Copy 23 of 30 copies

AD-A281 918

This Document Contains Page/s Reproduced From Best Available Copy

IDA DOCUMENT D-1047

PROCEEDINGS OF THE SECOND PORTABLE COMMON INTERFACE SET (PCIS) WORKSHOP: INTERFACE TECHNOLOGY ANALYSES (ITA2)

Clyde G. Roby

DTIC ELECTE JUL 15 1994 S B

October 1991

Prepared for
Ada Joint Program Office

This document contains blank pages that were not filmed

94-22202

Approved for public release, unlimited distribution: ~~18 March 1994.~~

94 7 14 051

INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311-1772

IDA Log No. HQ 91-040314

DEFINITIONS

IDA publishes the following documents to report the results of its work.

Reports

Reports are the most authoritative and most carefully considered products IDA publishes. They normally embody results of major projects which (a) have a direct bearing on decisions affecting major programs, (b) address issues of significant concern to the Executive Branch, the Congress and/or the public, or (c) address issues that have significant economic implications. IDA Reports are reviewed by outside panels of experts to ensure their high quality and relevance to the problems studied, and they are released by the President of IDA.

Group Reports

Group Reports record the findings and results of IDA established working groups and panels composed of senior individuals addressing major issues which otherwise would be the subject of an IDA Report. IDA Group Reports are reviewed by the senior individuals responsible for the project and others as selected by IDA to ensure their high quality and relevance to the problems studied, and are released by the President of IDA.

Papers

Papers, also authoritative and carefully considered products of IDA, address studies that are narrower in scope than those covered in Reports. IDA Papers are reviewed to ensure that they meet the high standards expected of refereed papers in professional journals or formal Agency reports.

Documents

IDA Documents are used for the convenience of the sponsors or the analysts (a) to record substantive work done in quick reaction studies, (b) to record the proceedings of conferences and meetings, (c) to make available preliminary and tentative results of analyses, (d) to record data developed in the course of an investigation, or (e) to forward information that is essentially unanalyzed and unevaluated. The review of IDA Documents is suited to their content and intended use.

The work reported in this document was conducted under contract MDA 903 89 C 0003 for the Department of Defense. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

© 1991 Institute for Defense Analyses

The Government of the United States is granted an unlimited license to reproduce this document.

IDA DOCUMENT D-1047

PROCEEDINGS OF THE SECOND PORTABLE COMMON INTERFACE SET (PCIS) WORKSHOP: INTERFACE TECHNOLOGY ANALYSES (ITA2)

Clyde G. Roby

October 1991

Approved for public release, unlimited distribution: 18 March 1994.



INSTITUTE FOR DEFENSE ANALYSES
Contract MDA 903 89 C 0003
Task T-D5-496

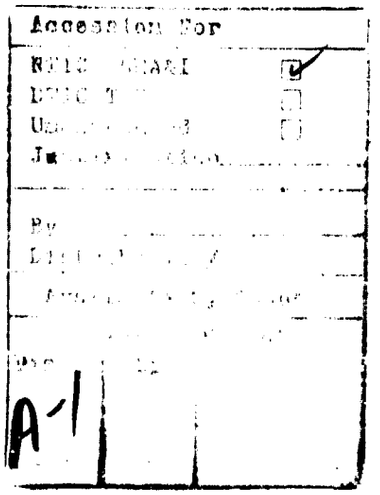
Preface

This document gives an overview of the Second PCIS (Portable Common Interface Set) Workshop, held 3-7 June 1991 in Redondo Beach, CA. It is intended to be used as a major source of technical input to the further refinement of requirements for the International Requirements and Design Criteria (IRAC) for a Portable Common Interface Set, as well as input, both technical and programmatic, for the "way forward" of the PCIS Programme. This document is directed to the participants of the aforementioned Workshop, the PCIS International Experts, and the management team of the PCIS Programme.

This document pertains to the requirements of the Statement of Work, items d. and e., of Task Order T-D5-496 (Amendment No. 5), NATO Special Working Group (SWG) on Ada Programming Support Environments (APSE), which request IDA to

d. "Provide technical assessments concerning SWG on APSE software interface activities to the US Representative on the IRB of the SWG on APSE; these include technical analyses of interface technologies for APSEs, including CAIS-A, PCTE+, and others (e.g., ECMA PCTE)."

e. "Interface with the PCIS Expert Team to perform technical reviews of the products developed by the PCIS Expert Team; provide programme leadership and guidance to members of the PCIS Experts Team and to PCIS Expert Reviewers in the development of these products."



This document contains
blank pages that were
not filmed

EXECUTIVE SUMMARY

The Second PCIS Workshop, in conjunction with the 5th NIST ISEE Workshop and the IWCASE Workshops, was held in Redondo Beach, California from 3-7 June 1991. The workshops were hosted by TRW and the Los Angeles SIGAda and sponsored by the U.S. Department of Defense, the Ada Joint Program Office (AJPO), the National Institute of Standards and Technology (NIST), the U.S. Department of Commerce, the NATO Special Working Group on Ada Programming Support Environments (NATO SWG on APSE), and the International Workshop on CASE (IWCASE).

The primary goals of the PCIS Requirements Validation Phase are to:

- Gather requirements from the software engineering environment (SEE) community,
- Validate requirements defined in the NATO Requirements and Design Criteria (NRAC), and survey available or emerging technologies,
- Produce the International Requirements and Design Criteria (IRAC) and Interface Technology Analysis (ITA) document, and
- Propose a way forward for the PCIS Programme to the NATO SWG on APSE, the PCIS Programme's sponsoring organization.

The workshops were all very important and complementary to the PCIS Programme. The reference model work of the NIST ISEE Workshop is complementary in that it identifies a superset of the services likely to be provided in a PCIS environment framework. It also provides a context in which to compare several existing or emerging systems of interest to the PCIS Programme. The IWCASE Workshop supports the second SWG on APSE goal by providing information about emerging technology relevant to the PCIS Programme.

The goal of the NIST ISEE program is to identify and establish a consensus for U.S. Federal government and industry to take in addressing the need for open system ISEE and software tools interface standards. The current NIST work on software engineering environment issues centers on Workshops on Integrated Software Engineering Environments (ISEE). The goal is to provide guidance to Federal agencies in acquiring an ISEE. The current NIST ISEE Reference Model defines the "concept" of an ISEE in terms of services and dimensions. Most of the items which are defined originated, for the most part, in the ECMA Technical Report, "A Reference Model for Computer Assisted Software Engineering Environment Frameworks," which was approved by ECMA TC33, September 1990. However, some of the definitions have been created by the NIST ISEE Working Group through its extension and modifications to the ECMA Reference Model Document. NIST is planning to publish the "ISEE Reference Model Technical Report," version 1.0, as well as the "Report on Summary of Results of the first NIST ISEE Reference Model Mapping" by the end of September or early October, 1991. NIST will continue the harmonization of joint efforts with ECMA TC33 to develop a standard ISEE for open system environments and will continue the effort of working towards a full ISEE development.

The objectives of the IWCASE Workshop are to update standardization working groups with status of other working group progress and to identify overlapping issues and facilitate coordination on these issues among these groups.

This document contains blank pages that were not filmed

CONTENTS

The Interface Technology Analysis (ITA) Document
Points of Contact
Editor's Remarks

The PCIS Workshops

The NIST ISEE Workshops

The IWCASE Workshops

Final Agenda

Chairman's Opening Remarks for 2nd PCIS Workshop
-- Gary Pritchett, US PCIS Experts Team Leader

NIST Integrated Software Engineering Environments (ISEE) Program: An Overview
-- William Wong, NIST

Portable Common Interface Set (PCIS) Programme: PCIS Workshop
-- Dr. John Solomond, PCIS International Programme Manager

PCIS Requirements Validation: First Workshop
-- John Dawes, European PCIS Experts Team Leader

NIST ISEE / PCIS / IWCASE Workshop: General Welcome
-- Lolo Penedo, TRW

Chairman's Closing Remarks for the Second PCIS Workshop
-- Gary Pritchett, US PCIS Experts Team Leader

Sponsor's Wrap-Up for the Second PCIS Workshop
-- Currie Colket, AJPO

LA SIGAda Presentation (after dinner Wednesday, June 5, 1991):
Goals and Strategies towards Domain-Specific Reuse Based Development
-- Teri F. Payton, Unisys

SECTION I: NIST ISEE Workshop

SECTION II: IWCASE Workshop

SECTION III: PCIS Workshop

Attendance List for Second PCIS Workshop

This Document Contains Page/s
Reproduced From
Best Available Copy

SECTION I -- NIST ISEE Workshop

National Institute for Science and Technology
Integrated Software Engineering Environment
NIST ISEE Workshop

ECMA Organisation
-- Hugh Davis, ICL

Next Generation Computer Resources (NGCR):
Project Support Environment Working Group (PSEWG)
-- Tricia Oberndorf, NADC

Reference Model Mappings

Feedback from AD/Cycle Mapping
-- Bob Ekman *for* B. F. Meyers, IBM

CAIS-A (MIL-STD-1838A) Mapping to NIST Reference Model
-- Geoff Clow, SofTech

Software Life Cycle Support Environment (SLCSE)
-- James Milligan, AFSC Rome Laboratory

CIS Experience in ECMA Reference Model Mapping
-- Hal Pierson, SPL

Summary of Reference Model Evaluation
-- Marvin V. Zelkowitz, NIST

Integration Services
-- Tricia Oberndorf, NADC

Object Management Working Group Summary
-- Lolo Penedo, TRW

Project Management Subgroup Summary
-- Hal Hart, TRW

User Interface Management Services Summary
-- Bob Bagwell, NIST

Summary of the 5th Workshop on Integrated Software Engineering Environments (ISEE)
-- Bill Wong, NIST

SECTION II -- IWCASE Workshop

International Workshop on CASE (IWCASE)
-- David Sharon, CASE Associates

CASE Standards Coordination Update Meeting
Standard Update Reports
-- David Sharon, CASE Associates

CALS, PDES and Software Products
-- Tom Baker, Boeing

CDIF
-- Richard Good, MITRE

PDES STEP CALS
-- Tom Baker, Boeing

IEEE-CS P1175 - A Standard Reference Model for Computing System Tool Interconnection
-- David Sharon, CASE Associates

CIS and ATIS
-- Eric Black, Atherton Technology

Project Support Environment Standards Working Group (PSESWG)
-- Tricia Oberndorf, NADC

The COHESION Framework Program
-- Ed Cuoco, DEC

AD/Cycle Platform: Blueprint for a More Productive Future
-- Bob Ekman, IBM

The SoftBench Experience
-- Dr. Huw Oliver, Hewlett-Packard

Adding Control Integration to PCTE
-- Dr. Huw Oliver, Hewlett-Packard

MIF Control Integration Reactive Integration (Implicit Invocation)
-- Dr. Robert Balzer, USC-ISI

Information Resource Dictionary Systems (IRDS)
-- Dr. James Emerson, RTI [unable to attend; slides in document]

SECTION III -- PCIS Workshop

Needs of the Environment User

Caught in the Minefield (35mm slides) [not included]
-- David Robinson, SD Scicon *for* Mark Gibbons, BT

US Air Force: The Software Technology Support Center
-- Bob Hanrahan, USAF

US Army: Software Engineering Requirements for the Strategic Defense System
-- Jackie Christina, USASDC

Needs of PCTE Programme: A Personal View
-- Hugh Davis, ICL

SEMATECH's Advanced Development Environment (ADE), or
Software Engineering Environments: The Needs of Manufacturing Users
-- Claude Baudoin, SEMATECH

Environments -- An Industry View
-- Leigh Power, MCC

Environment Users Presentation
-- Tim Shorrock, British Aerospace & Dave Robinson, SD Scicon

PCIS Wrap-Up Sessions

Report of the Environment Users Session -- Dr. Tim Lindquist, Arizona State University

Presentations

Process Driven Requirements, evolution and lego interfaces
-- Dr. Vic Stenning, Anshar

Role-based Requirements Analysis
-- John Leary, Martin Marietta

Needs for PCIS Administrative Services
-- Judy Kerner, Aerospace Corp.

The Requirements Process: Technology Push versus User Pull
-- Claude Baudoin, SEMATECH

Stability, Systems Engineering and Benefits
-- Robert Rankin, DRA RSRE, *for* M. Morron

Identifying Methods and Tools
-- Audrey Canning, ERA Technology

Report of the Environment Suppliers Working Group -- Geoff Clow, SofTech

Presentations

ATIS / PCTE Merger Experiment
-- Chris Nolan, DEC

Report of the Platform Providers [merged with Environment Suppliers]
Presentations
PCIS Working Group Needs of Platform Providers
-- Burt Rubenstein, Group Software Technology

Report of the Tool Users / Builders Working Group -- Herman Fischer, Mark V Systems
Presentations
PCIS Needs of the Tool Builders
-- Nicholas Wybolt, Cadre

THE INTERFACE TECHNOLOGY ANALYSIS (ITA) DOCUMENT

This Interface Technology Analysis (ITA) document is divided into three main sections, each corresponding to one of the Workshops described above. An introductory section giving an overview of the PCIS and NIST ISEE activities is included first. Section I contains presentations and summary reports of the NIST ISEE Workshop; section II contains presentations on current standards activities in the IWCASE workshop, and section III contains presentations and summary reports of working groups of the PCIS Workshop. Working groups met in parallel sessions during the PCIS and NIST ISEE Workshops.

POINTS OF CONTACT

A copy of the ECMA Reference Model document, *A Reference Model for Frameworks of Computer-Assisted Software Engineering Environments, ECMA TR/55*, December 1990, and a copy of the NIST Reference Model document, *A Reference Model for Computer Assisted Software Engineering Environment Frameworks, NIST version 1.0e*, May 29, 1991, were distributed as part of the package of materials which everyone received at the Workshop. Free copies of the ECMA Reference Model document can be obtained from:

ECMA
European Computer Manufacturers Association
114 Rue du Rhone
CH-1204 Geneva
Switzerland

Copies of the NIST Reference Model document can be obtained by contacting:

Mr. William Wong
NIST
Bldg. 225, Room B266
Gaithersburg, MD 20899
USA

Additional copies of this, the Interface Technology Analysis (ITA) document can be obtained by contacting (in North America):

Mr. Clyde Roby
Institute for Defense Analyses
1801 North Beauregard Street
Alexandria, VA 22311-1772
USA

or (in Europe):

Mr. Ken Hayter
Defence Research Agency (UK)
RSRE
St. Andrews Rd.
Malvern
Worcestershire
WR14 3 PS
UNITED KINGDOM

Video and Audio tapes of the plenary sessions were made during the meeting. Contact (in North America) Clyde Roby at the address given above, or (in Europe):

Mr. John Dawes
ICL
Eskdale Road, Winnersh
Wokingham
Berkshire RG11 5TT
UNITED KINGDOM

for copies of audio or video tapes.

EDITOR'S REMARKS

The contents of this document reflect the fact that three "working" Workshops were held. As is expected in such workshops, several vugraphs were hand-drawn and many of these were produced in real time in the Working Sessions and Working Groups, so their quality is not always the best. Also, not all slides have the same quality; some were reproduced from the actual vugraphs, others were reproduced from hardcopy. The editor appreciates the hard work in the preparation of all materials present in this document.

The editor is particularly grateful to the chairs of all the working groups and the working sessions of all the workshops for helping make this document possible. In particular, the editor would like to thank Mr. Bill Wong, Mr. Currie Colket, Mr. Gary Pritchett, and Mr. John Dawes for assistance in the final contents of this Interface Technology Analysis (ITA) document.

THE PCIS WORKSHOPS

At the NATO Special Working Group (SWG) on Ada Programming Support Environments (APSE) meeting in San Diego in December 1990, the SWG agreed to pursue an international co-operative programme with the goal of defining a Portable Common Interface Set (PCIS). The fusion of military and civil (commercial) requirements is seen as essential to ensure that PCIS will be a viable standard for next generation environments. Therefore, before the PCIS Definition Programme commences, a Requirements Phase is necessary in order to take into account the needs of the military and civil communities. This Phase has four objectives:

- o The validation of the existing NATO Requirements and Design Criteria (NRAC) and inclusion of non-military requirements.
- o An investigation into what is being provided by existing interface technology and an assessment of the perceived emerging technology over the next five years.
- o A comparison of the results with that which industry is currently providing.
- o An analysis of the differences leading to a prioritization and costing of future work.

Two public workshops were held to support the first two objectives. The First PCIS Workshop was held in the U.K. during the week of 29 April to 3 May 1991. The Second PCIS Workshop was held in California during the week of 3-7 June 1991.

Professor John Buxton, chairman of the First PCIS Workshop, said that one starting point for identifying requirements is to consider the principal aim of software engineering as one of improving the quality of software. In achieving quality, it is essential to link military and commercial requirements for environments -- because successful commercial software is likely to be software that is widely and extensively used, with the result that its defects are more likely to have been detected. As a consequence, the background aim of the PCIS Programme, which is to bring together the requirements of the military and commercial communities, should be regarded as a major step towards achieving quality in itself.

The PCIS Workshops were organized into general sessions and Working Group sessions. The general sessions provided presentations on emerging technology and the needs of special interests. These general sessions were aimed at both the specialist and the non-specialist. The Working Group sessions allowed participants to discuss detailed technology and commercial issues in depth. The Working Group Sessions addressed:

- o Needs of the Tool Builders
- o Needs of the Platform Suppliers
- o Needs of the Environment Suppliers
- o Needs of the Environment Users

The results of the PCIS Workshops are two documents, the Interface Technology Analysis (ITA) document and the Draft International Requirements and Design Criteria (IRAC) document. The ITA is the proceedings of both PCIS workshops.

The IRAC captures the needs of those groups mentioned above. When completed, the IRAC will represent a set of requirements for military and civil technical and programmatic requirements with accompanying rationale. It is denoted *International* to reflect the fact that PCIS is planned to be an international standard encompassing international requirements. The intent is to solicit comments from the public on the IRAC from tool builders, platform suppliers, environments suppliers and environment users. Input from organizations such as ADPESO, AIAA, CBEMA, ECMA, IEEE, IEPG, IWCASE, NGCR, NIST, SIGAda, and STARS/DARPA is also very welcome. A key to the success of the PCIS Programme is to accurately reflect the interface requirements from all segments of integrated software engineering environments communities.

THE NIST ISEE WORKSHOPS

The goal of the NIST ISEE program is to identify and establish a consensus for U.S. Federal government and industry to take in addressing the need for open system ISEE and software tools interface standards. Today, software engineering environments come in many shapes and sizes. They consist of a variety of tools and techniques which assist the software developer. Unfortunately, there are inherent problems in such environments:

1) there are approximately 249 existing activities identified that affect CASE tools;

2) many of these solutions are proprietary;

3) most of them do not support the entire life-cycle;

4) there is no consensus on a reference model or on standard interfaces which define how these essential elements of information can be shared either by the tools in an environment, or by tools across different software engineering environment boundaries; and

5) it is necessary that NIST serves as a "clearinghouse" for coordinating the efforts of key organizations on the establishment of a standardized ISEE, and acts as a neutral forum for discussion of these efforts. This will help reduce the duplication of efforts and redundant initiatives, and establish synergism between the participating groups.

The current NIST work on software engineering environment issues centers on Workshops on Integrated Software Engineering Environments (ISEE).

The objective of these workshops is to identify the coordination needed among key working group and relevant standards activities by:

1) identifying and exploring fundamental problems and issues in ISEE areas;

2) identifying a needed set of standards which define a comprehensive interface for integrating software tools, and developing guidelines on interface standards for an ISEE.

The goal is to provide guidance to Federal agencies in acquiring an ISEE. A series of workshops has been held over the last two years and a Workshop Working Draft was prepared and published for tracking the progress made by each meeting. At present, a core of individuals from government, industry, and academia of the NIST ISEE Working Group, have committed to developing a NIST ISEE Reference Model Technical Report. The current NIST ISEE Reference Model defines the "concept" of an ISEE in terms of services and dimensions. Most of the items which are defined originated, for the most part, in the ECMA Technical Report - "A Reference Model for Computer Assisted Software Engineering Environment Frameworks" which was approved by ECMA TC33, September 1990. However, some of the definitions have been created by the NIST ISEE Working Group through its extension and modifications to the ECMA Reference Model Document in the third and fourth ISEE Workshops.

The NIST ISEE Working Group had conducted the first NIST ISEE Reference Model Mapping Meeting at MCC in Austin, TX in March, 1991. Five existing Software Engineering Environment frameworks were selected and mapped into the developing NIST/ECMA ISEE Reference Model in order to determine the adequacy and completeness of the NIST/ECMA ISEE Reference Model. The summary of this mapping effort will be discussed in this workshop. This workshop will mainly focus on:

1) enhancing the NIST ECMA Reference Model Technical Report;

2) reviewing the results of the NIST ECMA Reference Model mapping exercise; and

3) identifying and defining the services related to integration.

NIST is planning to publish the "ISEE Reference Model Technical Report", version 1.0 as well as the "Report on Summary of Results of the first NIST ISEE Reference Model Mapping" by the end of September or early October, 1991. Finally, future ISEE Workshops and directions will also be discussed. NIST will continue the harmonization of joint efforts with ECMA TC33 to develop a standard ISEE for open system environments and will continue the effort of working towards a full ISEE development.

THE IWCASE WORKSHOPS

IWCASE provides a forum for the update of standardization activities through the IWCASE CASE standards coordination information exchange. The status of each standardization activity represented was presented.

Following the IWCASE information exchange, selected technologies were presented in technical detail as PCIS Emerging Technology. The intent of the Emerging Technology session was to present emerging interface technology that potentially will support production quality environments within the next five years.

Final Agenda
NIST ISEE / PCIS / IWCASE Workshop
3-7 June 1991

Hosts: TRW and Los Angeles SIGAda

Sponsors: U.S. Department of Commerce
National Institute of Standards and Technology (NIST)

U.S. Department of Defense
The Ada Joint Program Office (AJPO)

International Workshop on CASE (IWCASE)

Special Working Group (SWG)
on Ada Programming Support Environment (APSE)

All are invited to participate in all three workshops.

5th NIST ISEE Workshop
June 3-5

Objectives:

- o Enhance the NIST/ECMA Reference Model document.
  - Review and rewrite new services.
  - Identify and define the services related to integration.
- o Review the results of the mapping exercise.
- o Support the PCIS and IWCASE activities.

Results:

- o Revised Reference Model document.

2nd PCIS Workshop
June 3-7

Objectives:

- o Identify and establish the scope of the requirements for PCIS.
- o Assess the requirements of emerging technologies over the next five years.
- o Examine a range of candidate services to serve as the basis for PCIS requirements.
- o Examine a range of candidate technologies that should be leveraged by PCIS.
- o Support the NIST ISEE and IWCASE activities.

Results:

- o Updated version of International Requirements and Design Criteria (IRAC).
- o Updated version of Interface Technology Analysis (ITA).

IWCASE Workshop
June 6

Objectives:

- o Update standardization working groups with status of other working group progress.
- o Identify overlapping issues and facilitate coordination on these issues.
- o Support the PCIS and NIST ISEE activities.

Results:

- o Updated status of standardization activities.

MONDAY, 3 June 1991

11:00 - 13:30	Registration
13:30 - 15:00	NIST & PCIS Overview of Activities
	General Welcome by Host and Chairman
	Overview of the NIST ISEE Program - Bill Wong
	PCIS Programme Overview - Dr. John Solomond
	Review of First PCIS Workshop - John Dawes
15:00 - 15:15	Break
15:15 - 18:00	NIST/ECMA Reference Model - Dr. Anthony Earl

The NIST/ECMA Reference Model will be presented as a tutorial in order to serve as a frame of reference for NIST ISEE and PCIS activities.

18:00	Reception

TUESDAY, 4 June 1991

8:00 - 8:30	Registration
8:30 - 9:00	General Welcome - Lolo Penedo NIST ISEE Workshop Introductory Remarks - Bill Wong ECMA Organization - Hugh Davis NGCR - PSESWG - Tricia Oberndorf
9:00 - 12:00	NIST ISEE Reference Model Mapping Exercise Summary

The mapping exercise is to validate the NIST/ECMA Reference Model. To validate the model, 5 interface technologies have been mapped to the model. These technologies are:

CAIS-A, PCTE, CIS, SLCSE, and AD/Cycle.

12:00 - 13:30	Lunch
13:30 - 17:30	Parallel Tracks for NIST ISEE and PCIS

Track 1: PCIS Needs - A Focus on Needs of the Environment User

Caught in a Mine Field	Dave Robinson (SD SCICON) *for* Mark Gibbons (BT)
Needs of Air Force	Bob Hanrahan (U.S. Air Force)
Needs of SDI	Jackie Christina (U.S. Army)
Needs of ECMA	Hugh Davis (ICL)
Needs of Manufacturers	Claude Baudoin (SEMATECH)
Needs of Platform Suppliers	Burt Rubenstein (Bull)
Needs of Industry	Leigh Power (MCC)
Needs of Industry	Tim Shorrock (British Aerospace) & Dave Robinson (SD SCICON)

A summary of each presentation will be made available to those participating in the NIST ISEE Working Group Sessions.

Track 2: NIST ISEE Working Group Sessions:

1 Object Management	Lolo Penedo
2 Process & Task Management	Hal Hart
3 Interface & Platform Services	Patricia Oberndorf
4 User Interface	Bob Bagwill

These sessions are intended to review and rewrite new services to enhance the NIST/ECMA Reference Model. Discussions will focus on new services as a result of the mapping activity. PCIS Participants are encouraged to support this NIST ISEE activity as it will provide important input into the PCIS Needs Working Groups, especially for the Needs of the Tool Builder, Needs of the Environment Supplier, and the Needs of the Platform Supplier.

WEDNESDAY, 5 June 1991

8:30 - 12:00 Parallel Tracks for NIST ISEE and PCIS

Track 1: PCIS Working Group Sessions:

1 Needs of the Tool Builders

Chairman: Herman Fischer Co-Chairman: Dr. Hans Keus

2 Needs of the Platform Suppliers

Chairman: Bob Munck Co-Chairman: Gérard Memmi

3 Needs of the Environment Suppliers

Chairman: Geoff Clow Co-Chairman: Regis Minot

4 Needs of the Environment Users

Chairman: Dr. Tim Lindquist Co-Chairman: Dr. Vic Stenning

Track 2: 8:30 - 10:45 Continuation of NIST ISEE Working Sessions from Tuesday.

11:00 - 12:00 NIST/ISEE Plenary (Summary of NIST/ISEE Working Groups)

12:00 - 13:30 Lunch

13:30 - 17:30 Parallel Tracks for NIST ISEE and PCIS

Track 1: Continuation of PCIS Working Group Sessions from Morning

Track 2: NIST ISEE Plenary

13:30 - 16:30 Integration Services

The ECMA Reference Model does not completely address integration as a separate service but as activities of other services. Consequently, there are some integration specific services that are not identified in the ECMA Reference Model. The purpose of this session is to address these integration services.

16:30 - 17:30 ISEE Working Group Closing Remarks

This is the final session of the NIST ISEE Workshop. It includes a wrap-up of the ISEE WG activities and will address future workshop direction.

Note: The local chapter of ACM will be hosting a dinner presentation with Teri Payton speaking on national reuse initiatives. Please register by 17:00 Monday.

18:00 Cocktails & Social Hour
19:00 Buffet Dinner
20:00 Program

THURSDAY, 6 June 1991

8:00 - 8:30 Registration

8:30 - 12:00 IWCASE CASE Information Exchange & PCIS Emerging Technology

IWCASE will provide an update of standardization activities through the IWCASE CASE standards coordination information exchange. Dave Sharon will chair the morning session. The status of each standardization activity represented will be presented.

IWCASE Standard Update Reports	Dave Sharon (Chairman)
CALS & PDES	Tom Baker (Boeing)
CDIF	Richard Good
IEEE P1175	Dave Sharon (CASE)
ECMA PCTE	Hugh Davis (ICL)
ATIS	Eric Black (Atherton)
Needs of the Navy NGCR PSESWG	Patricia Oberndorf (U.S. Navy)

Following the IWCASE Information Exchange, selected technologies will be presented in technical detail as PCIS Emerging Technology. The intent of the Emerging Technology Session is to present emerging interface technology that potentially will support production quality environments within the next 5 years. Information Exchange & Emerging Technology Presentations include:

IEEE P1175	Dave Sharon (CASE)
CALS & PDES	Tom Baker (Boeing)

Certain technologies were presented at the First PCIS Workshop and will not be repeated here. These include CAIS-A, CDIF, CFI, and ECMA PCTE.

12:00 - 13:30 Lunch

13:30 - 17:30 Parallel Tracks for Technology Plenary and PCIS Working Groups

Track 1: Continuation of Information Exchange & Emerging Technology Sessions from Morning

ATIS and CIS	Eric Black (Atherton)
COHESION	Ed Cuoco (DEC)
AD/Cycle	Bob Ekman (IBM)
Adding Control Integration to PCTE	Dr. Huw Oliver (HP)
Module Interconnection and Reactive Integration	Dr. Robert Balzer (USC-ISI)
IRDS	Dr. James Emerson (RTI) (unable to attend - slides in proceedings)

Track 2: PCIS Working Group Sessions - WG Sessions of Wednesday afternoon will continue. Some WGs may want to participate in the in the Emerging Technology Session.

FRIDAY, 7 June 1991

9:00 - 12:00 Introduction - Gary Pritchett

Short Presentation by Session Chairmen followed by general discussion
Closing Address - Gary Pritchett
Closing Remarks - Currie Colket

This Document Contains Page/s
Reproduced From
Best Available Copy

NIST ISEE / IWCASE / PCIS

WORKSHOP

PLENARY SESSION

This document contains
blank pages that were
not filmed

Chairman's Opening Remarks for 2nd PCIS Workshop

Mr. Gary Pritchett, US PCIS Expert Team Leader

Welcome all to the 2nd PCIS Requirements Workshop.

I'm pleased that this workshop is being held in conjunction with the 5th NIST ISEE Workshop and the IWCASE Workshop. The activities of these workshops are very important and complementary to the PCIS program. For example, the reference model work being done at the NIST workshop is relevant to the PCIS program, complementary in that it identifies a superset of the services likely to be provided in a PCIS environment framework and provides a context in which to compare several existing or emerging systems of interest to the PCIS program. A rather full week of technical presentations and working group sessions is planned. I encourage you to support as many sessions as you can and expect this to be a very productive and informative workshop.

This is the 2nd of two workshops to validate requirements for the PCIS program. The primary goals of the requirements validation phase is to gather requirements from the SEE community, validate requirements defined in the NRAC, survey available or emerging technologies, produce an IRAC and ITA, and propose a way forward the PCIS program to the NATO Special Working Group on APSE which is the PCIS programme's sponsoring organization.

The requirements workshops are being held to support the requirements validation phase.

In support of the requirements validation, we have planned several technical presentations on the needs of various programmes or organizations. The requirements input is being gathered in the four parallel working sessions held during the workshop. There requirements are being discussed from the following points of view:

- Needs of the environment user
- Needs of the tool builder
- Needs of the environment supplier
- Needs of the platform supplier.

The input captured at these sessions will be the basis for producing the IRAC.

Presentations are being made to the entire workshop on available or emerging technologies. These presentations will provide input to the ITA.

The first workshop, held in London during 29 April through 3 May, had a similar format as this workshop. That workshop was planned and conducted by a European team headed by Wing Cmmdr. Dennis Longdon. They did an excellent job and the workshop was a huge

success. That workshop produced initial requirements input for the IRAC and good presentations were made on available and emerging technologies.

The tasks for this workshop are to: (1) conduct a quick review of the results of the first workshop, then capture any important additional requirements issues so the requirements validation can be completed and the IRAC can be produced and (2) collect additional input on emerging technologies as input for the ITA.

After the workshop is complete, a team of experts, selected from the SEE community in Europe and North America, will analyze the input received at the workshops, reconcile inconsistencies, wordsmith, and produce the final versions of the IRAC and ITA.

I strongly urge you to participate in the working sessions and voice any issues you feel are important for requirements consideration. The issues recorded during these sessions will form the basis for the experts preparation of the PCIS requirements document. Voicing your concerns is the best way to influence the PCIS requirements and programme.

NIST
Integrated
Software Engineering Environments (ISEE)
Program

AN OVERVIEW

William Wong

National Institute of Standards and Technology

U.S. Department of Commerce

June 3-7, 1991

GOAL

Identify and establish consensus directions for U.S. federal government and industry to take in addressing the need for open system ISEE and tools interface standards

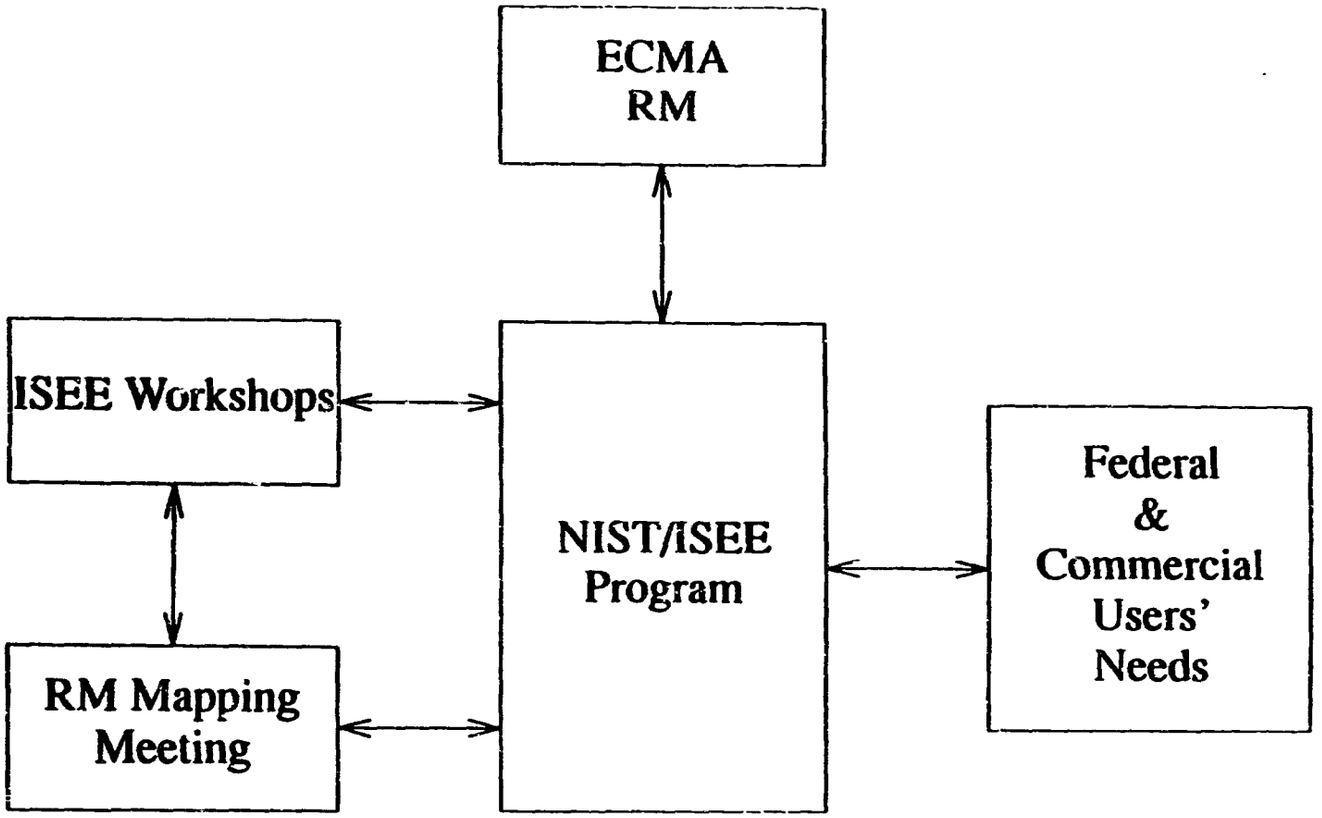
NIST/NCSL Initiatives

* Many existing disjointed activities
  - 249 standards activities identified affecting CASE tools
* Many solutions are proprietary
* Few activities attempt to address the entire software lifecycle
* No consensus on a reference model or standard interfaces
* Need neutral forum for discussion

* ISEE Workshops

FOCUS ON:

- NIST/ISEE Reference Model,
- User Interface,
- Process and Task management,
- Ojective management and Repository,
- Interface and platform services,
- RM Mapping Guidelines,
- ISEE Glossary Definition, and
- Evaluation of existing technologies



Overall NIST ISEE Program

NIST ISEE WG Charter:

* Identify the coordination needed among other key working group and relevant standards activities by:
  - identify and explore fundamental problems and issues in ISEE areas,
  - identify a needed set of standards which define a comprehensive interface for integrating software tools, and
  - developing guidelines on interface standards for an ISEE
* The objective is to provide guidance to Federal agencies in acquiring on ISEE

NIST ISEE Workshop

Chair: William Wong, NIST

NIST ISEE Executive Committee

Bob Bagwill, NIST

Miguel Carrio, Teledyne Brown

Dick Drake, IBM

Peter Fieler, Software Engineering Institute

Hal Hart, TRW

Sandra Mulholland, Rockwell

Tricia Oberndorf, Naval Air Development Center

Lolo Penedo, TRW

Ian Thomas, Hewlett Packard

Bill Wong, NIST

Marv Zelkowitz, U of Maryland and NIST

NIST ISEE Working Groups:

* Reference Model, led by Miguel Carrio, Teledyne Brown
  William Wong, NIST;
* User Interface, led by Bob Bagwill, NIST;
* Process and Task Management, led by Dick Drake, IBM
  Hal Hart, TRW;
* Object Management and Repository, led by
  Maria (Lolo) Penedo, TRW;
* Interface and platform services, primarily focusing
  on integration definition, led by Tricia Oberndorf, NADC;
* RM mapping guidelines, led by Sandra Mulholland,
  Rockwell; and
* Mapping selection criteria and selection of efforts,
  led by Marvin Zelkowitz, UMD and NIST.

ISEE Workshops Participants:

- Federal Agencies:

NIST, US AirForce, US Army, US Navy, DARPA/STARS,
AJPO/PCIS, IRS, HIS, US Senate, VA, NASA, NSF,
US Coast Guard, USDA, HHS, USDT, GSA, CAIS-A,
NIST/APP, SLCSE, NSA, IDA, USAF/STSC, USN/PSESWG

- Industries:

IBM, HP, SUN, TRW, Mitre, Rockwell, MCC, SPC, IEEE,
UNISYS, DEC, IDE, Teledyne Brown, Intermetrics, ATAC,
General Research Corp, Abacus Technology, CADRE, I-logic,
Honeywell, Boeing Aerospace, Geodymanics, Texas Instruments,
AVTEC, Mark V. Systems, MKL Software, EDPNS, SPS, AD/Cycle,
P1175, CDIF, PDES, IRDS, Charles Draper Lab, OMS, CIS, ATIS,
Integrated Systems, Softran, Emerging Technologies Group,
UNIX Adviser, AIT, IWCASE, OODBTG, AIAA, SoftTech

- Academia:

U of Maryland, SEI, U of Houston, Jersey City State College,
Georgetown U, U of VA, Johns Hopkins U, New Jersey Institute
of Technology, George Mason U.

- International

NEC/Japan, Hitachi/Japan, IPA/Japan, Netron/Canada, SPP/Brazil,
STL/UK, HP/UK, ECMA/TC33, Colin Tully Associatea/UK, ECMA,
III/Taiwan, Hong Kong Polytechnic/Hong Kong, PCTE, SIGMA/Japan

* Workshop Products

- NIST ISEE Reference Model Document
  - user interface,
  - process and task management,
  - object management and repository, and
  - interface and platform services
- Mapping Guidelines Document
  - mapping selection criteria and selection of efforts
- ISEE Glossary Definition Document
- Proceedings of the NIST/ISEE Workshops (1-4)

FIRST NIST ISEE Workshop SUMMARY

- Hosted and Sponsored By NIST, May 25, 1989

* Open Architecture Approach
* Strawman Requirements
* Reference Model
* Taxonomy and classification of tools and services
* Interfaces for environment services
* Information Interface Language
* Services/properties provided by an ISEE
* An assessment process to determine how close we are to complete environment
* coordination of related ISEE groups and activities

- Workshop Product

* Workshop Working Draft, July, 1989

SECOND NIST ISEE Workshop SUMMARY

- Hosted By Teledyne Brown Engineering and sponsored by NIST, Dec 5-6, 1989

* Identified and defined of an ISEE Reference Model

* Conducted a NIST ISEE Reference Model survey

* Identified and defined a set of ISEE end-user requirements

* Established synergism between the users, vendors and standards groups

* Promotion of convergence of ISEE standards and interfaces

- Workshop Product

* Workshop Working Draft, May, 1990

THIRD NIST ISEE Workshop SUMMARY

- Hosted By US NAVY and Sponsored by NIST, May 31 - June 1, 1990

* Reviewed Reference Models
* User Interface
* Process and Task Management
* Object Management and Repository
* Summary of the ISEE Survey
* Selected the ECMA RM as the base definition for the development of the NIST ISEE RM Document
* Reviewed the ECMA RM Version 3.0 (ECMA/TC33/TGRM/90/011, May 25, 90)
* Established a collaborative effort with ECMA/TC33 to develop a standardized ISEE RM Techical Report

- Workshop Products

* Comments on the Version 3.0 of the ECMA Reference Model, Version 1, August, 1990
* Workshop Working Draft, September, 1990

FOURTH NIST ISEE Workshop SUMMARY

- Hosted By IBM and Sponsored by NIST, October 11-12, 1990

* User Interface
* Process and Task Management
* Object Management and Repository
* Interface and Platform (Integration)
* RM Mapping Guidelines
* RM Selection Guidelines
  - ECMA PCTE, CAIS-A, CIS, SLCSE, AD/Cycle
* RM Mapping Meeting
  - hosted by MCC, March 12-13, 1990

- Workshop Products

* Summary of the 4th ISEE Workshop, October, 1990
* NIST ISEE RM Subset for Mapping, February, 1991
* NIST ISEE RM Mapping Guidelines, Version 1.2 March, 1991

FIFTH NIST ISEE Workshop SUMMARY

- Hosted By TRW and Los Angeles SIGAda, sponsored by NIST, AJPO, IWCASE, and SWG on APSE, June 3-5, 1991

* Enhance the NIST/ECMA Reference Model Technical Report

* Review the results of RM mapping exercise

* Identify and define the services related to integration

- Workshop Products

  * NIST ISEE Reference Model Technical Report, Version 1.0 (Target date: Sept. 30, 1991)

  * Summary of results of the 1st NIST RM Mapping Report (Target date: Sept. 27, 1991)

Future Workshops and Directions

- 6th Workshop will be hosted and sponsored by NIST, October ?, 1991 in Gaithersburg, MD

- 7th Workshop will be hosted by SEI and sponsored by NIST, May ?, 1992, in Pittsburgh, Penn

- 8th Workshop will be hosted and sponsored by NIST, October ?, 1992 in Gaithersburg, MD

- 9th Workshop will be hosted by STSC/USAF and sponsored by NIST, May ?, 1993

- Harmonize the joint effort with ECMA/TC33 to develop a standards ISEE for Open System Environments

- NIST ISEE effort is working towards an (full) ISEE development

William Wong
NIST/NCSL
Gaithersburg, MD 20899
(301) 975-3341
FAX # (301) 590-0932
wong@swe.ncsl.nist.gov

Portable Common Interface Set (PCIS) Programme

PCIS WORKSHOP

Dr. John Solomond
PCIS International Programme Manager

3 June 1991

OVERVIEW

- Importance of Interface Technology
- History of CAIS-A/PCTE+ Convergence Effort
- PCIS Programme
- ECMA, STARS, and NIST Participation
- SWG on APSE Tasking/Status

Importance of Interface Technology

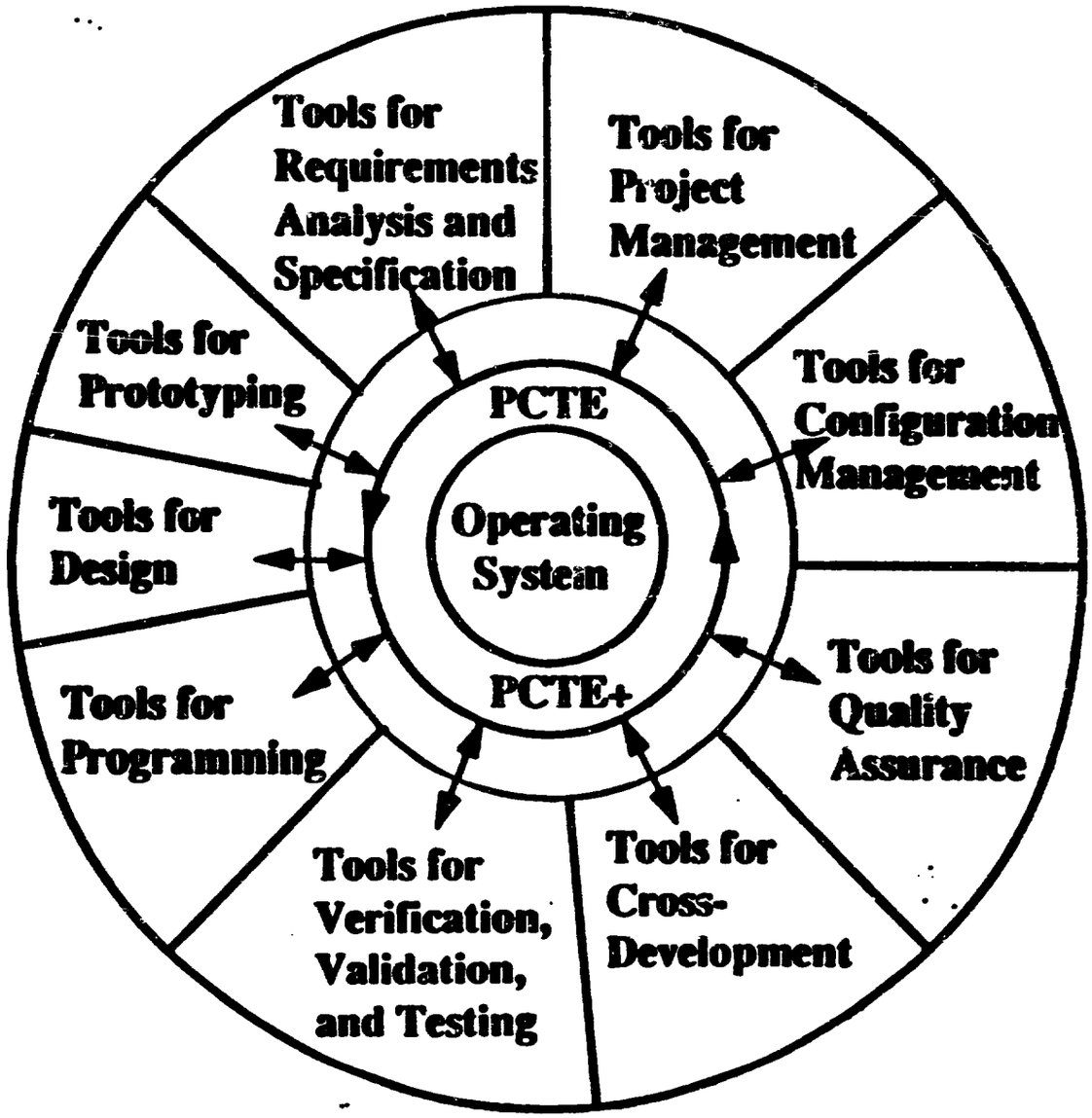
Provides:

- Integrated tools
- Integrity of databases
- Portability of users
- Portable tools
- Portability of databases
- Higher quality tools

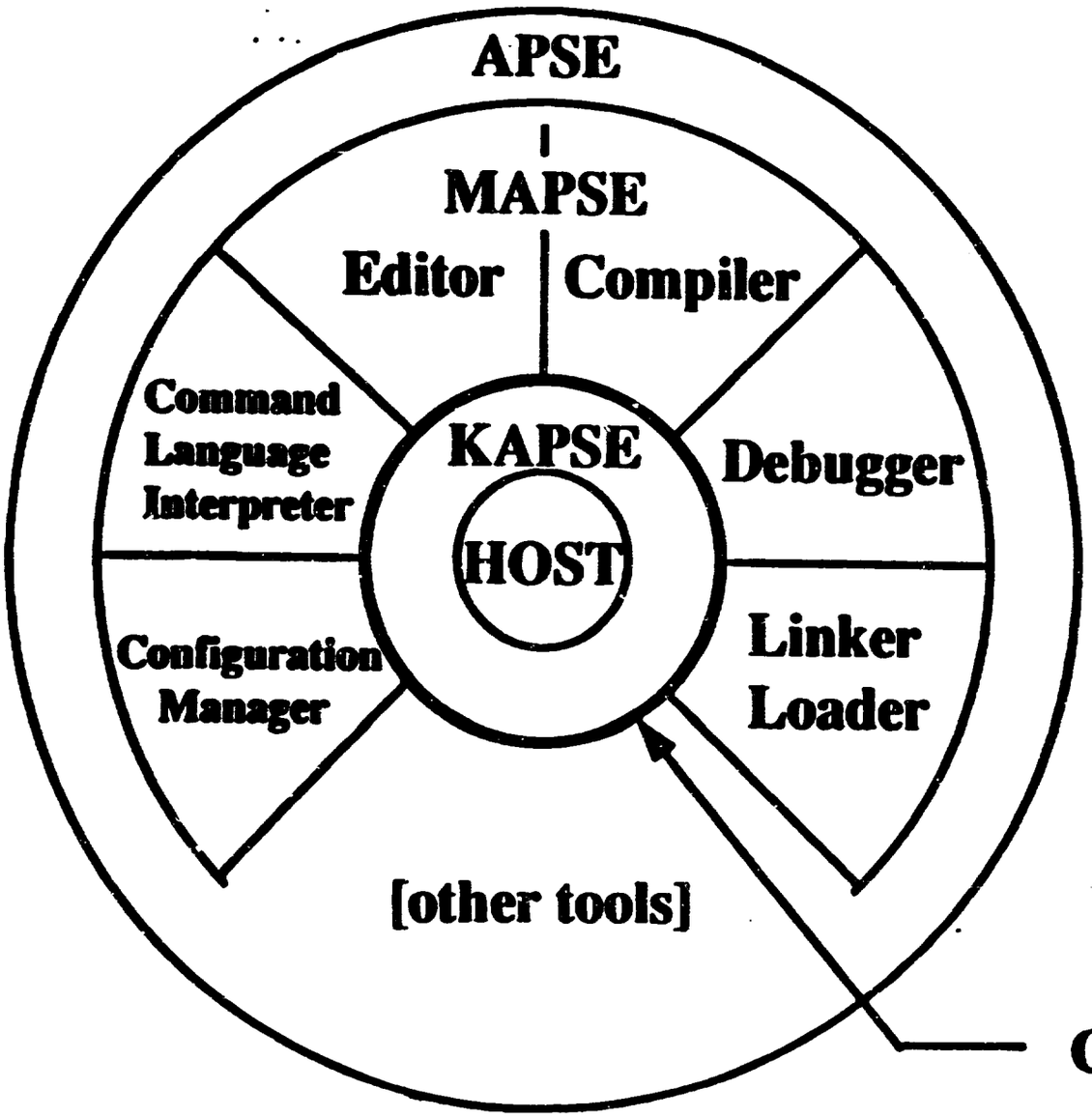
Common Standard:

Acceptable to both military and commercial communities is essential to achieve economies in the acquisition of APSEs.

