as well as to introduce the JSC/UH -Clear Lake cooperative research program. The program consisted of one and a half-days of sessions. Wednesday afternoon was devoted to presenting the RICIS concept and an overview of the research being conducted. On Thursday, there were five technical sessions featuring researchers presenting their research results and near-term plans. The five technical areas addressed were Education and Training, Computer Systems and Software Engineering, Information Management, Mathematical and Statistical Analysis and Artificial Intelligence and Expert Systems.

It was decided to publish an executive summary of this symposium, rather than the papers presented, since in most cases the researchers were not at a point in their research to publish results. It is noted that final reports of the research will be published and may be obtained from the RICIS Project Office as they become available.

A large number of people helped make RICIS Symposium '87 a big success. The organizing committee included Peter C. Bishop, Sam J. Bruno, Terry Feagin, Glenn B. Freedman, Cecil R. Hallum, Robert F. Hodgin and Charles W. McKay. A special thanks is extended to Sam J. Bruno for tending the many details required to make such an event a success.

The RICIS Project Office staff, particularly Kerry Ellison, Vickie Gilliland and Bassanio Wong, aided by other UH - Clear Lake staff , notably, Jean Hart, Mary Jo Westover, Olga Gonzales and Melinda Goyne, oversaw the correspondence and bookkeeping, maintained a participant database, assembled registration packets, and manned the registration booths. E.T. Dickerson, Dean of the School of Natural and Applied Sciences, UH - Clear Lake and R.B. MacDonald, Assistant to the Director for Technology Utilization Mission Support, Directorate, NASA Johnson Space Center, were very supportive and provided welcome guidance for conducting this symposium. Thanks are also extended to Amy Kennedy, Employee Development, NASA/JSC, and Connye McLendon, Administrative Office, Mission Support Directorate, NASA/JSC for working the registration and transportation issues, respectively, for JSC employees to attend this symposium.

Financial support for the Symposium came from NASA Johnson Space Center through Cooperative Agreement NCC 9-16.

A. Glen Houston
Director, RICIS

Contents

WELCOME AND OPENING

Thomas M. Stauffer, President, UH-Clear Lake

Paul J. Weitz, Deputy Director, NASA Johnson Space Center

The RICIS Symposium '87 opened with remarks by Dr. Thomas Stauffer, president of the University of Houston-Clear Lake, and by Paul J. Weitz, Deputy Director of NASA Johnson Space Center. Dr. Stauffer drew an analogy between software engineering research and the emerging superconductor technology. He indicated that while cooperation between academia and industry is sometimes difficult, joint research, like that which is being conducted through RICIS, is of significance to Houston, to Texas and to the entire nation.

Mr. Weitz discussed the importance of software systems to the space program. He said that without the appropriate major software systems, spacecraft design, flight simulator, mission control and the Space Shuttle are not possible.

JSC/UH-CLEAR LAKE COOPERATIVE RESEARCH PROGRAM

Joseph P. Loftus, Jr., Assistant Director for Plans, NASA Johnson Space Center

Charles S. Hardwick, Senior Vice President and ProvostUH-Clear Lake

Joseph P. Loftus, Assistant Director of Plans for JSC, and Dr. Charles Hardwick, Senior Vice President and Provost of UH-Clear Lake, offered comments about the cooperative nature of the JSC/UH-Clear Lake research program. Mr. Loftus discussed the space business as being extremely information-intensive. He pointed out that, aside from the returning Space Shuttle, an information stream is currently the only product we obtain from space. Hence, solutions are to be found in software.

Dr. Hardwick discussed the need for a "major revolution" in the methods by which managers and engineers are educated. Educational systems and curricula need to be developed to allow them to better cope with the complexity of software and to ensure the success of the space program.

THE RICIS CONCEPT

E.T. Dickerson, Dean, School of Natural and AppliedSciences, UH-Clear Lake

Robert B. MacDonald, Assistant to the Director for Technology Utilization, Mission Support Directorate, NASA Johnson Space Center

A. Glen Houston, Director, RICIS, UH-Clear Lake

The RICIS Concept

Dr. E.T. Dickerson, Dean of the School of Natural and Applied Sciences, UH-Clear Lake, Mr. Robert B. MacDonald, Assistant to the Director for Technology Utilization, Mission Support, Directorate, JSC and Dr. A. Glen Houston, Director of RICIS, UH-Clear Lake discussed the history and objectives of RICIS.

RICIS SYMPOSIUM '87
PROGRAM

WEDNESDAY, OCTOBER 14
BAYOU BUILDING

12:30-5:00	**REGISTRATION**	**ATRIUM 1**
1:30-1:45	WELCOME AND OPENING Thomas M. Stauffer, President, UH-Clear Lake Paul J. Weitz, Deputy Director, NASA Johnson Space Center	AUDITORIUM
1:45-2:05	JSC/UH-CLEAR LAKE COOPERATIVE RESEARCH PROGRAM Joseph P. Loftus, Jr., Assistant Director for Plans, NASA Johnson Space Center Charles S. Hardwick, Senior Vice President and Provost UH-Clear Lake	AUDITORIUM
2:05-2:50	THE RICIS CONCEPT E. T. Dickerson, Dean, School of Natural and Applied Sciences, UH-Clear Lake Robert B. MacDonald, Assistant to the Director for Technology Utilization, Mission Support Directorate, NASA Johnson Space Center A. Glen Houston, Director, RICIS, UH-Clear Lake	AUDITORIUM
2:50-3:05	**REFRESHMENT BREAK**	**AUDITORIUM FOYER**
3:05-4:45	RICIS RESEARCH **Computer Systems and Software Engineering** **Charles W. McKay, Director, High Technologies Laboratory and Software Engineering Research Center, UH-Clear Lake** **Artificial Intelligence and Expert Systems** **Terry Feagin, UH-Clear Lake** **Information Systems** **Peter C. Bishop, UH-Clear Lake** **Mathematical and Statistical Analysis** **Cecil R. Hallum, UH-Clear Lake** **Education and Training** **Glenn B. Freedman, Director, Center for Cognition and Instruction, UH-Clear Lake**	AUDITORIUM
4:45-5:00	**REFRESHMENT BREAK**	**AUDITORIUM FOYER**

5:00-5:30	**INVITED TALK** Lee B. Holcomb, Director, Human Sciences and Human Factors, OAST, NASA Headquarters **CLOSING REMARKS** Robert F. Hodgin, UH-Clear Lake	**AUDITORIUM**
6:30-7:30	**RECEPTION - CASH BAR**	**GILRUTH CENTER** Second Floor
7:30-8:30	**DINNER**	**GILRUTH CENTER**
8:30-9:00	**KEYNOTE ADDRESS** *The Real Technologies in Space Station Information Systems* John R. Garman, Director of Information Systems Services Space Station Program Office, NASA Headquarters	**GILRUTH CENTER**

THURSDAY OCTOBER 15

8:00-8:30	**REGISTRATION/CONTINENTAL BREAKFAST**	**ATRIUM 1**
8:30-8:35	**RICIS RESEARCH AREAS** Robert F. Hodgin, UH-Clear Lake	**ROOM 2-532**
8:35-9:45	**EDUCATION AND TRAINING** Conveners: Glenn B. Freedman, UH-Clear Lake Amy B. Kennedy, Employee Development, NASA/JSC *Review of the Education and Training Activities* Glenn B. Freedman, UH-Clear Lake *Software Engineering and the Transition to Ada* John McBride, SofTech, Inc. *Computer Based Ada Training Using Hypertext Systems* Jack Rienzo and Robert Wallace, SofTech, Inc.	**ROOM 2-532**
9:45-10:00	**REFRESHMENT BREAK**	**ROOM 2-532 FOYER**

10:00-12:00 **COMPUTER SYSTEMS AND SOFTWARE ENGINEERING** **ROOM 2-532**

Conveners: Charles W. McKay, UH-Clear Lake

Stephen A. Gorman, Head, Application Systems,
Spacecraft Software Division, NASA/JSC

Fault Tolerant Ada Software
Pat Rogers, UH-Clear Lake

A Study of Converting PCTE System Specifications to Ada
Kathy Rogers, Rockwell International

Proof-of-Concept Prototype of the Clear Lake Model for Ada Run Time Support Environment
Charles Randall, GHG Corporation

Testing And Verification of Ada Flight Software for Embedded Computers
David Auty, SofTech, Inc.

Ada Programming Support Environment Data Base
Anthony Lekkos, UH-Clear Lake

12:00-1:00 **BUFFET LUNCHEON** **FOREST ROOM**

1:00-2:30 **INFORMATION MANAGEMENT**

Conveners: Peter C. Bishop, UH-Clear Lake

William J. Huffstetler, Assistant to the Director, Engineering, NASA/JSC

The Need for Strategic Information at JSC
William J. Huffstetler, NASA/JSC

Research Projects in Information Management
Peter C. Bishop, UH-Clear Lake

Database Strategies and Prototypes
Timothy N. Tulloch, Vice President, TNT Consulting

Space Station
Documentation Technology and Strategies
Christopher Dede, Professor of Education, UH-Clear Lake

Future Research Opportunities
Lloyd R. Erickson, Electronics Engineer, NASA/JSC

2:30-2:45 **REFRESHMENT BREAK** **ROOM 2-532 FOYER**

2:45-3:30 **MATHEMATICAL AND STATISTICAL ANALYSIS**

Conveners: Cecil R. Hallum, UH-Clear Lake

David K. Geller, Mission Planning and Analysis Division, NASA/JSC

Space Station Momentum Management and Attitude Control
Bong Wie, University of Texas at Austin

Quantifying Software Reliability (Invited Presentation)
Patrick L. Odell, University of Texas at Dallas

3:30-5:40 **ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS**

Conveners: Terry Feagin, UH-Clear Lake

Timothy F. Cleghorn, Mission Planning Analysis Division, NASA/JSC

Introduction and Overview
Timothy F. Cleghorn

Communication and Tracking Expert Systems for the NASA Space Station
T.F. Leibfried, UH-Clear Lake

Simulation of Robotic Space Systems
Yashvant Jani, LinCom Corporation

Robotic Path Planning and Software Testbed Architecture
Richard D. Volz, University of Michigan

Fuzzy Set and Related Theory for Failure Detection and Control in Space Systems
Thomas B. Sheridan, Massachusetts Institute of Technology

A Computer Graphics Testbed to Simulate and Test Vision Systems for Space Applications
John B. Cheatham, Jr., Rice University

Demonstration of a 3D Vision Algorithm for Space Applications
Rui J.P. deFigueiredo, Rice University

CLOSING REMARKS

A. Glen Houston, UH-Clear Lake

N91-18617

The RICIS Concept

Robert B. MacDonald

In May 1986, JSC initiated a cooperative program with the University of Houston - Clear Lake to support research in computing and information systems. The objective of this program was and is to provide continuing long-term research in support of the numerous mission and mission-related endeavors of NASA/JSC. JSC defined a "cooperative agreement" as the appropriate contractual vehicle to facilitate both joint participation of researchers from NASA, industry and the university community, and sharing of supporting research facilities among the participants. Facilities are shared by networking among UH-CL's research and data computing resources and JSC's computing system.

A significant part of the cooperative program is its "gateway role." UH-CL is chartered to involve researchers from outside organizations throughout the US and the world in projects defined by professionals at NASA and UH-CL.

A particularly important set of activities being carried out under the cooperative program is in the area of "computer software development." Because it recognized these activities as a critical element of the cooperative program, the Office Aeronautics and Space Technology at NASA Headquarters supported the initiation of the Software Engineering Research Center (SERC).

In order to meet research and education needs associated with the engineering of large, real-time software systems for NASA's future numerous researchers at SERC are investigating:

1. engineering research issues central to large distributed systems for real-time and distributed systems with active embedded elements (such as for thespace station)

2. concepts, principles and methodologies for the engineering of such large software systems

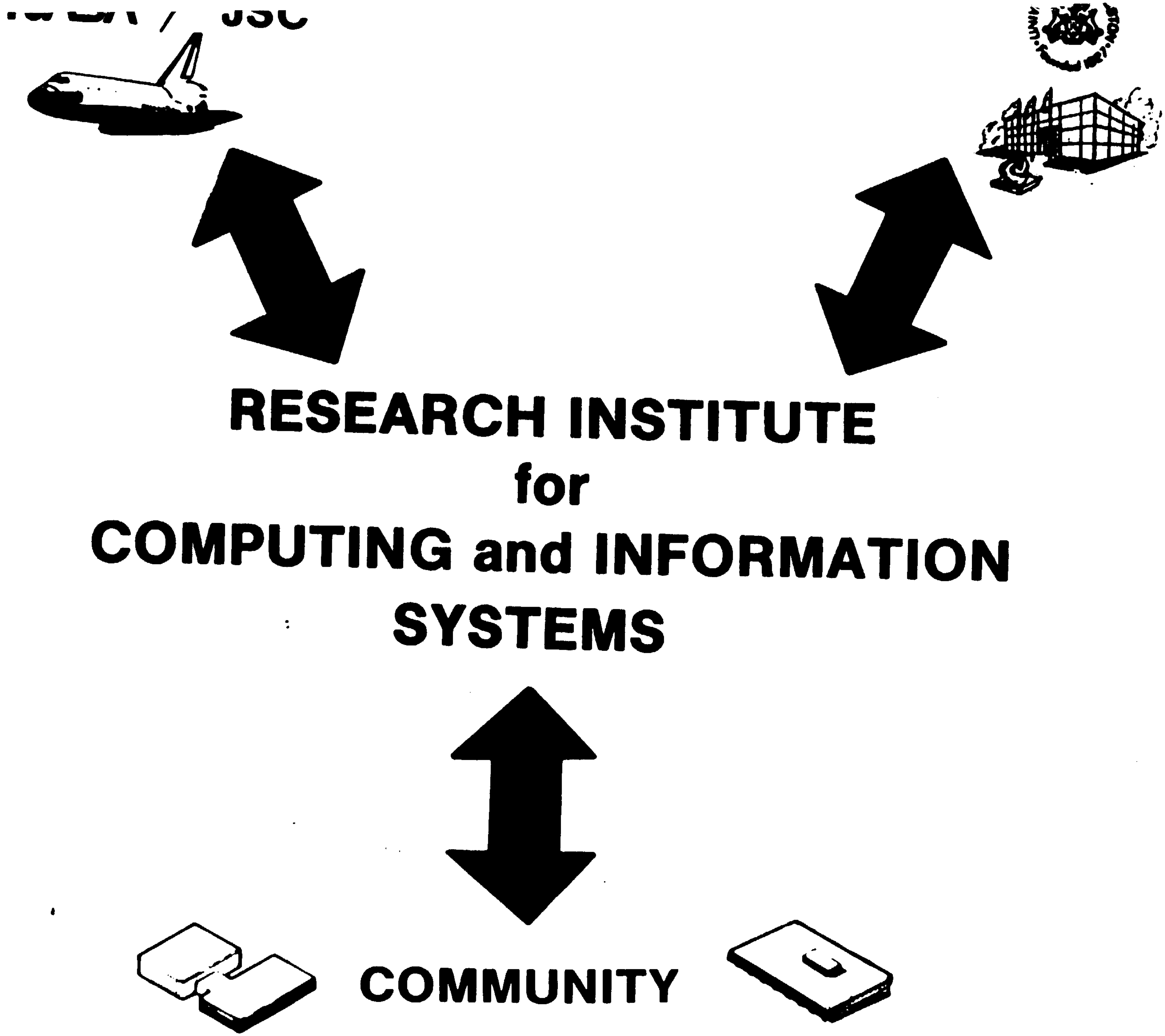
3. "computer aided software engineering environments" to advance the state of thepractice to achieve improvements in the quality productivity crucial to the application of engineering methodologies to software lifecycle phases

4. the establishment and incorporation into of future systems of appropriate standards.

With the growth of software development activities, comes the increase need for education. Currently most managers and professionals continue to emphasize the implementation and test phases of the "software lifecycle." More disciplined engineering approaches require that this managerial and professional workforce be educated in approaches to software development which emphasize requirement and design phases and designing for change.

The evolution of engineering methodologies and tools such as CASE and languages like ADA over the last ten years has created a severe shortage of individuals who are technically and emotionally prepared to exploit these advances. In response, the School of Education and the School of Natural and Applied Sciences at UH-Clear Lake have established the Software Engineering Professional Education Center (SEPEC). The objective of this new center is to interact with SERC and other organizations throughout the US, such as the SEI at Carnegie Mellon University, to develop and bring about suitable education and training at both professional and academic levels.

In short, the central goal of the SERC and SEPEC is to develop and make available the Engineering Knowhow, the qualified human resources and supporting to tools and rules to better "engineer large, distributed, real-time software systems of the future."


JSC
RESEARCH INSTITUTE
for
COMPUTING and INFORMATION
SYSTEMS
COMMUNITY

RICIS SYMPOSIUM '87

October 14, 1987

THE RICIS CONCEPT

by

E. T. DICKERSON

Dean, School of Natural & Applied Sciences, UH-Clear Lake

ROBERT B. MACDONALD

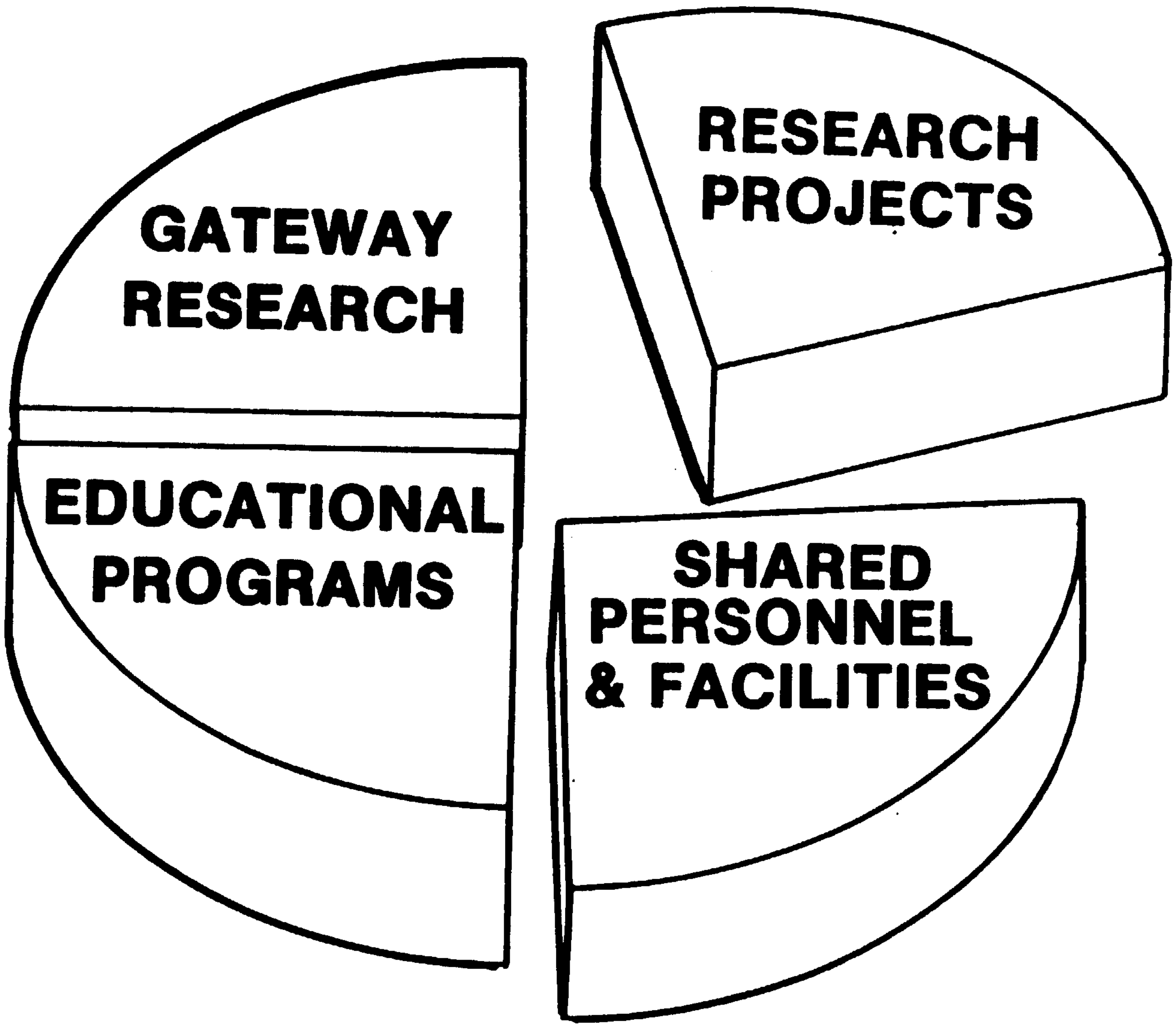
Mission Support Directorate, NASA Johnson Space Center

A. GLEN HOUSTON

Director, RICIS, UH-Clear Lake

ACTIVITIES

for
COMPUTING and INFORMATION SYSTEMS



BENEFITS

RESEARCH INSTITUTE for COMPUTING and INFORMATION SYSTEMS

- FOCUS FOR RESEARCH
- EVALUATION OF NEW TECHNOLOGIES
- INVESTIGATION OF STANDARDS
- DISSEMINATION OF INFORMATION
- COORDINATION OF EFFORT
- COST-EFFECTIVE USE OF FACILITIES

HISTORY

EVENTS LEADING TO RICIS

- TASK FORCE ESTABLISHED SUMMER 1983
  (HARDWICK - COHEN)
- MEMORANDUM OF UNDERSTANDING - NOVEMBER 1983
  (GRIFFIN - STAUFFER)
- TASK FORCE ESTABLISHED FOUR AREAS OF POSSIBLE COOPERATION
  - COMPUTERS
  - HUMAN PERFORMANCE
  - EDUCATION AND TRAINING
  - R&D MANAGEMENT

EVENTS LEADING TO RICIS
(CONT)

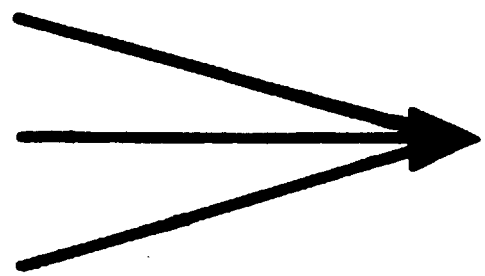
- RICIS CONCEPT ESTABLISHED BY UHCL-SST
- NASA - UHCL - SST
- RICIS CONCEPT SCOPE EXPANDED (1984)
- DIRECTOR APPOINTED (DEC 1985)
- UHCL STEERING COMMITTEE ESTABLISHED (JAN 1986)
- PROPOSAL TEAM APPOINTED (JAN 1986)
- UNSOLICITED PROPOSAL SUBMITTED TO NASA/JSC (MAR 1986)

ORGANIZATION

MISSION

RESEARCH INSTITUTE
for
COMPUTING and INFORMATION
SYSTEMS

- CONDUCT
- COORDINATE
- DISSEMINATE



RESEARCH

MISSION

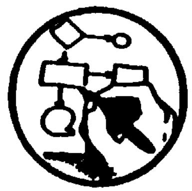
RESEARCH INSTITUTE
for
COMPUTING and INFORMATION
SYSTEMS

- CONDUCT
- COORDINATE
- DISSEMINATE

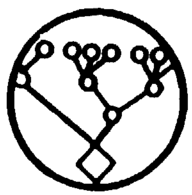
→ RESEARCH

AGENDA

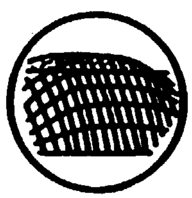
RESEARCH INSTITUTE for COMPUTING and INFORMATION SYSTEMS



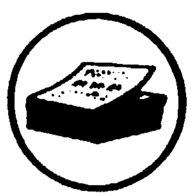
- COMPUTER SYSTEMS AND SOFTWARE ENGINEERING



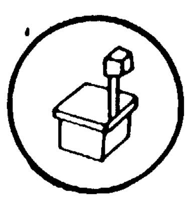
- ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS



- MATHEMATICAL AND STATISTICAL ANALYSIS

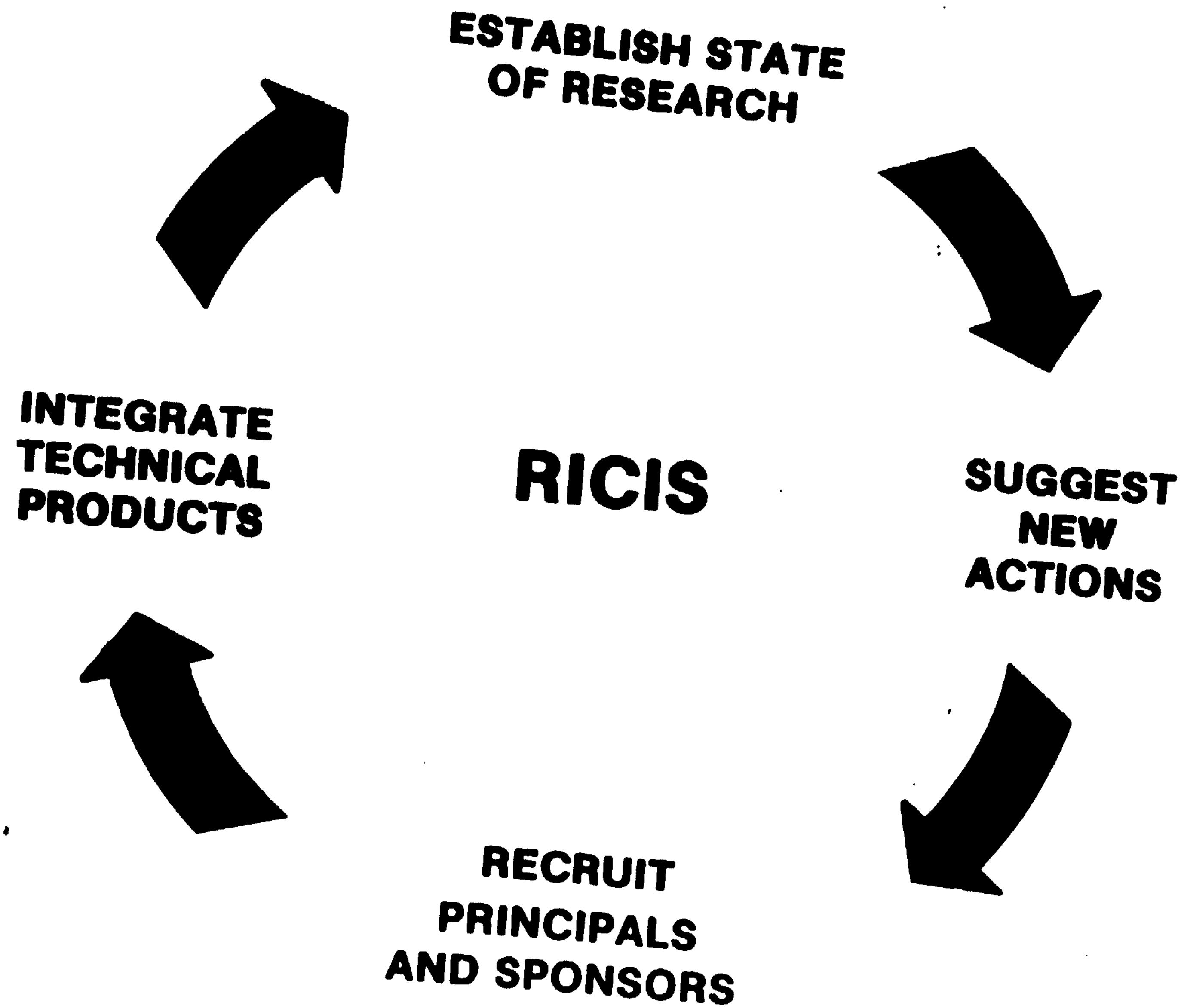


- INFORMATION MANAGEMENT

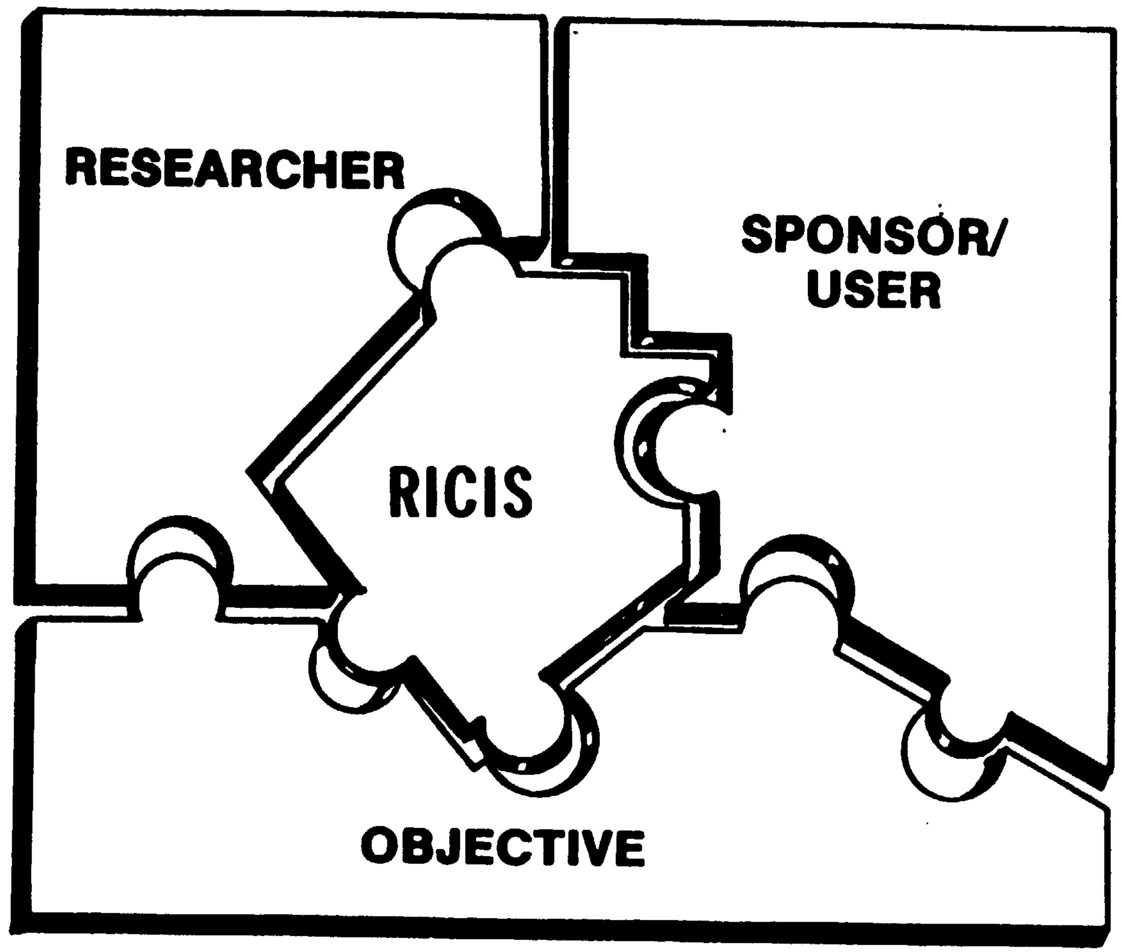


- EDUCATION AND TRAINING

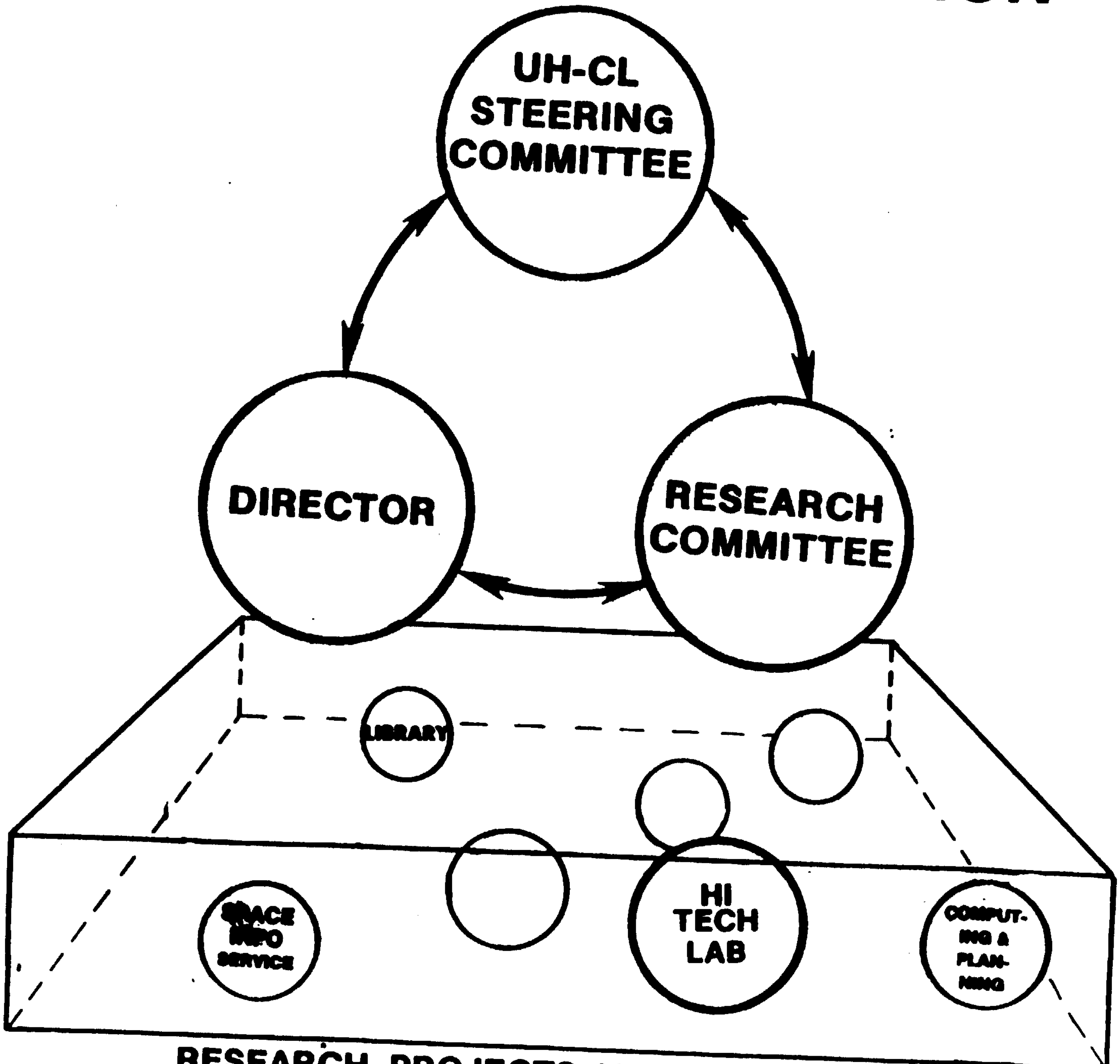
INTEGRATION STRATEGY



RESEARCH PROJECT "3-WAY MATCH"



UNIVERSITY ORGANIZATION



RESEARCH PROJECTS AND SUPPORT OFFICES

UH-CL STEERING COMMITTEE

RESPONSIBILITIES

- ESTABLISH POLICY-LEVEL DIRECTION
- PROVIDE OVERALL MANAGEMENT AUTHORITY
- OVERSEE FISCAL AFFAIRS
- DIRECT STRATEGIC PLANNING
- SELECT PRINCIPAL INVESTIGATOR FROM ITS MEMBERSHIP

DIRECTOR
RESPONSIBILITIES

- SUPPORT STEERING COMMITTEE
- COORDINATE TACTICAL PLANNING WITH JSC
- PROVIDE STAFF SUPPORT TO RESEARCH COMMITTEE
- ASSIST IN ESTABLISHING RESEARCH PROJECTS
- MONITOR PROGRESS OF RESEARCH PROJECTS
- MANAGE INSTITUTE OFFICE

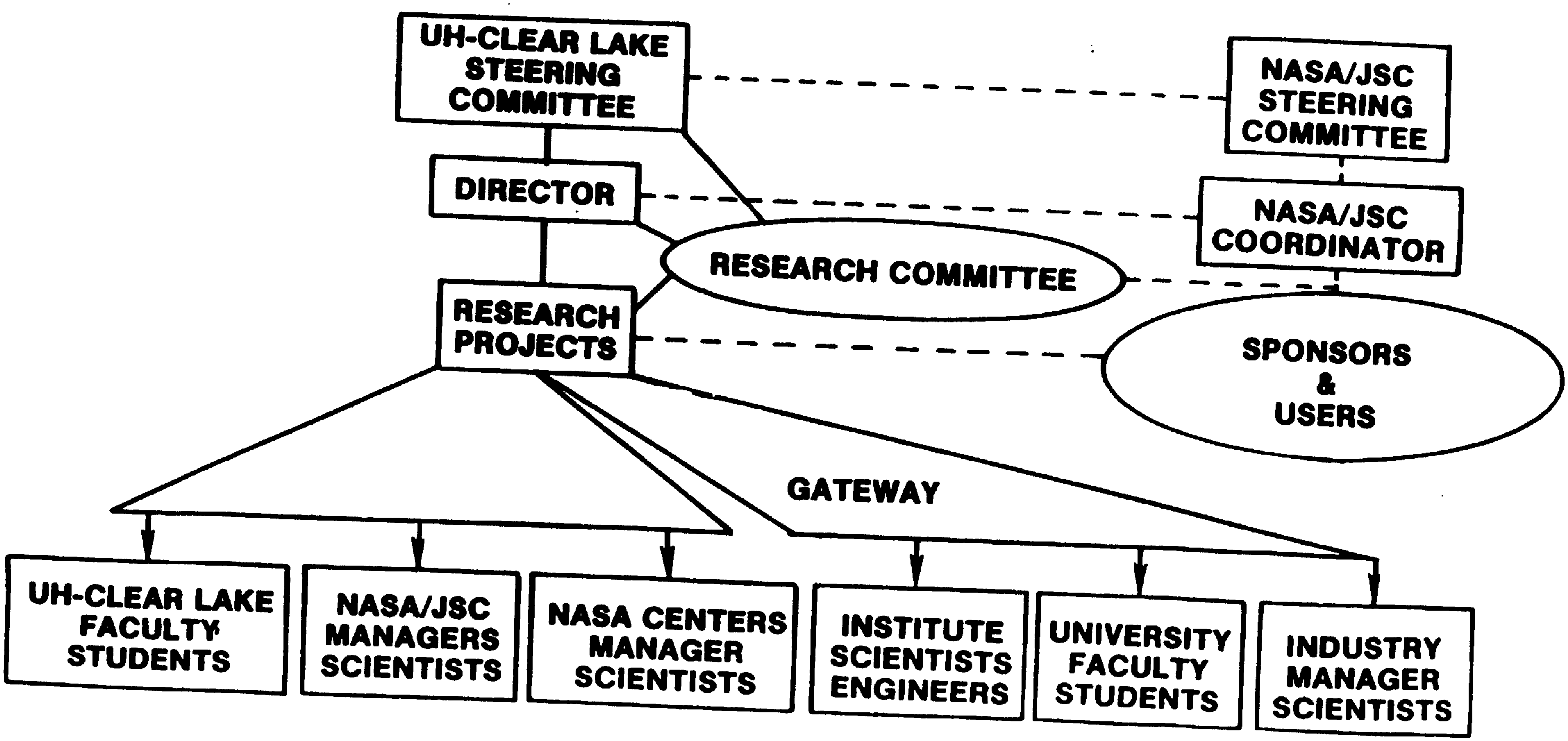
RESEARCH COMMITTEE

RESPONSIBILITIES

- PROVIDE TECHNICAL DIRECTION
- ESTABLISH STATE OF RESEARCH
- PROMOTE RESEARCH ACTIVITIES
- IDENTIFY RESEARCH ORGANIZATIONS/PRINCIPALS
- INTEGRATE RESEARCH PROPOSALS AND REPORTS
- RECOMMEND NEW RESEARCH INITIATIVES

RESEARCH INSTITUTE for COMPUTING and INFORMATION SYSTEMS

MANAGEMENT STRUCTURE



RESEARCH STATUS

RESEARCH ACTIVITY METHODOLOGY

- RESEARCH ACTIVITY MAY BE INITIATED BY RESEARCHER OR JSC SPONSOR
- RESEARCH ACTIVITY DESCRIPTION (RAD) INCLUDES:
  - RESEARCH OBJECTIVE
  - BACKGROUND
  - APPROACH
  - SCHEDULE
  - DELIVERABLES
  - BUDGET
- RAD REQUIRES TECHNICAL APPROVAL (JSC AND UHCL)
- GATEWAY RESEARCH REQUIRES PROPOSAL (RAD) FROM RESEARCH ORGANIZATION

RESEARCH ACTIVITY METHODOLOGY
(CONT)

- EACH RESEARCH ACTIVITY IS INITIATED (OR LATER MODIFIED) VIA A PROGRAM CHANGE REQUEST (PCR)
- PCR REQUIRES ADMINISTRATIVE APPROVAL (JSC & UH-CLEAR LAKE)

Table 1. Approved Research Activities as of September 30, 1987

RESEARCH INSTITUTE FOR COMPUTING & INFORMATION SYSTEMS
COOPERATIVE AGREEMENT NCC 9 - 16

Current Date: 11-Oct-87

	RESEARCH ACTIVITY	PROJECT DIRECTOR	IMPLEMENTING ORGANIZATION	JSC TECHNICAL MONITOR NAME	ORG.	PERIOD FROM	TO
AI.1	COMMUNICATION & TRACKING EXPERT SYSTEMS STUDY	LEIBFRIED	UH-CLEAR LAKE	SCHMIDT	EE7	6/1/87	12/31/8[illegible]
AI.2	COMPUTER GRAPHICS TESTBED TO SIMULATE & TEST VISION SYSTEMS FOR SPACE APPLICATIONS	FEAGIN & CHEATHAM	RICE UNIVERSITY	CLEGHORN	FM7	6/1/87	10/15/87
AI.3	ROBOTIC PATH PLANNING & SOFTWARE TEST-BED ARCHITECTURE	FEAGIN & VOLZ	UNIV. OF MICHIGAN	CLEGHORN	FM7	6/1/87	12/31/87
AI.4	APPLICATION OF FUZZY SET AND RELATED THEORY TO FAILURE DETECTION AND CONTROL IN SPACE SYSTEMS	FEAGAN & SHERIDAN	M.I.T.	CLEGHORN	FM7	6/1/87	12/31/87
AI.5	DEMONSTRATION OF A 3D VISION ALGORITHM FOR SPACE APPLICATIONS	FEAGIN & deFIGUEIREDO	RICE UNIVERSITY	CLEGHORN	FM7	6/1/87	9/30/87
AI.6	SIMULATION OF ROBOTIC SPACE OPERATIONS	GIARRATANO & JANI	LINCOM	CLEGHORN	FM7	6/1/87	9/15/87
AI.7	T IN THE CRAY X/MP	FEAGIN & HUDAK	YALE UNIVERSITY	SAVELY	FM72	6/1/87	5/31/88
AI.8	R&D FOR ONBOARD NAVIGATION (ONAV) GROUND BASED EXPERT/TRAINER SYSTEM	FEAGIN & JANI	LINCOM CORP.	SAVELY	FM72	10/1/87	5/31/88
AI.9	OBJECT ORIENTED PROGRAMMING & FRAME REPRESENTATION USING ADA	FEAGIN, AUTY & CHARNIAK	SOFTECH & BROWN UNIV.	SHULER	FR4	10/1/87	5/31/88
ET.1	SOFTWARE ENG. & ADA TNG	FREEDMAN	UH-CLEAR LAKE	GORMAN	FR43	1/16/87	5/31/87
ET.2	SOFTWARE ENGINEERING WITH ADA: A LIFE-CYCLE CURRICULUM	FREEDMAN	UH-CLEAR LAKE	GORMAN	FR43	6/1/87	12/31/87
ET.3	COMPUTER BASED ADA TRAINING SYSTEMS (CBATS)	FREEDMAN	UH-CLEAR LAKE	GORMAN	FR43	6/1/87	12/15/87
ET.4	SOFTWARE ENGINEERING AND ADA TRANSITION COURSE DEVELOPMENT	FREEDMAN & McBRIDE	UH-CLEAR LAKE & SOFTECH	KENNEDY	AH311	7/1/87	4/15/88
IM.1	SPACE MARKET MODEL	BISHOP	UH-CLEAR LAKE	DEMEL	KE	6/1/86	5/31/88
IM.2	CLEAR LAKE AREA COMPUTER CAPABILITY SURVEY	HODGIN	UH-CL BUREAU OF BUS. RESEARCH	MacDONALD	FA	2/1/87	6/30/87
IM.3	SPACE SHUTTLE PAYLOAD INFORMATION SYSTEM	BISHOP	UH-CLEAR LAKE	DENING	EX2	1/1/87	12/31/87
IM.4	ELECTRONIC DOCUMENTATION	DEDE	UH-CLEAR LAKE	GORMAN	FR43	6/1/87	5/31/88
IM.5	LONG-RANGE PLAN FOR THE COMMERCIAL DEVELOPMENT OF THE SPACE STATION	BISHOP & EVAN	CTR FOR SPACE AND ADV. TECH.	SVEGLIATO HENDERSON	KE EX4	6/1/87	9/31/88
IM.6	METHODOLOGIES FOR INTEGRATED INFORMATION MANAGEMENT SYSTEMS	BISHOP & MAYER	TEXAS A&M	SAVELY	FM72	6/1/87	12/31/87
IM.7	DEVELOPING INTEGRATED PARAMETRIC PLANNING MODELS FOR BUDGETING AND MANAGING COMPLEX DEV. PROJ.	ETNYRE & BLACK	UH-CLEAR LAKE	WHITTINGTON	KX2	7/1/87	1/15/88
IM.8	CLEAR LAKE AREA COMPUTER CAPABILITY CENSUS	HODGIN	UH-CLEAR LAKE	MacDONALD	FA	8/1/87	3/1/88
IM.9	MANAGEMENT INFORMATION AND DECISION SUPPORT ENVIRONMENT	BISHOP	UH-CLEAR CLEAR	ERICKSON	FM26	8/1/87	5/31/88

ORIGINAL PAGE IS
OF POOR QUALITY

Table 1. Approved Research Activities as of September 30, 1987

RESEARCH INSTITUTE FOR COMPUTING & INFORMATION SYSTEMS
COOPERATIVE AGREEMENT NCC 9 - 16

Current Date: 11-Oct-87

	RESEARCH ACTIVITY	PROJECT DIRECTOR	IMPLEMENTING ORGANIZATION	JSC TECHNICAL MONITOR NAME	ORG.	PERIOD FROM	TO
IM.10	RESEARCH IN IMAGE MANAGEMENT AND ACCESS	BISHOP & RORVIG	UNIV. OF TEXAS	PENROD	JA	10/1/87	5/31/88
MS.1	SPACE STATION MOMENTUM MANAGEMENT AND ATTITUDE CONTROL	FEAGIN & WIE	UNIV. OF TEXAS	BORDANO	FM4	6/1/87	1/31/88
PO.1	RICIS PROJECT OFFICE	HOUSTON	UH-CLEAR LAKE	MacDONALD	FA	6/1/86	5/31/88
SE.1	ADA PROGRAMMING SUPPORT ENVIRONMENT DATA BASE	LEKKOS & LIAW	UH-CLEAR LAKE	LOVEALL	FR121	6/1/86	1/15/88
SE.2	DMS TEST BED USER'S MANUAL DEVELOPMENT	McKAY & AUTY	SOFTECH, INC.	RAINES	EH421	6/1/86	10/31/86
SE.3	ADA-BASED S-O-A EXPERT SYSTEM BUILDER	McKAY & WILLIAMS	INFERENCE, INC.	SAVELY	FM72	9/1/86	6/30/87
SE.4	APSE BETA TEST SITE TEAM SUPPORT	McKAY & LEGRAND	SOFTECH, INC.	GORMAN	FR43	6/1/87	10/30/87
SE.5	BENCHMARKING OF ADA ON EMBEDDED COMPUTER SYS.	McKAY & AUTY	SOFTECH, INC.	HUMPHREY	EH431	7/1/86	5/31/87
SE.6	DEVELOPMENT OF A PROOF-OF-CONCEPT PROTOTYPE	McKAY & RANDALL	UH CLEAR LAKE & GHG INC.	GORMAN	FR43	6/1/86	12/31/87
SE.7	JOINT NASA/JSC UH-CL SERC	McKAY	UH-CLEAR LAKE	GORMAN	FR43	6/1/86	9/30/87
SE.8	WORK STATION EVALUATION	PERKINS & GRAVES	UH-CLEAR LAKE & BARRIOS	SCHWARTZ	EA121	6/1/86	10/31/87
SE.9	TESTING AND VERIFICATION OF ADA FLIGHT SOFTWARE FOR EMBEDDED COMPUTERS	McKAY & AUTY	SOFTECH, INC.	HUMPHREY	EH431	6/1/87	1/31/88
SE.10	ATOP : SERC - A STUDY OF CONVERTING PCTE SYSTEM SPECIFICATION TO ADA	McKAY, AUTY & ROGERS	SOFTECH, INC. & ROCKWELL	GORMAN	FR43	6/1/87	7/31/87
SE.11	SYMBOLIC INFORMATION IN ADA TESTING AND INTEGRATION	McKAY & AUTY	SOFTECH, INC.	GORMAN	FR43	6/1/87	2/15/88
SE.12	ATOP : SERC-SECURITY FOR SPACE SYSTEMS	McKAY & LeGRAND	SOFTECH, INC.	GORMAN	FR43	6/1/87	9/14/87
SE.13	ATOP : AUTOMATIC SOFTWARE VERIFICATION TOOLS	McKAY, COHEN & AUTY	SOFTECH, INC.	GORMAN	FR43	6/1/87	1/1/88
SE.14	IRDS PROTOTYPING W/ APPLICAT'N TO REPRESENTATION OF EA/RA MODELS	LEKKOS	UH-CLEAR LAKE	GORMAN	FR43	6/1/87	8/31/87
SE.15	FAULT TOLERANT ADA SOFTWARE	BOWN/DAVARI/ROGERS/McKAY	UH-CLEAR LAKE	GORMAN	FR43	6/1/87	12/31/87
SE.16	IMPLEMENT THE DISTRIBUTED ADA MODEL	McKAY, KAMRAD & RANDALL	HONEYWELL & GHG CORPORATION	GORMAN	FR43	6/1/87	5/31/88
SE.17	ADA ANALYSIS FOR NASA SPACE STATION PROGRAM OFFICE	McKAY & McBRIDE	SOFTECH, INC.	HALL	HQTS	6/1/87	5/31/88
SE.18	ESTABLISHMENT OF ADA TECHNOLOGY TRANSFER NETWORK: AdaNET	McKAY, BUTCHER & DIGMAN	MountainNET	BIVENS	HQTS	10/1/87	5/31/88

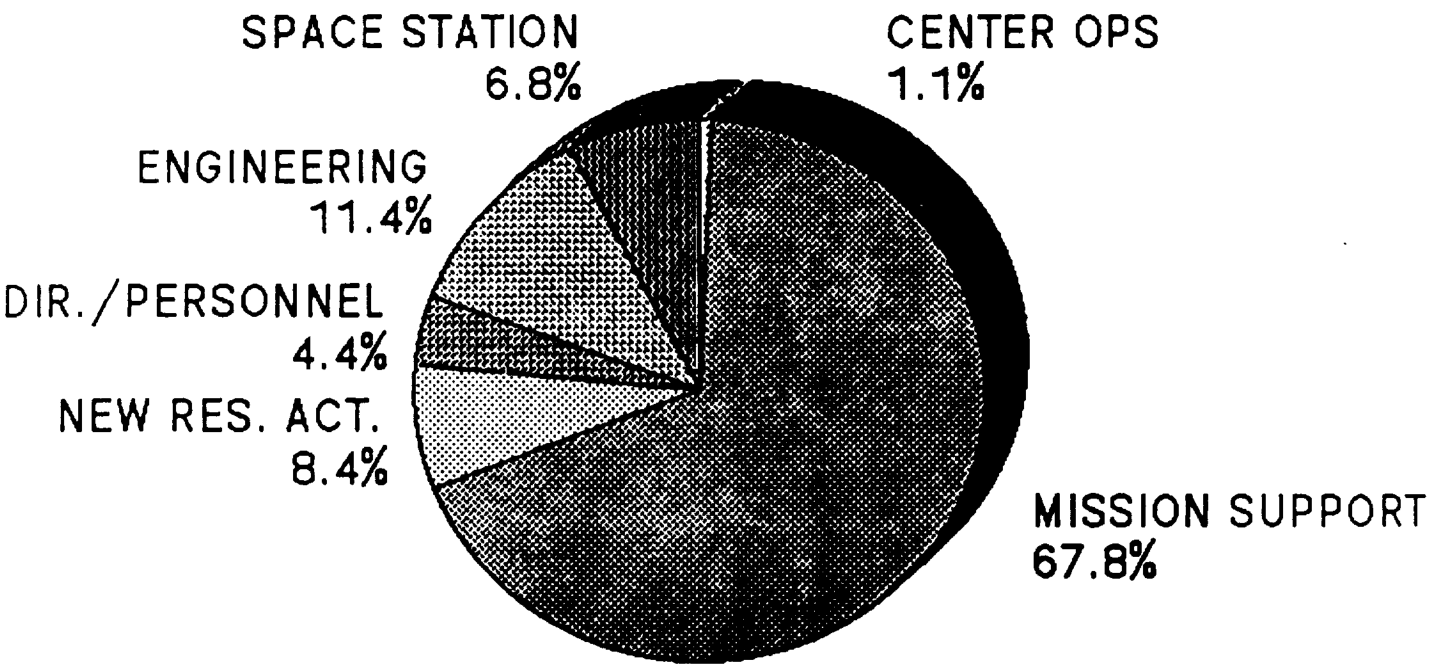
UH-CLEAR LAKE / NASA-JSC COOPERATIVE AGREEMENT NCC 9-16

THREE YEAR AGREEMENT BEGINNING JUNE 1, 1986

- ➡ EACH 12 MONTH SEGMENT FUNDED SEPARATELY
- ➡ $5.1M ALLOCATED FOR FIRST TWO YEARS
- ➡ EXPENDED $1.8M IN YEAR ONE
- ➡ LEAVES $3.3M FOR RESEARCH IN YEAR TWO

RICIS JSC SPONSORS

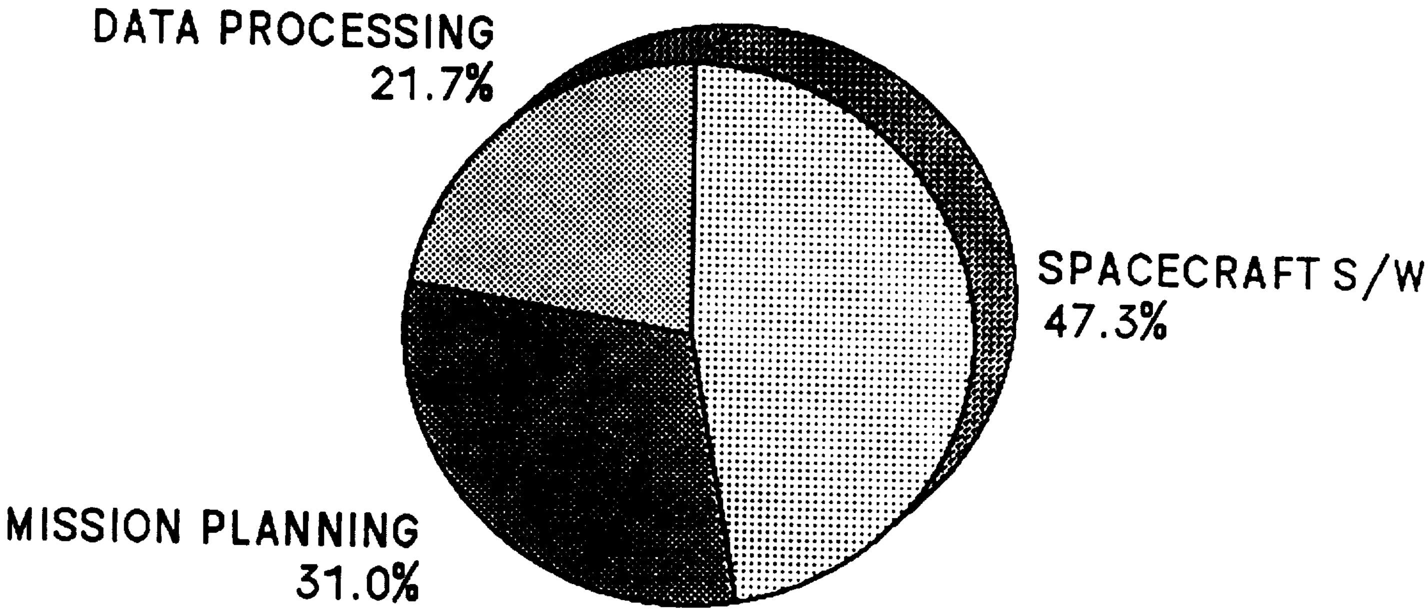
JUNE 1, 1986 - SEPTEMBER 30, 1987



FUNDS ALLOCATED TOTAL $5,059,942

RICIS JSC SPONSORS

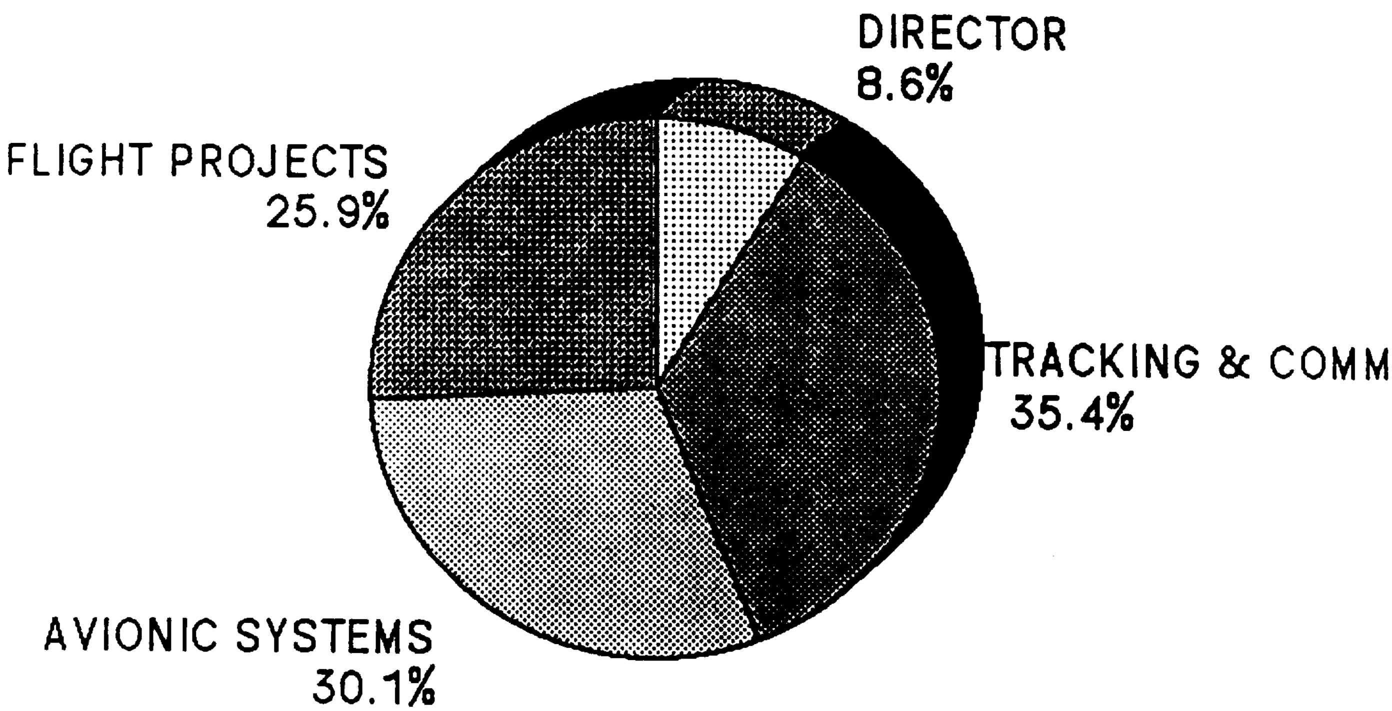
MISSION SUPPORT DIRECTORATE
(Includes funds from the Navy and the Air Force)



FUNDS ALLOCATED TOTAL $3,432,834

RICIS JSC SPONSORS

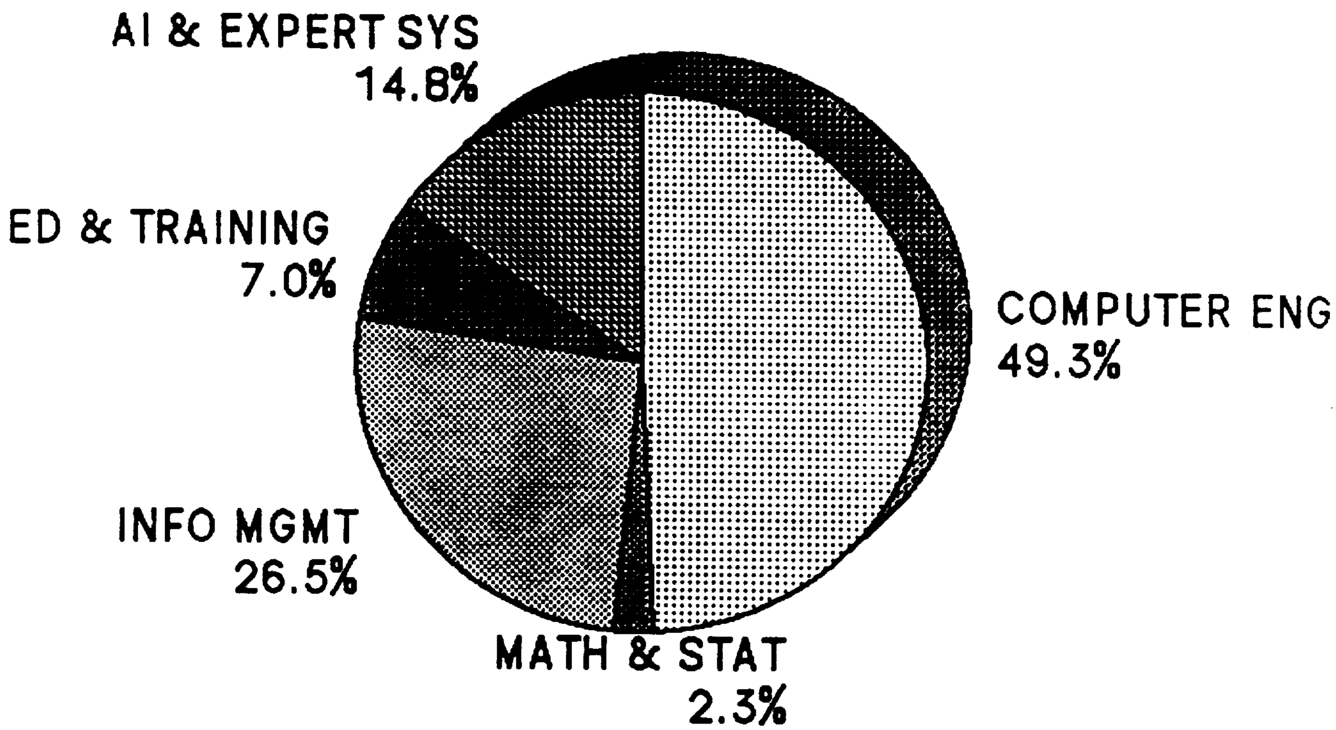
ENGINEERING DIRECTORATE



FUNDS ALLOCATED TOTAL $578,733

RICIS RESEARCH

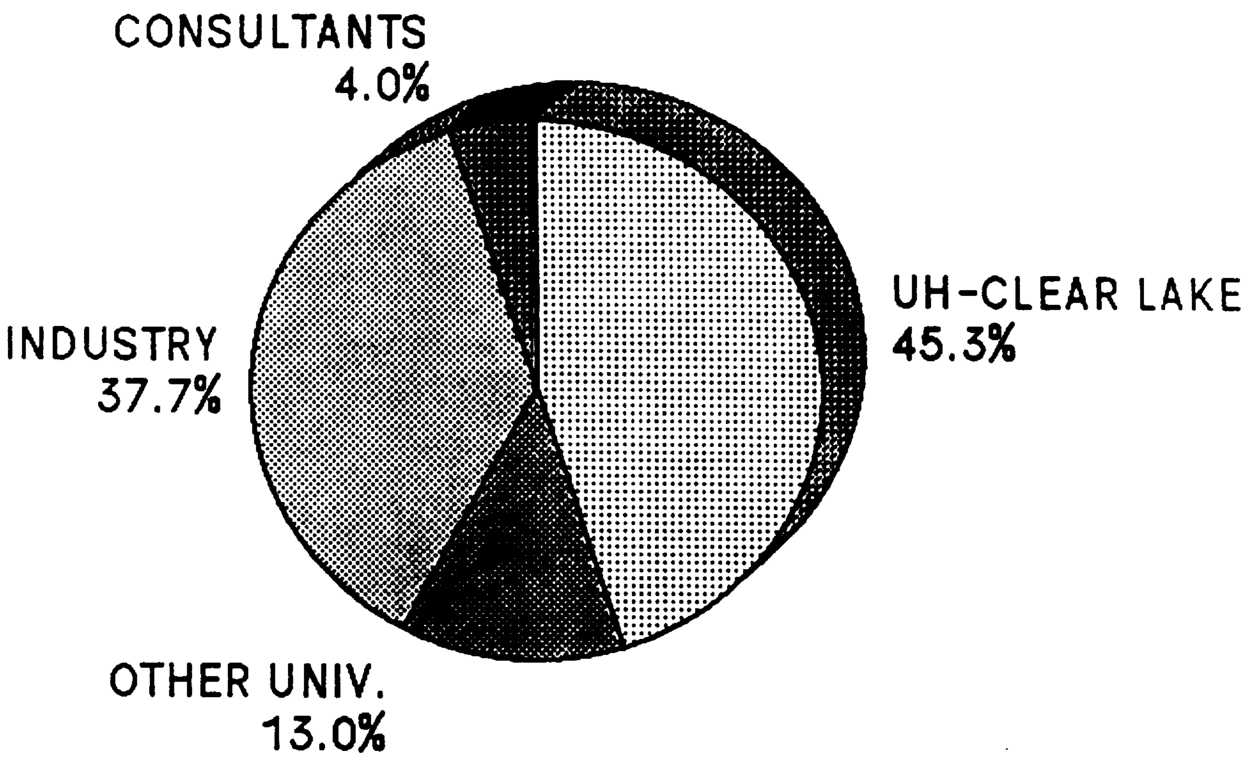
JUNE 1, 1986 - SEPTEMBER 30, 1987



APPROVED EXPENDITURES TOTAL $4,462,742

RICIS RESEARCHERS

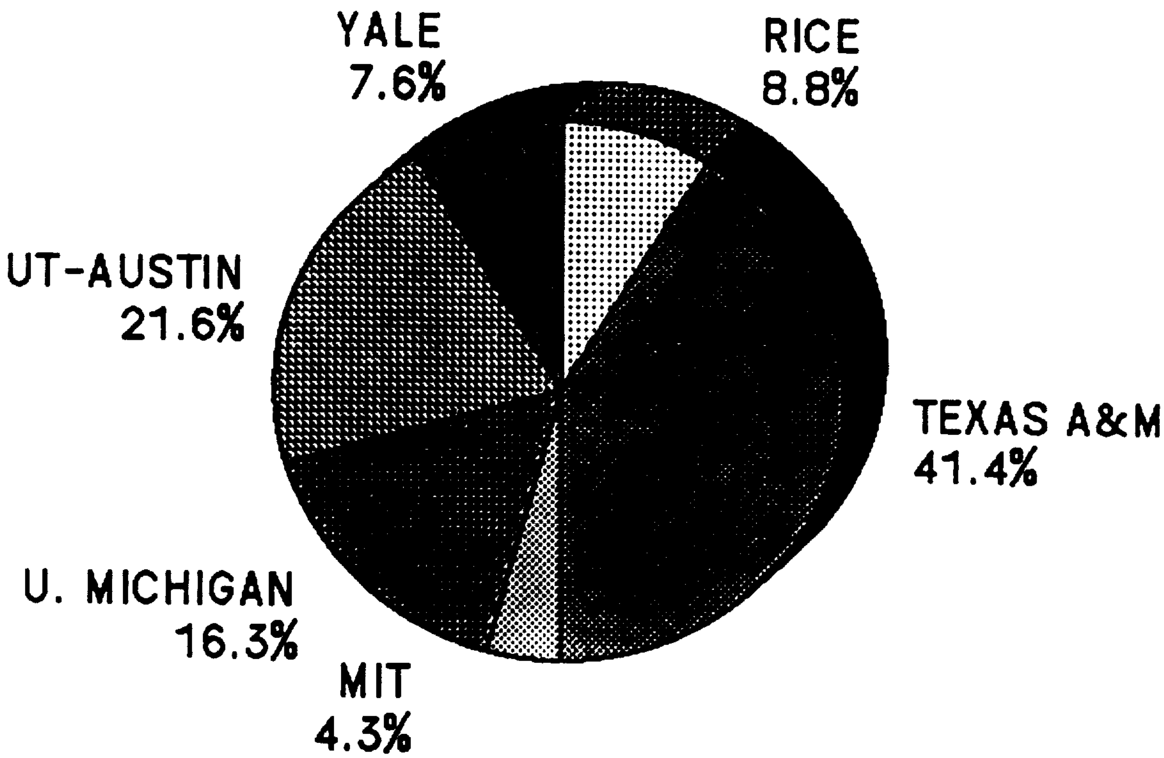
JUNE 1, 1986 - SEPTEMBER 30, 1987



APPROVED EXPENDITURES TOTAL $4,462,742

RICIS RESEARCHERS

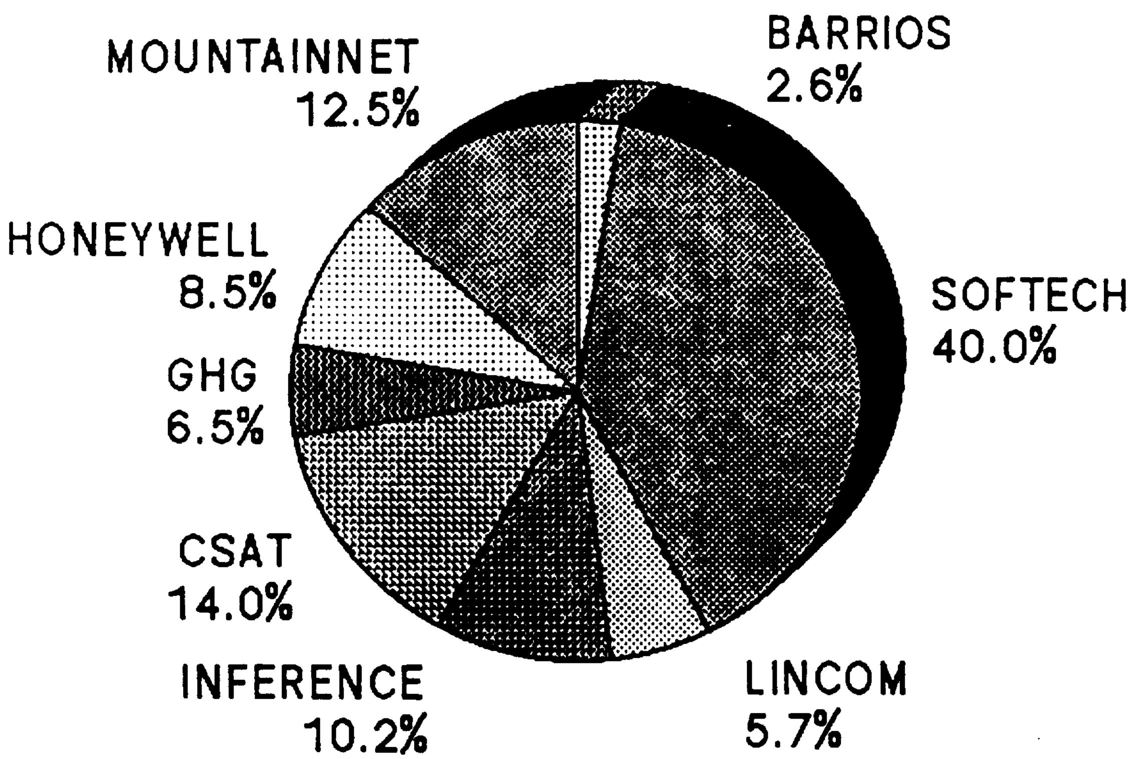
OTHER UNIVERSITIES



OTHER UNIV. EXPENDITURES TOTAL $ 581,808

RICIS RESEARCHERS

INDUSTRY



INDUSTRY EXPENDITURES TOTAL $1,680,901

RICIS COMPUTING RESOURCES

RCDF*
- -SEQUENT
- -HARRIS HCX-9

- -IBM 4381

UNIVERSITY COMPUTING
- -VAX 11/750
- -VAX 11/780 Academic
- -VAX 11/785

- -VAX 8250
- -VAX 8700 Administrative

LAN
- -IBM Token Ring
- -4 8228 MAU's
- -11 IBM PC/AT currently connected

JSC ACCESS

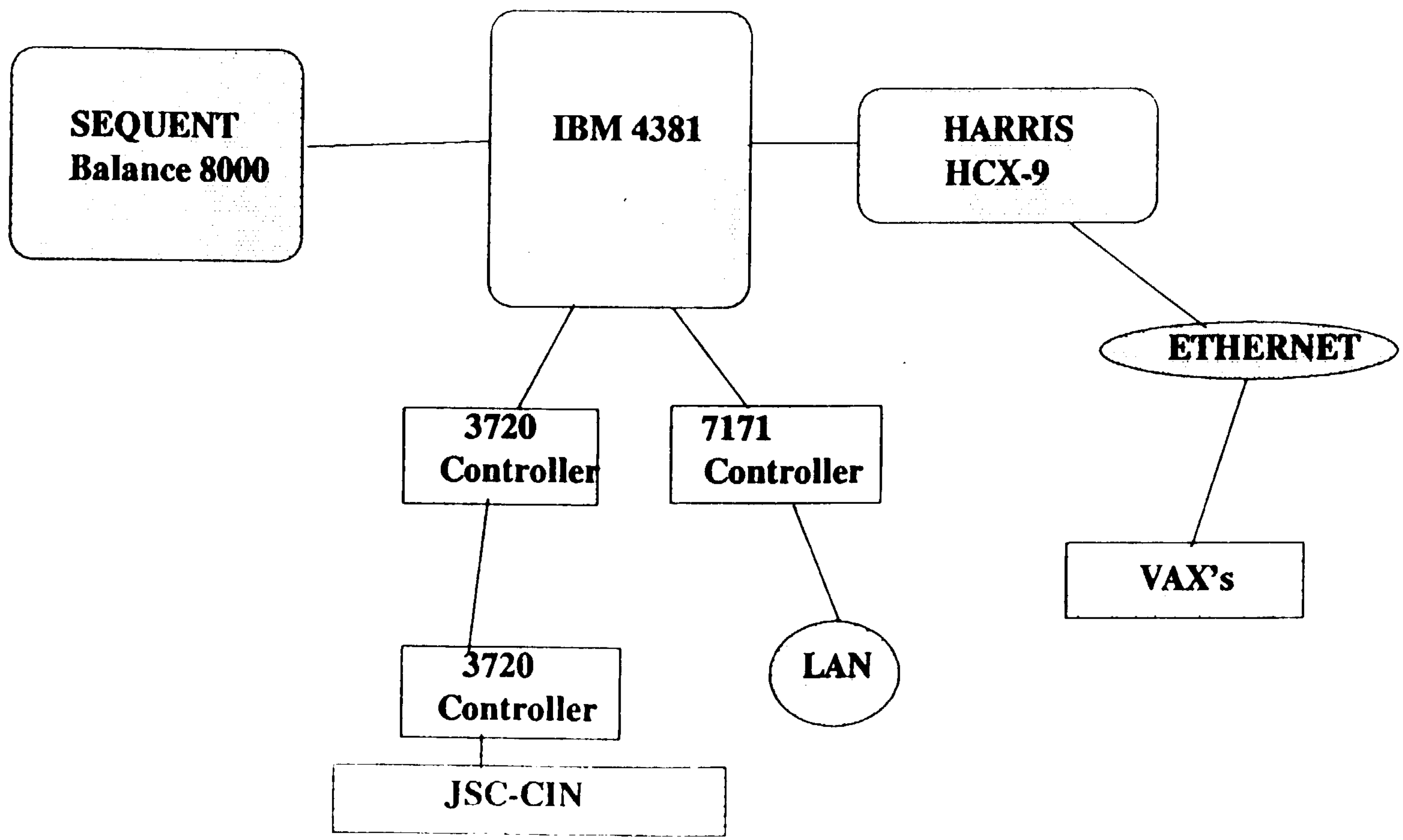
To:
- CIS-B (Profs)
- CIS-C (Oracle/SEAD)
- CIS-D (NOMAD/Shuttle Payload) (Text DBMS/ ?)

via the 3274 controller and dial-in

* Research Computing and Data Facility

JSC/UH-CL NETWORKING

Plans



N91-18618

RICIS RESEARCH

Computer Systems and Software Engineering

Charles W. McKay, Director, High Technology Laboratory and Software Engineering Research Center, Professor of Computer Science, UH-Clear Lake

Artificial Intelligence and Expert Systems

Terry Feagin, Professor of Computer Science, UH-Clear Lake

Information Systems

Peter C. Bishop, Director, Space Business Information Center, Associate Professor of Human Sciences, UH-Clear Lake

Mathematical and Statistical Analysis

Cecil R. Hallum, Associate Professor of Mathematics, UH-Clear Lake

Education and Training

Glenn B. Freedman, Director, Center for Cognition and Instruction, Associate Professor of Reading and Language Arts, UH-Clear Lake

An Overview of the Computer Systems and Software Engineering Component of RICIS

Charles W. McKay

The principal focus of this RICIS component is computer systems and software engineering in-the-large of the lifecycle of large, complex, distributed systems which:

* evolve incrementally over a long life,

* contain non-stop components, and

* must simultaneously satisfy a prioritized balance of mission and safety critical requirements for behavior at run time

This focus is believed to be extremely important at this time because of the contribution of the "scaling direction problem" to the current software crisis. That is, paradigms/models, techniques/methodologies and tools which often worked for yesterday's comparatively smaller, simpler, centralized systems have been shown to be an inadequate baseline to scale-up to meet the challenges of distributed systems. By contrast, models, methodologies, tools, and environments which are based on a sounder theoretical foundation to address these larger and more complex systems are capable of scaling-down to meet the needs of less demanding, centralized applications.

As shown in Figure 1, the Computer Systems and Software Engineering (CSSE) component addresses the lifecycle issues of three environments-host, integration and target. Solutions are proposed, specified, designed, developed, verified and sustained in the host environment. The solutions are deployed, monitored, interactively queried and operated int he target environment. Increasingly, components of both the host environment and the target environment are geographically as well as locally distributed. The solutions from the host environment are moved into the target environment under the control of the integration environment. The integration environment is responsible for monitoring and sustaining the current baseline of software, hardware and operational components in the target environment. The integration environment is also responsible for the test plans and for controlling the integration and evolution of advancing the target environment to the next baseline. Emergency interactions are also controlled through the integration environment.

As Shown in Figure 2, an integrated lifecycle support environment is becoming the common interface to four principal engineering activities: computer systems engineering, software engineering, hardware engineering, and the management of operations and logistics.

Currently there are more than 18 funded research activities in this technical area. Additionally, there is a larger number of CSSE coordinated research projects which are funded by companies working with the university. Several of these activities are deliberately structured interfaces to the other four components of RICIS.

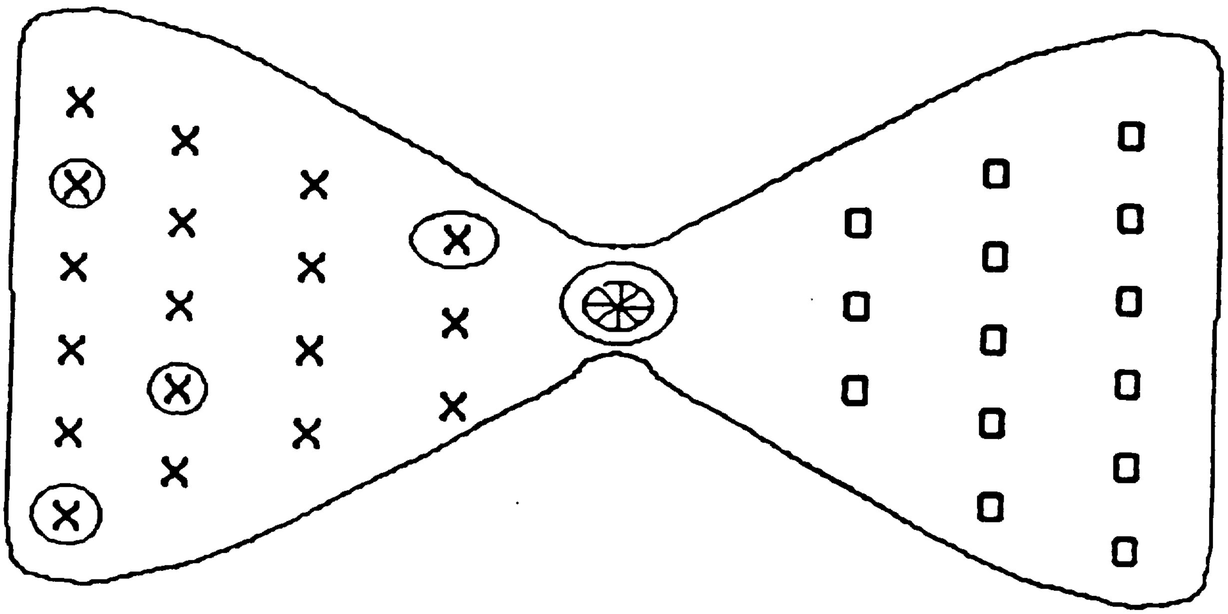
The goals for CSSE research during the next three years may be summarized as:

1. To develop a position of international leadership in the engineering of mission and safety critical components for the target and integration environments of large, complex non-stop, distributed systems.

2. To sustain a position of international leadership in the research issues of the host environment for the above applications.

3. To augment the Computer Systems and Software Engineering research base and provide support as needed to the other four technical areas addressed in RICIS.

TWO SCENARIOS FOR
SSP ENVIRONMENT
IN 2000+ A.D.



HOST ENVIRONMENTS:

- DEVELOP
- SUSTAIN

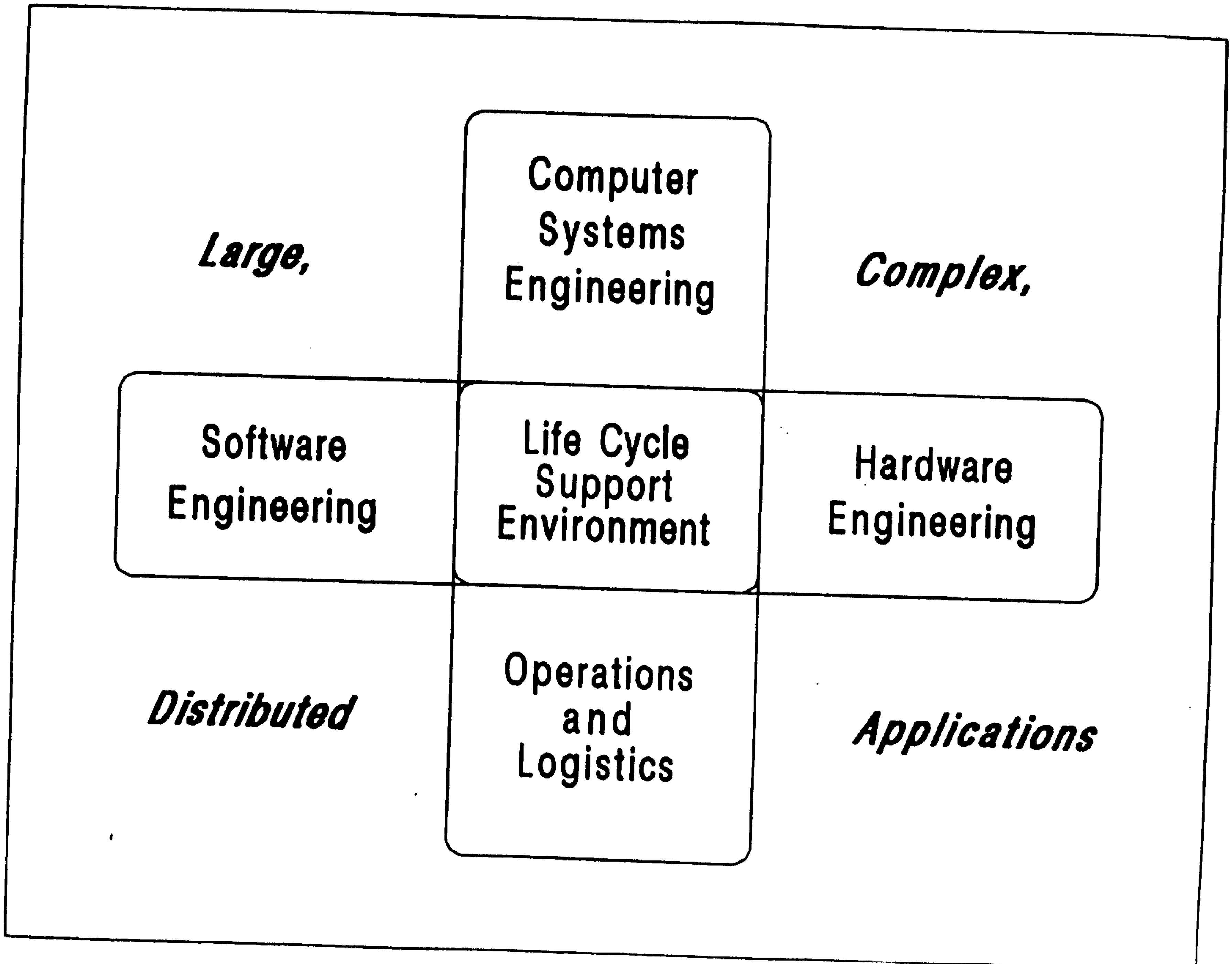
INTEGRATION ENVIRONMENT:

- CONTROL OF TGT. ENVIR. BASELINE
- INTEGRATION V&V FOR NEXT BASELINE AND TEST & INTEGRATION PLANS

TARGET ENVIRONMENTS:

- DEPLOY
- OPERATE

Figure 1



Large,
Computer Systems Engineering
Complex,
Software Engineering
Life Cycle Support Environment
Hardware Engineering
Distributed
Operations and Logistics
Applications

N91-18619

An Overview of the Artificial Intelligence and Expert Systems Component of RICIS

Dr. Terry Feagin

Artificial Intelligence (AI) is the study of how to simulate the intelligent behavior and problem-solving skills of humans using computational models. Expert Systems (ES) are AI application programs for accomplishing a task which requires expertise from within a particular domain, i.e. deciding where is the best place to drill for oil, determining how to configure a large computer system or finding the cause of a power outage. The areas of research in AI include knowledge representation, search, planning, learning and knowledge acquisition, computer vision, natural language understanding and speech, automatic inference and theorem proving, reasoning with uncertainty, logic programming, expert system and robotics. In solving these problems, the AI research scientist employs a number of specialized models, approaches and representations such as predicate calculus, semantic nets, scripts and frames, augmented transition networks, heuristic search, decision theory, constraint propagation, fuzzy logic, Bayesian inference networks, measures of belief/disbelief, default reasoning and production systems with forward and/or backward chaining.

For space applications, a number of problem areas that should be able to make good use of the above tools include resource allocation and management, control and monitoring, environmental control and life support, power distribution, communications scheduling, orbit and attitude maintenance, redundancy management, intelligent man-machine interfaces and fault detection, isolation and recovery. Research activities in this technical are researching solutions to these problems using the techniques of artificial intelligence and expert systems.

N91-18620

An Overview of the Information Management Component of RICIS

Peter C. Bishop

Information management is the RICIS research area devoted to the final customer of computing and information systems--the end-user. They are the people at the end of the long chain of information systems who don't care how their information is collected, manipulated or stored as long as the right information is in their hands at the right time.

Information **productivity** is the overall objective of the information management research area. In other words, people who use information systems should realize more value by using the system than by not using it. NASA in general and the Johnson Space Center in particular have a tremendous need to understand what makes an information system productive and to develop productive systems for its employees, contractors and customers. JSC, for instance, has the responsibility to manage the U.S. space operations--a task which requires an incredible amount of information. As a result, JSC should be a national leader in using information in a productive manner. The information management section of RICIS is set up to engage in those research projects which promote that end.

The approach of the research tasks within the information management section varies depending on the nature of the problem. Four types of tasks were initiated during the first year:

Surveys -- a description of the existing state of some area of computing and information systems.

A. Environmental Scanning for Information Processing
(Dr. Peter Bishop, UH-Clear Lake)

A survey of existing products for IBM mainframes in three applications areas--database management, full-text retrieval and optical character recognition

B. Clear Lake Area Computer Capability Census
(Dr. Robert Hodgin, UH-Clear Lake)

A survey of JSC and contractor computer capability in the Clear Lake area.

Forecasts -- a description of the alternative future states of some area

A. Space Station Advanced Virtual Electronic Documentation
(Dr. Chris Dede, UH-Clear Lake)

An analysis of information technology which could be available for use in the documentation of space station software when it is developed

Plans -- an approach to accomplishing some objective in the future

A. Long-Range Plan for Commercializing Space Station
(L.J. Evans, CSAT)

An analysis of the drivers and obstacles to commercial use of the space station along with strategies for overcoming the obstacles in order to maximize such use

B. Methodologies for Integrated Information Management Systems
(Dr. Richard Mayer, Texas A&M

A theoretical model for designing requirements for information systems

Demonstrations -- working prototypes and field trials to study the feasibility and the benefits of a particular information system

A. Space Shuttle Payload Information System
(Dr. Peter Bishop, UH-Clear Lake)

A study of the information available concerning the Space Shuttle

B. Space Market Model Development Project
(Dr. Peter, Bishop, UH-Clear Lake)

A study of the information available for space commercialization in general, including a design for providing the business community information which it does currently posses

C. Research in Image Management and Access
(Dr. Mark Rorvig, UT-Austin)

A study to increase the searchability of the keywords associated with the photographic and video archives at JSC

D. Management Information and Decision Support Environment
(Dr. Peter Bishop, UH-Clear Lake)

The prototype design for a computer interface whereby JSC managers can get information from JSC databases

An Overview of the Mathematical and Statistical Analysis Component of RICIS

N91-18621

Cecil R. Hallum

An aspect of computing that especially warrants input from the mathematical and statistical community is that which pertains to assessing the quality of a piece of software, or the trustworthiness of computer hardware and computer networks. Although much of the work in this area is probabilistic in nature, most of the work to-date has been done by engineers and published in the engineering literature. Numerous problems remain, however, whereby mathematicians and statisticians should get involved in order to provide supporting research, particularly in regard to design of hardware, the configuration of networks and policies for the development of reliable software.

Due to advances in microelectronics, problems regarding reliability are gradually shifting from hardware to software. Moreover, due to the overall expense of software (60 to 80 percent) relative to the whole system, and due to the fact that numerous failures are software connected, more emphasis is now being given to concerns for ensuring the reliable operation of the software system. Once again the mathematical and statistical community is a source for considerable insight into this problem area since they are well-qualified to address:

1. Quantification and measurement of software reliability.
2. Assessment of changes in software reliability over time (reliability growth).
3. Analysis of software-failure data.
4. Decision logic for whether to continue or stop testing software.

A fault-tolerant computer is one whose key features are the automatic detection, diagnosis and correction of errors (faults). A review of the existing literature shows that a satisfactory solution addressing this problem is not yet available. In particular, the research gap is evidenced by the fact that existing material is mostly qualitative; there appears to be potential for valuable contributions in this technical area.

In addition to the areas mentioned above, other areas that research in this technical area can be of particular aid in regard to problems of interest to NASA/JSC include the following:

1. Math modeling of physical systems.
2. Simulation.
3. Statistical data reduction.
4. Evaluation methods including robustness (stability), sensitivity analysis, perturbation theory, error analysis and development of test criteria.
5. Optimization.
6. Algorithm development.
7. Mathematical methods in signal processing.

N91-18622

An Overview of the Education and Training Component of RICIS

Glenn B Freedman

Research in education and training focuses on means to disseminate knowledge, skills and technological advances rapidly, accurately and effectively. A range of areas for study have been identified including artificial intelligence applications, hypermedia and full-text retrieval strategies, use of mass storage and retrieval options such as CD-ROM and laser disks, and interactive video and interactive media presentations.

The Education and Training area also provides necessary support activities for dissemination of research information from the other research areas. Further, this area serves as a link among the University, corporations, and government for information on training, curriculum development and education services.

Among the first-year accomplishments of education and training and practitioners researchers were the following:

1. market survey in software engineering and Ada training
2. establishment of the Software Engineering Professional Education Center
3. establishment of UH Clear Lake Software Engineering and Ada Training Forum
4. delivery of a hypertext training system for Ada
5. delivery of a software engineering training film for upper-level managers
6. creation of the course "Introduction to Software Engineering for Managers"
7. development of the course "Software Engineering and the Transition to Ada"
8. application for affiliation with the Software Engineering Institute

N91-18623

INVITED TALK

A NASA Initiative: Software Engineering for Reliable Complex Systems

Lee B. Holcomb, Director Human Sciences
and Human Factors, OAST, NASA Headquarters

Invited Talk

A NASA Iniative: Software Engineering for Reliable Complex Systems

Lee B. Holcomb

The objective of this initiative is the development of methods, technology and skills that will enable NASA to cost-effectively specify, build and manage reliable software which can evolve and be maintained over an extended period. The need for such software is rooted in the increasing integration of software and computing components into NASA systems.

As a result, the size, capability and complexity of NASA systems are increasing rapidly. This growing complexity causes a number of significant software issues. The prevention of software failure becomes critical.

Improvements in software productivity must catch up with and keep pace with software complexity. Functional descoping that has been caused by software complexity must be eliminated. And, the enormous maintenance costs generated by complex software must be reduced.

Current NASA Software Engineering expertise has been applied toward some of the largest reliable systems including shuttle launch and ground support, shuttle simulation and minor control and satellite tracking and scientific data systems. Research in NASA focuses on Software Engineering in Management and Environments, Fault Tolerant Software/Reliability Models and Performance Measurement. Several other governmental agencies, DoD, SPC, SEI, DARPA, NSF, AF/RADC and MCC are conducting related research but none specifically focusses on reliable software or management of complexity. In fact, no theory exists for reliable complex software systems.

NASA is seeking to fill this theoretical gap through a number of approaches. One such approach is to conduct research on theoretical foundations for managing complex software systems. The focus of this research includes communications models, new and modified paradigms and life-cycle models. Another approach is research into theoretical foundations for reliable software development and validation. Research here focuses upon formal specifications, programming languages, software engineering systems, software reuse, formal verification and software safety. Further approaches to address the need for reliable complex software involves benchmarking a NASA software environment, experimentation within the NASA context, evolution of present NASA methodology, and transfer of technology to the space station software support environment.

INVITED TALK:

" A NASA INITIATIVE: SOFTWARE ENGINEERING FOR RELIABLE COMPLEX SYSTEMS"

By

Lee Holcomb, Director, Human Sciences and Human Factors, OAST, NASA Headquarters

A NASA INITIATIVE :

SOFTWARE ENGINEERING

FOR RELIABLE COMPLEX SYSTEMS

OCTOBER 1987

AGENDA

OBJECTIVE

RATIONALE

NASA SOFTWARE ENGINEERING EXPERTISE

RELATED RESEARCH

APPROACH

TECHNOLOGY TRANSFER

A NASA INITIATIVE : SOFTWARE ENGINEERING FOR RELIABLE COMPLEX SYSTEMS

OBJECTIVE:

TO DEVELOP METHODS, TECHNOLOGY, AND SKILLS TO ENABLE NASA TO COST-EFFECTIVELY SPECIFY, BUILD, AND MANAGE RELIABLE COMPLEX SOFTWARE WHICH IS EVOLVABLE AND MAINTAINABLE OVER AN EXTENDED PERIOD OF TIME.

COMPLEX SOFTWARE

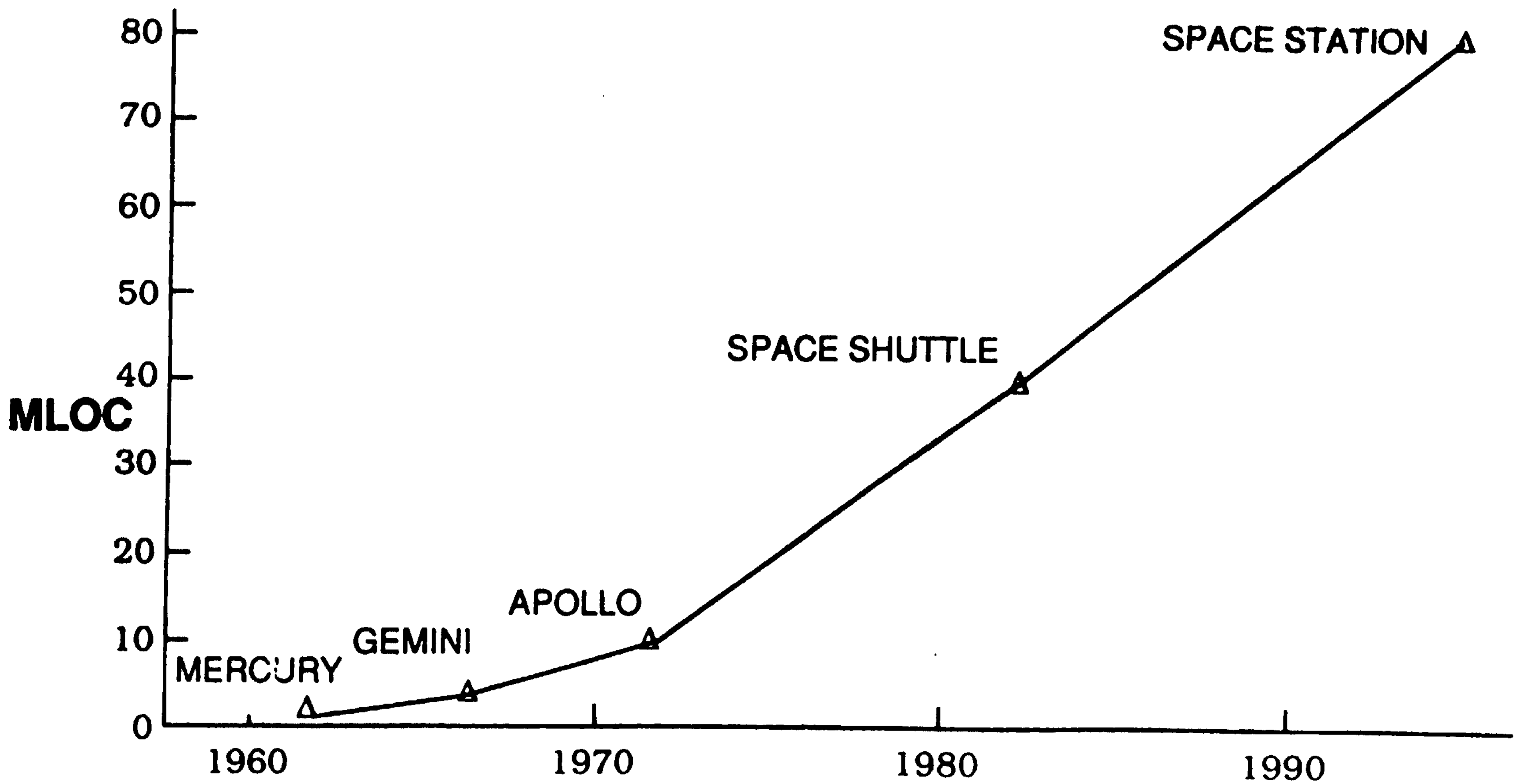
Software that is

- Large (LOC, many states, complex connectivity)
- Distributed (multiprocessor, LAN, WAN, multiple development teams)
- Multifunction (many capabilities, data complexity, KB systems)

Complex software systems generally have long operational life involving corrective maintenance, adjustment to changing computing environments, and enhancement for evolving requirements and newer technology.

GROWTH IN SOFTWARE SIZE

U.S. MANNED SPACE FLIGHT PROGRAM



RELIABLE SOFTWARE

Software that is

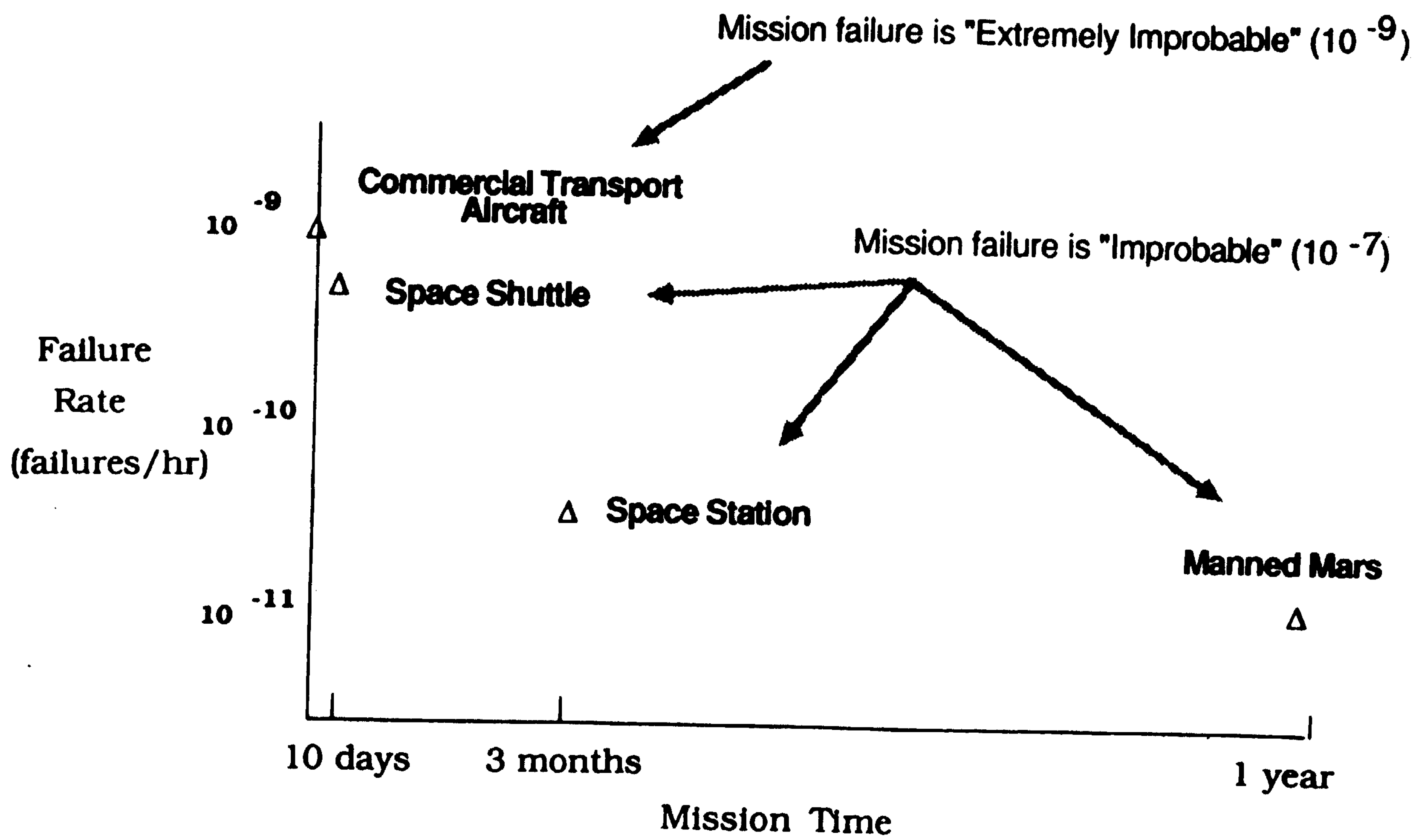
<u>Correct</u> (performs intended functions)

<u>Robust</u> (operates in spite of invalid inputs)

<u>Safe</u> (prohibits life-threatening operation)

Software reliability is the probability that software will not cause the failure of a system for a specified time under specified conditions.