PCTE & PCTE+

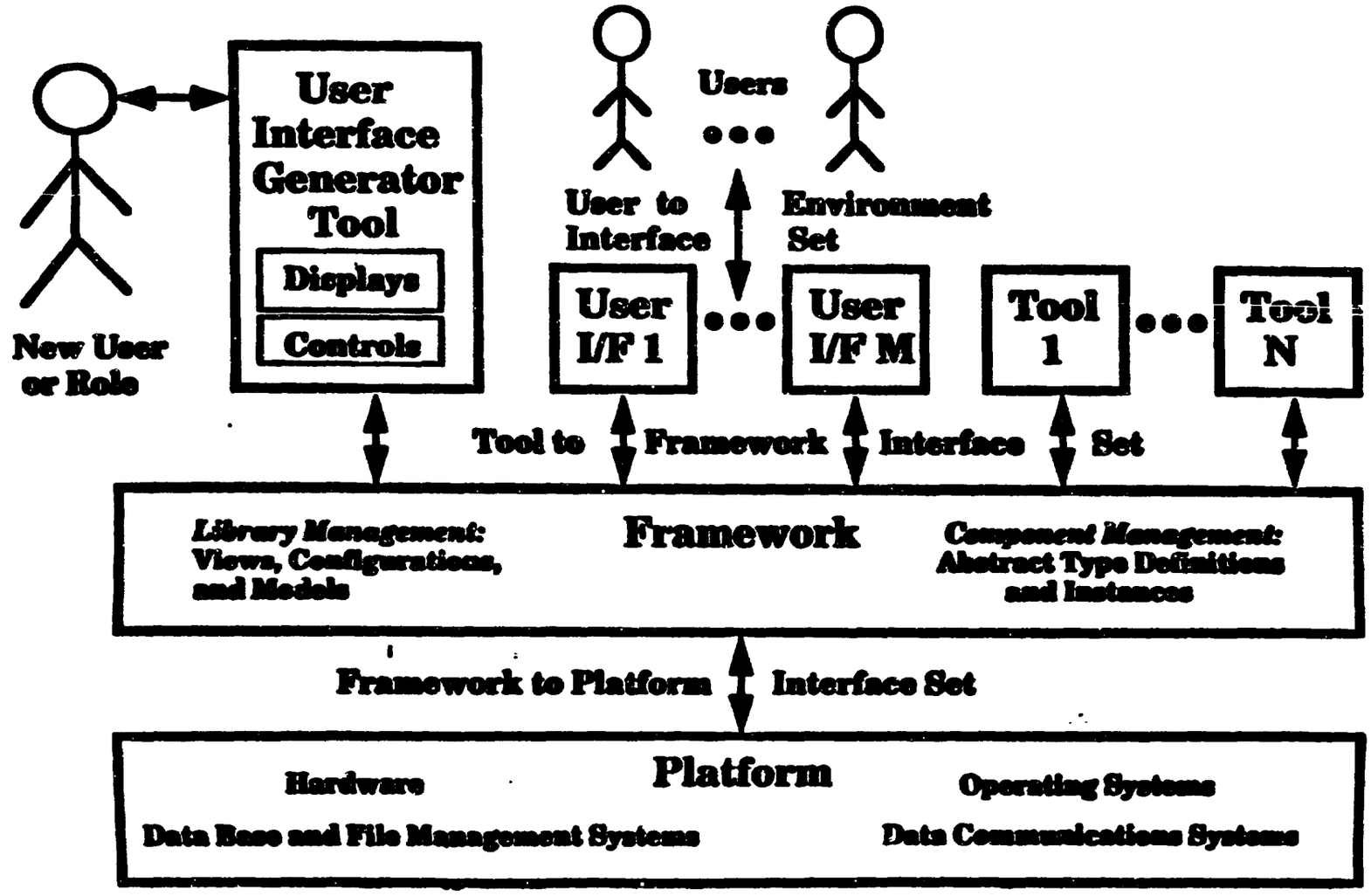

Tools for Requirements Analysis and Specification
Tools for Project Management
Tools for Prototyping
Tools for Configuration Management
PCTE
Operating System
Tools for Design
Tools for Quality Assurance
PCTE+
Tools for Programming
Tools for Verification, Validation, and Testing
Tools for Cross-Development

CAIS-A



PCIS-US, page 5

NIST's Preliminary Reference Model



Defined at Second Integrated Software Engineering Environment Workshop
5-6 December 1989, Gaithersburg, Maryland (Sponsored by NIST)

History - 1

Special Working Group (SWG) on APSE MOU:

- Cooperative effort for the enhancement of APSEs
- Ten nation effort from 1986-1993:
  Canada, Denmark, France, Germany, Italy, Netherlands, Norway, Spain, United Kingdom, and the United States + NACISA (NATO Communications and Information Systems Agency)
- 3 Objectives: Develop usable APSE on 2 distinct architectures
  Demonstrate/Evaluate interface technology
  Develop evolutionary interface standard
- Approximately $50 million combined effort

History - 2

Interface Review Board (IRB) created by SWG on APSE:

- Tasked to accomplish the third objective
  - NATO Requirements and Design Criteria (NRAC)
  - NATO Standard Interface Specifications (NSIS)
- Completed NRAC with Rationale on 6 December 1988
- Proposal for NSIS:
  - Work should be done by teams of experts
- Cost approximately $6 Million
- Most feasible approach to achieving the NSIS -

Get the CAIS-A and PCTE+ sponsors cooperating

History - 3

- Waltham Meeting of CAIS-A and PCTE+ designers (25 January - 5 February 1988):

  Objective: To identify similarities and differences

  Recommendation: Pursue goal of CAIS/PCTE Commonality

- AJPO/IEPG TA-13 Initiative (6 January 1989) -----> Experts Workshop (Winnersh, UK, 14-28 April 1989):

  Purpose: To determine if convergence feasible

  Recommendation: Convergence technically feasible resulting in evolutionary interface standard.

History - 4

AJPO/IEPG TA-13 Meeting (23 August 1989):

- Reviewed/Released Winnersh Report
- Agreed in principle to pursue such a program
- Identified 5 principles to govern its conduct

  Spans both defense and commercial applications
  Retains maximum possible investment from CAIS and PCTE
  Timed to derive maximum benefits from PCTE and CAIS
  Provides natural evolution of the PCTE and CAIS standards
  Ultimately be submitted for ISO Standardization

New interface specification to be available in mid-1994

PCIS

Portable Common Tool Environment

Common APSE Interface Set

PCIS Programme Established - 1

AJPO/IEPG TA-13 Meeting (14-15 February 1990):

- Reviewed Framework for PCIS Programme
- Reviewed options for conducting Programme
- Agreed

SWG on APSE MOU is recognized mechanism for conducting PCIS Definition Phase (Now Requirements Validation Phase)

SWG on APSE chairman to investigate possible mechanisms for remaining phases

Civil organizations encouraged to become actively involved

PCIS Programme Established - 2

AJPO/IEPG TA-13 meeting (13-14 August 1990):

- Confirmed support for PCIS

  As progressive evolution from CAIS to PCTE
  By incorporating soundly based emerging technologies

- Planned for April 1991 Workshop to identify PCIS Requirements

PCIS North American Expert Team

Team Leader:

Gary Pritchett	SofTech

Expert Team:

Frank Belz	TRW
Geoff Clow	SofTech
Dr. Tim Lindquist	Arizona State University
Bob Munck	UNISYS
Dr. Erhard Ploedereder	Tartan

Expert Consultants:

Herm Fischer	Mark V Systems
Gary McKee	McKee Consulting

European PCIS Expert Team

Team Leader:	Organization:
John DAWES	ICL
Expert Team:	
Francois AUDRAS	SYSECA
Professor Jean-Claude DERNIAME	CRIN
Regis MINOT	GIE EMERAUDE
Gérard MEMMI	BULL HN
Christian BREMEAU	GIE EMERAUDE
Boris GELDER	IABG
Expert Consultants:	
Claude MAUPETIT	CR2A
Gerard BOUDIER	GIE EMERAUDE
Dick FIKKERT	FEL-TNO

ECMA TC33 Participation

ECMA Meeting (7 November)

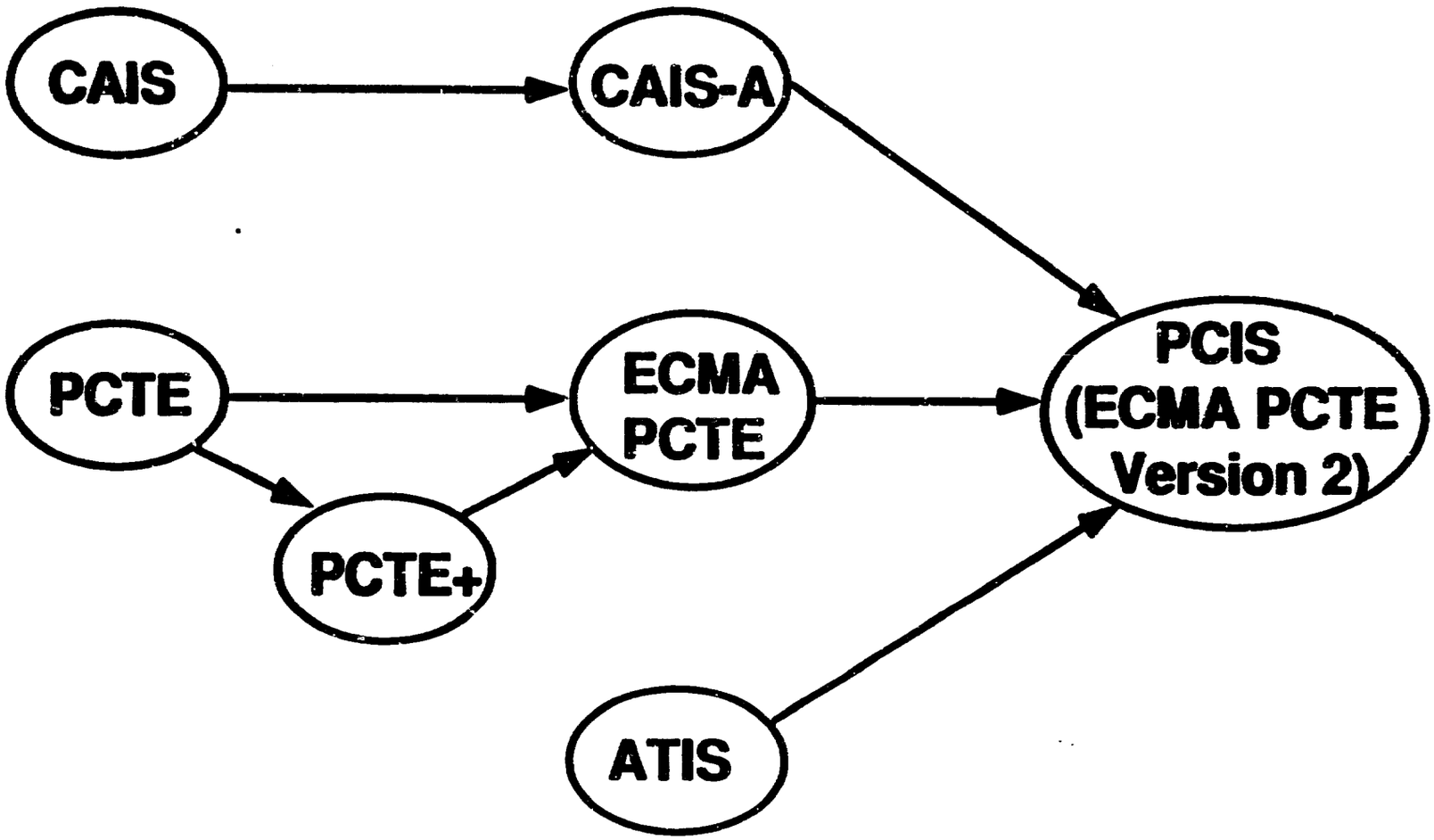
- Endorsed objectives of August 1990 AJPO/TA13 Communique
- Explore avenues of cooperation with SWG on APSE

  Objective to include construction of "Next Generation" PTI
  Open access to information
  To establish a satisfactory relationship

- ECMA TC33 resolved:

  TC33 Chairman to attend 11 December SWG on APSE
  TC33 explore flexibility of current workplan
  To meet with SWG to scope "Next Generation" & involvement
  TC33 begin gathering "civil requirements" for April Workshop

Interface Evolution



STARS Participation

STARS already has important impact to PCIS

- Support for PCIS Expert and Reviewers

AJPO/STARS Meeting (9 November 1990):

- Can commit to a layered POSIX/ATIS/PCIS Approach for STARS
- Support first and second PCIS Requirements Workshops
- Can support environment research in CAIS-A and PCTE
- Provide Demo Implementation / Assessment of PCIS
- Leading to PCIS part of STARS Program

Joint Memo

NIST Participation

NIST participation is essential as NIST represents the US commercial community

NIST has already supported 4 IPSE Workshops

Support PCIS Requirements Workshops

Solicit input / support from commercial community:

- ADPESO [Association of Data Processing Service Organizations]
- CBEMA [Computer Business Equipment Manufacturers Association]

Support PCIS Public Reviews

Ramifications to SWG on APSE

Definition of PCIS is evolving:

It is definitely NOT

- Simple convergence of CAIS-A and PCTE+
- Based only on military requirements

PCIS WILL

- Be based on both military and commercial requirements
- Support low-cost transition path from CAIS-A / PCTE+
- Include relevant emerging technology (Advance state of the Art)
- Interface with other popular standards
- Serve as viable commercial standard for "Next Generation" environments
- Support by NATO, AJPO, IEPG TA-13, STARS, ECMA, NGCR, NIST, et al

Potential Impact to all previous planning activities

SWG on APSE [11-12 Dec 1990]

- Selected PCIS Managers:

  International PCIS Programme Manager - John Solomond (AJPO, US)

  International PCIS Project Manager - Jacques Printz (CR2A, France)

- Tasked to complete requirements phase with industry by December 1991:

  - Validate NRAC and include civil requirements
  - Investigate what the present technology in interface requirements is providing, and assess the emerging technology over the next 5 years.
  - Compare the results of what industry is providing
  - Analyze these differences, prioritize, and cost the work

- SWG GO / NO GO Decision based on resources and time necessary for PCIS Definition Phase

PCIS Work Plan

PCIS/PLN/2, 26 March 1991
PCIS Requirements Validation Activity Work Plan

2 Public Workshops

Expert Team Analysis

Public Interface
for Comments

Expert Workshop

Presentation to the SWG

Preliminary PCIS Schedule - 1

23 Jan	PCIS Planning Meeting
24-25 Jan	ECMA/PCIS Coordination
27 Feb - 1 Mar	SWG #24
5 Mar - 11 Apr	PCIS Planning Meeting
29 Apr - 3 May	1st PCIS Workshop

-----> Preliminary International Requirements and Design Criteria (IRAC)

-----> Preliminary Interface Technology Analysis

3-7 Jun 2nd PCIS Workshop [5th NIST ISEE Workshop]

11-13 Jun SWG #25

----> Report on 2 workshops

Preliminary PCIS Schedule - 2

Jun-Jul	Expert Team Compare Results
Jul-Aug	Expert Team identification of PCIS work & cost
4 Sep	Deadline for comments from the Public
10-12 Sept	Expert Workshop
	Concur on PCIS Work Plan, Cost Estimate
	Finalize documents
17 Sep	Distribute to SWG Representatives
	Final IRAC, Interface Technology Analysis
	Final PCIS Requirements
	PCIS Work Plan / Schedule / Management Manual
8-10 Oct	SWG #26 - PCIS Programme presented to SWG
	3-5 December SWG #27 for Continuation Decision

Summary

Definition of PCIS is still evolving:

Results from first PICS Workshop indicate that the PICS Programme should have an emphasis of bringing Interface Technology to the environment user.

The first and second workshops will provide the SWG on APSE with valuable information to focus the PCIS Programme.

PCIS must be a viable commercial standard that satisfies military requirements.

PCIS Requirements Validation

First Workshop

Heathrow Park Hotel, UK

29/4/91 - 3/5/91

PCIS Requirements Validation First Workshop

Objectives:

- Start to establish requirements (IRAC)
- Start to establish candidate technologies (ITA)

Processes to be continued at Second Workshop

Outputs:

- **Workshop Proceedings**
  - Précis of plenary sessions
  - Sent to all attenders
- **Workshop Report**
  - Protodraft IRAC and ITA
  - Input to Second Workshop

PCIS Requirements Validation First Workshop

Timetable:

Monday p.m. General Introduction

Tuesday a.m. Needs of PCIS

Tuesday p.m. Emerging Technologies

Wednesday/Thursday Parallel Sessions:
- Platform Suppliers (P)
- Environment Suppliers (E)
- Tool Suppliers (T)
- Environment Users (U)

Friday a.m. Concluding Session
- Parallel session reports

PCIS Requirements Validation First Workshop

Wednesday/Thursday: Tool Suppliers

- Comprehensive PTI
- Integration: data, control, presentation
- Query language
- Tool intercommunication
- Tool registration
- Object orientation
- Interoperability
- Distribution costs
- Help services

Wednesday/Thursday: Environment Suppliers

- Environments and Frameworks
- Multipart structure and conformity
- Suppor. for long lifetimes and reuse
- Formal and informal definitions
- Object orientation (more or less)
- Query language
- Preservation of investment, foreign tools and data

PCIS Requirements Validation First Workshop

Monday afternoon: General Introduction

PCIS Programme	J.Solomond
STARS Programme	R.Munck
IEPG TA13 Programme	B.Gladman
PIMB	F.Sallé

PCIS Requirements Validation First Workshop

Tuesday morning: The Needs of PCIS

Tool Suppliers	J.-P. Bourguignon (SFGL)
Platform Suppliers	G.Sagols (IBM)
CEC Programme	D.Talbot
Environment Suppliers	R.Minot (GIE Emeraude)
Environment Users	T.Shorrock & J.Thornley (BAe)

PCIS Requirements Validation First Workshop

Tuesday afternoon: Emerging Technologies

Opening Address:	J.Derniame
Caught in a Minefield	M.Gibbons (BT)
ATIS	A.Argento
OSF	H.-J. Jeanrond
CFI	T.Rhyne
CDIF	H.Barlow
ECMA PCTE	M.Morron
PCTE+	B.Basdell
CAIS-A	G.Pritchett

PCIS Requirements Validation First Workshop

Wednesday/Thursday: Platform Suppliers

- Small group, useful discussion, no special focus
- Scalability, implementability
- Security
- Validation
- Education and Training
- Internationalization
- Public Domain

Wednesday/Thursday: Environment Users 1

- CM and PM including measurement and traceability
- Integration and scalability
- Openness
- Multiplatform, multilanguage, multimethod
- Support for evolution
- Support for commercial development

Wednesday/Thursday: Environment Users 2

Adoption of new technology to improve quality and productivity is a major challenge for any organisation

Requires partnership between suppliers, users, and standards groups to implement the change process:

- **justification (cost/benefit)**
- **initiation (incremental, organised)**
- **management (commitment, focus)**

Lessons for PCIS Programme:

- Understand user needs
- Show how PCIS-based technology meets user needs
- Roadmap to meet user needs
- Address barriers to PCIS use

NIST ISEE/PCIS/IWCASE Workshop

Redondo Beach

3-7 June 1991

Arcadia	**Sponsorship/Hosts**	TRW

- Hosts: **TRW and Los Angeles SIGAda**
- Sponsors:
  - U.S. Department of Commerce, National Institute of Standards and Technology (NIST)
  - U.S. Department of Defense, The Ada Joint Program Office (AJPO)
  - NATO Special Working Group (SWG) on Ada Programming Support Environment (APSE)
  - International Workshop on CASE (IWCASE)

Arcadia	**NIST/ISEE Workshop Objectives/Results**	TRW

Objectives:

- Review the results of the Reference Model (RM) mapping exercise.
- Enhance the NIST/ECMA Reference Model Document:
  - Review and rewrite new services
  - Identify and define the services related to integration
  - Incorporate feedback of mapping exercise.
- Support the PCIS and IWCASE activities.

Results: Generate a revised Reference Model document to be published late summer.

Objectives:

- Identify and establish the scope of the requirements for PCIS
- Assess the requirements of emerging technologies over the next five years.
- Examine a range of candidate services to serve as the basis for PCIS requirements.
- Examine a range of candidate technologies that should be leveraged by PCIS.
- Support the NIST ISEE and IWCASE activities.

Results: Generate an updated version of:

- International Requirements and Design Criteria (IRAC) document.
- Interface Technology Analysis (ITA) document.

Arcadia	**IWCASE Workshop Objectives**	TRW

Objectives:

- Update standardization working groups with status of other groups.
- Identify overlapping issues and facilitate coordination on these issues.
- Support NIST/ISEE and PCIS activities.

Results: Updated status of standardization activities.

Arcadia	**NIST/ISEE Agenda - Tuesday, June 4**	TRW

- 8:00- 8:30: Workshop Registration
- 8:30- 9:00: Workshop Introductory Remarks
- 9:00-12:00: NIST/ISEE Reference Model Mapping Exercise:
  - CAIS-A,
  - PCTE,
  - CIS,
  - SLCSE,
  - AD/Cycle
- 12:00-13:30: Lunch
- 13:30-17:00: NIST/ISEE Working Sessions
  - Object Management - M. Penedo
  - Process & Task Management - H. Hart
  - Interface & Platform Services - P. Oberndorf
  - User Interface - B. Bagwill

Arcadia	**NIST/ISEE Agenda - Wednesday, June 5**	TRW

- 8:30-10:45: NIST/ISEE Working Group Sessions (Cont.)
- 11:00-12:00: NIST/ISEE Plenary (summary of working groups)
- 12:00-13:30: Lunch
- 13:30-16:30: NIST/ISEE Integration Services
- 16:30-17:00: NIST/ISEE Workshop Closing Remarks

Chairman's Closing Remarks for the Second PCIS Workshop

Mr. Gary Pritchett

General Thanks

Thanks to the Chair and co-chair people for conducting valuable Working Sessions.

Thanks to the Presenters for giving us insights into their activities; these will prove to be useful to the PCIS Programme.

Thanks to the Participants in the workshop; without you it would not have been a valuable event.

Special Thanks

Thanks to Dr. John Solomond and the AJPO for sponsoring the meeting on behalf of the NATO Special Working Group on APSE.

Thanks to Hal Hart and everyone at TRW for hosting the meeting.

Thanks to Currie Colket for all the energy he put into organizing the PCIS Workshop and lining up speakers.

Thanks to Wing Commander Dennis Langdon and the European Expert Team that managed the London PCIS Workshop as they gave us a strong base on which to build at this PCIS Workshop.

What Will Happen Next

We've collected Requirements and Needs from Environment Users, Environment Suppliers, Tool Suppliers, and Platform Suppliers.

We've heard presentations on existing or emerging technologies relevant to the PCIS effort.

Over the next few months the PCIS Expert Team will analyze these inputs to produce the IRAC and the ITA documents.

The IRAC will be produced by analyzing requirements captured at the two PCIS workshops and in the NRAC.

- The Experts will be reconciling conflicts as best as possible.
- The Experts will be consolidating these inputs into a complete document.

The high level structure of the IRAC document will be:

- Requirements of Environment Users:

  From the point of view and from the level of the Environments Users, there will be requirements in the document about the PCIS Process, and there will be requirements on the Products of the PCIS Programme.

- Detailed Requirements:

  These requirements will be similar in structure to requirements in the existing NRAC.

- Required Services:

  An identification of services from the NIST/ECMA Reference Model that are required for PCIS will be located in this section.

When completed, the IRAC will be circulated for a Public Review.

The ITA will be a collection of Summaries of presentations with Presentation Materials. When completed, it will be publicly available.

After the Requirements and the ITA are complete, the Expert Team will formulate a way forward (politically, a set of alternative ways forward) for PCIS. These will be presented to the SWG for a Go/NoGo decision on the continuation of the PCIS Programme.

The challenge for the Experts, then, is to:

- Understand the inputs we have heard here, and
- To craft a way forward that:
  - -- is responsive to what we have heard
  - -- will produce a product that is acceptable to the SWG, and

-- does not invalidate the work and important progress made in the environment and PTI areas by existing programs so far.

The documents produced by the effort of the Experts are in the Public Domain and will be made available to anyone who wants them when they are completed.

Chairman's Impression of the 2nd Workshop

As I observed the different Working Group sessions, an interesting point made is that the level of discussion varied widely.

The Environment Users Group has already had a significant impact on the PCIS Programme.

- The sponsor and the PCIS Experts are much more aware of requirements in this particular area.

- There is so much awareness in this particular area that sometimes it seems that some people want to listen to only this area.

Throughout this Second PCIS Workshop, a popular question I've heard is: "What is PCIS?"

- Sometimes it was asked in the context of "you must tell me what it is before I can tell you what I need."

- Sometimes it was asked maybe to see if we've already decided what it is.

My encouragement to everyone is to let the answer to that question emerge out of the analyses of the inputs we've received so far and in the preparation of the way forward that will be presented to the SWG.

SECOND PCIS WORKSHOP

SWG on APSE Tasking

1. Validate the NRAC and include civil requirements.

2. Investigate what the present technology in interface requirements is providing, and assess the emerging technology over the next 5 years.

3. Compare the results of what industry is providing.

4. Analyze these differences, prioritize, and cost the work.

QUESTION FROM
FIRST & SECOND PCIS WORKSHOP

What is PCIS?

SECOND PCIS WORKSHOP

Results

1. Interface Technology Assessment (ITA)

   Contain presentation slides of PCIS, NIST ISEE & IWCASE

   Contain summary of PCIS Presentations

   Distributed to attendees of Both PCIS Workshops ~ 8 July

2. International Requirements and Design Criteria (IRAC)

   Will capture Environment User Needs

   Will capture Detailed Technical Requirements

   Will capture Required Services

   - Services in NIST/ECMA Reference Model

   Distribution to public ~22 July with solicitation of comments

   Deadline for public comments is 4 September

SECOND PCIS WORKSHOP

THOUGHTS

1. There should be a greater emphasis placed on the needs of the environment user. These requirements will be captured in the IRAC and used for the PCIS definition.

2. It is clear that PTI technology may provide a viable alternative to satisfy the needs of the environment user in the near term. It may provide the best alternative to satisfy the needs of the environment user in the long term.

3. There was excellent progress in updating the NRAC level requirements of the tool builder and suppliers.

4. There are perceived and real barriers to the use of PTI based technology. The PCIS Programme must address these barriers.

5. The 2 PCIS Workshops has provided the SWG on APSE with extremely valuable information.

LA SIGADA PRESENTATION

GOALS AND STRATEGIES TOWARDS DOMAIN-SPECIFIC REUSE BASED DEVELOPMENT

"Goals and Strategies towards Domain-Specific Reuse Based Development"

Teri F. Payton, Unisys

June 5, 1991

STARS REUSE

TECHNOLOGY BASE



STATE OF COTS

- Early metrics tools used in STARS project library
- Licensable general, fine-grained parts (e.g., GRACE, Booch)
- Emerging reengineering support
- Analysis tools/knowledge-based tools provide potential for domain knowledge capture
- No library mechanisms

STATE OF DoD PRACTICE

- *Ad hoc* reuse/scavenging/knowledge captured in people's heads
- Experimentation with domain-specific component sets
- SIMTEL 20 like "as is" parts

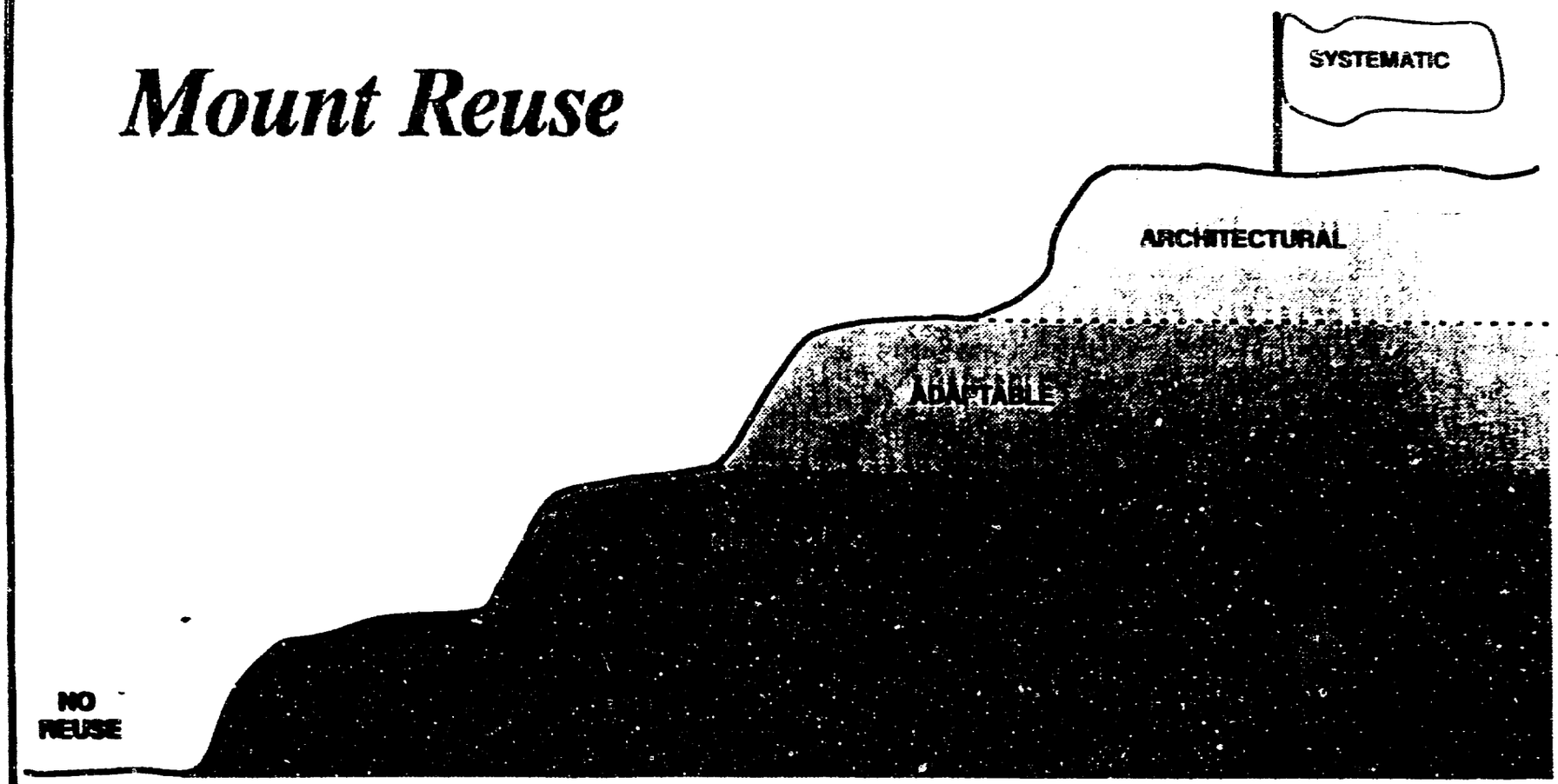
STATE OF TECHNOLOGY

- Move towards reuse-in-the-large
- Move towards application domain focus/application software architectures
- Prototype library techniques/mechanisms (hypertext, faceted, knowledge-based)
- Reuse paradigms not integrated into process

STATE OF RESEARCH

- DARPA ISTO megaprogramming focus
- Application software architectures (e.g., DARPA DSSA)
- Early experimentation in module interconnect languages (e.g., DARPA CPS)
- STARS breakthrough task: formal specifications and reuse libraries

Stages of Reuse



Time-phased Institutionalization of Reuse

(From JLC San Antonio I Reuse Panel, January 1991)

DEGREE OF DoD CONTROL

Model of reuse	0–1 yr	2–4 yr	5–10 yr
Institutionalized GFE Components	← Not Major Thrust →		
Mandatory use of Software QPL (qualified parts)	← Not Major Thrust →		
Centralized catalog	← Not Major Thrust → Some false starts		
Domain-specific Common Architecture Based	Intracompany (Some) Licensible library (Some)	Intracompany Intercompany (Closed) Intra-PEO	Intercompany Intraservice Components industry — Licensible library (COTS) — Support services (NDI) Black Box
Opportunistic	White Box; Public Domain Libraries	White Box	White Box

= Recommended model for DoD emphasis

What is needed to enable a transition to domain-specific reuse?

- Domain-specific software architectures – processes/techniques to create, represent, and utilize the architectures
- Consensus-building process to institutionalize architectures
- Requirements elicitation techniques enabling system requirements creation using a generic family of requirements
- Validation technology to access architectures
- Domain-specific asset assessment criteria and techniques
- Processes and prototyping to synthesize systems from reusable domain-specific assets

What is needed to enable a move towards black-box reuse?

- Precise asset descriptions (akin to hardware spec sheets)
  - Fully define external behavior
    - Semantics, side effects
    - Interconnections
    - Performance (machine processable/testable formulas)
    - Intended operating environment
    - Resource consumption
  - DARPA community MIF concept
- Means to reliably assess integrity/correctness of assets
- Testing technology to qualify assets for usage in a family of systems
- Assurance that an asset will be supported

What is needed to enable a move towards black-box reuse? (Cont.)

- Infrastructure to actively maintain an assets' track record
  - Feedback from users
  - Usage information
  - Problem tracking
  - Consumers reports
  - Recall mechanism
  - Update distribution

What is needed to enable a move towards black-box reuse? (Cont.)

- Infrastructure to actively maintain an assets' track record
  - Feedback from users
  - Usage information
  - Problem tracking
  - Consumers reports
  - Recall mechanism
  - Update distribution

What is needed to evolve reuse libraries into proactive software brokerages/exchanges?

- Easy access from individual's workstations
  - National file systems
- Distributed libraries
  - National file systems
  - Library interoperability
  - Heterogeneous databases
- Easy to find desired asset
  - Intelligent search
- Facilitation of inter-organization communication and asset sharing
- Active solicitation of feedback

What is needed to evolve reuse libraries into proactive software brokerages/exchanges? (Cont.)

- Infrastructure support for
  - Access control
  - High integrity electronic distribution
  - Electronic licensing
  - Update/recall mechanisms
  - Track record capture

What is STARS Doing to Support Reuse

A key STARS objective

- Establish a basis for a paradigm shift to reuse-based development

Focus on mechanisms to support domain architecture-based reuse-in-the-large

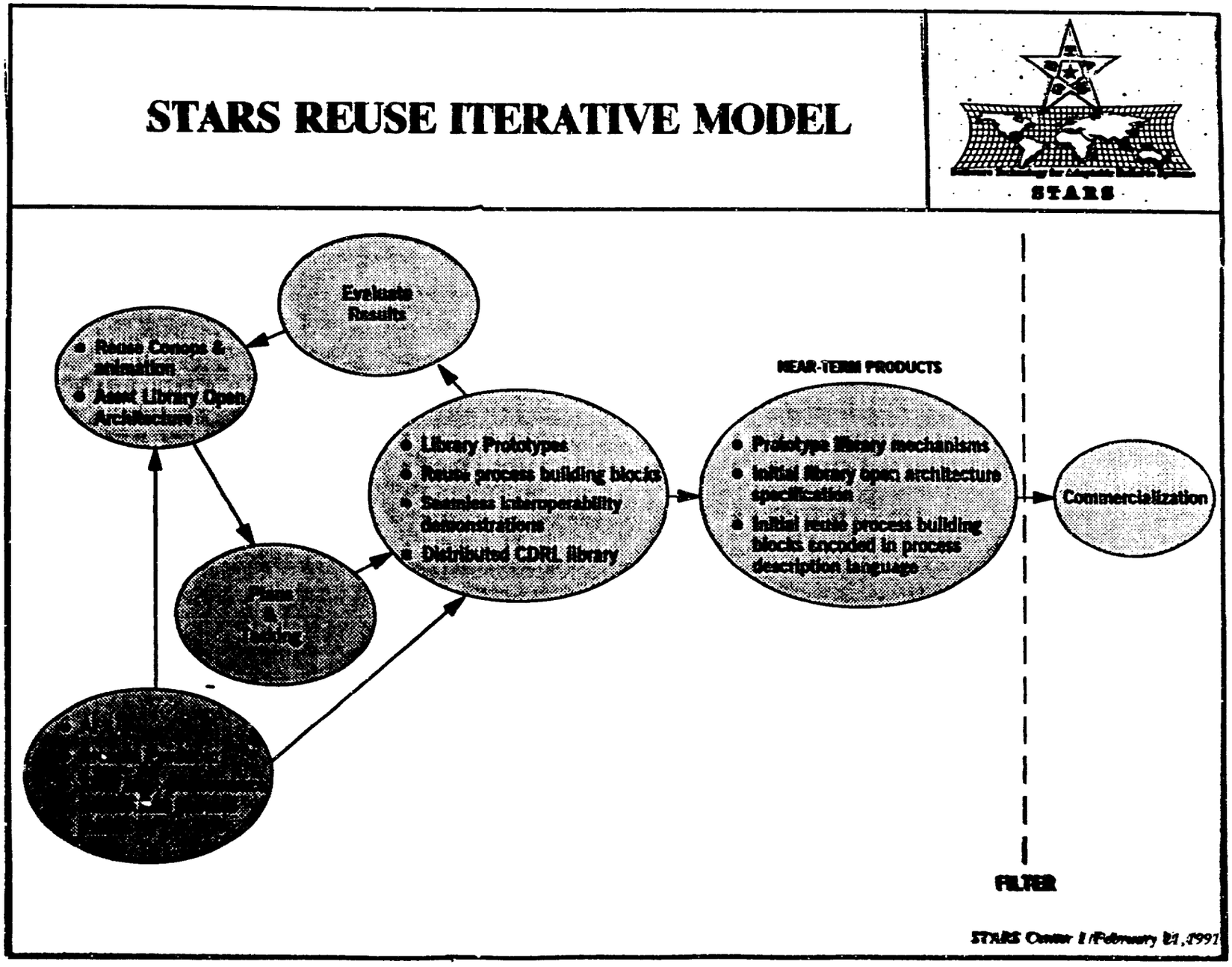
- Library mechanisms
- Reuse-based development processes
- Reuse support tools
- Library interoperability
- An operational ASSET reuse library

STARS MISSION



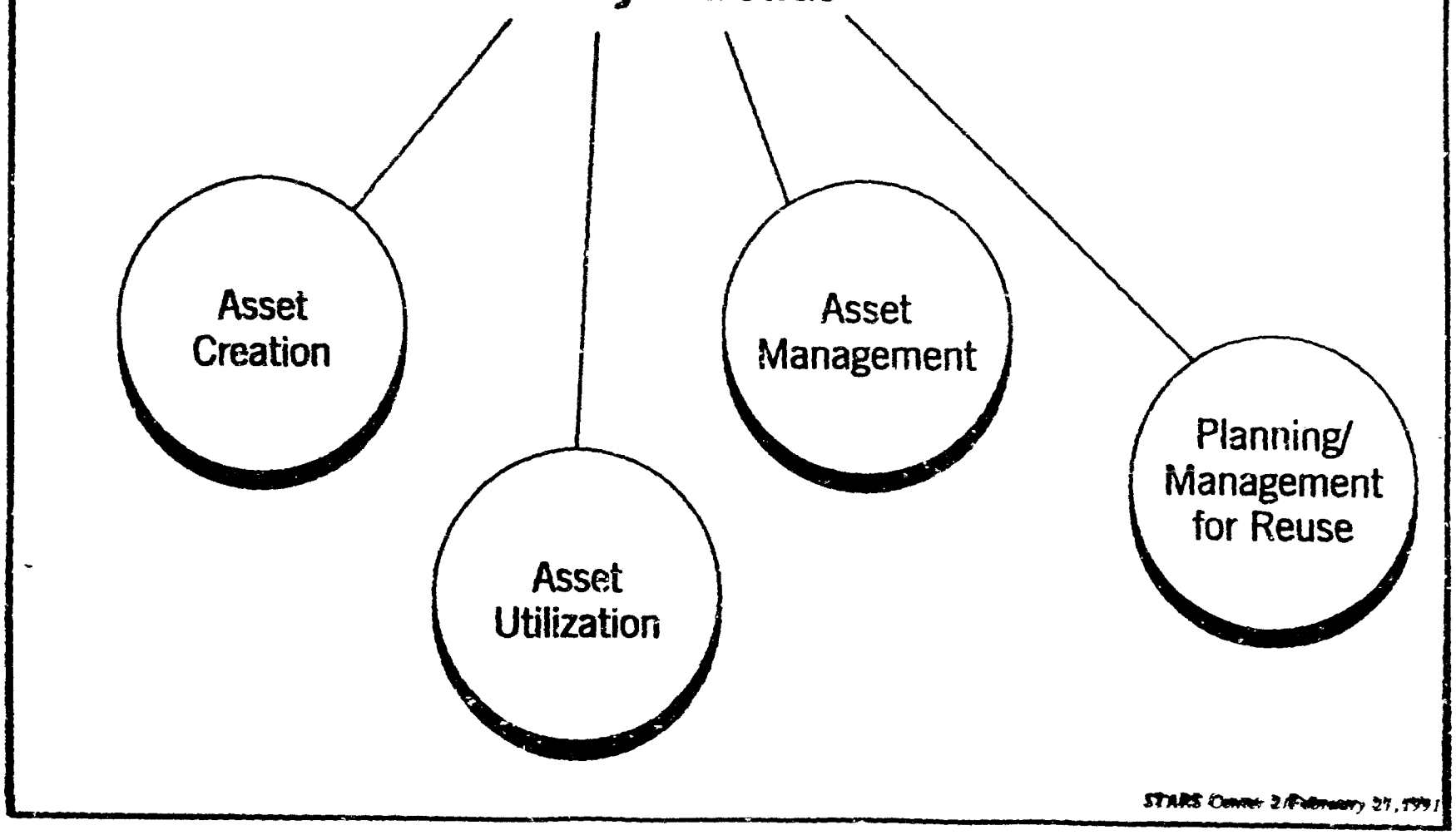
- **Accelerate the paradigm shift**
  - **Within DoD software–intensive system development and maintenance community**
  - **To a process–driven, domain–specific–reuse–based, technology–supported paradigm**
  - **This paradigm supports collaborative development across geographically dispersed project teams**

STARS REUSE ITERATIVE MODEL



REUSE PROCESSES

STARS



CANDIDATE REUSE PROCESS BUILDING BLOCKS



Creation

- Domain modeling/analysis
- Design recovery/reverse engineering
- Software architecture definition
- Reusable Ada component development
- Software generator creation

Management

- Asset metrics
- Asset description
- Classification
- Library capabilities
- Asset librarians
- Certification

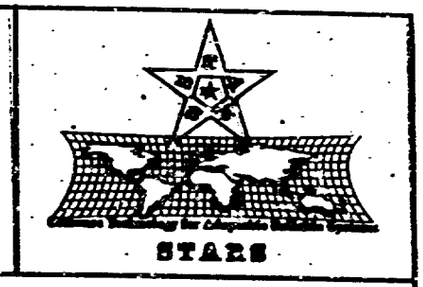
Utilization

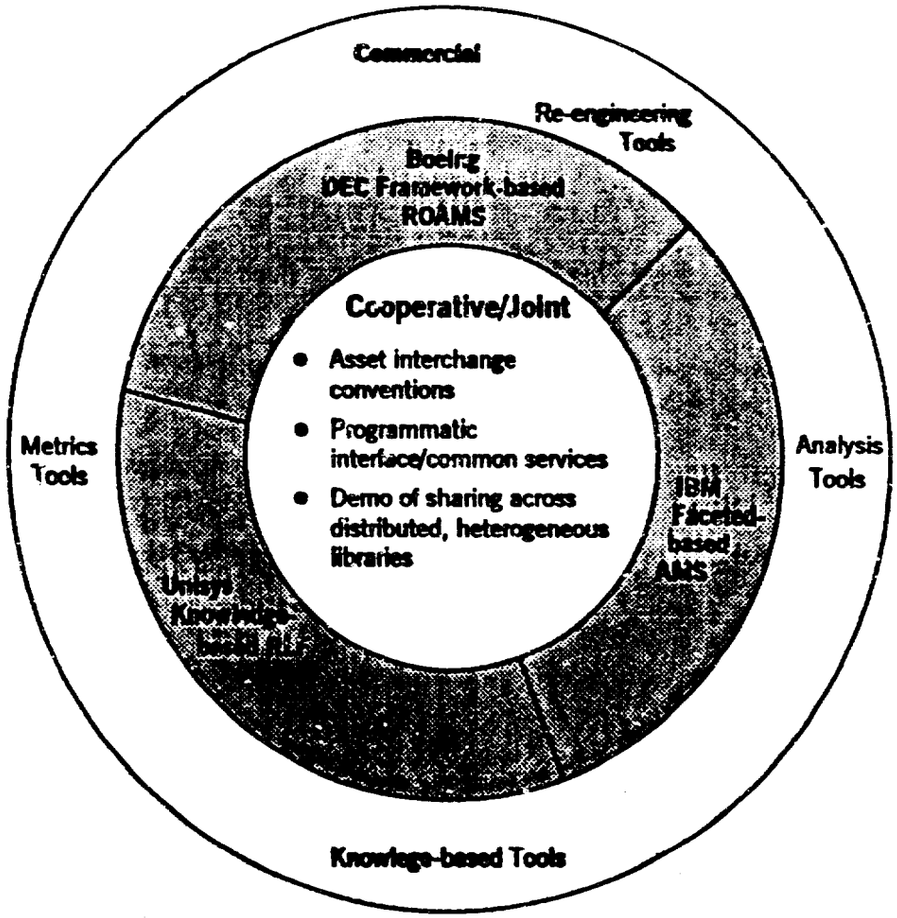
- Asset understanding
- Asset access
- Composition
- Re-engineering
- Evaluation/selection