GOALS FOR NASA RELIABLE SOFTWARE



RATIONALE

Software and computing components are becoming an increasingly integral part of NASA systems

The size, capability, and complexity of NASA systems are increasing rapidly with time

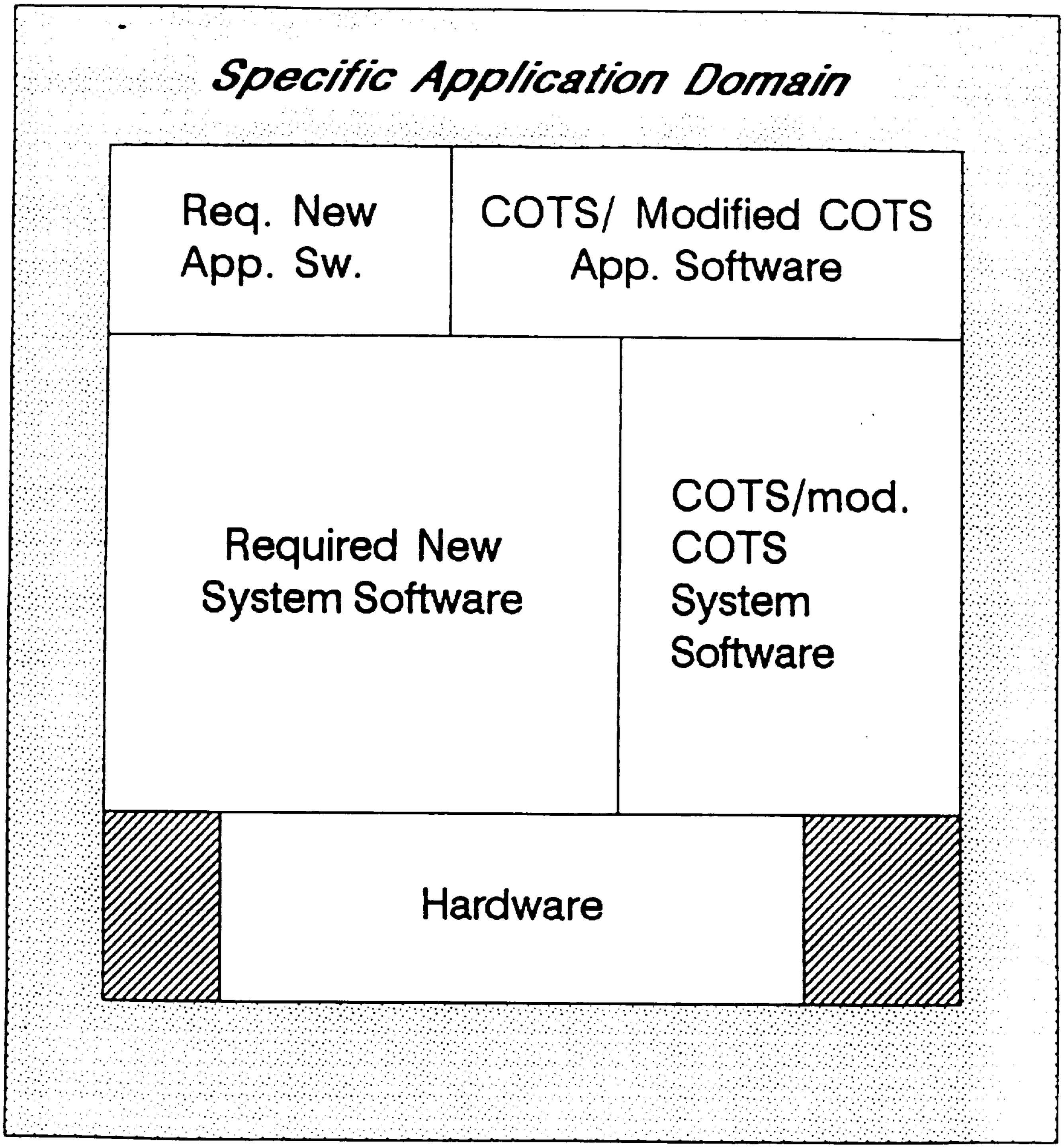
Software failures are intolerable

Software complexity outpaces improvements in software productivity

Software complexity has caused functional descoping

Complex software has enormous maintenance costs

Some "Perspectives" of Software Development



CSSE View of Requirements

Hard Work

CASE HISTORIES

CONTINUED

COMPLEXITY HAS CAUSED FUNCTIONAL DESCOPING

VOYAGER URANUS encounter endangered due to delayed software deliveries and descoping of software system

AFTI/F-16 first flight delayed more than 1 year, and then only normal modes used, not 6 advanced modes originally planned

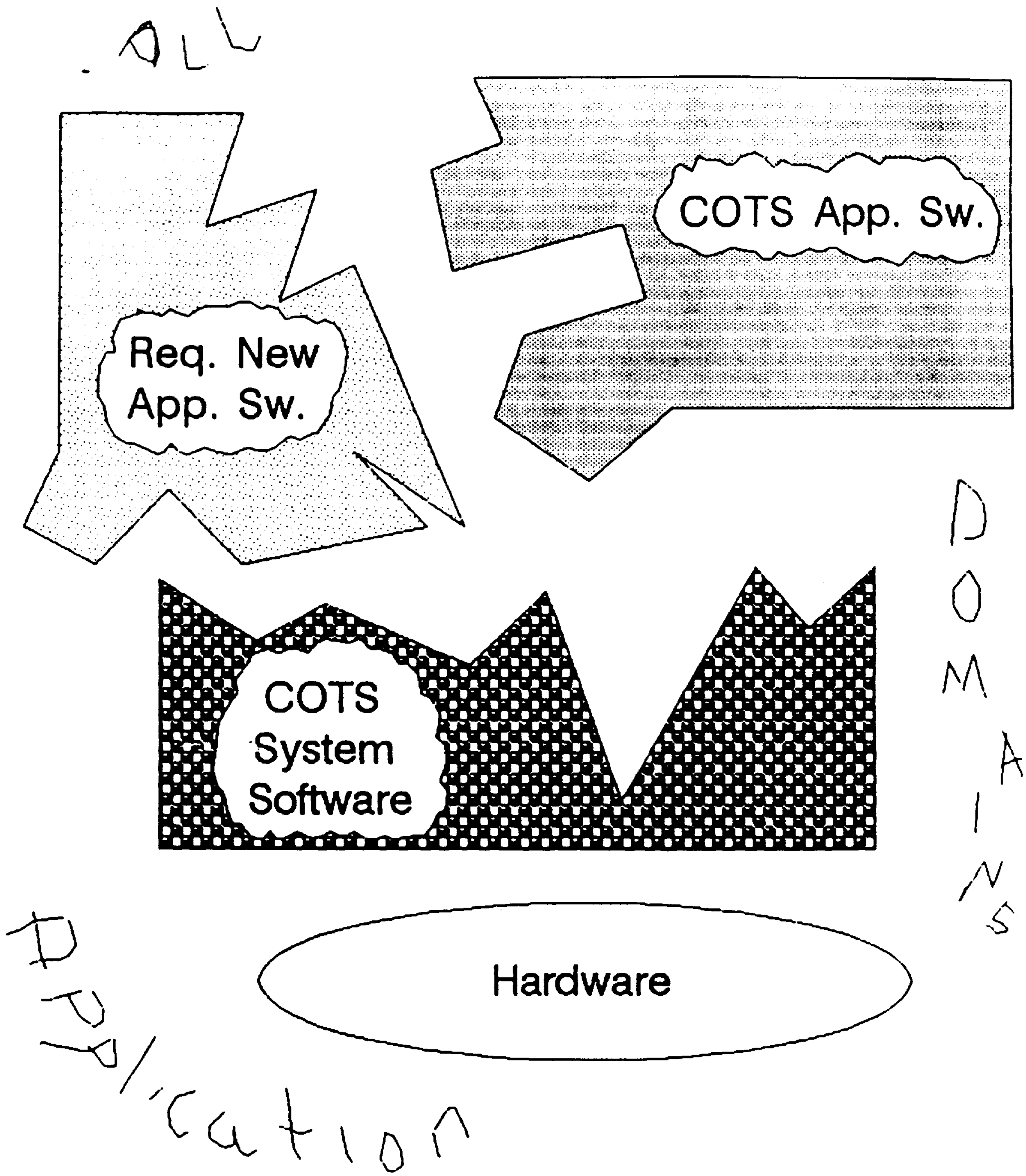
Full flight control system in X-WING cancelled due to software costs exceeding available budget

COMPLEX SOFTWARE INCURS ENORMOUS MAINTENANCE COSTS

Shuttle Ground Processing Support - 12M LOC, 420 people

Shuttle Software Production and Maintenance - 2.5M LOC, 378 people

Some "Perspectives" of Software Development



COTS Panacea View of Requirements Work/Play

RELATED RESEARCH

DoD -- Ada Standard language, run-time environments

SPC -- Reusability, prototyping to reduce costs

SEI --- Technology transition: Ada, environments, tools

DARPA -- Common LISP, formal specification language, operating systems, persistent object bases

NSF -- Research in software engineering

AF/RADC -- Knowledge-based software assistants

MCC/Software Technolgy Program

NONE are specifically focusing research on reliable software or management of complexity

A Proposed
Clear Lake Model for Computer Systems and Software Safety
in a
Portable Common Execution Environment (PCEE)

A baseline from which subsequent progress in the target environments and integration environments may be made

An extensible model which can also scale down to improve safety in smaller, simpler applications

A "lessons implied / learned" stimulus and opportunity to develop host environment methodologies and tools which better address the lifecycle issues of safety

APPROACH

RESEARCH THEORETICAL FOUNDATIONS FOR MANAGING COMPLEX SOFTWARE SYSTEMS

RESEARCH THEORETICAL FOUNDATIONS FOR RELIABLE SOFTWARE DEVELOPMENT AND VALIDATION

BENCHMARK NASA SOFTWARE ENVIRONMENT

EXPERIMENT IN THE NASA CONTEXT

EVOLVE PRESENT NASA METHODOLOGY

TRANSFER TECHNOLOGY TO SPACE STATION SOFTWARE SUPPORT ENVIRONMENT

RICIS Umbrella

Computer Systems and Software Engineering (CSSE)

plus

SERC (Software Engineering Research Center)

Research
Education and Training (SEPEC)

plus

Interfaces to:

Artificial Intelligence and Expert Systems
Information Management
Mathematical and Statistical Analyses

RESEARCH ON THEORETICAL FOUNDATIONS FOR RELIABLE SOFTWARE DEVELOPMENT AND VALIDATION

FORMAL SPECIFICATION

PROGRAMMING LANGUAGES

SOFTWARE PROTOTYPING

FAULT TOLERANCE

COMPUTER-AIDED SOFTWARE ENGINEERING SYSTEMS

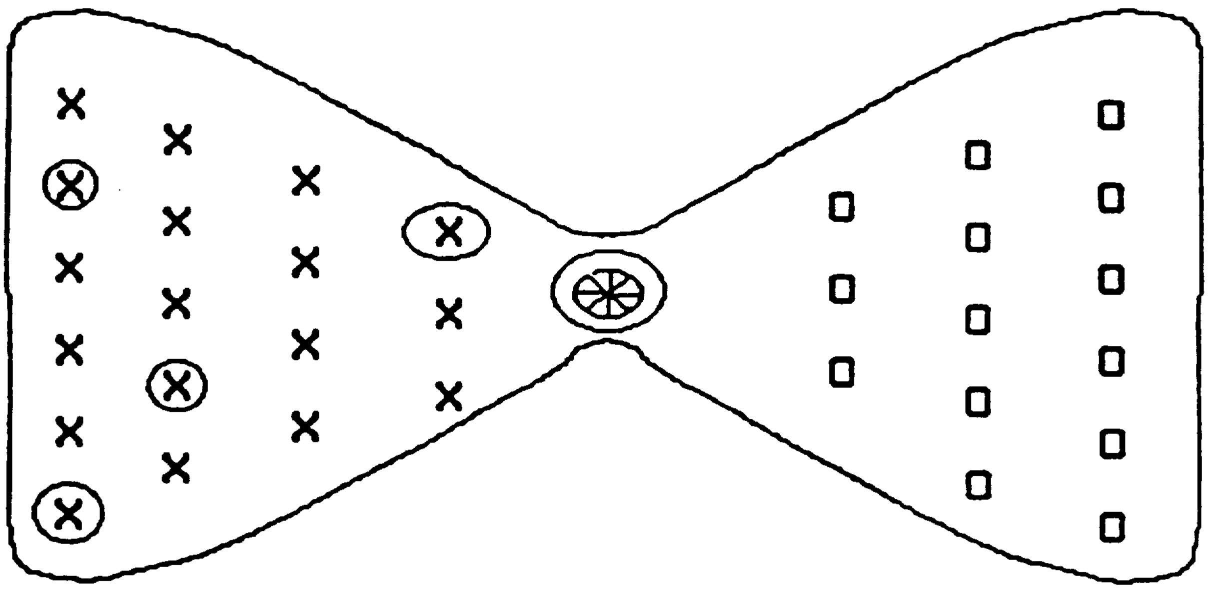
SOFTWARE REUSE

FORMAL VERIFICATION

SOFTWARE SAFETY

OTHER IDENTIFIED APPROACHES

TWO SCENARIOS FOR SSP ENVIRONMENT IN 2000+ A.D.



HOST ENVIRONMENTS:

- DEVELOP
- SUSTAIN

INTEGRATION ENVIRONMENT:

- CONTROL OF TGT. ENVIR. BASELINE
- INTEGRATION V&V FOR NEXT BASELINE AND TEST & INTEGRATION PLANS

TARGET ENVIRONMENTS:

- DEPLOY
- OPERATE

EVOLVE PRESENT NASA METHODOLOGY

CAPTURE MOST PROMISING EXPERTISE IN KNOWLEDGE BASE FOR EXPERT SYSTEM

TRANSFORM INTO PROCEDURE OR TOOL/ENVIRONMENT

Computer Systems and Software Engineering

Charles W. McKay

Director

High Technologies Laboratory

Software Engineering Research Center

UH – Clear Lake

TECHNOLOGY TRANSFER TO

SPACE STATION SOFTWARE SUPPORT ENVIRONMENT

A COMMON ENVIRONMENT FOR ALL SPACE STATION SOFTWARE DEVELOPMENT AND MAINTENANCE

CONTRACT AWARDED TO LOCKHEED MISSILES AND SPACE COMPANY WITH INTERIM SYSTEM FOR 2 YEARS, INCREMENTAL BUILDS OVER 6 YR

AN IDEAL TESTBED FOR NASA SOFTWARE RESEARCH RESULTS AND FOR TRANSITION OF NEW SOFTWARE TECHNOLOGY INTO PRACTICE

MEMORANDUM OF UNDERSTANDING BETWEEN OAST AND OSS CURRENTLY UNDER CONSIDERATION

Clear Lake Model

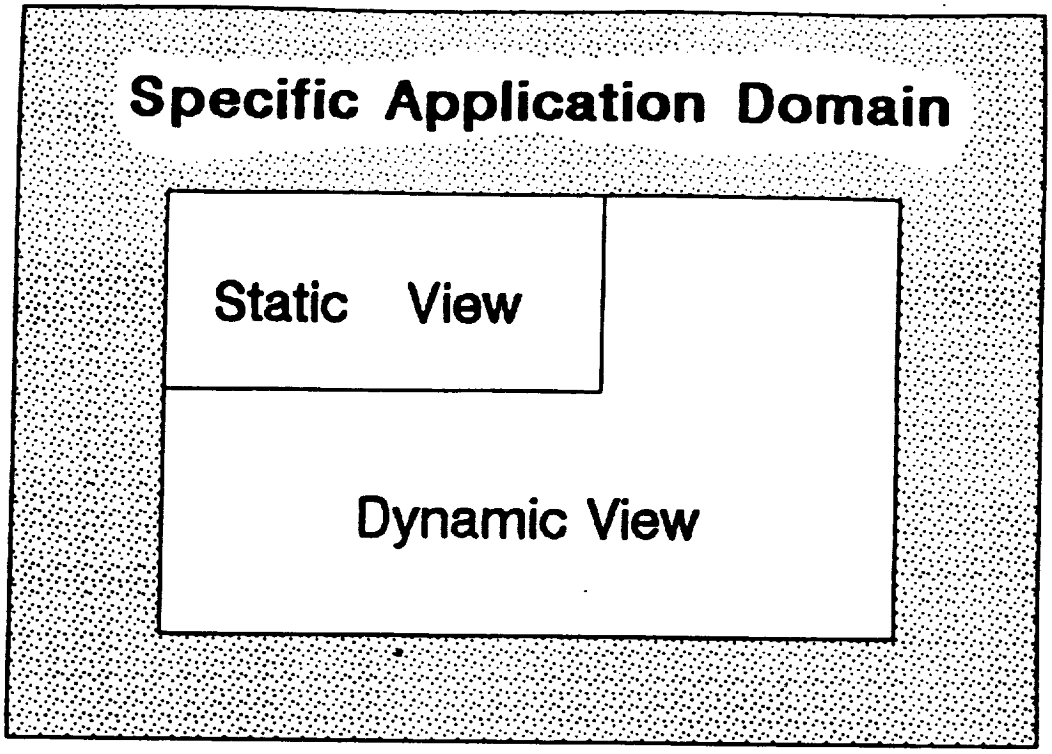
- Distribute Ada entities
- Interface to virtual Ada machine
- Provide functionality via XRTL
- Transparent reconfiguration
- Application directed reconfiguration
- Surrogates and agent tasks

KEYNOTE ADDRESS

The Real Technologies in Space Station Information Systems

John R. (Jack) Garman, Director of Information Systems Services, Space Station Program Office, NASA Headquarters

Some "Perspectives" of Software Requirements



Static "Gains" +

Software Development Team

Funcional and Nonfunctional Requirements

Behavioral Assertions

Software Quality Mgmt. Team

Funcional and Nonfunctional Requirements

Behavioral Assertions

Hardware Target Team

Normal and Exceptional CFg

Dynamic View of Software Requirements

Keynote Address

The Lead Technologies in Space Station Information Systems

John R. Garman

MASC Support Kernel Components

1. A tailorable RTSE developed & sustained in Ada

2. Software structuring which facilitates: firewalling, layered recovery capabilities, dynamic reconfiguration and extensibility

3. Pools of processes and processors capable of non-stop operation in a fault-tolerant environment

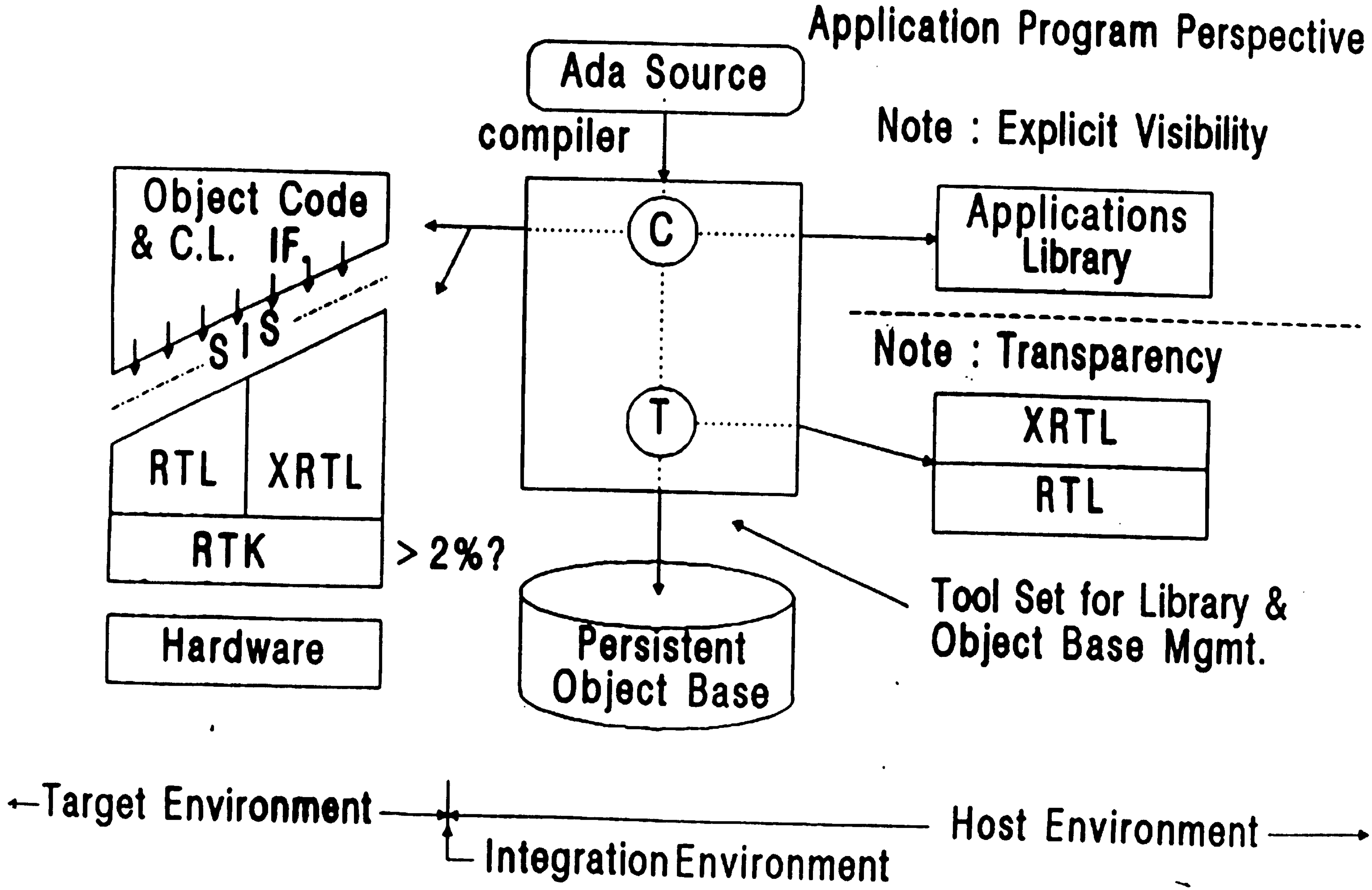
4. A command language interface between the SIS of the integration environment's PCEE and the SIS of the target environment's PCEE

5. System-wide, lifecycle-unique identification of all objects and transactions / subtransactions

6. Dynamic, multilevel security in the integration & target environments

RICIS RESEARCH AREAS

A Model for Supporting a 'Bare Machine' Philosophy for 'Safety Kernels' of Ada Runtime Support Environments (Ada RTSE's)



EDUCATION AND TRAINING

Conveners: **Glenn B. Freedman,** UH-Clear Lake

Amy B. Kennedy, Employee Development, NASA/JSC

Review of the Education and Training Activities

Glenn B. Freedman, UH-Clear Lake
Sue LeGrand, SofTech,Inc.

Software Engineering and the Transition to Ada

John McBride, SofTech, Inc.

Computer Based Ada Training Using Hypertext Systems

Sue LeGrand and **Gilbert Marlowe**, SofTech, Inc.

Mr. David Auty presented "Testing and Verification of Ada Flight software for Embedded computers." This presentation focused on the issues of storage management in Ada. Mr. Auty described certain aspects of the language which, if misused, could lead to errors which are difficult to detect. Examples include global variables, exception propagation, dynamic task interactions, generic subprograms parameters, and dynamic storage allocations. He distinguished among good software engineering approaches to utilizing these features, compiler dependencies which affect these features and the critical role of the run time support environment int he robust and correct management of storage. He then described some of the recommendations and guidelines that have resulted from this study.

The final presentation of this session was made by Dr. Morris Liaw. Dr. Liaw and his colleague, Dr. Anthony Lekkos, recently delivered an operational Software Engineering and Ada Database. He described the objectives and the history of the project as well as the architecture and the features of this unique resource. He further described the methodology used in design and development and concluded with the description of the planned enhancements for the future. A second release will be available in January 1988. The resource is being used by JSC, UH-Clear Lake and JSC's aerospace contractors.

Summary of the Education and Training Technical Session

Dr. Glenn B. Freedman

In the first presentation, Dr. Freedman reviewed two RICIS activities. He first presented the results of a contractor survey completed in Fall, 1986, in which 21 NASA contractors were interviewed to assess the extent to which they had undertaken software engineering and Ada training programs locally, what their perceptions were about these areas and what their plans were for training and educational activities for the next twelve months. The results indicated that at the time of the survey and interviews, the contractor community had very little software engineering training planned, but were beginning Ada syntax training, even though there was little Ada work in progress. The interviewees indicated that the commitment of NASA mid-level managers toward Ada was not firm and that this perception affected training plans. As one personnel person commented, there was "Ada talk from on high, but no Ada action."

Other findings were that the contractors had hardware, compilers and various tools available, they the general consensus at the time was that the tools and methods for Ada were immature. Training was typically defined in terms of language syntax and semantics, with little regard for the Ada culture that supports software engineering principles and goals. Interestingly, the companies perceived that there were sufficient numbers of programmers available, but few software engineers and design experts. Nonetheless, little in the way of design and software engineering training was planned. One of the most consistent findings was that no "transition to Ada" plans were mentioned, even though each company recognized that Ada would become a language they would be using and that the transition would be resource intensive to some extent.

Freedman also reported on the development of a number of training options for the aerospace community. On of the options was a videotape featuring modules about various aspects of software engineering. The tape, geared to upper level management, contains four modules: The Cost of Software, Software in the Space Station Era, Engineering Software and Building a Software Engineering Environment. A second development effort resulted in a one-half of three day presentation covering software engineering and the highlights of the Ada programming language. Other efforts led to an Ada glossary, a PC-based data base of software engineering and Ada training options, text resources, conferences and other educational and training information.

Among the deliverables to NASA there has also been a model for a comprehensive software engineering curriculum that features six planning dimensions and alternative training methods. Also, Freedman discussed the programs of the Clear Lake Software Engineering and Ada Training Forum, a monthly meeting of training experts from universities and industry, and the Software Engineering Professional Education Center, a complementary center to the Software Engineering Research Center that enables the university to offer a full range of services and research to the software engineering community.

Summary of the Computer Systems and Software Engineering Technical Session

Charles W. McKay

Because it was impossible to provide a meaningful presentation on more than 18 NASA funded activities as well as an even greater number of company funded activities that are coordinated by this component of RICIS, five related activities were selected for presentation. Mr. Pat Rogers introduced the five activities and then presented: "Lifecycle Support for Computer Systems and Software Safety in the Target and Integration Environments of the Space Station Program: Approaches to Fault Tolerant Software Systems."

Safety was defined as "the probability that a system, including all hardware and software and human-machine subsystems will provide appropriate protection against the effects of faults, which, if not prevented or handled properly, could result in endangering lives, health, property and environments." The past and present approaches to mission and safety critical components have been addressed through a static perspective of fault avoidance (i.e., considerations in the host environment only). That is, the development team was encouraged to design as well as possible to keep defects out of the system. The quality management team was encouraged to test as well as possible to identify defects that made it through the work of the development team so that the defects could then be removed before deployment. Post-deployment support depended almost entirely upon hardware techniques (e.g., redundant processors, built-in-tests, error coding) to sustain mission and safety critical components at run time.

Mr. Rogers advocated the CSSE team's position that a dynamic perspective of software assessment and control of run time behavior in the target and integration environments is needed to complement the static perspective which has previously been emphasized. Specifically, as shown in Figure 1, additional software processes should be deployed in the target environment to accompany all mission and safety critical components. For applications programs, these additional processes help to monitor the behavior of each of the critical components. These processes are needed to insure the fastest possible identification of faults that have entered any porion of the system state vector, to firewall their propagation, to analyze which of the predetermined recovery mechanisms are most appropriate, and to effect recovery. At the systems software level, such processes apply to all shareable services and resources which mission and safety critical components of application software depend upon.

Mr Rogers then described the CSSE team's proposal for a Portable Common Execution Environment (PCEE). The two principal components of this proposal are an extended run time support environment library and a Mission and Safety Critical (MASC) kernel. Underneath the MASC kernel are 12 distinct but highly interactive models believed essential to maximize the support for mission and safety critical requirements. Figure 2 depicts the extended run time library model and Figure 3 depicts the list of 12 models underneath the system interface set of the MASC kernel.

EDUCATION AND TRAINING SESSION

By

Dr. Glenn Freedman, UH-Clear

John McBride, SofTech

Sue LeGrand, SofTech

Gilbert Marlowe, SofTech

COMPUTER SYSTEMS AND SOFTWARE ENGINEERING

Conveners: **Charles W. McKay,** UH-Clear Lake

Stephen A. Gorman, Head, Application Systems, Spacecraft Software Division, NASA/JSC

PRODUCT DEMONSTRATION FORUM

NOVEMBER 9, 1987

9:00 A.M. - 12:00 NOON

BUILDING 12

PDF ROOM

RESEARCH ACTIVITIES

ET.1

SOFTWARE ENGINEERING AND ADA TRAINING, PHASE 1 (FREEDMAN)

ET.2

SOFTWARE ENGINEERING AND ADA TRAINING PHASE 2 (FREEDMAN)

SE.17

ADA ANALYSIS FOR SPACE STATION OFFICE (SUE LEGRAND, SOFTECH, INC.)

ET.4

SOFTWARE ENGINEERING AND ADA TRANSITION COURSE (JOHN MCBRIDE, SOFTECH, INC.)

ET.3

COMPUTER BASED ADA TRAINING SYSTEM (SUE LEGRAND, SOFTECH, INC.)

SPACE STATION

Final Report on A Study of System Interface Sets (SIS) For the Host, Target, and Integration Envrionments of the Space Station Program (SSP)

(SE.10)

written by
Dr. Charles McKay, SERC
David Auty, SofTech, and
Kathy Rogers, Rockwell International

presented to

RICIS First Year's Research Symposium

presented by:

Kathy Rogers

Rockwell International
Space Station Systems Division

Software will migrate among the three PCEE environments

SPACE STATION

HOST - Software development, documentation, etc. is developed and sustained

TARGET - Executable versions of the software are deployed and operated

INTEGRATION - Configuration control of the current target baseline

Responsible for the test and integration plans used to interactively advance the target baseline

Controls interaction with target environment during emergencies

7. A message interface which supports three forms of communication among clusters: asynchronous send/receive with 'no waits', remote procedure call, Ada rendezvous

8. Hierarchical runtime structure of the threads-of-control

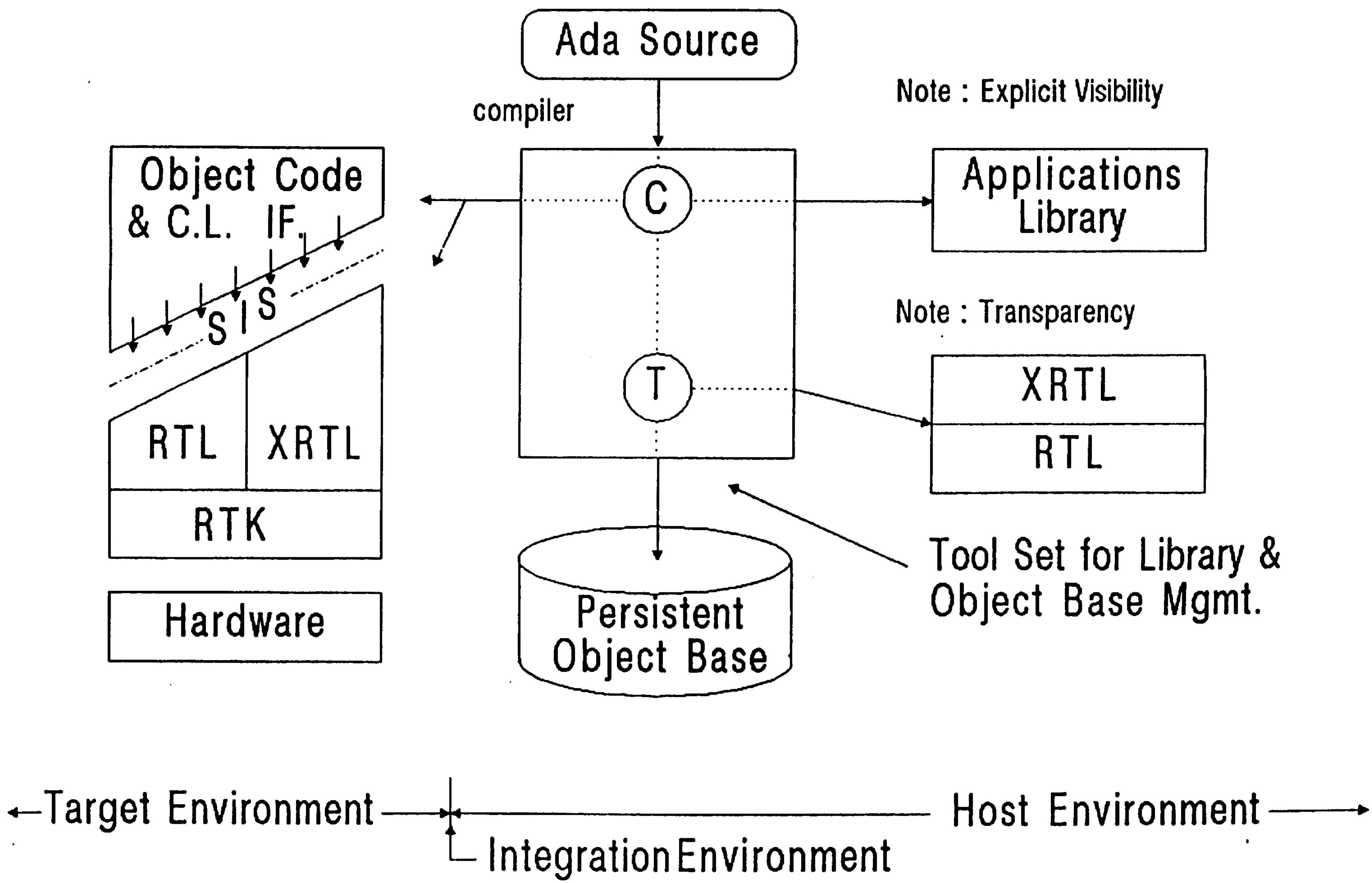
9. A redundancy management subsystem for services and resources which life and property depend upon

10. A stable storage subsystem for each cluster

11. A management subsystem for distributed, nested transactions

12. A multiversion, fault-tolerant programming capability with a granularity within any program which extends at least to the subtransaction level and explicitly identifies the recovery capabilities at that level

A Context for the Software Safety Model



MASC Support Kernel Components

1. A tailorable RTSE developed & sustained in Ada

2. Software structuring which facilitates: firewalling, layered recovery capabilities, dynamic reconfiguration and extensibility

3. Pools of processes and processors capable of non-stop operation in a fault-tolerant environment

4. A command language interface between the SIS of the integration environment's PCEE and the SIS of the target environment's PCEE

5. System-wide, lifecycle-unique identification of all objects and transactions / subtransactions

6. Dynamic, multilevel security in the integration & target environments

The Extended Runtime Library

RTL – routines to support ANSI/MIL–STD–1815A (Ada)

XRTL – legal extensions to language via RTL interfaces

ARTEWG CIFO

What is Ada ?

What should a language be expected to support ?

Fault Tolerance, Realtime, Distribution, etc....

PCEE – available to system software *and applications software*

Ada Runtime Environment Working Group
ARTEWG

Catalog of Interface Features and Options *CIFO*

Currently => uniprocessor-based interfaces

Dynamic Priorities, Bounded delays, Fast interrupts, etc

Under development =>

Multiprocessor-based interfaces

Interfaces for

- Distributed Systems
- Multiprogramming Issues
- Fault Tolerance Issues
- Model Runtime Interfaces (compiler-object code)

if

'safety' is adequately addressed in the host environment throughout development/acquisition

then

the support of this explicitly addressed set of safety specifications will be dependent upon a runtime environment built to:

1. Monitor the system and detect faults that enter the system state vectors ASAP

2. Firewall their propagation

3. Analyze their effects

4. Recover safely

end if;

The SERC PCEE Approach

- Extended Runtime Library
  - Sharable Resources and Services
  - Not "Ada++" !!

- Mission And Safety Critical (MASC) Kernel
  - A model for the underlying Runtime System (RTS)
  - Consists of 12 distinct submodels

Fault *Avoidant* System

- Hardware is the last (only ?) line of defense

Fault *Tolerant* System

- Faults in the target environment do not bring down the system
- Faults are recognized as inevitable in the design of
  - application software
  - system software

The PCEE is a model for such a fault tolerant system, which supports safety in all three environments, especially the integration and target environments

Host Environment

Target Environment

Distributed, Nonstop, Realtime, Data-driven, ...

Integration Environment

Focal point for "configuration" management

Focal point for controlling computer-based activities of
system and application hardware (processors, devices, etc...)
system and application software

Requires an interface to the computer system
"Command Language"

Requires a powerful database representation (EA/RA)

"If it ain't broke, don't fix it."

Anonymous

Its "broke" if it no longer meets requirements,
or
if somebody else is doing it better.

Safety : the probability that a system, including all hardware and software and human – machine subsystems, will provide appropriate protection against the effects of faults, which, if not prevented or handled properly, could result in endangering lives, health, property and environment (CWM, 1987)

Present Approach(s) => *Fault Avoidance*

Design as well as possible
Test as well as possible
Apply hardware redundancy and hope for the best

Who tests the testers ? ==> Faults DO get into the target !

"Acts of God", if no other reason ...

MI 50-

Lifecycle Support for "Computer Systems & Software Safety" in the Target & Integration Environments of the Space Station Program

Approaches to Fault Tolerant Software Systems

Pat Rogers
Software Engineering Research Center

Overview of Presentations

<u>David Auty</u>

- SIS issues for Host and Integration Environments
  - New issues in Testing, V&V, etc.

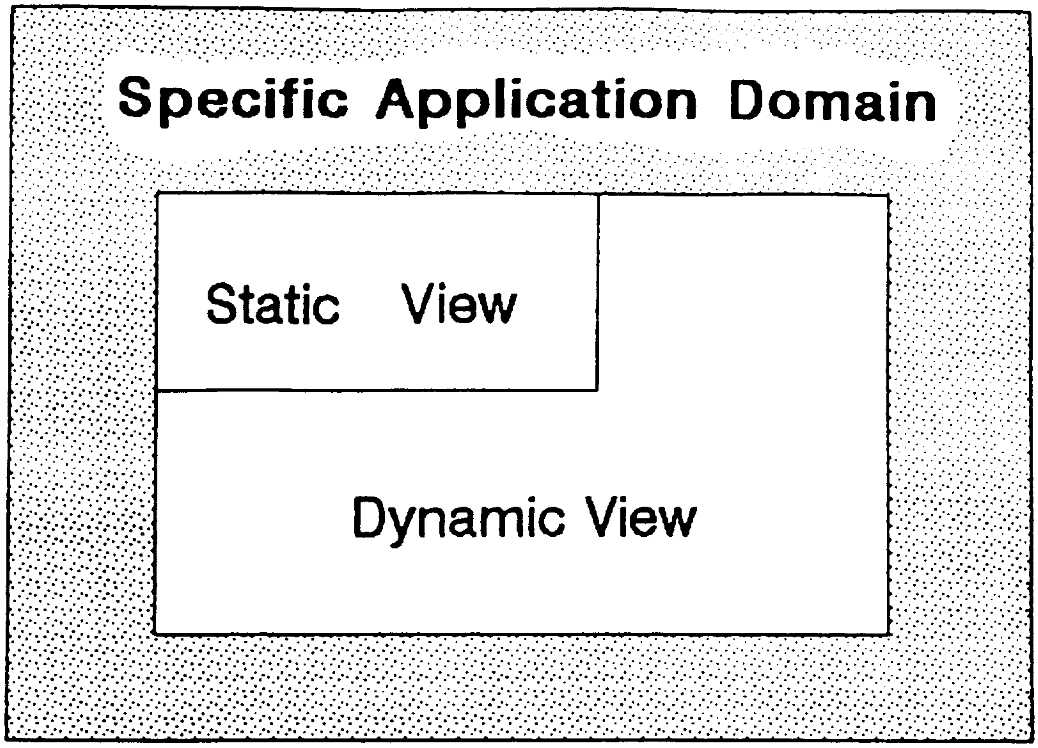
<u>Charlie Randall</u>

- Integration and Target Environment SIS issues
  - Distributed Ada

<u>Kathy Rogers</u>

- SIS issues for all three environments
  - CAIS, CAIS-A, PCTE for SSP

Some "Perspectives" of Software Requirements



Static "Gains" +

Software Development Team

Funcional and Nonfunctional Requirements

Behavioral Assertions

Software Quality Mgmt. Team

Funcional and Nonfunctional Requirements

Behavioral Assertions

Hardware Target Team

Normal and Exceptional CFg

Dynamic View of Software Requirements

Some Examples of CSSE Activities

CSSE / SERC Research Examples

- See Thursday's presentations

CSSE / SERC Education & Training Examples

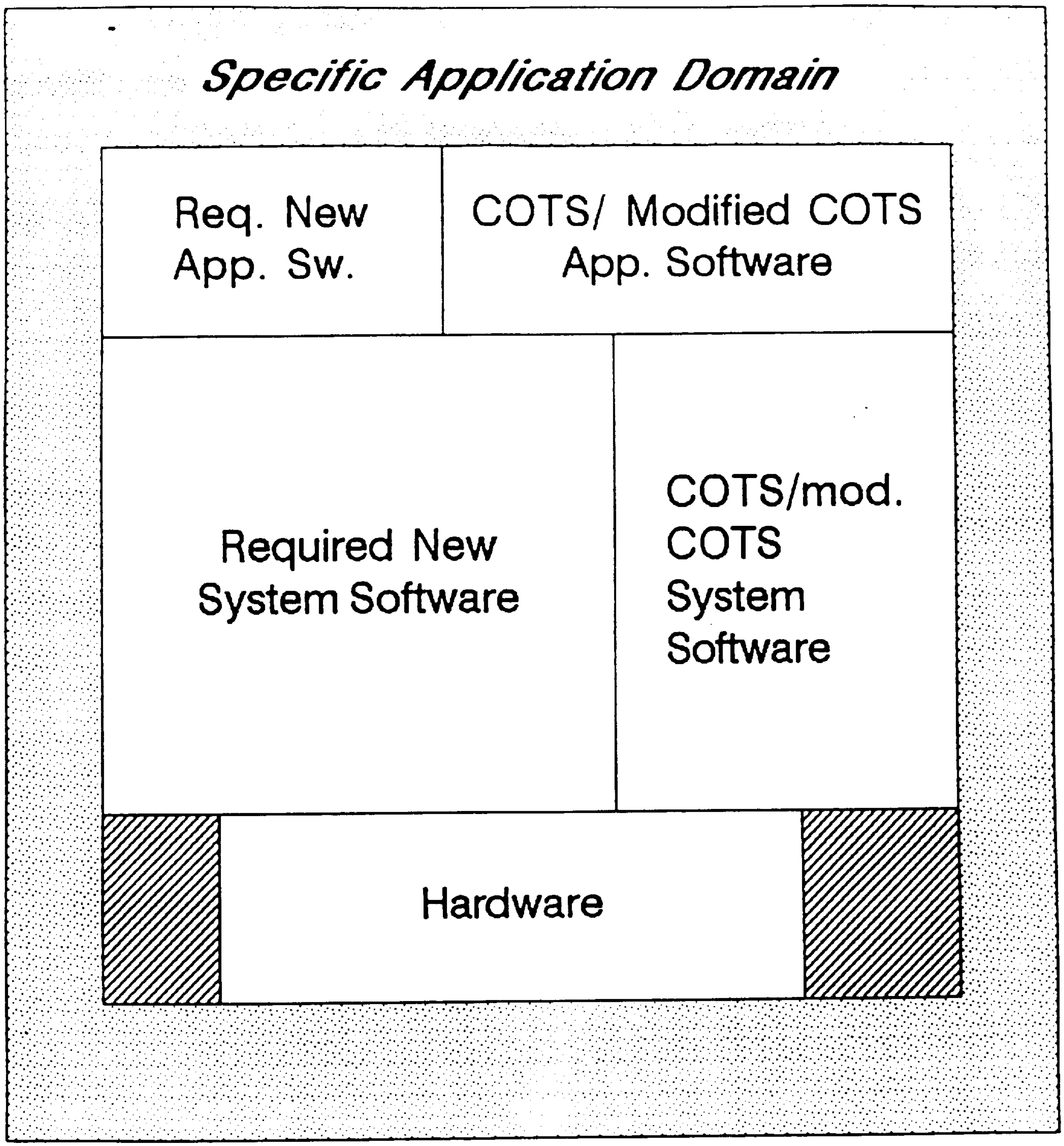
See Dr. Freedman's SEPEC presentation

CSSE / SERC Research on Interfaces to :

- AI / Expert Systems
  - Project with : Inference, Intellimac, JSC / MPAD, UH CL
- Information Management
  - IRDS
  - Object – Based Management Systems
    - U of Colorado, Boulder*
- Math/Stat Analyses
  - Metrics / Instrumentation Design
    - U of Maryland*
  - Reliability Modelling
    - Purdue*

* Reusability : UHCL + 6 others universities

Some "Perspectives" of Software Development



CSSE View of Requirements

Hard Work

RATIONALE

Software and computing components are becoming an increasingly integral part of NASA systems

The size, capability, and complexity of NASA systems are increasing rapidly with time

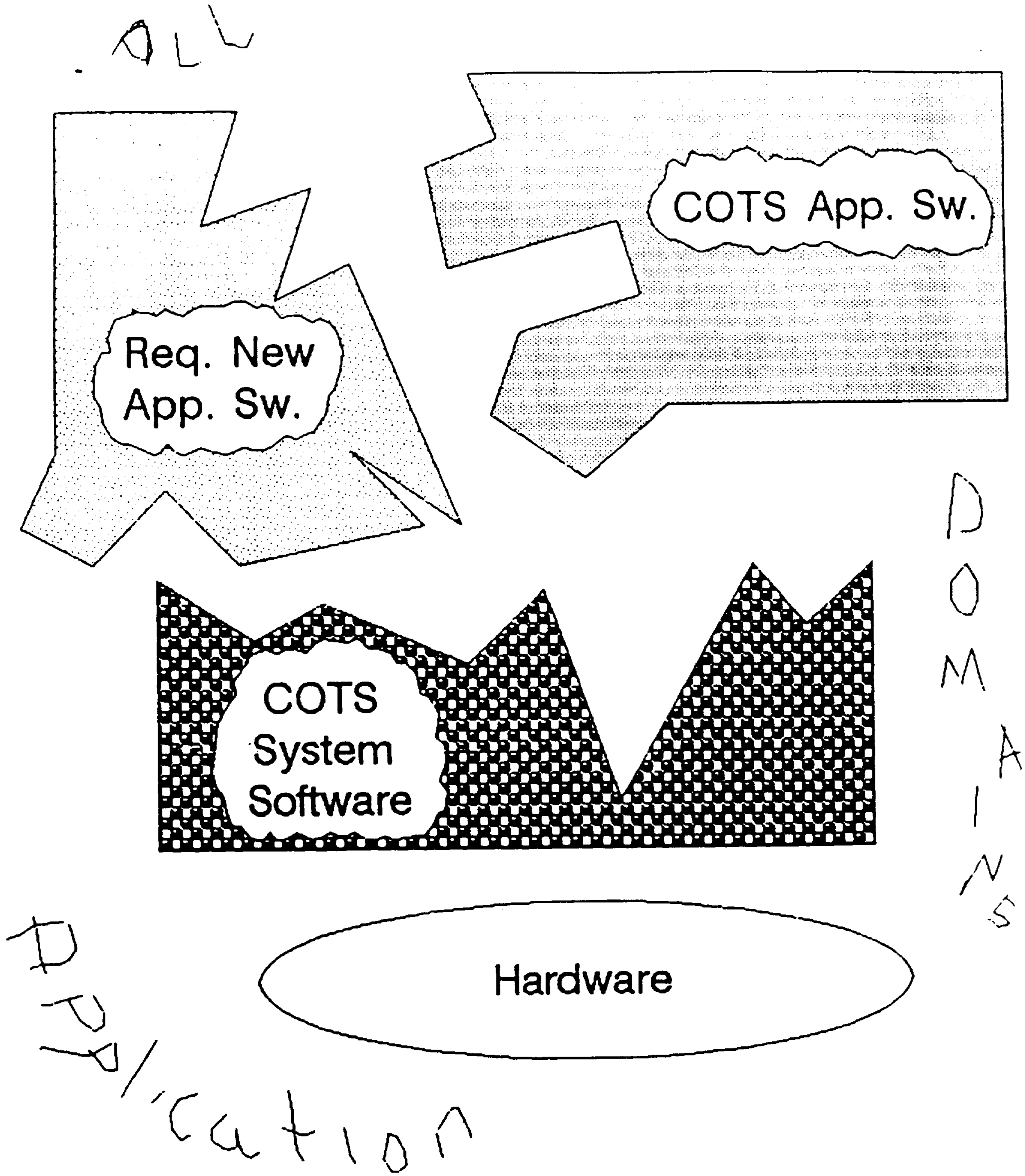
Software failures are intolerable

Software complexity outpaces improvements in software productivity

Software complexity has caused functional descoping

Complex software has enormous maintenance costs

Some "Perspectives" of Software Development



COTS Panacea View of Requirements Work/Play

CASE HISTORIES
CONTINUED

COMPLEXITY HAS CAUSED FUNCTIONAL DESCOPING

VOYAGER URANUS encounter endangered due to delayed software deliveries and descoping of software system

AFTI/F-16 first flight delayed more than 1 year, and then only normal modes used, not 6 advanced modes originally planned

Full flight control system in X-WING cancelled due to software costs exceeding available budget

COMPLEX SOFTWARE INCURS ENORMOUS MAINTENANCE COSTS

Shuttle Ground Processing Support - 12M LOC, 420 people

Shuttle Software Production and Maintenance - 2.5M LOC, 378 people

A Proposed
Clear Lake Model for Computer Systems and Software Safety
in a
Portable Common Execution Environment (PCEE)

A baseline from which subsequent progress in the target environments and integration environments may be made

An extensible model which can also scale down to improve safety in smaller, simpler applications

A "lessons implied / learned" stimulus and opportunity to develop host environment methodologies and tools which better address the lifecycle issues of safety

RELATED RESEARCH

DoD -- Ada Standard language, run-time environments

SPC -- Reusability, prototyping to reduce costs

SEI --- Technology transition: Ada, environments, tools

DARPA -- Common LISP, formal specification language, operating systems, persistent object bases

NSF -- Research in software engineering

AF/RADC -- Knowledge-based software assistants

MCC/Software Technolgy Program

NONE are specifically focusing research on reliable software or management of complexity

RICIS Umbrella

Computer Systems and Software Engineering (CSSE)

plus

SERC (Software Engineering Research Center)

Research
Education and Training (SEPEC)

plus

Interfaces to:

Artificial Intelligence and Expert Systems
Information Management
Mathematical and Statistical Analyses

APPROACH

RESEARCH THEORETICAL FOUNDATIONS FOR MANAGING COMPLEX SOFTWARE SYSTEMS

RESEARCH THEORETICAL FOUNDATIONS FOR RELIABLE SOFTWARE DEVELOPMENT AND VALIDATION

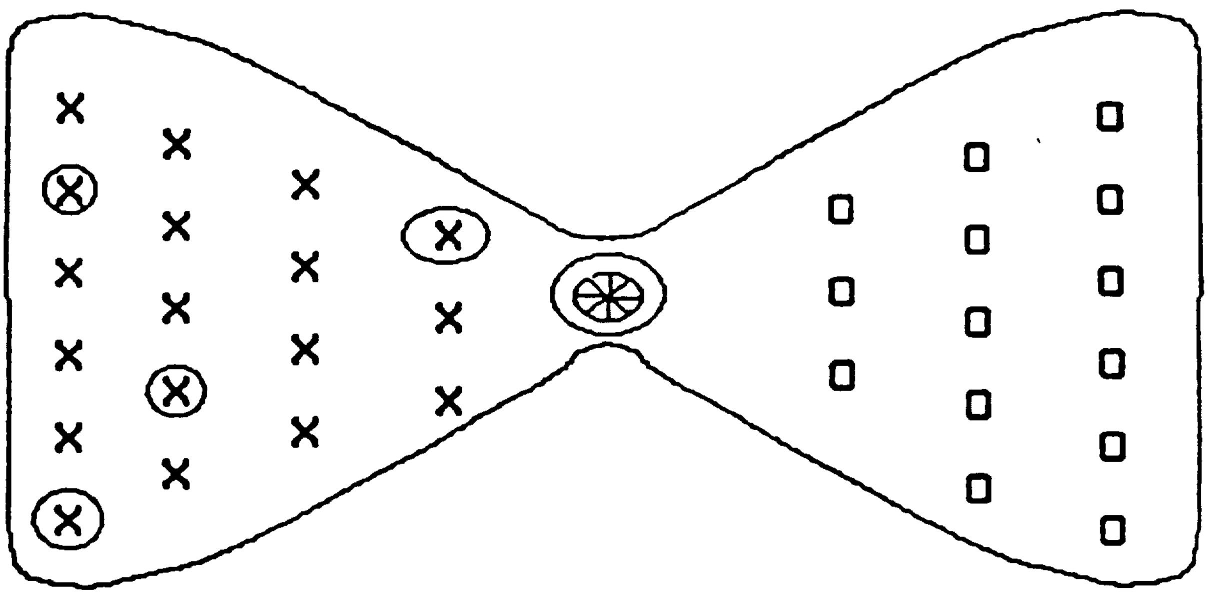
BENCHMARK NASA SOFTWARE ENVIRONMENT

EXPERIMENT IN THE NASA CONTEXT

EVOLVE PRESENT NASA METHODOLOGY

TRANSFER TECHNOLOGY TO SPACE STATION SOFTWARE SUPPORT ENVIRONMENT

TWO SCENARIOS FOR SSP ENVIRONMENT IN 2000+ A.D.



HOST ENVIRONMENTS:

- DEVELOP
- SUSTAIN

INTEGRATION ENVIRONMENT:

- CONTROL OF TGT. ENVIR. BASELINE
- INTEGRATION V&V FOR NEXT BASELINE AND TEST & INTEGRATION PLANS

TARGET ENVIRONMENTS:

- DEPLOY
- OPERATE

RESEARCH ON THEORETICAL FOUNDATIONS FOR RELIABLE SOFTWARE DEVELOPMENT AND VALIDATION

FORMAL SPECIFICATION

PROGRAMMING LANGUAGES

SOFTWARE PROTOTYPING

FAULT TOLERANCE

COMPUTER-AIDED SOFTWARE ENGINEERING SYSTEMS

SOFTWARE REUSE

FORMAL VERIFICATION

SOFTWARE SAFETY

OTHER IDENTIFIED APPROACHES

Computer Systems and Software Engineering

Charles W. McKay

Director

High Technologies Laboratory

Software Engineering Research Center

UH – Clear Lake

EVOLVE PRESENT NASA METHODOLOGY

CAPTURE MOST PROMISING EXPERTISE IN KNOWLEDGE BASE FOR EXPERT SYSTEM

TRANSFORM INTO PROCEDURE OR TOOL/ENVIRONMENT

Clear Lake Model

- Distribute Ada entities
- Interface to virtual Ada machine
- Provide functionality via XRTL
- Transparent reconfiguration
- Application directed reconfiguration
- Surrogates and agent tasks

GHG

TECHNOLOGY TRANSFER TO

SPACE STATION SOFTWARE SUPPORT ENVIRONMENT

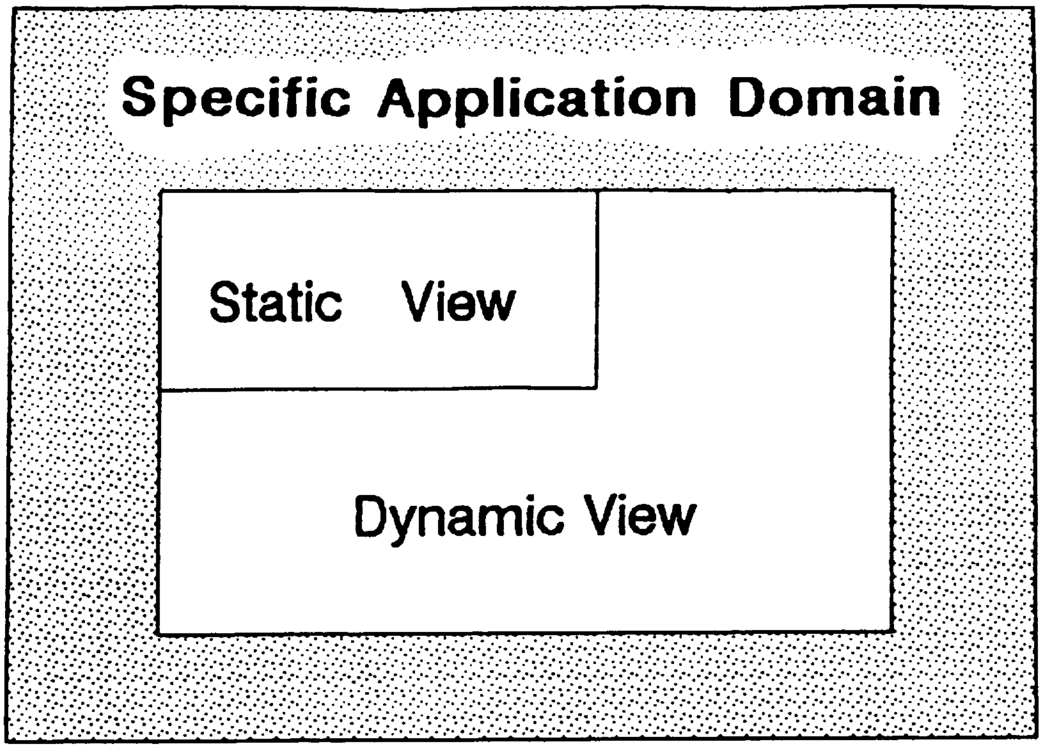
A COMMON ENVIRONMENT FOR ALL SPACE STATION SOFTWARE DEVELOPMENT AND MAINTENANCE

CONTRACT AWARDED TO LOCKHEED MISSILES AND SPACE COMPANY WITH INTERIM SYSTEM FOR 2 YEARS, INCREMENTAL BUILDS OVER 6 YR

AN IDEAL TESTBED FOR NASA SOFTWARE RESEARCH RESULTS AND FOR TRANSITION OF NEW SOFTWARE TECHNOLOGY INTO PRACTICE

MEMORANDUM OF UNDERSTANDING BETWEEN OAST AND OSS CURRENTLY UNDER CONSIDERATION

Some "Perspectives" of Software Requirements



Static "Gains" +

Software Development Team

Funcional and Nonfunctional Requirements

Behavioral Assertions

Software Quality Mgmt. Team

Funcional and Nonfunctional Requirements

Behavioral Assertions

Hardware Target Team

Normal and Exceptional CFg

Dynamic View of Software Requirements

KEYNOTE ADDRESS

The Real Technologies in Space Station Information Systems

John R. (Jack) Garman, Director of Information Systems Services, Space Station Program Office, NASA Headquarters

MASC Support Kernel Components

1. A tailorable RTSE developed & sustained in Ada

2. Software structuring which facilitates: firewalling, layered recovery capabilities, dynamic reconfiguration and extensibility

3. Pools of processes and processors capable of non-stop operation in a fault-tolerant environment

4. A command language interface between the SIS of the integration environment's PCEE and the SIS of the target environment's PCEE

5. System-wide, lifecycle-unique identification of all objects and transactions / subtransactions

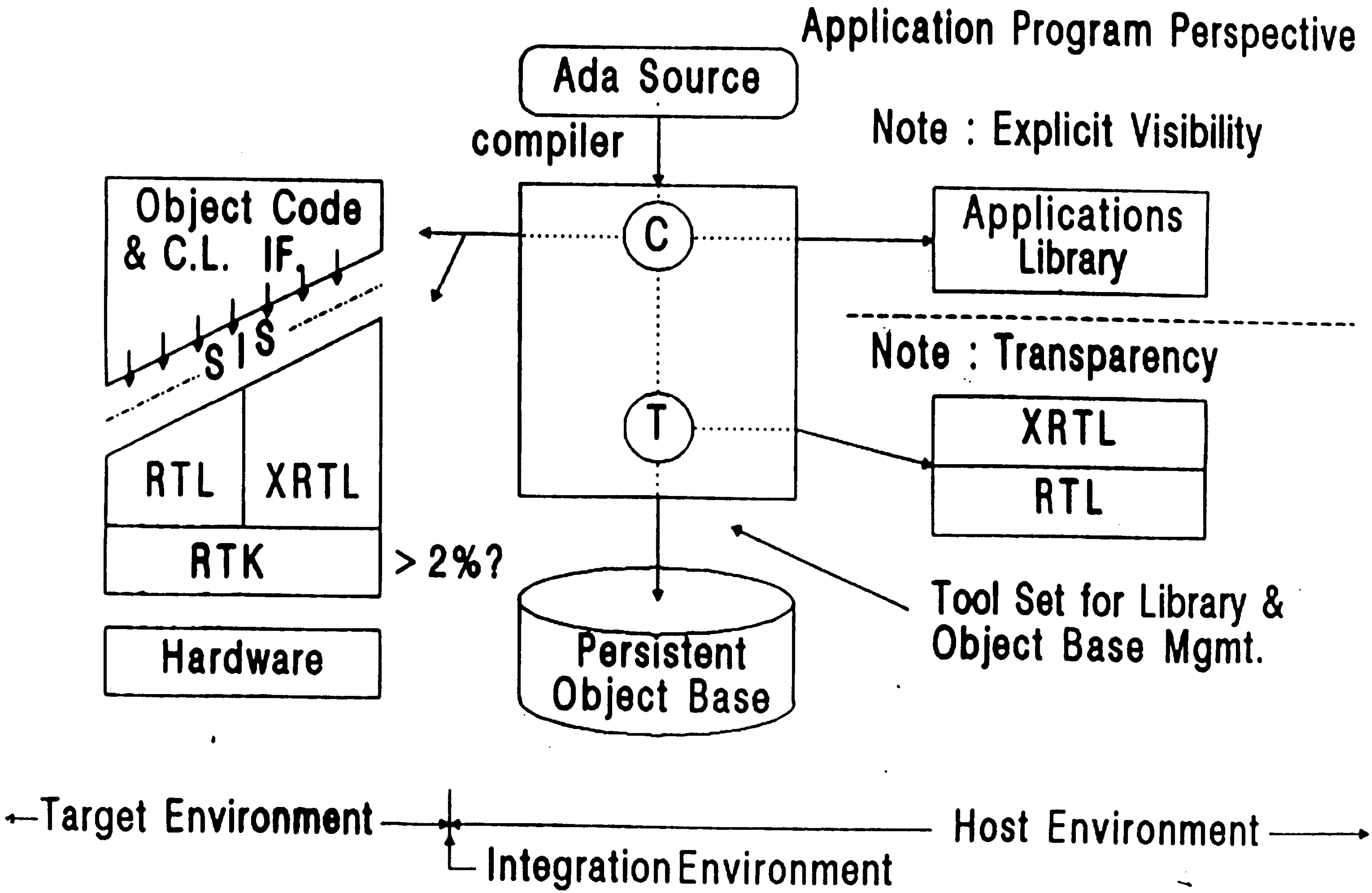
6. Dynamic, multilevel security in the integration & target environments

Keynote Address

The Lead Technologies in Space Station Information Systems

John R. Garman

A Model for Supporting a 'Bare Machine' Philosophy for 'Safety Kernels' of Ada Runtime Support Environments (Ada RTSE's)



RICIS RESEARCH AREAS

Mr. David Auty presented "Testing and Verification of Ada Flight software for Embedded computers." This presentation focused on the issues of storage management in Ada. Mr. Auty described certain aspects of the language which, if misused, could lead to errors which are difficult to detect. Examples include global variables, exception propagation, dynamic task interactions, generic subprograms parameters, and dynamic storage allocations. He distinguished among good software engineering approaches to utilizing these features, compiler dependencies which affect these features and the critical role of the run time support environment int he robust and correct management of storage. He then described some of the recommendations and guidelines that have resulted from this study.

The final presentation of this session was made by Dr. Morris Liaw. Dr. Liaw and his colleague, Dr. Anthony Lekkos, recently delivered an operational Software Engineering and Ada Database. He described the objectives and the history of the project as well as the architecture and the features of this unique resource. He further described the methodology used in design and development and concluded with the description of the planned enhancements for the future. A second release will be available in January 1988. The resource is being used by JSC, UH-Clear Lake and JSC's aerospace contractors.

EDUCATION AND TRAINING

Conveners: **Glenn B. Freedman,** UH-Clear Lake

Amy B. Kennedy, Employee Development, NASA/JSC

Review of the Education and Training Activities

Glenn B. Freedman, UH-Clear Lake
Sue LeGrand, SofTech,Inc.

Software Engineering and the Transition to Ada

John McBride, SofTech, Inc.

Computer Based Ada Training Using Hypertext Systems

Sue LeGrand and **Gilbert Marlowe**, SofTech, Inc.

Summary of the Computer Systems and Software Engineering Technical Session

Charles W. McKay

Because it was impossible to provide a meaningful presentation on more than 18 NASA funded activities as well as an even greater number of company funded activities that are coordinated by this component of RICIS, five related activities were selected for presentation. Mr. Pat Rogers introduced the five activities and then presented: "Lifecycle Support for Computer Systems and Software Safety in the Target and Integration Environments of the Space Station Program: Approaches to Fault Tolerant Software Systems."

Safety was defined as "the probability that a system, including all hardware and software and human-machine subsystems will provide appropriate protection against the effects of faults, which, if not prevented or handled properly, could result in endangering lives, health, property and environments." The past and present approaches to mission and safety critical components have been addressed through a static perspective of fault avoidance (i.e., considerations in the host environment only). That is, the development team was encouraged to design as well as possible to keep defects out of the system. The quality management team was encouraged to test as well as possible to identify defects that made it through the work of the development team so that the defects could then be removed before deployment. Post-deployment support depended almost entirely upon hardware techniques (e.g., redundant processors, built-in-tests, error coding) to sustain mission and safety critical components at run time.

Mr. Rogers advocated the CSSE team's position that a dynamic perspective of software assessment and control of run time behavior in the target and integration environments is needed to complement the static perspective which has previously been emphasized. Specifically, as shown in Figure 1, additional software processes should be deployed in the target environment to accompany all mission and safety critical components. For applications programs, these additional processes help to monitor the behavior of each of the critical components. These processes are needed to insure the fastest possible identification of faults that have entered any porion of the system state vector, to firewall their propagation, to analyze which of the predetermined recovery mechanisms are most appropriate, and to effect recovery. At the systems software level, such processes apply to all shareable services and resources which mission and safety critical components of application software depend upon.

Mr Rogers then described the CSSE team's proposal for a Portable Common Execution Environment (PCEE). The two principal components of this proposal are an extended run time support environment library and a Mission and Safety Critical (MASC) kernel. Underneath the MASC kernel are 12 distinct but highly interactive models believed essential to maximize the support for mission and safety critical requirements. Figure 2 depicts the extended run time library model and Figure 3 depicts the list of 12 models underneath the system interface set of the MASC kernel.

Summary of the Education and Training Technical Session

Dr. Glenn B. Freedman

In the first presentation, Dr. Freedman reviewed two RICIS activities. He first presented the results of a contractor survey completed in Fall, 1986, in which 21 NASA contractors were interviewed to assess the extent to which they had undertaken software engineering and Ada training programs locally, what their perceptions were about these areas and what their plans were for training and educational activities for the next twelve months. The results indicated that at the time of the survey and interviews, the contractor community had very little software engineering training planned, but were beginning Ada syntax training, even though there was little Ada work in progress. The interviewees indicated that the commitment of NASA mid-level managers toward Ada was not firm and that this perception affected training plans. As one personnel person commented, there was "Ada talk from on high, but no Ada action."

Other findings were that the contractors had hardware, compilers and various tools available, they the general consensus at the time was that the tools and methods for Ada were immature. Training was typically defined in terms of language syntax and semantics, with little regard for the Ada culture that supports software engineering principles and goals. Interestingly, the companies perceived that there were sufficient numbers of programmers available, but few software engineers and design experts. Nonetheless, little in the way of design and software engineering training was planned. One of the most consistent findings was that no "transition to Ada" plans were mentioned, even though each company recognized that Ada would become a language they would be using and that the transition would be resource intensive to some extent.

Freedman also reported on the development of a number of training options for the aerospace community. On of the options was a videotape featuring modules about various aspects of software engineering. The tape, geared to upper level management, contains four modules: The Cost of Software, Software in the Space Station Era, Engineering Software and Building a Software Engineering Environment. A second development effort resulted in a one-half of three day presentation covering software engineering and the highlights of the Ada programming language. Other efforts led to an Ada glossary, a PC-based data base of software engineering and Ada training options, text resources, conferences and other educational and training information.

Among the deliverables to NASA there has also been a model for a comprehensive software engineering curriculum that features six planning dimensions and alternative training methods. Also, Freedman discussed the programs of the Clear Lake Software Engineering and Ada Training Forum, a monthly meeting of training experts from universities and industry, and the Software Engineering Professional Education Center, a complementary center to the Software Engineering Research Center that enables the university to offer a full range of services and research to the software engineering community.

COMPUTER SYSTEMS AND SOFTWARE ENGINEERING

Conveners: **Charles W. McKay,** UH-Clear Lake

Stephen A. Gorman, Head, Application Systems, Spacecraft Software Division, NASA/JSC

EDUCATION AND TRAINING SESSION

By

Dr. Glenn Freedman, UH-Clear

John McBride, SofTech

Sue LeGrand, SofTech

Gilbert Marlowe, SofTech

PRODUCT DEMONSTRATION FORUM

NOVEMBER 9, 1987

9:00 A.M. - 12:00 NOON

BUILDING 12

PDF ROOM

SOFTECH

sepec

RESEARCH ACTIVITIES

ET.1

SOFTWARE ENGINEERING AND ADA TRAINING, PHASE 1 (FREEDMAN)

ET.2

SOFTWARE ENGINEERING AND ADA TRAINING PHASE 2 (FREEDMAN)

SE.17

ADA ANALYSIS FOR SPACE STATION OFFICE (SUE LEGRAND, SOFTECH, INC.)

ET.4

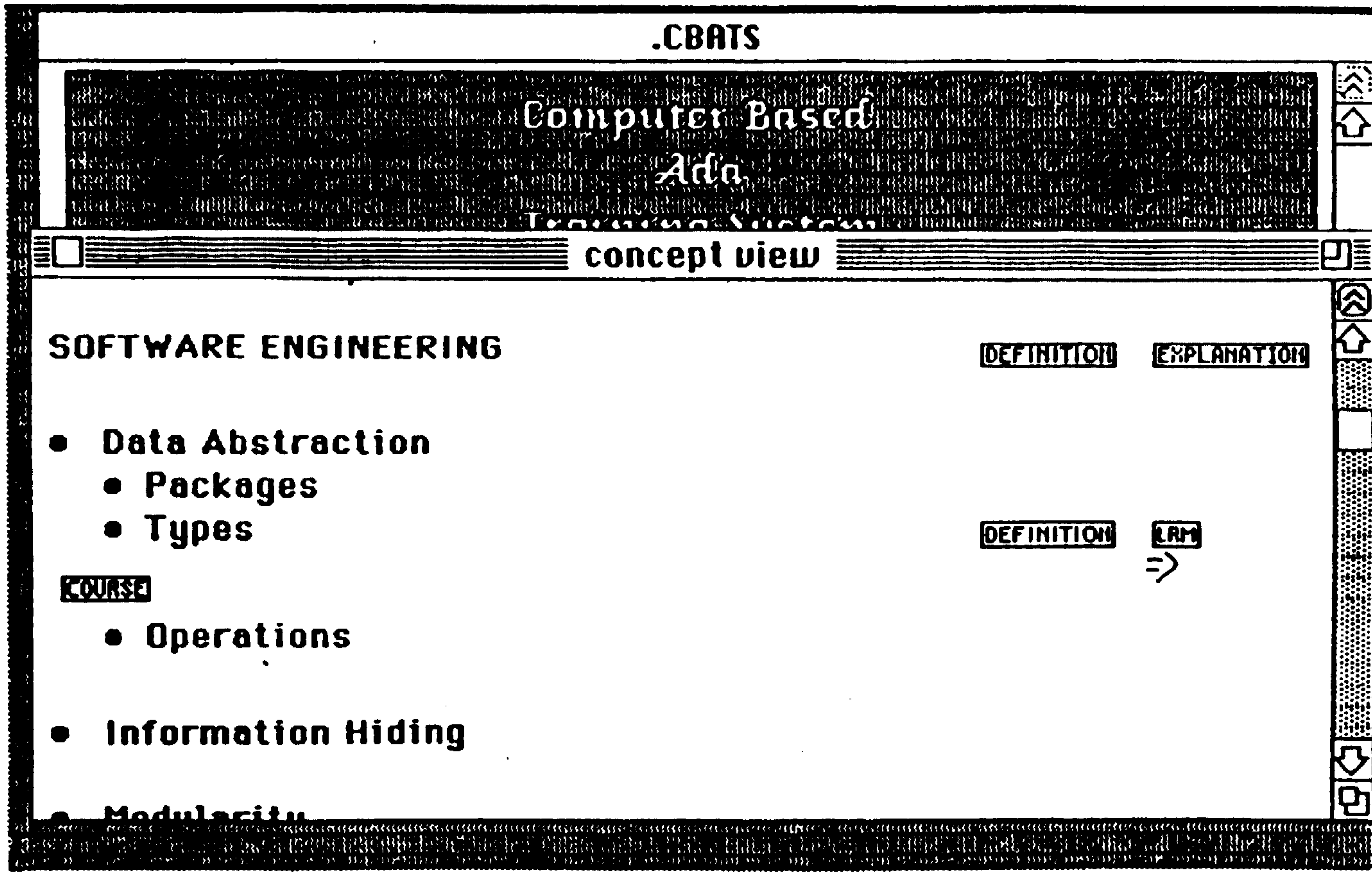
SOFTWARE ENGINEERING AND ADA TRANSITION COURSE (JOHN MCBRIDE, SOFTECH, INC.)

ET.3

COMPUTER BASED ADA TRAINING SYSTEM (SUE LEGRAND, SOFTECH, INC.)

FROM: TYPES

"CLICK" ON: L R M



SOFTECH

sepec

October, 1987

E.T. 1 and E.T. 2
SOFTWARE ENGINEERING AND Ada TRAINING

<u>DELIVERABLES (EXCERPTS)</u>

MARKET SURVEYS

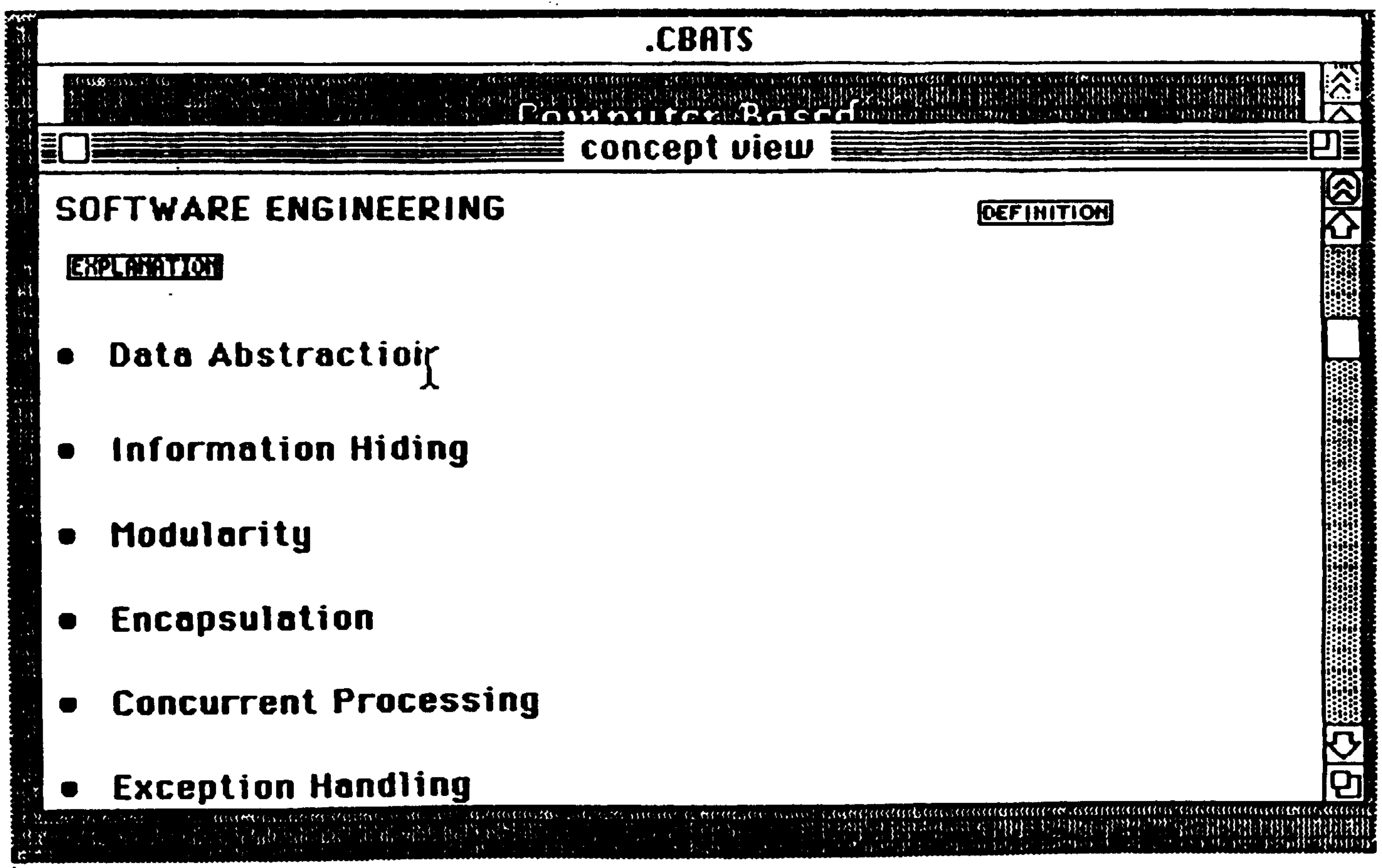
AWARENESS FOR UPPER-LEVEL MANAGEMENT

TRAINING MODELS AND CURRICULAR GUIDELINES

TRAINING INFRASTRUCTURE AND NETWORKS

TRAINING RESOURCES REPOSITORY

FROM:	SOFTWARE ENGINEERING
"CLICK" ON:	DATA ABSTRACTION



SOFTECH

sepec

October, 1987

VIDEOTAPE:

INTRODUCTION TO SOFTWARE ENGINEERING FOR LARGE SYSTEMS

SECTION I
THE COST OF SOFTWARE

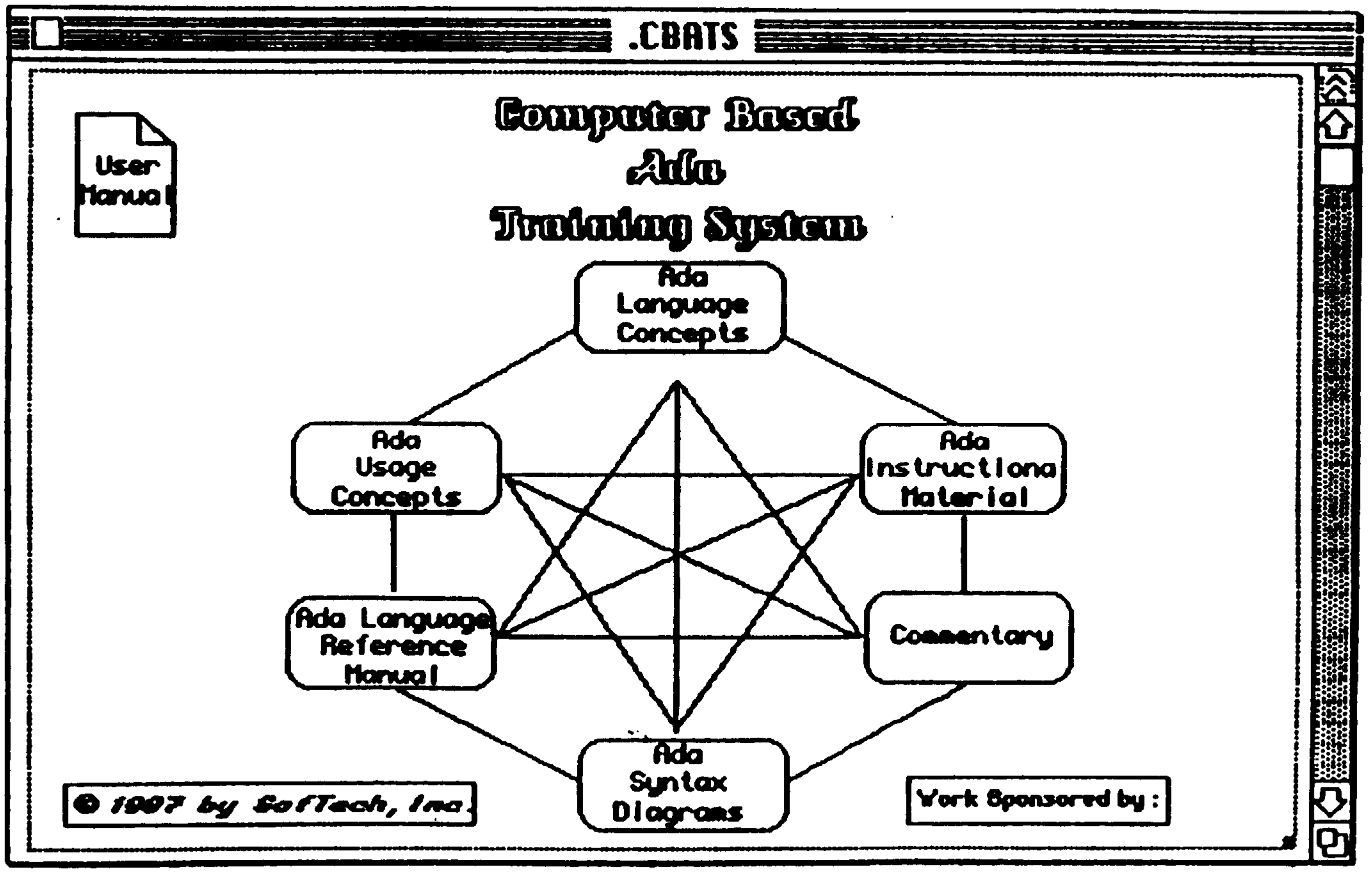
SECTION II
SOFTWARE IN THE SPACE STATION ERA

SECTION III
ENGINEERING SOFTWARE

SECTION IV
BUILDING A SOFTWARE ENGINEERING ENVIRONMENT

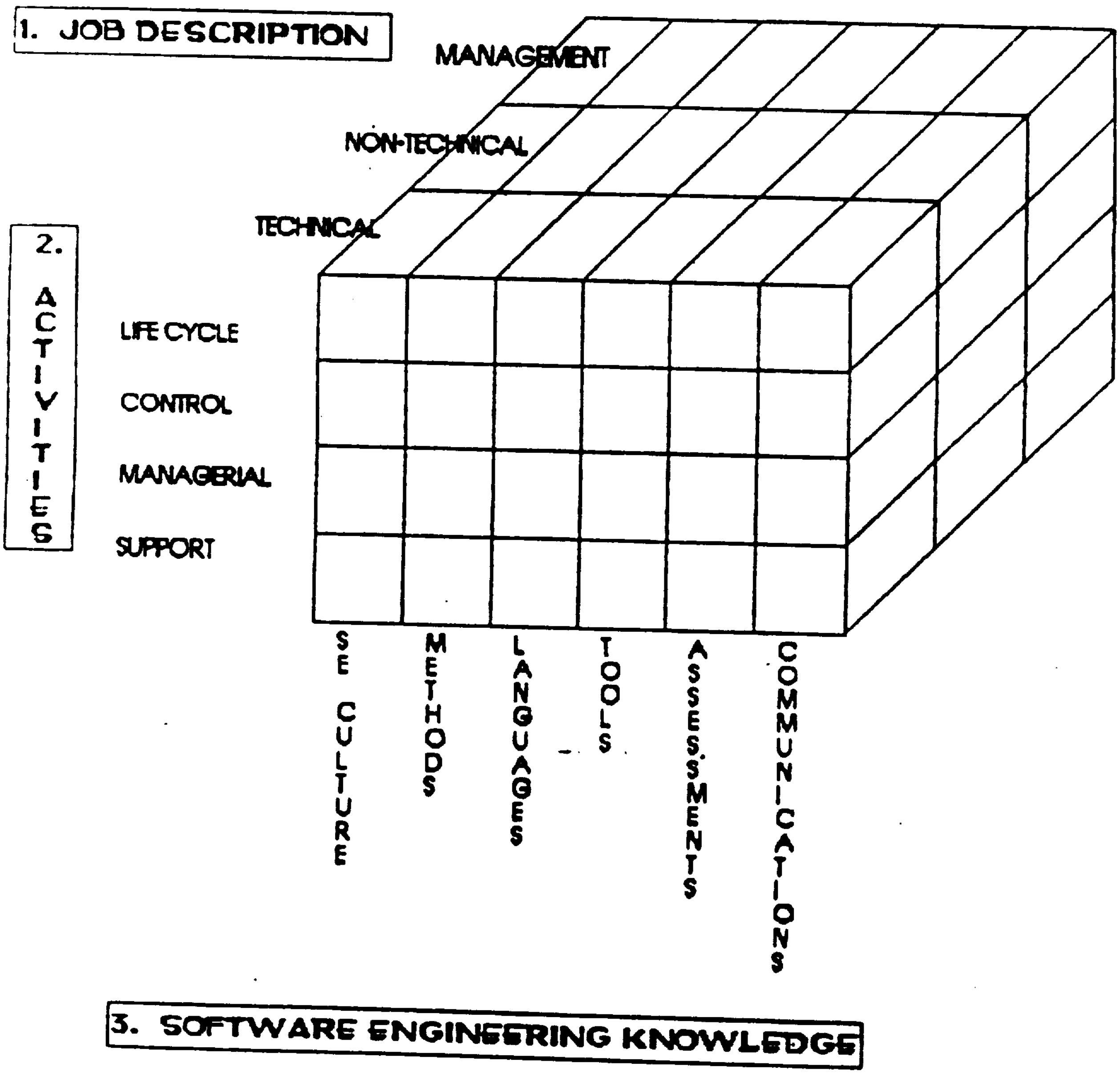
FROM: CBATS LOGO

"CLICK" ON: ADA LANGUAGE CONCEPTS



SOFTECH

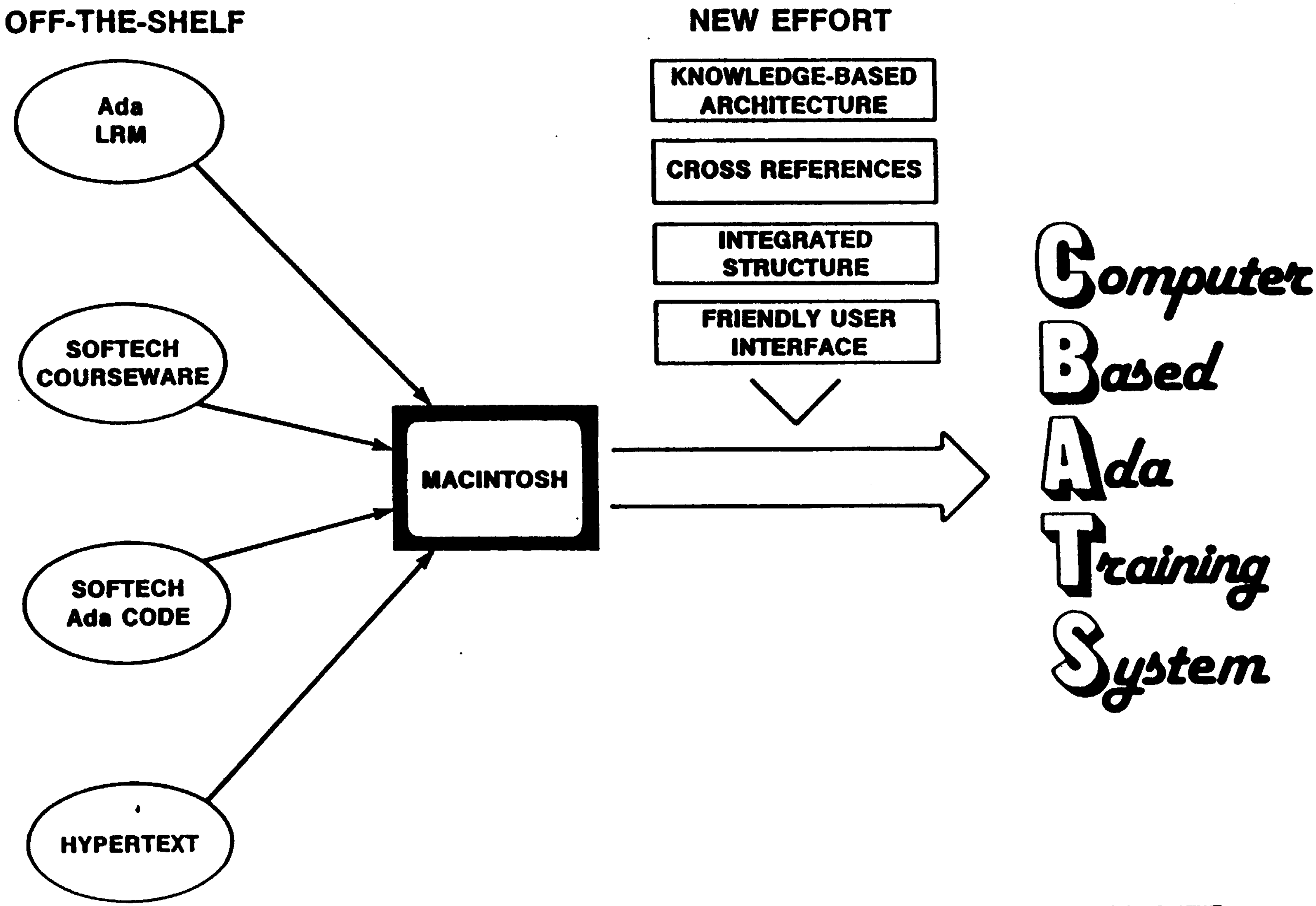
SOFTWARE ENGINEERING WITH ADA:
A DEFINITION OF THE FIELD
WITH CURRICULAR OPTIONS



4. ENVIRONMENT: HOST, TARGET, INTEGRATION
5. SKILL LEVEL: INTRODUCTORY, INTERMEDIATE, ADVANCED
6. PROJECT SIZE/COMPLEXITY: SMALL, LARGE COMPONENT, AI-BASED

8710

SOFTECH APPROACH



DATA BASE FILES

EDUCATION AND TRAINING

ORACLE
dBASE III

COURSE

ORGANIZATION

CONFERENCE

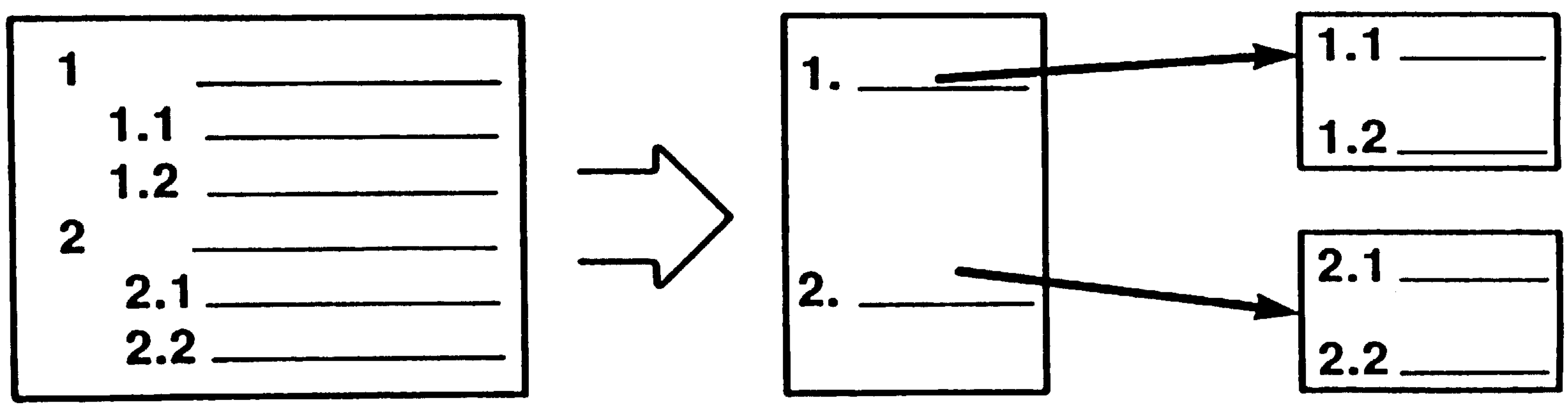
PRODUCT

PUBLICATION

PERSON

BOOK

HYPERTEXT



TEXT COLLAPSED INTO MULTIPLE LEVELS

SOFTECH

NASA SOFTWARE ENGINEERING AND Ada TRAINING REQUIREMENTS

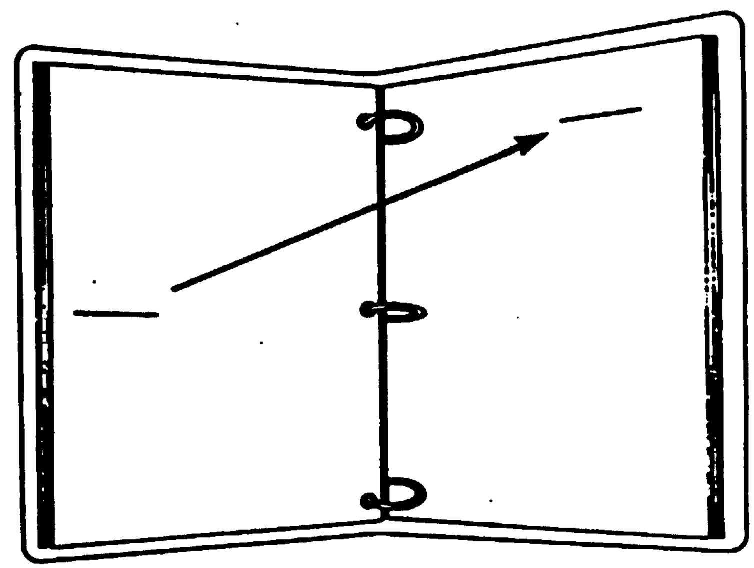
SE. 17

SUE LEGRAND, PRINCIPAL INVESTIGATOR

DR. CHARLES MCKAY, RICIS MONITOR

ROBERT NELSON, NASA MONITOR

HYPERTEXT



REFERENCE WITHIN DOCUMENTS

APPROACH

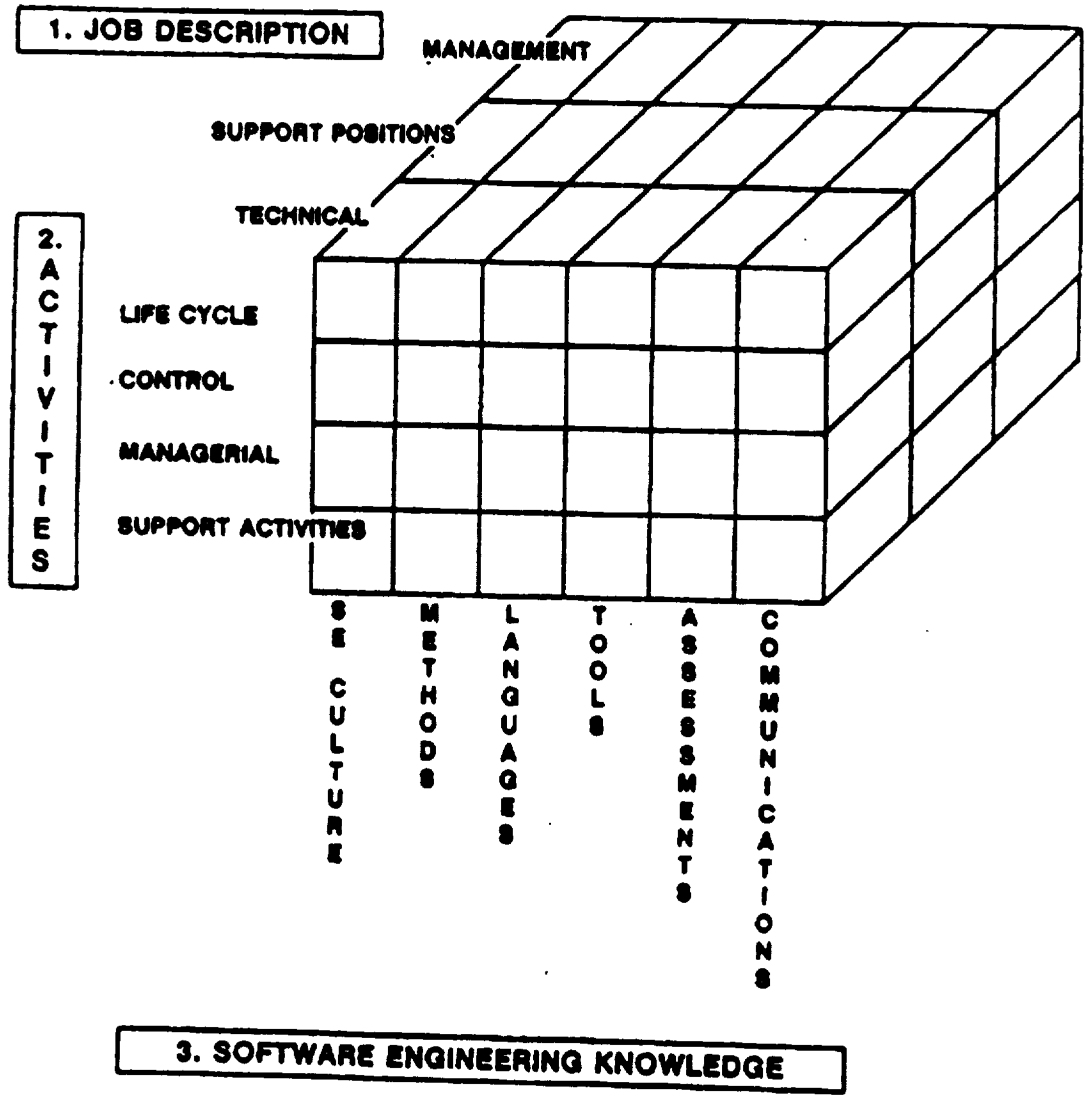
- SURVEY NASA EDUCATION OFFICES FOR PREVIOUS AND PLANNED TRAINING INFORMATION, SUCH AS:
  - PRESENT TRAINING AND EVALUATION POLICIES
  - COURSE SELECTION CRITERIA AND SOURCES
  - RECOMMENDED ASSISTANCE TO TRAINING OFFICE PERSONNEL

AGENDA

- OBJECTIVES
- Ada TRAINING PROBLEM
- DEFINITION OF HYPERTEXT
- SOFTECH APPROACH
- EXAMPLES

figure 2.6a

**SOFTWARE ENGINEERING WITH Ada:
A DEFINITION OF THE FIELD
WITH CURRICULAR OPTIONS**



4. ENVIRONMENT: HOST, TARGET, INTEGRATION
5. SKILL LEVEL: INTRODUCTORY, INTERMEDIATE, ADVANCED
6. PROJECT SIZE/COMPLEXITY: SMALL, LARGE
COMPONENT, AI-BASED

SUMMARY

- THE USE OF MODERN SOFTWARE ENGINEERING AND Ada INVOLVES CRITICAL TRANSITION STEPS

- THE SEMINAR BRIDGES THE CONCEPTS OF SOFTWARE ENGINEERING, Ada AND SOFTWARE SUPPORT ENVIRONMENTS.

- THE SEMINAR PRESENTS ISSUES AND MANAGEMENT OPTIONS

STATUS

- TRAINING RECOMMENDATIONS BEING FORMED FOR PERSONNEL IN:
  - MANAGEMENT
  - TECHNICAL
  - SUPPORT
- IMPLEMENTATION PLAN PROPOSED FOR:
  - CORE CURRICULUM
  - TECHNICAL TOPICS
  - ON-THE-JOB TRAINING

WHO WILL BENEFIT FROM THIS SEMINAR?

MANAGEMENT PERSONNEL INVOLVED IN THE TRANSITION PROCESS MAY INCLUDE:

- PROGRAM MANAGERS
- PROJECT MANAGERS
- TECHNICAL TEAM LEADERS
- COMPUTER RESOURCE MANAGERS
- CONTRACT MANAGERS

THE SEMINAR IS DESIGNED FOR THOSE INVOLVED IN PLANNING OR IMPLEMENTING A TRANSITION TO SOFTWARE ENGINEERING WITH Ada

SofTech

INTRODUCTION

- THE APPLICATION OF MODERN SOFTWARE ENGINEERING PRINCIPLES AND THE Ada LANGUAGE PROMISES HIGHER PRODUCTIVITY AND LOWER LIFE CYCLE COSTS.

- THE TRANSITION TO THESE NEW METHODS, HOWEVER, POSES RISKS.

- A SEMINAR IS BEING DEVELOPED TO ADDRESS TRANSITION ISSUES.

SofTech

WHAT DOES THE SEMINAR ADDRESS?

- SOFTWARE TRENDS
- SYSTEM AND SOFTWARE LIFE CYCLES
- SOFTWARE ENGINEERING PRINCIPLES AND METHODS
- HOW Ada SUPPORTS SOFTWARE ENGINEERING

WHAT IS NEEDED?

MANAGEMENT NEEDS:

- RATIONALE FOR USING MODERN SOFTWARE DEVELOPMENT METHODS.
- OVERVIEW OF SOFTWARE ENGINEERING METHODS.
- COMPREHENSIVE VIEW OF TRANSITION ISSUES.

SofTech

WHAT IS NEEDED?

MANAGEMENT NEEDS:

- RATIONALE FOR USING MODERN SOFTWARE DEVELOPMENT METHODS.
- OVERVIEW OF SOFTWARE ENGINEERING METHODS.
- COMPREHENSIVE VIEW OF TRANSITION ISSUES.

SofTech

WHAT DOES THE SEMINAR ADDRESS?

- SOFTWARE TRENDS
- SYSTEM AND SOFTWARE LIFE CYCLES
- SOFTWARE ENGINEERING PRINCIPLES AND METHODS
- HOW Ada SUPPORTS SOFTWARE ENGINEERING

INTRODUCTION

- THE APPLICATION OF MODERN SOFTWARE ENGINEERING PRINCIPLES AND THE Ada LANGUAGE PROMISES HIGHER PRODUCTIVITY AND LOWER LIFE CYCLE COSTS.

- THE TRANSITION TO THESE NEW METHODS, HOWEVER, POSES RISKS.

- A SEMINAR IS BEING DEVELOPED TO ADDRESS TRANSITION ISSUES.

SofTech

WHO WILL BENEFIT FROM THIS SEMINAR?

MANAGEMENT PERSONNEL INVOLVED IN THE TRANSITION PROCESS MAY INCLUDE:

- PROGRAM MANAGERS
- PROJECT MANAGERS
- TECHNICAL TEAM LEADERS
- COMPUTER RESOURCE MANAGERS
- CONTRACT MANAGERS

THE SEMINAR IS DESIGNED FOR THOSE INVOLVED IN PLANNING OR IMPLEMENTING A TRANSITION TO SOFTWARE ENGINEERING WITH Ada

SofTech

STATUS

- TRAINING RECOMMENDATIONS BEING FORMED FOR PERSONNEL IN:
  - MANAGEMENT
  - TECHNICAL
  - SUPPORT
- IMPLEMENTATION PLAN PROPOSED FOR:
  - CORE CURRICULUM
  - TECHNICAL TOPICS
  - ON-THE-JOB TRAINING

SUMMARY

- THE USE OF MODERN SOFTWARE ENGINEERING AND Ada INVOLVES CRITICAL TRANSITION STEPS

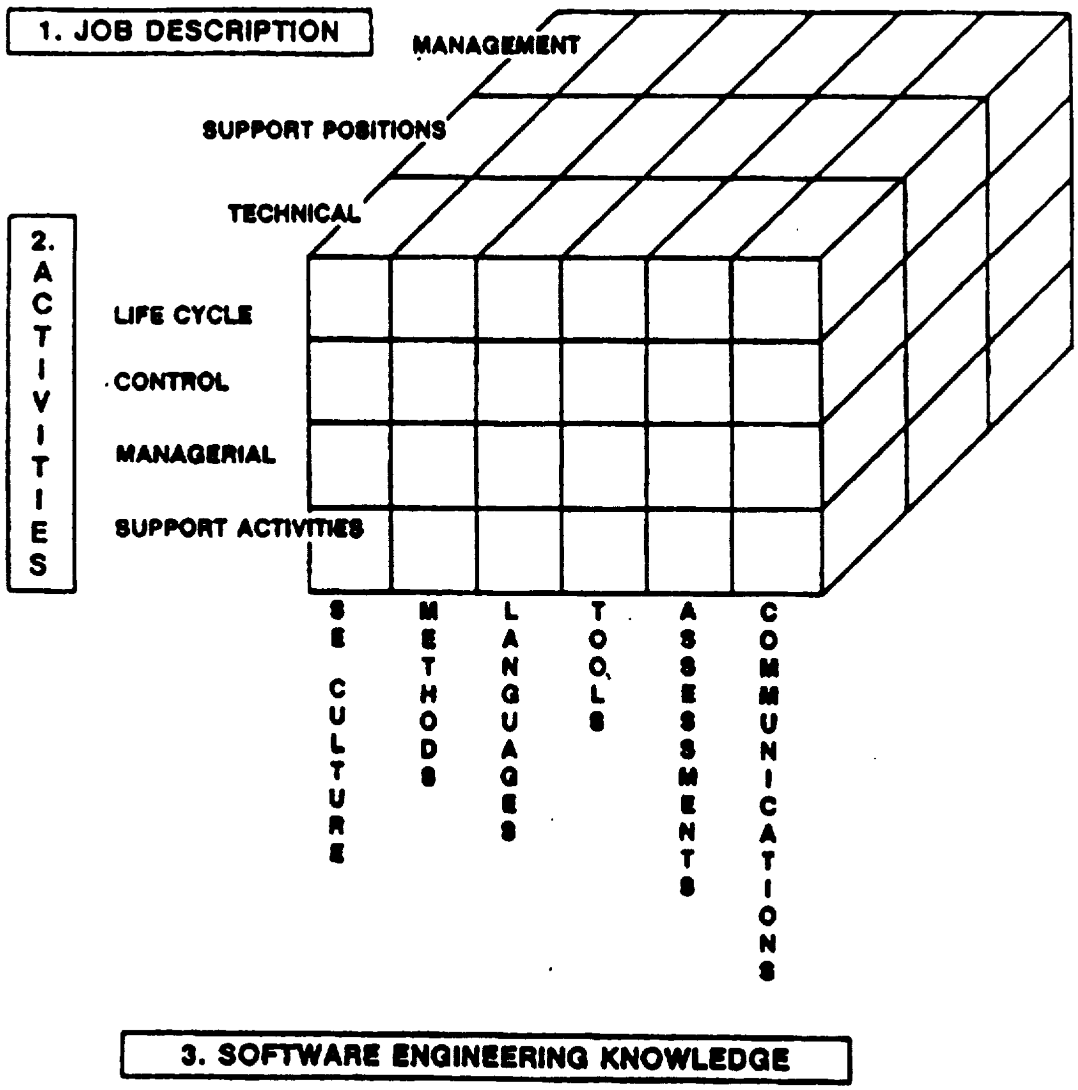
- THE SEMINAR BRIDGES THE CONCEPTS OF SOFTWARE ENGINEERING, Ada AND SOFTWARE SUPPORT ENVIRONMENTS.

- THE SEMINAR PRESENTS ISSUES AND MANAGEMENT OPTIONS

SofTech

figure 2.6a

SOFTWARE ENGINEERING WITH Ada:
A DEFINITION OF THE FIELD
WITH CURRICULAR OPTIONS



4. ENVIRONMENT: HOST, TARGET, INTEGRATION
5. SKILL LEVEL: INTRODUCTORY, INTERMEDIATE, ADVANCED
6. PROJECT SIZE/COMPLEXITY: SMALL, LARGE COMPONENT, AI-BASED

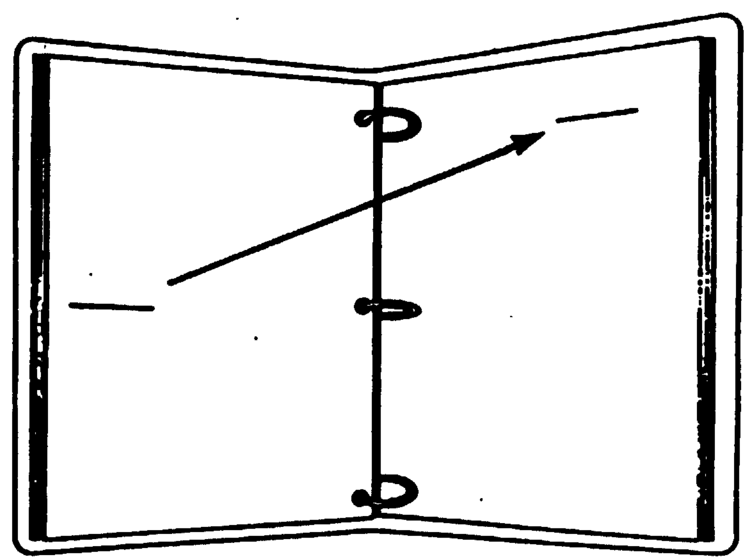
AGENDA

- OBJECTIVES
- Ada TRAINING PROBLEM
- DEFINITION OF HYPERTEXT
- SOFTECH APPROACH
- EXAMPLES

APPROACH

- SURVEY NASA EDUCATION OFFICES FOR PREVIOUS AND PLANNED TRAINING INFORMATION, SUCH AS:
  - PRESENT TRAINING AND EVALUATION POLICIES
  - COURSE SELECTION CRITERIA AND SOURCES
  - RECOMMENDED ASSISTANCE TO TRAINING OFFICE PERSONNEL