Planning/Management

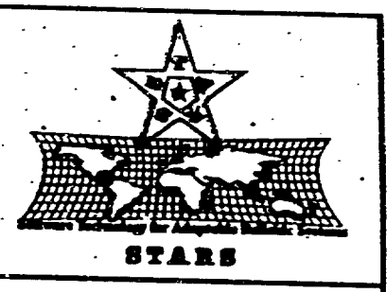
- Develop reuse strategy
- Incorporate reuse into process
- Process measurement

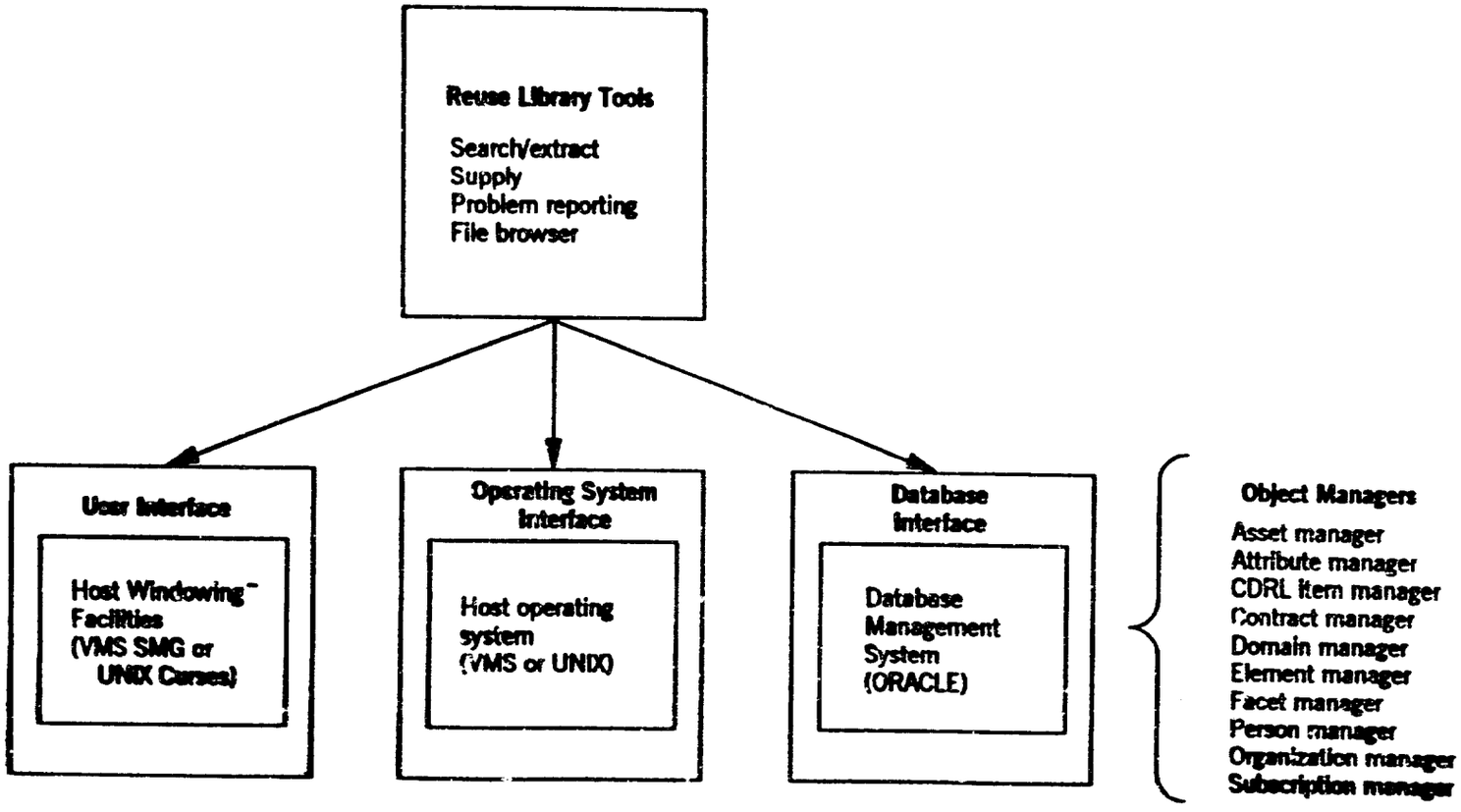
STARS LIBRARY MECHANISMS/ OPEN ARCHITECTURE



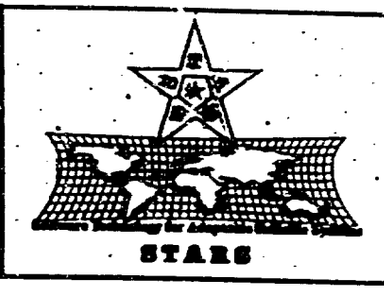


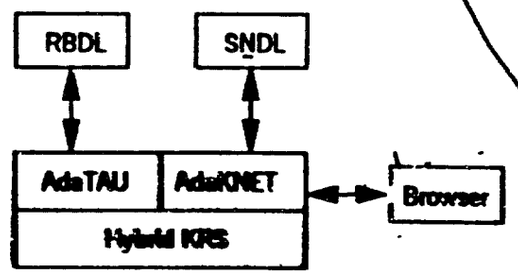
IBM/SAIC FACETED-BASED REUSE LIBRARY



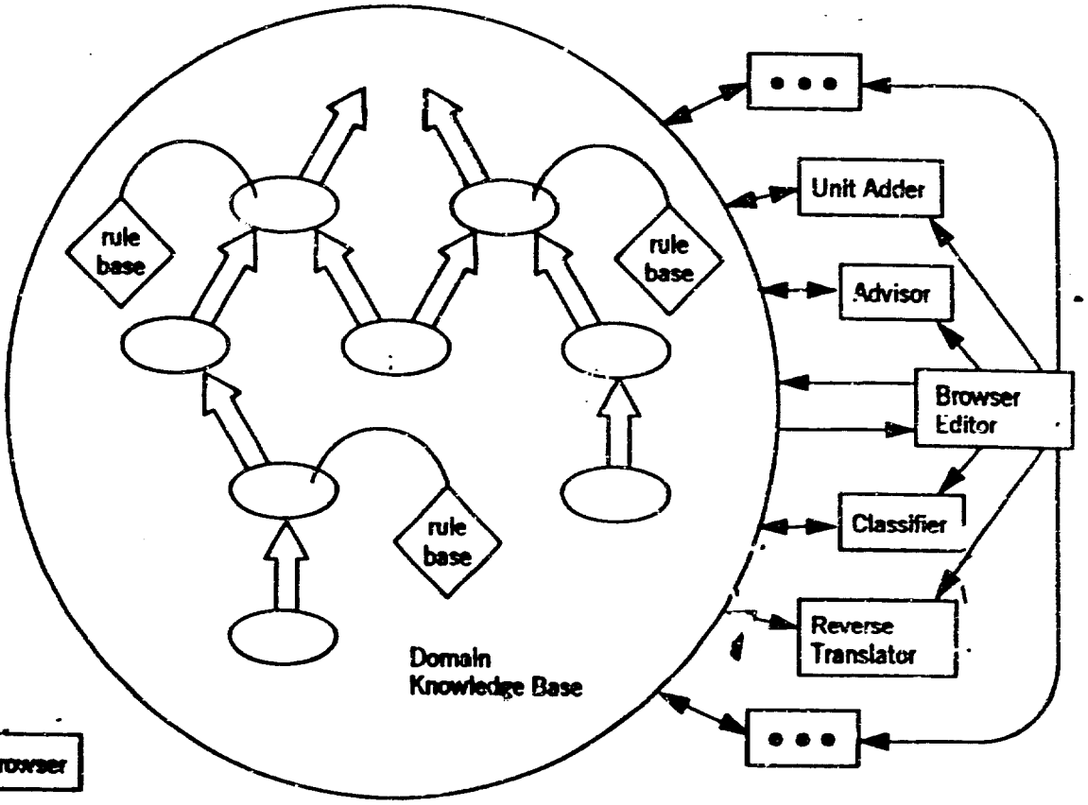


UNISYS KNOWLEDGE-BASED RLF



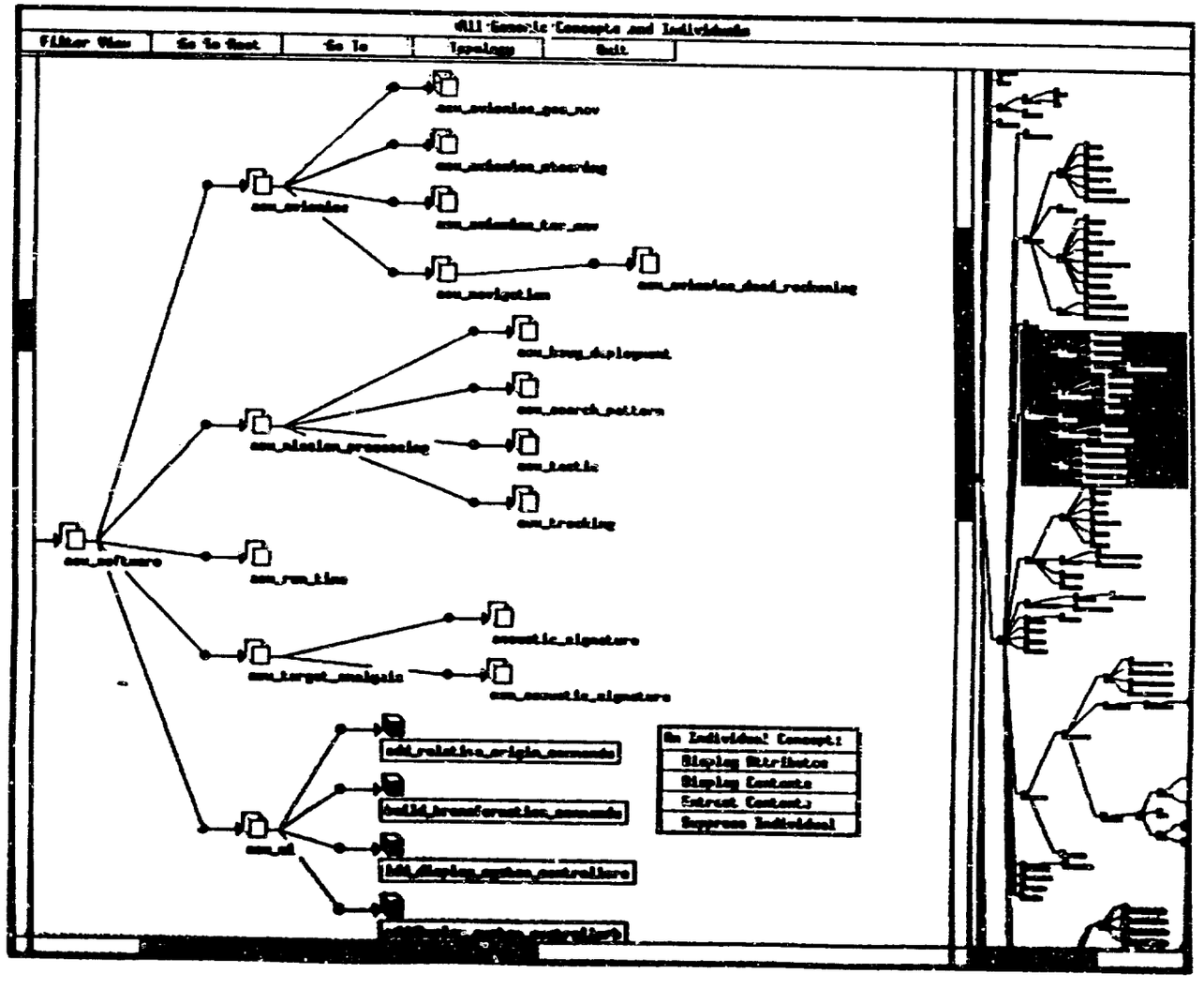


RLF Architecture



Librarian Architecture

RLF Graphical Browser



ASSET ROLE

Short term

- Focal point for software reuse within the defense industry

Long term

- Help stimulate the creation of a US software reuse industry
- Ensure DoD is supported by that industry

How?

- Provide networked electronic access to DoD software assets
- Invite private industry to offer their software assets in the same way

ACTIVITIES

- Demonstrate open architecture for interchange among reuse libraries
- Acquire software assets
- Categorize those software assets
- Preserve those software assets
- Distribute software assets
- Recall software assets

VALUE-ADDED ROLES OF ENTREPRENEURS

- Electronic and hardcopy (re) distribution
- Proprietary search and selection methods
- Consulting in specific application areas
- Systems integration in specific application areas
- Setting up local reuse programs and repositories
- "Yellow pages" for proprietary goods and services
- Privatization of ASSET's functions

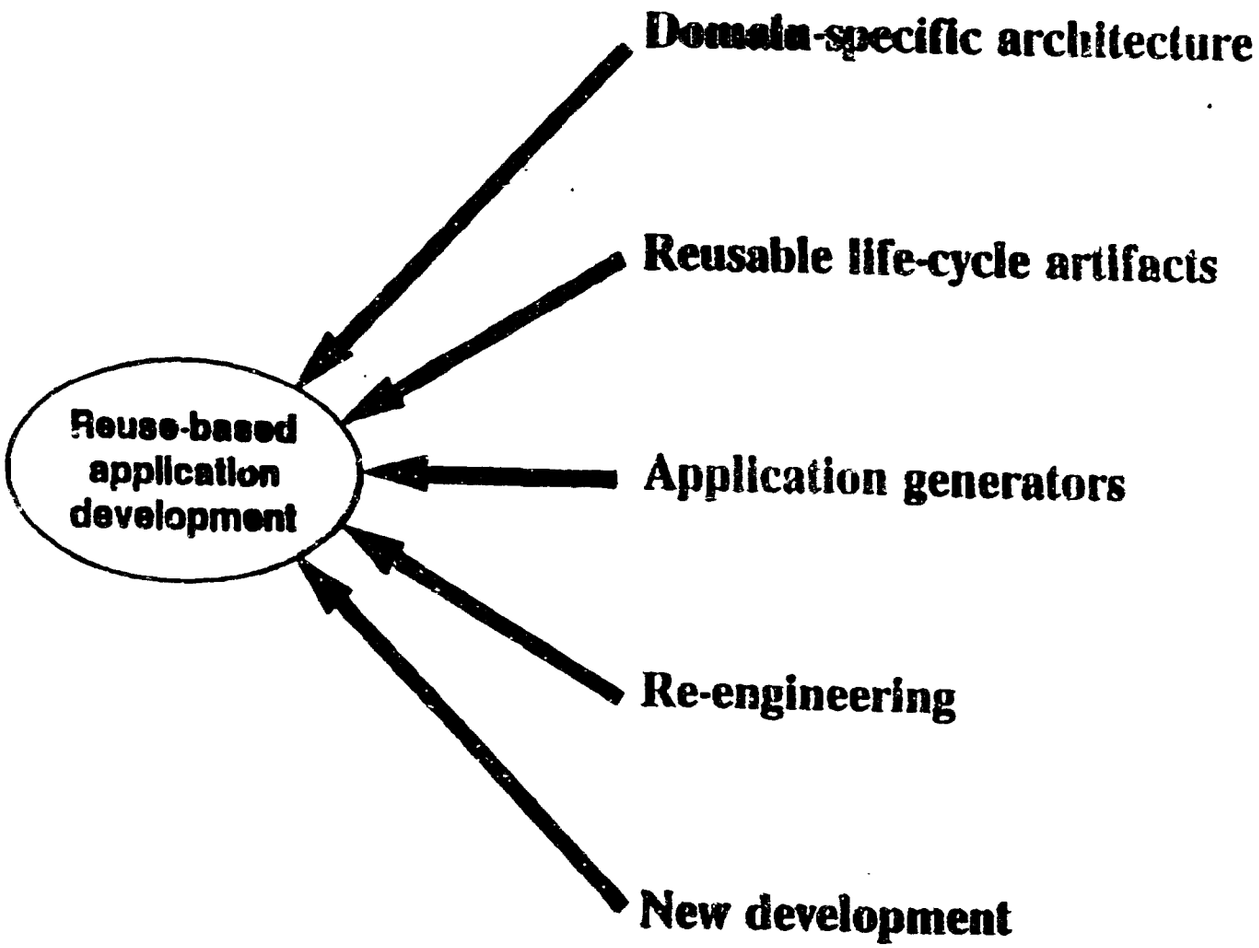
SUMMARY



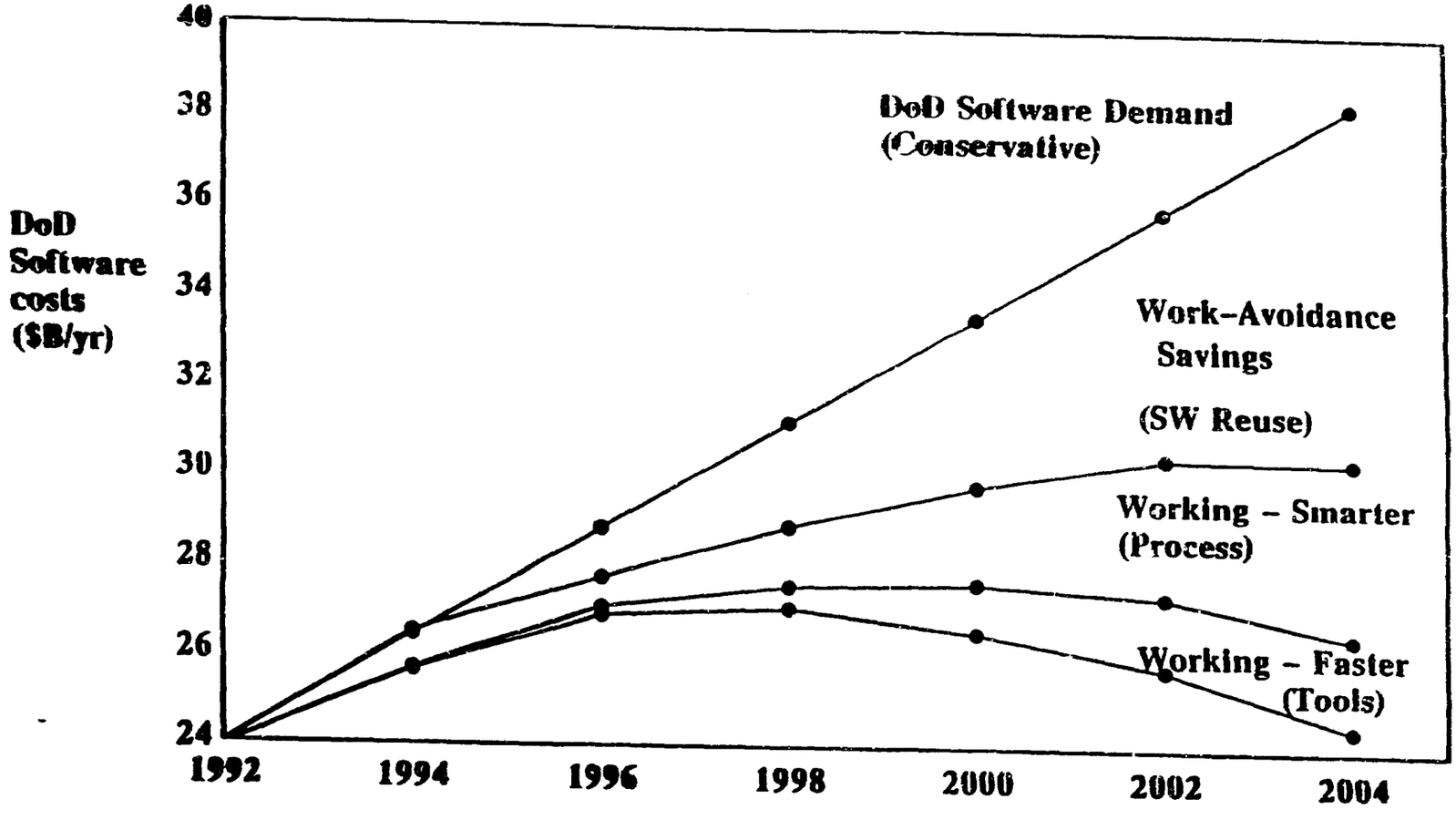
- Support continuous process improvement
- Transition from ad–hoc to high–impact, large–scale, domain–architecture based reuse
- Integrating process and reuse capabilities
  - Within COTS open–standards based SEEs

STARS REUSE

REUSE-BASED DEVELOPMENT SCENARIO



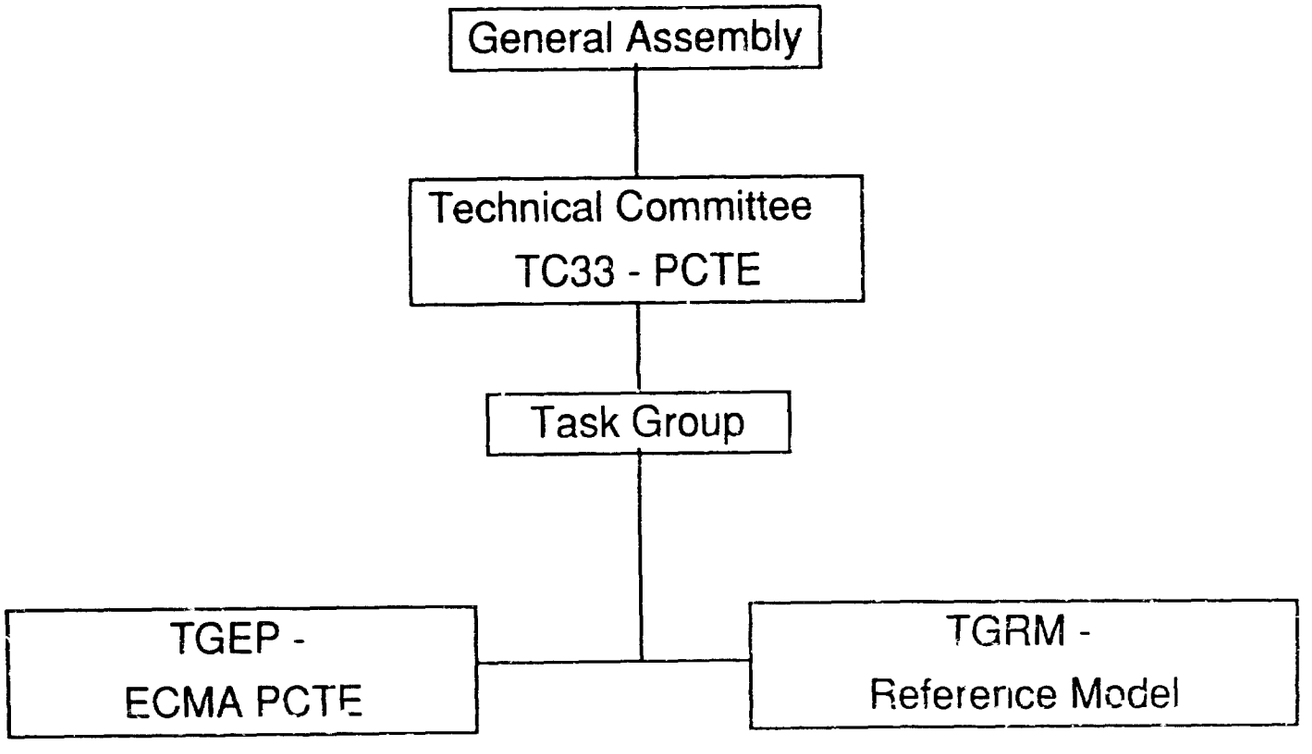
EXAMPLE PROGRAM PRIORITIZATION ANALYSIS: DOD SOFTWARE TECHNOLOGY USAGE/SAVINGS MODEL



SECTION I

NIST ISEE WORKSHOP

ECMA Organisation



ECMA = European Computer Manufacturers Association

ICL

ECMA TC33 TGRM POSITION

- ECMA TC33 TGRM remains actively committed to progressing the CASEE RM work
- By the September meeting of NIST, ECMA TC33 TGRM will provide ECMA's feedback on the changes made to the CASEE RM to NIST
- ECMA TC33 TGRM is performing a set of mapping exercises from which to validate the CASEE RM. These will be shared with NIST

MG/STDS/

ECMA TC33 TGRM
Organisations and Mappings

Organisations

- 0 Eureka Software Factory
- 0 EAST Environment
- 0 Digital
- 0 BNR
- 0 Syseca
- 0 HP
- 0 BT
- 0 University of Dortmund

Mappings

- 0 ESF
- 0 EAST
- 0 ATIS
- 0 Entreprise 2
- 0 Softbench .....
- 0 PCTE

MG/STDS/

ECMA TC33 TGRM: - TIMETABLE

0 Next TGRM Meeting: 26th June 1991

- Heathrow, London
- discuss NIST changes & status/progress to date on mapping work

0 Next TC33 Meeting: 3-4th September 1991

- Nice, France
- invite NIST representation
  - discuss mapping results to date
  - review changes made, and lessons learned with respect to the CASEE RM
  - clarify publishing and labelling details to ensure document consistency

MG/STDS/

The Navy's

Next Generation Computer Resources (NGCR)

PROJECT SUPPORT ENVIRONMENT

WORKING GROUP

(PSEWG)

Tricia Oberndorf

Naval Air Development Center,
Warminster, PA
(215)441-2737
tricia@nadc.nadc.navy.mil

APPROACH

OBJECTIVE: select industry-based PSE interface standards for use in support of Navy systems

* Joint industry/Navy working group
  - Navy co-chairmen (one military, one civilian)

* Subgroups according to needs
  - Approach/Requirements/Available Technology
  - by interface area

* Benefit from/coordinate with as many other related projects as possible (e.g., NIST, STARS)

* Start: ~ April 1991

* Completion: ~1997

ISSUES

* possible goals
  - tool "mix & match"
  - minimize training
  - maximize ease of transition to PDSS
  - maximizing tool commonality
  - host interchangability
  - attaining particular SEI assessment level
  - compatibility with other NGCR standards

* scope of the PSE
  - software only?
  - Ada only?
  - what application mix?

* level/extent of standardization
  - choose toolset
  - choose OS,DBMS, etc.
  - standardize on interfaces key to the PSE framework

* user interface

Feedback from AD/Cycle Mapping

Environment Characterization: Commercial product initially directed for business oriented customers AD

Time required for mapping: ≈ 1 hr/service

Feedback on mapping process:
- Mapping workshop mandatory
- Example PCTE mapping helpful for reference

Was the right information drawn out?
- In general, topics in current RM good
- What is missing:
  - User interface standard & integration
  - Security: make complete or give a pointer to security RM

Strengths of RM:
- Good for educational purposes
- Good to understand "complete" environment
- Good to have input to standards process
- Useful to answer questions about other environments
  ex: "How does PCTE do X ?"

3 June 91 BF Meyers-IBI

AD/Cycle Mapping Feedback (cont.)

Weaknesses / Recommendations for RM:

- Tool Registration should be consolidated
- Integration information should be consolidated (sections 9 & 18)
- "Require" all dimensions that have sections in the RM document
- Guidance needed for how to handle services provided by underlying operating system
- Clarify differences between 7.1 Data Model & 7.2 Data Storage service

Continuing interest?: Yes

3 June 91 BF Meyers-IBM

AD/Cycle Mapping Feedback (Cont)

Example of New Service proposed:

- Add 4th type of integration criteria
  - → FUNCTION INTEGRATION – common functions to be used by all AD/Cycle tools
    - allows for semantic consistency
    - Examples of common functions:
      - installation procedures
      - tool registration / invocation
      - communication protocols
      - editors - text & graphic
      - display / list services
      - help facilities
      - exception handling
      - security services
      - project management integration
      - :
      - :
      - :

CAIS-A (MIL-STD-1838A) MAPPING TO NIST REFERENCE MODEL

Geoff Clow
SofTech, Inc.
Clow@NOSC.MIL

Purpose

Mapping Effort

Guidelines, NIST Support

Benefits to Author

Services
- **Service Omissions**
- **Service Incompleteness**

Service Dimensions
- **Dimension Redundancies**

PURPOSE

Assist Evolution of the RM and the Guidelines.

Not to e.g. present CAIS and/or the RM.

CAIS

Standard interface specification for:

> **Programming of sophisticated, integrated project support environments.**
>
> **Portability of projects (tools, databases, users).**

MAPPING EFFORT

2 Day kick-off meeting

Orientation, Guidelines, Practise.

2 Weeks, 1 person mapping.

1 pass, little review and revision.

1 - 2 Weeks more for proper public draft

Intended, for revised RM.

Mapper Background

CAIS author and implementer

Familiarity with subject system.

Participant in Waltham and Winnersh CAIS/PCTE joint studies

Experience with similar exercises (mapping to an independent model).

GUIDELINES, NIST SUPPORT

Valuable kick-off meeting

Uncertainties inevitable

Number of service areas and dimensions,
Level of detail,
Only one example from which to generalize.

Critique of lab exercise was single most helpful experience.

Future mapping efforts would benefit from:

Existence of additional mappings.

Simulating lab experience in Guidelines

Examples (correct and less correct) with critique;

Stress information partitioning and standard characterizations, where possible (provides, supports, available, unavailable).

BENEFITS TO AUTHOR

Identify incompleteness and redundancy in a system or its documentation.

Identify redundancy, alternatives and complements between multiple systems.

Analogous comparison exercises:

Framework vs Requirements
e.g. CAIS, PCTE vs RAC, NRAC, EURAC

Framework vs Framework
e.g. CAIS vs PCTE in Waltham, Winnersh

Framework vs Reference Model
e.g. Mappings to ECMA & NIST RMs

Framework vs Framework, through Mappings
Objective means to all of the above.

SERVICES

Over 50 functional areas

Over half directly applicable to general-purpose PTIs such as CAIS.

> More for frameworks defining more policy (development process management, detailed integration conventions).

Extremely complete for CAIS.

SERVICE OMISSIONS

Time Services

Error Handling Services

Input-Output

> Distinguish uninterpreted (e.g. file) data from interpreted (captured in data model).

SERVICE INCOMPLETENESS

State vs Event Monitoring

- Inadequately distinguished in descriptions.
- Interchangeable examples.
- Events iff DB state change, often, so logically related.

Data vs Task Transaction

- Task transaction service is under development.
- Task transactions are "supported" by CAIS: Should such potential applications be discussed?
- Guidelines are needed:
  - How much coverage of potential applications?
  - Standard terminology to capture availability of functions (defined vs presupposed vs facilitated).

SERVICE INCOMPLETENESS, cont'd

Data Interchange addresses common format but overlooks other essential services.

- **Support for detection and reconciliation of multiple instances (representations) of the same entity.**
- **Prevention (or intermediate form) of multiple entities with same unique identifier.**
- **Exchange and reference of "foreign" unique identifiers.**
- **Convenient exchange of objects' dependencies, such as typing information, components.**

SERVICE DIMENSIONS

13 divisions potentially applicable within each Service.

Attempted to apply 10 uniformly.

ICE: Internal - Conceptual - External
ROD: Rules - Operations - Data
Related Services
TIM: Types - Instances - Metadata

DIMENSION REDUNDANCIES

Data is redundant with e.g. Internal, External, Instances

Omit.

Types are a subset of Metadata, which is both a Dimension and a Service area.

Merge.

ICE: Internal - Conceptual - External
ROD: Rules - Operations - Related Services
TIM: Types/Metadata - Instances

SOFTWARE LIFE CYCLE SUPPORT ENVIRONMENT (SLCSE)

OUTLINE:

- EFFORT CHARACTERIZATION
  - History
  - Operational Concept
  - Future Directions
- MAPPING EXERCISE
  - Results
  - Time Span
  - Mapping Guidelines
  - Mapping Process
- COMMENTS ON REFERENCE MODEL
  - Strengths
  - Weaknesses
  - Additional Service Descriptions
- SAMPLE MAPPING OF SLCSE SERVICE
  - Conceptual
  - Operations
  - Relationships Between Services

JAMES MILLIGAN
AFSC
ROME LABORATORY
GRIFFISS AFB, NY

SLCSE

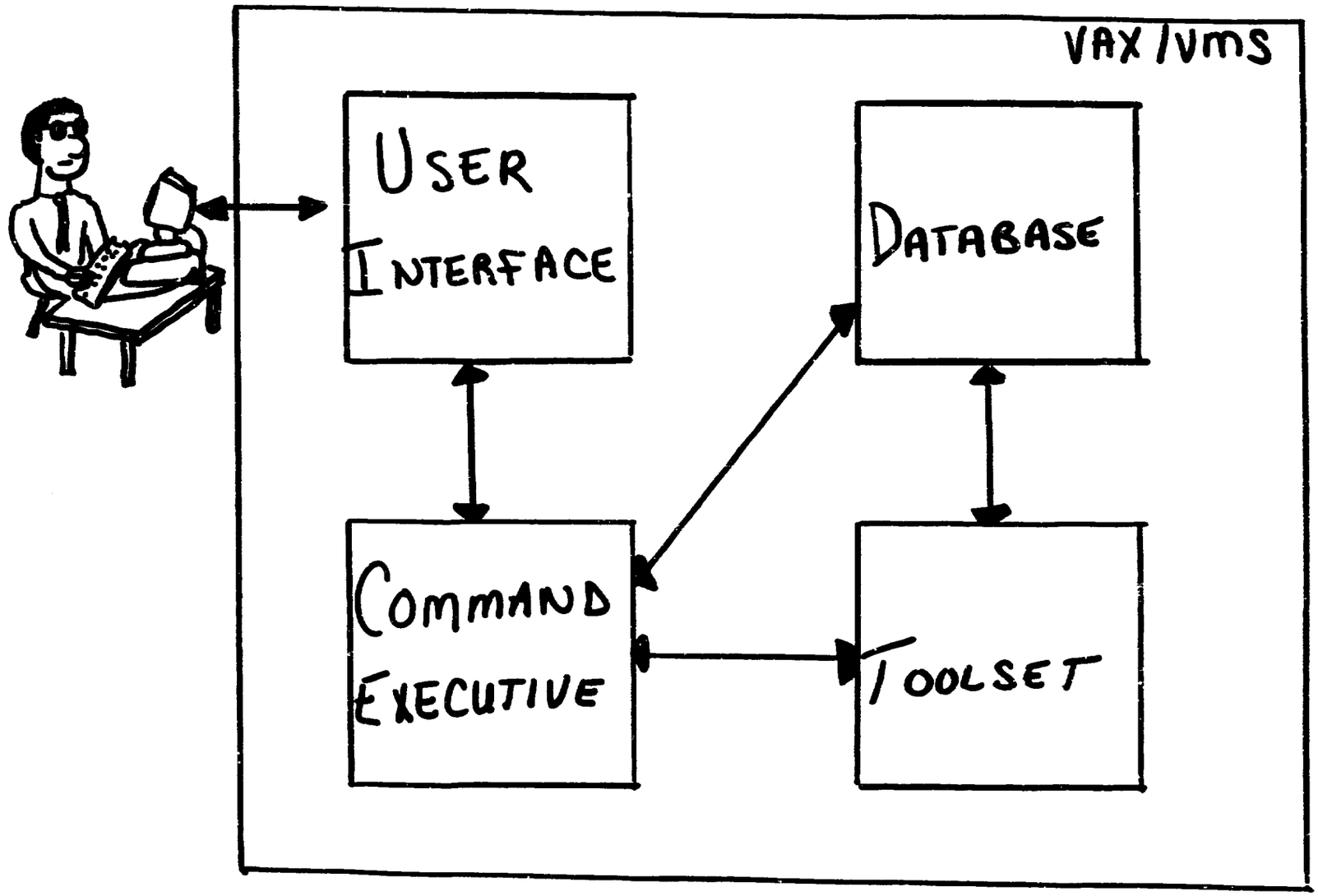
History

Prior to 1986	Exploratory SEE Research
1986 - 1989	SLCSE Version 3.5 Implementation
1989 - 1990	SLCSE Beta Test Sites - 3 AFLC ALCs
1991	SLCSE Version 3.8 Delivered with SLCSE Project Management System (SPMS)

- Rome Laboratory Sponsored Effort
- $4 Million Total Investment
- Developers: General Research Corp (Prime)
  Software Productivity Solutions (sub)

SLCSE

OPERATIONAL CONCEPT



SLCSE

Operational Concept

- User Interface
  - Common & Consistent Presentation Style
  - Window-based, menu driven
  - Character-oriented (e.g., VT100 terminals)

- Command Executive
  - Controller of SLCSE operations (e.g. tool invocation)

- Database
  - Repository for formal life cycle information
  - Integrating mechanism for tools to share info
  - Based on an Entity-Relationship (ER) model interface to an underlying RDBMS

- Toolset
  - Initial set of tools integrated
  - Consists of 9 broad categories (reqts to testing)
  - Flexible & Extensible, integrate any number
  - Data integration services for remote, network tools

SLCSE

FUTURE DIRECTIONS

- SLCSE Enhancements (2 years)
  - X-Windows User Interface
  - Additional RDBMS Support
  - Open Architecture
  - Improved Performance
  - Based on Beta Test site & User Concerns

- SLCSE Productization (5 years)
  - Goal to transition supported product to AF/DOD/DOD Contractors

- SLCSE Special Customer Services
  - Task order based contract mechanism
    - installation
    - customization
    - training
    - on-site maintenance

SLCSE

MAPPING EXERCISE

- Results:
  - Services of SLCSE mapped to Reference Model in 301 pages.

- Timespan:
  - Period covered: Mar 13 - May 7, 1991.
  - Actual time spent: Approx. four 40-hour wk.s.

- Mapping Guidelines:
  - Appropriate, but marginally useful after lessons learned during 12-13 March meeting.

- Mapping Process:
  - Useful in assessing areas for enhancements.
  - Useful in gaining in-depth understanding of the system (for both the mapper and the reader).
  - Time-consuming: may exist ways to stream-line the process without compromising the objectives of having the Reference Model.

SLCSE

COMMENTS ON REFERENCE MODEL

- STRENGTHS:
  - Generally correct and appropriate service descriptions.
  - Dimensions provide multiple perspectives/descriptions of a single service.

- WEAKNESSES:
  - More examples needed.
  - New services need refinement.
  - "Dimensions" (ICE, ROD, TIM), "points" (Internal, Conceptual, Operations, etc.), and other "factors" often referred to interchangeably.

- ADDITIONAL SERVICE DESCRIPTIONS:
  - RM Section 9: Integrated Tools
  - RM Section 14: Framework Definition/Modification
  - " Tool Integration Service
  - " Repository Creation/Modification
  - " Environment Definition/Modification
  - " Environment Deletion

A.3 Service Mapping

Effort Name: SLCSE

Service Name: Replication/Synchronization Service (7.12)

Applicable definitions and mapping guidelines which apply to this section are provided in Section 4.2.1 of this Mapping Guideline document. Fill out one form for each sevice listed in Section A.2. If your model has additional services not covered by the list in Section A.2, then please attach additional forms for these. Add New Service to the Service name at top of form.

A: Does your ISEE effort support the overall concept of this service?

_____ Not applicable (Discuss in comments below)
_____ No (Discuss in comments below)
_____ Supports alternate concept (Discuss in comments below)
_____ Unable to determine (Discuss issue in comments below and identify reason as to ambiguity or incompleteness in NIST model or as to ambiguity or incompleteness in your own effort.)
__X__ Yes (If yes, continue below)
  _____ Fully implements described set of services
  __X__ Implements a subset of services (List unsupported services)
  _____ Implemented as part of another service (Give reference to these other services)
  _____ Implements a superset of services (Give additional services supported)
  _____ Other (Discuss in comments below)

B: Dimensions (Check all dimensions that are mapped for this service. Include a "Service Dimension Form" for each such mapped dimension for each service.)

__X__ Conceptual [required]
__X__ Operational [required]
__X__ Related services [required]
_____ External [recommended]
_____ Data [optional]
_____ Rules [optional]
_____ Types [optional]
_____ Instances [optional]
_____ Metadata [optional]
_____ Internal [optional]

COMMENTS:

This service is not provided for Infrastructure Database or Project Files Hierarchy Objects, since they are never replicated within a distributed environment for any constructive purpose in the context of SLCSE.

A.4 Service Dimension Form

Effort: SLCSE

Service: Replication/Synchronization Service

Dimension: ICE - Conceptual

Description of how dimension applies to this service:

The SLCSE provides a Replication/Synchronization Service for Project Database Objects.

Within a heterogeneous network of computer nodes, it is possible for remote applications to "check-out" a subset of information contained in a SLCSE Project Database, manipulate that subset of information, and to "check-in" the modified information to the Project Database. Database integrity is important in a multi-user environment such as SLCSE, and therefore, it is possible to "lock" the instances that are checked-out, and "unlock" them when they are checked back in.

This facility is provided using a client-server architecture, where a client process on the remote node requests (over the network) a subset of data from a server process running on the host platform of the Project Database. Both the client and the server work through an interface to the Entity-Relationship Interface (ERIF) called the High Level ERIF (HLERIF), which is a highly portable Ada package. This package provides the capability to operate within the memory constraints of the remote computer via efficient, self-automated "swapping" operations to and from the local file space, and also provides data file formats that can be relatively easily transformed into the format required for use by a native application.

Give an example of this dimension for this service:

The SLCSE Project Management System (SPMS) is an example of a system that uses the Replication/Synchronization Service provided by SLCSE, as described above. The SPMS contains Commercial Off-The-Shelf (COTS) project management tools implemented on a Macintosh workstation that communicates to the SLCSE Project Database over the network. The Project Management Assistant (PMA) facet of the RL Knowledge-Based Software Assistant (KBSA) program and the QUality Evaluation System (QUES) are other examples where this service was applied for tools implemented on workstations.

**********End of Service Dimension Form**********

A.4 Service Dimension Form

Effort: SLCSE

Service: Replication/Synchronization Service

Dimension: ROD - Operations

Description of how dimension applies to this service:

The basic set of operations (create, query, update, and delete) applicable to this service for the Project Database are provided by the "High-Level Entity-Relationship Interface (HLERIF)", and are listed below:

Create:
"Add_Monitor_Action"
"Duplicate"
"Insert"

Query:
"Attribute_Error_Message"
"Collection_Error_Message"
"Condition"
"Count"
"Finalize"
"Find_Backward"
"Find_Forward"
"First"
"Get"
"Get_Current"
"Get_Error"
"Get_Instance_Storage"
"Get_Monitor_Action"
"Get_Next_Event"
"Get_Swap_Count"
"Get_Test_Error"
"Goto_First"
"Goto_Last"
"Goto_Next"
"Goto_Previous"
"Hlerif_Error_Message"
"Image"
"Initialize"
"Instance_Error_Message"
"Last"
"Local_Collection_Exists"
"Login"
"Logout"
"More_Errors"
"More_Monitor_Actions"
"More_Test_Errors"

"Print"
"Retrieve_From_Local"
"Retrieve_From_Slcse"
"Retrieve_Monitor_Actions"
"Test_Servers"
"Value"

Update:
"Save_To_Local"
"Save_To_Slcse"
"Set"
"Set_Instance_Storage"
"Set_Matching"
"Sort"

Delete:
"Delete"
"Destroy"
"Destroy_Local_Collection"
"Remove_Monitor_Action"

Give an example of this dimension for this service:

On a Macintosh workstation, retrieve from SLCSE all entities of the PROBLEM entity type, passing the Retrieve_From_Slcse operation the boolean value of 'True' for the "Reserve" parameter. This locks these entities while the retrieving application operates on the local entity collection until the Save_To_Slcse operation is used with a boolean value of 'True' for this operation's "Release" parameter.

**********End of Service Dimension Form**********

A.4 Service Dimension Form

Effort: SLCSE

Service: Replication/Synchronization

Dimension: Relationships Between Services

Description of how dimension applies to this service:

An Entity-Relationship model of the dependencies between each of the SLCSE services was developed to determine the relationships between services. Each service was modeled as an entity with various "depends_on" relationships to other services. An analysis on the model using the SLCSE analyzER tool resulted in the generation of forward and backward "trace" reports that were optimized to eliminate redundant relationship information. Forward trace reports on a service show the services that are required by the service. Backward trace reports on a service show the services that require the service.

This service requires the following services which are provided by SLCSE:

```
OPTIMIZED TRACE REPORT
========= ===== ======

TRACE ON ENTITY 7_12_replication FORWARD
     9 LEVEL RELATIONSHIP
          -- AS A MEMBER OF [all] SUBSETS.

7_12_replication (FORWARD) [object_management]
  1-  7_12_replication   depends_on   7_2_data_storage
     -->2-  7_2_data_storage   depends_on   11_1_message_delivery
     -->2-  7_2_data_storage   depends_on   15_1_2_common_data_descr
     -->2-  7_2_data_storage   depends_on   7_16_global_schema
        -->3-  7_16_global_schema   depends_on
15_1_2_common_data_descr
        -->3-  7_16_global_schema   depends_on   7_1_data_model
```

This service is required by the following services which are provided by SLCSE:

```
OPTIMIZED TRACE REPORT
========= ===== ======

TRACE ON ENTITY 7_12_replication BACKWARD
     9 LEVEL RELATIONSHIP
          -- AS A MEMBER OF [all] SUBSETS.

7_12_replication (BACKWARD) [object_management]
  1-  15_1_6_consistency_mgt   depends_on   7_12_replication
```

CIS Experience in ECMA Reference Model Mapping

- CIS committee / ATIS interface spec

- time taken:
  - ~3 hours in committee
  - ?? hours recording mapping document (not finished due to illness & workload)

- first impression: voluminous format

- second impression: mapping object oriented architecture requires awkward splitting of services -- but reasonable for purposes of discussion

⇨ RM is for discussion / comparison
It is <u>not</u> an architecture
Avoid the tendency to confuse the two

Specific weaknesses of RM

- ambiguities in partitioning
  → what goes where?
  e.g. fuzzyness between
  Data Repository, Data Integration,
  Communication, Process Mgt. services

  → ambiguity leads to mappings
  which can not be compared

- nowhere to put function mapping/binding
  [now NIST RM has function attachment]

- tool / service distinction is quite artificial
  [esp. when tool functionality is installed as
  a method in the environment/repository]

- because services are implemented as class
  methods, their specific definition is not a
  hard-wired part of the framework (meta-framework?)