HYPERTEXT



REFERENCE WITHIN DOCUMENTS

SOFTECH

NASA SOFTWARE ENGINEERING AND Ada TRAINING REQUIREMENTS

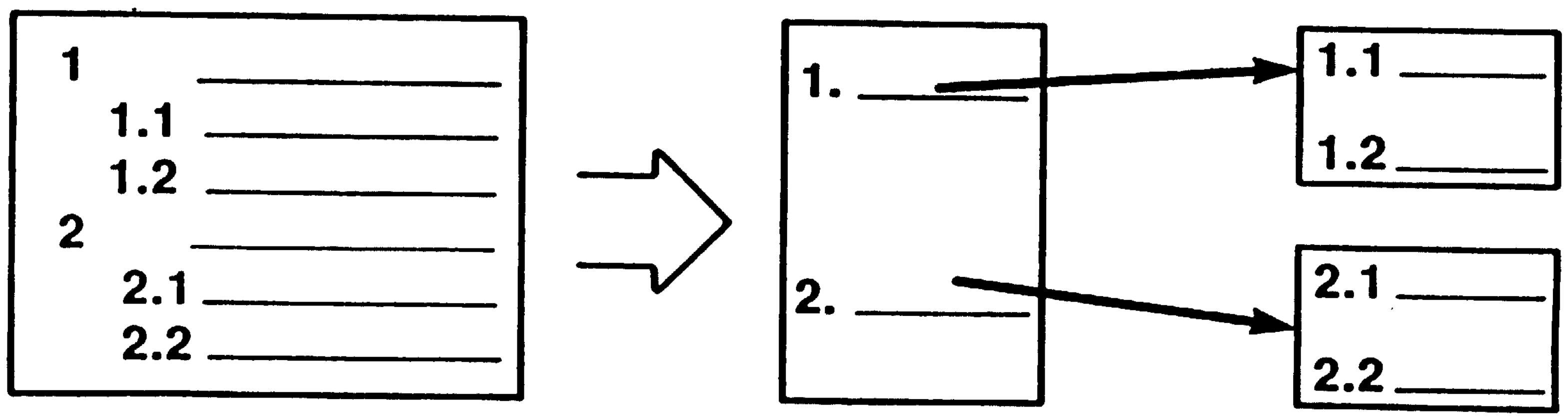
SE. 17

SUE LEGRAND, PRINCIPAL INVESTIGATOR

DR. CHARLES MCKAY, RICIS MONITOR

ROBERT NELSON, NASA MONITOR

HYPERTEXT



TEXT COLLAPSED INTO MULTIPLE LEVELS

DATA BASE FILES

EDUCATION AND TRAINING

ORACLE
dBASE III

COURSE

ORGANIZATION

CONFERENCE

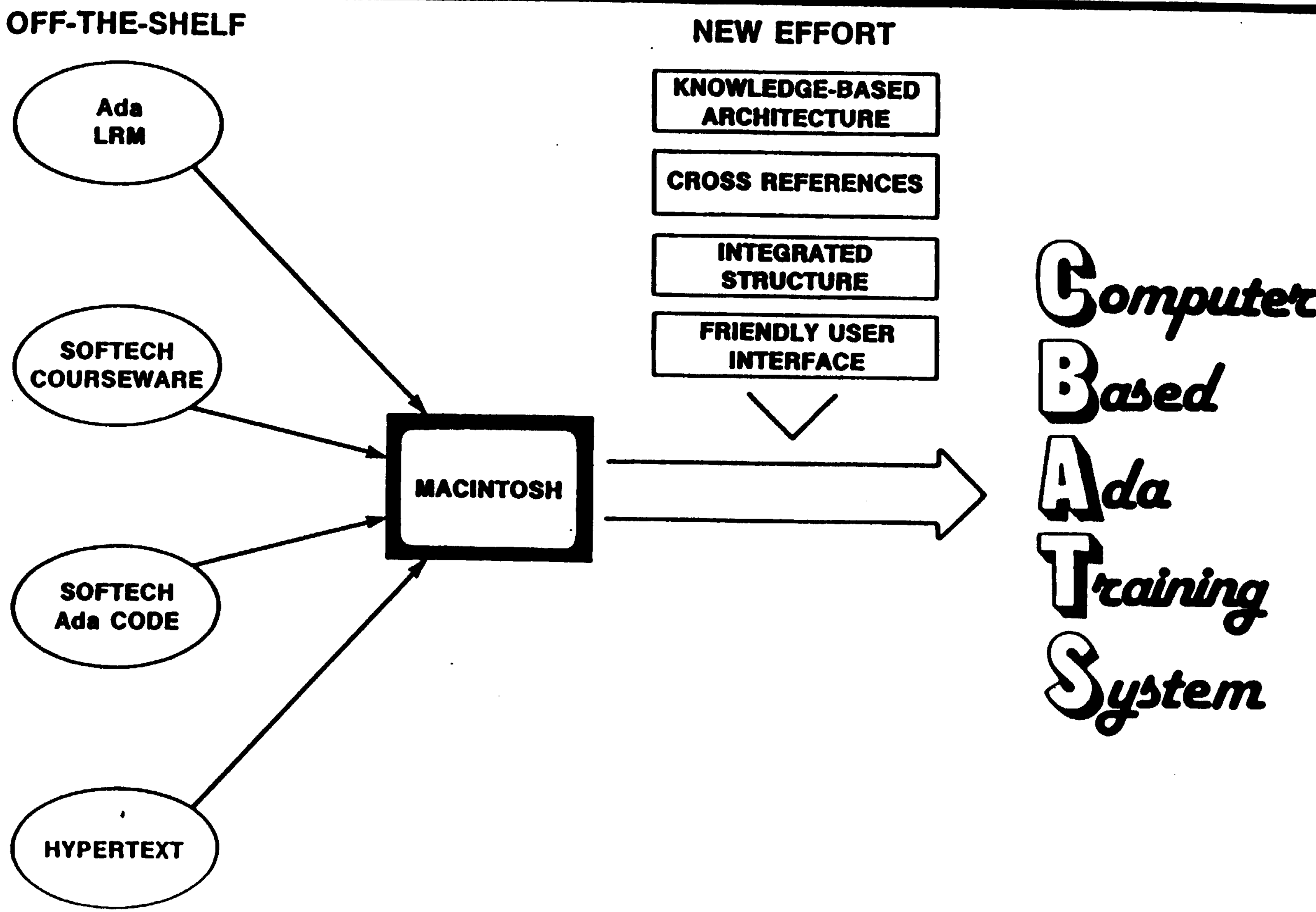
PRODUCT

PUBLICATION

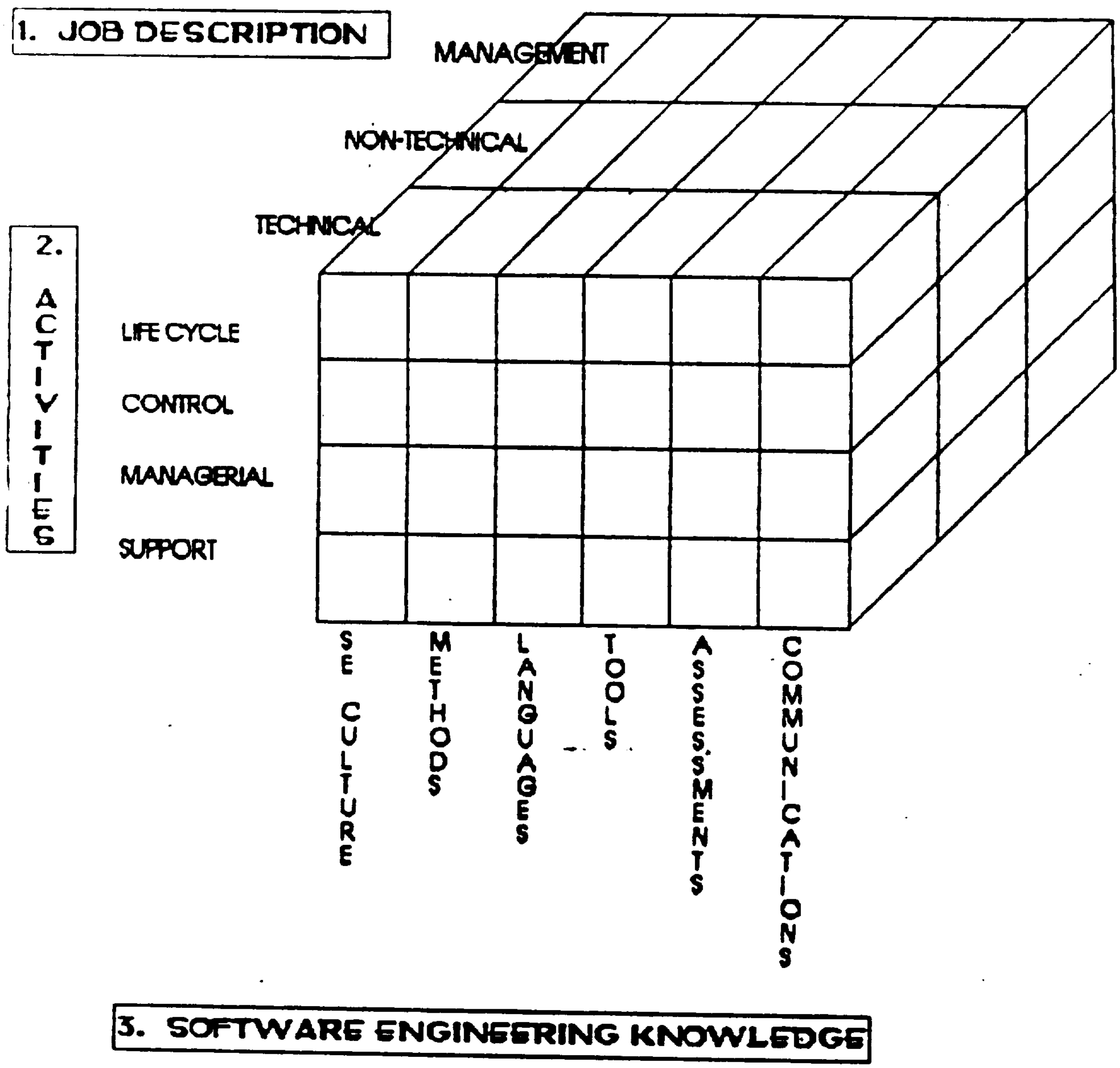
PERSON

BOOK

SOFTECH APPROACH



SOFTWARE ENGINEERING WITH ADA:
A DEFINITION OF THE FIELD
WITH CURRICULAR OPTIONS

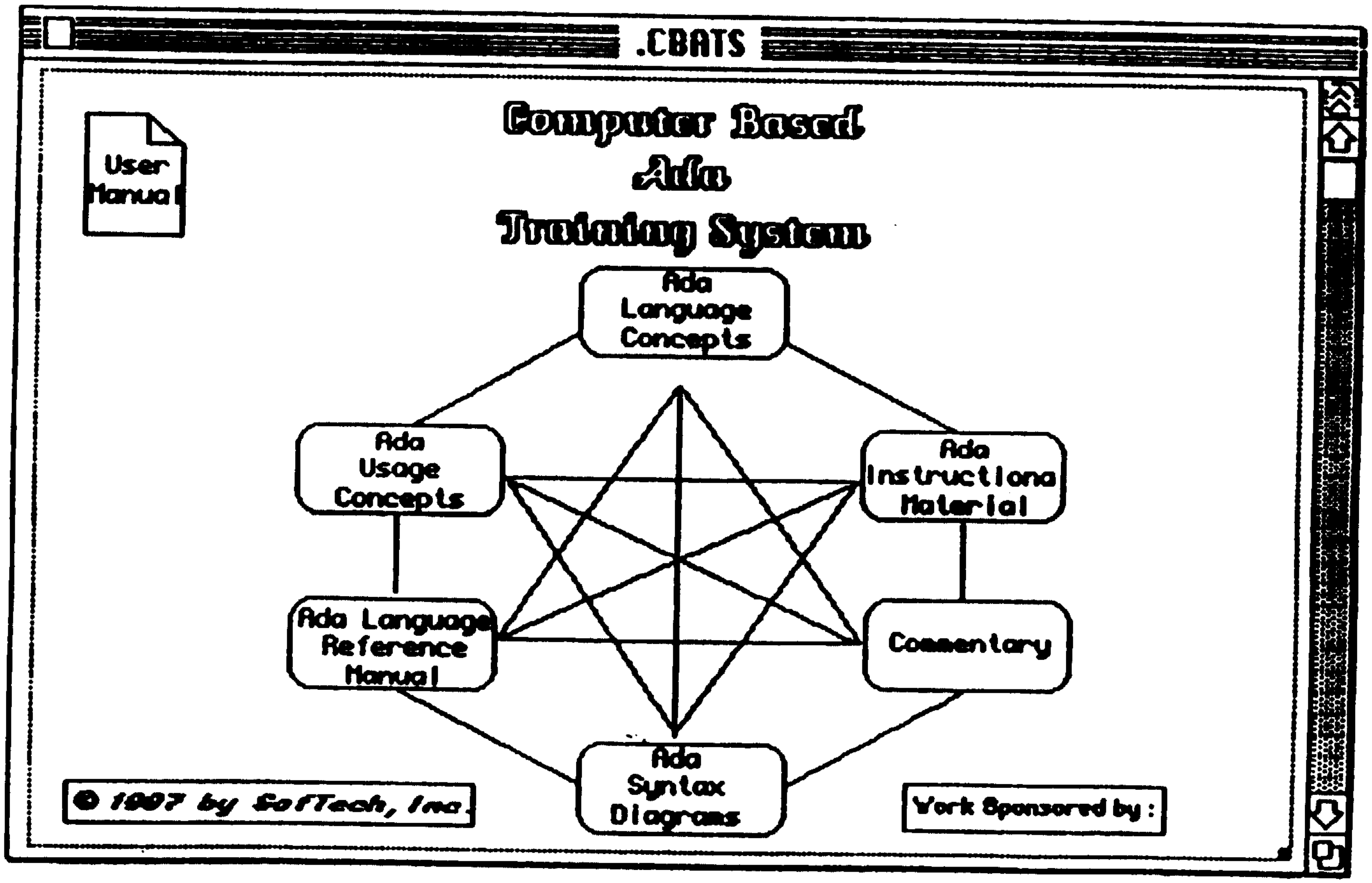


4. ENVIRONMENT: HOST, TARGET, INTEGRATION
5. SKILL LEVEL: INTRODUCTORY, INTERMEDIATE, ADVANCED
6. PROJECT SIZE/COMPLEXITY: SMALL, LARGE COMPONENT, AI-BASED

8710

FROM: CBATS LOGO

"CLICK" ON: ADA LANGUAGE CONCEPTS



SOFTECH

sepec

October, 1987

VIDEOTAPE:

INTRODUCTION TO SOFTWARE ENGINEERING FOR LARGE SYSTEMS

SECTION I
THE COST OF SOFTWARE

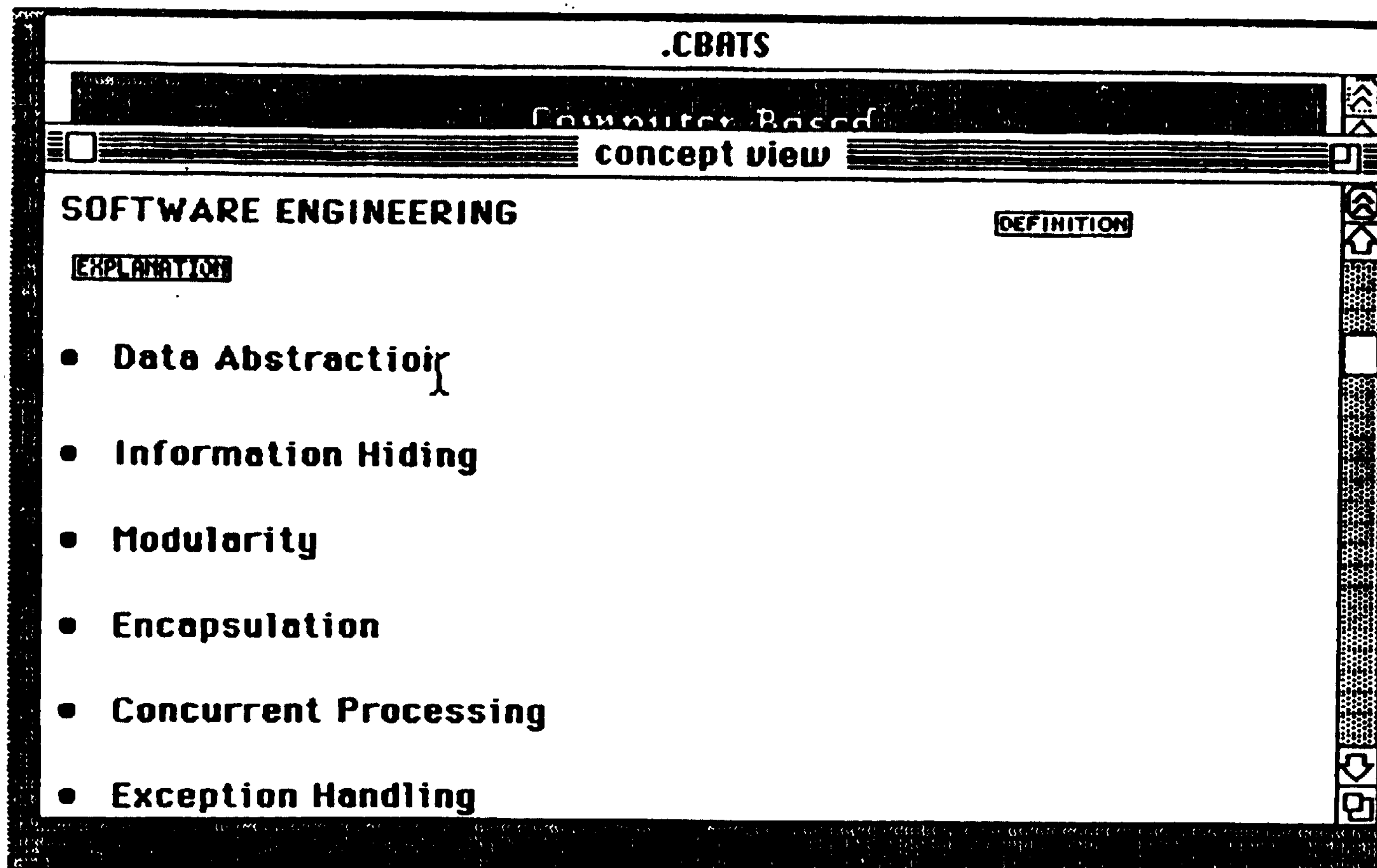
SECTION II
SOFTWARE IN THE SPACE STATION ERA

SECTION III
ENGINEERING SOFTWARE

SECTION IV
BUILDING A SOFTWARE ENGINEERING ENVIRONMENT

FROM: SOFTWARE ENGINEERING

"CLICK" ON: DATA ABSTRACTION



E.T. 1 and E.T. 2
SOFTWARE ENGINEERING AND Ada TRAINING

DELIVERABLES (EXCERPTS)

MARKET SURVEYS

AWARENESS FOR UPPER-LEVEL MANAGEMENT

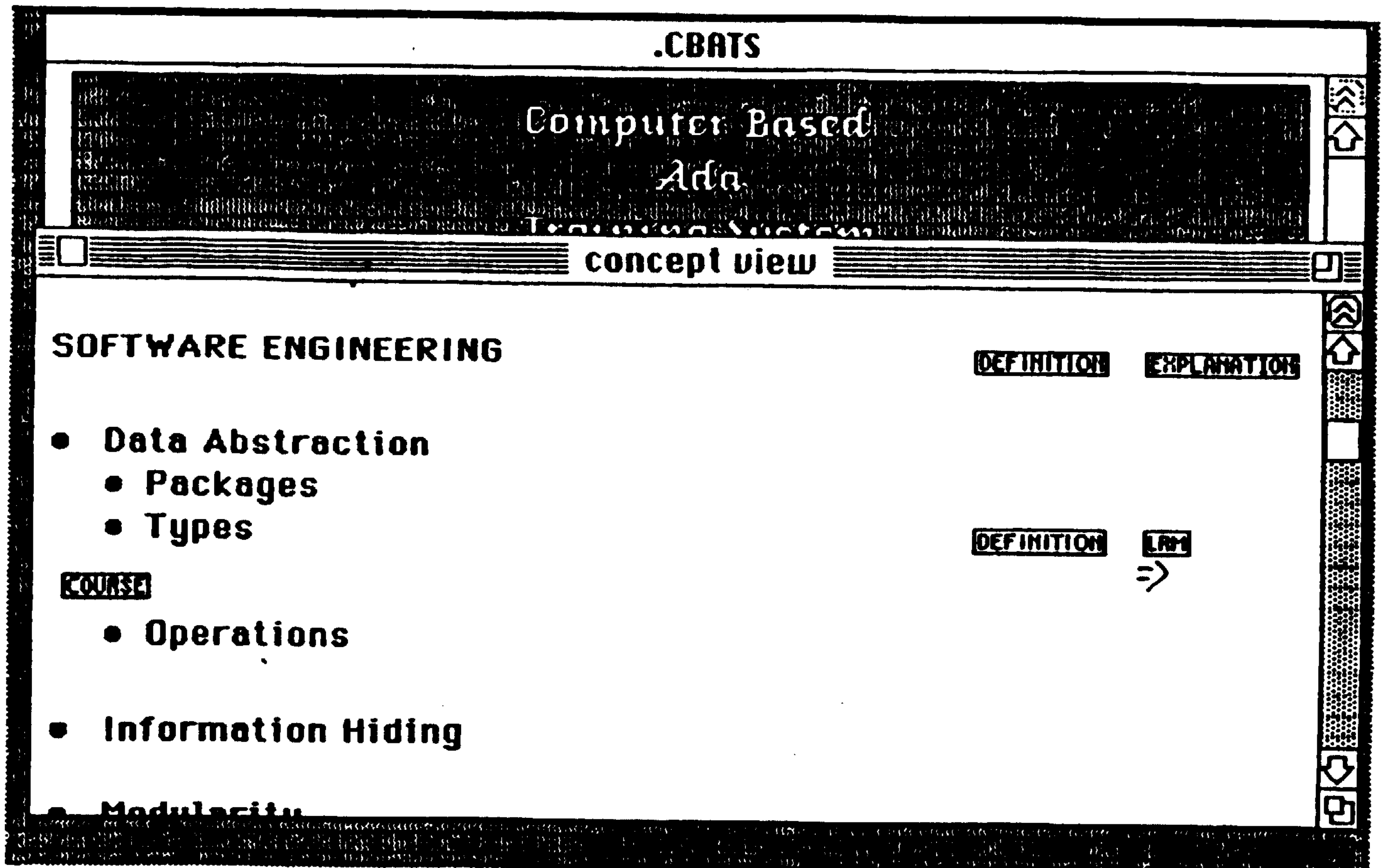
TRAINING MODELS AND CURRICULAR GUIDELINES

TRAINING INFRASTRUCTURE AND NETWORKS

TRAINING RESOURCES REPOSITORY

FROM: TYPES

"CLICK" ON: L R M



SOFTECH

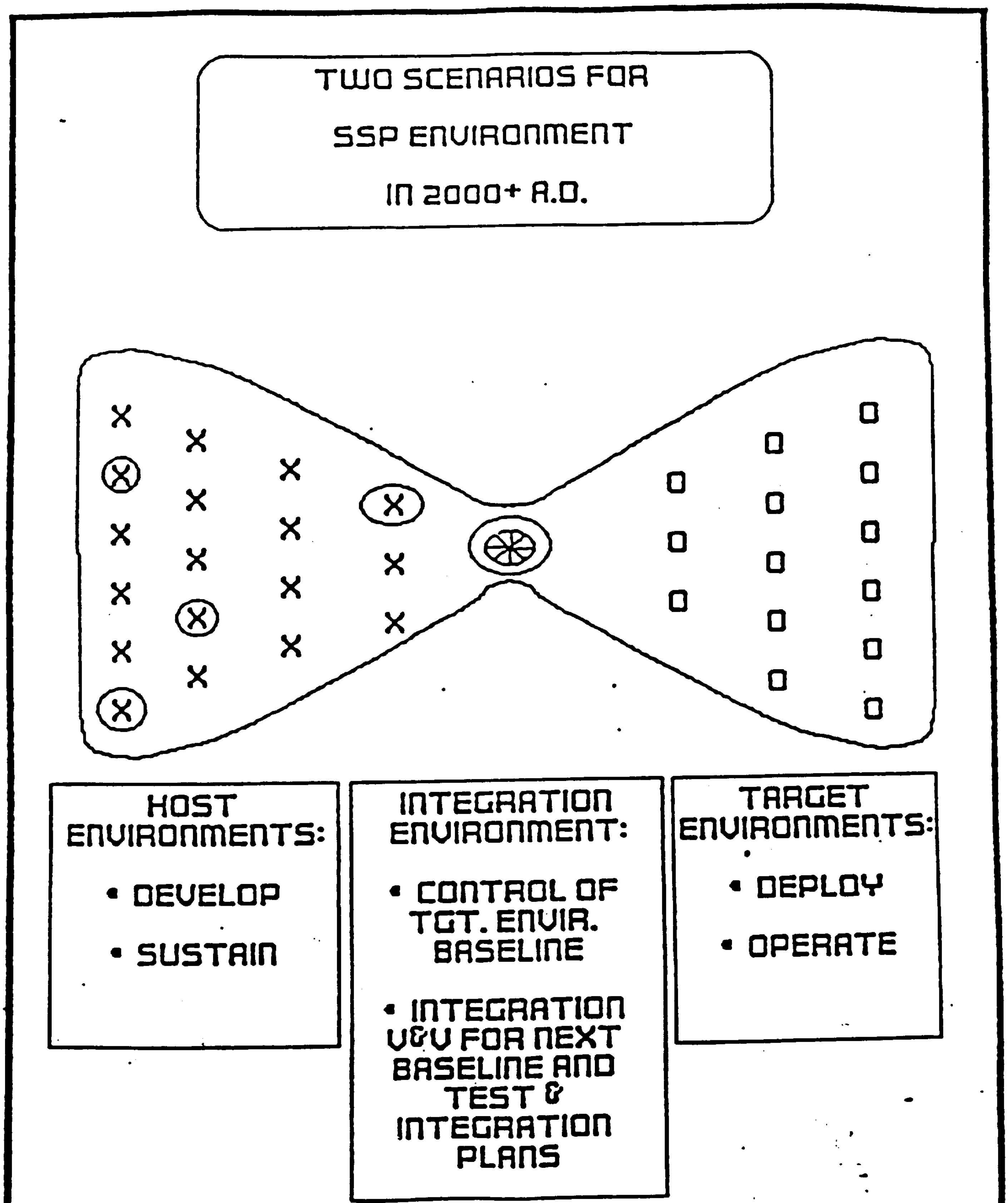


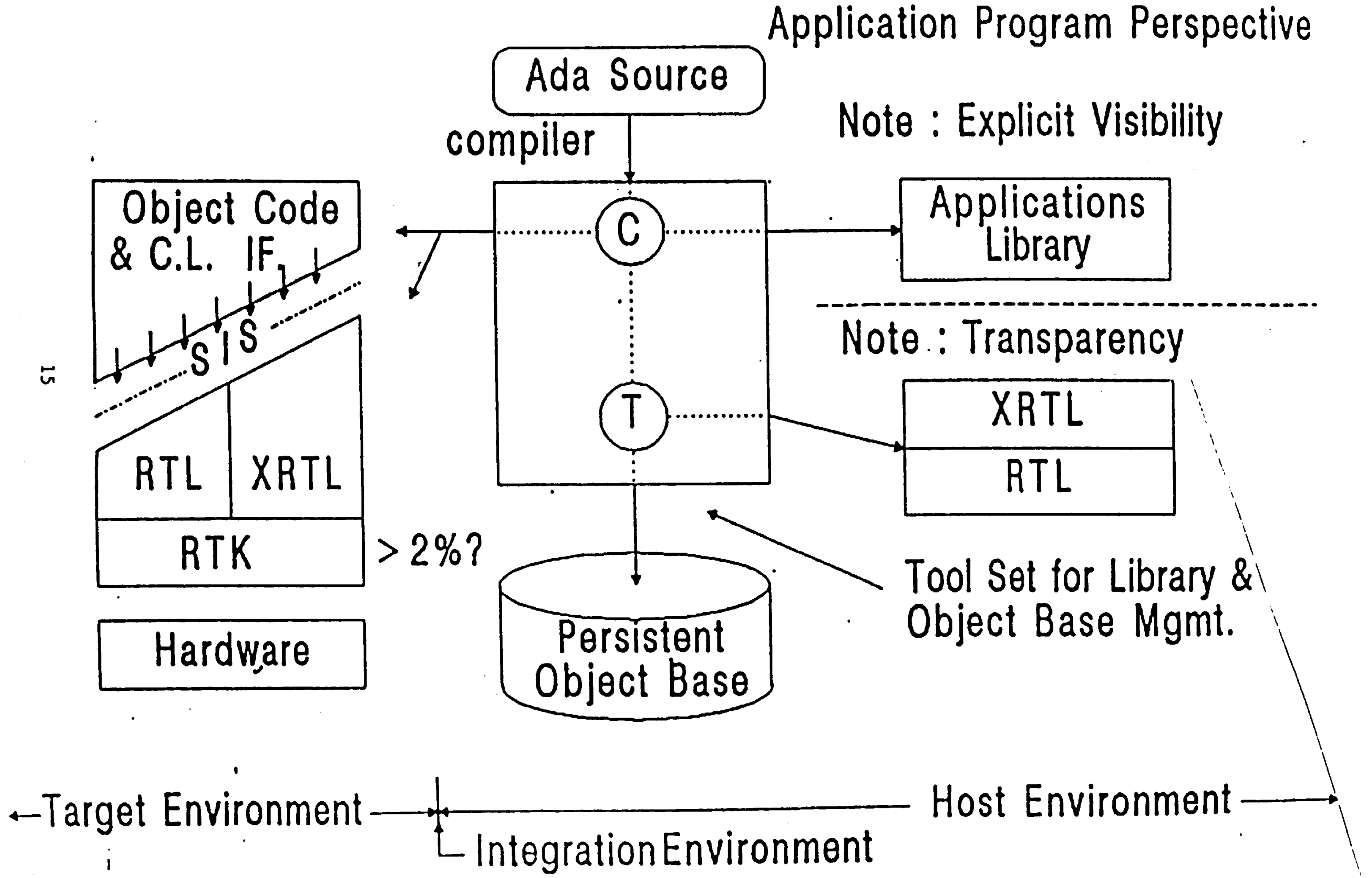
FIGURE 1-5 THREE SOFTWARE ENVIRONMENTS

Bare Machine philosophy ensures integrity

The policies, procedures and standards which apply to the deployed executable code should also apply to the compiler that produces the code, the runtime library modules that provide services required by the code, and the entire execution environment that supports the code

FIGURE 1-6

A Model for Supporting a 'Bare Machine' Philosophy for 'Safety Kernels' of Ada Runtime Support Environments (Ada RTSE's)



The Portable Common Execution Environment (PCEE) consists of

SPACE STATION

A standard set of interfaces

A common object representation

A supporting conceptual archiecture

Stable Framework Components

SPACE STATION

Closed sets of SIS's with their own unique attributes to identify them

Models of enclosed collections/instances of objects and object sets (which include firewalled protection capability) which can be strongly typed

Model of conceptual architecture of overall stable framework for a particular environment

The SIS consists of

SPACE STATION

Policies for management of services and resources to be provided to the application programs

Management modules to enforce the policies

Set of rules for modifications and extension

There are two environment perspectives

SPACE STATION

- Static - encompasses all host environment lifecycle phases except deployment and operation

CAIS predominantly satisfies this viewpoint

- Dynamic - encompasses all of the actions and requirements during the execution of programs

Extensions are required for this viewpoint

Feature	PCEE	CAIS	CAIS-A	PCTE
Status	Definition Stage	MIL-STD	In Progress	Completed
Validation Suite	Required	In Progress	In Progress	Completed (based on XPC)
Basis	ARTEWG CIFO and Clear Lake Model	unique	CAIS	UNIX SVID
Representation	Object	Node	Node	Object
Information Management	Extensible EA/RA (based on IRDS)	unique EA/RA	unique EA/RA	unique EA/RA
Kernel	"bare machine", operating system	"bare machine", operating system	"bare machine", operating system	operating system
Security	Full TCS "Puce Book"	minimal	TCS B3 class MLS	minimal
Cooperating Environments	required	not supported	supported	LAN only
Location	Migratable	Fixed	Migratable	LAN migratable
Processor Types	Heterogeneous	Homogeneous	Heterogeneous	Homogeneous
Common External Data Format	required	not supported	some support	minimal support
Communications Implementation	Full OSI	NA	TBD	Four layers of OSI
Distribution	RANs of Integrated LANs	single site	Some RAN and LAN support	LAN
Unique Names	Objects, Processes, Transactions, Relationships, and Attributes	Nodes, Relationships, and Attributes	Nodes, Relationships, and Attirbutes	Objects, Relationships, and Attributes
Transaction Management	Distributed Nested	NA	Single Level	Distributed Nested
Data Access	Synchronized	NA	NA	Synchronized
Stable Storage	required	NA	NA	NA
Granularity of Representation	Each thread of control for each program	program	program	program
Interoperability	data, tools, control	data	data	data
Goals	portability, performance stable baseline & safety across all environments	portability, performance	portability, performance	portability, performance, stable base
Support for Multiprocessors	required	NA	NA	NA
Support for nonfunctional requirements	ARTEWG CIFO and Clear Lake Model	NA	NA	NA
Environments	Host,Target and Integration	Host	Host	Host (some Target)
I/O	graphics, windows and other devices	character-oriented terminals	graphics and windows	graphics and windows

Figure 3-1 Comparison of Features for a PCEE

There are three SIS service perspectives

SPACE STATION

Tool Writer - what is needed beyond the syntax and semantics of Ada in the area of process control and external interactions

Project Information Manager - what controls should be built in the system to ensure proper information management

System Administrator - interaction between systems as well as coordination of resources, and management of distribution

Acronym List:

SPACE STATION

APSE - Ada Program Support Environment
ARTEWG - Ada Runtime Environment Working Group
CAIS - Common APSE Interface Set
CIFO - Catalog of Interface Features and Options
ISO - International Standards Organization
LAN - Local Area Network
O/S - Operating system
OSI - Open Systems Interconnection
PCEE - Portable Common Execution Environment
PCTE - Portable Common Tool Environment
UIS - User Interface Set
SIS - Stable Interface Set
WAN - Wide Area Network

PCEE Recommendations

SPACE STATION

Adopt CAIS as an extensible subset of the SIS and the UIS for the host and integration environments

Incorporate lessons learned along with some of the functionality of PCTE

Adopt ARTEWG CIFO as a start toward a runtime environment interface

Support ISO/OSI standard for WAN's of LAN's, especially in the context of unreliable communications

Utilize IRDS as the EA/RA modeling technique

Describe the PCEE in conceptual layers and conceptual contexts

Establish test bed support for a PCEE to investigate issues of safety, mission requirements, extensibility and adaptability

Assign responsibility to an appropriate group to effect the integration of the SIS and the UIS across all three environments

Ada Programming Support Environment (MAPSE) to Support the Life Cycle of Large, Complex, Non-Stop, Distributed Systems, SERC, July 1986.

Military Standard Common APSE Interface Set (CAIS), MIL-STD-1838 31 January 1985.

Space Station Software Support Environment Functional Requirements Specification, National Aeronautics and Space Administration, Johnson Space Center, JSC 30500, Draft 3.0, (6 April 1987).

Notkin, D. et. al. Heterogeneous Computing Environments: Report on the ACM SIGOPS Workshop on Accommodating Heterogeneity.

PCTE A Basis for a Portable Common Tool Environment, Project Report, ESPRIT Technical Week 86.

PCTE A Basis for a Portable Common Tool Environment Ada Functional Specification, First Edition, Volume 1.

Rogers, P. and C. McKay. "Distributed Program Entities in Ada", Proceedings of the First International Conference on Ada Programming Language, 2-5 June 1986, p B.3.4.1.

Rogers, K. "Extending the Granularity of Representation and Control for CAIS Process Nodes", Proceedings of the First International Conference on Ada Programming Language, 2-5 June 1986, p D.2.3.1.

Thall, R. and S. LeGrand. "The CAIS 2 Project", Proceedings of the First International Conference on Ada Programming Language Applications for the Space Station Program, 2-5 June 1986, p D.2.6.1.

Bibliography

ACM SIGAda Ada RunTime Environment Working Group (ARTEWG), A Catalog of Interface Features and Options for the Ada Runtime Environment, ARTEWG Interfaces Subgroup3, Release 2, 23 July 1986.

Auty, David, Ada and Operating Systems Practice and Experience in Targeting Ada, presentation to NASA/JSC, April 1986.

CAIS Panel Discussion, First International Conference on Ada Programming Language, 2-5 June 1986, Session D.5.1.

Chen, C. "Conceptual Architecture for an Ada Run-Time Environment", Rockwell SSSD IR&D Progress Report 86567, (fall 1987).

Dolk, R.D. and R. A. Krisch II. "A Relational Information Resource Dictionary System", Communications of the ACM, Vol. 30, No.1 (January 1987).

Fisher, Herman, PCTE Overview and CAIS Comparison Impressions, 9 September 1985.

Fisher, Herman, PCTE Ada Conceptual Design (PCD), Mark V Business Systems, Draft of 22 November 1986.

KAPSE Interface Team (KIT), DoD Requirements and Design Criteria for the Common APSE Interface Set (CAIS), 13 September 1985.

KIT Meeting, presentation by CAIS-A contractor, April 1987.

Mark V Business Systems/Systems Designers PLC., PCTE Ada Interface Requirements, Version 1.1, 27 December 1986.

McKay, C. "Distributed Computer Systems and Software Safety", SERC Lecture Notes, April-June 1987.

McKay, C. A Proposed Framework for the Tools and Rules to Support the Life Cycle of the Space Station Program, COMPASS '87 Conference Proceedings, IEEE, June 1987.

McKay, C. Life Cycle Support For "Computer Systems and Software Safety" in the Target And Integration Environments of the Space Station Program, SERC Memo, 15 June 1987.

McKay, C. "CWM's Perspective of:

- Probable enhancements to transition CAIS to CAIS-A
- Potential implications for 3 environments of the Space Station Program (host, target, integration)", SERC

Memo, May 5, 1987.

McKay, C., R. Charette, D. Auty Final Report on: A Study to Identify Tools Needed to Extend the Minimal Toolset of the

Clear Lake Model for Ada* RTSE Prototype

RICIS project
SE.6

Charlie Randall
October 15, 1987

* Trademark of the Ada Joint Programming Office

GHG

Researchers

UH-CL/Software Engineering Research Center

Dr. Charles McKay

Pat Rogers

GHG Corporation

Charlie Randall

GHG

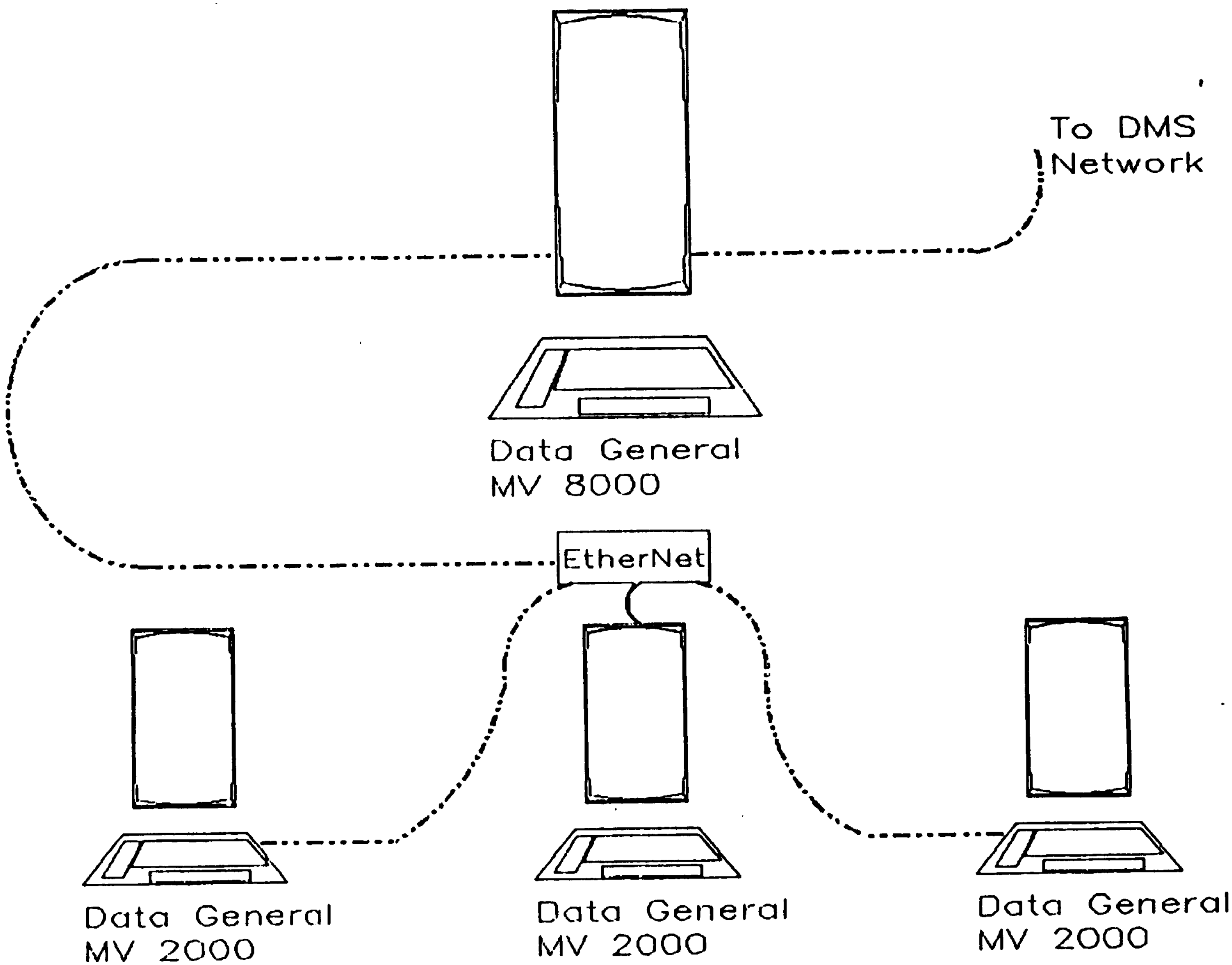
Goals

Develop proof-of-concept prototype of the Clear Lake Model for Ada Run Time Environments

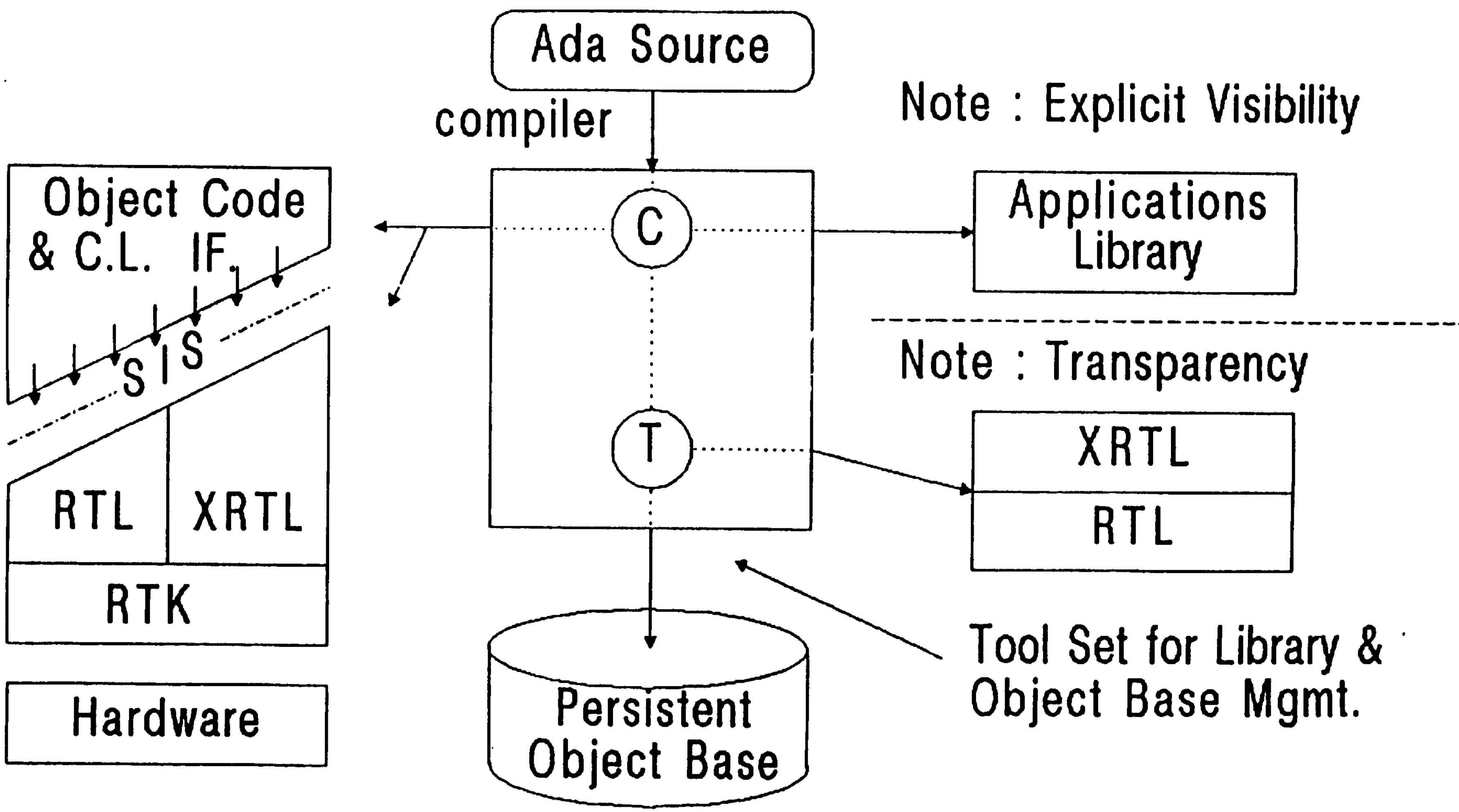
Investigate distributed Ada programming

Explore monitoring/instrumentation issuses and compare to method used by the Shuttle project (FEID)

GHG



Distributed Ada Testbed


Ada Source
compiler
Note : Explicit Visibility
Object Code
& C.L. IF.
S I S
C
Applications Library
Note : Transparency
T
XRTL
RTL
RTL
XRTL
RTK
Hardware
Persistent Object Base
Tool Set for Library & Object Base Mgmt.
GHG

Technical Background

- Runtime Kernel (RTK)
- Runtime Library (RTL)
- Extended Runtime Library (XRTL)
- Program Environments
  - Host Environment
  - Integration Environment
  - Target Environment

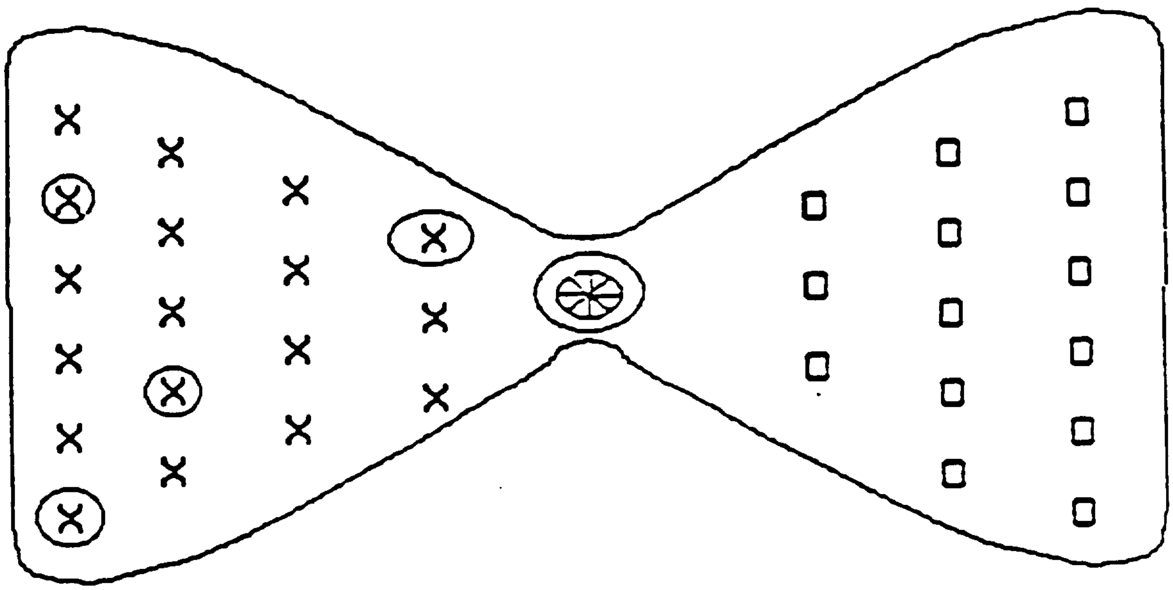
GHG

Clear Lake Model

- Distribute Ada entities
- Interface to virtual Ada machine
- Provide functionality via XRTL
- Transparent reconfiguration
- Application directed reconfiguration
- Surrogates and agent tasks

GHG

TWO SCENARIOS FOR SSP ENVIRONMENT IN 2000+ A.D.



HOST ENVIRONMENTS:

- DEVELOP
- SUSTAIN

INTEGRATION ENVIRONMENT:

- CONTROL OF TGT. ENVIR. BASELINE
- INTEGRATION V&V FOR NEXT BASELINE AND TEST & INTEGRATION PLANS

TARGET ENVIRONMENTS:

- DEPLOY
- OPERATE

Surrogates

Independent "processes" within each runtime system that reside on each of the distributed processors in the distributed system.

Justification for distributing Ada entities

Reliability

Hardware cost

Extensibility

Resource sharing

Fidelity

Performance efficiency issue

GHG

Status

Developed communications demo

Designing and developing prototype and related tools

Investigating the use of:

- DG runtime system software
- DIANA
- Command Language Interface/ System Interface Set/ User Interface Set

OUTLINE

- Objective
- History
- Components
- Architecture
- Features
- Methodology
- Feature Enhancements
- Research Problems
- Conclusions

SOFTWARE ENGINEERING AND ADA DATABASE (SEAD)

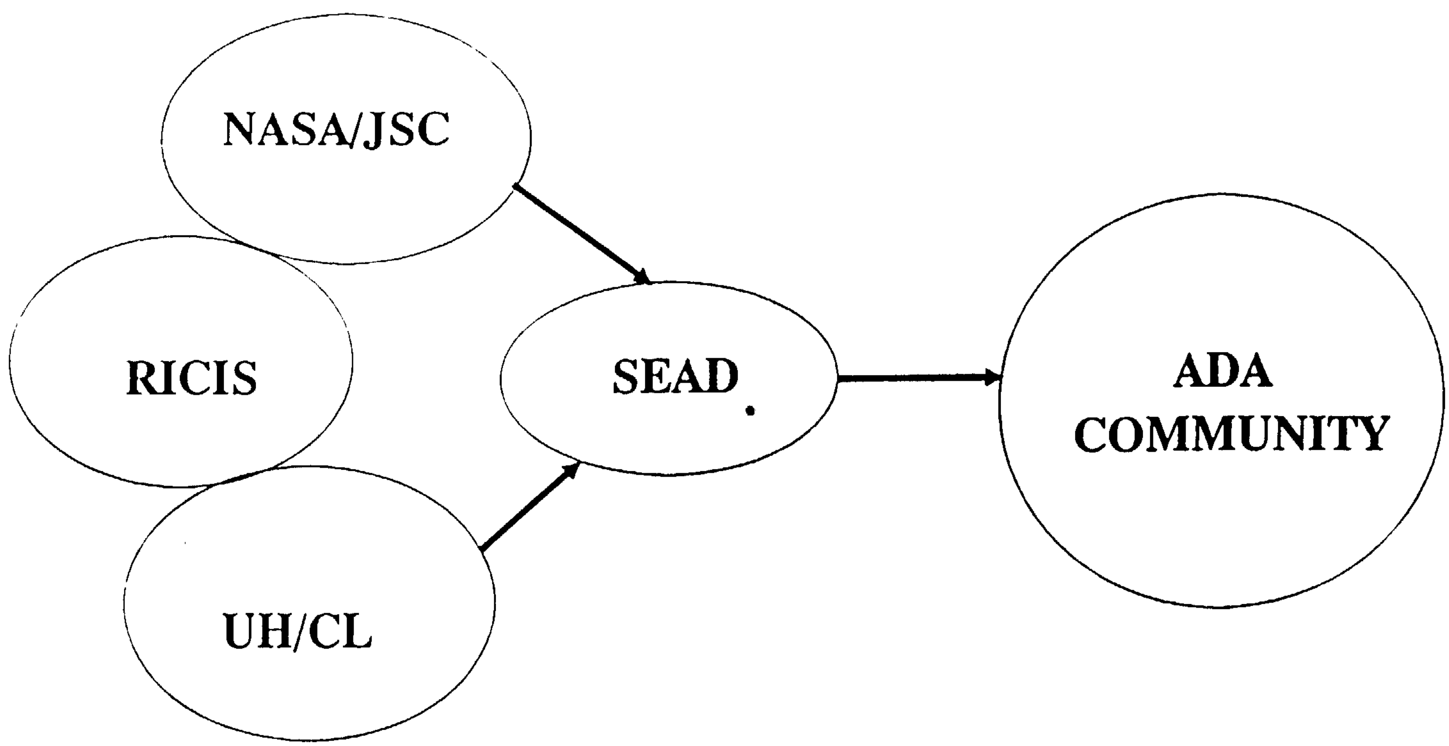
October 1987

Morris Liaw

HISTORY

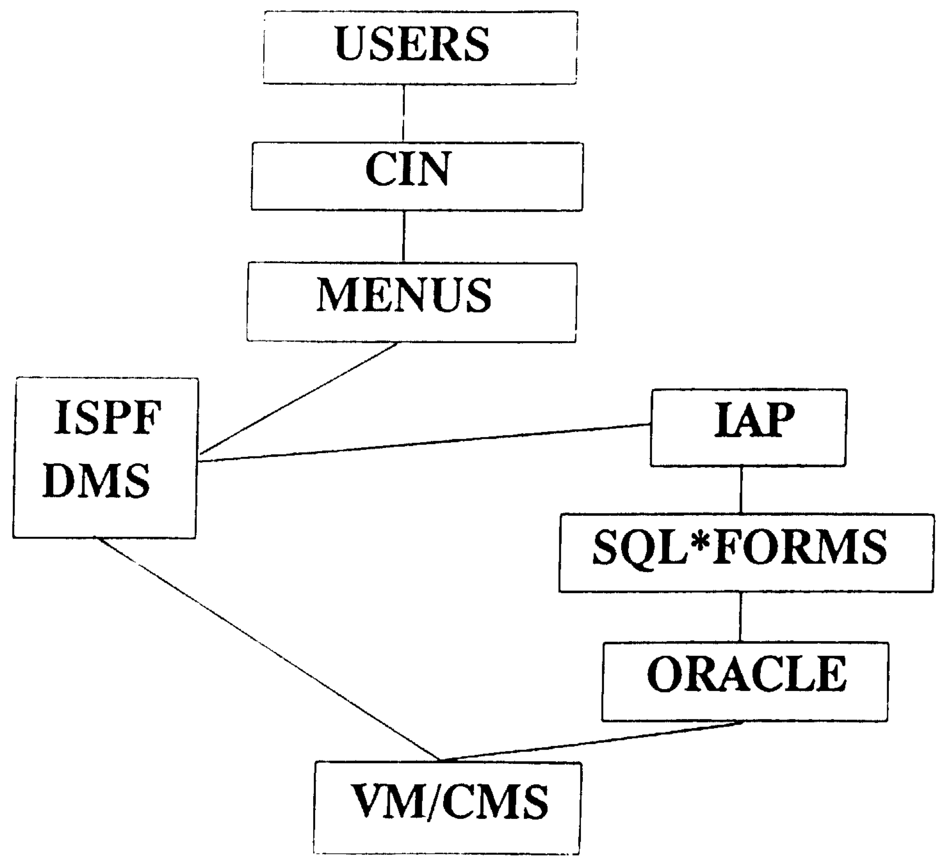
June 1986	- Project Began
October 1986	- Prototype Demostration on VAX at UH/CL
January 1987	- Computer Transfer of SEAD from UH/CL to JSC IBM 4381
February 1987	- Alpha Testing on IBM 4381
April 1987	- System Finalized and Operational
July 1987	- Completed Transfer of SEAD from IBM 4381 to CIS-C
August 1987	- More Data Gathered and Loaded
September 1987	- Modifications for Second Release Approved

OBJECTIVE

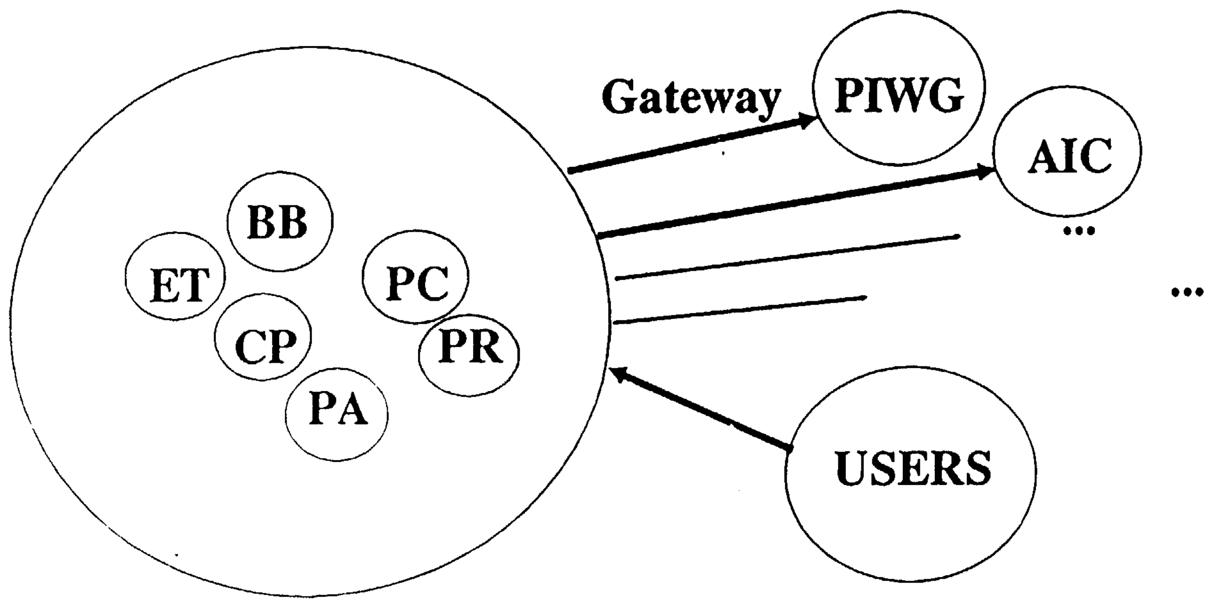


SEAD - Software Engineering and Ada Database

ARCHITECTURE



COMPONENTS



BB - Bulletin Board
ET - Education & Training Resources
PC - Publications & Conferences
CP - Compilers & Products
PR - Projects
PA - Reusable Packages

METHODOLOGY

DATABASE

- Top Down Approach
- Computer Aided Design
- DK/NF

APPLICATION

- Menu Driven
- Form
- Macro
- Trigger
- User Exit

FEATURES

EASINESS

- Menus and Forms
  - Menu Driven
  - Single Screen Per Menu
- On-Line HELP Facilities
  - System Level
    - Beginner Information
    - Function Key Map
  - Menu Level
    - Help Window
    - Procedure for Browsing
    - Function Key Map

ACCESSIBILITY

- Fully Accessible
- History File

INTEGRITY

- Key Constraints
- Domain Constraints
- Data Consistency
  - Trigger

SECURITY

- System Level
  - Public Account
  - Individual Account
- Menu and Field Level

RESEARCH PROBLEMS

- New classification scheme for reusable packages
- Text processing for ORACLE
- Designer's tools for SQL*FORMS

FUTURE ENHANCEMENTS

- A BULLETIN BOARD for information about meetings and announcements
- ON-LINE FORMS for user data contribution
- LOCATION for books and publications
- COMPILER as a separate category
- ACVC VERSION NUMBER, DATA FIELD and PERFORMANCE information for compilers
- ELECTRONIC MAIL for person information
- CROSS-REFERENCE for projects, publications, compilers, products, and reusable packages, etc.
- CONTACT PERSON DATA BASE for DBA to use

MATHEMATICAL AND STATISTICAL ANALYSIS

Conveners: **Cecil R. Hallum,** UH-Clear Lake

David K. Geller, Mission Planning and Analysis Division, NASA/JSC

Space Station Momentum Management and Attitude Control

Bong Wie, University of Texas at Austin

Quantifying Software Reliability (Invited Presentation)

Patrick L. Odell, University of Texas at Dallas

CONCLUSIONS

- We have modelled the Software Engineering and Ada Database, and also delivered the product.

- Second release will be available in January 1988.

- The project is beneficial to JSC, UH/CL and Ada Community.

Summary of the Mathematical and Statistical Analysis Technical Session

David Geller
and
Cecil Hallum

PRECEDING PAGE BLANK NOT FILMED

MATHEMATICAL AND STATISTICAL ANALYSIS

COMPONENT CHARTER

SUMMARY OF PRIORITY SUPPORT AREAS

CURRENT ACTIVITIES/STATUS

RICIS SYMPOSIUM 87'

Mathematical and Statistical Analyses

Space Station Attitude Control and Momentum Management
Presented by Dr. Bong Wie

Summary

The space station momentum management project is being monitored by the MPAD Guidance and Navigation Branch. The primary investigator, Dr. Bong Wie of the University of Texas at Austin, presented a preliminary design for station atttiude control and momentum managment.

First, Dr. Wie reviewed the space station Phase 1 configuration and described a station CMG (control momentum gyro). He then explained how the CMG's control the attitude of the station and why CMG momentum management is needed.