- tool registration / installation for O-O

SUMMARY OF REFERENCE MODEL EVALUATION

Marvin V. Zelkowitz
Computer Systems Laboratory
B248 Technology
National Institute of Standards and Technology
Gaithersburg, MD 20899
(301) 975 - 3342
mvz@swe.ncsl.nist.gov

Acknowledgements

- Special Thanks to ...
  - Geoff Clow, SofTech, Inc.
  - Anthony Earl, HP Laboratories
  - Barbara F. Meyers, IBM Corporation
  - James R. Milligan, Rome Laboratory

Hal Pierson SPK

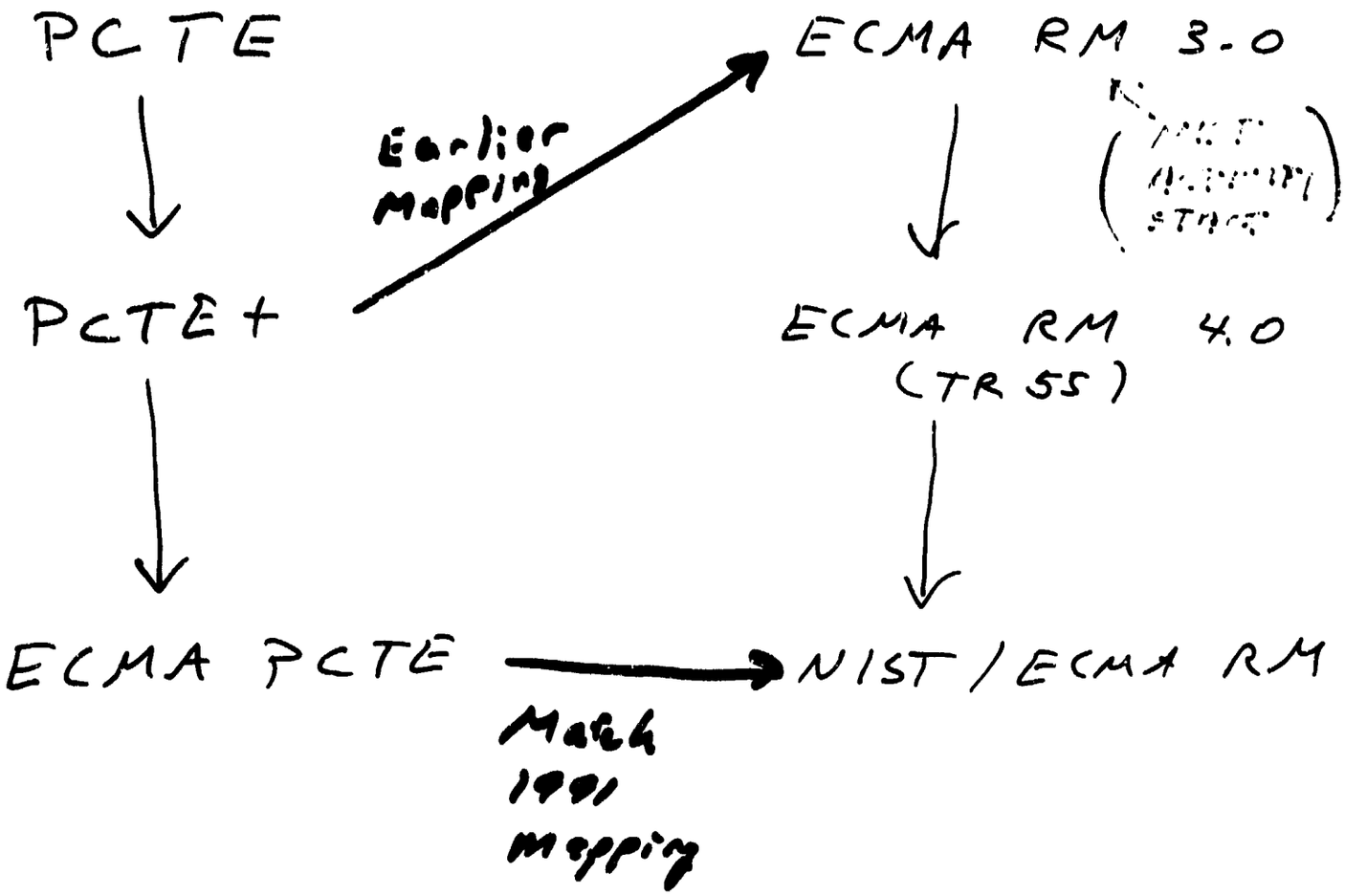
- The mapping forms were a significant undertaking and NIST and the NIST ISEE Working Group appreciates their time and effort in helping in this activity.

Disclaimers

- **Purpose of mapping exercise is to evaluate the effectiveness of the reference model to describe environment frameworks.**

- **The purpose is NOT to evaluate any individual framework.**

- **Opinions expressed in this talk are the personal observations of the author and do not represent NIST, the NIST ISEE Working Group or any of those involved in the mapping exercise.**


PCTE
ECMA RM 3.0
Earlier Mapping
PCTE+
ECMA RM 4.0 (TR 55)
ECMA PCTE
NIST / ECMA RM
March 1991 Mapping

Process

- Reference model based upon a set of services. Each service described by up to 13 characteristics (dimensions).

- A mapping form was developed.
  - Each service described on a separate form.
  - Three dimensions were requested for each service.
    - Conceptual -- What service does
    - Operations -- How it does it
    - Related services -- Impact on other services in framework

- Analysis based upon forms received from CAIS-A, ECMA PCTE, AD/cycle and SLCSE

Mapping Process

March Meeting

- Developed Form
- Discussion of reference model
- Sample mapping of one service with feedback
- Each mapper had a liaison with ISEE WG
- One month to do mapping

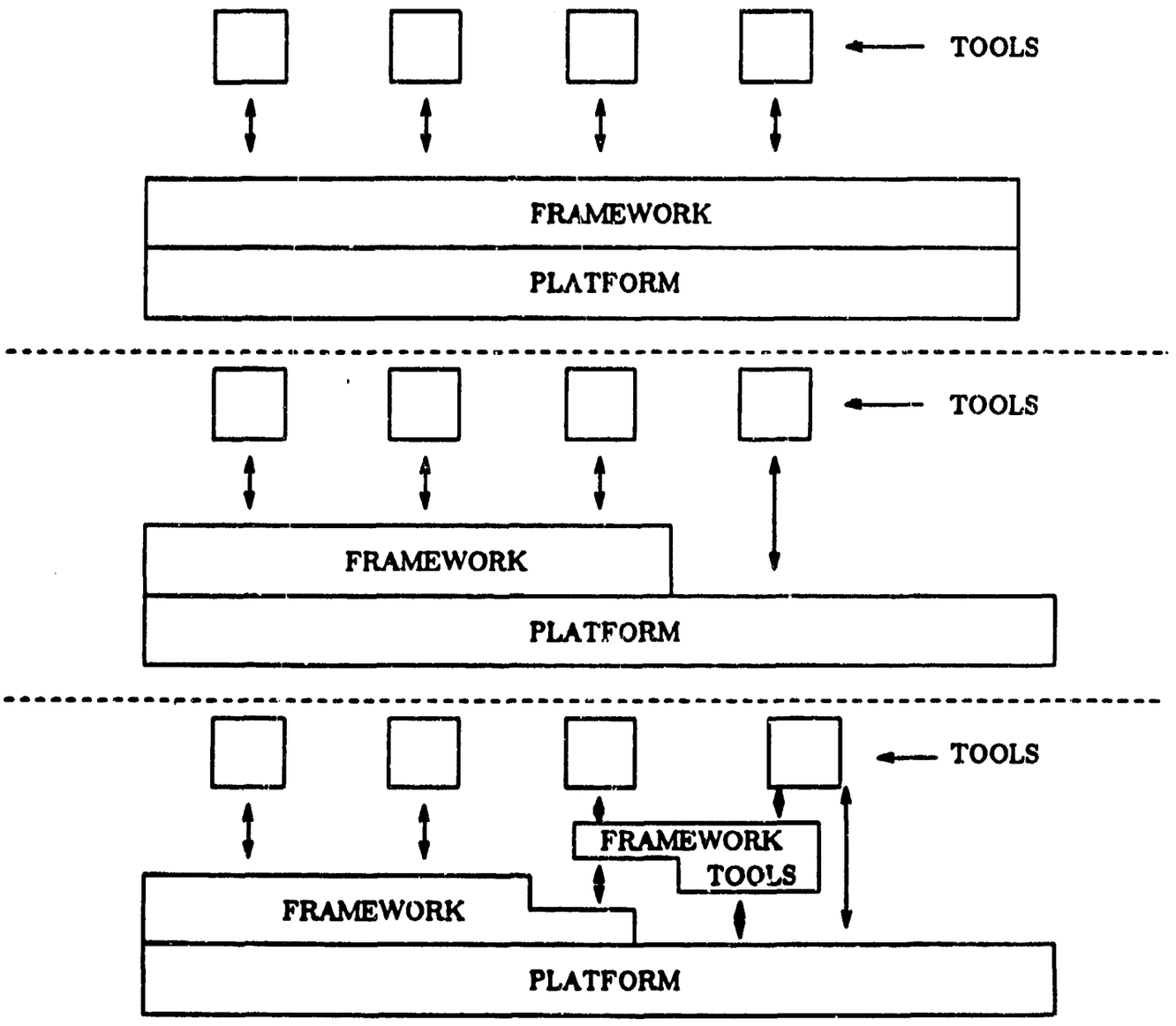
June Meeting

- Mappers believed process was useful and sucessful
- Kickoff meeting valuable
- SLCSE, ECMA, PCTE and CAIS-A mapping forms available through PSESARCH@NADC.NAVY.MIL archive

Positive Feedback to Mapping

- Reference model useful
  - to understand and explain architectures
  - as a guide to improve system's descriptions/ documentations
  - as a common basis for discussion
- All serviced seem applicable to SEE frameworks
- New Services suggested

Environment [redacted]



Issues raised by mapping - 1

- Group assigned responsibility to respond to item
  - EC - NIST ISEE Executive Committee
  - PL - Platform Services
  - OM - Object Management Services
  - TM - Task Management Services
  - UI - User Interface Services

- Data dimension possibly redundant [*EC*]
  - Data relates to internal, external and instances.
  - Proposal — Delete Data, [redacted] Merge Types and Metadata dimensions

- Tool Registration — Too distributed. Merge into one chapter [*PL*]

- [redacted] Inconsistent in handling underlying operating systems services (e.g., process support, communication, security, backup, UI, etc. when under control of the underlying platform) [*EC*]

- State monitoring (in OM) and event monitoring (in TM) seem similar. Should there be a common service? [*TM*]

Issues raised by mapping - 2

- Data interchange (in OM) and archive service (in OM) seem similar. Merge? [*OM*]
- State monitoring triggers when data states change in repository. Some frameworks include relationship changes also (e.g., full pre- and post-conditions on events). Extend state monitoring? [*OM*]
- Frameworks versus environments — How much should reference model cover? (e.g., How much of degree of tool integration is part of reference model?) [*PL*]
- Security models, role of framework security, and relationship of security chapter with access control services needs to be addressed. [*PL*]
- User Interface sections obviously incomplete. How much can and should be specified? [*UI*]

Issues raised by mapping - 3

- All four sample mappings are based upon an ER data model. Reference model needs to be checked against an object-oriented framework. [*OM*]
- New services need to be considered [*ALL*]

- Data Integration - Every tool supports it to a different degree.
- Add Services for environment/framework definition, tailorability and update.

Additional Services to consider

- Time
- Error reporting
- Framework definition/modification (Changing roles, names, tools, personnel, etc.)
- Tool integration (encapsulation "wrappers," "plug and play" tool slots)
- Object Management Navigation
- Function integration
- 2-way communication services

INTEGRATION

Definitions

Categorizations

Levels

→ BREAK

Incorporation in the RM document

Integration is ...

treat process and its support mechanisms as coherent parts of single complex system (Thomas & Nejmeh)

achieve synergy and thus productivity (Brown & McDermid)

user efficiency depends on coordination of interfaces of different tools for consistency (Nanard, Nanard, Pingand)

deals with cooperation of functionalities provided by different tools (ibid)

provides [an illusion of?] homogeneity (B&M)

ultimate goal involves collecting information from multiple tools and deriving new information not available from any single tool (Wasserman)

supports (static or dynamic) combination of tools so that they harmoniously cooperate

allows reconciliation of dual aspects of Project Management: Tasks & Deliverables (EAST)

... and Integration is...

minimum: requires agreement on at least one dimension.

effective: requires agreement on all 3 dimensions

(Wasserman)

use of same components or common services

a property of a relationship between two elements (T & N)

(n?)

semantics, data, behavior

ICSE13 Panel	**Integration in SEEs**	TRW

Distinction between User Interaction and Architecture perspectives:

- Environment User Interaction - how the user communicates and perceives the environment to operate in order to perform life-cycle activities/tasks.
- Environment Organization - how the environment is architected and how its various components interoperate.

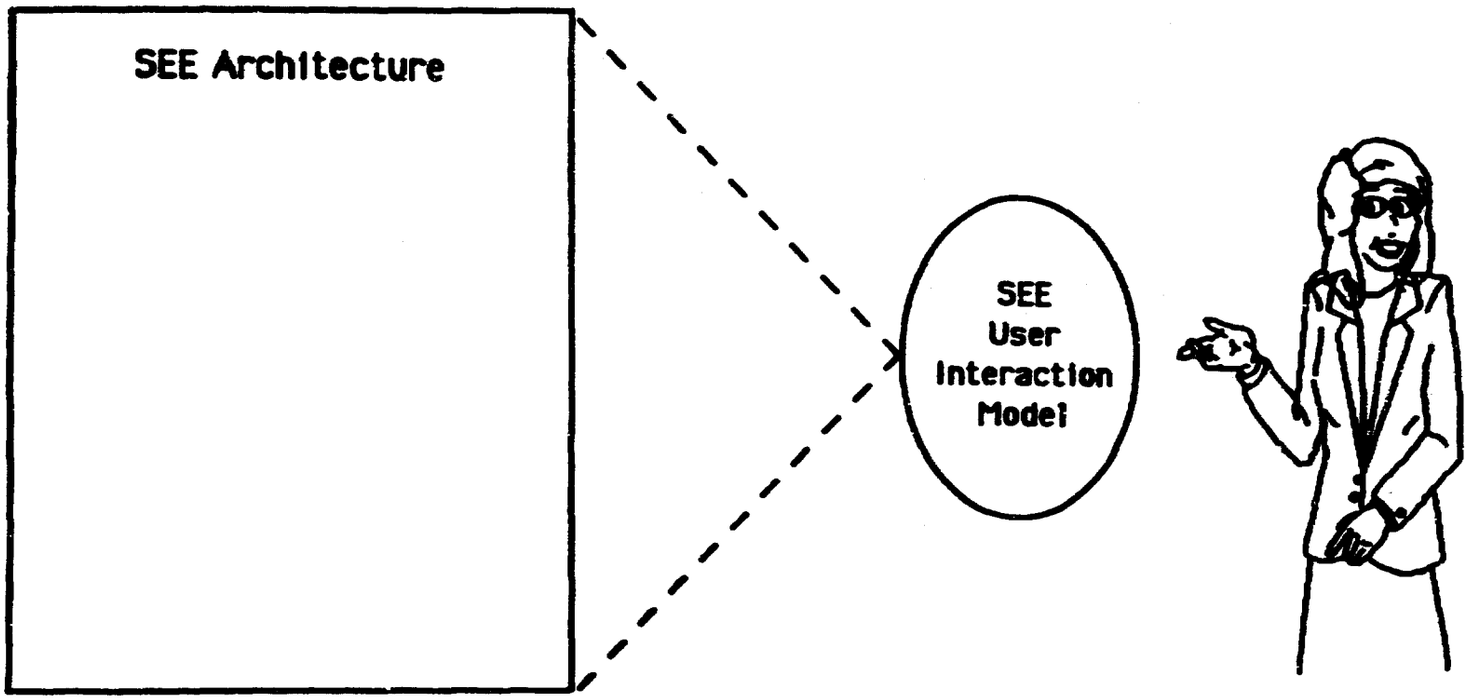
164

Definition: *Tool* is a piece of software that can be invoked and that performs some function in support of the life-cycle project users.

M. PENEDO
TRW

ICSE13 Panel	**User Interaction and Architecture perspectives**	TRW

Integration in SEEs



Env./Tool Builder Perspective:

- Platform Integration
- Data Integration
- Control Integration
- Communication Integration
- Process Integration
- User Interface Integration
- Tool Building Integration

Env. User Perspective:

- Invocation Integration
- Object Integration
- Tool Kit Integration
- Process Integration
- Presentation Integration

- **Invocation Integration.** Ability to invoke tools in the same manner, independent of their physical location (may reside on different machines).
- **Object Integration.** Ability to access the same objects via distinct tools.
  - objects produced by one tool can be accessed/modified by another tool;
  - multiple simultaneous views/displays of the same object are kept up to date.
- **Tool-kit (Function) Integration.** Ability to have access to functionally related tools from within a tool, based on changes to objects or specific tool actions.
- **Process Integration.** Ability to access tools based on a pre-defined development process.
- **Presentation Integration.** Ability to interact with tools with similar screen appearance and similar modes of interaction.

- **Platform Integration.** The extent to which tools work on the same platform (e.g., virtual operating system, network services, etc).
- **Data Integration.** The extent to which data is shared/communicated among tools.
  - file system
  - canonical data forms (e.g., CDIF)
  - translation mechanisms
  - common object store/common schema
  - multiple object stores with interoperating mechanisms

- **Control Integration.** The ways in which tools can be executed (i.e., registered, activated, de-activated, suspended, resumed) and combined.
- **Communication Integration.** The ways in which tools communicate with each other and the environment (i.e., infrastructure services).
- **Process Integration.** The extent to which tool ordering and execution is based on a separately defined automated software development process.
- **User Interface Integration.** The extent to which tools rely on common services for user interface functionality.
- **Tool Building Integration.** The extent to which tools rely on common services/automated mechanisms in support of tool building.

Copyright © 1991 Cadre Technologies Inc.

Tool Integration Issues

Styles of Integration (cont'd)

Control:

- Tools retain control of their own data
- Tools able to initiate requests and accept requests
- Event detection, translation & delivery

Semantic:

- Each tool has own semantics for data & behavior
- Integrating two tools yields a third tool with new semantics
- Framework must be able to model new semantics
- Reconcile conflicts in data models & behaviors

Integration
what – why – how

putting parts together
getting parts to work together
- smoothly
- correctly
- easily
- cost-effectively

goals:
perspective (user, provider, ...)

transitivity?
service vs. mechanism?

QUESTIONS

1. Is Integration a layer?
   who "owns" it
2. Is Integration a service?
3. What are the things that affect the fundamental architectures of tools?

4. Are there services which are unique to integration? Yes, e.g., CDR, triggers

Yes: Encapsulators, filters

5. What is the role of "common services" in integration?

1. Is Integration an RM dimension

2. Is Integration a policy? ~~agreement~~

3. Should there be a separate section for Integration Services?

3a. Discuss only whats known today
- issues, laying out space
- relative to existing services

4. Should Integration be covered in a separate model/document?

4a. Do proper development of Integration reference model.

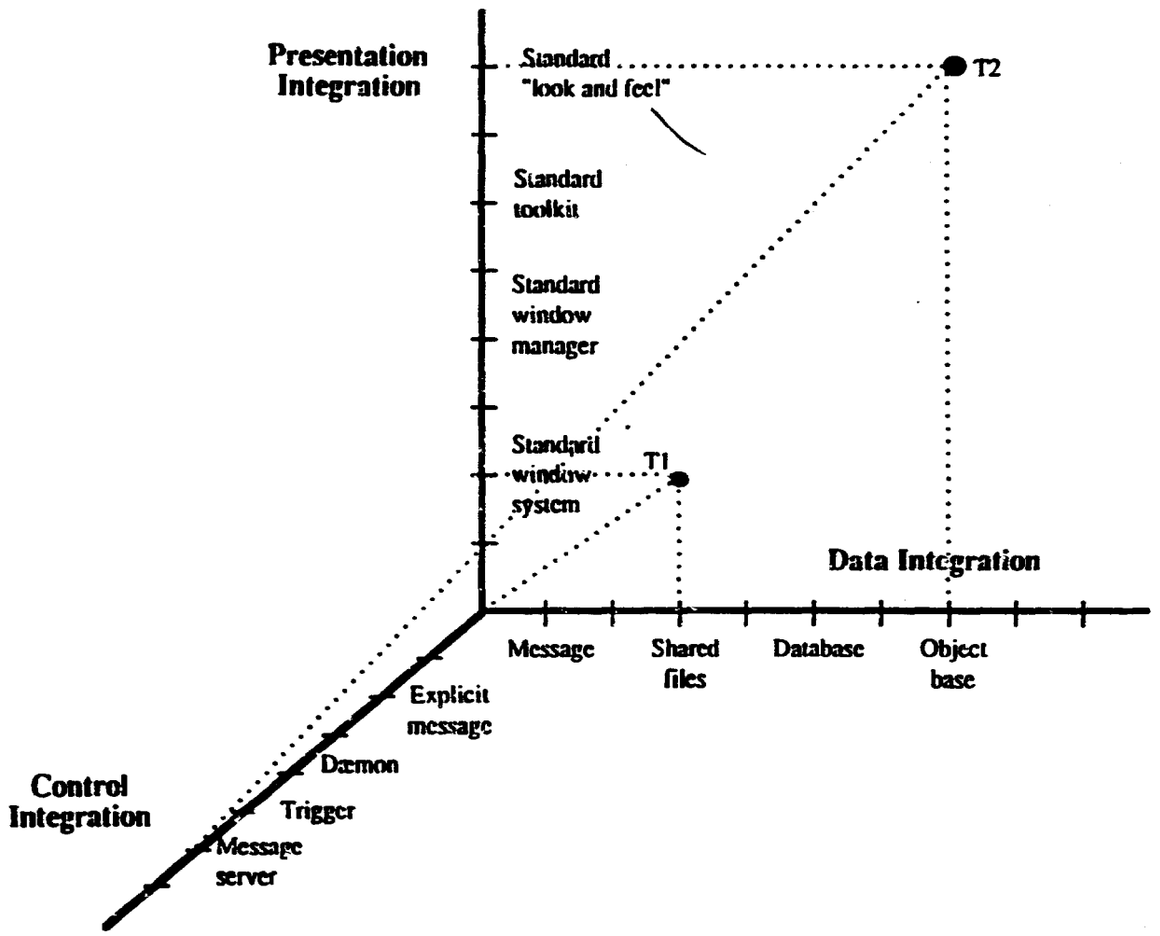
— an RM is descriptive

— Integration is one of several ways of looking at/describing frameworks

---

Pragmatics: ① dont create a part of the document that is changing faster than the rest
② who is working on what parts of such an RM?

Toward Standard Frameworks for Tool Integration

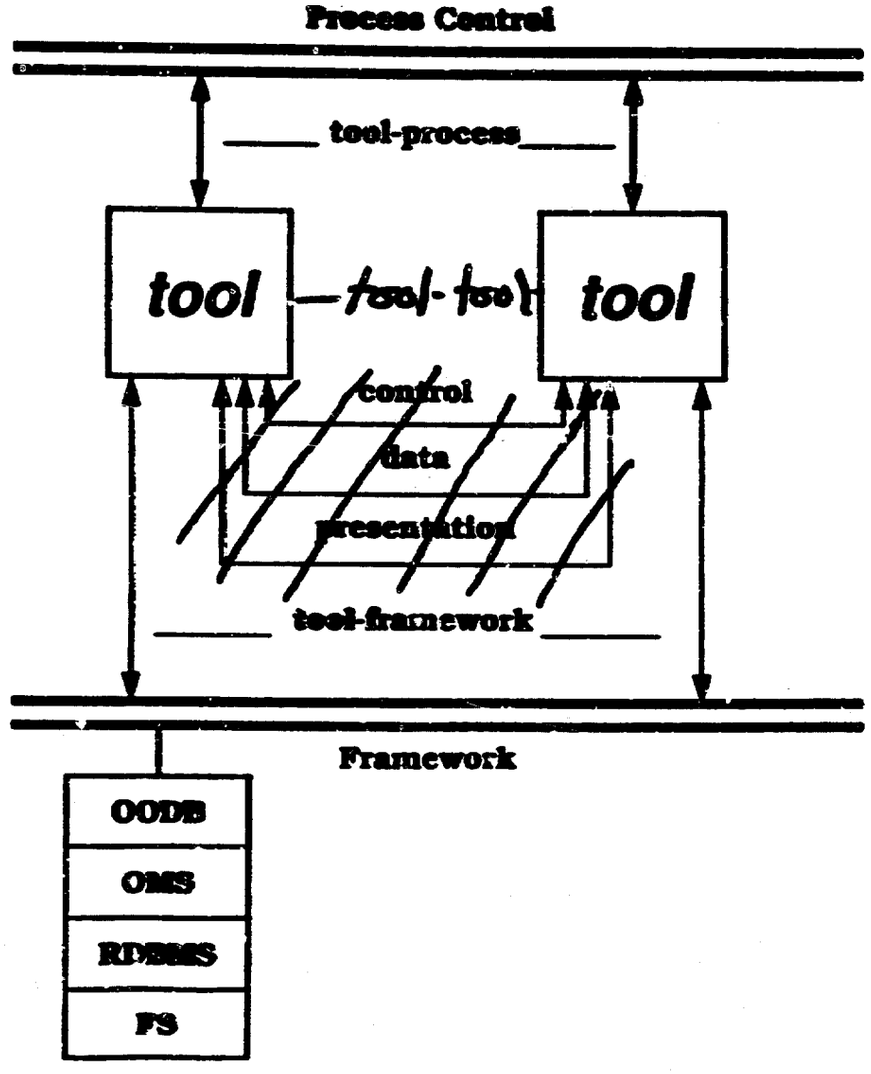


Copyright ©, 1991,
Interactive Development Environments, Inc.

INTERACTIVE DEVELOPMENT ENVIRONMENTS

Dimensions of Tool Integration

data
control
presentation
process



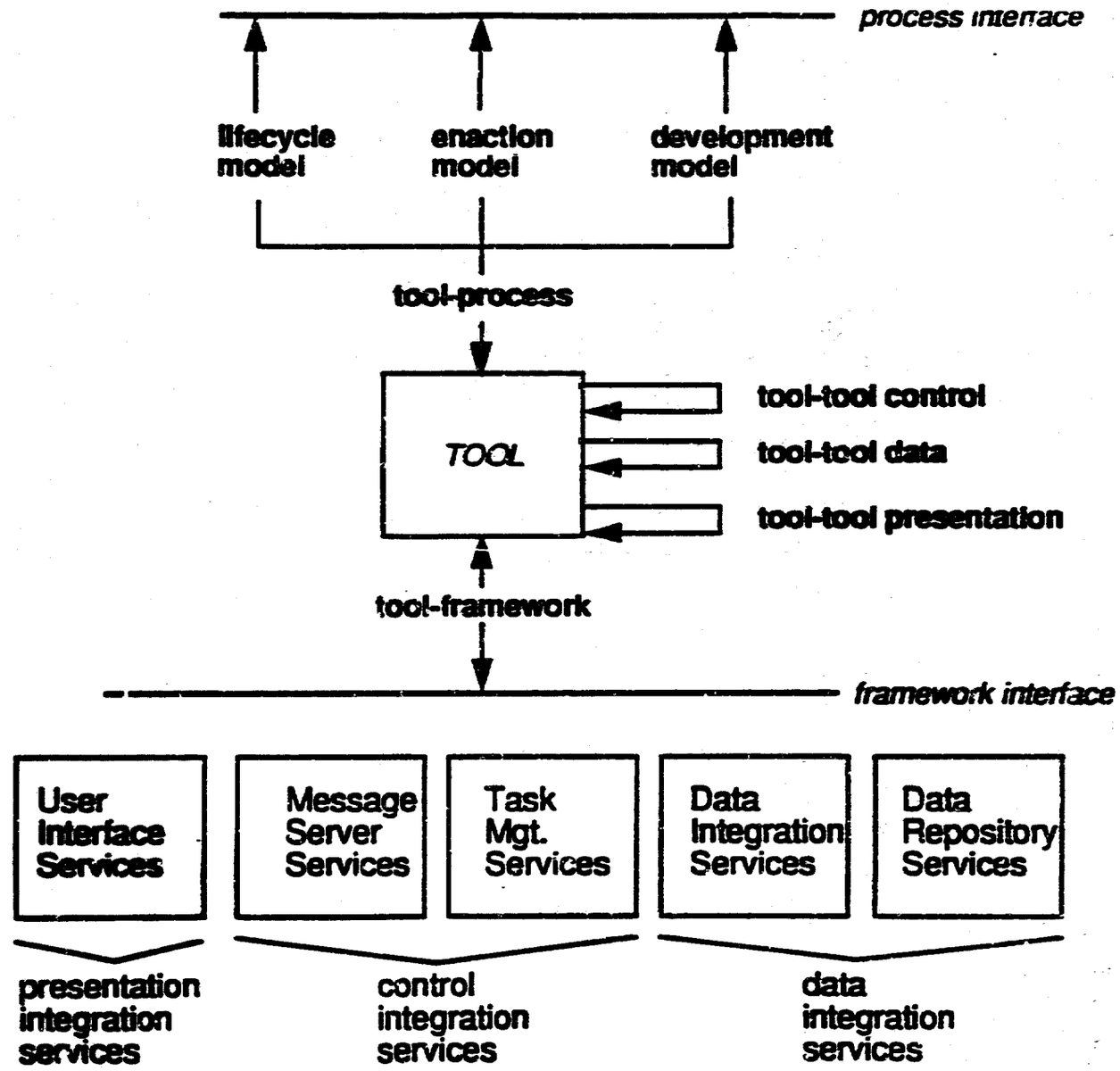
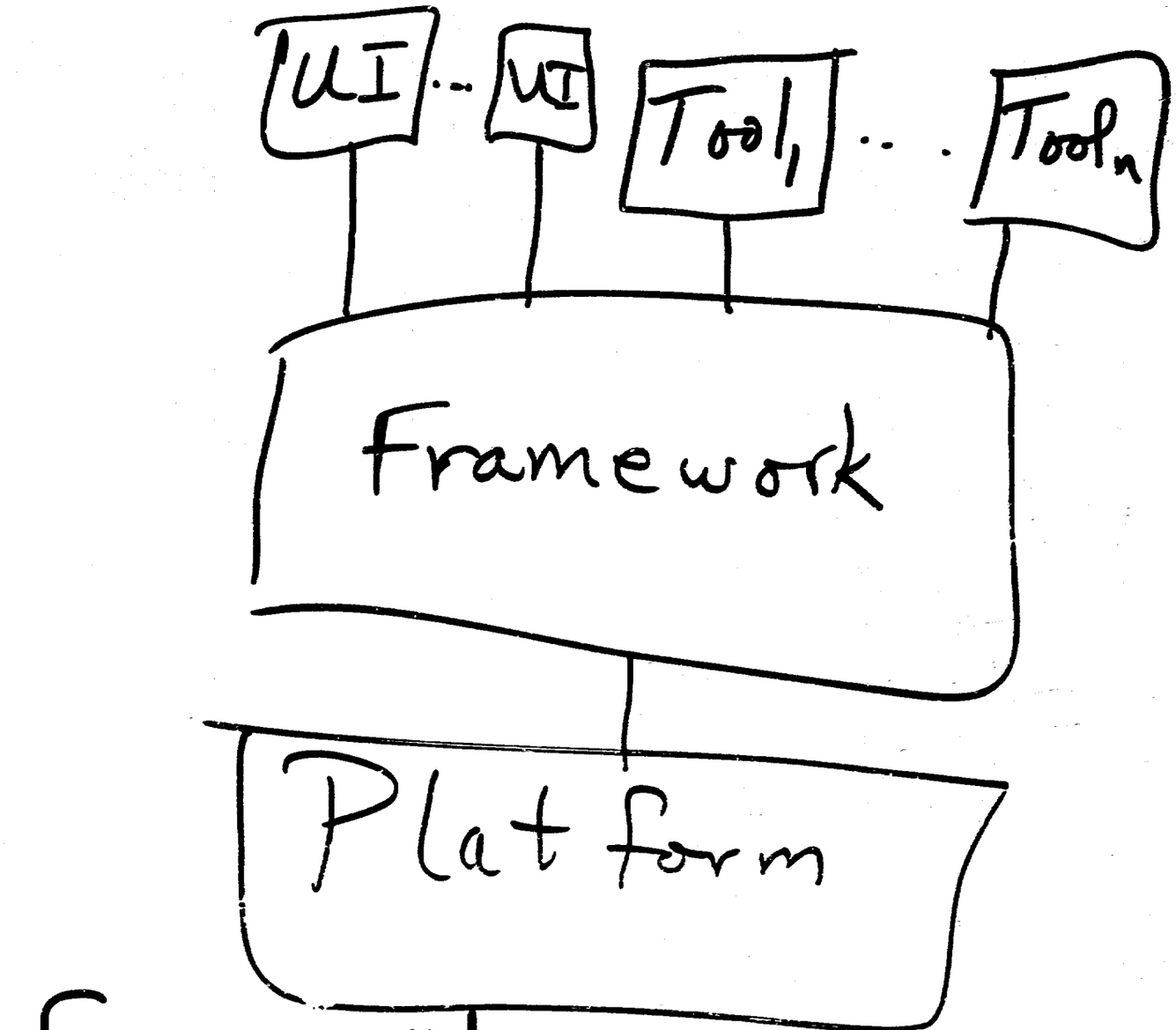
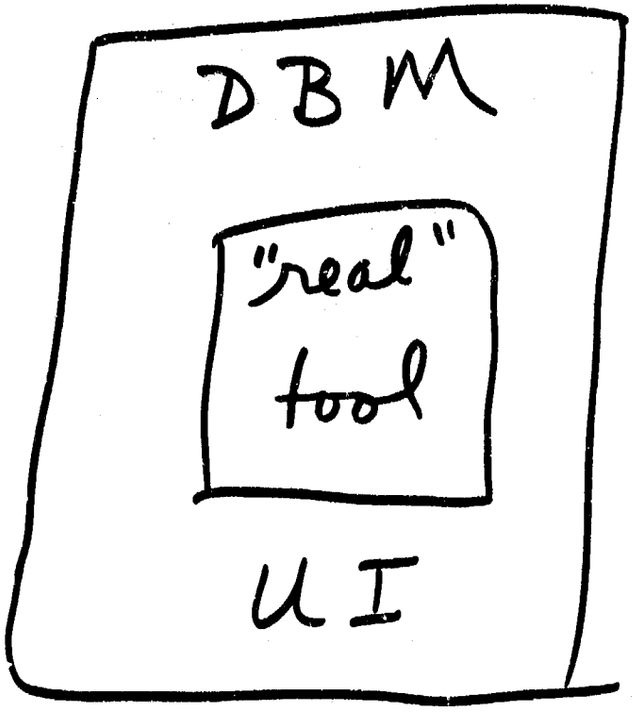


Figure 3-1 Types of Integration

platform services, framework services, and environment services. The following relationship describes the way these concepts are related:

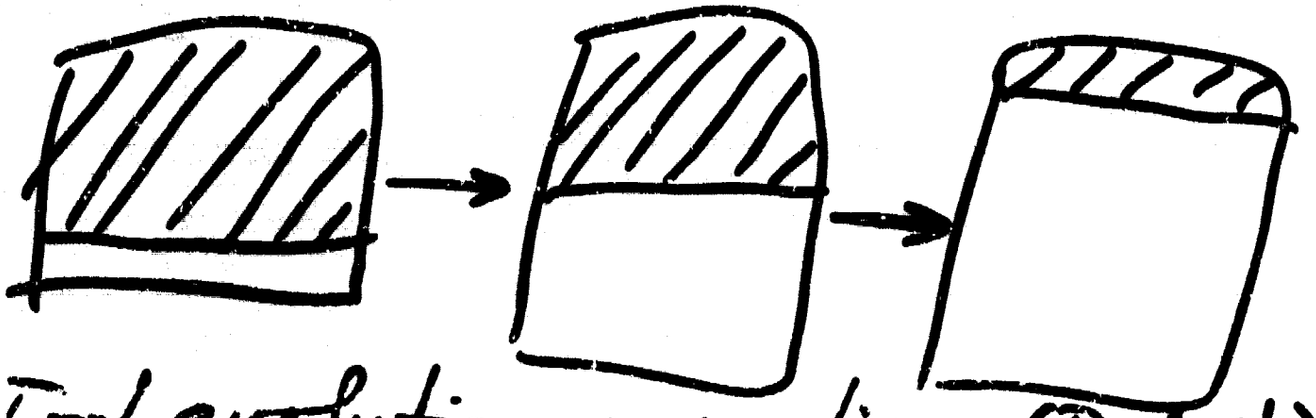


from 2nd NIST/SEE Wkshop



Tool

Tool evolution over time (Rudmik)



Tool evolution over time (Rudmik)

DATA INTEGRATION

managing data as a consistent whole

(m) common data representation

common models (a)

(m) central repository, format, storage, carrier

to preserve semantics and provide tool interoperability

sharing of data and managing relationships among objects from different tools

files, IPC

? no uncoupled? information redundancy (dep. on goals)

type/data interoperation

Geoff Lewis:
linkage
interchange
sharing

sharing: when / how often

PROCESS

to ensure that tools interact well in support of a defined process

way of guiding end user

tool invocations are ruled by process def.

process implements the organizations life-cycle

must support resource & role concepts

life-cycle model, enaction model, development model

facilitate interworking of teams

INTEGRATION MECHANISMS

DI = Data Interoperability

PM = (Life-Cycle) Process Management

DATA INTEGRATION MECHANISMS

Internal forms for programs, (by-) products of life-cycle process, documents [e.g. IRIS, DIANA]

data translators between OMSs [-> DI]

data exchange mechanisms, canonical data forms [e.g., CAIS-A CEF, CDIF, P1175, IRDS export/import; -> DI]

common schemas [e.g., PMDB]

type interoperability mechanisms [e.g., SLI]

name conflict resolution, name servers [covered by OM]

schema migration, exchange, and translation

self-describing data forms [e.g., SCML, CAIS-A CEF, CDIF; -> DI]

data translators

(data) model translators

semantic integrity

method (OO) inheritance

merging data, schemas, and models(?) [note: related to translation]

synchronization across duplicates & (distributed) replicates [covered by OM]

PROCESS MANAGEMENT INTEGRATION MECHANISMS

control communication mechanisms [e.g., state mechanisms, event mechanisms, concurrency services, message services, dynamic multiplatform (RPC+) comm. services, invocation, tool interaction across platforms and languages (homogeneous to heterogeneous), network services; -> PM]

dynamic sychronization of multiple views, multiple contexts, multiple presentations

control executive model/tool composition paradigm [note: this is closely related to communication mechanisms; e.g., transactions; pipes, scripts, attribute monitors, general life-cycle process model enactment; -> PM]

control vitual operating system

process translators (?)

(message) protocol translation

life-cycle process (method) integrity

USER INTERFACE INTEGRATION MECHANISMS:

user interface "look and feel" packages

MISCELLANEOUS:

tool encapsulation/de-encapsulation (?)

conventions [e.g., for naming]

common programmatic (call) interfaces

tool registration and deregistration

user customization & extension features/packages

policy-related [note: related to semantic integrity; e.g., management, security]

INTEGRATION IS ABOUT AGREEMENT

a precondition for integration

- Framework services provide the vocabulary in which to express the agreement

  OMS — schema
  RM — Information Model

  BMS — message formats & semantics

- Framework services enforce the some agreement

  DIANA tree service support

Integration is about parts/entities working together towards some goal.

- A service that reduces the consequence of disagreement

  HP Encapsulator

  tool writer's platform assumpt.
  vs
  actual platform

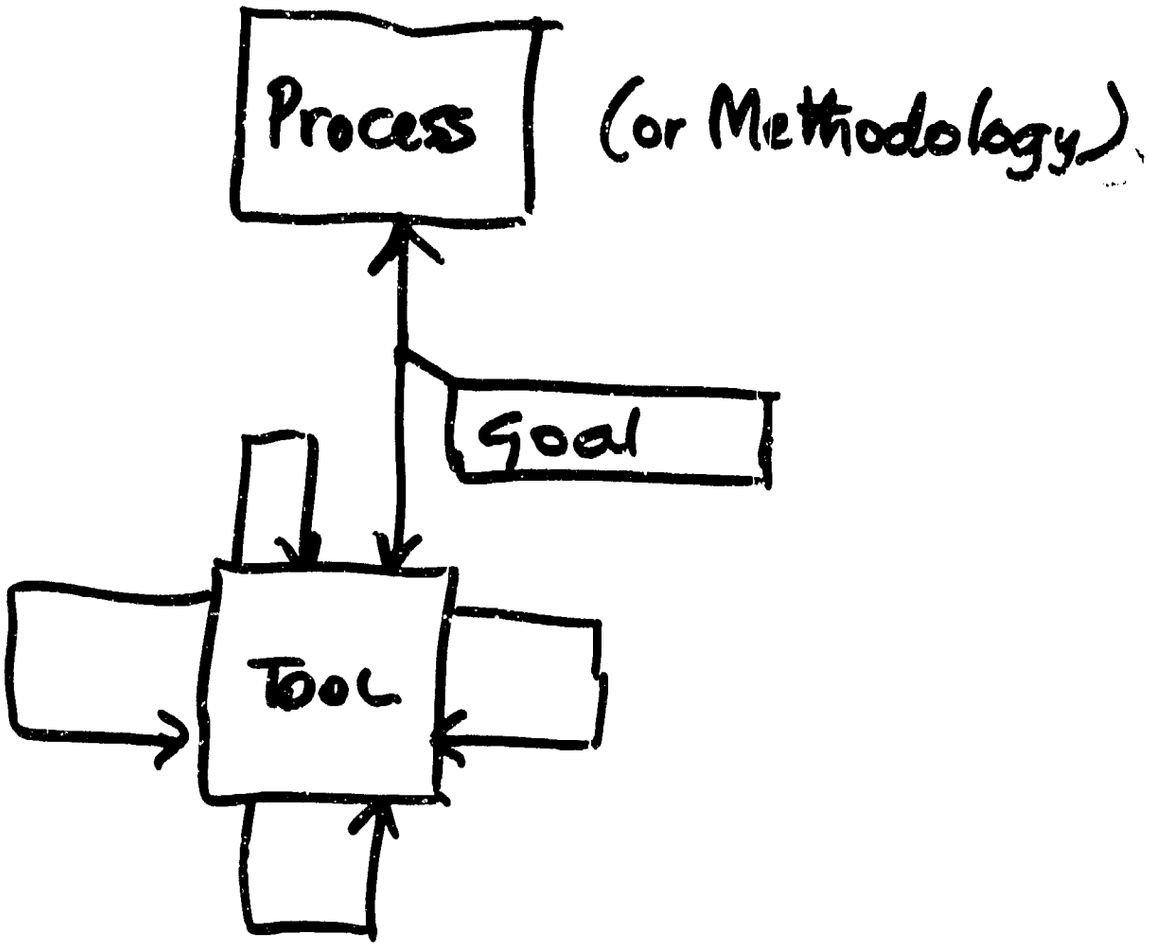
  Data translation / Formatting

RATIONALE FOR FRAMEWORKS

- Factorisation

- Factorisation with enforcement (not "good citizen" assumption)

- Integration
  -
  -
  -


Process
(or Methodology)
Goal
Tool

JUSTIFICATION FOR INTEGRATION

WE ARE TALKING ABOUT A REF. MODEL FOR AN ISEE, NOT A SEE

M. PENEDO

Object Management WG

6/5

Summary

- Workers:
  - J. Peterson, TRW
  - N. Goldman, USC/ISI
  - B. Cureton, SUN Microsystems
  - T. Strelich, GRC
  - N. McQuage, Boeing/STARS

- Expertise/familiarity w/ RM:

  little ⟵ ⟶ very

- Activities:
  - one pass over services + dimensions
  - discussions
  - actual review of service descriptions
  - provide answers/solutions to issues
    - mapping exercise
    - reviews

ACCOMPLISHMENTS

- "GOOD" REVIEW OF EACH SERVICE DESCRIPTION

- SOLVED SOME ISSUES

- CREATED LIST OF ISSUES TO BE DEALT LATER

- ACTION ITEMS:

  LP - [SECTION BOSS FOR DOCUMENT
  DUE TO NARV-NIST EARLY JULY

  WG ATTENDEES - [FURTHER DATA THROUGH EMAIL by next Wedn.
  REVIEW FINAL WRITE-UP

DISCUSSION ITEMS (OM)

1. OS issues as related to "Process Support" service

→ clarification of service description

↓ "OS" process treated as object by OMS"

1a. Generalize Suria

2. Deleting "data" dimension

no apparent problems

3. Merging "types" + "metadata" dimension

titled new section "Types / Metadata"

4. Should "NOT APPLICABLE" dimensions be noted?

unclear - they may be applicable to specific systems

5. Unclear distinction between "State Monitoring" and "Constraint" Services ⇒ Maz"

DISCUSSION ITEMS

6. Data Model seems out of place
   => belongs in "Metadata" Service

7. Redundancy of "Tool Registration" service across RM => - deleted from OM grouping
   - included/Kept Framework Adm.

8. Object navigation => ~~get~~ in Query
   and/or Archive
   " Relationship
   " Data Storage

9. Data vs Task transaction => added clarification text

10. Data/DBMS-like administration services
    => future discussion

11. Configuration/Composite Object service-naming
    => [illegible]

12. Shall non-persistent data be addressed?

⇒ FUTURE DISCUSSION

13. Consistency in terminology

OM, DM ⇒ OM & DM Terms used as synonims

14. Heterogeneous object interchange

⇒ FUTURE

ISSUES "Process Management Subgroup

1. Differences of Data & Event triggers
2. Task Transactions
3. New Services
   - Metrication

4. Framework / Definition / Modification Facility
5. Function / attachment integration
6. Identify classes of process info.
7. What to do about phf & wh terminology list CHECK conformance
8. Process operations
9. Revision of section structure & contents.
10. Granularity
   CHECK that the text does not assume coarse granularity

10

STRUCTURE OF CHAPTER

PROCESS MANAGEMENT SERVICES

- Introductory text

- Process Definition Services 2.1
  operation dimension should cover operations on processes

- Process Enactment Services 2.2
  operation dimension should cover operations on processes being enacted.
  includes selection of enactment agents

- "Visibility / Scoping" Services 2.3
  includes communication between enacting processes.

- Process State and History Services 2.4
  includes instantiated process description and its execution state.

Separate sections → Process Monitoring
→ Event Management (ex 2.5)

- Process ~~Supervision~~ Control Services
  eg scheduling
  auditing & accounting ↓2.6 other metrics too
  (as specializations of metrics)
  selection of which elements of history to record
  simulation / analysis — "what if..."

- Registration of enactment agents
  Resource management of enactment agents
  ~~Registration of~~

- Resource Management Services 2.7 2.8
  eg definition of resources
  - enactment agents
    people & machines
  - tools / process fragments

"Interface & Platform"
(a.k.a. Fatpoint)

Emphasis: Integration

Questions:

- Completeness of "other" services?
  - Tools
  - Message Services
  - Security
  - Framework Admin & Configuration
- Missing service groups?
- Where are the interfaces? (operations? external view?)
- What are the integration mechanisms?

Session this afternoon;
Integration plenary tomorrow

Interface & Platform

OLD		NEW	
9	Tools	9	Small
11 ←	Message } Communication	7	Med
13	Security	10	Much
14	Framework Admin.	11	Human
15	Integration	12	

9 Tools 9

1. Consolidate "Integration"
2. PCTE+ mapping comments [re: Tool Integr]
3. Need Integrated Tools section (SLCSE)
4. SLCSE mapping pp 180-205 [re: Tool Integr]

1. Move Tool Integration to Sec 12
2. Keep section (somewhere)
3. Rename "Domain-specific _____"
   "Life-Cycle Support _____"
4. Use software-oriented tools list as an example; others are CAD/CAM, Office Automation, ...

Communication

1. Communication protocols
2. I/O [postpone]
3. SLCSE mapping pp 242, 243

1. session protocols
   with/without state
2. Services
   - message delivery
   - RPC
   - "file" sharing
   - "pipes"
   - session dialogue

} connec/CL

Security

✓ Sec. 13.4 is not a service.

1. Location
   → as is
   - after TM
   - under Framework Admin
   - at end
2. Name:
   ? generalize to Policy Enforcement Svcs

Munch.