Next, a controller for the space station pitch axis was introduced. It was shown that the cyclic aerodynamic torques acting on the station produced large undesirable pitch oscillations. However, when a periodic disturbance rejection filter (tuned to the aerodynamic torque frequency) was added to the controller, the pitch oscillations were be completely eliminated. In addition, pitch CMG momentum was shown to be minimized.

Following this, the space station yaw/roll controller was introduced. In this case the aerodynamic torques acting about the station's yaw/roll axis produced large yaw/roll oscillations. When the disturbance rejection filter was added to the controller, it was found that only yaw oscillations could be eliminated. Roll oscillations were minimized, but not eliminated.

Dr. Wie also made some comments related to space station flight software. The control laws that would be implemented in the flight software were pointed out, and the required state integrators were noted. It was also shown that the computer storage requirement for gain scheduling could be minimized using a proposed decoupled feedback controller.

It was noted that Dr. Wie and the University of Texas have been extremely productive and have been providing excellent results.

MATHEMATICAL AND STATISTICAL ANALYSIS

(PROPOSED) PRIORITY SUPPORT AREAS

RELIABILITY ASSESSMENT OF SOFTWARE, FAULT-TOLERANT COMPUTERS, AND COMPUTER NETWORKS

Background

- Much research has been largely probabilistic in nature
- Much has been undertaken by the Engineering community
- Numerous publications in Engineering literature
- Interface between Computer Science and Statistics
  - Statistical community has largely focused on the aspect of application
  - Statistical emphasis shifting now to investigation

SOFTWARE RELIABILITY – STATISTICIANS ARE WELL-QUALIFIED TO ADDRESS:

- Quantification and measurement of software reliability
- Assessment of changes in software reliability over time, (reliability growth)
- Analysis of software-failure data
- Decision logic for whether to continue or stop testing software

MATHEMATICAL AND STATISTICAL ANALYSIS

RELIABILITY OF FAULT-TOLERANT COMPUTERS AND SOFTWARE

Complex system involving automatic detection, diagnosis, and correction of errors (faults)

Large research gap

Existing material mostly qualitative

Potential for valuable contributions from Statistical community here

NETWORK RELIABILITY METHODS - CONSIDERABLE RESEARCH EMPHASIS TO-DATE

OTHER CRITICALLY RELEVANT EXPERTISE AREAS INCLUDE:

Math Modeling of Physical Systems

Simulation

Statistical Data Reduction

MATHEMATICAL AND STATISTICAL ANALYSIS

Evaluation Methods

- Robustness (Stability)
- Sensitivity Analysis
- Perturbation Theory
- Error Analysis
- Development of Test Criteria

Optimization

Optimal Experimental Designs

Algorithm Development

Math Methods in Signal Processing

Consultant and Team Member

CURRENT ACTIVITIES/STATUS

SPACE STATION MOMENTUM MANAGEMENT AND ATTITUDE CONTROL

Bong Wie, J. Speyer, and D. Hull
Guidance and Control Group
Dept. of Aerospace Engineering and Engineering Mec
UT/Austin

QUANTIFYING SOFTWARE RELIABILITY

Professor Patrick L. Odell
Department of Mathematics
UT/Dallas

ON QUANTIFYING SOFTWARE RELIABILITY

Patrick L. Odell
University of Texas at Dallas

October 1987

PERSPECTIVES

Life Cycle.

(1)Requirement Definition Cycle
(2)Design Cycle
(3)Construction Cycle (Includes VVT)
(4)Operation and Maintenance Cycle

The Actors and Advocates.

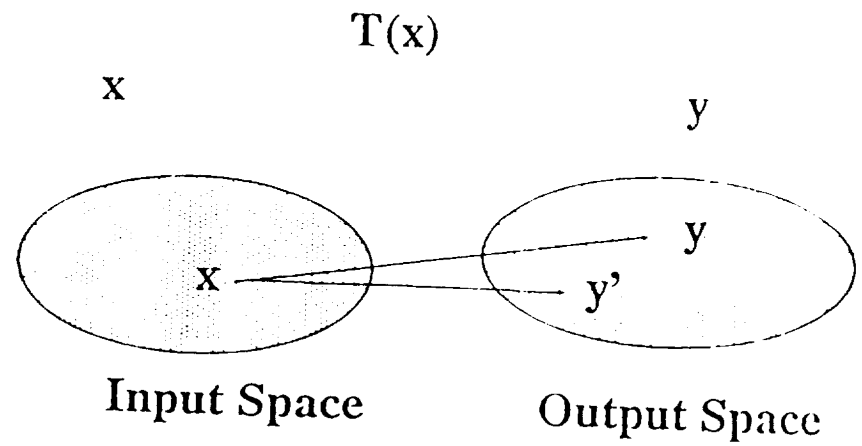
(1)Managers
(2)Coders and Computer Scientists
(3)System Engineers
(4)Reliability Engineers

A Little Culture.

E valuation and Development
E valuation Separate from Development
R eliability Report submitted to manager and then manager makes decision
M anagers and Producers are not part of final evaluation team
Q uality Assurance versus Manufacturing

QUANTIFYING SOFTWARE RELIABILITY

A PROGRAM. $P(x) : X \to Y$ $P(x)$ $T(x)$



<u>The Literature.</u>

HARDWARE RELIABILITY:

1. Shooman M.L., "Probabilistic Reliability: An Engineering Approach," 1968.

2. Martz H.F. and Waller, R.A., "Bayesian Reliability Analysis," 1982.

SOFTWARE RELIABILITY:

3. Musa, J.D., Iannino, A., and Okumoto, K., "Software Reliability; Measurement, Prediction, Appplication," 1987.

4. Miller E. and Howden, E.H., "Tutorial: Software Testing and Validation Techniques," 1981.

5. ACM Computing Surveys 14, No. 2 (1982).

QUANTIFYING SOFTWARE RELIABILITY

HALSTED'S METHOD. *Total number of bits required to specify the program*

$$V = (N_1 + N_2)\log_2(n_1 + n_2)$$

Number of "Mental Lapse"

$$N = V/E$$

E denotes mean number of mental discriminations between lapse.

CAPTURE-RECAPTURE METHOD.

(1) $\hat{N} = N_S \frac{n_{ns}}{n_s}$ *Duran/Duran Wiorkowski*

(2) $\hat{N} = \frac{mn}{k}$ *Rudner*

MEAN TIME TO FAILURE (MTTF).

(1) *Errors* $= M(1 - \exp(-Ct/MT))$ *where M denotes Total No. of Errors, T denotes MTTF at start of test.*

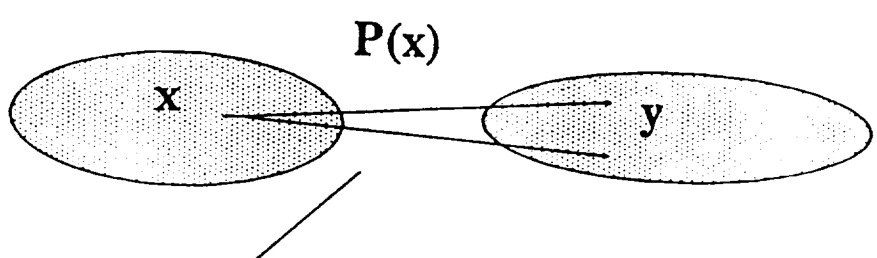
(2) $R = \exp(-t/MTTF)$ *denotes reliability — Musa, Iannino, and Okumoto.*

QUANTIFYING
SOFTWARE RELIABILITY

DIRECT METHOD.

$$R = \frac{\text{No. of successful runs}}{\text{Total No. of runs}}$$

FUNCTIONAL TESTING (Alternative)



$$P_m(x) = P(x) + P_e(x)$$

$P_m(x)$ - mutation

RESEARCH AREAS

Ref..

(1)Software Testing Validation, Section 7

(2)Validation, Verification and Testing of Computer Software (NSF), Vol. 14, No. 2, ACM Computing Surveys.

(3)Chapter 7,; Musa, Iannino, and Okumato

Areas.

(1)Theoretical Foundations for a) Testing b) Evaluatiing Software, and c) Evaluating-Expert Systems.

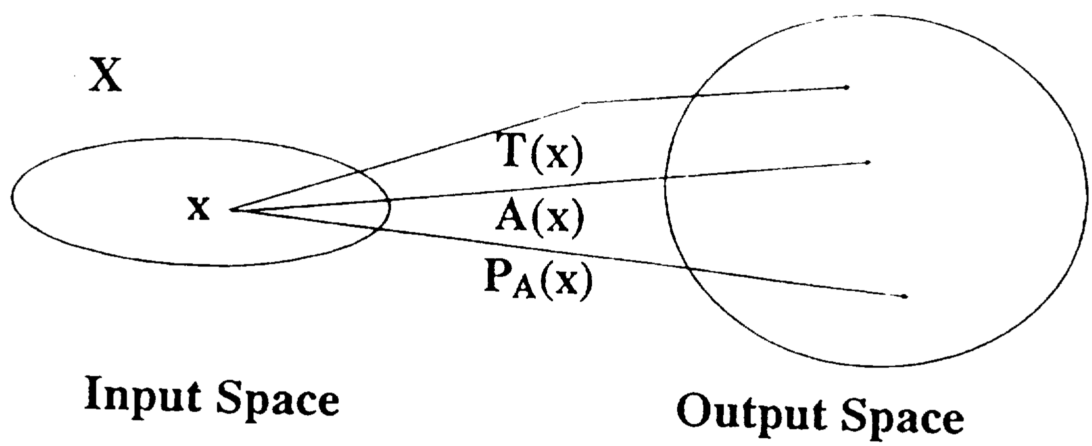
(2)Develop "Pert Chart like" software development system to monitor and/or guide software development.

(3)Make precise the notion of "how valid software should be?"

(4)A method for packaging validation tools.

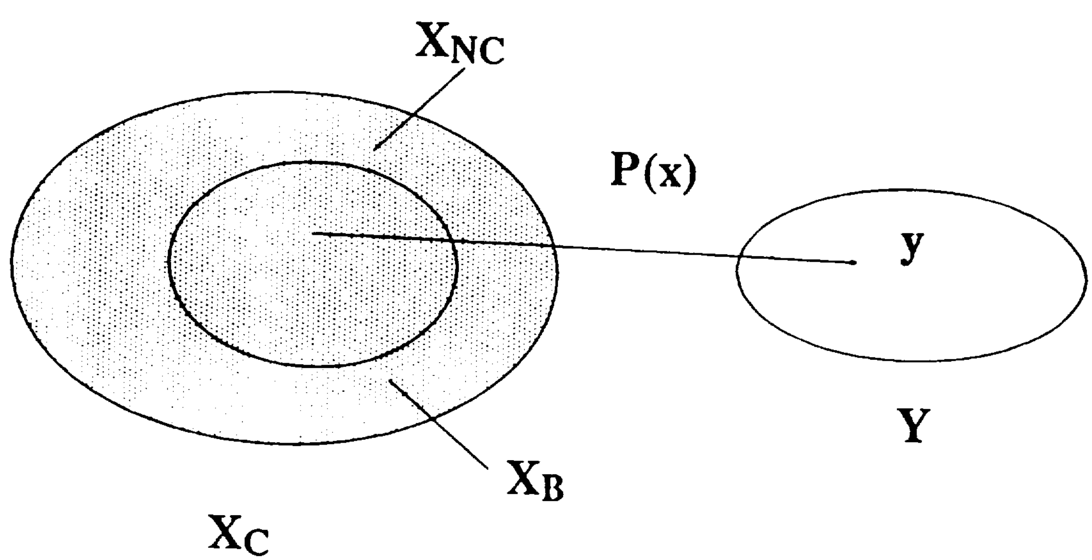
(5)Study interaction effects between software and hardware in order to estimate system reliability.

(6)Develop methods for "tearing" a program apart to facilitate functional testing.



$$P_A(x) \equiv T(x) \quad \text{for all} \quad x \in X$$

$$\| P_A(x) - T(x) \| \leq E(x) \quad \text{for all} \quad x \in X$$



X_B, X_C, X_{NC}

INFORMATION MANAGEMENT

Conveners: **Peter C. Bishop,** UH-Clear Lake

William J. Huffstetler, Assistant to the Director, Engineering, NASA/JSC

The Need for Strategic Information at JSC
William J. Huffstetler, NASA/JSC

Research Projects in Information Management
Peter C. Bishop, UH-Clear Lake

Database Strategies and Prototypes
Timothy N. Tulloch, Vice President, TNT Consulting

Space Station Documentation Technology and Strategies
Christopher Dede, Professor of Education, UH-Clear Lake

Future Research Opportunities
Lloyd R. Erickson, Electronics Engineer, NASA/JSC

INFORMATION MANAGEMENT
Presentations

Peter C. Bishop, PhD
Associate Professor, Human Sciences
Director, Space Business Information Center
University of Houston-Clear Lake

Information management is the RICIS research area devoted to the final customer of computing and information systems--the end-user. They are the people at the end of the long chain of information systems who don't care how their information is collected, manipulated, or stored as long as the right information is in their hands at the right time.

Information productivity, therefore, is the overall objective of the information management research area. In other words, people who use information systems should realize more value by using the system than by not using it. NASA in general and the Johnson Space Center is particular have a tremendous need to understand what makes an information system productive and to develop productive systems for its employees, contractors and customers.

Our first speaker in this session will address this issue directly. He is William Heffstetler, Assistant to the Director of Engineering at the Johnson Space Center. Mr. Huffstetler has served in a number of different capacities during his time in government services. Most recently, he was chief of the JSC Office of Flight Projects Engineering, a JSC organization group which helps academic and industrial customers develop payloads for the Space Shuttle. In that capacity, Mr. Huffstetler served on the NASA Commercialization Task Force which wrote the NASA Policy on Space Commercialization. I have asked Mr. Huffstetler to comment on how information systems can enhance productivity at JSC.

I will take floor following Mr. Huffstetler's remarks and describe the main projects within the RICIS information management area. One project I am personally involved in is the Space Market Model Development Project. This project, conceived almost three years ago, is designed to study the information needs of the business community in their search for and evaluation of space ventures. That project has resulted in a prototype information center at the University, entitled the Space Business Information Center, which is collecting and distributing space information to members of the space industry. I will describe the results of the first phase of research activity.

Another objective of the Space Market Model Project is to develop and evaluate electronic tools for the collection and dissemination of information. Our work is focused on using the NOMAD2 database management system. Mr. Timothy Tulloch, Vice-President of TNT Consulting, is a support contractor on this aspect of the Space Market Model. He will describe the principles of information management that we are trying embed in a tool which we have developed.

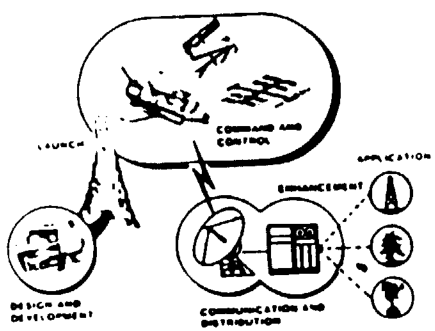
The management of textual data has received less attention than the management of numeric data because the technology to handle large amounts of text did not exist. Dr. Chris Dede will describe how that situation is changing, however. He will report the results of long-term forecasting project he directed. The project was designed to assess the state of the art and the state of the practice in the area of electronic documentation in support of the Space Station software development and management.

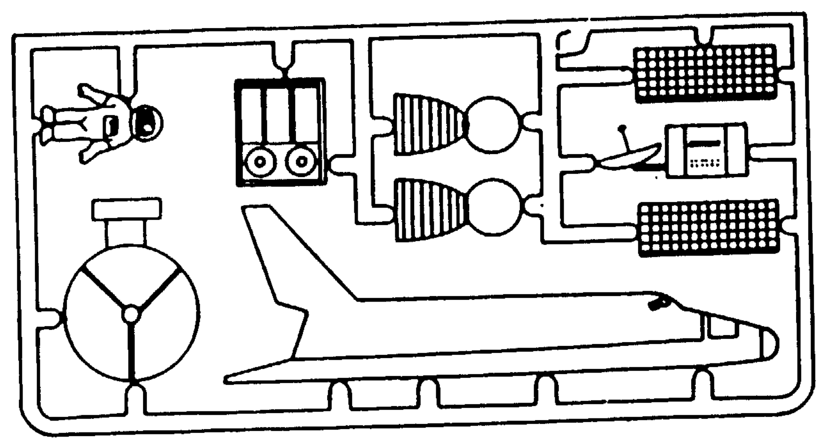
Finally, Mr. Lloyd Erickson from JSC/MPAD will describe a new project we are just beginning. JSC has made great strides in hardware connectivity over the last few years. Individuals can now get access to most mainframe and many of the minicomputers under JSC control. The data which resides on those computers, however, is still largely out of reach through lack of a suitable interface. The Management Information and Decision Support Environment task is designed to prototype interface strategies so that JSC managers and technical staff can use one tool to access a variety of JSC databases.

The sessions this afternoon then contain a blend of the old and the new. Beginning with the need for productive information management, expressed by Mr. Huffstetler, we will get report results of research projects currently underway was well as the objectives of research projects which are only now beginning. True information productivity is a tremendous goal which will require a long-term to achieve. Fortunately, we have begun.

SPACE MARKET MODEL

DEVELOPMENT PROJECT
PHASE I REPORT



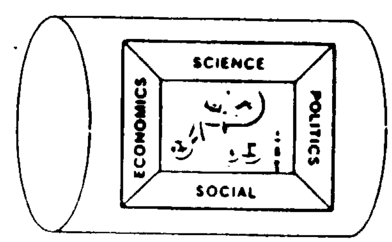


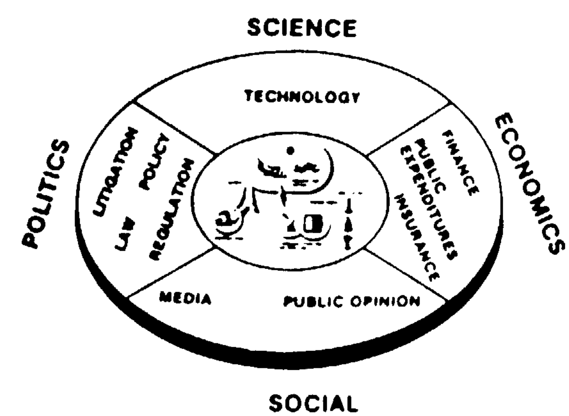
SPACE MARKET MODEL DEVELOPMENT PROJECT
PHASE I
Purpose

- To study the information needs of the space business community
- To study the information available to the space business community
- To design and test a system to deliver useful information to the space business community

BENEFITS OF MARKET INFORMATION

- More Understanding
- Better Decisions
- Less Risk
- More Rapid Market Development





NATIONAL SPACE POLICY

"The Congress declares that the general welfare of the States requires that the National Aeronautics and Space Administration seek and encourage, to the maximum extent possible, the fullest commercial use of space."

-- *Public Law 98-361*

KEY ELEMENTS OF NASA POLICY

- STANDARDIZE BUSINESS ARRANGEMENTS
- PROMOTE SPACE MARKET OPPORTUNITIES
- INSTITUTIONALIZE SPACE COMMERCIALIZATION
- COORDINATE RELATIONS WITH INDUSTRY

SPACE MARKET MODEL DEVELOPMENT PROJECT

Organizations

- **Sponsored by**

  Office of Commercial Programs
  Space Station Customer Utilization
  Johnson Space Center

NASA
National Aeronautics and
Space Administration

- **Conducted by**

  University of Houston-Clear Lake
  Research Institute for Computing
  and Information Systems

UNIVERSITY OF HOUSTON • Founded 1927 •

SPACE MARKET MODEL DEVELOPMENT PROJECT

Milestones

- **Space Shuttle Payload Information System (SSPIS)**

Phase I Start	December 1985
Phase I Report	September 1986
Phase II Start	January 1987

- **Space Market Model Development Project (SMMDP)**

Phase I Start	August 1986
Phase I Report	April 1987
Phase II Start	May 1987
SSPIS and SMMDP Phase II Report	December 1987

SPACE MARKET MODEL DEVELOPMENT PROJECT

Information Available

	COVERAGE
News media	Excellent
Scientific & Technical Literature	Excellent
Organizational Directories	Partial (breadth > depth)
Statistical Directories	Partial (not marketed)
Business Statistics	Partial (govt > private)
Analytical Reports	Partial (expensive)
On-line coverage	News media Tech literature Gov't contracting

SPACE MARKET MODEL DEVELOPMENT PROJECT

Space Information

GOVERNMENT Information	BUSINESS Information
R&D	Commercial
Knowledge	Profit
Present, Future	Past, Present, Future
Benefits	Revenues
Performance	Productivity
Expenditures	Profits
ABUNDANT	SCARCE

SPACE MARKET MODEL DEVELOPMENT PROJECT

Research Approach

Interviews

- 40 respondents
- 5 categories
  - Government
  - Aerospace
  - Entrepreneurs
  - Business service
  - Information

- Purpose
  - information they had
  - information they needed

Acquisitions

- studies
- periodicals
- directories
- databases

SPACE MARKET MODEL DEVELOPMENT PROJECT

General Conclusions

- **Respondents wanted more information about space commercialization**
  - available information is difficult to obtain
  - much information is not available
- **Directory information wanted**
  - who the players are
  - what they do
  - who the contact is
- **Other information wanted**
  - business statistics
  - government activity
  - international activity

SPACE MARKET MODEL DEVELOPMENT PROJECT

Conclusions by Sector

- **Business service**
  - firms involved in a market
  - market statistics
  - rating of a firm's chances
- **Aerospace**
  - government contracting
- **Entrepreneurs**
  - customers
  - technology
  - financing
  - government and aerospace contracting
- **Government**
  - little need in general
  - potential players and market statistics for those in space commercialization

SPACE MARKET MODEL DEVELOPMENT PROJECT

Comparison by Sector

	NEED	INTERNAL CAPABILITY	RESOURCES
Business Service	Hi	Lo	Hi
Aerospace	Hi	Hi	Hi
Entrepreneurs	Hi	Lo	Lo
Government			
• general	Lo	Hi	Hi
• space commercialization	Hi	Lo	Hi

SPACE MARKET MODEL DEVELOPMENT PROJECT

Comparison by Sector
(High Need Sectors)

ABILITY TO GENERATE OWN INFORMATION			
	HIGH	...	Aerospace
	LOW	Entrepreneurs	Business Service Govt. (Space Comm)
		LOW	HIGH
		ABILITY TO BUY INFORMATION	

SPACE MARKET MODEL DEVELOPMENT PROJECT

Overall Conclusion

- The information **available** for space commercialization

  Government R&D and procurement

- The information **required** for space commercialization

  Private-sector R&D and products

SPACE MARKET MODEL DEVELOPMENT PROJECT
PHASE II
GOAL

To build and test a clearinghouse for space business information

SPACE BUSINESS INFORMATION CENTER

SPACE BUSINESS INFORMATION CENTER

Goals

RESEARCH

- To continue to study the information needs of the space business community
- To continue to investigate productive techniques to meet those needs

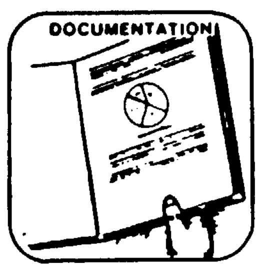
PROTOTYPE

- To collect available information for ready access
- To develop new information for commercial use

SPACE BUSINESS INFORMATION CENTER

A Complete Information System

People



Information

Tools

SPACE BUSINESS INFORMATION CENTER

Space > aerospace

Business > science & technology

Information > policy analysis

SPACE BUSINESS INFORMATION CENTER

Information System

SPACE BUSINESS

APPLICATION MARKETS

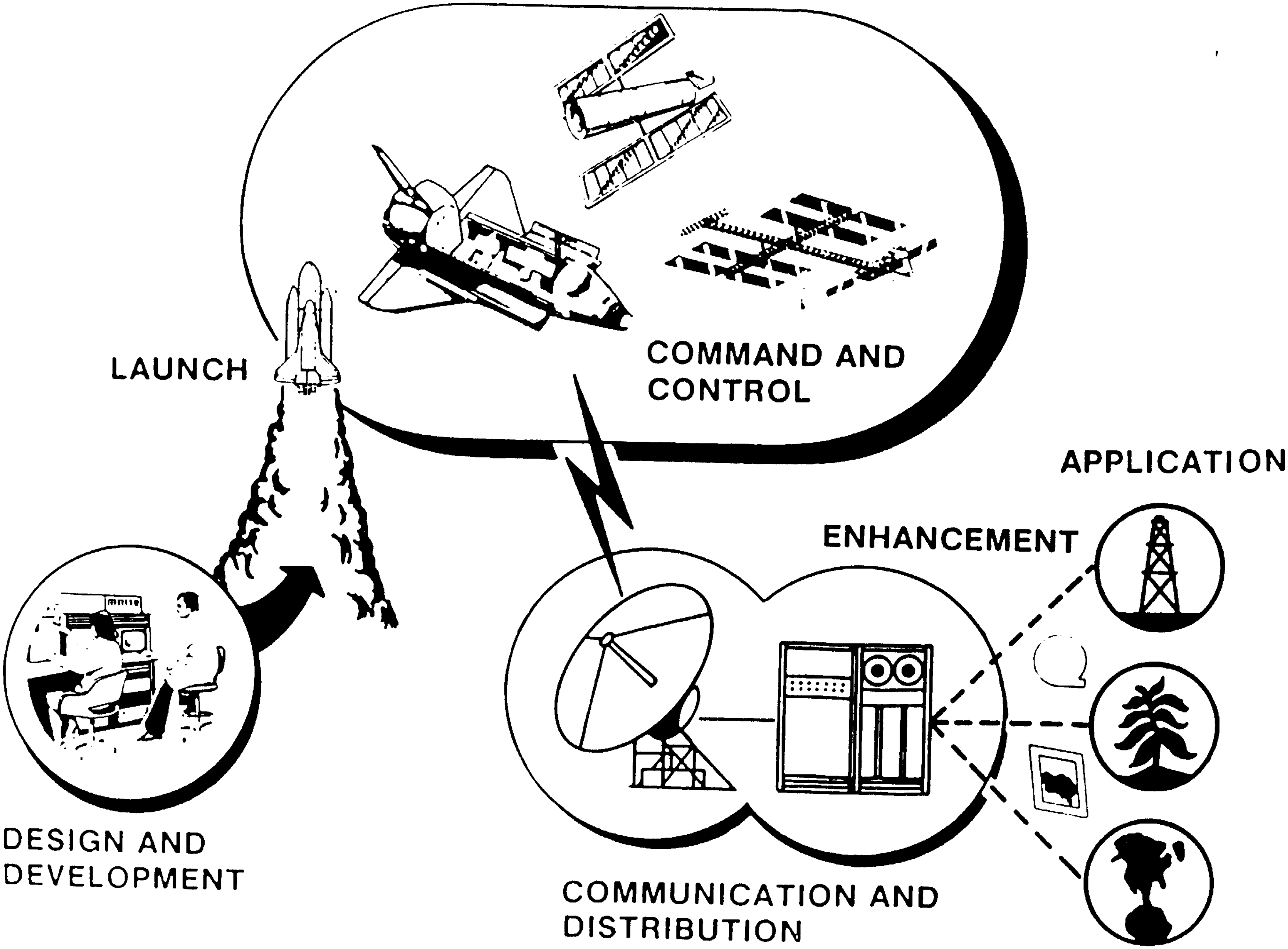
- Communication
- Remote Sensing
- Material Processing

INFRASTRUCTURE MARKETS

- Manufacturing
- Transportation
- Support Service

A SPACE BUSINESS

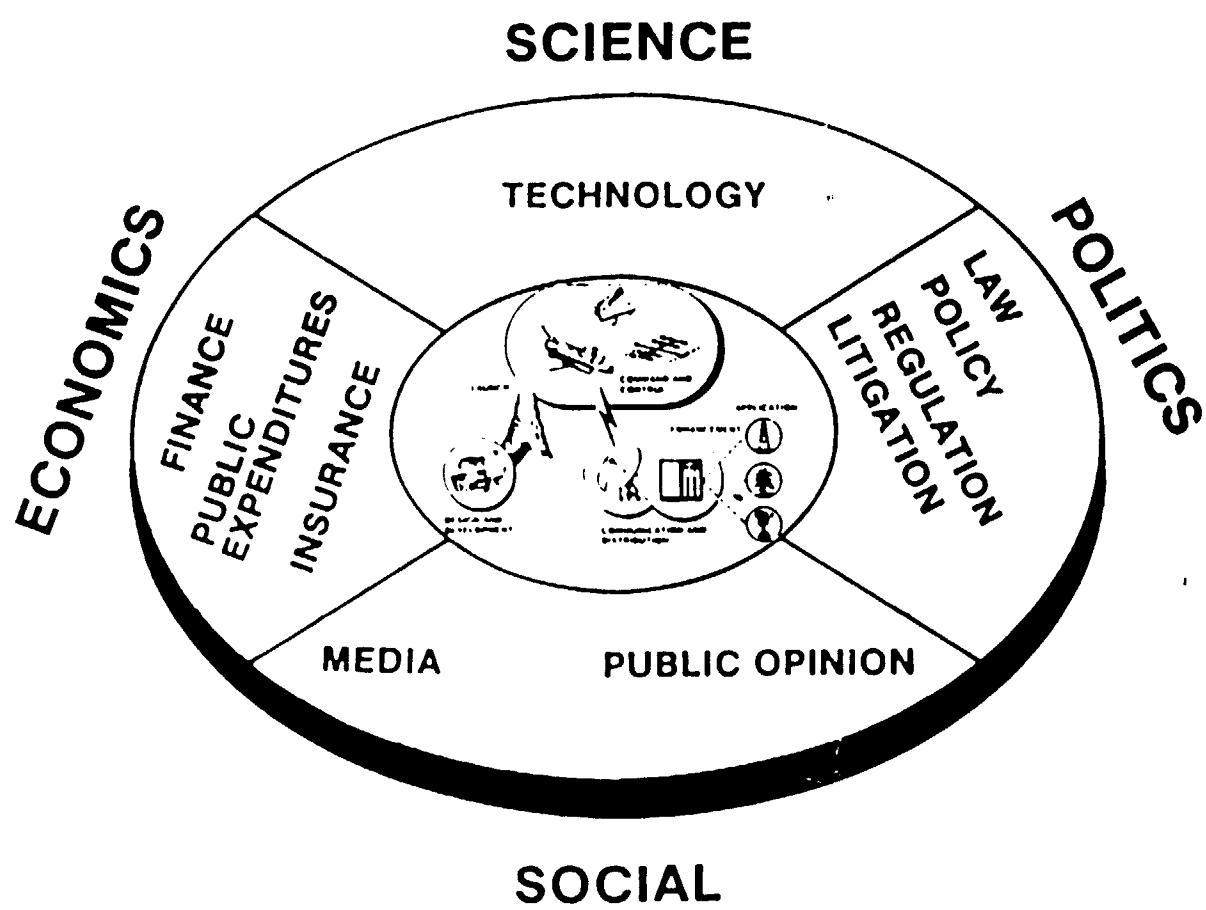
EXAMPLE: REMOTE SENSING



SPACE BUSINESS INFORMATION CENTER

Information System

CONTEXTUAL AREAS

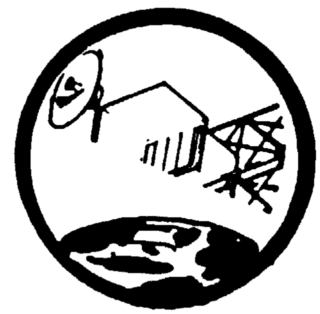


SPACE BUSINESS INFORMATION CENTER

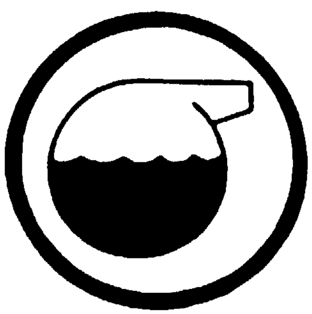
Information System

PROTOTYPE MARKETS

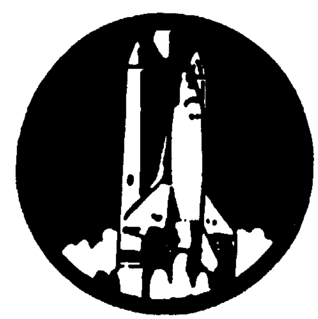
Small Shuttle Payloads



Remote Sensing of Land Areas



Processing of Biological Materials



Commercial Space Transportation

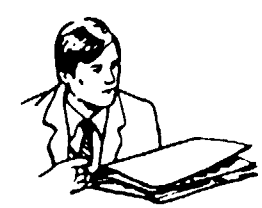
SPACE BUSINESS INFORMATION CENTER

Information System

PEOPLE

Information Specialists

Market Analysts

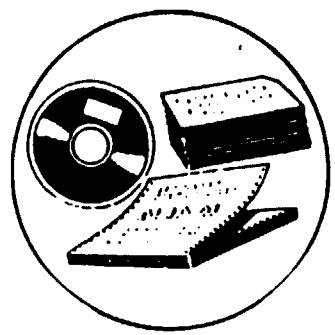


Client Representatives

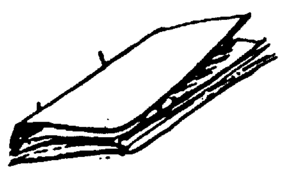
SPACE BUSINESS INFORMATION CENTER

Information System

FORMS

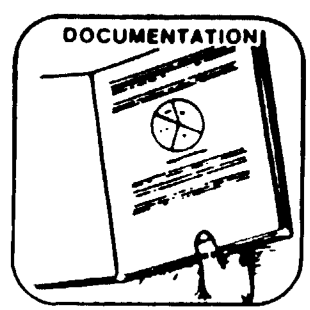


Data



Periodicals

Experts



Documents

SPACE BUSINESS INFORMATION CENTER

Information System

TOOLS

- DBMS modules
- Local processor
- Electronic networks

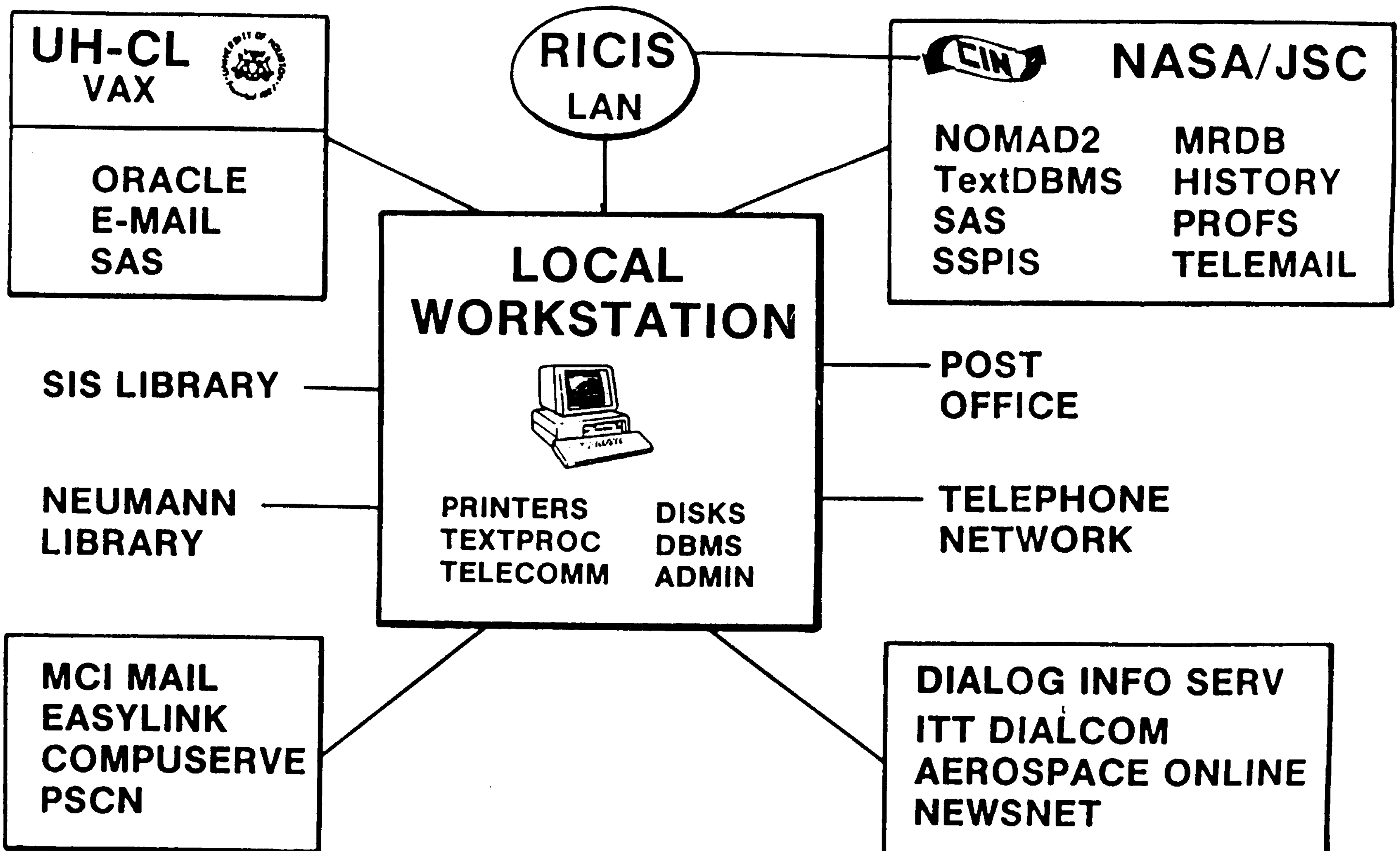
SPACE BUSINESS INFORMATION CENTER

Information System

DATABASES

- Products & services
- People & organizations
- Events & plans
- Physical infrastructure
- Budgets & expenditures
- Information resources

INFORMATION WORKSTATION



SPACE BUSINESS INFORMATION CENTER

Test Products and Services

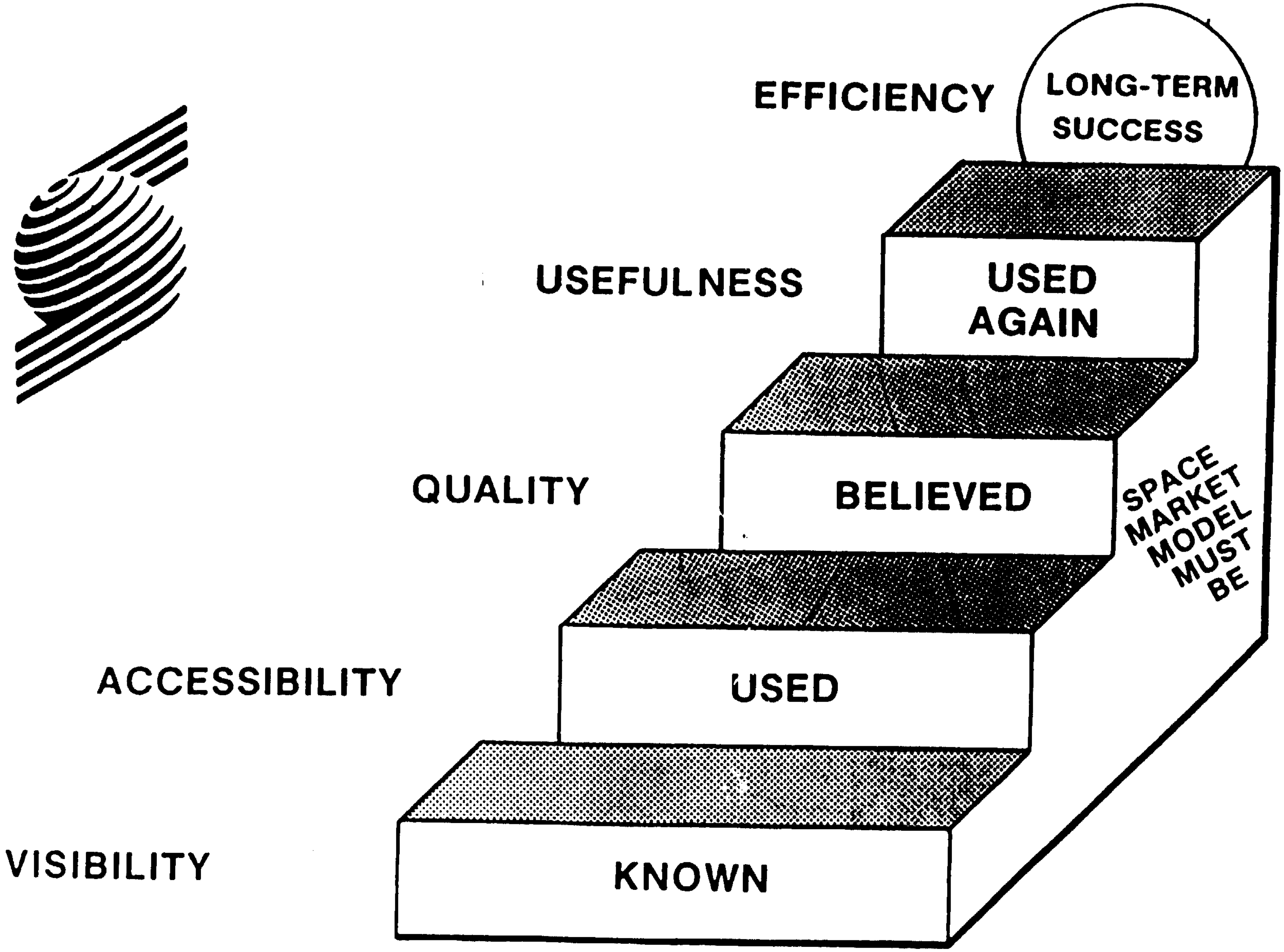
- **Publications**
  - guides to shuttle payloads
    - remote sensing
    - biotechnology in space
    - space transportation
  - background papers on significant issues
- **Information Clearinghouse**
  - existing information on request
  - accessible, accurate, timely

SPACE BUSINESS INFORMATION CENTER

Test Products and Services

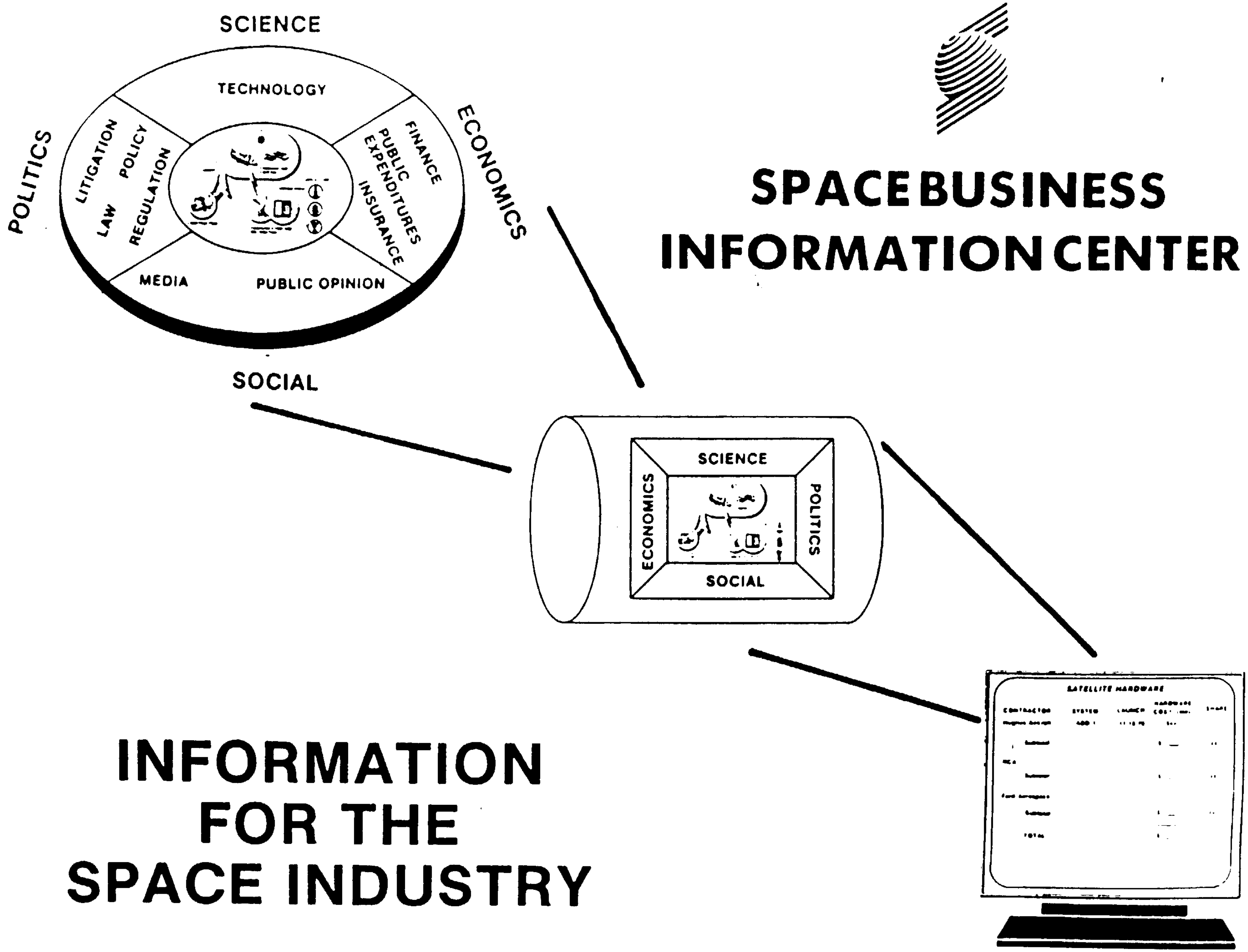
- **Research Reports**
  - primary data collection
  - original analysis
- **Briefings**
- **Custom Databases**
- **Economic Model**

SPACE BUSINESS INFORMATION CENTER



4 STEPS TO SUCCESS

SPACE BUSINESS INFORMATION CENTER



INFORMATION FOR THE SPACE INDUSTRY

Research Institute
for
Computing and Information Systems

INFORMATION MANAGEMENT

Dr. Peter C. Bishop
Studies of the Future
University of Houston-Clear Lake

ADP Strategic Plan

NASA Johnson Space Center

Connected Systems

Common Tools

Iterative Development

Strategic and Tactical Information

OBJECTIVE

Productive Information Technology

Information Management

TARGET

Manager...

...Customer

Information Management

PRODUCTIVE TOOLS

ALREADY	NOT YET
Word Processing	Text Retrieval
Spreadsheet	Data Retrieval
Electronic Mail	Graphics

RESEARCH APPROACH

- Surveys
- Forecasts
- Plans
- Demonstrations

Information Management

SURVEYS

What is out there?

- ➡ Environmental Scanning for Information Processing
- ➡ Clear Lake Area Computer Capability Census

Information Management

FORECASTS

What might be out there?

➡ Space Station Advanced Virtual Electronic Documentation (SSAVED)

Information Management

PLANS

How to get there?

- ➡ Word One Replacement

Information Management

DEMONSTRATIONS

What might it look like?

- Space Shuttle Payload Information System (SSPIS)
- Space Market Model Development Project (SMMDP)

OUTCOME

Productive
Information
Technology

Information Management
FY88 Research Projects

- PLANS

  - ➡ Long-Range Plan for the Commercial Development of the Space Station
  - ➡ Methodologies for Integrated Information Management Systems
  - ➡ Integrated Parametric Planning Models for Budgeting and Managing Complex Development Projects

Information Management

FY88 Research Projects

- DEMONSTRATIONS

  ➡ Management Information and Decision Support Environment (MIDSE)

  ➡ Image Management and Access (Project ICON)

OUTCOME

Productive Information Technology

DATABASE STRATEGIES AND PROTOTYPES

Timothy N. Tulloch
Vice-President
TNT Consulting

PURPOSE:

1. Describe the principles utilized to develop productivity tools.

2. Illustrate the use of a tool which utilizes these principles.

Definition of a Productive

Tool

1. Independent of Application
2. Enhances User's Capability
3. Operates Intuitively
4. Flexible Entry Techniques
5. Keeps Track of Detail
6. Remembers History
7. Modifiable Parameters
8. Access to Related tools

SPACE STATION ADVANCED VIRTUAL ELECTRONIC DOCUMENTATION

Work in Progress

Presented by Dr. Chris Dede

September, 1987

SSAVED

FOCUS: Evolution of SSP Documentation Systems

- forecast probable shifts in hardware capabilities, knowledge representation format, user interface functions, database approaches, and cognitive structure
- prepare technical and managerial personnel for new electronic approaches to knowledge creation, capture, transfer, and utilization

By anticipating changes, can facilitate
an orderly progression
of increasingly sophisticated strategies

SSAVED

PROJECT OUTCOMES:

-- synthesis of advanced research

-- network of external resources

-- menu of high leverage topics

Target: concepts practical in mid-range time frame
specific to SSP needs
not currently major issues

SSAVED

<u>Different Perspectives Based on Orientation</u>

-- "electronic documentation"

-- "configuration management and control"

-- "knowledge creation, capture, transfer, and utilization"

-- ???

depends on role of information in task performance

C-4

SSAVED

Changing Aspects of Documentation Systems:

- information technology synthesis
- hardware power
- standardization and connectivity
- advances in conventional DBMS
- different representational formats
- mimetic, intelligent user interfaces
- computer-supported cooperative work

SSAVED

EMERGING ISSUES:

- evolution of technical workstations
- data bases ---> knowledge bases
- "hypermedia" for knowledge representation
- advanced user and organizational interface capabilities

an information abundant environment will alter management policies, training, implementation, and organizational culture

SSAVED

Illustration: User Interface Capabilities

* managing trouble
  - --self explication through direct manipulation
  - --microworlds and artificial realities

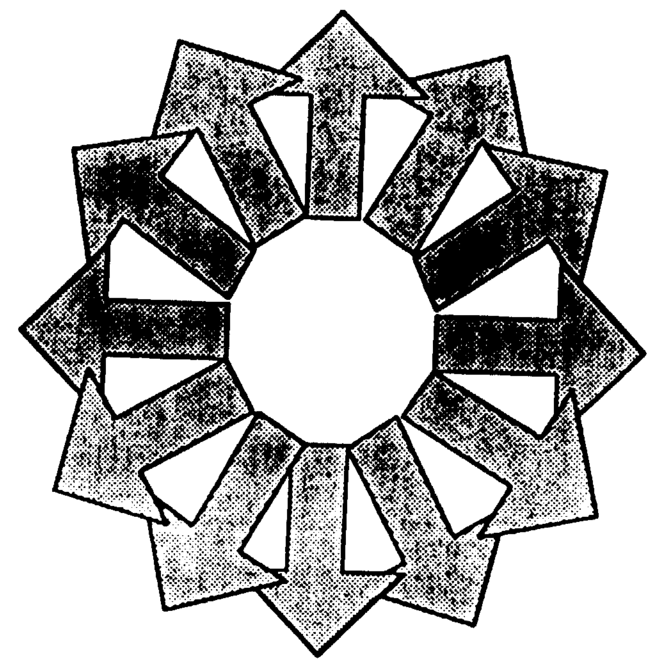
* controlling multiple, simultaneous processes
  - --linearizing
  - --advice giving interfaces

* resolving task performance ambiguities
  - --natural language
  - --semi-autonomous agents

Cognition Enhancers

SSAVED

EXTERNAL FORCES:



- artificial intelligence
- cognitive science
- software engineering
- computer engineering
- synthesis of all information technologies

initial SSP information systems must be *very* flexible

INFORMATION MANAGEMENT

YEAR 2

LLOYD ERICKSON
OCTOBER 15, 1987

MANAGEMENT INFORMATION DECISION SUPPORT ENVIRONMENT (MIDSE)

- PROBLEM
- GOALS
- STRATEGY
- APPROACH
- FUTURE

PROBLEM

- LOTS OF DATA
- LITTLE OR NO ELECTRONIC ACCESS BY DECISION MAKERS

JSC DATA ENVIRONMENT

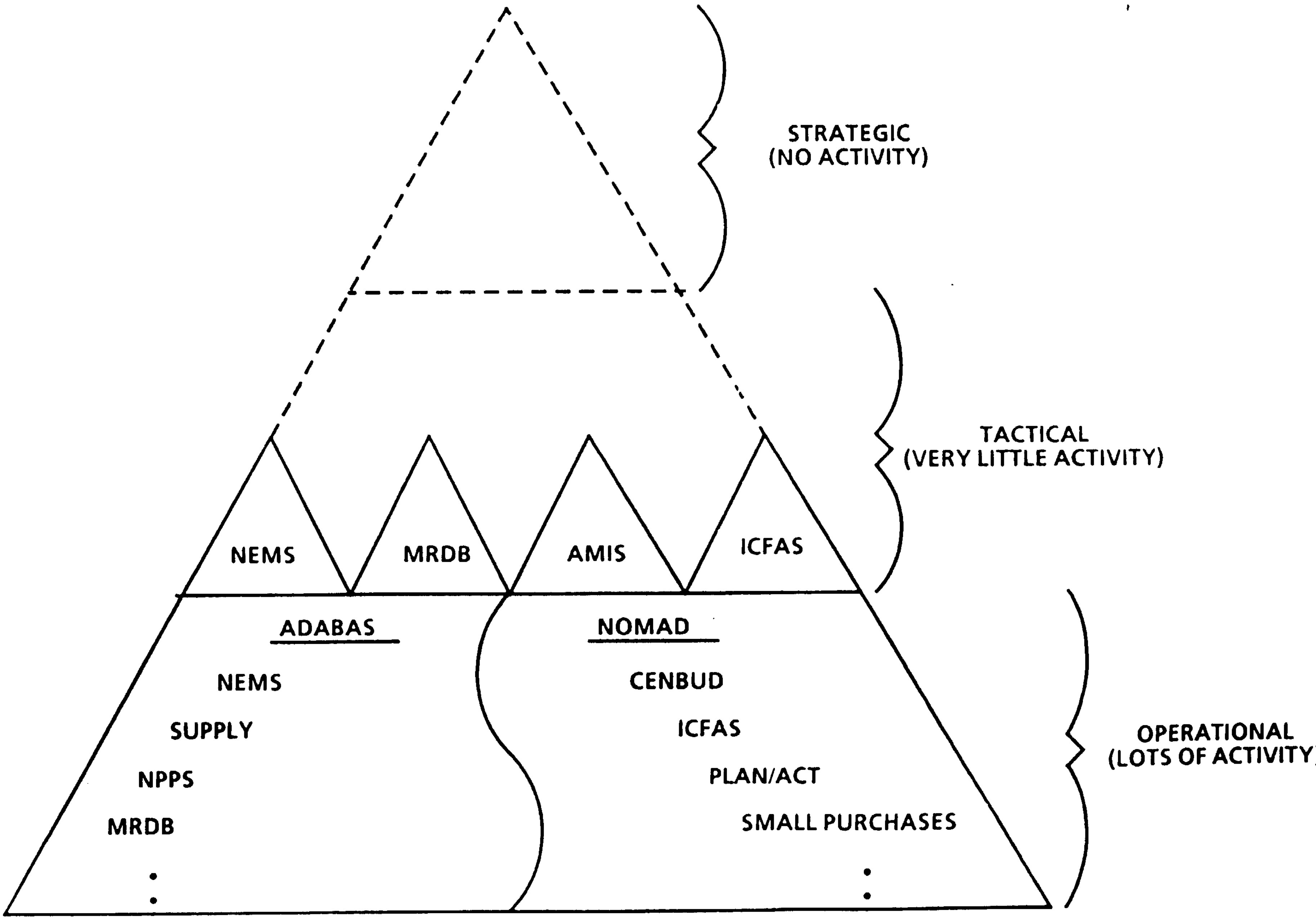
- DIFFERENT KINDS OF DATA
  - TABULAR *
  - TEXTUAL
  - IMAGE

- DIFFERENT HARDWARE
  - IBM M/F (+ TERADATA) *
  - VAX
  - P/C'S *
  - MACINTOSH
  - APOLLO'S & SUN'S
  - CIN & ASYNC *
  - TERMINALS *

- DIFFERENT DATA HANDLING SOFTWARE
  - NOMAD *
  - ADABAS *
  - ORACLE
  - 3 GL (FORTRAN, COBOL, ETC.) + FILES

* INCLUDED IN INITIAL MIDSE EFFORT

JSC INFORMATION SYSTEMS STATUS



GOALS OF MIDSE

PROTOTYPE-SYSTEMS

- PROVIDE ACCESS TO DATA IN SYNTAX-FREE, INTUITIVE ENVIRONMENT
- TRANSFER DATA TO SPREADSHEET, GRAPHICS, WORD PROCESSING AND PROFS
- CREATE SAME ENVIRONMENT ON PC AND MAINFRAME
- FUNCTIONALLY RICH, EXTENSIBLE SYSTEM FOR ADVANCE END-USERS

DEVELOPMENT ENVIRONMENT

- PROVIDE TOOLS TO AUTOMATE DEVELOPMENT OF MIS APPLICATION

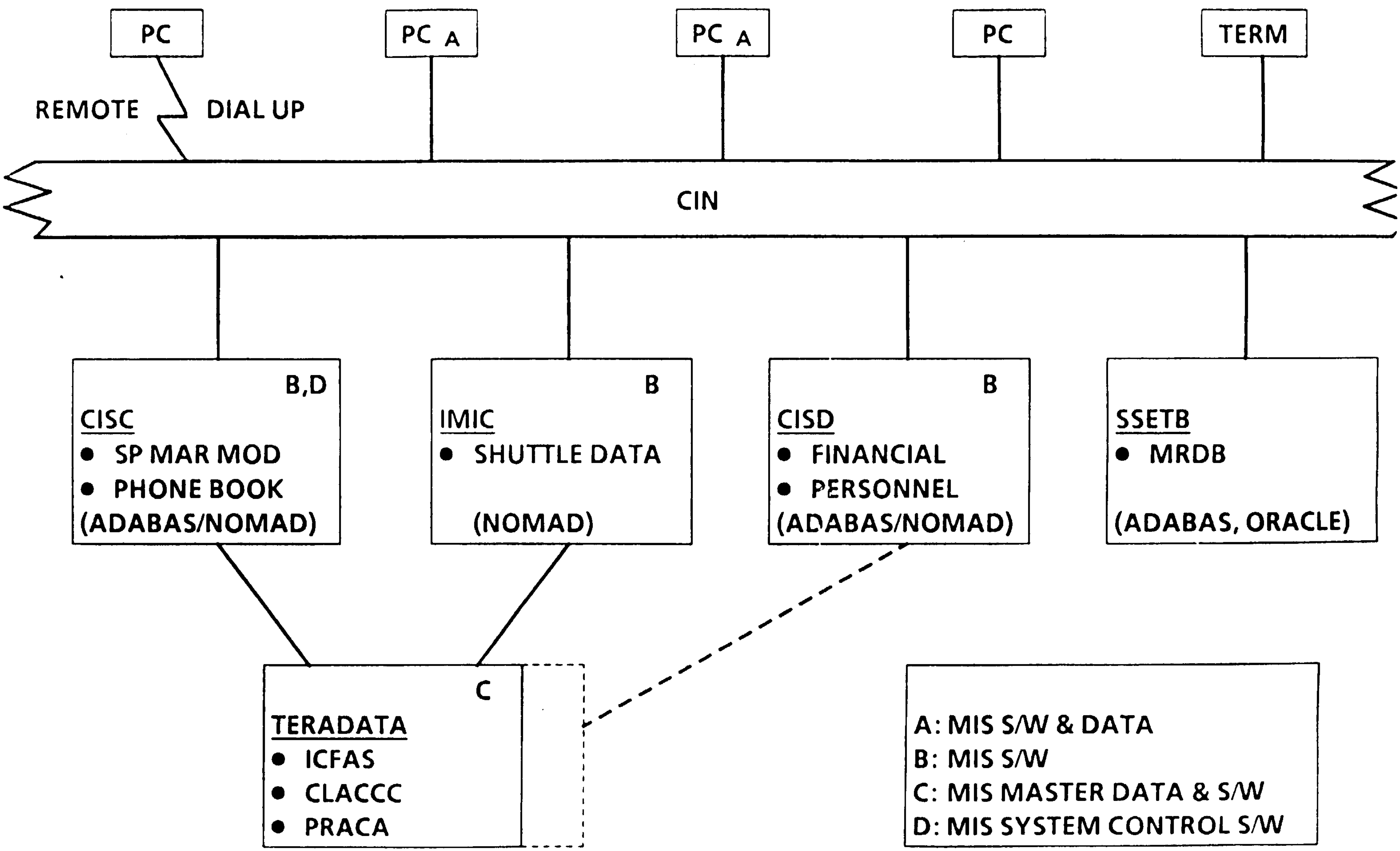
MIDSE STRATEGY

- SMALL INCREMENTAL STEPS
- QUICK PROTOTYPES
- HIGH INTERACTION WITH USERS
- LOW COST R&D
- LEVERAGE DEVELOPMENT WITH VENDORS AND OTHER USERS
- EXISTING HARDWARE AND SOFTWARE

MIDSE APPROACH

- SELECT MIS APPLICATION
- INTERACT WITH PANEL
- DEVELOP TOOLS

JSC MANAGEMENT INFORMATION ENVIRONMENT
LATE FY 88



INFORMATION MANAGEMENT FUTURES AT JSC

- DATA TYPES
  - TABULAR: NEEDS PROTOTYPE DEVELOPMENT
  - TEXT: NEEDS APPLIED RESEARCH
  - IMAGES: NEEDS BASIC RESEARCH
- HARDWARE
  - PC'S CONTINUE TO GET MORE POWERFUL
  - FASTER NETWORKS FACILITATE DISTRIBUTED APPLICATIONS
- SOFTWARE
  - 0S2 WILL IMPROVE PC FUNCTIONALITY
  - GRAPHIC ORIENTATION FACILITATE EASE OF USE

ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS

Conveners: **Terry Feagin**, UH-Clear Lake

Timothy F. Cleghorn, Mission Planning and Analysis Division, NASA/JSC

Introduction and Overview
Timothy F. Cleghorn

Communication and Tracking Expert Systems for the NASA Space Station
T.F. Leibfried, UH-Clear Lake

Simulation of Robotic Space Systems
Yashvant Jani, LinCom Corporation

Robotic Path Planning and Software Testbed Architecture
Richard D. Volz, University of Michigan

Fuzzy Set and Related Theory for Failure Detection and Control in Space Systems
Thomas B. Sheridan, Massachusetts Institute of Technology

A Computer Graphics Testbed to Simulate and Test Vision Systems for Space Applications
John B. Cheatham, Jr., Rice University

Demonstration of a 3D Vision Algorithm for Space Applications
Rui J.P. deFigueiredo, Rice University

Summary of the Artificial Intelligence and Expert Systems Technical Session

Dr. Terry Feagin introduced the technical session by identifying the various research projects underway in the technical area. Dr. Timothy Cleghorn gave an overview of the projects in this technical area that are funded by the Mission Manning and Analysis Division. All of the projects except the work on the communications and tracking expert system fall in this category. These projects are ultimately directed at the formation of a Robotics Software Simulation Testbed within MPAD at JSC.