11 Framework Admin 11

1. Consolidate "Tool Registration" → Tools
2. Installation procedures → Env. Admin?
   incl. Tool Mgt — Integration?
3. Framework Definition / Modification
   → (Subenvironment ~~Support~~ (Views) Svc,
4. Environment Definition / Modification
   → Env. Admin
5. Environment Deletion
   → Env. Admin

from 11.3
OMS Schema Mgt → 5
Groupware Mgt → 5
Rsrc Regis. & Mapping → 5

Sec 11:

11.1 Metrication

Other Feedback

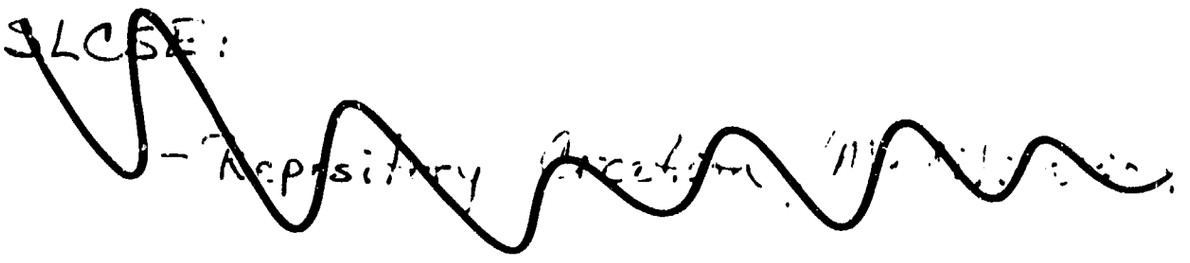
1. Services provided by underlying OS / Platform
   → Mapping Guidelines issue
2. Exception Handling / Error Handling services
   → 5.14, 6.5, Integration
3. Project Management Integration
   → TM
4. Tool invocation (incl. cmd. interp.?)
   → Tool Integration
5. Editors (textual, graphical)
   → Common Services / Encapsulation
6. Display / List Services
   → Common Services / Encapsulation
7. Help facilities
   → Common Services / Encapsulation
8. Time Services
   → ? (not I&P)
9. Repository Creation / Modification
   → Env. Admin
10. Tool encapsulation / de-encapsulation
    → Tool Integration

Mappers' Comments we didn't address:

Under "add function integration" (4.1):

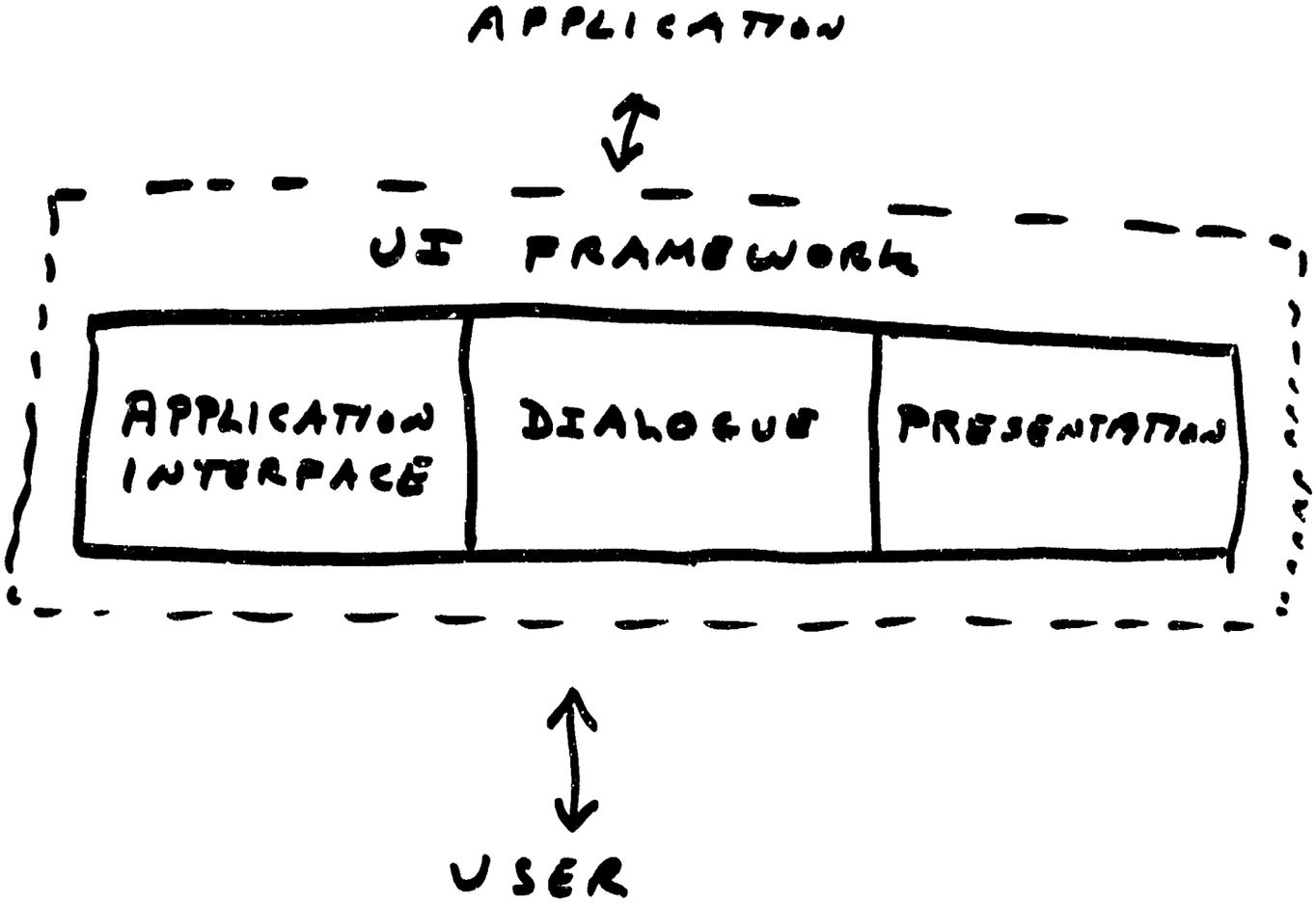
- interchange formats
- presentation styles

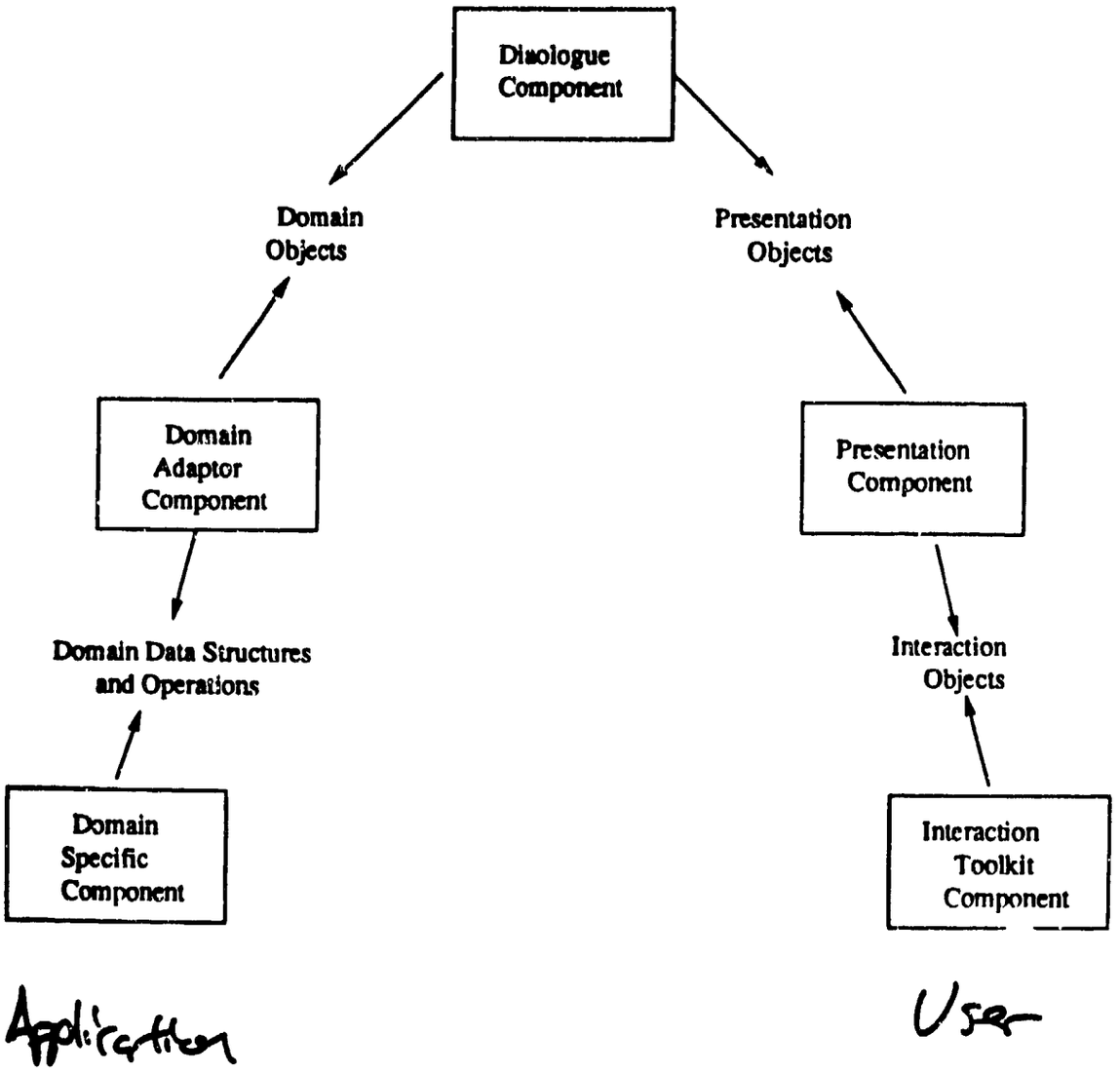
SLCSE:



User Interface Management Services

- Session Services
  - Create, Delete, Query, Modify, Register, Negotiate
- Security Services → framework Security Services
- Profile Services
  - User, Tool, Session, Role
- Name and Location Services (→ Internationalization)
- Transfer Services
  - Get/Put, Cut and Paste, Capture
- Metadata Services
  - UI Model
    - Create, Delete, Modify, etc.
- Dialog Services
  - Transaction, Concurrency, Functional Attachment
  - State Monitoring, Constraints
- User Assistance Services


APPLICATION
UI FRAMEWORK
APPLICATION INTERFACE
DIALOGUE
PRESENTATION
USER

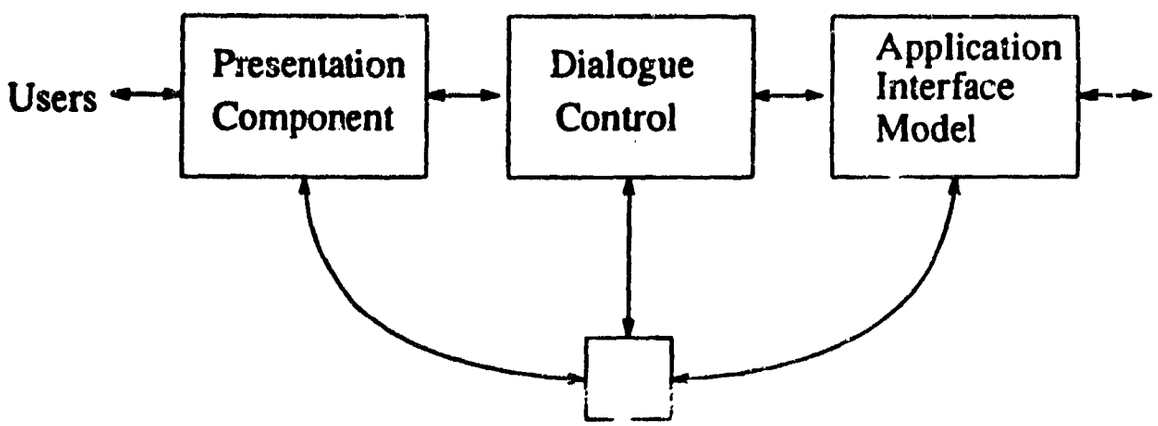

Dialogue Component
Domain Objects
Presentation Objects
Domain Adaptor Component
Presentation Component
Domain Data Structures and Operations
Interaction Objects
Domain Specific Component
Interaction Toolkit Component
Application
User

User Interface Services involve all aspects of the framework

Conceptual

- Seeheim Model

  Since 1982, the most influential user interface reference model has been the so-called Seeheim model, defined by a group of designers at a workshop sponsored by Eurographics and IFIP in Seeheim, Germany.



Summary of the 5th Workshop on Integrated Software Engineering Environments (ISEE)

July 26, 1991

William Wong
NIST/CSL
Gaithersburg, MD 20899
(301) 975-334
Fax # (301) 590-0932
wong@swe.ncsl.nist.gov

The Fifth Workshop on Integrated Software Engineering Environments (ISEE) in conjunction with the PCIS (June 3-7) and IWCASE (June 6) workshops took place in Redondo Beach, California on June 3-5, 1991. The workshops were hosted by TRW and Los Angeles SIGAda and sponsored by the National Institute of Standards and Technology (NIST), the U.S. Department of Commerce, the Ada Joint Program Office (AJPO), the U.S. Department of Defense, the International Workshop on CASE (IWCASE), and the Special Working Group (SWG) on Ada Programming Support Environment (APSE).

The goal is to identify and establish a consensus for the U.S. Federal government and industry to address the need for open system ISEE and software tools interface standards. The agenda of the workshops were designed to support each other's activities and all participants were encouraged to participate in all the workshops. This will provide a better understanding of the overall activities from the various working groups and will help to reduce the duplication of efforts and redundant initiatives and establish synergism between the participating groups.

The ISEE workshops objectives are:

- to identify and explore fundamental issues in ISEE areas;
- to identify the needed set of standards that define a comprehensive interface for integrating software tools;
- to develop guidelines on interface standards for ISEEs; and
- to provide guidance to Federal agencies in acquiring software development and maintenance environments.

The workshops were attended by approximately 100 software professionals drawn from the U.S. Federal government, industry, and academia. Participants also included European and Japanese software professionals.

The ISEE workshop primarily focused on:

1) enhancing the NIST ECMA ISEE Reference Model Technical Report by reviewing and rewriting new services, and identifying and defining the services related to integration, and

2) reviewing the results of the NIST ECMA ISEE Reference Model mapping exercise.

NIST is scheduled to publish the "ISEE Reference Model Technical Report", version 1.0 as well as the "Report on Summary of Results of the First NIST ISEE Reference Model Mapping" by the end of September or early October, 1991. Working Groups included:

- Object Management, led by Lolo Penedo, TRW;
- Process and Task Management, led by Hal Hart, TRW;
- Interface and Platform, led by Tricia Oberndorf, NADC; and
- User Interface, led by Bob Bagwill, NIST.

The general format included:

1) the NIST ECMA ISEE Reference Model tutorial session given by Anthony Earl of HP in UK,

2) the NIST ECMA ISEE Reference Model Mapping Exercise Summary chaired by Sandi Mulholland of Rockwell International, and the results of the ISEE reference model mapping exercise summarized by Marv Zelkowitz of the University of Maryland and NIST,

3) the Integration Services session chaired by Tricia Oberndorf of NADC, and

4) four parallel working group sessions.

This workshop was organized and coordinated by William Wong of NIST, Lolo Penedo and Hal Hart of TRW, and Currie Colket of AJPO. Special thanks are due to members of the NIST ISEE Executive Committee and other individuals for their valuable assistance and support in the planning of the workshop. They are: Bob Bagwill of NIST, Currie Colket of AJPO, Hal Hart of TRW, Roger Martin of NIST, Sandra Mulholland of Rockwell, Tricia Oberndorf of Naval

Air Development Center, Lolo Penedo of TRW, Ian Thomas of Hewlett Packard, and Marv Zelkowitz of the University of Maryland and NIST.

The 6th NIST ISEE Workshop is scheduled for October, 1991 in Gaithersburg, MD. NIST will be the host and sponsor of the upcoming workshop. The date and agenda will be announced at a later time. The lists of action items for the work to be accomplished before the next workshop, the summary of each working group, reference model mapping exercise and integration services sessions are included as follows:

a) summary of action items;

b) summary of schedule for completion of the NIST ISEE Reference Model document, version 1.0;

c) summary of status report from each working group;

d) summary of the NIST ECMA ISEE Reference Model Mapping Exercise session; and

e) summary of the Integration Services session.

a) **Summary of Action Items.**

Action Items	Responsible persons	Deadlines
Coordinate and prepare the final workshops participation list	Hal Hart Clyde Roby	6/21/91
Coordinate and prepare the Proceedings of NIST ISEE / PCIS / IWCASE Workshops	Clyde Roby Bill Wong	6/28/91
Update the NIST ISEE working group e-mail distribution list	Tricia Oberndorf	7/19/91
Working Group Status Reports:		6/14/91
Object Management	Lolo Penedo	
Process Management	Hal Hart	
Interface and Platform	Tricia Oberndorf	
User Interface	Bob Bagwill	
Integration session	Tricia Oberndorf	
Mapping Exercise session	Marv Zelkowitz	

Summary of the ISEE Workshop	Bill Wong	6/21/91
Coordinate with ECMA TC33	Bill Wong	ongoing
Forward new sections to Marv (the RM editor)	Lolo Penedo Hal Hart Tricia Oberndorf Bob Bagwill	7/15/91
Forward the RM document to NISTEC, Anthony Earl for review	Marv Zelkowitz	8/1/91
Review of NIST ISEE RM document (version 1.0)	Marv Zelkowitz Anthony Earl NISTEC	8/23/91
Forward the revised document to NIST ISEE, ECMA TC33, other relevant groups	Marv Zelkowitz Bill Wong	8/30/91
Coordinate with ECMA TC33 for comments	Bill Wong	9/5/91
Collect all the comments on RM document	Marv Zelkowitz Bill Wong	9/15/91
Finalize the RM document	Marv Zelkowitz NISTEC	9/30/91
Publish the joint NIST and ECMA ISEE RM document (version 1.0)	Marv Zelkowitz Bill Wong	10/91
Arrange and hold NIST ISEE Executive Committee Meeting - plan for the upcoming workshop	Bill Wong	TBD
Arrange and hold 6th Workshop	Bill Wong	10/91 or 11/91

b) **Summary of detailed schedule for completion of NIST ISEE Reference Model Document, version 1.0**

Marv Zelkowitz is designated as the editor of the NIST ISEE Reference Model Document with the assistance of Anthony Earl.

1) New sections from all the group leaders to Marv - July 15

   Object Management - Lolo Penedo
   Process and Task Management - Hal Hart
   Interface and Platform - Tricia Oberndorf
   User Interface - Bob Bagwill

2) RM document from Marv to the following individuals for review: - August 1

   - NISTEC
   - Anthony Earl

3) Comments due back to Marv - August 23

4) Revised document for review from Marv to: - August 30

   - NIST ISEE
   - ECMA TC33
   - other relevant groups

5) Comments due back to Marv - September 15

6) Finalize the document - September 30

7) Publish the ISEE RM document Version 1.0 - October ?

c) **Summary of status report from each working group.**

1) Object Management Working Group

Chair: Lolo Penedo, TRW
(213) 812-0595
penedo@trwarcadia.sdd.trw.com

Summary:

Expertise/familiarity w/ RM: little to very much

Group Activities:

- o reviewed OM services & dimensions definitions
- o discussed issues/question pertaining to OM services and global RM
- o reviewed service descriptions (each service was allocated to at least one person).
- o provided answers/solutions to issues from mapping exercise and reviews

Accomplishments:

- o Reviewed all text associated with the OM services
- o Provided recommended solutions for issues
- o Identified issues for future discussion

Action items:

- o T. Strelich - will review 3 services by June 14;
- o L. Penedo - is section boss for document, i.e., will review, modify and integrate all comments about OM services into document, which is due to Marv of NIST as the editor of the RM document in early July.
- o WG Attendees - will review final text

DISCUSSION ITEMS (OM)

1. Issue: OS issues as related to "Process Support" service.
   Resolution: This service is not supposed to deal with processes in general; it is there only in the case that OS processes are treated like objects in OMS. Text will be changed in the document.

2. Issue: Deleting "data" dimension.
   Resolution: no apparent problems for OM services.

3. Issue: Merging "types" & "metadata" dimension.
   Resolution: No problems, titled new section "Types/Metadata".

4. Issue: What to do when some dimensions are not described in RM document? Should "Not Applicable" dimensions be noted?

Resolution:

a) unclear - try to put text whenever possible and applicable
b) N/A is ambiguous - don't use it for the time being

5. Issue: Unclear distinction between "State Monitoring" and "Constraint" Services.
Resolution: Merge those services.

6. Issue: Data Model service seems out of place.
Resolution: Include in conceptual dimension of "Metadata" Service.

7. Issue: Redundancy of "Tool Registration" service across RM.
Resolution: Deleted Tool Registration service from OM grouping (assuming it is kept in Framework Adm.).

8. Issue: Is Object navigation covered anywhere?
Resolution: In Query and/or Archive and/or Relationship and/or Data Storage services.

9. Issue: What is the difference between Data and Task transaction?
Resolution: Added clarification text in data transaction.

10. Issue: No services for Data/DBMS like administration services.
Resolution: Future discussion.

11. Issue: Does the configuration service refer to CM?
Resolution: No, it deals with Composite Objects which may support CM.

12. Issue: Shall non-persistent data be addressed?
Resolution: Future discussion.

13. Issue: Consistency in terminology OM, DM.
Resolution: Review document carefully. OM does not mean O-O, nor ER; it means data management in general.

14. Issue: Heterogeneous object interchange.
Resolution: Future discussion.

Change items for change in document:

- o clarify text in OS Process support.
- o delete Data sections and merge Types with Metadata.
- o merge Constraint and State Monitoring Services.
- o delete Tool Registration.

- o clarify data versus task transaction.
- o clarify meaning of "object".
- o merge Data Model and Metadata services.

Future Issues:

- o Delete OS process support service (in OM grouping) or generalize service to deal with OS related services (beyond process).
- o Add new services to deal with Database administration services.
- o Deal with non-persistent data.
- o Deal with heterogeneous object interchange.

2) Process and Task Management Working Group

Chair: Hal Hart, TRW
(213) 812-0661
halhart@ajpo.sei.cmu.edu

Summary:

Expertise/familiarity w/ RM: This group of participants combined to yield extensive familiarity and expertise with both the RM and the subject area (life-cycle process management).

Group Activities:

- o Developed and discussed problems/issues with current RM organization and definitions relating to Process Management (PM).
- o Gained consensus on moderate reorganization of RM section (including some added and some grouped services) and approach for dealing with appearance of overlap with OM.
- o Broke into 1- or 2-man subgroups to author new descriptions of each re-organized service section (will become mainly the "Conceptual" sections); distributed each to PM WG.

Accomplishments:

- o Renamed section "Process Management Services" to be more consistent with what we perceive to be common community terminology.

- o Committed to converging RM PM terminology with SEI Process Glossary distributed recently by Watts Humphrey and Peter Feiler; we will provide feedback to SEI where we have detected inadequacies and with their glossary.

- o Developed revised list of PM services and corresponding new list of RM PM subsections.

- o Began revision or re-writing of new RM PM subsections.

Action Items:

- o Hal Hart - is section boss for document, i.e., will review, modify and integrate all drafts and revisions about PM services into document; Anthony Earl intends to assist with final writing:

  - draft due to WG approx. June 25
  - completed draft due to NIST in early July.

- o Jim King - will continue to work terminology/glossary normalization actions, including both review of new RM PM usage against SEI Process Glossary and feedback to SEI about our detected inadequacies and incompletenesses.

- o All Participants - will review revisions and drafts distributed at workshop and provide feedback to section boss and authors (hopefully via e-mail so all of WG can see it); and will review June 25 draft consolidated PM section.

Discussion Items:

General Overlaps with OM:

General Issue Resolution: Keep OM and PM sections independent, include redundancy (e.g., storage services) where it is known that some SEE's will provide distinct sets of services for OM and PM. But, clarify in introductory text the potential for some SEE's to deal with both uniformly or identical services, and be sure that usage of the RM (e.g., mappings) provides distinction between SEE's with totally separate OM & PM services versus partially/totally merged services.

1. Issue: Differences between Data and Event Triggers.
   Resolution: Data triggers always deal with changes to persistent data, but events may occur as result of changes to the execution state of processes which may or may not involve persistent data in different SEE's. Hence, Event Triggers must be uniquely provided in the PM section, but the potential for cases of overlap or equivalence with Data Triggers should be noted.

2. Issue: Process Transactions versus Data Transactions.
   Resolution: Meaningful distinction for SEE's where events are independent of changes to persistent data, but general case will be that Process Transactions also involve some notion of data transactions. General "Scope" issues of visibility, propagation, applicable constraints, and relation of scope to history management apply. Will be dealt with by retitled/revised section 6.3 titled "Visibility/Scoping Services."

Others:

3. Issue: New Services, esp. Metrication.
   Resolution: Previous section 6.6 ("Audit and Accounting Services") will be generalized into "Process Control Services," including scheduling, general metrics specification & collection (of which auditing, accounting, & emerging process metrics are specializations), history recording selection, and simulation/analysis.

4. Issue: Framework Definition/Modification Facility.
   Resolution: Not dealt with; probably deferred to Framework Administration section.

5. Issue: Function/Attachment Integration.
   Resolution: Opinion was that this is dealt with adequately in the OM section (as an accommodation to object-oriented approaches?) and does not impact PM section.

6. Issue: Identify Classes of Process Information.
   Resolution: Identified classes that provide context for all Process service descriptions:
   - (0) Description of process fragments (process assets)
   - (a) Description of software process (interpretable by humans)
   - (b) Description of software process (in some enactable form)
   - (c) Description of the execution state of a software process:
     - products manipulated by a process have a product state
     - some parts of the product state may also be part of the execution state
     - some parts of product and execution state will be linked
   - (d) Description of history of software process execution.

7. Issue: What to Do about SEI Terminology List.
   Resolution: We will adapt to SEI Process Glossary where it applies, and provide feedback to SEI where we detect incompletenesses and inconsistencies.

8. Issue: Process Operations.
   Resolution: Discussion identified a list of Process Meta-Model Operations (which is reflected in revised set of PM services below, and eventually their "Operations" dimensions, with refinement):
   - process scheduling
   - metrication & its specializations
   - modifications to process descriptions
   - modifications to descriptions of executable processes
   - modifications of descriptions of executing processes (& hence to their execution states)
   - history record selection
   - simulation/analysis

9. Issue: Revision of Section Structure and Contents.
   Resolution: See "Change Items" below.

10. Issue: Granularity -- Does RM imply that only Coarse-Grained Storage/Manipulation Support is needed?
    Resolution: Check that text does not assume coarse granularity; check that Process Definition (and OM storage services for SEE's where OMS provides PM support) recognizes distinction.

11. Issue: Inconsistency between Meanings of "Task" versus "Process".
    Resolution: Term "Process" will be used consistently henceforth in the NIST ISEE Reference Model, in the sense of the implied adjectives "life-cycle" or "software development" always being present.

CHANGE ITEMS FOR RM PM SECTION RESTRUCTURING:

New PM Section Structure (with mapping to previous & draft authors):
(NOTE: Although new subsections 6.3 - 6.6 are moderate regroupings or extensions of scope, the WG expects that almost all existing text will be reused; however, significant new and revised text is also expected).

6 Introductory Text (Ian Thomas)
6.1 Process Definition Services (6.1; Hal Hart & Anthony Earl)
6.2 Process Enactment Services (6.2; Hal Hart & Anthony Earl)
6.3 Visibility/Scoping Services (6.3; Joe Morin & Ron Peterson)
6.4 Process State and History Services (6.4, 6.5; Bob Balzer)

6.5 Process Control Services (6.6; Ken Shere)
6.6 Resource Management Services (6.7, 6.8; Bill Ett, Jim King)

Future Issues:

- o Synchronize with OM section for clear handling of SEE instances where service classes are independent versus those SEE's where OM services serve some dual or support role for PM (esp. definition storage; transactions; triggers)
- o Negotiation with Framework Administration section for handling of tool registration
- o Negotiation with Framework Administration section for handling of metrication services

3) Interface and Platform Working Group

Chair: Tricia Oberndorf
(215) 441-2737
tricia@nadc.navy.mil

Summary:

The Interface and Platform Working Group of the NIST ISEE Workshop met on the afternoon of Tuesday, June 4, 1991. On Wednesday morning, time was to be used to generate the assigned text. The first priority of the group is to address all of the sections of the NIST ISEE RM that are not covered by the other working groups, so we covered the following sections during our meeting:

7 - Communication Services
9 - Tools
10 - Security Services
11 - Framework Administration and Configuration

It has previously been agreed that the new Integration section will not be completed for the September document, so no discussion was held regarding this section. (See the summary of the Integration session held Wednesday afternoon.)

The attendance was very small, leaving us only one or two authors per section. Comments from the mappers as well as outstanding issues were addressed. The results are provided in the following paragraphs.

SECTION 7 - Communication Services

We agreed with the initiative already taken in the recent work on the RM which suggested that the "Message Services" section should be broadened to include other sorts of communication services, including RPC, "file" sharing, pipe-like mechanisms, and session dialogues and protocols, as well as the message delivery services already discussed in the section. We also discussed the role of such session issues as connection-oriented versus connectionless protocols and synchronous versus asynchronous mechanisms. One author took an assignment to work on broadening the section in accordance with our discussions.

One question we considered in this section was the one raised by the CAIS-A mapping regarding where to place CAIS-A's I/O protocols, model, and services. It was agreed that this would logically fit into the Communication section of the document, but that it should be postponed until some revision after the September document.

SECTION 9 - Tools

In a move to consolidate the coverage of Integration in the RM, it was agreed that the Tool Integration section would be moved to Section 12 (Integration). We do believe that there is a place in the RM for a section on Tools, although we discussed several different placements, drawing no final conclusion. Because of the ambiguities inherent in the term "tools", we discussed alternate titles (e.g., "Domain-specific _____" or "Life-Cycle Support _____"), but we came to no final conclusion here either. We also agreed that we should keep a software-oriented tools list such as is currently in the RM, noting that it is an example and that there are similar tool lists for CAD/CAM, Office Automation, and other disciplines of interest. One author took an assignment to work on rewriting the general text regarding tools; a second author took an assignment to consolidate the "Tool Registration" text currently spread throughout the document (see discussion of Section 11).

SECTION 10 - Security

It was agreed that it was important to keep a section on Security and that there are likely to be a number of other similar sections which should be added. Making this one part of a larger section on "Policy Enforcement Services" was discussed, but it was felt that it was premature to attempt that for the September document. We also discussed location of the section, but decided it should be left where it is (i.e., after Tools and before Framework). We agreed that the contents needed to be better articulated and more in line with the format used in all the other sections (i.e., providing a discussion according to the applicable dimensions). One author agreed to work on this assignment, based on a related paper which he has authored for a related effort.

SECTION 11 - Framework Administration and Configuration

The first issue discussed here was with regard to the disposition of a consolidated section on Tool Registration. Although the group originally agreed that this should be placed in the Tools section, the sentiment was not extraordinarily strong and this decision was subsequently reversed by the executive committee, leaving Tool Registration in this section. Several other issues raised by the mappers were considered for placement in this section, but upon discussion all of them seemed to fall into some other category, most notably one that would be called "Environment Administration" as opposed to Framework Administration. As there is no home for such a section within the current scope of the document, it was agreed that these additions would have to wait. Thus one author, with the assistance of another attendee, agreed to work on the Metrication section as well as the general introduction.

Several other questions raised by the mappers, along with a few noted in the Framework Administration section of the current draft, were considered. The following is a list of those questions and the disposition of them proposed by the IPWG.

- Installation Procedures, including Tool Management - include in Environment Administration or perhaps Integration.
- Framework Definition/Modification - include in the Subenvironment (Views) Service.
- Environment Definition/Modification - include in Environment Administration.
- Environment Deletion - include in Environment Administration.
- OMS Schema Management - include in Section 5 (OM Services).
- Groupware Management - include in Section 5 (OM Services).
- Resource Registration and Mapping - include in Section 5 (OM Services).
- Services provided by underlying OS/Platform - discuss how to handle these in the Mapping Guidelines.
- Exception Handling/Error Handling Services - include in 5.14, 6.5, and Integration.
- Project Management Integration - include in Section 6 (PM Services).
- Tool Invocation (including command interpreter?) - include with Tool Integration.
- Editors (textual, graphical) - include in Common services/Encapsulation.

- o Display/List services - include in Common Services/Encapsulation.
- o Help facilities - include in Common Services/Encapsulation.
- o Time services - not clear where it goes, but it is not the responsibility of the IPWG within the current document.
- o Repository Creation/Modification - include in Environment Administration.
- o Tool encapsulation/de-encapsulation - include with Tool Integration.

Mappers' comments which we did not address, on the assumption that they would naturally be picked up by other working groups, are:

- o interchange formats
- o presentation styles

Two of the five assignments were completed prior to close of the meeting. The remaining three are agreed to be delivered within two weeks following the close of the NIST ISEE meeting.

4) User Interface Working Group

Chair: Bob Bagwill, NIST
(301) 975-3282
rbagwill@nist.gov

Summary:

Expertise/familiarity with UI RM:

Except for Marv Zelkowitz, all the participants were new to the UI group. Bob Bagwill took over as leader when Brian Clapper could no longer participate in the NIST ISEE activity. Bob Bagwill presented two alternative UI models, the Seeheim and Arch models.

Group Activities:

- o reviewed the contents of the previous draft, which was identical to the ECMA TR/55 Technical Report.

- o re-confirmed the decision that the ECMA version was insufficient as a basis for a user interface services reference model.
- o decided X11 would be retained only as an example of an architecture to be mapped against the UI reference model.
- o discussed appropriateness of Seeheim and Arch models.
- o mapped Object Management Services against the UI domain.
- o added two service areas Internationalization and User Assistance Services
- o changed name of section to "User Interface Management Services."

Action items:

- o Bob Bagwill - will prepare rough draft of new UI section and email to attendees
- o Marv Zelkowitz - will review draft
- o WG attendees - will review the text

d) Summary of the NIST ECMA ISEE Reference Model Mapping Exercise Session.

Chair: Sandi Mulholland, Rockwell
Marv Zelkowitz, University of Maryland and NIST

Summary:

The NIST ISEE Reference Model Mapping Exercise Summary session of the 5th NIST ISEE Workshop was held Tuesday morning. All attendees of the 3 workshops were invited. During the Fourth NIST ISEE Workshop, it was recognized that in order to test the effectiveness of the Reference Model, it would be necessary to map the set of services provided by various activities into the set of services provided by the NIST/ECMA ISEE reference model. Accordingly, in March, 1991 a meeting was held to discuss this mapping exercise. The following individuals and activities were represented:

1. Geoff Clow of SofTech representing CAIS-A

2. Anthony Earl of HP representing ECMA PCTE

3. Barbara Meyers of IBM representing AD/cycle

4. James Milligan of Rome Laboratory representing SLCSE

5. Hal Pierson of SPC representing the CIS ATIS activity

A mapping form was generated and reports from each of the 5 activities was due in mid-May. This session of the Fifth Workshop was a report on the results of this mapping exercise.

The reference model is based upon 13 characteristics (dimensions) for each of the approximately 40 services of the model. In order to make the overall effort for each mapper reasonable, only three dimensions were requested for each service:

1. Conceptual - What the service does

2. Operations - How it does it

3. Related Services - Which other services are related to this dimension

During the session, four of the mappings were discussed. The CIS mapping was delayed and will be conveyed to NIST at a later date. (Note: Copies of the SLCSE, ECMA PCTE and CAIS-A mappings were submitted to the PSESWG archive and can be retrieved by anyone with internet access. Send mail to PSESARCH@NADC.NAVY.MIL with the subject line 'help' to get further information. The AD/cycle mapping has to be obtained directly from IBM.)

General comments on the mapping exercise were:

1. There was agreement that such a mapping exercise is a useful activity. It helped each of the mappers to understand their own architecture better and helped to determine places where either their own designs or their own documentation needed to be altered.

2. The reference model and the mapping exercise provided a common basis for discussing alternative environment frameworks.

3. All services in the reference model seemed relevant to the mappers. In addition, a few new services were suggested for inclusion into a later draft of the model.

Some issues raised by the mappers included the following:

1. There was general confusion concerning the placement of platform - supplied services outside of the services explicitly designed for the environment framework. For example, many native systems included underlying backup services (e.g., DEC's VMS for SLCSE, IBM's MVS for AD/cycle), so it was not clear if such services are explicitly part of the environment framework. It was clear that the reference model document needs to explain this better. The reference model is a description of the services provided to tools executing on the framework. Whether these services are provided specifically by the framework implementation or passed through by the underlying native operating system, is an architectural issue and outside of the realm of whether the service is supplied or not.

2. Tool registration concepts need to be collected in one place in the document. Also, the scope of the document -- environment frameworks versus full life cycle tool support that needs to be clarified.

3. The data dimension was deemed redundant (e.g., it only appears once in the version of the reference model document used at the meeting). The internal, operations and external dimensions provided sufficient detail to replace this.

4. The metadata dimension was also redundant. There is a special metadata service and most of the other metadata issues seem like cases of the types dimension.

5. Several services seemed redundant. The various working groups were to consider merging the following pairs:

   a. State monitoring and Event monitoring
   b. Data interchange and Archive service

6. Several chapters were obviously incomplete and needed additional service descriptions: Security, Framework administration, Communication services and User interfaces.

7. All four reported mappings were based upon an ER model of the data repository. It would be useful to obtain an object-oriented mapping (e.g., ATIS) to test against the reference model service descriptions.

8. New services that were proposed included:

   a. Timing features -- elapsed time, time of day

   b. Standardized error reporting

c. Framework definitions -- changing roles, names, tools, personnel, etc.

d. Tool integration issues -- encapsulation "wrappers," "plug and play" tool slots

e. Environment servicing -- tool installation, modification, tailoring

e) **Summary of the Integration Services Session.**

Chair: Tricia Oberndorf

Summary:

The Integration session of the 5th NIST ISEE Workshop was held Wednesday afternoon. All attendees of the 3 workshops were invited.

BACKGROUND

At the 4th NIST ISEE Workshop, there was a brain-storming session during which everyone was encouraged to speak up with anything they felt constituted an environment "integration mechanism". No time was spent on defining just what that meant; instead we relied on a common intuition to come up with something that would be representative of the wide range of services that might be needed. Following that workshop, the list was organized slightly, and the result is attached to this report.

OBJECTIVE

For the purposes of the 5th NIST ISEE Workshop, a more structured approach was taken to the Integration session, since the real question now is how Integration should be represented in the NIST ISEE RM. This session began by discussing various possible definitions for "integration" and examining several ways in which various authors have attempted to characterize it. Following those discussions, we took up the question of how to address it in the RM.

DEFINITIONS AND CHARACTERIZATIONS

The session started with the examination of a number of definitions and characterizations of "integration" from various authors. Through discussion of these aspects, a common

understanding of integration developed among the participants. Currently this common understanding can best be expressed in a series of statements as follows.

- Integration is about putting parts together in such a way that they work together towards some goal.
- This goal involves the parts working together smoothly, correctly, easily, cost-effectively, etc.
- Integration is about agreement.
- Integration is not monolithic; it is complex and is only understood by understanding a number of different areas from a number of different perspectives.
- To completely describe integration, one must consider answering at least the questions, "what", "why" (which relates to the "goal" mentioned above), and "how" (which relates to the mechanisms that resulted from the 4th ISEE brainstorming session).
- Integration is not something that a single entity either has or does not have; instead, it is a property of the relationship between two or more things. (Thomas and Nejmeh)

In addition to the typical characterizations of integration (data, process, control, presentation), we discussed "semantic" integration and the goal of reducing tools to their "essence", with all other features provided by the framework into which they are integrated.

After examination of different attempts to characterize integration and integration features, it was possible to answer a few questions. It seemed to be the group's feeling that "Integration" is not a layer, as it is sometimes depicted in reference model or architectural diagrams. Nor is it a service per se', although there are a few services which it was felt are unique to integration (e.g., a common data repository, triggers, encapsulators, filters).

REPRESENTATION IN THE REFERENCE MODEL DOCUMENT

One of the difficulties in getting a handle on Integration in the context of the reference model appears to be the fact that the scope of the current reference model is environment frameworks and frameworks are largely motivated by desires for achieving integration. In other words, since the rationale for having frameworks is that they provide factorization with enforcement, which naturally leads one to desire maximum commonality within that factorization, it is hard to tell a "framework service" from something that one might call an "integration service" or "integration mechanism". However, following the discussion of definitions, it was possible to address this separation to some extent.

Four questions were put before the group:

1. Is Integration a Reference Model dimension?

2. Is Integration a policy?

3. Should there be a separate section for Integration Services?

4. Should Integration be covered in a separate model/document?

Although there was some support for answering "yes" to the first question, it seemed to be the sense of the group that, if done at all, this would not suffice. No one really seemed to pick up on the second question, which would suggest that that did not make much sense. The conclusion of the group was a combination of short-term and long-term approaches which can be characterized as:

3a. For the September document, discuss only what is known today

- the issues, laying out the "space" to be covered by an eventual Integrati on section or model

- the relationship of integration to the existing services

4a. Do a proper development of an Integration reference model for future presentation, either as a separate document or as an eventual section of an evolving NIST ISEE RM document

- keep in mind that an RM is descriptive

- keep in mind that Integration is one of several ways of looking at and describing frameworks.

Some pragmatics were also brought up. It would be unwise to create a part of the current RM document which would undoubtedly be changing faster than the rest of the document. It will also be important to find out who else is working on what parts of such an Integration RM.

SECTION II

IWCASE WORKSHOP

INTERNATIONAL WORKSHOP ON CASE (IWCASE)

CASE STANDARDS COORDINATION UPDATE MEETING

Thursday, June 6, 1991
8:30 am - Noon
Sheraton Hotel
Redondo Beach, CA

In Conjunction with NIST ISEE and PCIS Workshops

STANDARD UPDATE REPORTS

David Sharon
North American Chairman
CASE Standards Coordination Committee
c/o CASE Associates Inc.
15686 S. Bradley Road
Oregon City, OR 97045
(503) 656-0986

IWCASE
CASE STANDARDS COORDINATION UPDATE MEETING

Standard Update Reports

- CALS Industry Steering Group - Software Products Committee
- CDIF - EIA CASE Data Interchange Format Technical Committee
- ECMA TC23 PCTE
- IEEE-CS P1175 - A Standard Reference Model for Computing System Tool Interconnections
- IGES/PDES - Software Products Committee
- NIST/CSL - NIST IEEE Working Group
- NGCR/PSESWIG - U.S. Navy Next Generation Computer Resources/Project Support Environment Standards Working Group
- U.S. TAG for ISO/IEC JTC1/SC7

Expect to Receive Reports From:

- CAIS-A
- PCIS
- ANSI IRDS
- ISO IRDS
- AD/Cycle
- Cohesion
- STARS
- OMG

and others

IWCASE Inc.

STANDARDS COORDINATION UPDATE MEETING

In Conjunction with NIST ISEE and PCIS Workshops

Thursday, June 6, 1991
8:30 am - Noon
Sheraton Hotel
Redondo Beach, CA

STANDARDS UPDATE REPORT

Organization Name: CALS Industry Steering Group

Contact Person: Thomas G. Baker (206) 234-6234

Name of Project/Working Group: Software Products Committee (SPC)

Purpose of Standard:
The Computer-Aided Acquisition and Logistic Support Initiative represents an effort by the DoD to improve quality and decrease costs in the acquisition and support of weapon systems through the automation of integrated processes. The SPC is addressing methods of bringing software products into the CALS environment.

Scope of Standard:
The CALS initiative will impact DoD standards related to all aspects of weapon system acquisition and support. The SPC is addressing the subject of CALS compatibility with the software functional standards DOD-STD 2167A and 7935A.

Objective of Standard:
The objective of the SPC is to study, document and recommend on the CALS compliance of software functional standards, life cycle processes and deliverables for the:

Near Term - Define requirements for digital delivery of information products.

Long Term - Define issues and potential solutions for accessing the data contained in information products.

Current Status:

The SPC has completed its analysis of Near-Term objectives and is finalizing the Near-Term Report. The group will continue work on Long-Term objectives.

Plans (with Milestones and Schedule Dates):

Submit Near-Term Report

Prepare updated Position Paper

Continue analysis of Long-Term issues.

Publications Available/Produced:

Software Products Committee (SPC) Position Paper, dated April 6, 1990.

Liaison/Coordination with Other Standard Groups:

The group has conducted joint meetings and presented to several software standards making organizations including the IEEE CS-P1175, EIA-CDIF, ISO TC184/SC4(STEP), and PDES.

Future Plans/Trends:

1. Act in advisory role in DOD implementation of Near-Term strategy.
2. Integrate results of Near-Term implementation into Long-Term strategy.
3. Provide requirements to PDES.

Problems/Issues/Needs:

The group would benefit from additional participation of people who use and maintain the software products. New members are welcomed.

Please provide the Reference Model used by your standard group and other graphical representations used to depict/position your standard.

KEEP YOUR REPORT TO 3 PAGES and FAX to Dave Sharon at (503) 656-3207 by May 31, 1991. Copies will be distributed to those attending the June 6th meeting.

Thank you!

David Sharon
North American Chairman
FAX: (503) 656-3207
Phone: (503) 656-0986

Organization Name:
CDIF - EIA CASE Data Interchange Format Technical Committee

Contact Person :
Stds. Coord. : Burt Parker (703) 883 5519
Chair : Mike Imber 011 44 71 636 4213

Name of Project/Working Group :
CDIF

Purpose of Standard :
To provide a neutral format for the interchange of data between CASE Tools, in the form of a family of 3 standards, covering Framework, Transfer Format and Standardized CASE Meta-model.

Scope of Standard :
To cover all information required to be interchanged between CASE Tools, both Semantic and Presentation, through Standardized CASE Meta-model and extensibility mechanism.

Objective of Standard :
to aid the development of open systems, by enabling users to move information between CASE Tools in different environments and platforms.

Current Status :
Aim to produce Interim Standard as basis for prototyping effort in mid 1991. In parallel will prepare draft for Proposed Standard based on expanded scope and feedback from Prototypes and coordination with other Standards Groups.

Plans (with Milestones and Schedule Dates) :
To produce Interim Standard in mid 1991, then proceed with drafts for Proposed Standard based on feedback and consultation with other Standards Groups. Timescale dependant on consultation progress.

Publications Available/Produced :
Internal Drafts of all three standards

Liaison/Coordination with other Standards Groups :
Regular liaison with PDES/SPC, PDES Data Dictionary, IEEE P1175 and CALS/SPC. Proposal for joint working group with ECMA TC33 (PCTE). Contact with ISO and ANSI IRDS.

Future Plans/Trends:
see Plans

Problems/Issues/Needs
Need ongoing coordination with other standards groups, as already commenced, to ensure greatest applicability of the CDIF work to other standards efforts.

IWCASE Inc.

STANDARDS COORDINATION UPDATE MEETING

In Conjunction with NIST ISEE and PCIS Workshops

Thursday, June 6, 1991
8:30 am - Noon
Sheraton Hotel
Redondo Beach, CA

STANDARDS UPDATE REPORT

Organization Name: ECMA

Contact Person: M W MORRON

Name of Project/Working Group: TC33 PCTE

Purpose of Standard:

To provide a public tool interface.

Scope of Standard:

Systems engineering tools repository interface.

Objective of Standard:

To enable tool integration and portability.

2/4

Current Status:

Reference model ECMA TR/55 approved Dec 1990
Abstract specification ECMA-149 standard approved Dec 1990
C Binding standard Final draft in review
Ada Binding standard First draft in review

Plans (with Milestones and Schedule Dates):

C Binding standard approved 13 June 1991
Ada Binding standard approved Autumn 1991
C++ Binding standard approved 1992

Publications Available/Produced:

All of the above. Hundreds of working papers available.

Liaison/Coordination with Other Standard Groups:

Liaison with contributors to SC21 WG7 ORP
Starting to cooperate with CDIF and hopefully P1175 on common meta meta model.
Closely working with PCIS and in contact with CFI.

Future Plans/Trends:

To enter ISO standardization via fast track this Autumn.
Plan to have ISO standard approved 1992.

3/4

Problems/Issues/Needs:

Need to formalise cooperation on common models.

Please provide the Reference Model used by your standard group and other graphical representations used to depict/position your standard.

KEEP YOUR REPORT TO 3 PAGES and FAX to Dave Sharon at (503) 656-3207 by May 31, 1991. Copies will be distributed to those attending the June 6th meeting.

Thank you!

David Sharon
North American Chairman
FAX: (503) 656-3207
Phone: (503) 656-0986

4/4

** TOTAL PAGE.004 **

IWCASE Inc.

STANDARDS COORDINATION UPDATE MEETING

In Conjunction with NIST ISEE and PCIS Workshops

Thursday, June 6, 1991
8:30 am - Noon
Sheraton Hotel
Redondo Beach, CA

STANDARDS UPDATE REPORT

Organization Name: IEEE-CS Task Force on Professional Computing Tools

Contact Person: Bob Poston, Chairman

Name of Project/Working Group: P1175 - A Standard Reference Model For Computing System Tool Interconnections

Purpose of Standard:

To help builders, buyers, testers and users of tools to integrate Tools into family of tools. The standard provides a:

1. Reference model for tool to organization interconnections
2. Reference model for tool to platform interconnections
3. Standard text language (STL) for information transfer between tools.

Scope of Standard:

P1175 describes interconnections that must be considered when buying, building, testing, or using computing system tools. Computing system tools are special purpose tools that are developed for use in creating computing systems. These tools include:

1. Computer-Aided Engineering Tools (CAE)
2. Computer-Aided Software Engineering Tools (CASE)
3. Computer-Aided System Engineering (CASE)

Current Status:

P1175 Final draft review concluded at ICSE May 16-18, 1991.
P1175 has been submitted for balloting as a "Trial Use Standard"

Plans (with Milestones and Schedule Dates):

Final Review - May 16-18, 1991
Balloting - May 20 - August 23, 1991
Conflict Resolution - August 25-27, 1991.
IEEE Standards Board Approval - Sept. 1991

Publications Available/Produced:

P1175 / D11 May 22, 1991

Liaison/Coordination with Other Standard Groups:

CDIF, PDES and CALS

Future Plans/Trends:

Extend the STL to include other life cycle phases and tool types

IEEE-CS P1175

Problems/Issues/Needs:

Gaining acceptance

Resolving feedback during Trial use period.