The work in expert systems for communications and tracking was presented by Dr. T.F. Leibfried, who described how the team of three faculty and four students were studying various ways of approaching the problems of detecting, isolating and recovering from faults in the communications and tracking systems to be used on the space station. He described that as software in the area is developed, a software simulator must be employed to exercise and test the system. Two systems for fault detection and diagnosis (as designed by Harris and TRW) are being reviewed closely for ideas about the best way to approach this problem. Both of these systems are based upon using expert systems for the diagnostic portion of the system, as various alternative possible causes for the observed problems are evaluated. The expert systems are written in ART (an expert system shell language) and run on the Symbolics LISP machine. The TRW system runs on several machines and much of the code is written in the language C. Other work under this project involves the development of an explanation facility for the expert systems, a distributed collection of cooperating expert systems, and extremely fast fault diagnosis for single point failures using bit-strings.

The work on robotic path planning at the University of Michigan was described by Dr. Kang Shin. He explained how one can evaluate the various possible routes that a robot might take through a set of obstacles in order to reach a desired destination. He also explained how one could take into account a measure of safety as the various obstacles were circumvented, so as to avoid collisions due to small deviations in the path due to an inability to control the robot's movements precisely. Also, he showed how blind alleys and unacceptable paths could be labeled recursively until acceptable paths were identified for further evaluation so as to determine the optimal path for the robot to follow. He also discussed the generalization of the approach to three dimensional problems.

The work on fuzzy sets for failure detection and control was presented by Dr. Thomas B. Sheridan. He described how fuzzy sets could be used to model uncertainty and how this approach could be used effectively in failure detection and control. He described how objective measures of uncertainty could be obtained.

The work on a graphics testbed for computer vision systems was described by Dr. John Cheatham of Rice University. He described how they are developing a graphics system for simulation scenes that might require analysis by a robotic computer vision system. The system is presently able to simulate the appearance of an artificial satellite under various lighting conditions.

The work on 3-D vision algorithms for computer vision was described by Dr. Rui de-Figueiredo of Rice University. The algorithms being developed allow selected objects viewed by the robotic computer vision system to be recognized readily. Various invariant properties of the objects are used to aid in the identification process.

Dr. Yashvant Jani described the work on simulation of robotic space systems presently being conducted at LinCom Corporation. The research involves applying expert systems to support Du-board navigation (ONAV) ground controllers in operational training associated with monitoring the status of navigation sensors during the entry phase of shuttle flights.

T.F. Leibfried

University of Houston

Clear Lake

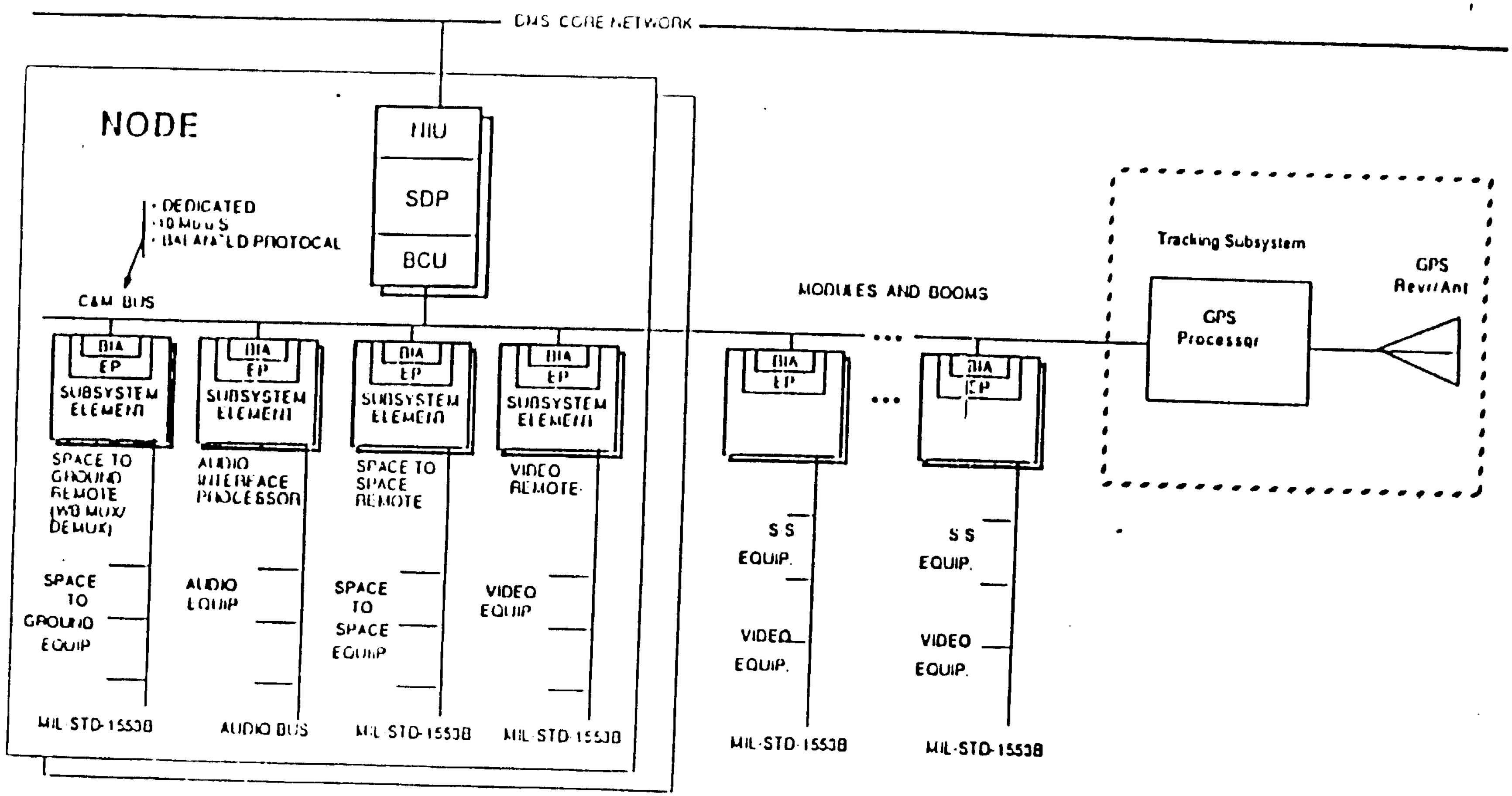


Figure 1: C&MS HARDWARE ARCHITECTURE

ORIGINAL PAGE IS
OF POOR QUALITY

AI-1 Expert Systems Study for T&C

Supervisor Oron Schmidt EE7

UHCL Team Members :

Faculty	Students
T. F. Leibfried	David Overland
T. Feagin	Dennis Stevens
J. Giarratano	Gary Young
	Albert Rodriguez

Objectives

Contractor Studies

Harris Corp. (Mlb. Fla.)
TRW (Redondo Beach Cal.)

Lessons Learned

Future

Executive Summary
RICIS Research Activity AI.1
Expert Systems for Space Station T&C
17 June 1987

Section 1:

Objective: Study Methodologies and Techniques for Optimal Development of Configuration Management, Fault Detection Isolation and Recovery and Explanation Expert Systems

Overview : The space station environment will require malfunction detection, isolation and recovery from all causes. Station personnel will probably not be proficient in communication systems analysis and management. There will be a need for on-board expert systems to provide assistance to the crewmen. These systems will have to be distributed in nature because of the nature of the hardware design. The techniques and principles of good expert systems software design need to be researched so as to provide workable implementation guidelines for production system contractors.

Status: Preliminary analysis shows that modularization and and using a mixture of so-called Expert System Languages and more conventional procedural languages is feasible. Reusable expert systems code and the feasibility of automated knowledge acquisition are in the early stages of investigation.

Section 2

Technical Issues:

1. Should the development environment for ES be different than the Implementation environment ? (The answer so far is "probably".)

2. What can be done to make expert system language software more maintainable and efficient ?

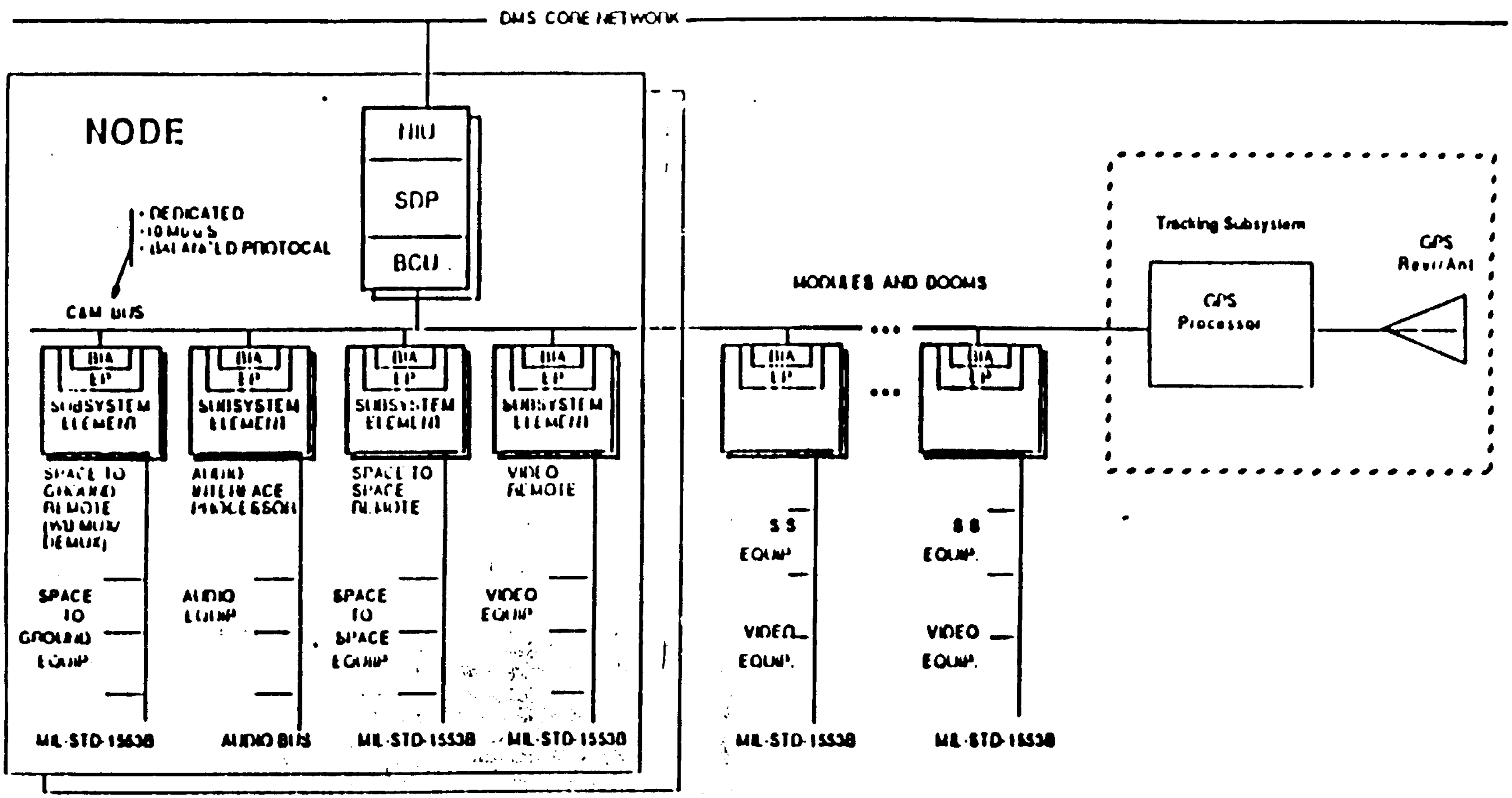


Figure 1: C&MS HARDWARE ARCHITECTURE

ORIGINAL PAGE IS OF POOR QUALITY

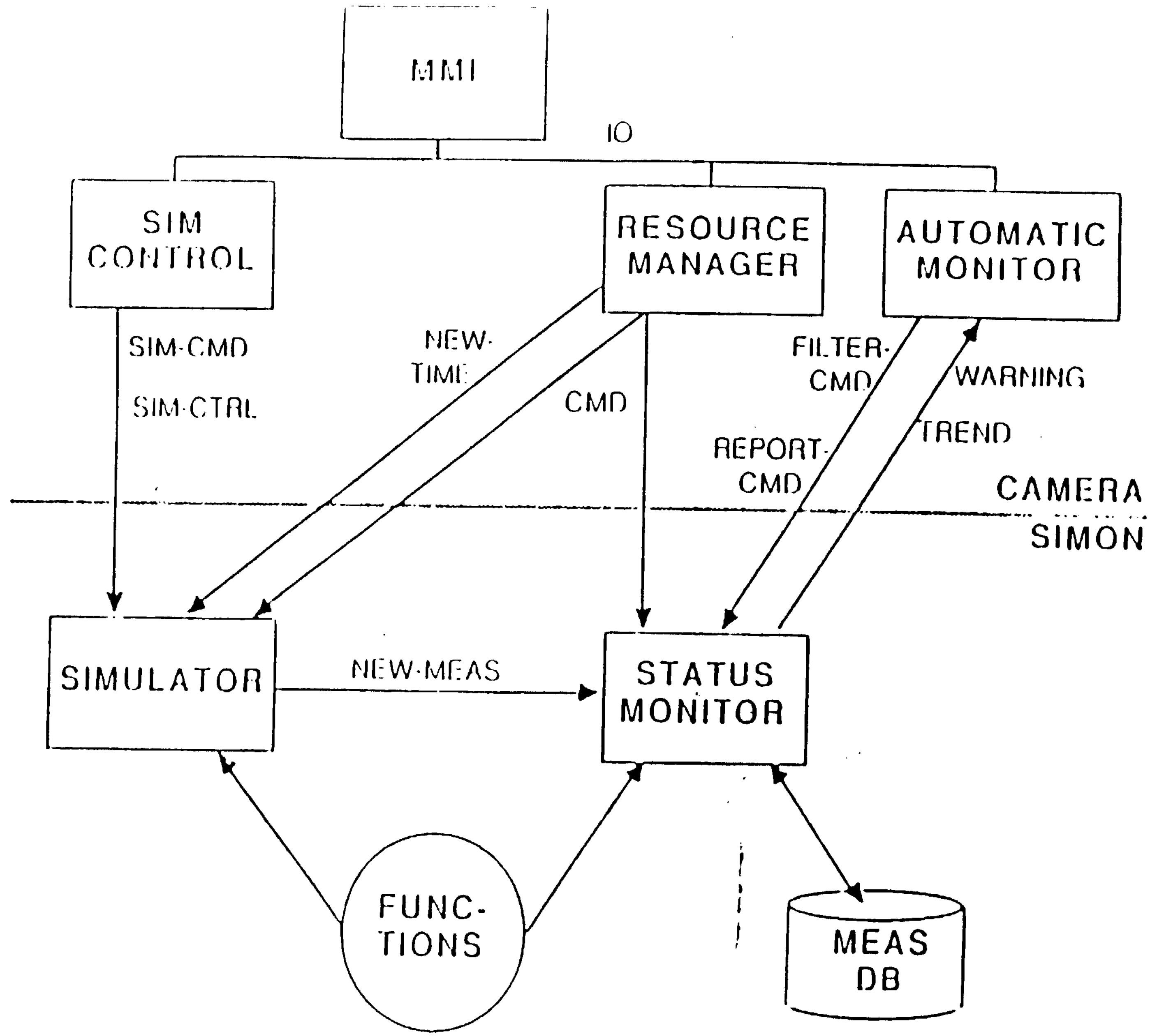


Figure 2. CAMERA Architecture

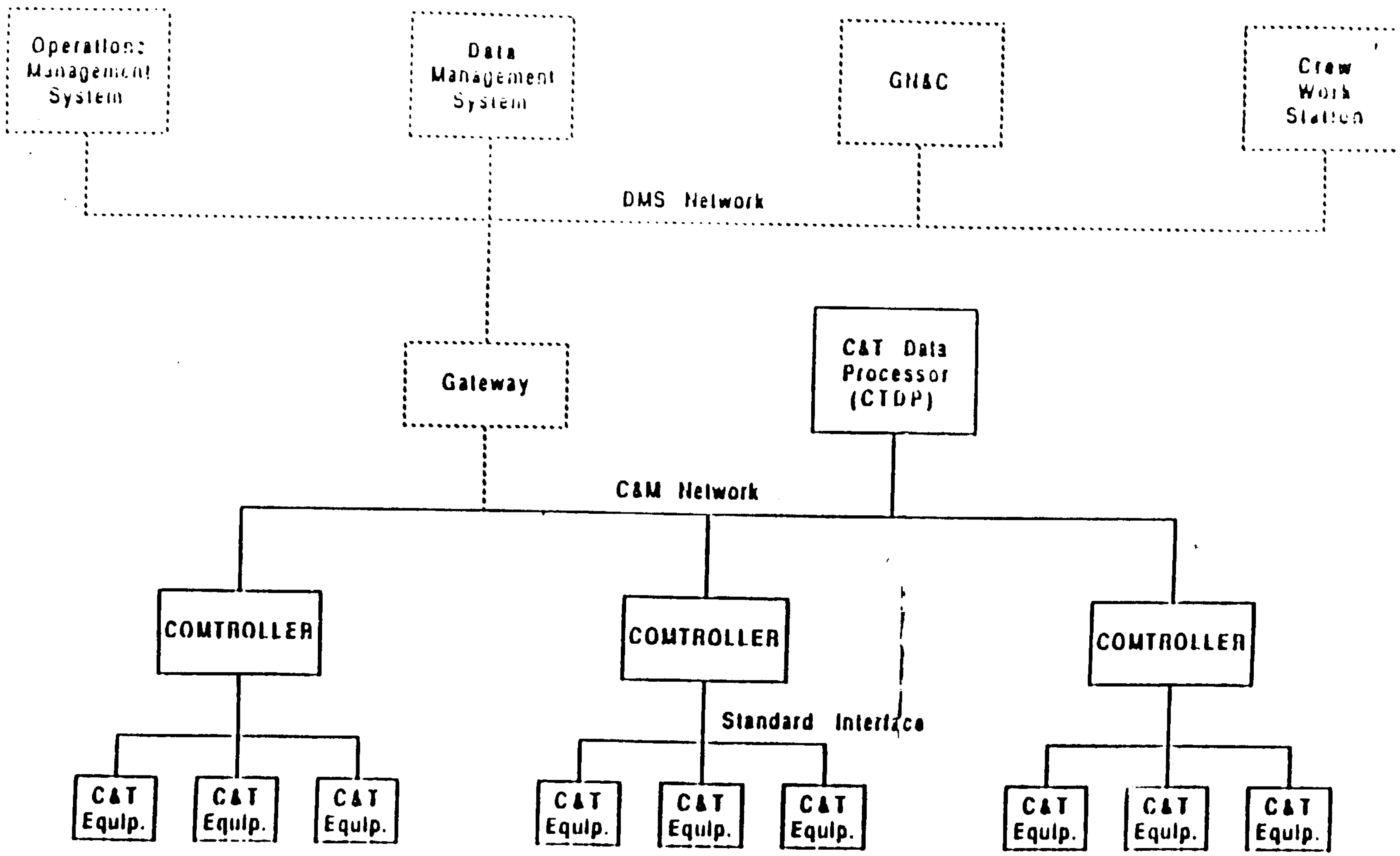


Figure 3-1 Control and Monitoring Subsystem Architecture

TRW

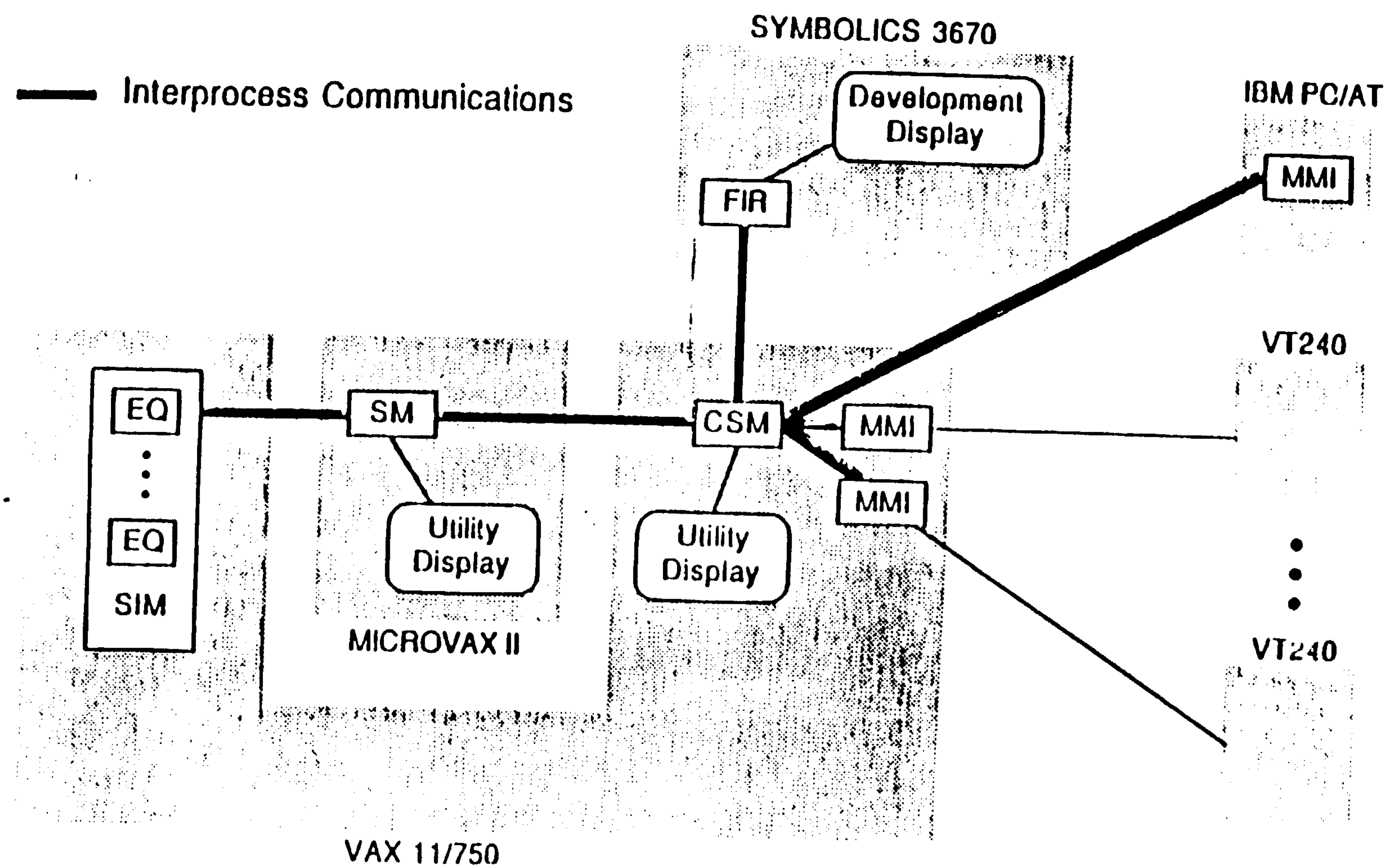


Figure 3-3 Typical C&M Testbed Software Configuration

TRW

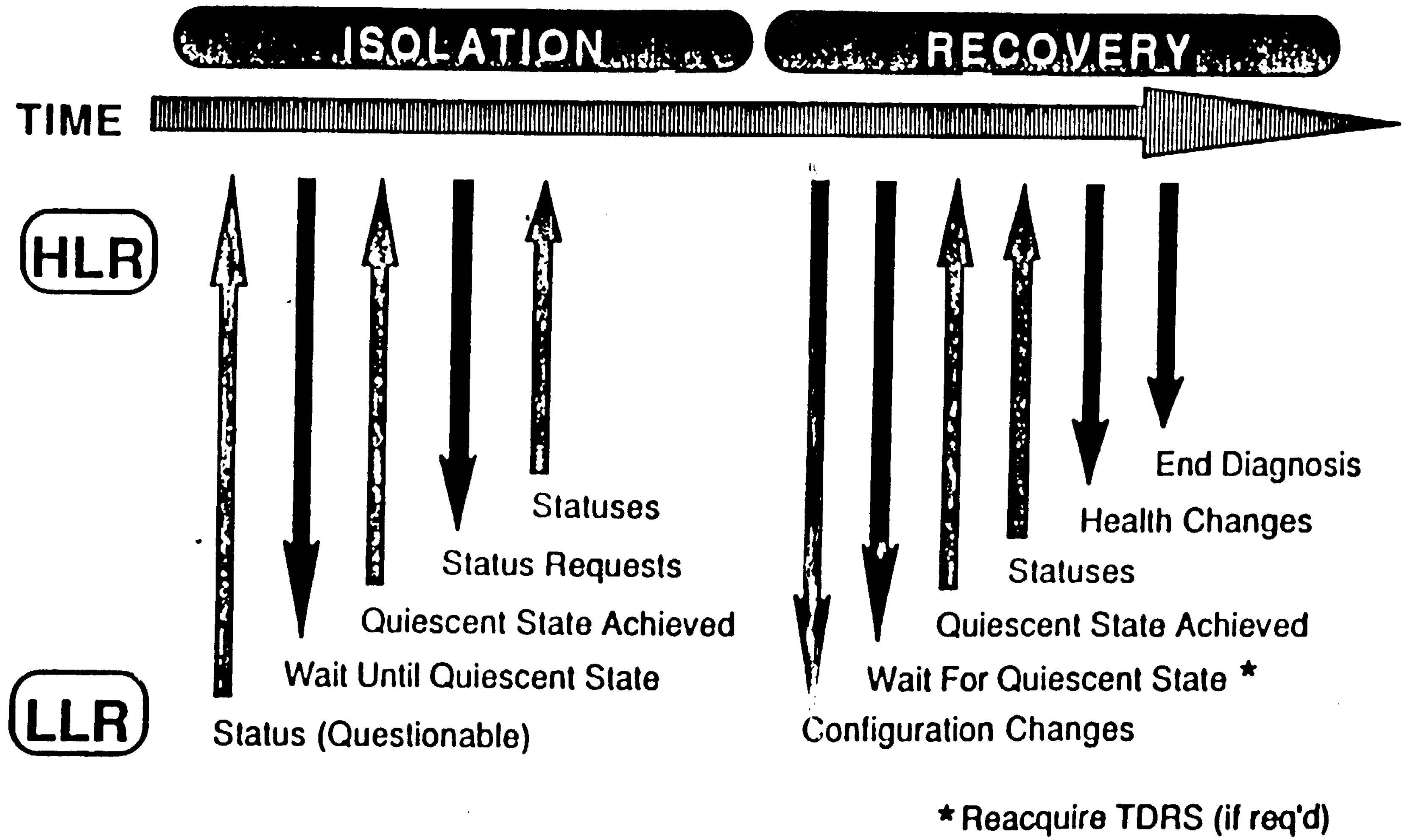


Figure 7-1
Messages plotted against the Isolation and Recovery Phases and against time.

Lessons Learned

Simulators

Function

In support of FDIR
Test and Explanation

Implementation Environment

FDIR

Function

In support of Resource Management
Isolation and Explanation

Development Environment

Implementation Environment

Low level Detection and Isolation
Higher level Reasoning

Future Investigation

Generic Simulators (Table driven)

ANS for Modeling ?

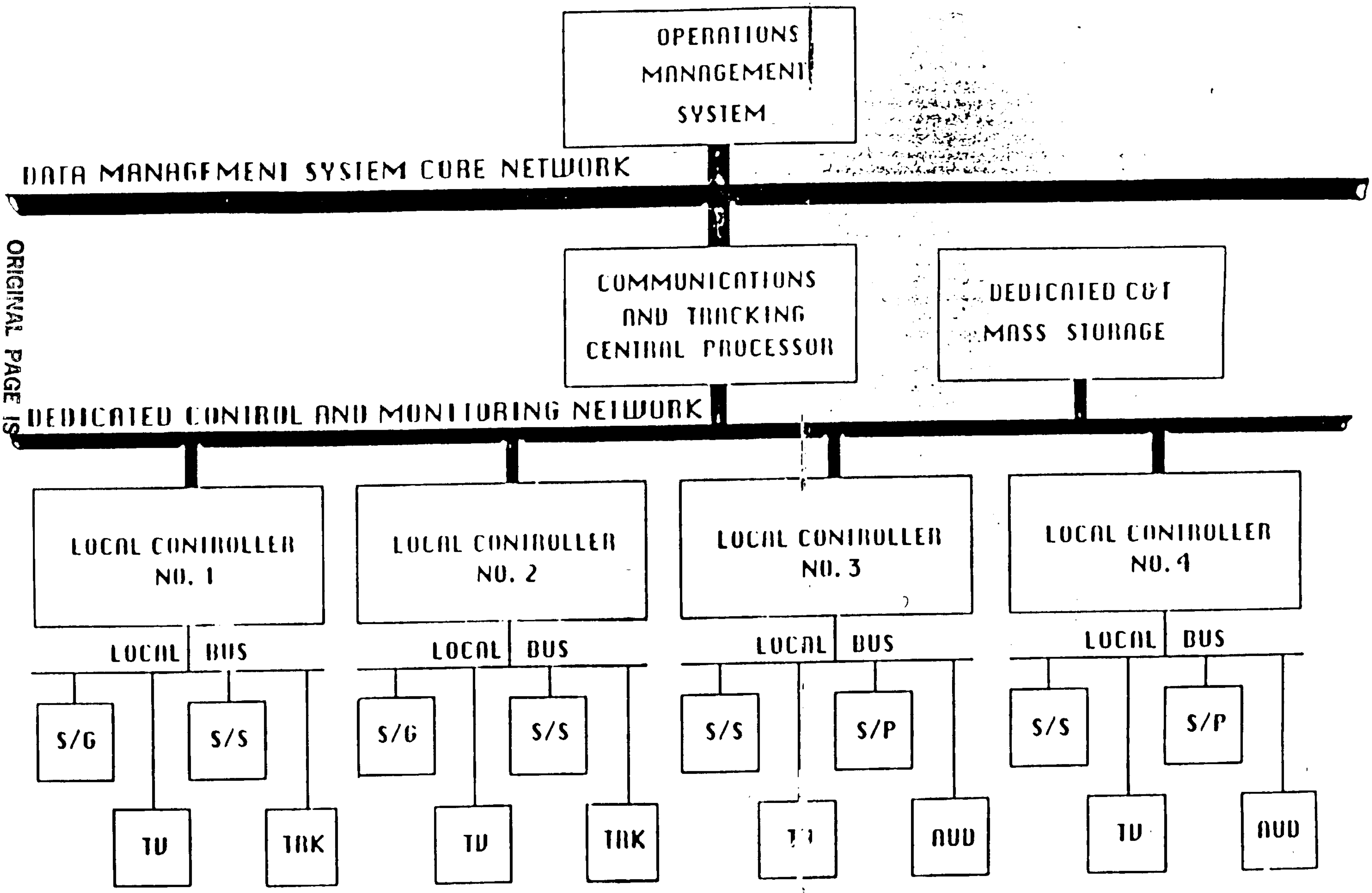
MMI Needs

Expert System Implementations

Rule-Based
Procedural

Rule-Based Conversion to Procedural ?

Automated Knowledge Acquisition



COMMUNICATIONS AND TRACKING CONTROL AND MONITORING SUBSYSTEM

SIMULATION OF ROBOTIC SPACE OPERATIONS

INTEGRATION OF VISION ALGORITHMS INTO AN ORBITAL OPERATIONS SIMULATION

YASHVANT JANI, PhD
WILLIAM L. OTHON
LinCom CORPORATION

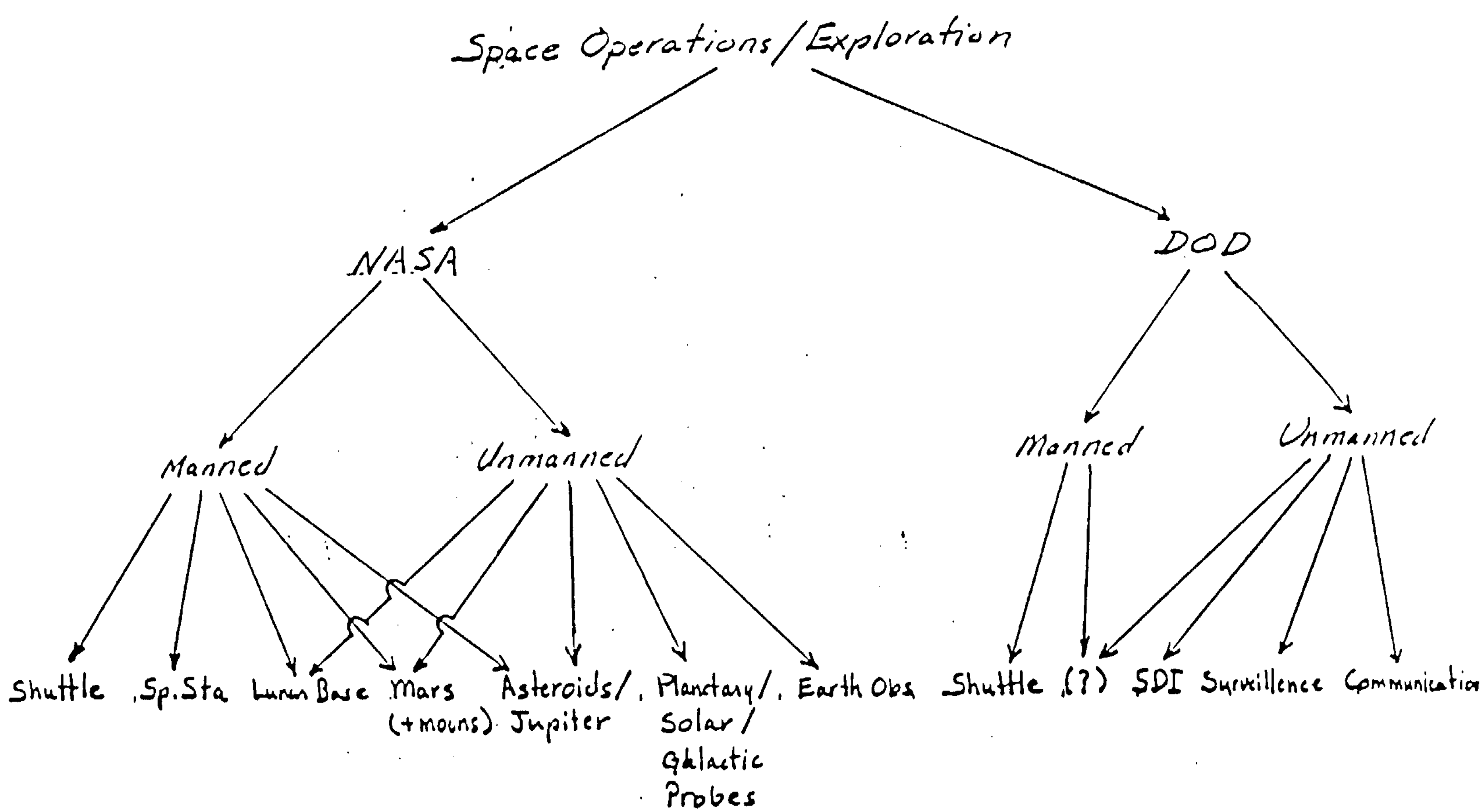
RICIS Symposium
15 October 1987

AGENDA

- OBJECTIVES
- USE OF SIMULATION
- INTEGRATION OF ROBOTICS / VISION ALGORITHMS INTO AN ORBITAL OPERATIONS SIMULATION
- CURRENT EFFORT: INTEGRATION OF VISION ALGORITHMS FROM RICE UNIVERSITY WITH ORBITAL MANUVERING VEHICLE (OMV) MODEL
- PROJECT STATUS
- FUTURE EFFORT

OBJECTIVES

- DEVELOP A *TESTBED* FOR INTEGRATION OF ROBOTICS SUBSYSTEMS AND SPACE VEHICLES SIMULATION
  - IMPLEMENT VISION/ROBOTICS ALGORITMS
  - PERFORM SYSTEMS INTEGRATION ANALYSIS
- STUDY OPERATIONAL ASPECTS OF ROBOTIC SPACE SYSTEMS AND MISSIONS


Space Operations/Exploration
NASA
DOD
Manned
Unmanned
Manned
Unmanned
Shuttle
Sp. Sta
Lunar Base
Mars (+moons)
Asteroids/Jupiter
Planetary/Solar/Galactic Probes
Earth Obs
Shuttle
(?)
SDI
Surveillence
Communication

SCOPE:

- NASA MISSIONS/VEHICLES WITH ROBOTICS
  - SHUTTLE
  - SPACE STATION
  - OMV
  - SURFACE AUTONOMOUS EXPLORATION VEHICLES (AEV)
    - LUNAR
    - MARS (+ MOONS)
    - OTHER SOLID PLANETARY - SIZED BODIES
  - MATERIAL PROCESSING PLANTS
    - LUNAR
    - MARS (+ MOONS)
    - ASTEROIDS
    - COMETS
    - OTHERS

OBJECTIVES:

- TO DEVELOP AN UNDERSTANDING OF THE COMPUTING, COMMUNICATIONS AND CONTROL TECHNOLOGIES NEEDED FOR THE COORDINATED ROBOTIC MANIPULATION OF OBJECTS IN A SPACE ENVIRONMENT.

- TO DEVELOP INTEGRATED SUBSYSTEM CONCEPTS IN SUFFICIENT DEPTH TO IDENTIFY AND DESCRIBE:

  - REQUIREMENTS FOR INTEGRATING COORDINATED ROBOTICS UNITS INTO THE EVOLUTIONARY SPACE STATION COMPUTING, COMMUNICATIONS, AND CONTROL SYSTEMS.

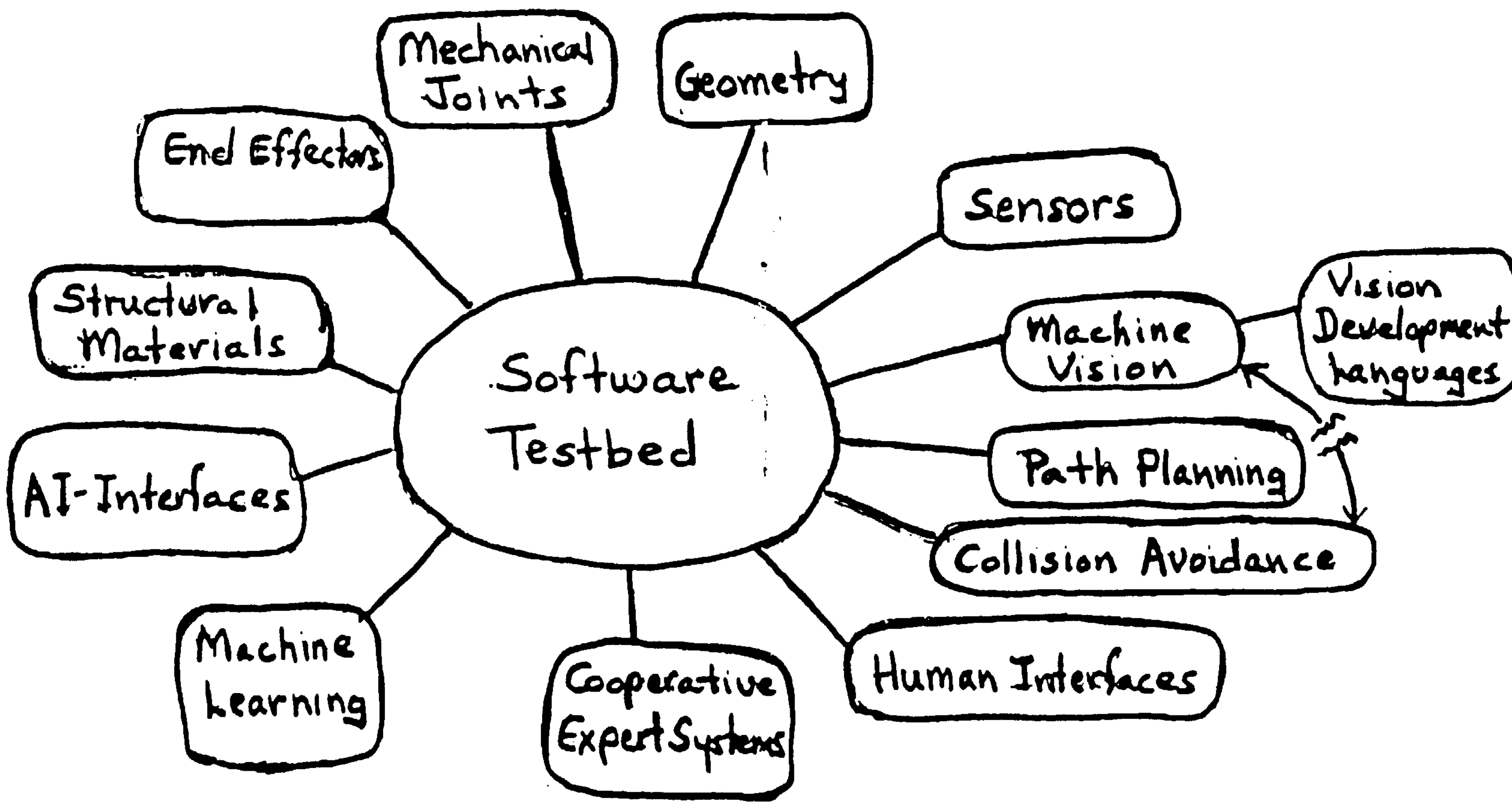
  - COMMON SERVICE AND HARDWARE / SOFTWARE ELEMENTS.

- TO DEFINE BOTH ENABLING AND ENHANCING ROBOTICS AND AUTOMATION TECHNOLOGIES NEEDED IN SUPPORT OF THE NATIONAL SPACE TRANSPORTATION SYSTEM.

- TO DEVELOP AN UNDERSTANDING OF THE TRADE - OFFS AND COMPLEMENTARY RELATIONSHIPS BETWEEN GRAPHICAL SIMULATION MODULES AND TESTBED ITEMS WITH RESPECT TO DESIGN, DEVELOPMENT, AND PROOF - OF - CONCEPT DEMONSTRATIONS.

ROBOTICS SOFTWARE SIMULATION TESTBED

- INTEGRATE A TESTBED WHICH INCORPORATES
  - COMPUTER VISION
  - ARTIFICIAL INTELLIGENCE INTERFACES
  - GRAPHICS
  - VOICE RECOGNITION AND NATURAL LANGUAGES
  - END EFFECTORS/SENSORS
  - MECHANICAL JOINTS
- DEFINE A PARALLEL ARCHITECTURE WHICH WILL PERMIT THE SUCCESSFUL INTEGRATION OF THESE COMPONENTS FOR REAL-TIME OPERATIONS
- DEVELOP AN UNDERSTANDING OF THE COMPUTING, INTERFACING AND CONTROL CAPABILITIES OF A SPACE BASED ROBOTICS SYSTEM


Software Testbed
Mechanical Joints
Geometry
End Effectors
Sensors
Structural Materials
Machine Vision
Vision Development Languages
AI-Interfaces
Path Planning
Collision Avoidance
Machine Learning
Cooperative Expert Systems
Human Interfaces

COMPONENTS:

GEOMETRIES
END EFFECTORS
MECHANICAL JOINTS
MOBILITY SYSTEMS
MATERIALS
PATH PLANNING
COLLISION AVOIDANCE
VISION SYSTEMS / ALGORITHMS

OTHER SENSOR SYSTEMS
- LASER
- FORCE / TORQUE FEEDBACK
- RADAR

ARTIFICIAL INTELLIGENCE INTERFACES
FAULT DIAGNOSIS / RECONFIGURATION

COOPERATIVE EXPERT SYSTEMS
MACHINE LEARNING

HUMAN - MACHINE INTERFACES
- SWITCHES
- VOICE RECOGNITION
- BALL
- JOY - STICK
- MASTER - SLAVE ARMS
- ANTHROPOMORPHIC HANDS

PROJECT TITLE: RESEARCH ON APPLICATIONS OF FUZZY SET AND RELATED THEORY TO FAILURE DETECTION AND CONTROL IN SPACE SYSTEMS

PRINCIPAL INVESTIGATOR: PROFESSOR THOMAS SHERIDAN, MASSACHUSETTS INSTITUTE OF TECHNOLOGY

OBJECTIVE:

- EVALUATE AND DEVELOP NON-DETERMINISTIC SET TECHNIQUES FOR DETECTION AND DIAGNOSIS OF FAILURES IN SPACE SYSTEM GN&C
- EXPLORE DECISION CRITERIA APPROPRIATE TO CONTROL, SYSTEM RECONFIGURATION AND REDUNDANCY MANAGEMENT
- PERFORM ANALYTICAL AND SIMULATION STUDIES OF SENSITIVITY OF FAILURE DETECTION TO MODELING IMPERFECTIONS, DATA IMPRECISION OF REAL-TIME RESPONSE. COMPARE FUZZY SET TECHNIQUES WITH OTHER FAILURE DETECTION/DIAGNOSIS TECHNIQUES.