Please provide the Reference Model used by your standard group and other graphical representations used to depict/position your standard.

KEEP YOUR REPORT TO 3 PAGES and FAX to Dave Sharon at (503) 656-3207 by May 31, 1991. Copies will be distributed to those attending the June 6th meeting.

Thank you!

David Sharon
North American Chairman
FAX: (503) 656-3207
Phone: (503) 656-0986

IWCASE Inc.

STANDARDS COORDINATION UPDATE MEETING

In Conjunction with NIST ISEE and PCIS Workshops

Thursday, June 6, 1991
8:30 am - Noon
Sheraton Hotel
Redondo Beach, CA

STANDARDS UPDATE REPORT

Organization Name: IGES/PDES Organization

Contact Person: Thomas G. Baker - (206) 234-6234

Name of Project/Working Group: Software Products (SP)

Purpose of Standard:

The development of a series of Standards for the Exchange of Data to provide a neutral mechanism capable of describing product data throughout the life cycle of a product, independent from any particular Computer-Aided system. The nature of this description will make it suitable not only for neutral file exchange but also as a basis for implementing and sharing product databases and archiving. SP was formed to bring software and its associated products into the STEP environment.

Scope of Standard:

The scope of the software considered includes all software embedded in, or associated with, any product describable by PDES/STEP. Software product data describes requirements, design, implementation (code), test and support documentation such as user's manuals, installation instructions, etc.

Objective of Standard:

The objective of Software Products is to develop the STEP models for software product data supporting the entire life cycle.

Current Status:

The group has evaluated EIA-CDIF and IEEE CS-P1175 software standards for potential reuse in the SP Application Reference Model (ARM). Development of the SP ARM is underway. Other software standards will be evaluated as appropriate.

Plans (with Milestones and Schedule Dates):

The group is awaiting final approval as a project under STEP. The groupwill develop a draft ARM and will start development on the Application Activity Model. The group will select computing tools for project use. A language for internal modeling efforts will be determined.

Publications Available/Produced:

Liaison/Coordination with Other Standard Groups:

The group has been very active in coordination efforts. Joint meetings and technical sessions have been conducted with EIA-CDIF, IEEE CS-P1175, and CALS ISG Software Products Committee. The Group has also coordinated with other PDES Committees.

Future Plans/Trends:

After the Application Reference Models are completed, the group will work with the Integration Committee to develop an Application Interpreted Model. Software Products Application Protocols will be included in STEP. The group will continue coordination with related activities.

Problems/Issues/Needs:

The group is interested in obtaining models from other organizations addressing the exchange of software information. New members are welcomed. As this is a fairly new group, participation provides a great opportunity for members to make a significant contribution to Software Products representation in STEP.

Please provide the Reference Model used by your standard group and other graphical representations used to depict/position your standard.

KEEP YOUR REPORT TO 3 PAGES and FAX to Dave Sharon at (503) 656-3207 by May 31, 1991. Copies will be distributed to those attending the June 6th meeting.

Thank you!

David Sharon
North American Chairman
FAX: (503) 656-3207
Phone: (503) 656-0986

IWCASE Inc.

STANDARDS COORDINATION UPDATE MEETING

In Conjunction with NIST ISEE and PCIS Workshops

Thursday, June 6, 1991
8:30 am - Noon
Sheraton Hotel
Redondo Beach, CA

STANDARDS UPDATE REPORT

Organization Name: NIST/CSL

Contact Person: William Wong (301)975-3341

Name of Project/Working Group: NIST ISEE Working Group

Purpose of Standard:

Provide guidance to Federal agencies in acquiring software development and maintenance environments (ISEEs).

Scope of Standard:

Identify and stimulate the plans and coordination needed among key software industry parties and relevant standards activities for consensus direction on open systems ISEEs.

Objective of Standard:

Identify and explore fundamental issues in ISEEs areas; identify the needed set of standards that define a comprehensive interface for integrating software tools; and develop guidelines on interface standards for ISEEs.

Current Status:

Enhance the NIST/ECMA Reference Model document for NIST publication, and review the results of reference model mapping exercise.

Plans (with Milestones and Schedule Dates):

Complete the revised NIST/ECMA Reference Model document and publish it as the first version of the NIST Reference Model document by the end of September 1991.

Publications Available/Produced:

The NIST/ECMA Reference Model Working Draft and the NIST ISEE Reference Model Mapping Guidelines draft.

Liaison/Coordination with Other Standard Groups:

ECMA/TC33, DARPA/STARS, AJPO/PCIS, NGCR/PSEWG, USAF/STSC, POSIX, IEEE, IWCASE, AIAA, ECMA/PCTE, CIS, CAIS-A, OMG, PDES, IRDS, P1175, P1201.

Future Plans/Trends:

Develop the NIST Reference Model for a full environment.

Problems/Issues/Needs:

Need volunteers help from software and system communities, Federal agencies and academic, especially in the areas of user interface, integration and security services.

Please provide the Reference Model used by your standard group and other graphical representations used to depict/position your standard.

KEEP YOUR REPORT TO 3 PAGES and FAX to Dave Sharon at (503) 656-3207 by May 31, 1991. Copies will be distributed to those attending the June 6th meeting.

Thank you!

David Sharon
North American Chairman
FAX: (503) 656-3207
Phone: (503) 656-0986

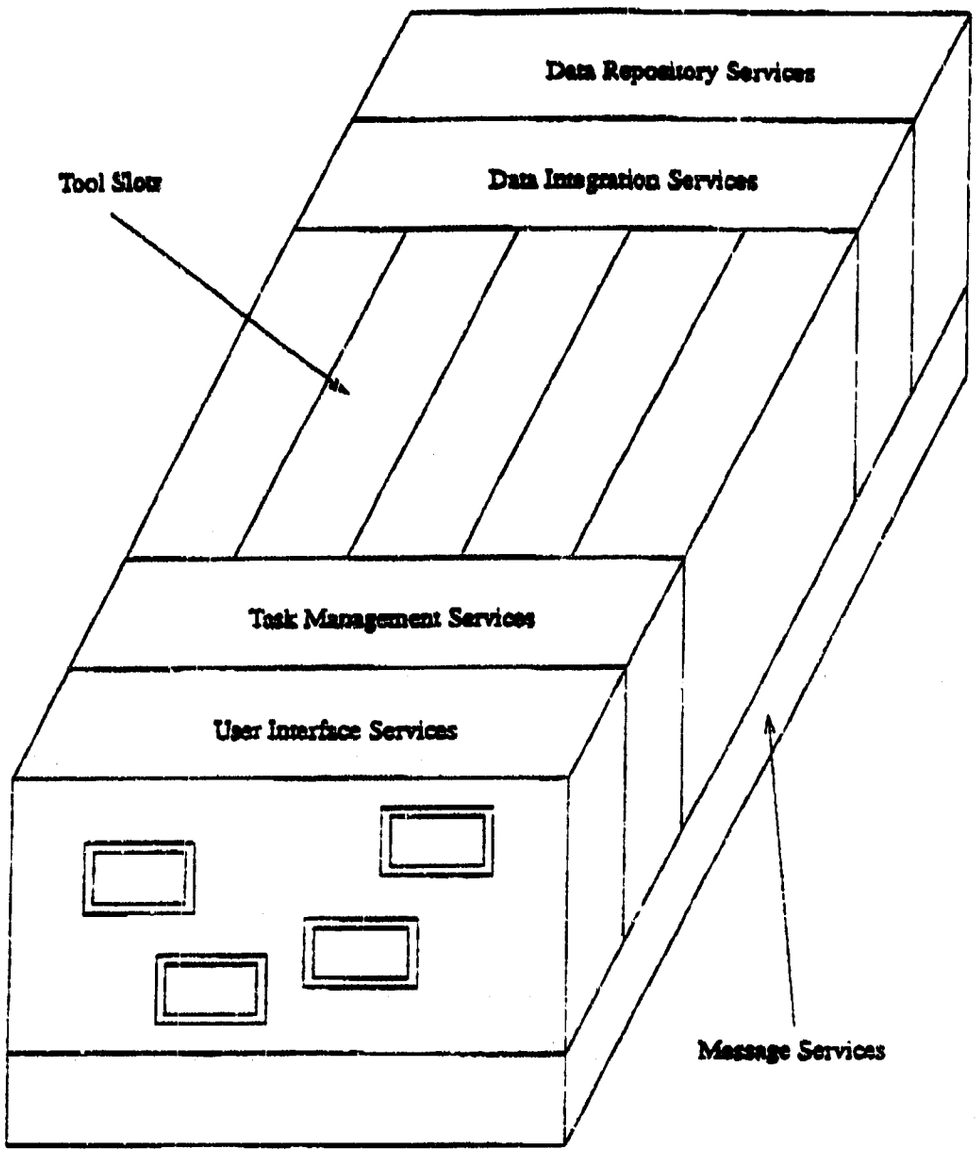


Figure 1. Overall Reference Model Structure

TOTAL P.005

IWCASE Inc.

STANDARDS COORDINATION UPDATE MEETING

In Conjunction with NIST ISEE and PCIS Workshops

Thursday, June 6, 1991
8:30 am - Noon
Sheraton Hotel
Redondo Beach, CA

STANDARDS UPDATE REPORT

Organization Name: U.S. Navy

Contact Person: Patricia Oberndorf

Name of Project/Working Group: Next Generation Computer Resources (NGCR) Project Support Environment Standards Working Group (PSESWG)

Purpose of Standard:

To support Navy acquisition of modern, cost-effective PSEs

Scope of Standard:

All interfaces, particularly framework interfaces, relevant to acquisition of PSEs for the support of Navy systems.

Objective of Standard:

To create a profile of PSE-related interface standards. To the greatest extent possible, the Navy standard will adopt existing or emerging industry standards. Vehicle is open meetings of a joint Navy/industry/other government/academia working group.

Current Status:

One meeting was held (Feb 26-28 '91) and the second will be 29-31 May. Subgroups are working on a comprehensive Reference Model and a compendium of Available Technologies.

Plans (with Milestones and Schedule Dates):

Meetings will be quarterly (third meeting is 4-6 Sep; generally will be [redacted], Feb, May, August, November).
Ref Model – Feb '92; first increment – Feb '93; final MIL-STD – Sep '98

Publications Available/Produced:

None.

Liaison/Coordination with Other Standard Groups:

We expect to mostly adopt standards from such groups as IEEE, ECMA, and ISO. We have a formal relationship with STARS and an informal one with NIST.

Future Plans/Trends:

We plan to identify all interfaces of potential interest and then to narrow down that list by assessing availability of candidates and payoff for standardization. We will then address each interface we choose to standardize by developing requirements, identifying candidates, and selecting one based on an evaluation process.

Problems/Issues/Needs:

---

* **Please provide the Reference Model used by your standard group and other graphical representations used to depict/position your standard.**

KEEP YOUR REPORT TO 3 PAGES and FAX to Dave Sharon at (503) 656-3207 by May 31, 1991. Copies will be distributed to those attending the June 6th meeting.

Thank you!

David Sharon
North American Chairman
FAX: (503) 656-3207
Phone: (503) 656-0986

* We do not yet have a Reference Model. We will closely follow the ECMA and NIST ISEE work in this area, as we would prefer to adopt one rather than introduce a new one. Our general "position" covers all interfaces relevant to the construction of PSEs, including networking, user interfaces, frameworks, and tool-to-tool interfaces

IWCASE, Inc.

STANDARDS COORDINATION UPDATE MEETING

In conjunction with NIST ISEE and PCIS Workshops

Thursday, June 6, 1991
8:30 am - Noon
Sheraton Hotel
Redondo Beach, CA

STANDARDS UPDATE REPORT

Organization Name:	U.S. Technical Advisory Group (TAG) for ISO/IEC Joint Technical Committee 1 (JTC 1) Subcommittee 7 (SC7)--Software Engineering
Contact Person:	Roger U. Fujii 213/831-0611 x2420 Logicon, Inc: 222 West Sixth Street San Pedro, CA 90733
Name of Project/Working Group:	U.S. TAG for ISO/IEC JTC 1/SC7

Purpose of Standard:

The U.S. TAG for ISO/IEC JTC 1/SC7 is the organization that provides the representation of various U.S. interests related to the scope and program of work of ISO/IEC JTC 1/SC7. See Current Status for standards projects active in ISO/IEC JTC 1/SC7.

Scope of Standard:

The scope and program of work of the U.S. TAG for ISO/IEC JTC 1/SC7 encompasses the projects included in ISO/IEC JTC 1/SC7. Mission statement for ISO/IEC JTC 1/SC7 includes, "*. . .standards for management techniques, supporting methods and tools, and procedures for design, development, testing, and maintainance of quality software.*" During the past three years, the U.S. delegates to ISO/IEC JTC 1/SC7 meetings have been influential in expanding the scope to include software engineering. The standards projects do not include projects for programming languages, computer graphics, office document interchange format and document content architecture, and database management systems included in the work of other ISO/IEC JTC 1 subcommittees. The scope of each standard includes applicability for international use and is, in most cases, based on a National standards.

Objective of Standard:

Scope and objectives of each standard are included in the proposal for developing the standard and is usually based on consensus developed during the early stages development. It is influenced by the reference document and by the delegates attending and participating in the technical discussions of the working groups.

Current Status:

JTC 1/SC7 Working Group 1:

- Charting Techniques for Software Development, including use in CASE tools, process flow, data modeling, state transition, data flow, and data structure [adding more techniques]
- Conventions for Usage of Symbols and Icons [new project]
- Survey of Diagramming and Charting for Inference Systems [new project]

JTC 1/SC7 Working Group 2:

- Software System Documentation [revision of current international standard]
- Support Graphics for Consumer Software [new project]
- Management of Information Transfer Between Life Cycle Phases [new project]

JTC 1/SC7 Working Group 3:

- Selection and Evaluation of CASE Tools [new project]
- Evaluation of Software Product Quality Characteristics, includes subcharacteristics, measurement and rating [near approval as standard]
- Life Cycle Management Processes [early stages of work]
- Software Configuration Management [new project]
- Quality Requirements and Testing Directives [new project]
- Software quality Engineering [new project]

JTC 1/SC7 Working Group 5:

- Reference Model for Information System Engineering [final stages of work]
- Mapping of Standards to Reference Model for Information System Engineering [development stage]
- Classification of Software [early stages of work]

Plans (with Milestones and Schedule Dates): Contact R. Fujii for details.

Publications Available/Produced:

U.S. TAG meeting summary, ISO/IEC JTC 1/SC7/WG working papers, committee correspondence, and ISO/IEC JTC 1/SC7 meeting summary are distributed to U.S. TAG members. Approved ISO/IEC JTC 1/SC7 standards are available through the American National Standards Institute. U.S. TAG does no have any formal publications.

Liaison/Coordination with Other Standards Groups:

U.S. ASC X3, DoD, NISO, EIA, IEEE/CS, NIST, NSIA, ASQC, NASA

Other: ISO TC156/SG2, TC159, JTC 1/SC1, JTC 1/SC18, JTC 1/SC1; IEC TC56, TC65.

Future Plans/Trends:

Plan includes advancing U.S. National Standards in the area of software engineering as International Standards. Principal sources of U.S. National Standards includes voluntary accredited standards-developing organizations such as the IEEE/CS Software Engineering Standards Subcommittee, government, and other National Standards. Trend is toward describing frameworks within which standards are developed or used to attain desired degrees of software system functionality and interoperability. Attached is an example of such a framework for standardization currently being considered for use in ISO/IEC JTC 1/SC7. See attached "SC7 Overview: Work Items."

Problems/Issues/Needs:

Broader representation of interests and more active participants in the U.S. TAG activities are needed to represent U.S. national positions during international meetings.

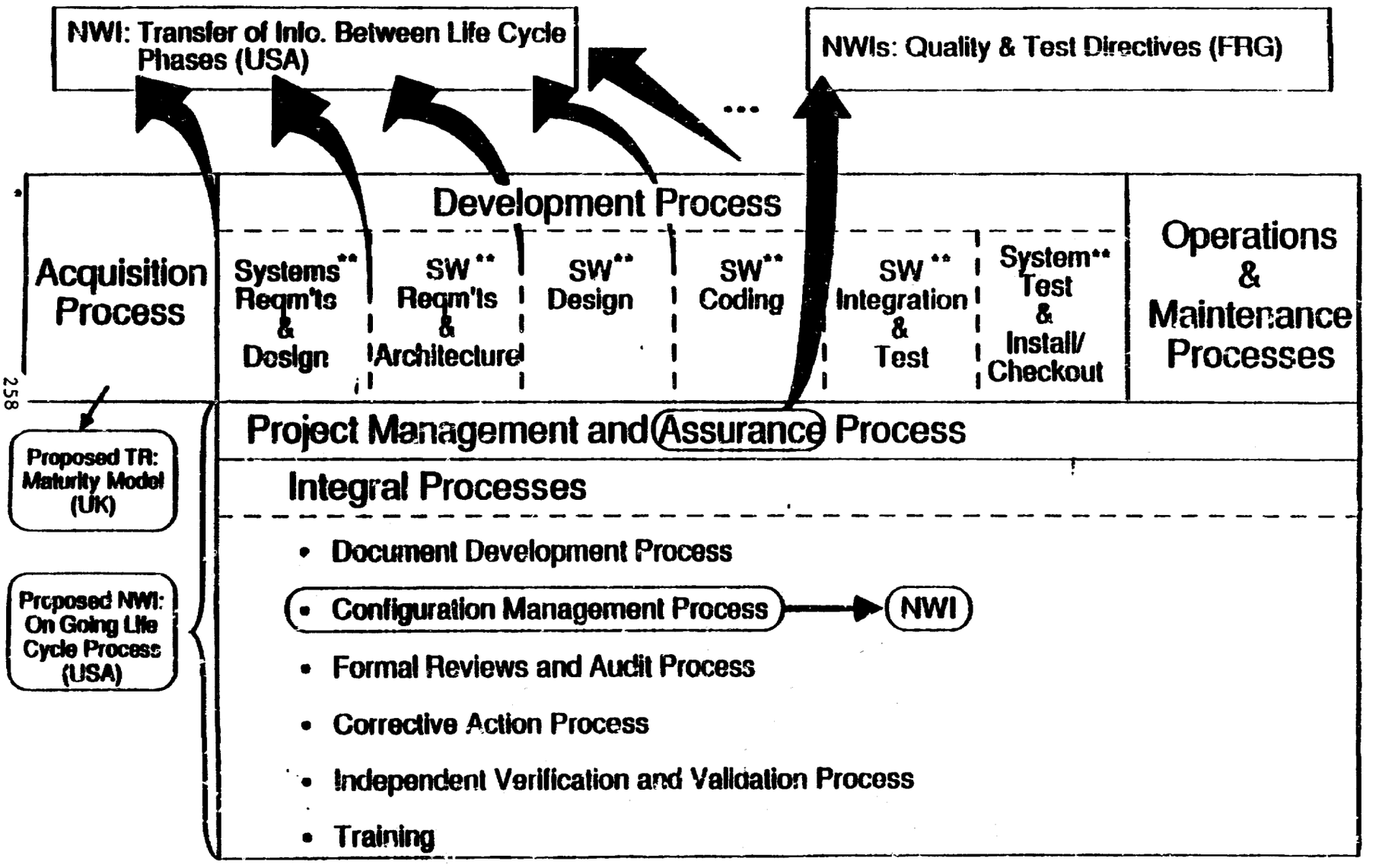
Extensive commitment of time and resources is needed to be an effective participant at any level of standardization in addition to the understanding of procedures and protocols for success.

Coordination workshops such as the IWCASE Coordination, NIST/ISEE, PCIS, and Joint X3 Database Systems Study Group (DBSSG) and ISO/IEC JTC 1/SC21 TAG sponsored meeting on the "Convergence of Open Systems Interconnection (OSI) and Data Management Standards," are necessary to foster communication among standards developers investigating interoperability issues.

A global framework of software engineering processes including software quality metrics and indicators, evaluation and selection of CASE tools, practical guidance about the management of the products of each life cycle phase, and services associated with an integrated software engineering environment is needed.

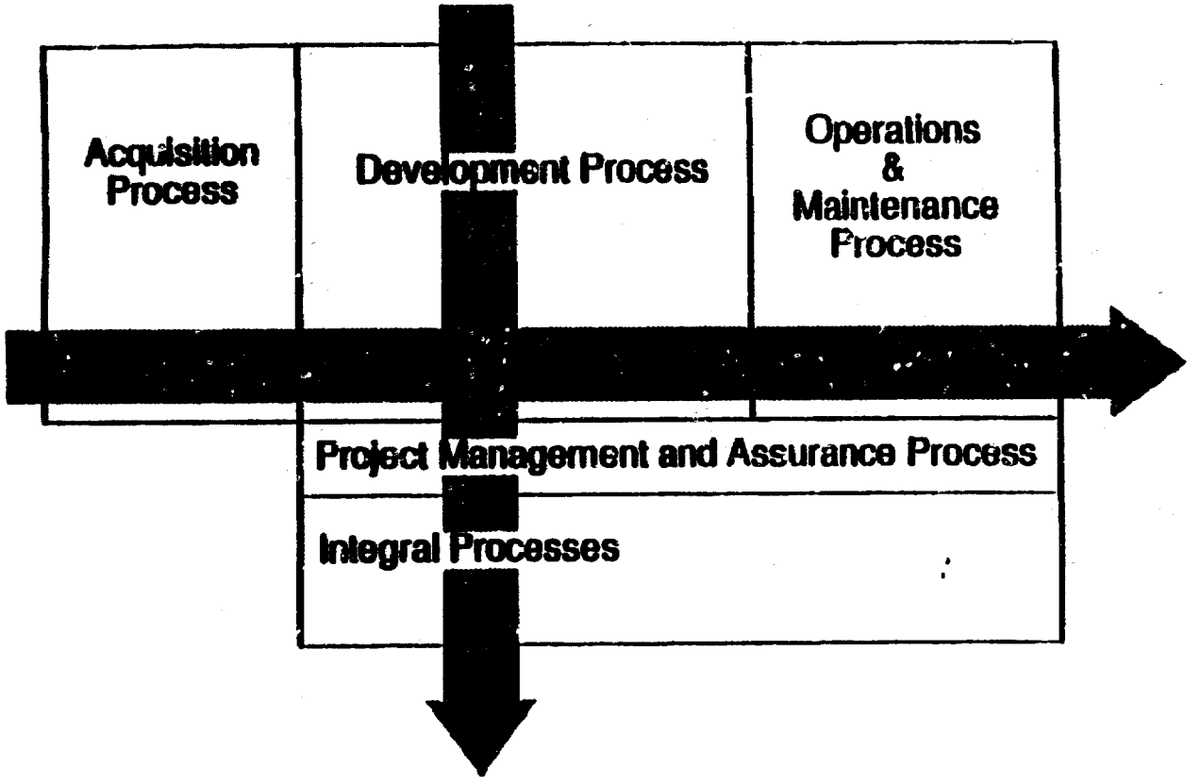
Submitted by: T.M. Kurihara
Member, US. TAG for
ISO/IEC JTC 1/SC7
703/695-4470
May 5, 1991

SC7 OVERVIEW: WORK ITEMS



*Entire Life Cycle Model = NWI (USA)
**Proposed NWI: Phases of the Software Development Process (USA)

SC7 OVERVIEW: WORK ITEMS (CONT.)



Tools/Techniques

1. 7.19.02 Charting Techniques for SW Development and Maintenance
2. NWI: Usage of Symbols and Icons
3. NWI: Survey of Diagrams/ Charting Techniques for Inference Systems
4. NWI: Evaluation and Selection of CASE Tools

Tools/Techniques

- Metrics
  1) DIS 9126 - Software Quality Characteristics (Japan)
- Indicators
  1) Consideration for a Proposed NWI: Quality Indicators (USA)

IWCASE Inc.

STANDARDS COORDINATION UPDATE MEETING

In Conjunction with NIST ISEE and PCIS Workshops

Thursday, June 6, 1991
8:30 am - Noon
Sheraton Hotel
Redondo Beach, CA

STANDARDS UPDATE REPORT

Organization Name: CASE Integration Services (CIS)

Contact Person: Dr. Andres Rudnik, SPS (Melbourne, FL)

Name of Project/Working Group: A Tool Integration Services (ATIS)

Purpose of Standard:

An Integrated Project Support Environment (IPSE) interface specification addressing those aspects of the IPSE critical to the tool integrator or implementor and which pertain to integration in particular.

Scope of Standard:

Focuses primarily on Data Integration, some Control Integration; intentionally does not address Presentation Integration. Requires services of underlying repository/database, but does not now specify interface to that service.

Objective of Standard:

Provide a sufficient base of IPSE services for effective and convenient integration of tools into a cohesive project support environment. Portability is viewed as a problem to be addressed separately.

Current Status:

A version of ATIS was submitted to ANSI IRDS as proposed std, "X3H4 Working Draft: IRDS ATIS". Not currently active in standardization process, but expected to begin soon. Work on ATIS is now conducted by CASE Integration Services (CIS) committee.

Plans (with Milestones and Schedule Dates):

- update Base Document (7/91)
- eliminate conflicts with PCTE, in coordination with PCTE groups (?/??)
- submit as proposed standard (?/??)

Publications Available/Produced:

Current version of ATIS is "CIS Base Document V1-0". Available from CI> committee secretary: Ms. Kathy Chapman, DEC, 110 Spit Brook Rd., Nashua NH, 03062, USA

Liaison/Coordination with Other Standard Groups:

CIS has proposed to ANSI SPARC the formation of a Technical Committee to evaluate & adopt standards pertaining to software development environments; ATIS will be submitted to this TC. CIS includes representation from ECMA, OMG.

Future Plans/Trends:

Harmonization with PCTE

Problems/Issues/Needs:

Note: "ATIS" is the interface specification.
"CIS" is the committee which is developing ATIS.

Please provide the Reference Model used by your standard group and other graphical representations used to depict/position your standard.

KEEP YOUR REPORT TO 3 PAGES and FAX to Dave Sharon at (503) 656-3207 by May 31, 1991. Copies will be distributed to those attending the June 6th meeting.

Thank you!

David Sharon
North American Chairman
FAX: (503) 656-3207
Phone: (503) 656-0986

CALS, PDES and Software Products

Presentation to the

NIST-ISEE / PCIS / IWCASE Workshop

June 3-7, 1991 in Redondo Beach, California

Tom Baker, Boeing

(206) 234-6234

Overview

- Software Products - what do we mean?
- CALS
  - What is it? - Concept, DoD Initiative
  - CALS ISG organization - where's Software Products?
- CALS implementation standards
  - CALS, MIL-STDs, industry
  - IPO (PDES) - United States effort supporting STEP
  - ISO (STEP) - international product data standard
- Software Products
  - Standards coordination
  - CALS Software Products Committee
  - PDES/STEP Software Products

Software Products

What do we mean by "Software Products"?

- Software, just like an airplane or a building, can be a product. It can also be a tool. It depends upon your point of view in any given situation.
- The subject here is software as a product; how it is described, how it is developed, how it relates to other products, etc.
- The subject is not the use of software to satisfy some need.
- This definition covers not only the code, but related information products such as requirements and design specifications, user manuals, etc.

Software Products

Background

- Boeing doesn't build Boeing airplanes; it assembles parts that are mostly built by others. Boeing is really an integrator.
- Increasingly, in both military and commercial systems, the "glue" that holds things together is software - software that is an integral part of the product being built.
- Software is a growing component of military and commercial systems. It is becoming larger, more important, more costly, and more critical to system success.
- The software component of a delivered system was being ignored in the product standards development arena.

CALS - The Concept

- An improved way of doing business
- A way of doing business which involves users, customers, contractors, subcontractors, and suppliers
- A way of integrating all business processes through improved information
  - Connectivity, management and communication
  - Across the entire life cycle
  - Define information once and use it often
- Facilitates Total Quality Management, concurrent engineering processes, increased productivity, higher states of system readiness, and reduced life cycle cost

CALS - The Concept

- Documents are a view of information
- We must start thinking in terms of delivering information, not documents, and let the user create the views that they require.
- Complete information is available and accessible
- Near Term - deliverable in CALS digital form:
  - Capability to view selected information
  - Capability to cross reference information
  - Easy to change or add supporting information
  - Identification of changes
- Long Term - integrated product/process deliverable:
  - Information referencing/retrieval capability
  - Views of information can be created by the user as needed
  - Requirements, design, test, etc. information can be kept coordinated and up-to-date

CALS - The DoD Initiative

What is it?

- CALS is a joint DoD - Industry initiative to define/develop an ongoing strategy to:
  - Lower life cycle cost of major DoD acquisitions
  - Increase Total Quality Commitment
  - Decrease the flow of paper
  - Establish standards for data exchange
  - Promote concurrent engineering
  - Change the way DoD does business
- DoDI 5000.2 of 23Feb91 demands data "delivered in digital form unless clear and convincing analysis shows this not to be cost effective when assessed across the life cycle"

CALS - The DoD Initiative

The evolution of CALS

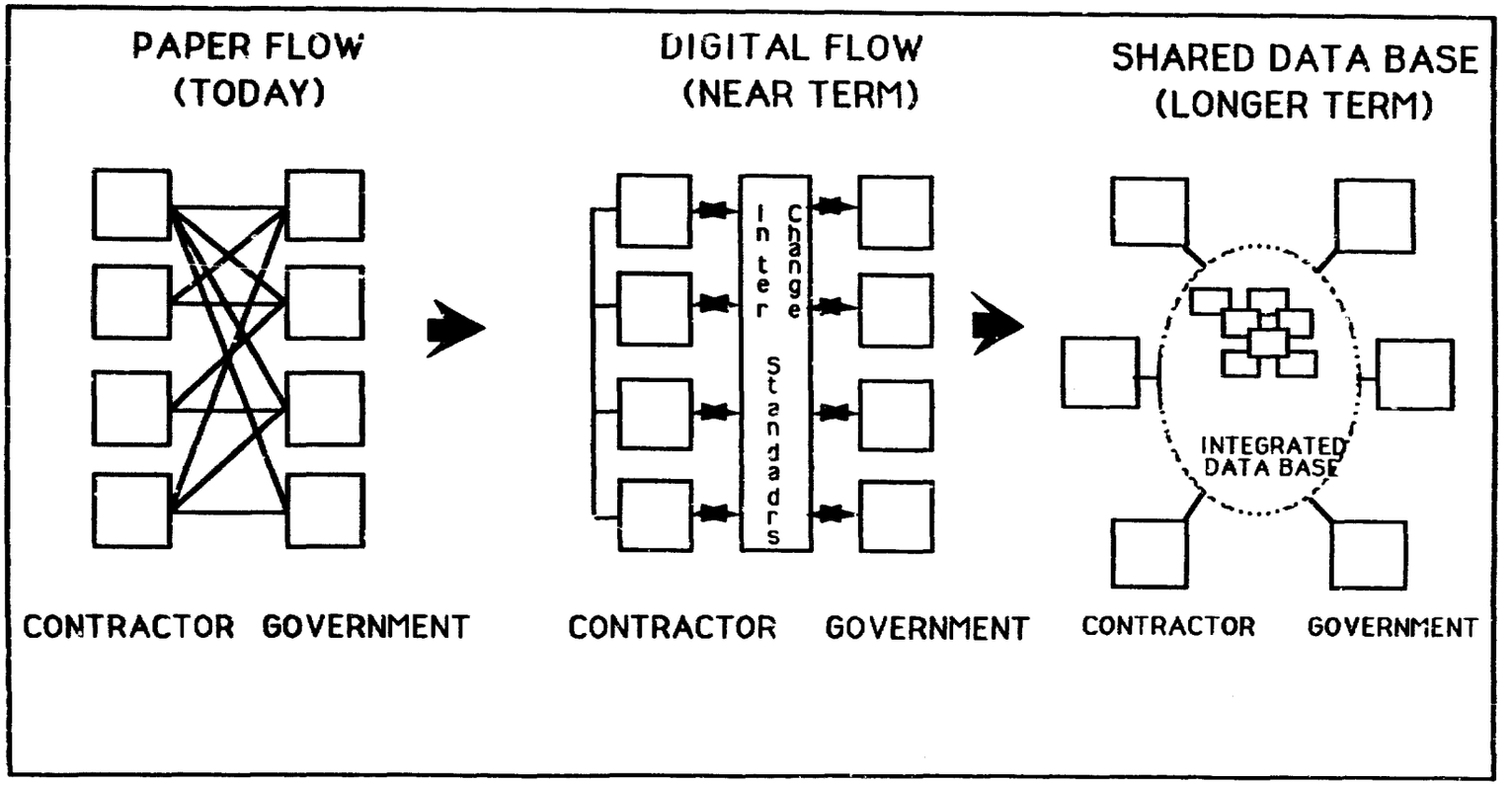
Design **Build** **Support**

Years of continuous transition

- Moving from paper to digital
- Electronic interfacing with customers, partners, and suppliers
- Evolution of integrated information management baseline processes and standards
- Migrating from islands of automation to interfaced systems to integrated systems environment
- Phased approach - near term, long term - evolutionary

CALS - The DoD Initiative

Evolutionary approach leading to smarter use of data



CALS - Concept vs DoD Initiative

CALS Concept vs the DoD Initiative

- Umbrella strategy
  - Do smart, value-added things
  - Eliminate dumb, non-value-added things
  - Use information effectively
  - Customer focus
  - Rethink and reevaluate <u>everything</u>
  - The "ilities"

CALS - a business decision

CALS - The Concept

- Some people think CALS is about data exchange standards.
  - Well, you have to crawl before you can walk.
- Some people think CALS is about a big integrated database that everybody can access.
  - Well, you have to walk before you can run.
- I think CALS is about standing the entire engineering profession, indeed all technical professions, on their heads. CALS is about structuring information, building access paths to it, building tools to use it, and then integrating that information so completely into our work that the job becomes constructing a database to describe our work. The resulting "product", be it missiles, software, or buildings, is merely a manifestation of that database.

CALS ISG Organization

- President (NSIA) - W. H. Robinson, Jr.
- Chairman - R. N. Longuemare
- Vice Chairman - Chief Operating Officer - H. B. Stormfeltz
  - Planning, coordination, advisors
  - Liaison activities
  - Task Groups
    - Concurrent Engineering
    - Information Management
    - Acquisition
    - Logistics Processes
    - Education / Training
    - Small Business
    - International

CALS ISG Organization

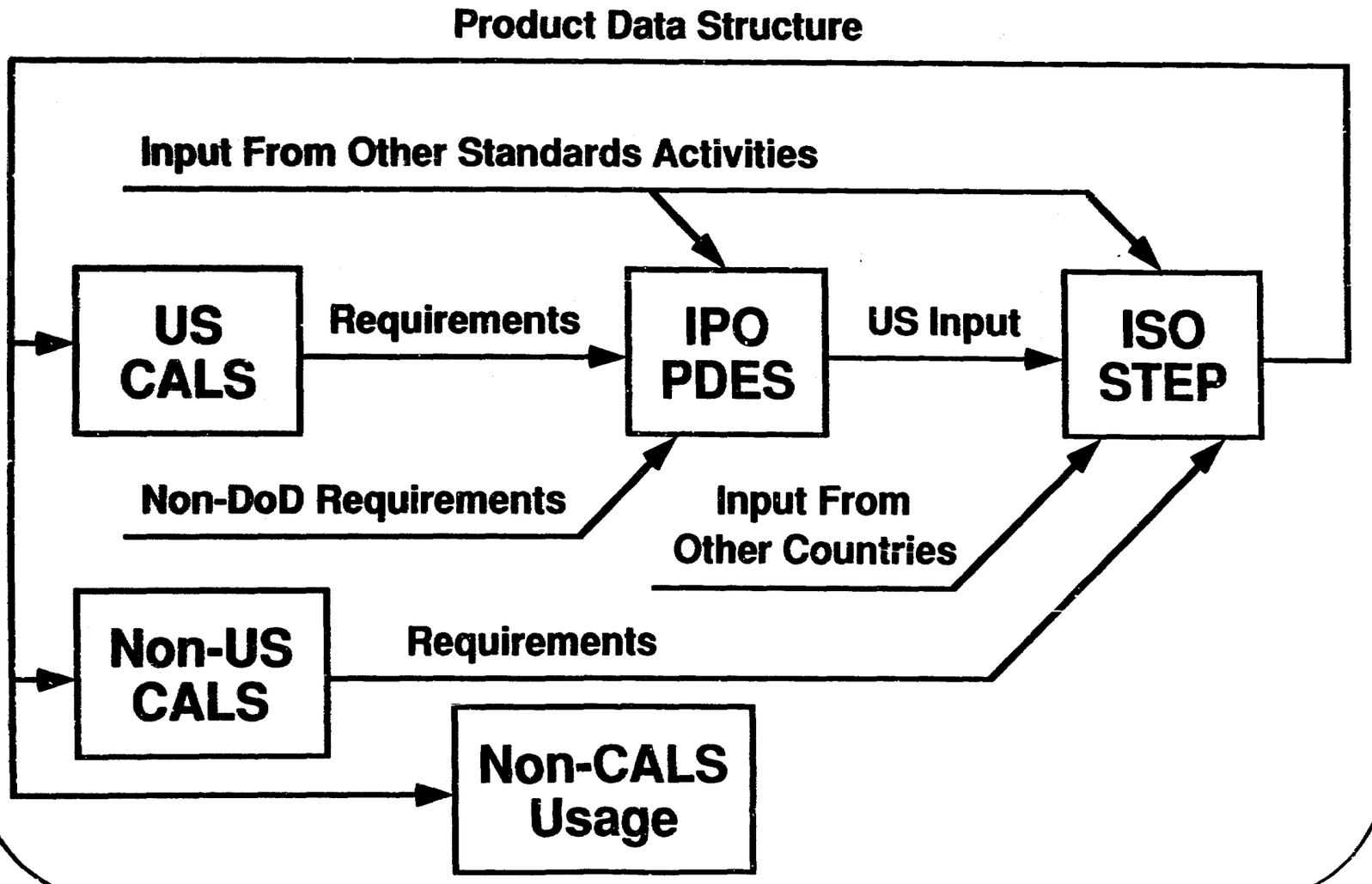
Information Management Task Group

- System / Data Protection Working Group
- Configuration Management Working Group
- Information Integration Working Group
- Standards Working Group
- Information Technologies, Software and Modeling Working Group
  - Software Committee
    - Software Products Committee (also reports to Standards WG)
    - Systems Products Committee
    - COTS Products Committee
  - Modeling Committee
  - Communications / EDI Committee
  - Data Base Management Committee
  - Intelligent Systems Committee

CALS Implementation Standards

- Relationship between activities
- CALS standards
- Other industry standards
- PDES / STEP
  - What is it?
  - IPO - the IGES/PDES Organization
  - ISO TC184/SC4 (STEP)

Relationship Between Activities



CALS Standards

- One cornerstone of the CALS strategy is the use of standards to facilitate the sharing and exchange of information.
- CALS standards:
  - Select from the field
  - Supplement a standard e.g. provide an application profile for a more generic standard
  - Fill in the gaps - or identify the gaps to be filled
- MIL-STD-1840A plus standards referenced by it

MIL-STD-1840A

- MIL-STD-1388
- MIL-D-28000
  - IGES
- MIL-M-28001
  - ISO 8879 - SGML
- MIL-R-28002
  - FIPS PUB 150 - CCITT Group 4
  - ISO 8613 - ODA
  - ISO 8824, 8825 - ASN.1
- MIL-D-28003
  - FIPS PUB 128 - ANSI X3.122 - CGM

Other Industry Standards

- IEC
- ISO
- ANSI
- Standards developing organizations such as the IEEE Computer Society, EIA, ...
- Recent ATA requirements for digital format
  - Vector graphics - CGM
  - Raster graphics - CCITT, Group 4
  - Text - SGML
- Cross pollinization of standards efforts is being greatly facilitated by cross participation

PDES / STEP

Setting the stage

- The challenge is cutting time and cost in an increasingly complex technological environment.
- The solution is an entirely new, more integrated way of doing business - industrial automation on a whole new scale - requiring a common electronic method of accurately communicating product design and manufacturing information.
- This requires the creation of a new kind of standard - the product data standard

PDES / STEP

Product Data Exchange using STEP (PDES)

- **A universal computer language and format standard for the exchange and sharing of all data necessary to describe a product and how it is made.**
- **An <u>international</u> standard facilitating:**
  - **Inexpensive, specific applications working from a common, complete definition of a product - suppliers**
  - **High power, broad application tools (often expensive and with their own internal proprietary database) working together - reliability analysis with cost analysis**
  - **Concurrent Engineering - improvements here are probably the biggest payoff in both flow time and dollars**

IPO

IPO - IGES/PDES Organization

- Began as a government/industry initiative
- Currently over 100 companies and organizations
- Seeks the ability to build better products in less time and cost through effective and efficient communications
- Development of a specification for product data exchange among current and future CAD/CAM systems
  - A robust specification for the exchange and sharing of product data across all product life cycles phases
  - Achieve and document industry/government consensus on requirements for exchange and sharing of product data
  - Maximize the use of existing technology and develop the new technology necessary for completion

IPO

IGES - Initial Graphics Exchange Specification

- Defines a neutral data format which allows the digital exchange of information among CAD systems
  - Two and three dimensional wireframe representations
  - Surface representations
  - Geometry of finite elements
  - Process instrumentation diagrams
  - 3-D piping models
  - Support for electronic products
- Assumes a person is available to interpret data meaning, i.e. a hole vs a rod

IPO

PDES - Product Data Exchange using STEP

- United States development activity in support of the international standard known as STEP (Standard for the Exchange of Product Model Data)
- Focuses on exchanging product models with sufficient information content so they can be interpreted directly by advanced CAD/CAM/CAE application programs
- Geometry plus nongeometric data such as manufacturing features, tolerance specifications, material properties, surface finish specifications, . . .
- Not "just" a graphical representation, not "just" geometry, but a complete definition of the product and its associated data across all life cycle phases

ISO TC184 / SC4 (STEP)

STEP - Standard for the Exchange of Product Model Data

- The International Organization for Standardization (ISO) facilitates the national interchange of goods and services and encourages cooperation in economic and technological endeavors
- ISO Technical Committee 184 (TC184) - industrial automation systems
- Subcommittee 4 (SC4) - representation and exchange of digital product data - the effort known as STEP
- Documents developed as part of the IPO PDES activity are periodically submitted to ISO in support of the STEP effort.
- Many IPO PDES members are actively involved in ISO STEP

SC4 (STEP) - Organization

ISO TC184 / SC4 - representation and exchange of digital product data

- **Project Management Advisory Group**
- **Editing Committee**
- **WG2 - Standard Parts Library**
- **WG3 - Product Modeling**
  - **All product representation areas including**
  - **P16 - Software Products**
- **WG4 - Qualification and Integration**
- **WG5 - STEP Development Methods**
- **WG6 - Conformance Testing**
- **WG7 - Implementation Specifications**

Software Products

- Standards Coordination
  - What groups are involved
  - What's been accomplished
- CALS Software Products Committee
  - Problem
  - Charter
  - Results, directions
- PDES/STEP Software Products
  - Scope of effort
  - Tasks in work

Standards Coordination

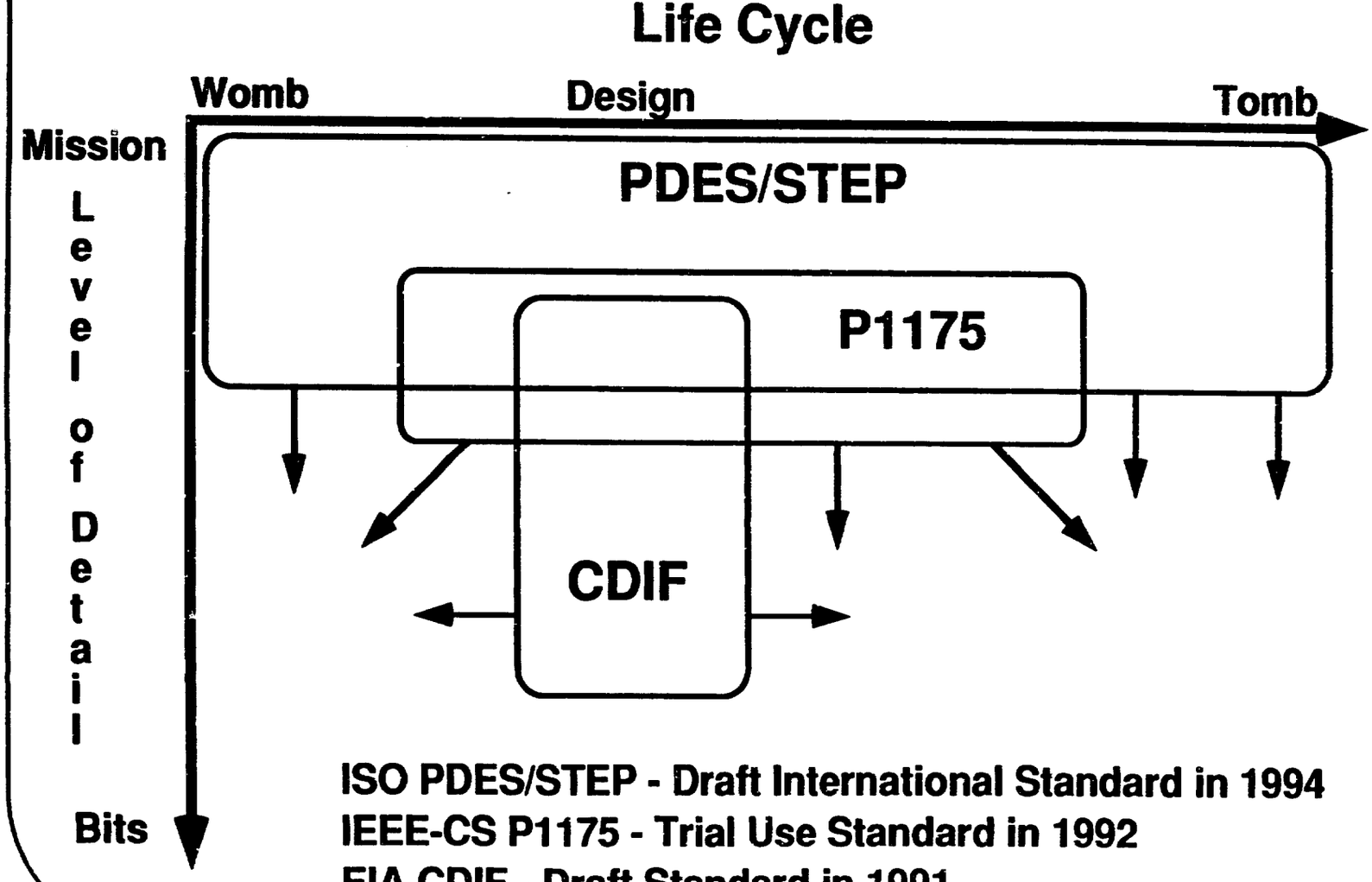
Principal Players

- EIA CASE Data Interchange Format (CDIF)
- IEEE-CS Task Force on Professional Computing Tools (P1175)
- CALS ISG Software Products Committee
- PDES Software Products Committee
- STEP Software Products (future)

Other Interested Parties

- PDES Dictionary/Methodology (especially the SUMM project)
- STEP Product Functionality (future)
- IBM AD/Cycle
- Potentially IRDS, SC7, P1074, ...

Software Product Standards Coverage and Future Directions



ISO PDES/STEP - Draft International Standard in 1994
IEEE-CS P1175 - Trial Use Standard in 1992
EIA CDIF - Draft Standard in 1991

CDIF

Electronics Industry Association (EIA) - CASE Data Interchange Format (CDIF)

- **This is an international effort leading to a standard for interchange of information between tools that support Computer Aided Software Engineering (CASE).**
- **The emphasis is on models of the data used by such tools. CASE tool vendors are heavily involved. A large portion of the work is being done in the United Kingdom.**
- **Some of the participants are aware of US DoD software practices, most are not.**

P1175

Institute for Electrical and Electronic Engineers (IEEE) Computer Society - Task Force on Professional Computing Tools (P1175)

- **This is a two pronged international effort leading to a standard to:**
  - **describe the interconnections that must be considered when buying, building, testing, or using computer tools**
  - **specify an interchange syntax for transferring information between such tools.**
- **This is primarily a user's group. The majority of the work has been done in the USA but there is a strong international flavor.**
- **While some of the participants have a working knowledge of US DoD software practices, most do not.**

Agreements on Overall Approach

- Future cooperation and coordination will be a common goal. Sometimes, it might be necessary to temper the cooperative effort due to specific goal or schedule constraints of a particular group.
- We will move toward a common meta-meta-model. This will take time. Papers on the subject will be cross-circulated between the groups.
- Each group will formally comment on the public drafts of the other groups.
- For meta-model coordination, we are considering dividing up the work on different subject areas between the groups, ensuring that they "fit together" rather than several groups all investing major efforts in the same subject areas. Groups will be encouraged to comment on, and reuse, the modeling efforts of the other groups.
- Extensive use of joint meetings, telecons, cross mailings, etc.

Progress to Date

- Presentations were made by each group to every other group.
- There has been cross attendance at each others meetings.
- A coordination meeting was held in September.
- A joint meeting was held on January 8, 1991 in conjunction with the PDES and CDIF meetings. There were joint technical presentations and discussion/resolution of issues of interest to the groups (e.g. the meta-meta-model, subject areas, the concept of data flow).
- There is a common calendar of meetings.
- There have been several joint telecons.
- A mailing matrix was developed between the groups. All work generated by one group is being mailed to each of the other groups.

Progress to Date

- P1175/STL being considered as an alternate syntax for CDIF
- CDIF a potential interconnection "tool" for use by the P1175 reference model
- CDIF giving P1175 increased rigor
- PDES providing P1175 with reverse engineered data model
- P1175 and PDES giving CDIF more powerful representations
- CDIF and P1175 providing PDES SPC with baseline models
- PDES may provide a functional model for CDIF and P1175 in the future
- P1175 has benefitted by applying some CDIF concepts to its work.

CALS Software Products Committee

Problem: Software deliverables, a major part of systems, do not currently benefit from the CALS strategy - neither process nor deliverables. Current deliverables are characterized by:

- **Poor access to information**
- **Virtually no access to relationships between information**
- **Similar, related information delivered many different ways**

Solution must address:

- **Current state of affairs**
- **Delivery of accessible info in a standard electronic form**
- **More effective information structure**
- **Migration from electronic accessible product to PDES information structure**

CALS Software Products Committee

- And so, the unanswered questions were:
  - How to deliver a Software Product in a CALS environment?
  - What effects does this have on the development of that product?
  - What about the non-CALS environment of the future?
- The answers affect <u>all</u> software development activities and activities related to software development.

CALS SPC Charter

The committee's charter is to study, document, and make recommendations on the CALS compliance of software functional standards, life cycle processes, and deliverables for the:

- **Near term - define requirements for digital delivery of software products**
- **Longer term - define issues and potential solutions for more effective utilization (e.g. concurrent engineering) of software products information in an integrated database environment**

CALS SPC Charter

Notice that the charter is to study, define issues and requirements, and make recommendations.

The charter is <u>not</u> to implement.

- **Implementation for near term**
  - **DoD contracted or in-house effort**
- **Implementation for longer term**
  - **Coordinated effort with other groups such as PDES**

CALS SPC Charter

Software Products being considered:

- MIL-STD-2167A and supporting DIDs for Mission Critical Systems
- MIL-STD-7935A and supporting DIDs for Automated Information Systems
- These two standards account for the vast majority of DoD software.
- Define, fairly precisely, document organization and content
  - Much more like a blueprint than a novel
  - The document <u>structure</u> and <u>content</u> define the product to be built

CALS SPC - Near Term

Efforts focussed on near term:

- Determined overall approach to applying CALS to software product deliverables
- Validated the development approach
- Obtained government and industry needs, goals, and benefits desired from software products in a CALS environment
- Drafting the final report

Near term solution will utilize SGML to identify the content of, and the relationships between, tagged textual information.

CALS SPC - Near Term

Benefits and payoffs of near term efforts:

- Information is portable between platforms, systems, and tools
- Paper documents become electronic information - not just a stream of characters, but information with meaning
- Documents, and sets of documents, become a computer library of information
- A bridge to CALS long term activities
- Allows automation of processes surrounding documents
  - Semi-automatic generation of documents
  - Automatic "non-technical check"
  - Automated consistency check between documents

CALS SPC - Long Term

Long term objectives:

- Define the requirements for "the software product" in an integrated database world
- Review, validate, and test implementations to ensure that the defined requirements are being properly met
- Interface with related technical areas/efforts
- Determine the benefits/impacts of working in this type of environment and educate the acquisition and program management communities
- Develop a transition plan

CALS - Summary of Findings

- An analysis of information products is required to define accessible information
- Analysis of several different programs is required to develop a common information set
- Priority of addressing information products must be based on return on investment to government and industry
- Factors of complexity along with product usefulness are major considerations for establishing priority in developing CALS information product standards
- Preparing activities of the functional standards for information product content and structure must be involved in the analysis, enhancement, and introduction of CALS into those standards

PDES/STEP Software Products

Scope

The Software Products Committee will develop the STEP models for software product data supporting the full life cycle. The scope of software considered includes all software embedded in, or associated with, any product describable by PDES/STEP. Software Product data describes requirements, design, implementation(code), test, and support documentation such as user manuals, installation instructions, etc. Project administration data is outside of this scope.

STEP/PDES Software Products Tasks

Management and administration

- Planning
  - Goals, tasks, resources, assignments, schedules, interdependencies, meetings, ...
  - Ongoing
- New Project within STEP for Software Products
  - Complete July 1991
- Enterprise Model development
  - Draft July 91
- Member recruitment and support
  - Ongoing

PDES/STEP Software Products Tasks

Software Products Model development

- Support for model development
- Develop Software Products Information Model
  - Develop Software Products Scope Component
  - Develop Software Products Process Component
    - Use P1074 as a base
  - Develop Software Products Data Component
    - Use P1175, CDIF, and P1074/2167A/SLCSE mappings as a base
  - Develop Software Products Dynamic Component
  - Keep Components correlated
- Identify partitions in Software Products Information Model
- Build the identified parts
  - Resource Models and Application Protocols as required

STEP/PDES Software Products Tasks

Liaison efforts

- CALS, CDIF, P1175, IWCASE, AD/Cycle?, IRDS, SLCSE, SC7, others?
- CASE vendors, SPC, SEI, AIA, GUIDE, others?
- Determine overlaps and interfaces to other PDES/STEP efforts
  - Dictionary/Methodology, Product Functionality, Electrical, Product Life Cycle, PSCM, Library Structures, Integration, others?

STEP/PDES Software Products Tasks

Committee Repository

- Develop interim repository
  - Develop list of items
  - Catalog, store, file, existing items
  - Develop a bibliography
  - Develop an Issues Log
  - Basic capability October 90, up-to-date July 91
- Develop a prototype repository capability
- Transfer appropriate paper based information to the electronic repository
- Maintain repository information

CDIF Committee Participants

* Aerospace Corporation
* Anderson Consulting
- ATA
* Boeing Computer Sciences
* Cadre Technologies
* CCTA (England)
* DEFT
* DuPont Company
* General Dynamics
* George Mason University
* Georgia State University
* Hewlett Packard
- IBM
* Index Technology
- Institute of Software Engineering
* LBMS (England
- Lockeed
- Mark V System
- Mentor Graphics
* MITRE Corporation
* Oracle (Europe)
* Popkin Software and Systems
* Sage Software
* Siltronics
* University of Lowell
- USAF (EDS/AVF)
* Yourdon, Inc.

*Current members

CDIF Prototype Experimenters

Advanced Systems Technology
Asset-Logic/RDD
Boeing
Cadre
Digital
ESA
ICONIX
IDE
Index Technology
Knowledgeware
LBMS
MESA Systems
Oracle
Sage

CDIF Background

- CDIF effort = two-year old effort
- EIA Interim Standard = fall 1991
- CDIF prototype development = training in May 1991
- Associated standards activities
  - CALS: Computer-Aided Acquisition and Logistics Support
  - IGES: Initial Graphics Exchange Specification
  - IRDS: Information Resource Dictionary System
  - PDES: Product Data Exchange using STEP
  - STEP: Standard for Exchange of Product Model Data
  - SUMM: STEP Unification Meta-Model

CDIF--A Set of Standards

- Framework for Modeling and Extensibility (EIA-PN2387)
- Standardized CASE Interchange Meta-Model (EIA-PN2329)
  - Part 1: Semantic Model
  - Part 2: Presentation Model
- Transfer Format Definition (EIA-PN2389)
  - Part 1: General Rules for CDIF Syntaxes and Encoding
  - Part 2: CDIF Transfer Format Syntax - SYNTAX.1
  - Part 3: CDIF Transfer Format Encoding - ENCODING.1

Software Products

What do we mean by "Software Products"?

- **Software, just like an airplane or a building, can be a product. It can also be a tool. It depends upon your point of view in any given situation.**
- **The subject here is software as a product; how it is described, how it is developed, how it relates to other products, etc.**
- **The subject is <u>not</u> the use of software to satisfy some need.**
- **This definition covers not only the code, but related information products such as requirements and design specifications, user manuals, etc.**

ISO TC184 / SC4 (STEP)

STEP - Standard for the Exchange of Product Model Data

- The International Organization for Standardization (ISO) facilitates the national interchange of goods and services and encourages cooperation in economic and technological endeavors
- ISO Technical Committee 184 (TC184) - industrial automation systems
- Subcommittee 4 (SC4) - representation and exchange of digital product data - the effort known as STEP
- Documents developed as part of the IPO PDES activity are periodically submitted to ISO in support of the STEP effort.
- Many IPO PDES members are actively involved in ISO STEP

PDES/STEP Software Products

Scope

The Software Products Committee will develop the STEP models for software product data supporting the full life cycle. The scope of software considered includes all software embedded in, or associated with, any product describable by PDES/STEP. Software Product data describes requirements, design, implementation(code), test, and support documentation such as user manuals, installation instructions, etc. Project administration data is outside of this scope.

PDES/STEP Software Products Status

- Expect final SC4 approval at July meeting
- Software Products Model Development
  - Develop Software Products Information Model (SPIM)
    - Scope, Process, Data Components
    - Use of IEEE-CS P1074 as a baseline for high level processes and data
  - Partition the SPIM for individual applications
- Highly involved in liaison efforts
  - IEEE-CS P1175, EIA CDIF, CALS ISG SPC, IRDS, ...

CALS SPC Charter

The committee's charter is to study, document, and make recommendations on the CALS compliance of software functional standards, life cycle processes, and deliverables for the:

- **Near term - define requirements for digital delivery of software products**
- **Longer term - define issues and potential solutions for more effective utilization (e.g. concurrent engineering) of software products information in an integrated database environment**

CALS SPC - Near Term

Efforts focussed on near term:

- Determined overall approach to applying CALS to software product deliverables
- Validated the development approach
- Obtained government and industry needs, goals, and benefits desired from software products in a CALS environment
- Drafting the final report

Near term solution will utilize SGML to identify the content of, and the relationships between, tagged textual information.

CALS SPC - Long Term

Long term objectives:

- Define the requirements for "the software product" in an integrated database world
- Review, validate, and test implementations to ensure that the defined requirements are being properly met
- Interface with related technical areas/efforts
- Determine the benefits/impacts of working in this type of environment and educate the acquisition and program management communities
- Develop a transition plan

NIST ISEE AND PCIS WORKSHOP

June 6, 1991
Sheraton Hotel
Redondo Beach, CA

IEEE-CS P1175 - A Standard Reference Model for Computing System Tool Interconnection

By: David Sharon
President
CASE Associates Inc.
15686 S. Bradley Road
Oregon City, OR 97045
(503) 656-0986

Member of the CASE Technology Group

IEEE-COMPUTER SOCIETY'S TASK FORCE ON PROFESSIONAL COMPUTING TOOLS

Outline

- Purpose and Scope
- Tool Interconnections
- Organization
- Membership
- Status

WHO ARE THE PROFESSIONALS?

Professionals are individuals who earn income by working with computing systems.

- **Managers**
- **Systems Analysts**
- **Systems, Software, Hardware Engineers**
- **Programmers**
- **Testers**
- **Teachers**
- **Consultants**

WHAT ARE COMPUTING SYSTEMS?

Computing systems include:

- **Systems made of hardware only**
- **Systems made of software only**
- **Systems made of hardware and software**

WHAT ARE COMPUTING SYSTEM TOOLS?

"Computing System Tools" include two types of tools a professional would use to develop computing systems.

General purpose tools:

Specific purpose tools:

WHAT ARE COMPUTING SYSTEM TOOLS?

General purpose tools:

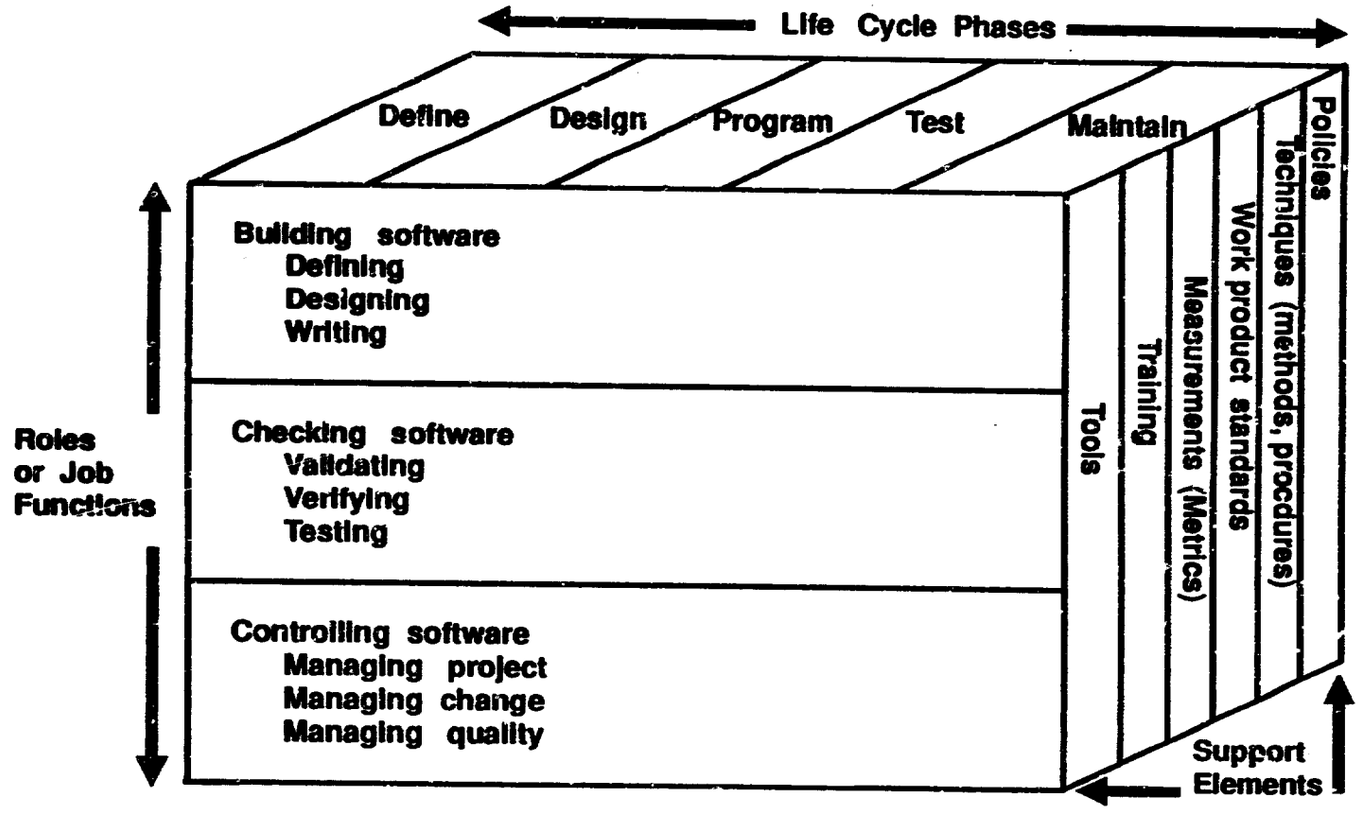
- Tools developed for use in many applications
- Word processors, database systems, etc.
- These tools are not addressed by the Task Force.

Specific purpose tools:

- Tools developed explicitly for creating, testing, and/or maintaining computing systems
- Requirements specifiers, compilers, test tools, etc.
- These tools are the focus of the Task Force.

IEEE
COMPUTER SOCIETY'S
TASK FORCE
ON
PROFESSIONAL
COMPUTING TOOLS

Tool to Organization Interconnections
Organizational Context for Tools



Charter, Purpose and Scope of the Task Force Services

Services include:

Standards impact assessment reports

Charter: Itemize and evaluate existing and emerging standards that could impact professional tools to avoid redundant standards and identify needed standards.

On-line tools inventory

Charter: Build an on-line tools database that is updated by vendors and accessed by users to keep professionals current relative to tool availability.

Charter, Purpose and Scope of the Task Force Services

Services include:

Tool openness guidelines

Charter: Develop a scale of openness (ie. guidelines which characterize degrees of openness).

Tool interconnection standards

Charter: Build a standard reference model (framework) for interconnecting tools to allow old, current, and future tools to work together.

Charter, Purpose and Scope of the Task Force Computing System Tools

CASE Tools are included and may mean:

Computer Aided System Engineering

Computer Aided Software Engineering

Concentrated Attempt to Snow Everyone

IEEE CS Tools Task Force
Organization

Executive Board:

Chairperson	Robert Poston, PEI
Vice Chairperson	Francois Coallier, Bell Canada
Treasurer	Corey Clinger, BELLCORE Research
IEEE-CS Finance Coordinator	Richard Cain, CS
Secretary	Dave Sharon, CASE Consulting Associates
Assistant Secretary	Delores Wallace, NBS
Publicity Chairperson	To Be Appointed
IEEE-CS Coordinator	Lee Blue, CS

Working Groups

Standards Impact Assessment	Tom Kurihara, DoD
Tools Inventory	Peter Eirich, Westinghouse
Tool Interconnections	Robert Poston, Dave Sharon
Scale Of Openness	Sam Redwine, SPC
Tool Training	Paul Radding, Technology Decisions
	Paul Jorgensen, Research & Technology Institute

IEEE CS Tools Task Force
Membership

MEMBERSHIP IS OPEN. Anyone may participate.

At balloting time only members of the

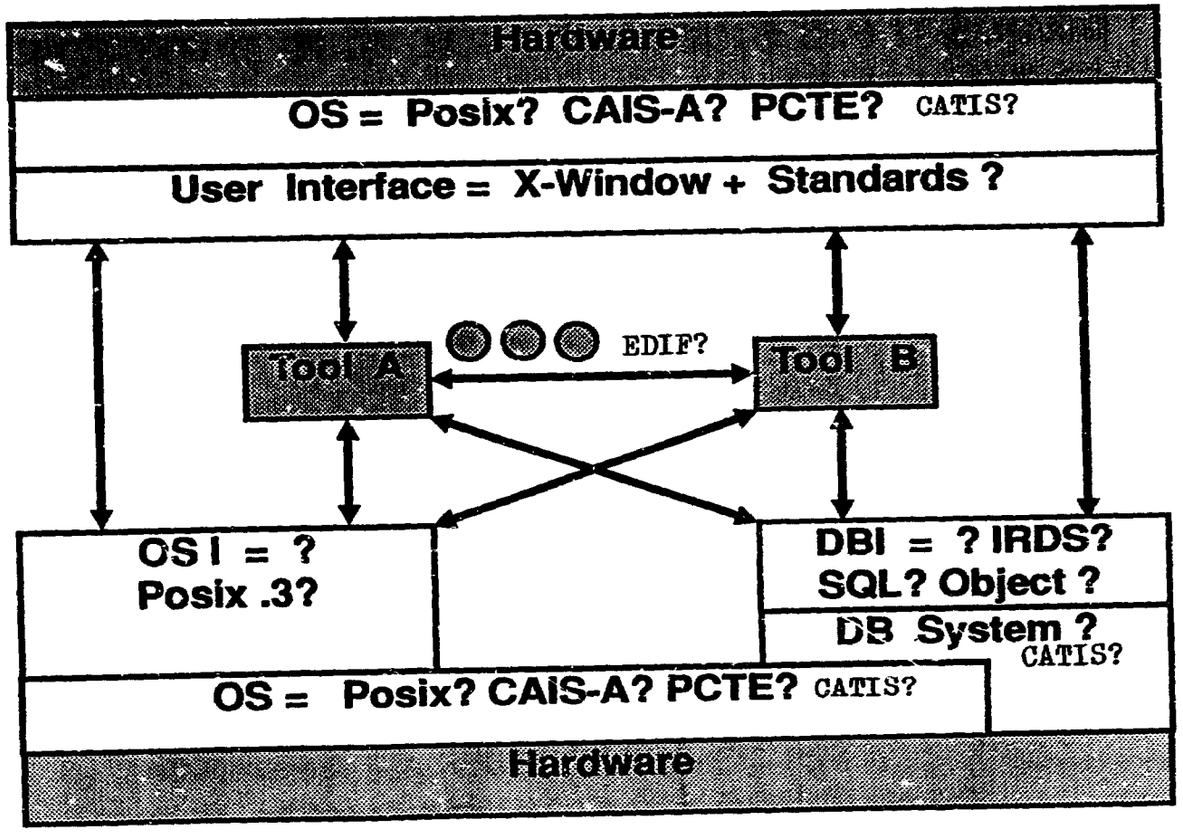
IEEE,

Computer Society, or

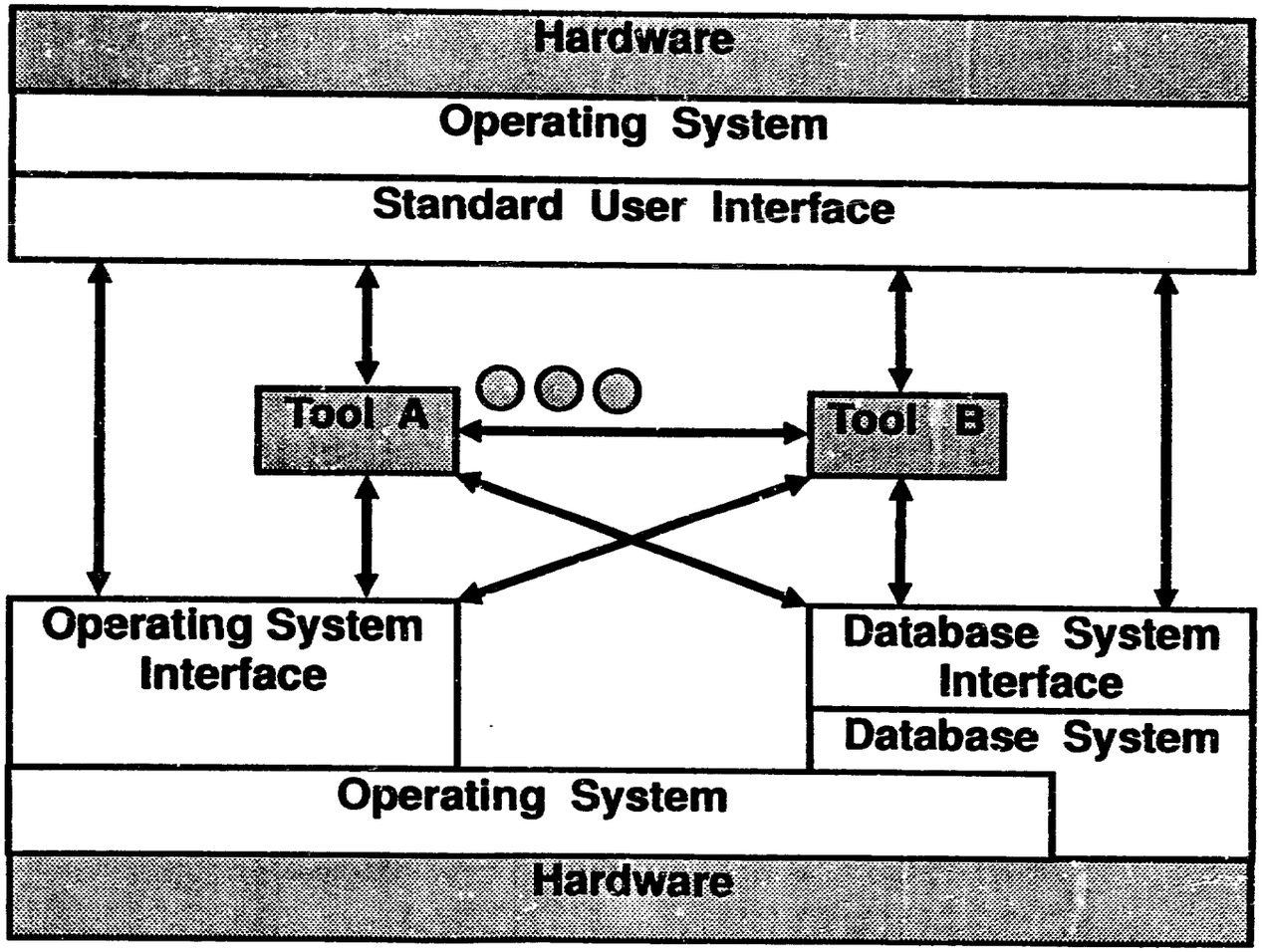
Affiliate organizations may officially ballot.

Every ballot from an active working group participant is recorded and negative comments are resolved by the working group.

TOOL INTERCONNECTIONS
Database Interconnections: Tool-to-Database-to-Tool



Tool to Platform Interconnections
Platform Context for Tools



Tool to Platform Interconnections
Platform Context for Tools

Many industry (professional and defacto) standards for tool to platform interconnections already exist.

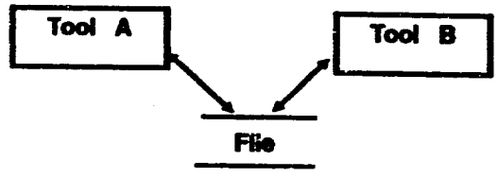
Operating Systems:	**POSIX, VMS, MVS, etc.**
Database Systems:	**SQL, IRDS, etc.**
Human Interface Systems:	**Open Look, Motif, etc.**
Communication Systems:	**OSI, Ethernet, etc.**
Hardware Systems:	**PC, VAX, Apple, etc.**

Interconnection Methods

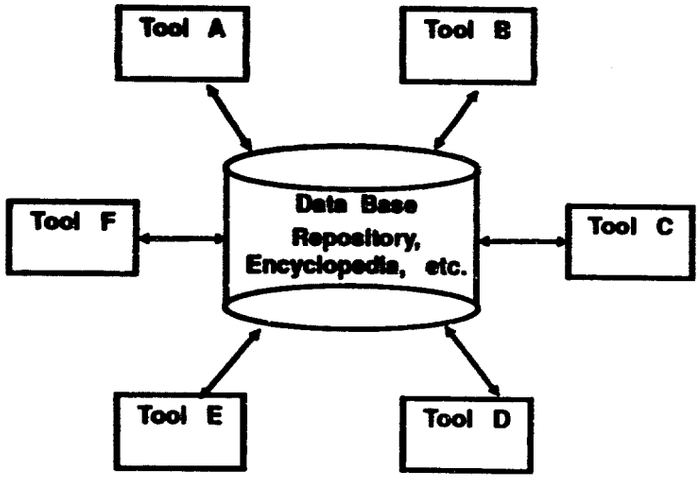
Direct



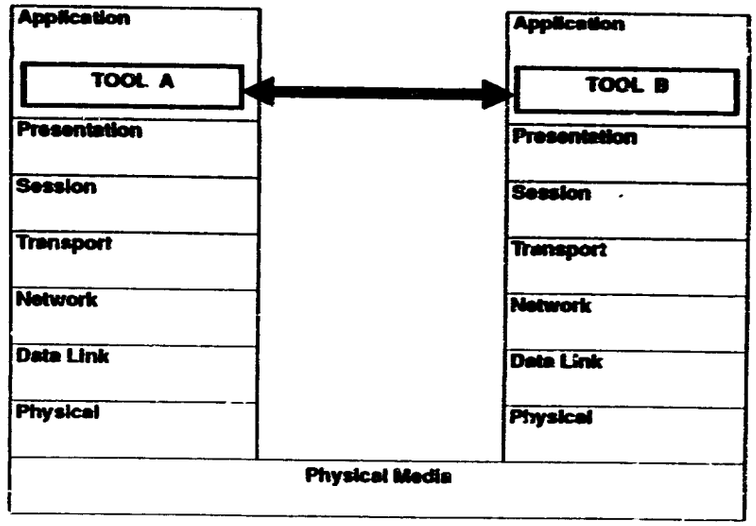
File



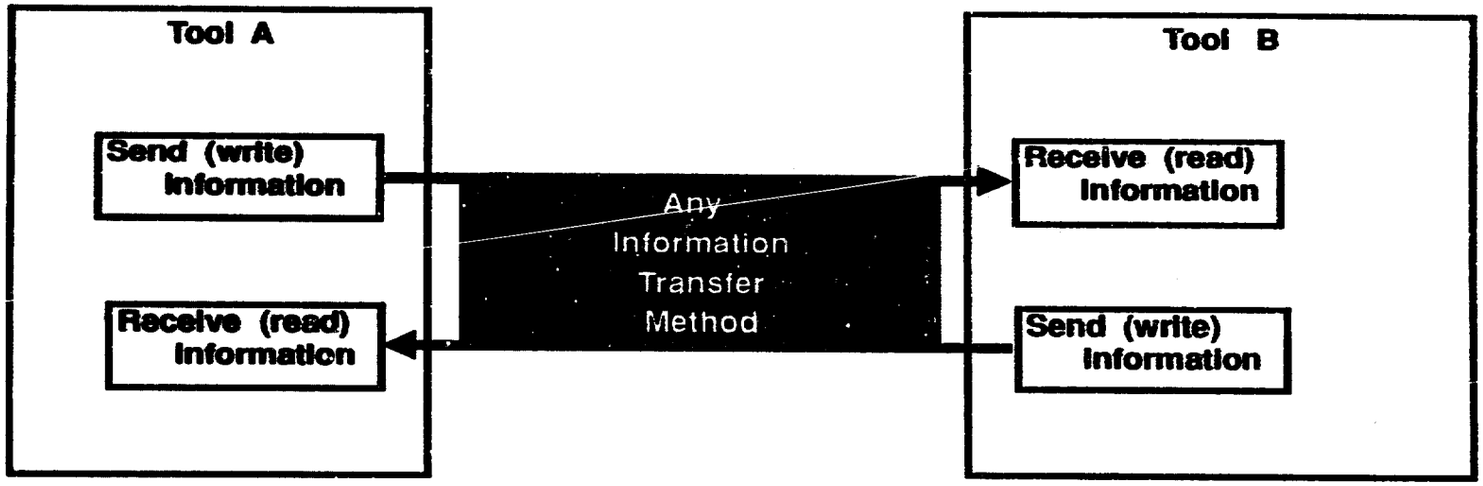
Database



Communication System



Interconnection Methods



2.3 Support view of a tool

What does it take to make a tool successful in an organization?

Perhaps the most indirect interconnection between an organization and a tool is the support elements. Support elements are those things that an organization must provide to maximize the benefits of a tool:

1. Policies: Policies are written descriptions of what job functions must perform what activities in which life cycle phases. Policies may also be called Directives, Instructions, or Methodologies.
2. Techniques: Techniques are written descriptions of how to perform an activity. Techniques may also be called methodologies, methods, or procedures. See IEEE P1016.2, Guide to Software Design Descriptions, as an example.
3. Work product standards: Work product standards are written descriptions of the items (documents, code, or data) that must be produced in an activity. Work product standards may also be called documentation format standards. See the following standards as examples:
   - 3.1. ANSI/IEEE Std 830-1984, Software Requirements Specifications
   - 3.2. ANSI/IEEE Std 1016.1, Software Design Descriptions
   - 3.3. ANSI/IEEE Std 829-1983, Software Test Documentation
   - 3.4. ANSI/IEEE Std 828-1989, Software Configuration Management Plans
   - 3.5. ANSI/IEEE Std 730-1984 (Rev 1989), Software Quality Assurance Plans
   - 3.6. ANSI/IEEE Std 1012-1986, Software Verification and Validation Plans
   - 3.7. IEEE Std 1074, (in process), Software Life Cycle Processes
   - 3.8. IEEE Std 1058.1-1987, Project Management Plans
4. Measurements: Written descriptions of how to quantitatively evaluate work products (measurements may also be called metrics). See the following standards as examples:
   - 4.1. IEEE P1044, (in process) Standard for Classification of Software Errors, Faults, and Failures
   - 4.2. IEEE P1045 (in process), Standard for Software Productivity Metrics
   - 4.3. IEEE Std 982.1-1988, Dictionary of Measures to Produce Reliable Software
   - 4.4. IEEE P1061, Standard for Software Quality Metrics Methodology
5. Training: Training is experience in the application of support elements.
6. Tools: Tools are mechanizations that aid or replace human effort.

Support elements aid a person who is new to a job or tool by providing the answers to the following questions:

1. What am I supposed to do? When do I do it? - Policies
2. How am I supposed to do it? - Techniques
3. What am I supposed to produce? - Work product standards
4. How will I know it is a good work product? - Measurements
5. Where do I get the answer to these questions? - Training
6. What is the easiest way to do the right thing? - Tools

2.4 Tool to organization interconnection standard profile

A context for designing, using, or testing a tool is established by the organization that will use the tool. One step toward growing knowledge of the context is to identify each of the interconnections or relationships between an organization and its tools. The "Tool to Organization Interconnection Standard Profile" form at the end of this chapter has been designed to help identify all of these interconnections.

Figure 2. Tool to Organization Interconnection Profile

Tool to Organization Interconnection Standard Profile

Tool Name: _____ Date: _____

Interconnections	Name (s) of Applicable Standard (s)
Job function	
Primary user	
Secondary users	
Final user	
Life cycle	
Phase of first use	
Phase of second use	
Phase of final use	
Support element (s)	
Policy(ies)	
Technique(s) (Methodologies)	
Work product standard (s)	
Measurement(s)	
Training Course(s)	

Copyright © IEEE, 1990

3 Reference Model for Tool to Platform Interconnections

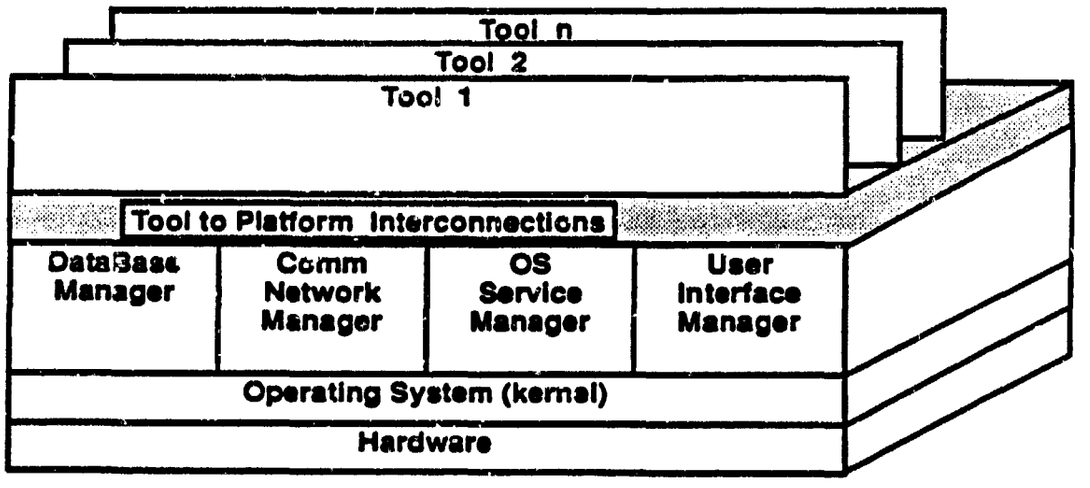
Just as tools are interconnected to the organizations that use them, tools are interconnected to the platforms where they operate. A platform is a collection of hardware and software components. The hardware components include devices such as central processing units, disk drives, printers, and displays. The software components include devices such as operating systems, database systems, communication systems, and human interface systems. A tool platform may also be called a development environment or an operating environment.

An organization may use only one platform and the tool will be operational on that platform. However, some organizations may use a variety of platforms and platform configurations, including:

1. stand-alone work-centers (terminals, workstations, and/or personal-computers)
2. individual work-centers connected to a central host
3. individual work-centers interconnected through communication networks.

This variety presents two issues for interconnecting tools. The first issue is the ease of moving a tool from one platform to another platform -- portability. The second issue is the ease of using platform services to allow tools to exchange information within a platform or among platforms -- connectability.

Figure 3. Tool to Platform Interconnections



How well tools interconnect with an organization's available platforms may be considered from the following perspectives:

1. interconnectivity of platforms
2. distribution of tools among platforms
3. configuration of individual platforms

How well a tool interconnects with a specific platform may be considered from the perspective of how the tool uses the standard, available platform services, and controls:

1. operating system services: access, manipulate, and control hardware capabilities for all software programs on a platform
2. database system services: store and retrieve data, text, and graphics information for the application software programs (tools) on a platform
3. human interface system services: accept and display data, text, and graphical information on a platform for the human user of an application program (tool)
4. programming language systems: generate, control, compile, link, load, and debug programs
5. communication system services: send, receive, encode, decode, and route information and service requests between applications software programs (tools) running on different platforms
6. data file exchange systems: either use standard file formats or convert from proprietary formats to standard file formats
7. document exchange systems: transfer complete documents

The productivity and effectiveness of a tool can be reduced if there is work or effort involved in moving data from a tool on one platform to a second tool on a different platform.

Attributes of a well-defined interface specification for any tool-to-platform interconnection are described in the POSIX (IEEE P1003.0) system architecture. An interface specification should be open, independent, shared, and documented.

1. Open: The interface has
   - 1.1. the capability of allowing a user or application operating on one platform to communicate with a user or application on another platform
   - 1.2. documented, nonproprietary interface specifications
   - 1.3. standardized connectivity and coupling for each mode of operation with implicit responsibilities to preserve interfaces
2. Independent: The interface supports
   - 2.1. software reuse, platform architecture, and application internals
   - 2.2. multiple releases
   - 2.3. recognized authorized users
   - 2.4. functional recovery
3. Shared: The interface is designed for invocation by more than one application
4. Documented: The interface has
   - 4.1. well-defined syntax, semantics, and services
   - 4.2. well-defined set of parameters invoking equally well-defined sets of actions and responses

3.3 Tool to platform interconnection standard profile

Context for designing, using, or testing a tool is established by the platform where the tool will be used. One step toward growing knowledge of the context is to identify each of the interconnections or relationships between a tool and its platform. The "Tool to Platform Interconnection Standard Profile" form at the end of this chapter has been designed to help identify all of these interconnections.

Figure 4. Tool to Platform Interconnection Profile

Tool To Platform Interconnection Standard Profile

Tool Name: Date: _____

Interconnection _____ Applicable Standard (s) _____

Hardware _____

Operating Systems(s) _____

Database System(s) _____

Language Systems(s) _____

Communications System(s) _____

Human Interface System(s) _____

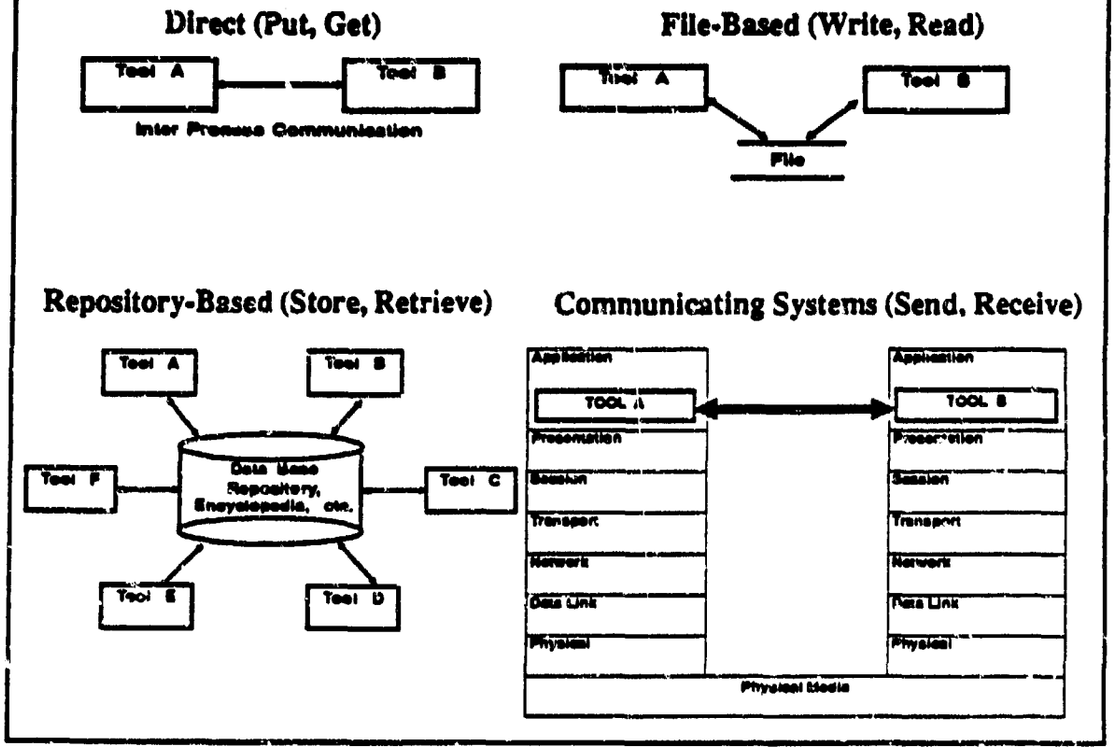
Data File Exchange Format(s) _____

Copyright © IEEE, 1990

4 Methods for Information Transfer Among Tools

Information transfer among tools may be visualized in many different transfer methods.

Figure 5. Example Information Transfer Methods

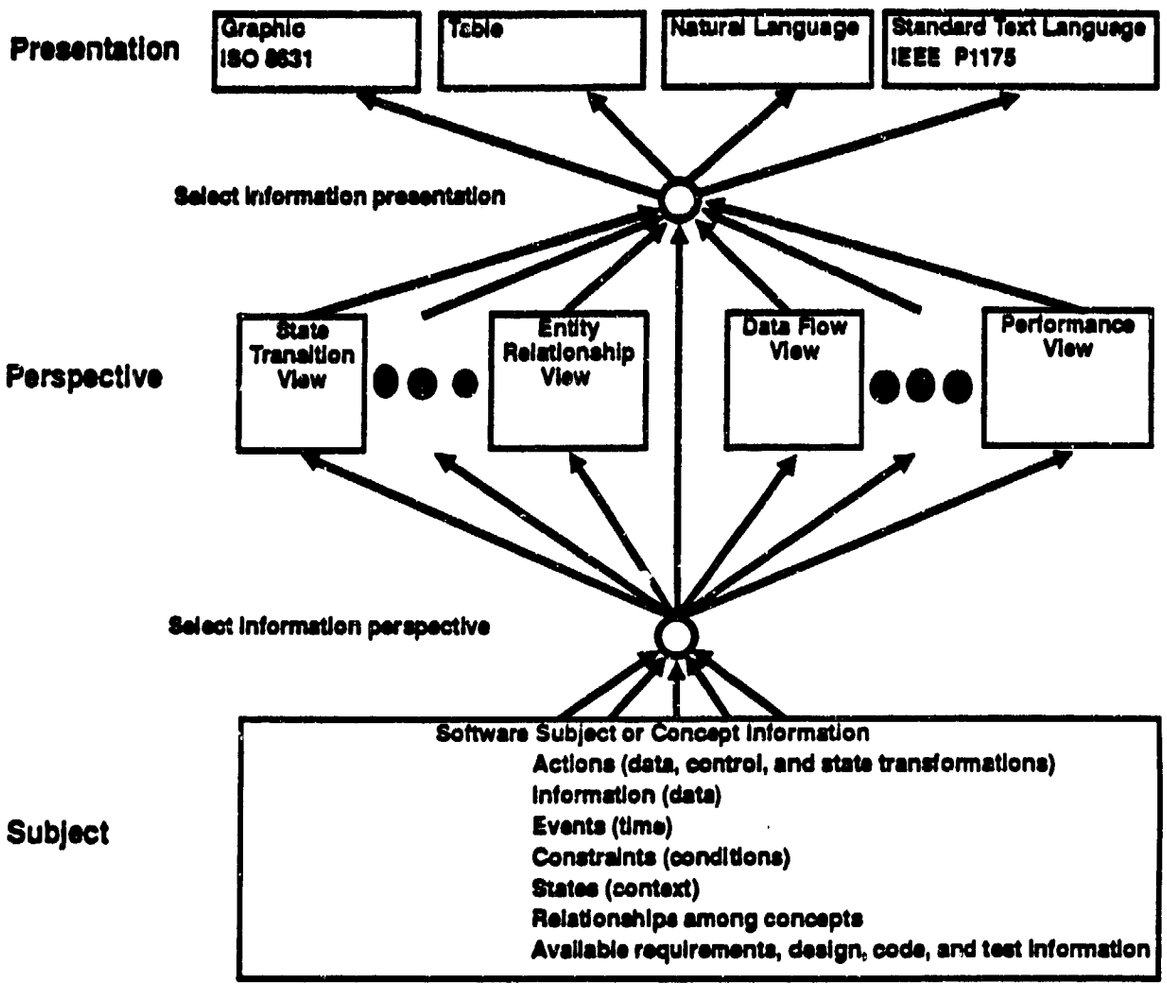


Each of the methods serves a particular need and is best for some particular situation. The direct transfer method is the most efficient from a response time perspective. The file transfer method is best from a simplicity of implementation perspective. The central data base or repository method is best from a tools integration perspective. An in-memory shared working area method of transfer fits somewhere between direct and repository-based transfer methods. The communicating system method is best from an open systems perspective. If the communication systems and repositories are considered to be tools, then all of these information transfer methods may be represented with a standard process model and a standard information model as illustrated in Figure 6.

The next two chapters of this standard describe information transfer among tools from the perspectives of

1. The processes of information transfer
2. The information transferred.

Figure 7. Subjects, Perspectives, and Presentations



4.3 Descriptions of information being transferred

If information is to be transferred among tools it must be described in a form that a tool can deal with. Each increment of transferred information must have its syntax (form) and its semantics (meaning) defined.

4.3.1 Syntactic (format) information

Syntactic information describes the physical structure or form of subject information. Syntactic information may or may not be transferred among tools, but it will be used in the sending and receiving processes. ISO 8824 and ISO 8825 provide the standards necessary for defining syntactic information. ISO 8824:1987(E) Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1) describes the notation recommended by this standard. ISO 8825:1987(E) Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for ASN.1 describes encoding rules for use with ASN.1. Together 8824 and 8825 describe the following information:

- Atomic data elements (bits) and how those atomic data elements will be grouped for interpretation (coded) into primitive DataItems such as characters and numbers.
- How primitive (unstructured) DataItems may be grouped together into structured DataItems such as words, fields.
- The delimiters (characters, symbols, or counts of characters) that separate primitive DataItems.
- How structured DataItems may be grouped together to form DataItems of larger structures of such as lists, records, files, tables, and databases.

4.3.2 Semantic (meaning) information

Semantic information is the information that represents the meaning of transferred information. In practice, semantic information may be transferred among tools, or it may be expressed as an agreement among the tools and not transferred at all. Here are some examples.

- The concept "process" means a manipulation or derivation of data.
- The concept "functional state" means a collection of one or more operations.
- The concept "state transition" means moving from one state to another state.
- The concept of "data state" means the characteristics of data at an instant in time.

5. Semantic Transfer Language (STL) Overview

This chapter contains an overview of the STL and a summary of its syntactic elements. The STL is presented in top-down order. First, a complete information transfer is described. Next, the language structures are described. Finally, the language primitives are defined.