STATUS:

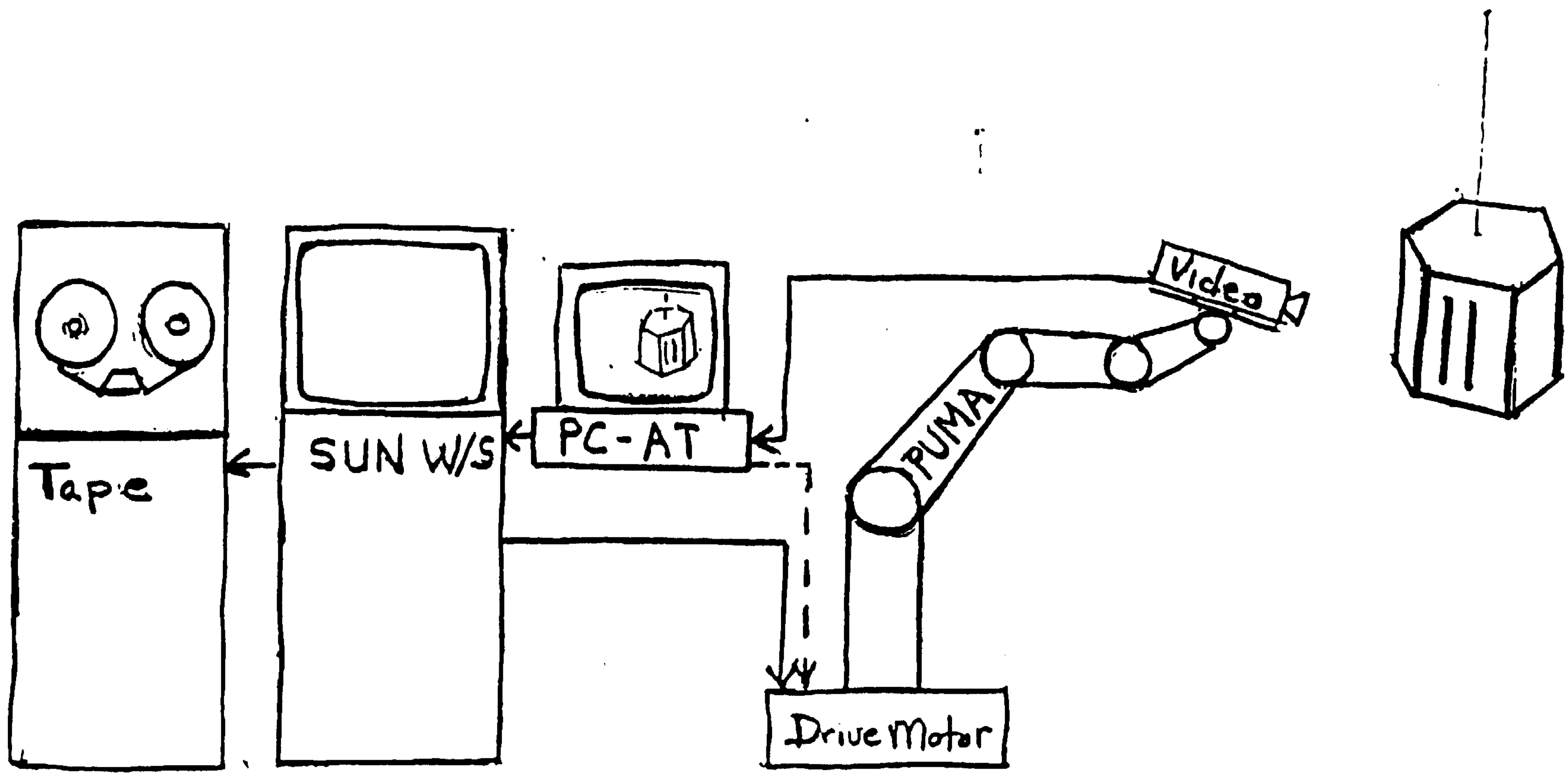
- THE TEST VEHICLE HAS BEEN SELECTED (HUBBLE SPACE TELESCOPE), AND THE GN&C AND POINTING SYSTEMS MODELS ARE BEING COLLECTED
- THE NON-DETERMINISTIC SET CRITERIA ARE BEING DEFINED

- PROJECT TITLE: COLLISION-FREE PLANNING/ROBOTICS SOFTWARE TESTBED ARCHITECTURE (I)
- PRINCIPAL INVESTIGATOR: PROFESSOR KANG SHIN, UNIVERSITY OF MICHIGAN
- OBJECTIVE: DEVELOP DYNAMIC MODELS FOR COLLISION-FREE PATH PLANNING FOR ROBOT MANIPULATORS
    - SELECTION OF INTERMEDIATE POSITIONS
    - PATH SEGMENT GENERATION
    - PATH OPTIMIZATION WITH RESPECT TO TIME, SAFETY
- STATUS: A SERIES OF COLLISION AVOIDANCE AND PATH OPTIMIZATION ALGORITHMS HAVE BEEN DEVELOPED AND TESTED
    - COORDINATION OF DUAL ROBOT ARMS USING KINEMATIC REDUNDANCY
    - COOPERATIVE CONTROL OF A ROBOT AND A POSITIONING DEVICE
    - VARIATIONAL DYNAMIC PROGRAMMING FOR OPTIMIZATION
    - SELF-TUNING PREDICTED CONTROL FOR ROBOT TRAJECTORY TRACKING

- PROJECT TITLE: COLLISION-FREE PATH PLANNING/ROBOTICS SOFTWARE TESTBED ARCHITECTURE (II)
- PRINCIPAL INVESTIGATOR: PROFESSOR RICHARD VOLZ, UNIVERSITY OF MICHIGAN
- OBJECTIVE: DEVELOP THE ARCHITECTURAL FRAMEWORK FOR THE NASA/MPAD ROBOTICS SOFTWARE TESTBED
    - CHARACTERIZE THE SOFTWARE PACKAGES, FUNCTIONS AND USAGES
    - PARALLEL PROCESSORS/REAL-TIME OPERATIONS
    - DISTRIBUTED LANGUAGES AND DATABASES
    - INTEGRATION CONSIDERATIONS
- STATUS: A SERIES OF ARCHITECTURES AND TESTBED ITEMS AT UNIVERSITY OF MICHIGAN HAVE BEEN EXAMINED
    - LANGUAGE DEFINITION
    - DISTRIBUTED TIME MANAGEMENT
    - PERFORMANCE EVALUATION
    - IMPLEMENTATION AND TRANSLATION STRATEGIES
    - PARALLEL PROCESSING ON HYPERCUBE MACHINES

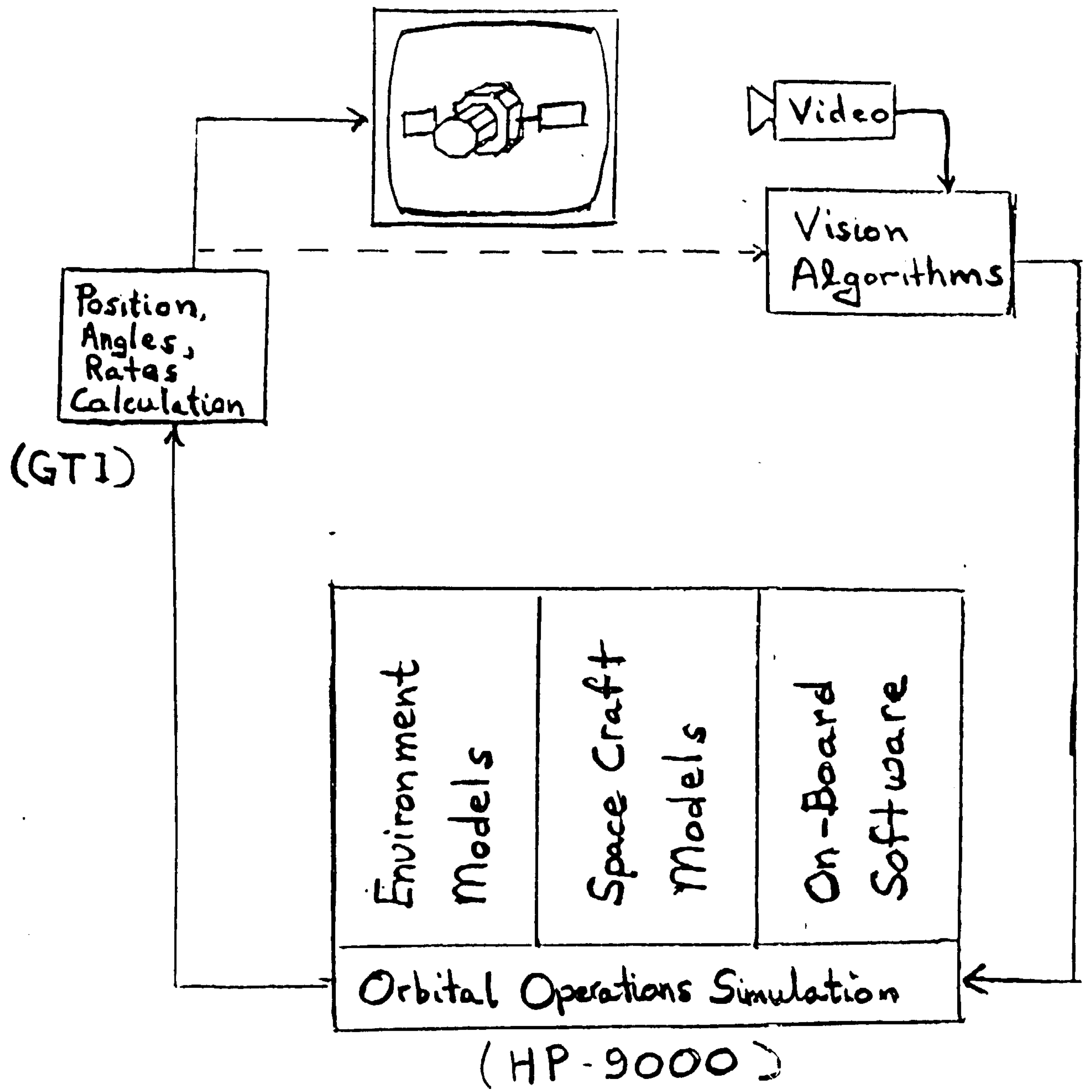
- PROJECT TITLE: A COMPUTER GRAPHICS TESTBED TO SIMULATE AND TEST VISION SYSTEMS FOR SPACE APPLICATIONS
- PRINCIPAL INVESTIGATOR: PROFESSOR JOHN CHEATHAM, RICE UNIVERSITY
- OBJECTIVES:
  - CREATE A COMPUTER GRAPHICS TESTBED FOR MACHINE VISION ALGORITHMS RELATING TO SPACE ROBOTICS ACTIVITIES
  - PROVIDE VIDEO IMAGES OF A MODEL ILLUSTRATING THE FEEDBACK CONTROL OF A ROBOT ARM
  - DEVELOP MACHINE VISION ALGORITHMS FOR SPACE APPLICATIONS USING A COMMERCIALLY AVAILABLE VISION DEVELOPMENT LANGUAGE
- ACCOMPLISHMENTS:
  - PHYSICAL MODELS OF SATELLITE SERVICING OPERATION, USING MULTIPLE SENSORS TO CONTROL A ROBOT ARM
  - CORRESPONDING MODELS DEVELOPED OR BEING DEVELOPED USING A SUN COLOR GRAPHICS WORKSTATION
  - DEVELOPMENT OF LOW-LEVEL VISION ALGORITHMS USING THE 3M CORPORATION VDL SYSTEM

- PLANS:
- CONTINUATION AND EXTENSION OF PAST WORK, INCLUDING:
  - ADDITIONAL PHYSICAL MODELS USING THE PUMA 560 ROBOT TO SIMULATE THE OMV AND A UMI/RTX ROBOT TO CONTROL THE POSITION OF A TARGET SATELLITE
  - ADDITIONAL SENSOR DEVICES, INCLUDING FORCE/MOMENT AND PROXIMITY SENSORS
  - GRAPHICS SOFTWARE TO INCLUDE SHADOWS, MULTIPLE LIGHT SOURCES, PHONG SHADING, AND IMPROVED CONTROL OF AMBIENT, DIFFUSE, AND SPECULAR LIGHT SIMULATIONS, AND BETTER CONTROL OF MODEL POSITION AND ORIENTATION
  - INTERMEDIATE LEVEL VISION ALGORITHMS, USING THE 3M VISION DEVELOPMENT LANGUAGE, IE. OBJECT RECOGNITION
  - USE OF THREE CAMERAS TO CONTROL THE ROBOT OPERATIONS


Tape
SUN W/S
PC-AT
PUMA
Video
Drive Motor

- PROJECT TITLE: DEMONSTRATION OF A 3-D VISION ALGORITHM FOR SPACE APPLICATIONS
- PRINCIPAL INVESTIGATOR: PROFESSOR RUI DE FIGUEIREDO, RICE UNIVERSITY
- OBJECTIVES:
  - MODIFY EXISTING VISION ALGORITHMS FOR SPACE APPLICATIONS
  - PERFORM A DEMONSTRATION OF THESE ALGORITHMS USING BOTH COMPUTER GRAPHICS AND ACTUAL ROBOT MANIPULATORS
  - DELIVER SOFTWARE FOR THESE ALGORITHMS TO THE MPAD GRAPHICS LABORATORY
- ACCOMPLISHMENTS:
  - MODIFICATIONS TO THE ALGORITHMS HAVE BEEN COMPLETED AND TESTED AT RICE UNIVERSITY
  - DEMONSTRATIONS OF THE MODIFIED ALGORITHMS HAVE BEEN PERFORMED AT RICE UNIVERSITY
  - SOFTWARE HAS BEEN DELIVERED TO LINCOM CORPORATION FOR INCLUSION IN THE ROBOTICS SOFTWARE TESTBED

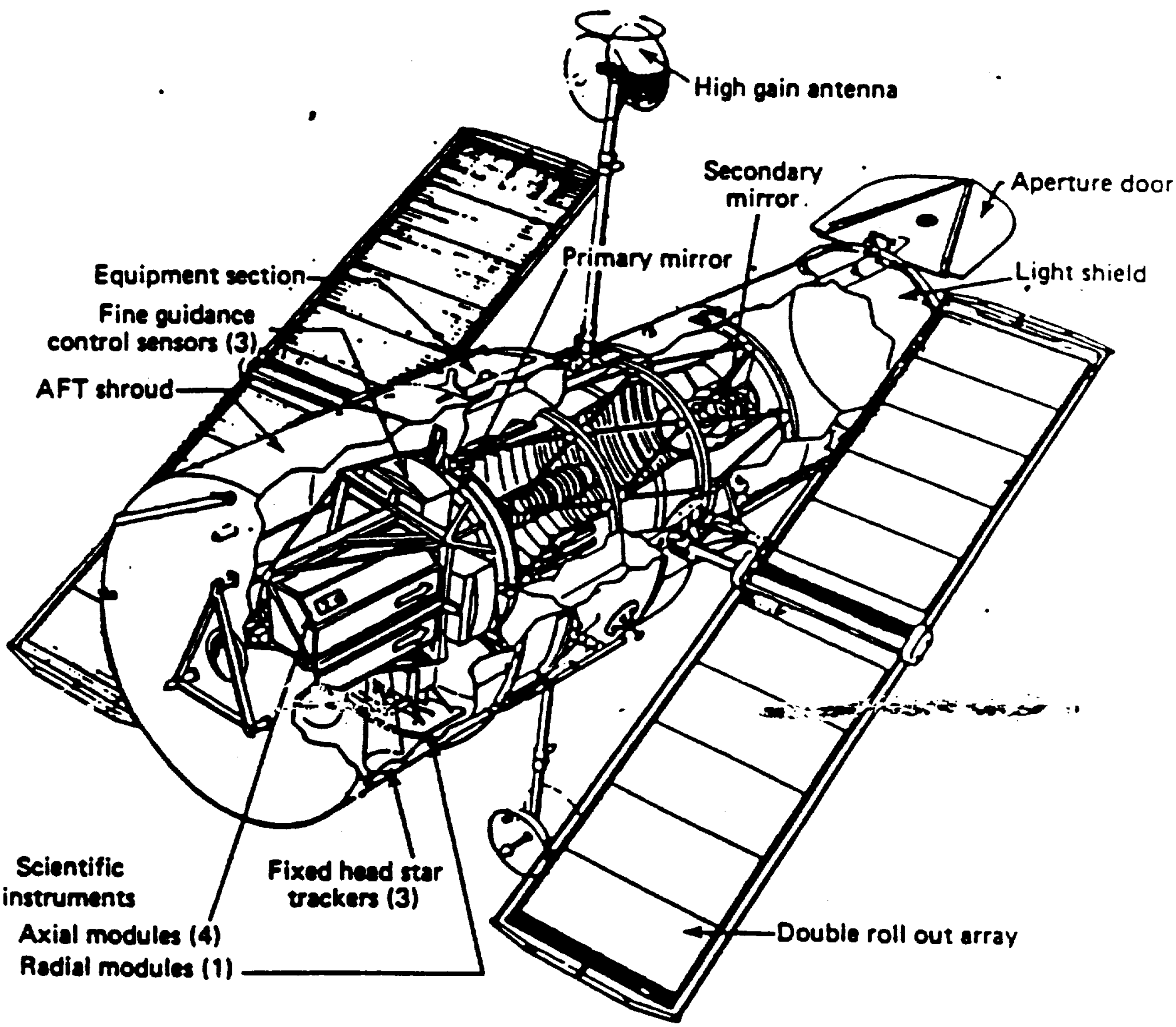
- PROJECT TITLE: SIMULATION OF ROBOTICS SPACE OPERATIONS
- PRINCIPAL INVESTIGATOR: DR. YASHVANT JANI, LINCOM CORPORATION
- OBJECTIVE: INTEGRATE THE ALGORITHMS INTO THE ROBOTICS SOFTWARE TESTBED GRAPHICS SYSTEM
  - DEFINE REQUIREMENTS FOR VISION SENSOR SIMULATION AND RELATED ROBOTICS OPERATIONS
  - IMPLEMENT SENSOR ALGORITHMS IN THE OOS AND ASSOCIATED GRAPHICS SOFTWARE
  - VALIDATE THE VISION SENSOR ALGORITHMS WITH THE RESPECT TO THE OOS AND THE RELATED ROBOTICS SIMULATIONS
- STATUS:
  - SOFTWARE PACKAGES HAVE BEEN RECEIVED FROM THE RICE UNIVERSITY TASKS, AND ARE BEING PREPARED FOR INCLUSION IN THE OOS
  - CODE CHANGES IN THE OOS HAVE BEEN COMPLETED TO ACCOMMODATE THE NEW ALGORITHMS



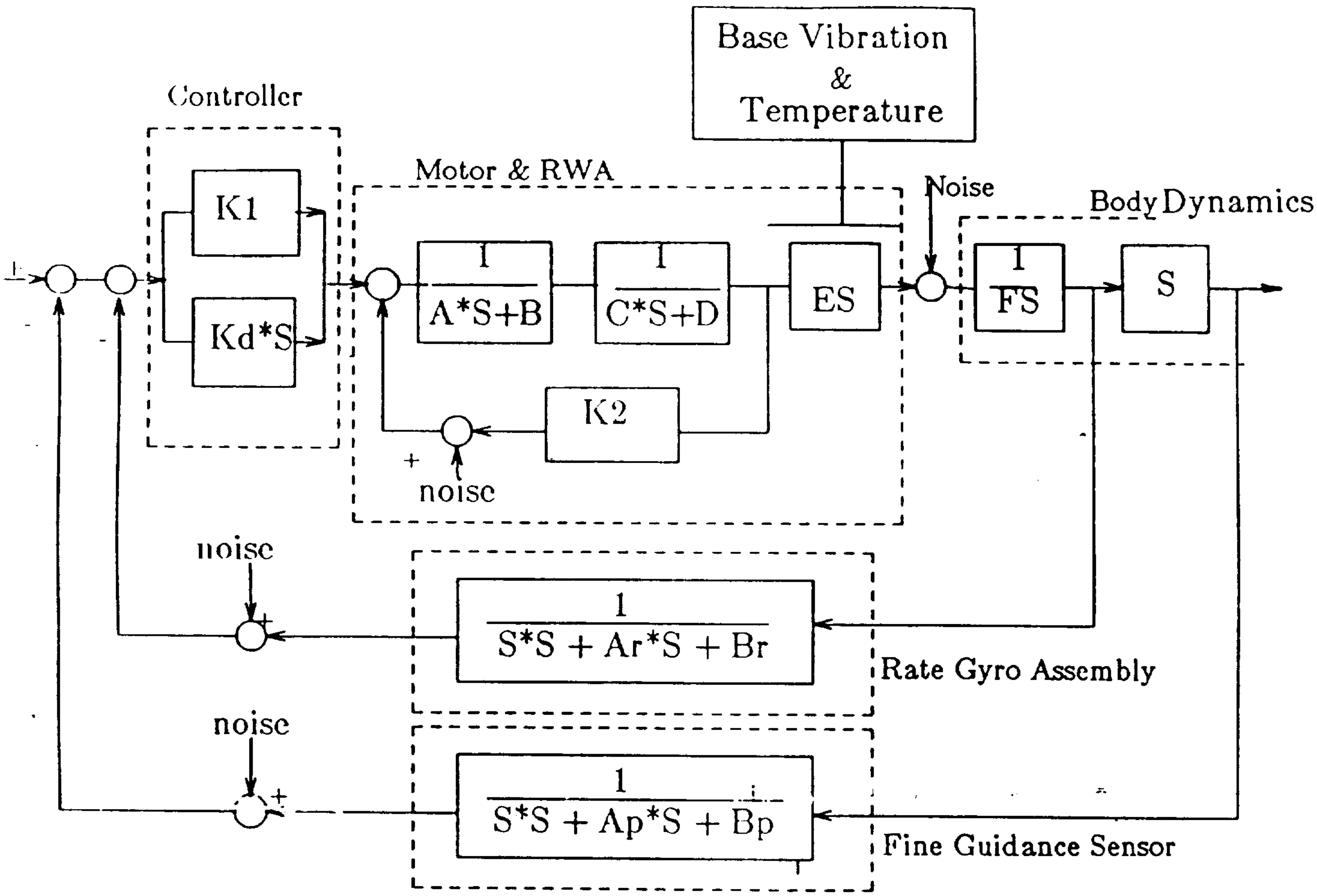
MSD
Video
Vision Algorithms
Position, Angles, Rates Calculation
(GTI)
Environment Models
Space Craft Models
On-Board Software
Orbital Operations Simulation
(HP-9000)

Thomas Sheridan

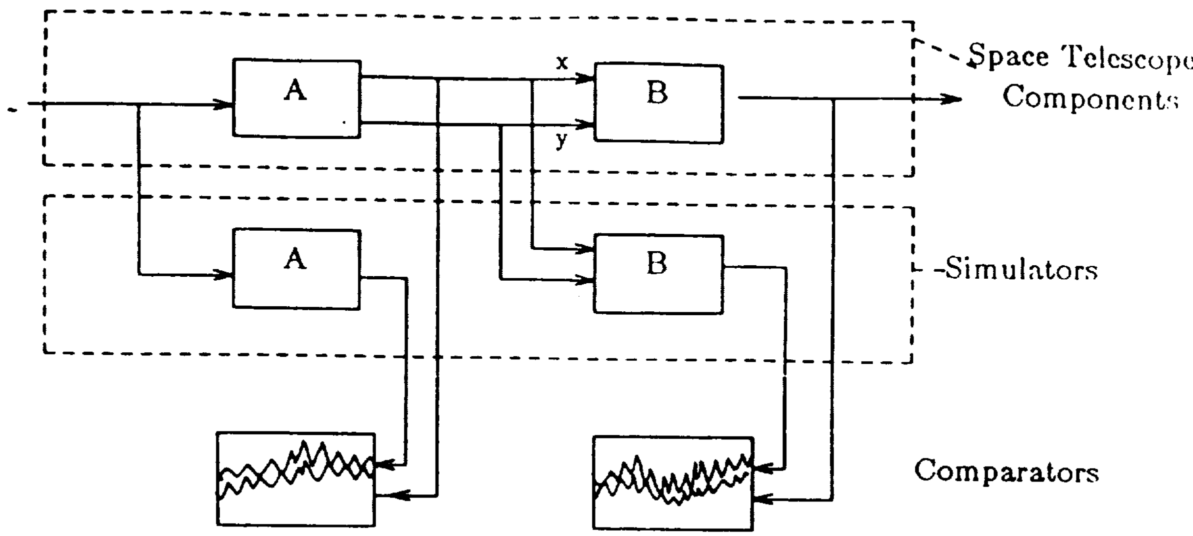
MIT



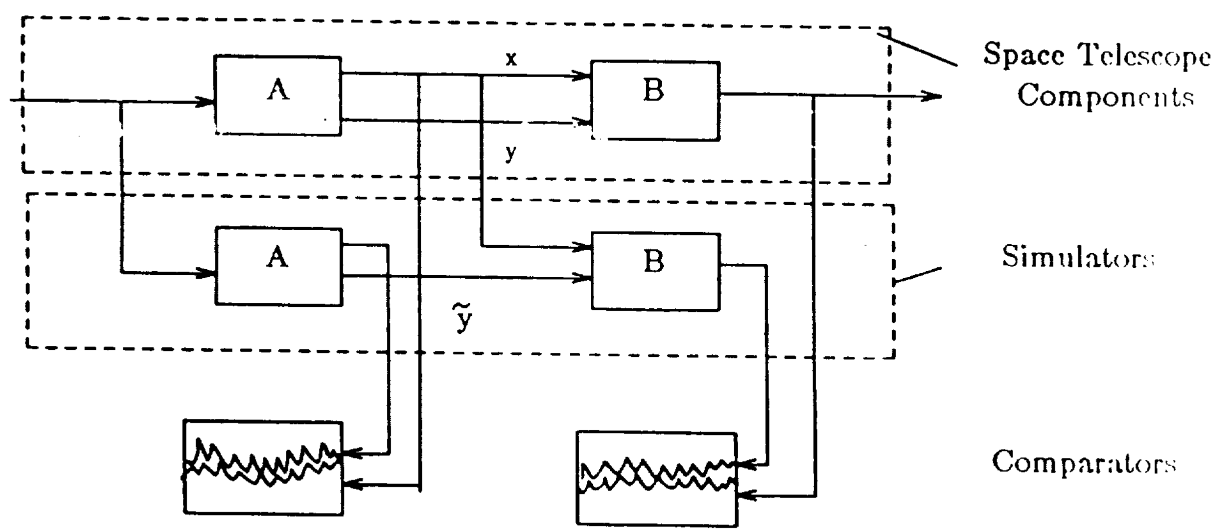
ST Antenna Pointing System



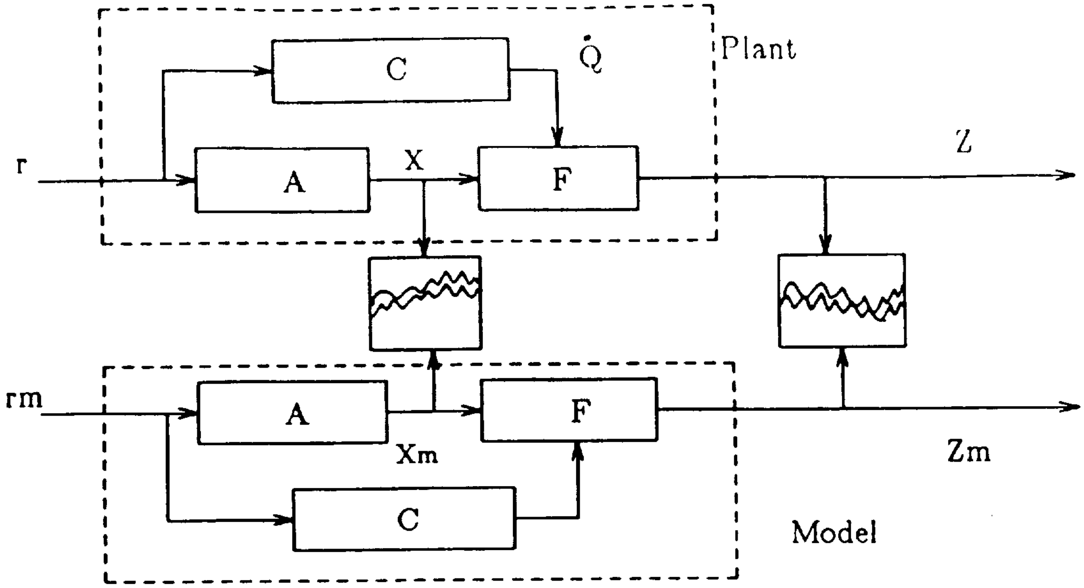
Space Telescope Attitude Control Simulation



All Variables are measureable.



Variable ỹ is not measureable.

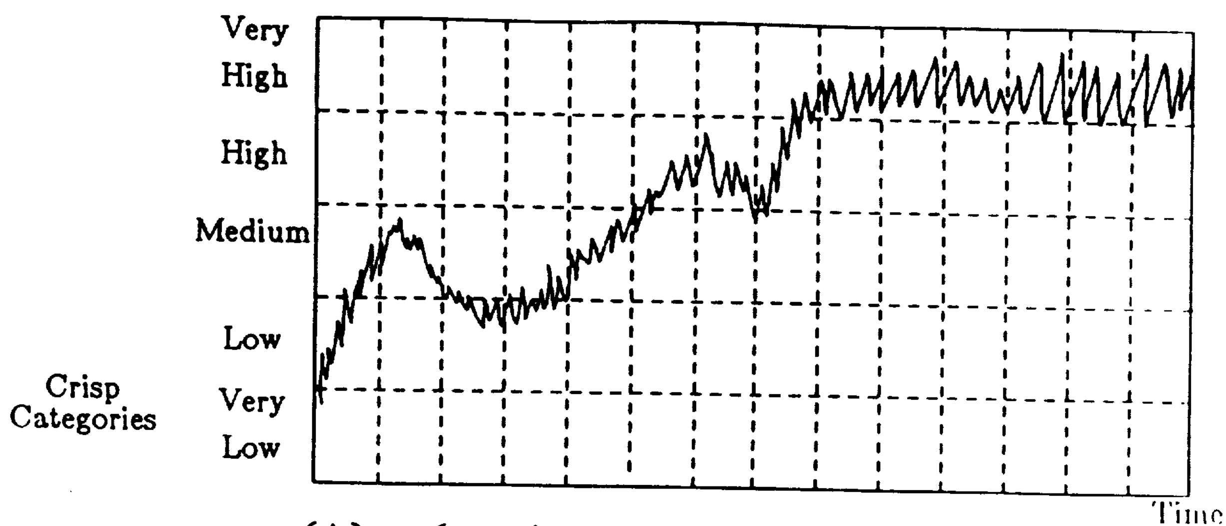


Inadquate Measurements

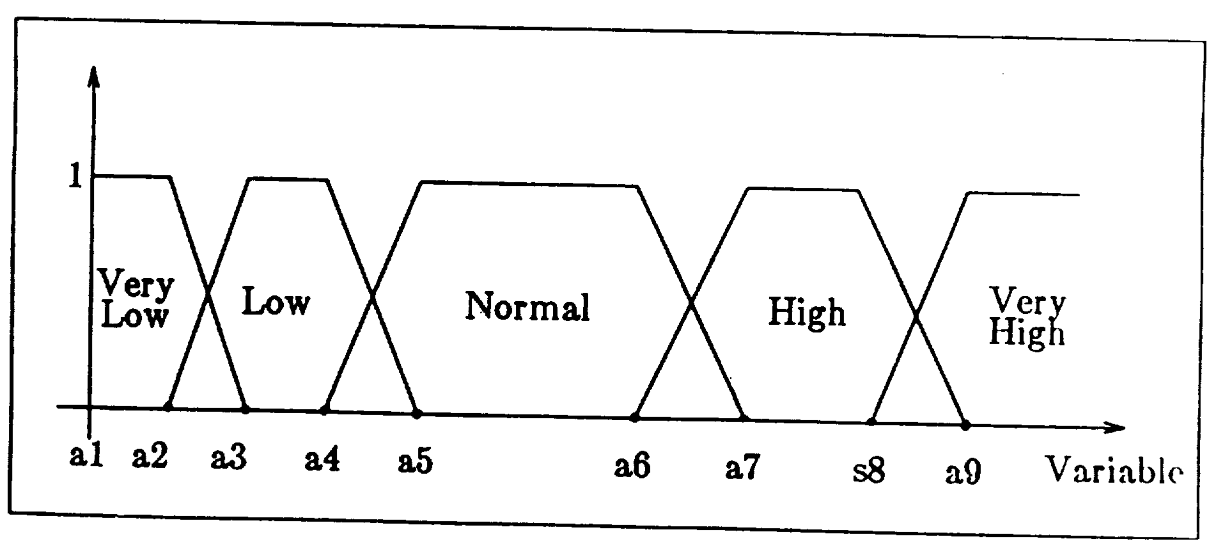
$$\{F\} = \{\,0.1/1,\ 0.3/2,\ \overset{0.5\text{ - Cut}}{|}\ 0.7/3,\ 0.9/4,\ 1.0/5\}$$

$$\{F\} = \{0/1,\ 0/2,\ 1/3,\ 1/4,\ 1/5\}$$

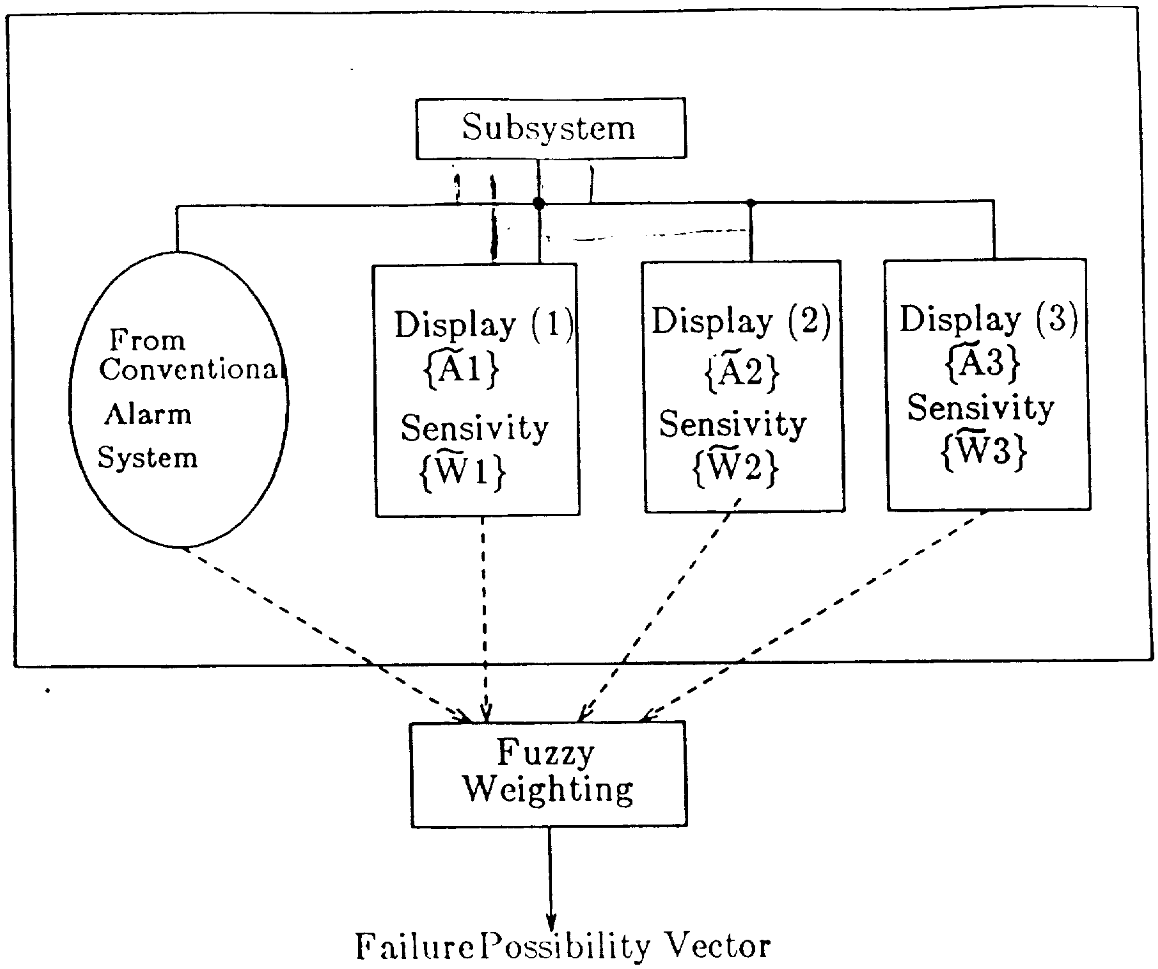
Fuzzification of Discrypancy
for Given Time Period



{A} = { 0.1/1, 0.2/2, 0.2/3, 0.8/4, 1.0/5}



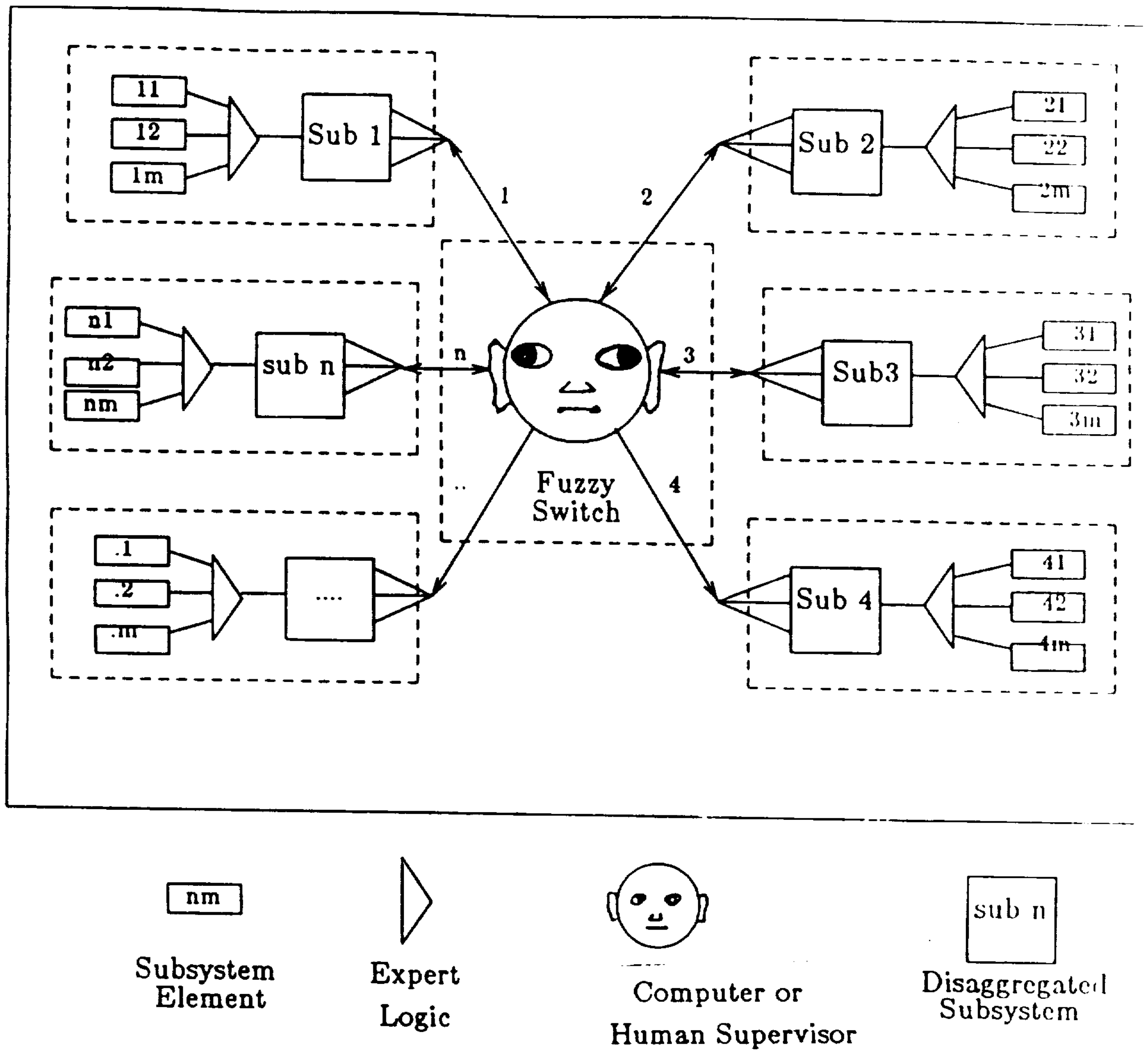
Fuzzy Membership for Unmodelable Variables



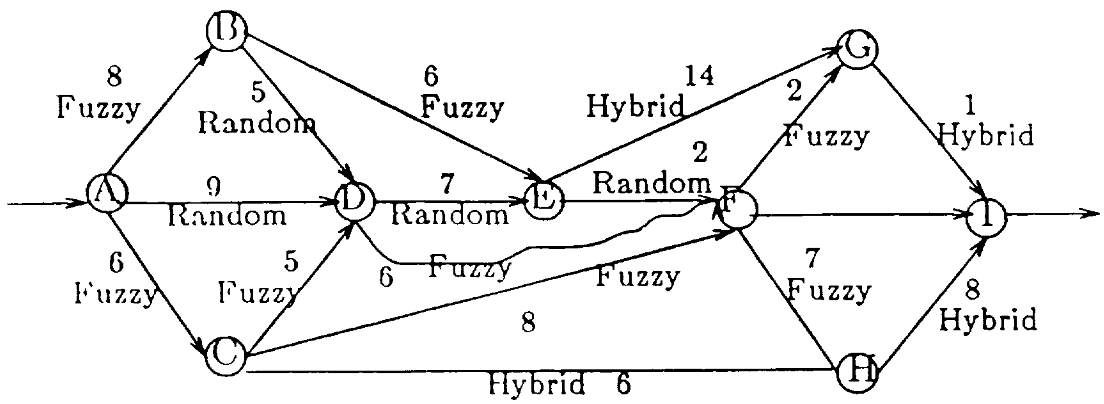
$$\left\{\tilde{F}_0\right\} = \frac{\sum_{i=1}^{i=4}\left\{\tilde{W}_i\right\} * \left\{\tilde{A}_i\right\}}{\sum_{i=1}^{i=4}\left\{\tilde{W}_i\right\}}$$

Failure Possibility for A Subsystem

Fuzzy Expert System



1. System disaggregated into n subsytems.
2. At each node, the failure possibility is computed via fuzzy sets.
3. When failure possibility is high, fuzzy switch triggers expert system for details.



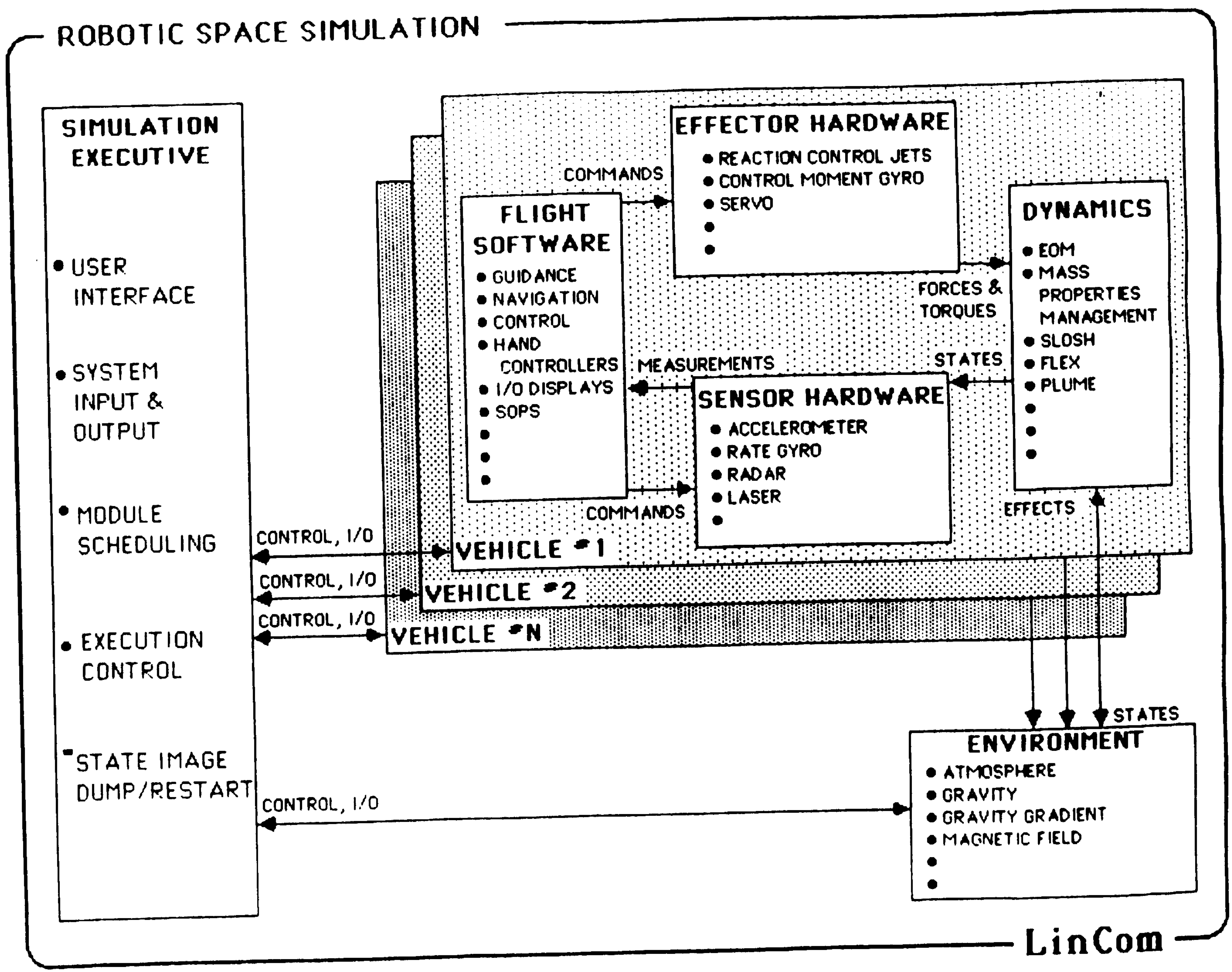
Optimization of Reconfiguration
Based on Fuzzy and Random Numbers

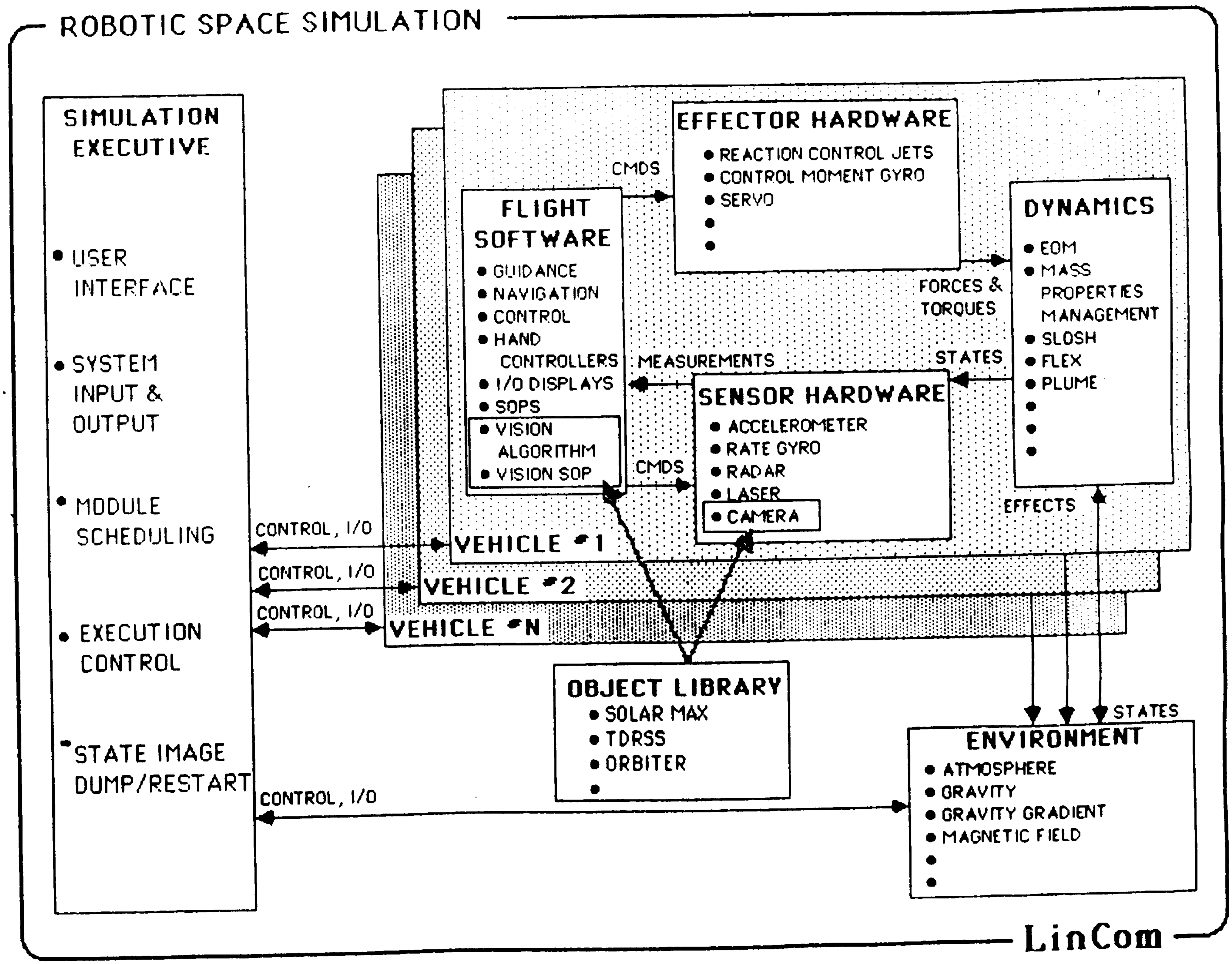
USE OF SIMULATION

- PRE-FLIGHT ANALYSIS
  - DEFINITION OF MISSION REQUIREMENTS
  - PERFORMANCE ENVELOPES
  - FLIGHT ASSESSMENT
- DEVELOPMENT OF MISSION SCENARIOS
  - OPERATIONS
  - PROCEDURES
  - INTEGRATION OF SEVERAL VEHICLES AND SUBSYSTEMS INTO A COORDINATED SCENARIO
- INTRODUCTION OF NEW VEHICLES / SUBSYSTEMS
  - SPECIFICATION AND ANALYSIS
  - SUBSYSTEMS REQUIREMENTS ANALYSIS

INTEGRATION OF ROBOTICS/VISION ALGORITHMS INTO AN ORBITAL OPERATIONS SIMULATION

- TESTBED REQUIREMENTS
  - MODULARITY
  - RAPID PROTOTYPING
  - FIDELITY
- ROBOTICS COMPONENTS IN OOS
  - VISION
  - REMOTE MANIPULATOR SYSTEM (RMS)
  - AUTOMATED FLIGHT / EXPERT SYSTEMS


ROBOTIC SPACE SIMULATION
SIMULATION EXECUTIVE
• USER INTERFACE
• SYSTEM INPUT & OUTPUT
• MODULE SCHEDULING
• EXECUTION CONTROL
• STATE IMAGE DUMP/RESTART
CONTROL, I/O
CONTROL, I/O
CONTROL, I/O
CONTROL, I/O
FLIGHT SOFTWARE
• GUIDANCE
• NAVIGATION
• CONTROL
• HAND CONTROLLERS
• I/O DISPLAYS
• SOPS
COMMANDS
EFFECTOR HARDWARE
• REACTION CONTROL JETS
• CONTROL MOMENT GYRO
• SERVO
FORCES & TORQUES
DYNAMICS
• EOM
• MASS PROPERTIES MANAGEMENT
• SLOSH
• FLEX
• PLUME
STATES
MEASUREMENTS
SENSOR HARDWARE
• ACCELEROMETER
• RATE GYRO
• RADAR
• LASER
COMMANDS
EFFECTS
VEHICLE #1
VEHICLE #2
VEHICLE #N
STATES
ENVIRONMENT
• ATMOSPHERE
• GRAVITY
• GRAVITY GRADIENT
• MAGNETIC FIELD
LinCom



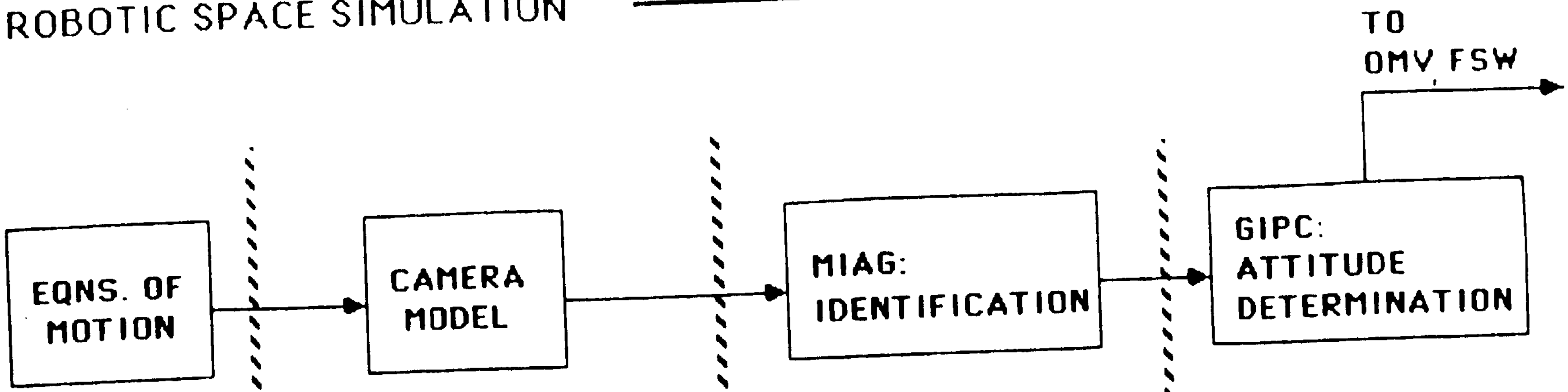
ROBOTIC SPACE SIMULATION
SIMULATION EXECUTIVE
• USER INTERFACE
• SYSTEM INPUT & OUTPUT
• MODULE SCHEDULING
• EXECUTION CONTROL
• STATE IMAGE DUMP/RESTART
CONTROL, I/O
CONTROL, I/O
CONTROL, I/O
CONTROL, I/O
FLIGHT SOFTWARE
• GUIDANCE
• NAVIGATION
• CONTROL
• HAND CONTROLLERS
• I/O DISPLAYS
• SOPS
• VISION ALGORITHM
• VISION SOP
CMDS
EFFECTOR HARDWARE
• REACTION CONTROL JETS
• CONTROL MOMENT GYRO
• SERVO
FORCES & TORQUES
DYNAMICS
• EOM
• MASS PROPERTIES MANAGEMENT
• SLOSH
• FLEX
• PLUME
STATES
MEASUREMENTS
SENSOR HARDWARE
• ACCELEROMETER
• RATE GYRO
• RADAR
• LASER
• CAMERA
CMDS
EFFECTS
VEHICLE #1
VEHICLE #2
VEHICLE #N
OBJECT LIBRARY
• SOLAR MAX
• TDRSS
• ORBITER
STATES
ENVIRONMENT
• ATMOSPHERE
• GRAVITY
• GRAVITY GRADIENT
• MAGNETIC FIELD
LinCom

CURRENT EFFORT

INTEGRATION OF VISION ALGORTHMS WITH ORBITAL MANUVERING VEHICLE (OMV) MODEL

- VISION ALGORITHMS FROM RICE UNIVERSITY
  - OBJECT IDENTIFICATION
    - MOMENT INVARIANT/ATTRIBUTED GRAPH (MIAG):
  - ATTITUDE DETERMINATION
    - GENERALIZED IMAGE POINT CORRESPONDENCE (GIPC):
    - MIAG EXTENSION (TENSORS)
- OMV MODEL
  - RIGID BODY DYNAMICS
  - REACTION CONTROL SYSTEM (RCS) JETS
  - OMV FLIGHT SOFTWARE (CONTROL SYSTEM, GUIDANCE, ETC)
  - CAMERA MODEL
    - FOCAL LENGTH , RANGE , FIELD OF VIEW
    - EXTRACTION OF 2D WIREFRAME (LOW-LEVEL IMAGE PROCESSING)

ROBOTIC SPACE SIMULATION



VEHICLE DYNAMICS	SENSORS	FSW	FSW
- PROPAGATES EQUATIONS OF MOTION OF TARGET AND CHASER (OMV)	- CHECKS IF OBJECT IS WITHIN RANGE AND IN FIELD OF VIEW - CALCULATES THE ORIENTATION OF THE TARGET IN CAMERA FRAME - FUNCTIONAL WIREFRAME EXTRACTION ROUTINE	- MATCHES TARGET WIREFRAME WITH MODELS IN OBJECT LIBRARY	- UNIQUELY IDENTIFIES POINTS USING MAP FROM MIAG - DETERMINES ATTITUDE OF OBJECT IN CAMERA FRAME
OUTPUT: "TRUE VEHICLE STATE"	OUTPUT: WIREFRAME OF ROTATED AND TRANSLATED OBJECT MODEL	OUTPUT: TARGET ID AND COORESPONDENCE MAP BETWEEN TARGET AND CHASER	OUTPUT: ATTITUDE AND RATE OF TARGET

LinCom

CURRENT STATUS

- ALGORITHMS IMPLEMENTATION COMPLETE
  - •• CAMERA MODEL
  - •• FUNCTIONAL WIREFRAME EXTRACTION
  - •• MIAG IDENTIFICATION AND GIPC ATTITUDE DETERMINATION IN OOS
- INTEGRATION TESTING IN PROGRESS
  - •• MODULE INTERFACES COMPLETE
  - •• NEW EVENT-DRIVEN OMV SEQUENCER GENERATED
- TEST CASE DESCRIPTION
  - •• THREE VEHICLES IN SAME ORBIT
  - •• OMV WITH CAMERA IN LOWER ORBIT
  - •• AS OMV APPROACHES TARGET, THE VISION ALGORITHMS WILL IDENTIFY OBJECT AND COMPUTE ATTITUDE AND ATTITUDE RATES

END OF PRESENTATION

*U.S. GOVERNMENT PRINTING OFFICE:1988 661-008/80691