5.1. STL Goals

Design decisions for the STL were based on satisfying the following goals, in order of decreasing priority:

1. The STL must be parsable. This goal is the minimum requirement for allowing tool interconnection. If information is to be transferred among tools, the tools must be able to read the information. This absolute requirement does not require that the STL be easy to parse, only that it be parsable.
2. The STL must be easy to read by programmers who do not have special training in the STL and do not have special tools. The goal is to express information about software in a form that requires the least amount of specialized knowledge to read. In particular, a software description written in the STL should be readable without needing a special tool. A text editor should be sufficient for reading an STL file.
3. The STL must be easy to write. Programmers who have limited training in STL and no access to special tools must be able to write a description in STL easily.
4. It should be possible to convert the STL into a compact transfer form to make efficient use of machine resources.

Goals 2 and 3 imply that the STL should be a language that is close to natural language. Goal 4 implies that the STL should be a sparse, machine-like language. To meet these diverse goals, the STL is provided in a human-readable, highly stylized sentence form.

5.2. STL Sentence Form

Figure 7 contains a sequence of special natural language sentences. They are special because

1. every sentence is about one, and only one, subject (ie., Action A01).
2. every sentence contains a non-compound verb or verb phrase.
3. every sentence expresses only one relation to or attribute of the subject.
4. every sentence has the same order of words:
    1. a subject with a descriptor
    2. a verb or verb phrase
    3. a direct object with a descriptor or a prepositional or an adverbial phrase

Figure 8 contains an STL sentence that contains the same information as the natural language sentences presented in Figure 7.

Figure 7. Natural Language Sentences

```
Action A01 is allowed in state normal.
Action A01 receives eventitem warning_interrupt.
Action A01 transmits eventitem safety_action1.
Action A01 uses dataitem water_temperature_reading.
Action A01 produces dataitem safety_action_report.
Action A01 has maximum duration time 3 seconds.
```

Figure 8. STL Sentence Example 1

```
Action A01
    is allowed in state normal ;
    receives eventitem warning_interrupt ;
    transmits event safety_action1 ;
    uses dataitem water_temperature_reading ;
    produces dataitem safety_action_report ;
    has maximum duration time 3 ;
    has duration time units "seconds".
```

5.5.3. STL_Clauses

In Figure 11 the first clause in the STL example sentence describes a relation between Action A01 and state normal. It is a relation_clause. The last clause in the example sentence describes one measurable fact or attribute of Action A01. The last two clauses are attribute clauses. Each STL_clause describes either one relationship for or one attribute of the subject of the sentence where the clause appears. The BNF syntax for an STL_clause is described as follows:

```
<STL_clause> ::= <relation_clause> | <attribute_clause> | NULL
```

Clauses may or may not be present. When clauses are present, they may appear in any order. In any STL_sentence a particular clause may be present once and only once. STL_clauses must begin with a keyphrase and are separated by a semicolon. The last clause in a sentence is properly followed by a period, not a semicolon. However, for the convenience of STL users, the construct ";." is allowed at the end of a sentence by the NULL alternative above.

Figure 11. STL Sentence Example 2

Action A01
is allowed in state normal ;
receives eventitem warning_interrupt ;
transmits eventitem safety_action1 ;
uses dataitem water_temperature_reading ;
produces dataitem safety_action_report ;
has maximum duration time 3 ;
has duration time units "seconds".

5.5.3.1. Relation_clauses

A relation_clause defines a relationship between the system or software concept instance being defined in the STL_sentence and one or more concept instances defined in other STL_sentences. Relation_clauses may describe relations such as abstraction, aggregation, connection, presentation, and restriction. The set of relation_clauses is unique for each pair of software concepts, so the relation_clauses are defined with the STL_sentences in Chapter 6. The BNF form of the relation_clause is as follows:

```
<relation_clause> ::=
        <relation_keywords>
        <relation_list>
```

The BNF sentence is read as follows: a relation_clause is defined as relation_keywords followed by a relation_list. In Figure 12 the example STL sentence is presented with relation_keywords underlined and relation_lists emboldened.

Figure 12. STL Sentence Example 3

Action A01
is allowed in state **normal** ;
receives eventitem **warning_interrupt** ;
transmits eventitem **safety_action1** ;
uses dataitem **water_temperature_reading** ;
produces dataitem **safety_action_report** ;
has maximum duration time 3 ;
has duration time units "seconds".

The last word in each set of relation_keywords is the name of a concept whose instances can appear in a relation_list. This word is called a classifier. The classifier allows type checking and forward referencing of sentence_identifiers.

The relation_list references (names) the concept instances that participate in the relationship. The BNF form of the relation_list is as follows:

```
<relation_list> ::= <relation_member> { , <relation_member>}
<relation_member> ::= <sentence_identifier> | TBD
```

6. STL Semantics: Subject and Concept Information

This Chapter is organized as follows: Sections 6.1 and 6.2 form an introductory overview; Section 6.3 is in the form of a reference handbook, with each STL concept presented in alphabetical order.

6.1. STL Concept Organization

STL concepts are logically organized according to subject, purpose, and type. This logical organization gives a classification hierarchy with three levels. The classification shows the commonality which exists among the concepts and the differences which distinguish between them.

The classification is exhaustive, and the distinctions make them mutually exclusive as well. Thus, each concept fills a well-defined information transfer need. The figure below illustrates the classification hierarchy.

Figure 14. Classification Hierarchy for Concepts

Category, Group, Concept	Subject, Purpose, Type of Information
Identification Category	**Subject: information transfer unit**
Identification Group	Purpose: identification of information
$_Packet	- information labeling and references
Interpretive Support Category	**Subject: elements of description**
Value Group	Purpose: value instance representations
Literals	- quantification of properties
Constant	- reference names for literals
Organization Group	Purpose: user focus
Collection	- groupings of software description elements
Object	- combine and encapsulate properties and transforms
Image Group	Purpose: visual presentations in drawings
GraphicSymbol	- references to named image instances
Software Behavior Category	**Subject: operations of software**
Data Group	Purpose: property representation
DataItem	- labeled instances of datatypes in transforms
DataKey	- identifier property representation
DataType	- property representation abstractions
DataPart	- property component structuring
DataRole	- rules for interaction instance
DataView	- entity subtype structure
Time Group	Purpose: marking of time
EventItem	- labeled appearances of recognized time
EventType	- time recognition abstractions
Logic Group	Purpose: decisions
CondExpressions	- property value instance discriminator
Condition	- named property value instance discriminators
Transform Group	Purpose: changes of property representation
Action	- observable changes of properties
DataStore	- retention of properties between actions
Coupling Group	Purpose: propagation of changes
ConnectionPath	- information propagator to transforms
Evolution Group	Purpose: sequencing of changes
State	- property characterization and transform access
StateTransition	- allowed changes of state

This classification is described in the remainder of this section. Appendix A contains entity-relationship diagrams for the STL which illustrate the interrelationships between the concepts. The set of diagrams is organized by subtypes and interrelationships.

6.4. STL Summary

6.4.1. Keywords

For context-independent parsing, the words in the attribute and relation clauses are reserved words in the STL. These reserved words are presented in the figure below, in alphabetical order. They are listed with the label "keyword" in the STL Profile (see Chapter 7).

Figure 18. STL Keywords

a	continuous	fixed	offers	statetransition
accepted	control	flow	only	statetransitiontype
accepts	creator	for	optional	statetype
access	criticality	foreign	order	std
accessed	data	format	outer	stored
action	dataitem	from	partitioned	stores
actiontype	datakey	goes	partitions	string
activate	datapart	graphicsymbol	pause	stringlong
acts	datarole	graphicsymboltype	permits	stringshort
allowed	datastore	grouped	pictured	structured
allows	datatype	groups	pictures	subject
alternate	datatypeclass	has	placeholder	subtype
an	dataview	identifier	played	subtypes
array	delay	if	plays	supertype
as	depletable	in	primary	supplied
asynchronous	description	initial	process	supplies
average	designunit	inner	produced	suspend
bandwidth	desirable	instance	produces	synchronous
basis	destructive	instances	provides	table
bidirectional	did	instantiates	purpose	through
boolean	disable	integer	range	time
by	disallowed	intermediate	ranges	timestamp
cardinality	discrete	internal	real	timing
carried	domain	into	received	to
carries	duration	involved	receives	transfer
caused	effect	involves	referenced	transform
causes	enable	is	references	transmits
character	encapsulated	label	relationship	transmitted
characteristic	encapsulates	mandatory	resume	trigger
characterized	entity	maximum	retention	true
characterizes	erd	membership	retrieval	unbounded
choice	established	merged	satisfied	units
coincident	establishes	merges	satisfies	unknown
collection	event	method	second	unspecified
component	eventitem	minimum	sequence	unstructured
compound	eventtype	nondepletable	sequential	usage
concurrent	eventtypeclass	nondestructive	shares	used
condition	exclusive	nonexclusive	size	uses
connected	exhaustive	nonexhaustive	specified	value
connectionpath	expression	null	specifies	values
connectiontype	external	object	split	valuetype
connects	false	occurrences	splits	version
consequent	final	of	standard	volume
constant	first	offered	state	with
content				

6.4.2. Attributes

Several attribute clauses are present in more than one concept sentence template. These clauses are gathered here into an alphabetized list of the unique occurrences.

Figure 19. STL Attribute Clauses

has action effect	has order
has average volume	has outer cardinality
has content timestamp	has placeholder value
has content version	has process method
has creator	has purpose
has criticality	has retention time units
has description	has retrieval effect
has duration time units	has size range
has expression	has standard usage
has external basis	has subject
has external usage	has subtype ranges
has external value	has time units
has fixed size	has transfer timing
has flow characteristic	has transform purpose
has format	has units
has identifier purpose	has value
has inner cardinality	has value false
has label	has value range
has maximum bandwidth	has value true
has maximum delay	has value unknown
has maximum duration time	has values
has maximum retention time	has volume units
has maximum volume	is actiontype
has membership domain	is connectiontype
has minimum bandwidth	is datatypeclass
has minimum delay	is eventtypeclass
has minimum duration time	is graphicsymboltype
has minimum retention time	is statetransitiontype
has minimum volume	is statetype
has null occurrences	is valuetype
has occurrences for	

6.4.3. Reciprocal relations

The binary relations between concept instances in an S_Packet can be expressed in either or both directions. Therefore, each relation clause has a reciprocal relation clause. These clause pairs are collected here and organized as follows.

- The unique pairs of concept names (CN1,CN2) are generated as those pairs of names for which CN2 is alphabetically greater than or equal to CN1.
- These unique pairs are alphabetized first by the first member of the pair.
- Pairs having the same first member are alphabetized by the second member of the pair.
- The reciprocal relation clauses for each unique pair are listed below the pair identification line.

There are two special cases. For conditions, the defining clause "is true if" takes a parenthesized, nested logic expression, represented below as "(CondExpression)." That expression may contain "component" conditions, but there is no explicit "has component" clause. The "is component of" clause is regarded here as the reciprocal of "is true if."

For graphicsymbols, there is a single "pictures" clause used to relate a graphicsymbol to any concept instance. This clause is represented below with a parenthetical entry for the intended sentence identifier. It is regarded here as the reciprocal of the "is pictured with" clause in the other concept templates

Figure 20. STL Reciprocal Relationships: Actions

action - action		
action	has component	action
action	is component of	action
action - collection		
action	is grouped into	collection
collection	groups	action
action - condition		
action	acts only if	condition
condition	permits	action
action	satisfies	condition
condition	is satisfied by	action
action - connectionpath		
action	is connected from	connectionpath
connectionpath	connects from	action
action	is connected to	connectionpath
connectionpath	connects to	action
action - dataitem		
action	uses	dataitem
dataitem	is used by	action
action	produces	dataitem
dataitem	is produced by	action
action - eventitem		
action	receives	eventitem
eventitem	is received by	action
action	transmits	eventitem
eventitem	is transmitted by	action
action - graphicsymbol		
action	is pictured with	graphicsymbol
graphicsymbol	pictures	(action)
action - object		
action	is encapsulated in	object
object	encapsulates	action
action	is offered by	object
object	offers	action
action - state		
action	is allowed in	state
state	allows	action
action - statetransition		
action	causes	statetransition
statetransition	is caused by	action

The following paragraphs describe the derivation of the STL Profile.

The analysis of the concept sentence templates to produce the STL Profile is illustrated with a simple example. Figure 21 contains a short STL sentence template example. This example is formed from the Action sentence template in Section 6.3.1 by ignoring all but one clause in each part of the template. A partial listing of STL Profile which corresponds to this example is displayed in Figure 22.

Figure 21. STL Sentence Template Example

```
concept_name:                Action
concept_meaning:             a transform internal or external to the subject software,
                             having inputs and outputs and changing states of action
                             or data in the subject software
concept_text_presentation:
    sentence_keyword:        Action
    sentence_identifier:     ActionID
    possible_attributes:
        has label                        label_string
        is actiontype                    internal |
                                         external
    possible_relations:
        uses dataitem                    DataItemID { , DataItemID }
    notes:
        1. The label_string is a stringValueShort.

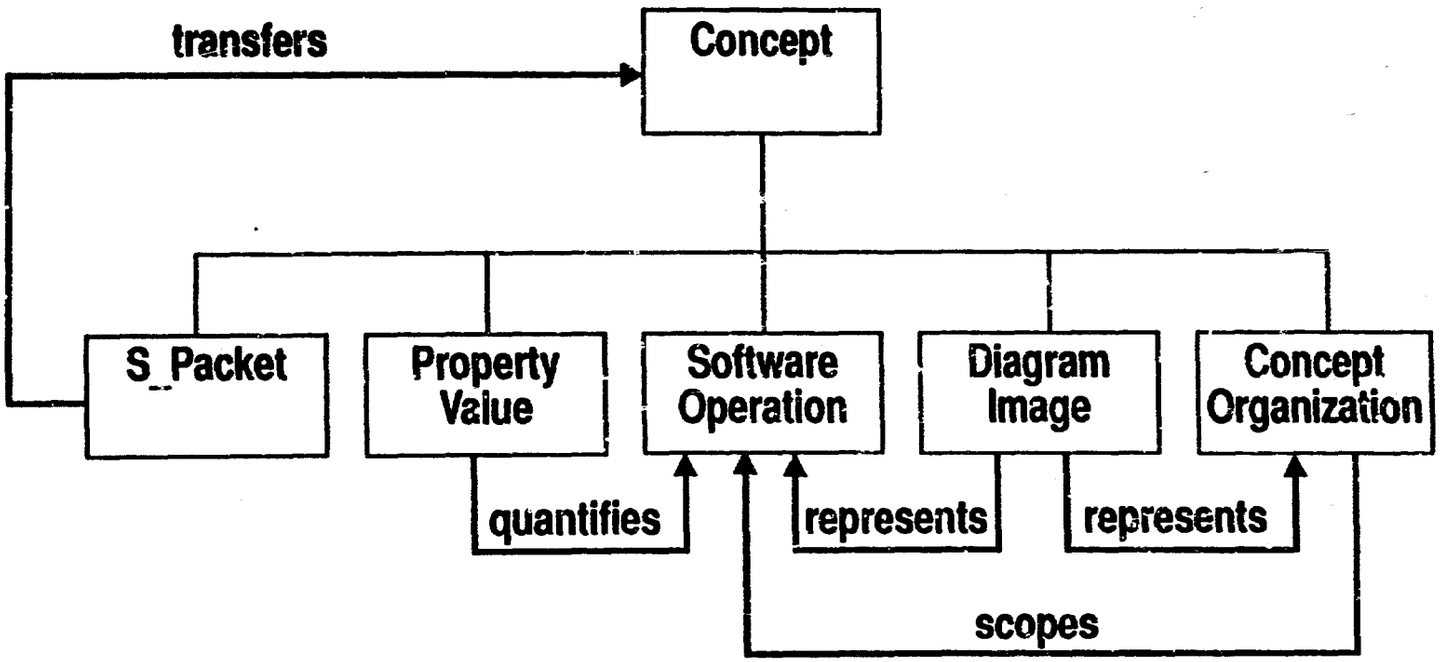
    additional_clauses:      for actiontype  internal
    possible_attributes:
        is actiontype                    internal
        has process method               method_description
    possible_relations:
        is allowed in state              StateID { , StateID }

    additional_clauses:      for actiontype  external
    possible_attributes:
        is actiontype                    external
    possible_relations:
        none
```

The initial steps in reducing the STL sentence templates to an STL profile profile are these.

- Identify all the terms needed to express the STL. Identify the symbols which are to represent sentence identifiers such as "ActionID", etc. Identify all keywords such as "action", "has", "label", "is", "actiontype", "internal", etc. Also, identify the mnemonics for the value types to appear in an instance of the sentence such as "label_string" for a "stringValueShort", etc.

Diagram A.2 STL Concepts



Note: Excluding the S_Packet, all entities on this diagram are abstractions of exclusive subsets of STL concept definitions. Similarly, excluding the "transfers" relation from S_Packet, the indicated relations are abstractions of actual relation clauses. Therefore, cardinalities are not included on this diagram.

Diagram A.3 Software Operation Concepts

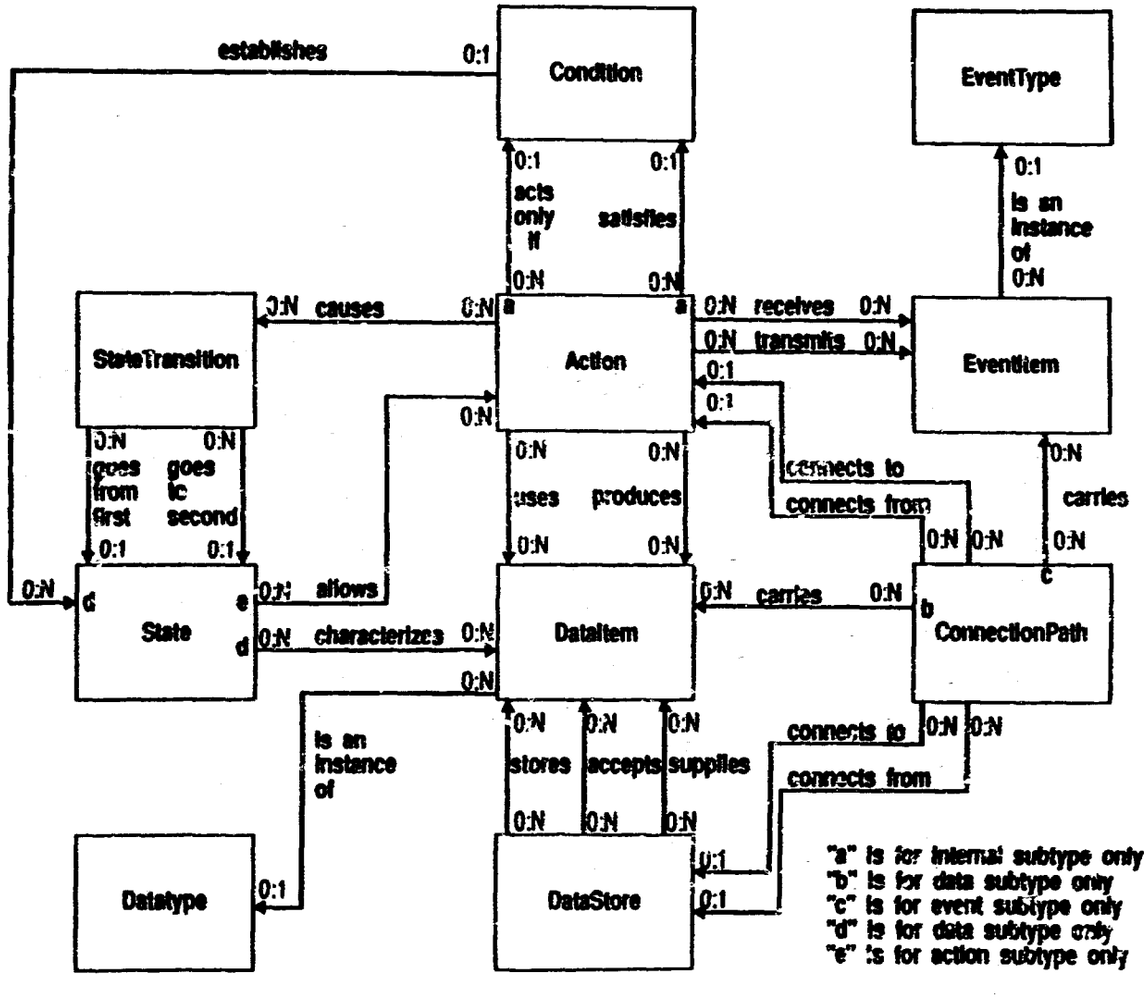


Diagram A.4 Data Concepts

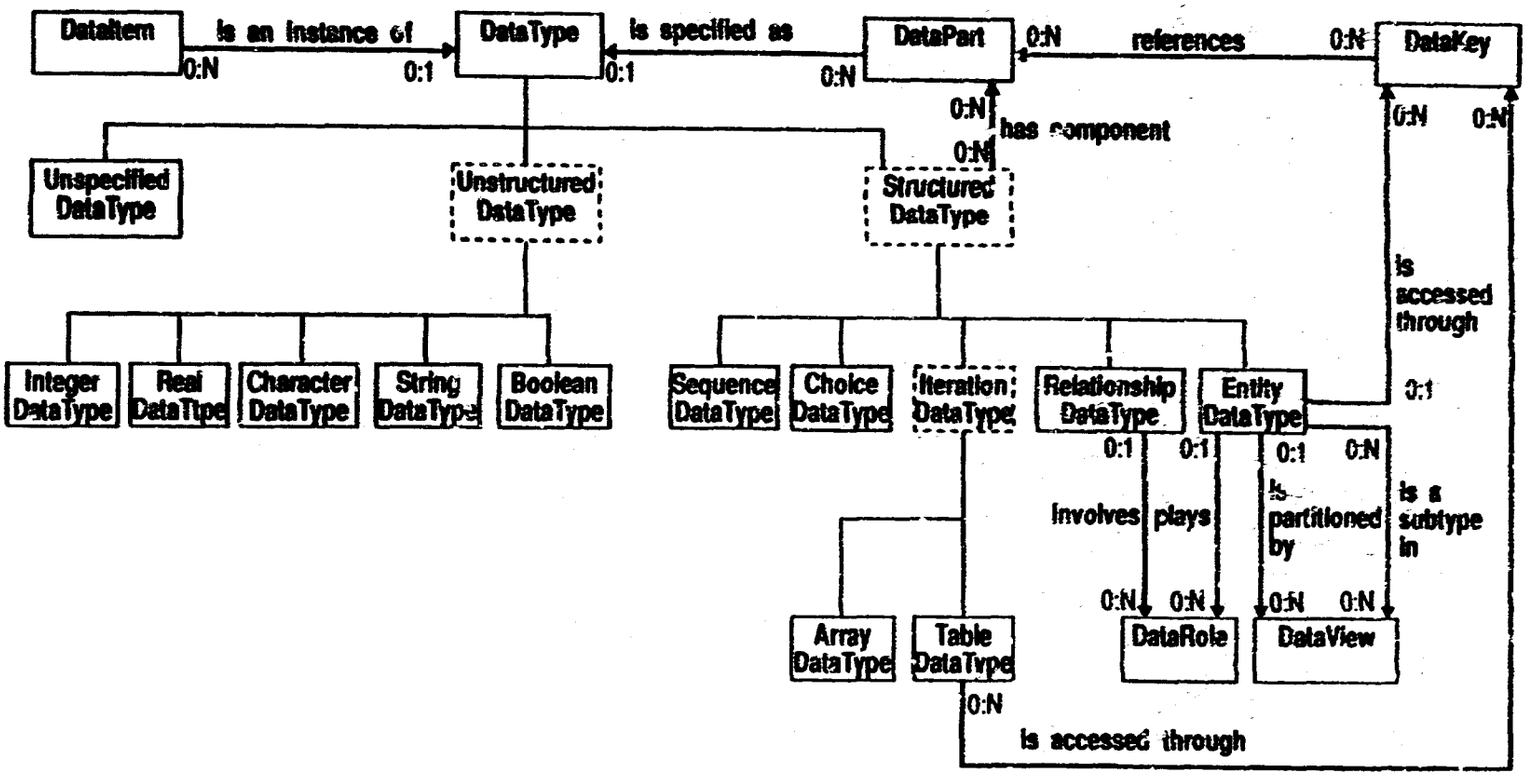


Diagram A.5 Event Concepts

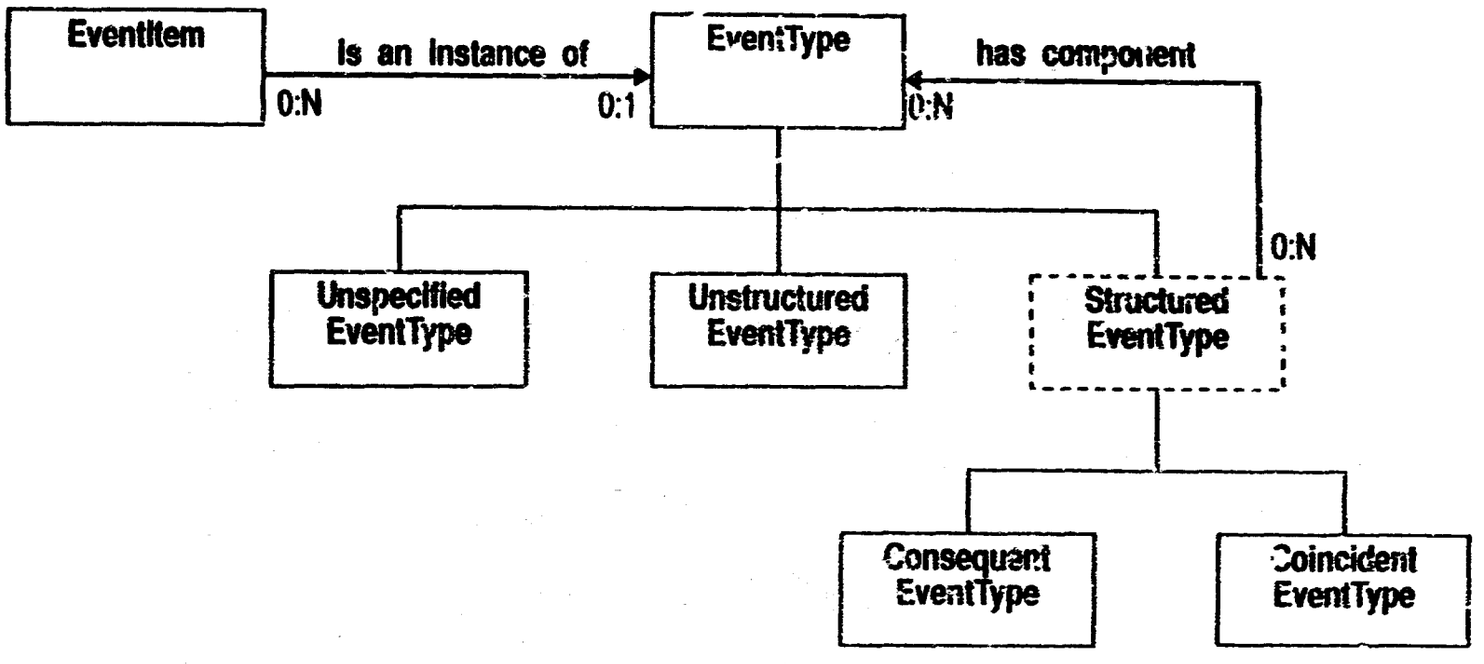


Diagram A.6 Action Concepts

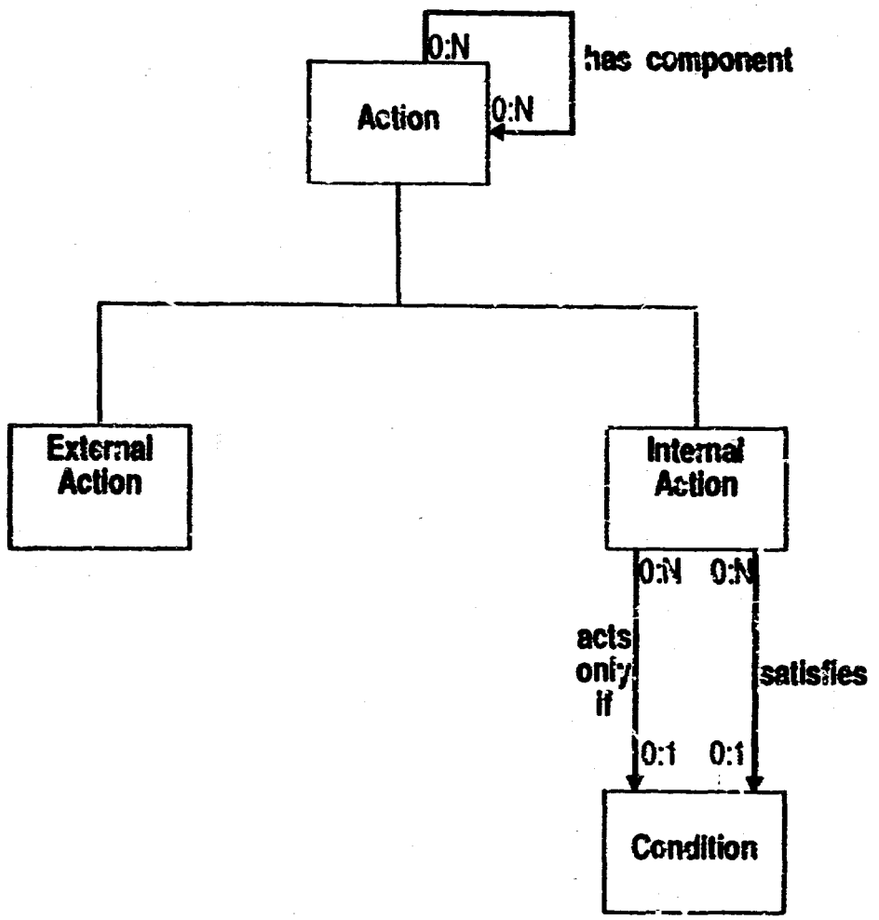


Diagram A.7 ConnectionPath Concepts

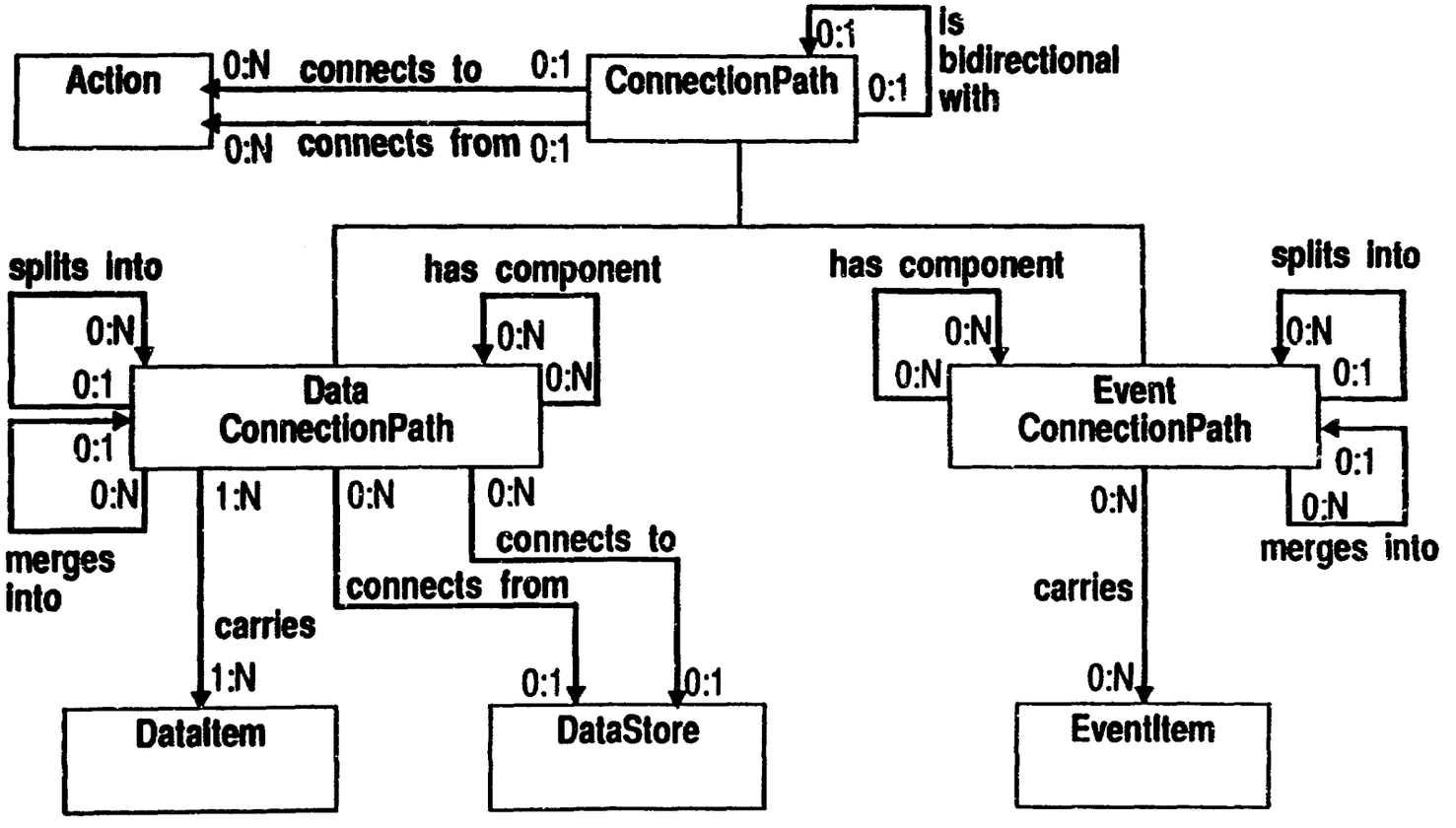


Diagram A.8 State Concepts

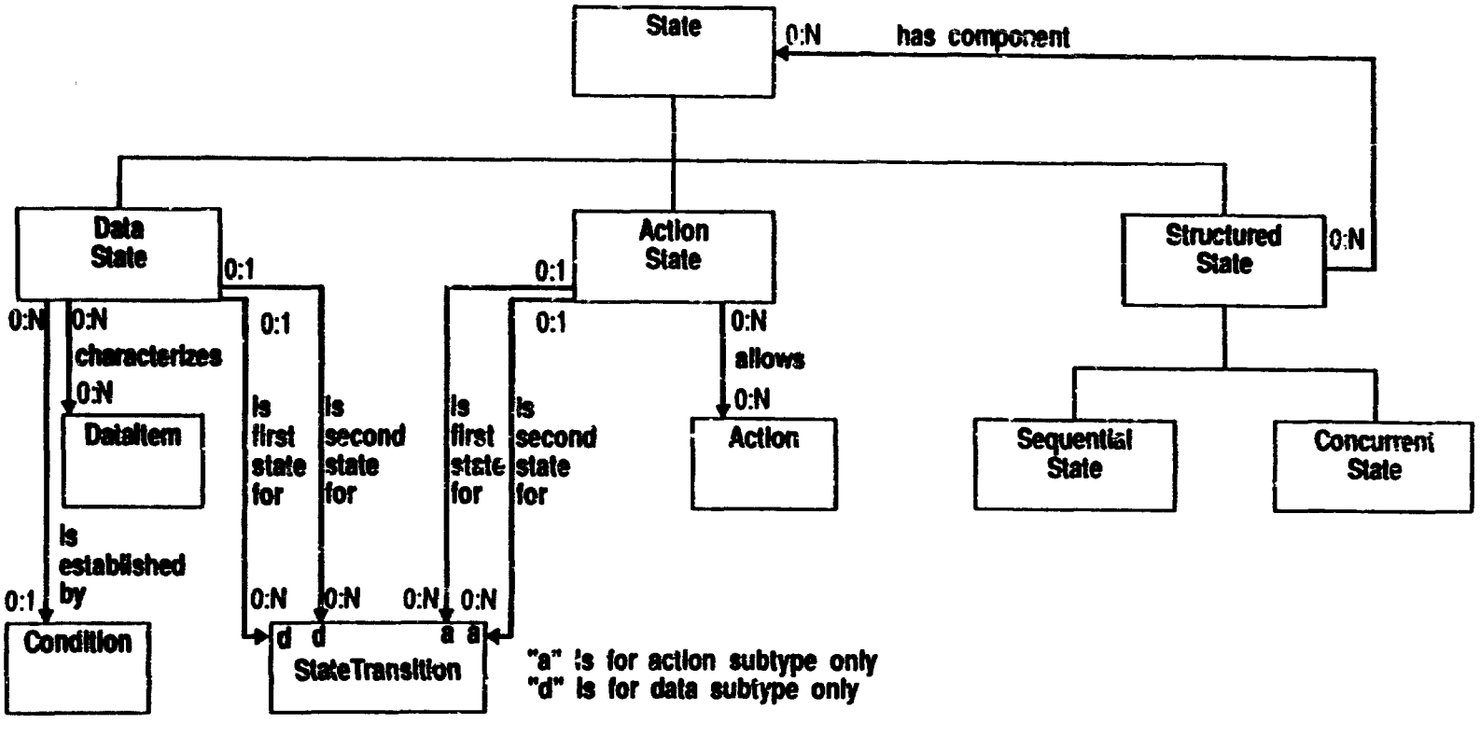


Diagram A.9 StateTransition Concepts

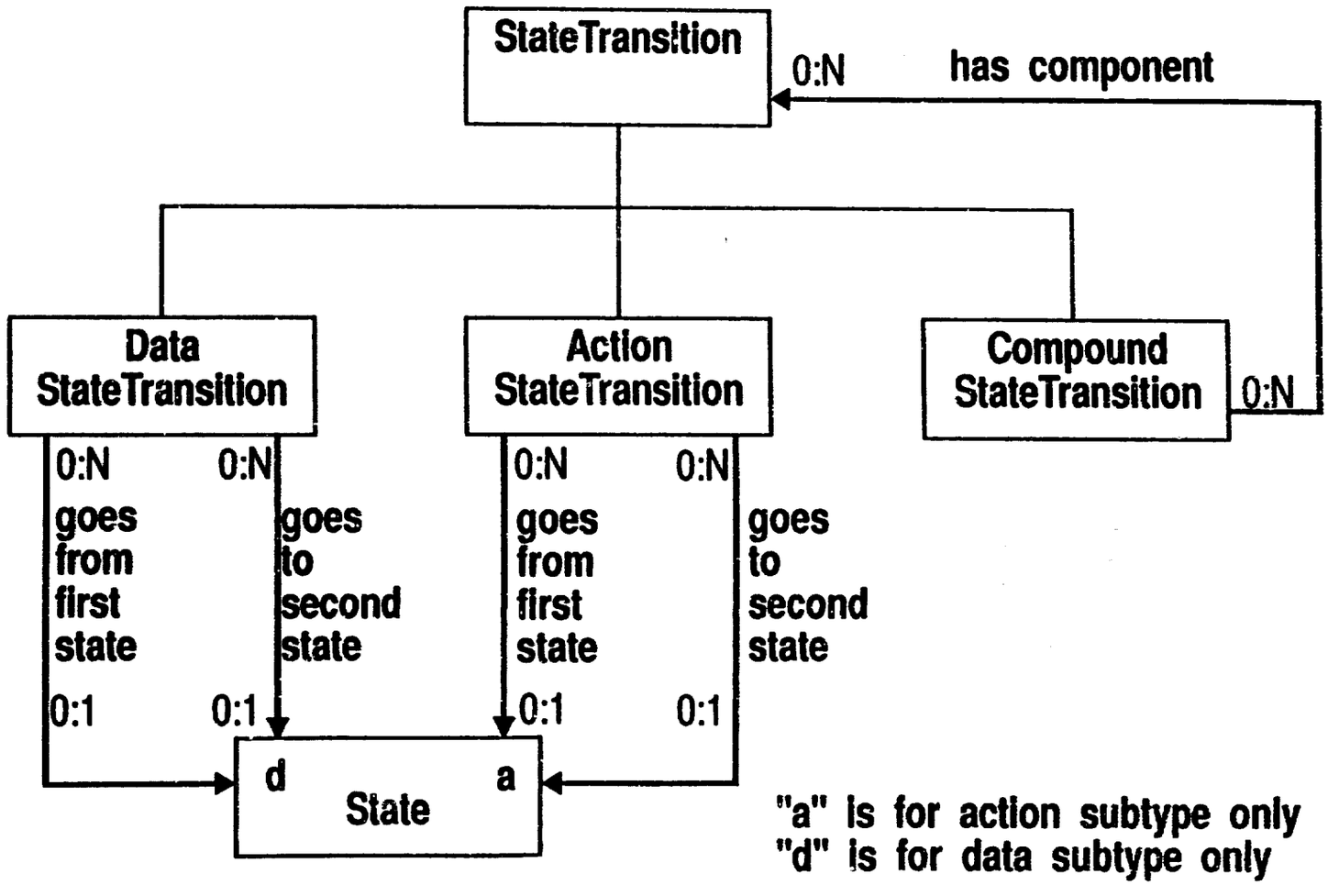
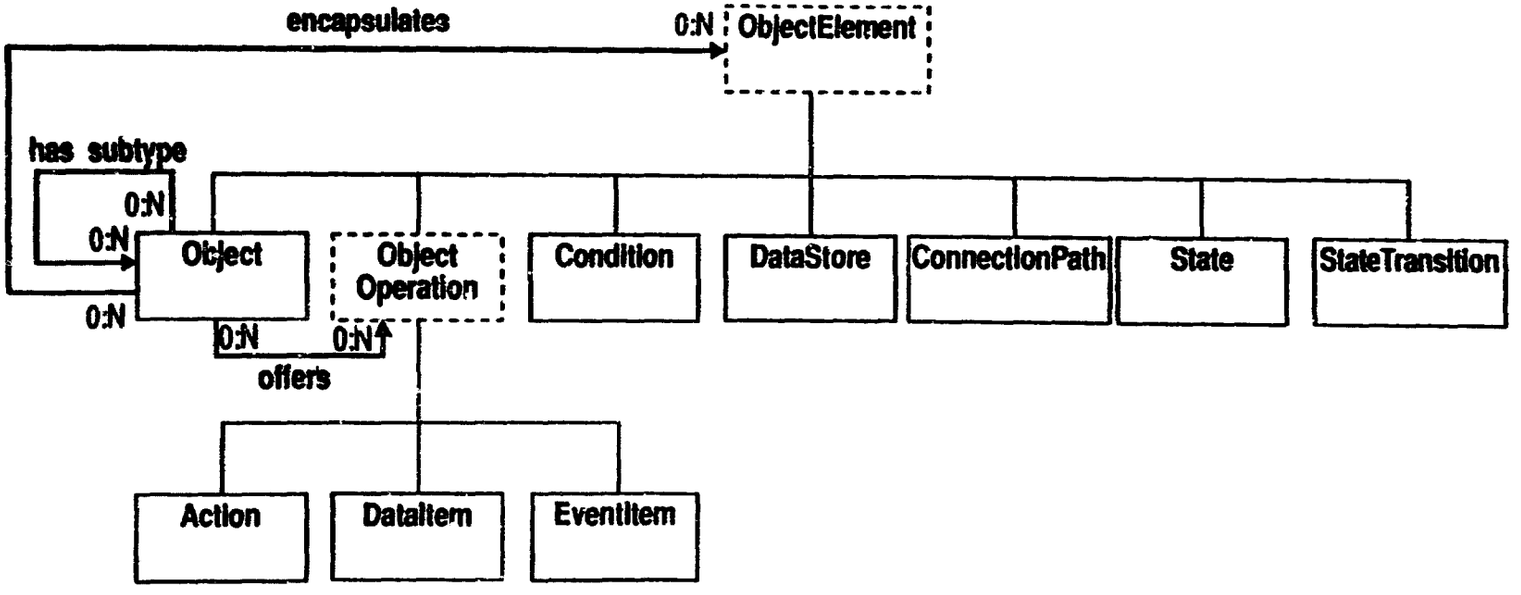


Diagram A.10 Object Concepts



B. Appendix B: STL Entity-Relationship Diagram Example

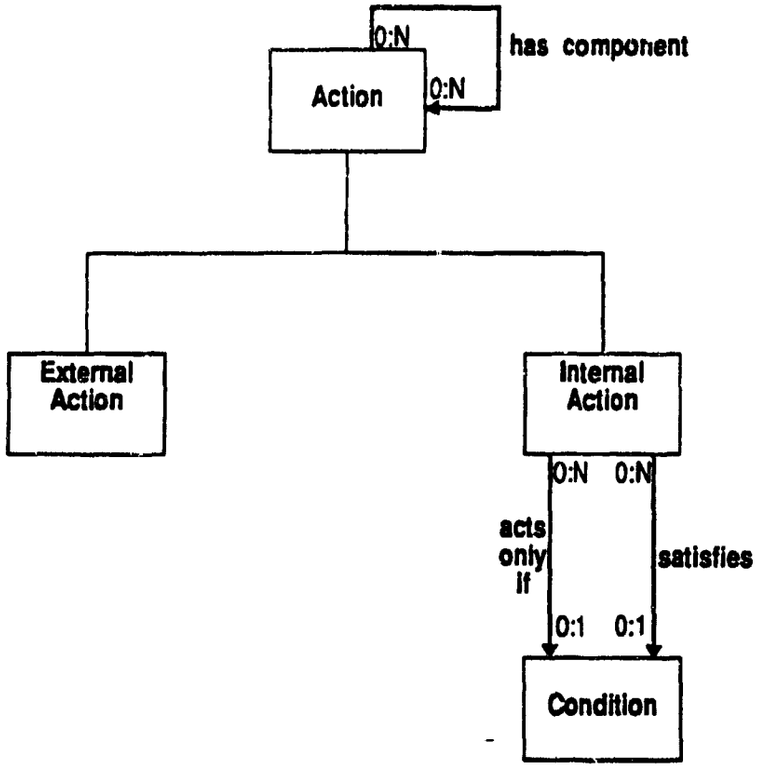
(This Appendix is not a part of IEEE Std 1175-19XX, IEEE Trial-Use Standard Reference Model for Computing System Tool Interconnections, but it included for information only.)

This appendix provides an example of the STL as applied to an entity-relationship diagram.

The subject is Diagram A.6, Action Concepts, in Appendix A. The figure is repeated here for convience.

The STL text here is actually more complete than the diagram: reciprocals for the relationships are included, and part of the entity definition for Action is given.

Diagram B.1 STL Entity-Relationship Example



CIS and ATIS

Eric Black: Atherton Technology

PRECIS OF PRESENTATION

ATIS is an IPSE interface specification which addresses those aspects of the IPSE which are critical to the tool integrator (or implementor, if the tool is designed from the start to be part of the IPSE). Integration of a tool with the IPSE (and with the other tools which are already integrated) involves three distinctly different yet overlapping aspects of integration:

- data integration: managing the tool's data and sharing data among tools
- control integration: invoking the correct tool at the correct time and in the correct way
- presentation integration: integrating the user interface of the tool with that of the IPSE and the other tools

ATIS focuses primarily on data integration, but treads into portions of control integration as well. It intentionally does not address presentation integration at all. It requires the services of an underlying data repository, which might be provided by an Object-Oriented, Entity-Relationship, Relational, or other suitable database, but does not at present specify the interface to that service. In terms of the ECMA Reference Model for Engineering Frameworks (August 17, 1990), ATIS addresses the Data Repository, Data Integration, Message, and Task Management Services.

The CASE Integration Services (CIS) committee is an industry consortium of hardware and software vendors and users chartered with developing or adopting a set of standard interfaces that promote the integration of software engineering tools. A number of environment interface specifications are being or will be evaluated by CIS, including ECMA PCTE, CAIS-A, IRDS, and ATIS. Related work being observed with interest includes that of OMG, OSF, CFI, CDIF, and others.

At the March 1991 ANSI SPARC meeting, CIS proposed the formation of a Technical Committee with essentially the same charter as CIS (that is, to adopt, promote, and/or develop a set of standard interfaces for software engineering environments). A vote is to be taken at the July SPARC meeting, and a positive answer is expected.

The continuing development and evolution of the ATIS specification is now "owned" by the CIS committee, which no doubt leads to further confusion over the difference between ATIS and CIS. The distinction is quite simple: ATIS is a specification, CIS is a committee.

Because there have been multiple versions of the ATIS specification published during its evolution to date, there is also confusion as to which ATIS is the "real" ATIS. The current version of the ATIS specification is the one which was adopted by CIS as its starting basis, and is titled the "CIS Base Document V1.0". It is available from the CIS committee secretary.

An earlier special version of ATIS was submitted to ANSI IRDS as a proposed standard, with numerous additions which are not present in the current specification (the Base Document). It was accepted by ANSI IRDS, yielding the document titled "ANSI X3H4 Working Draft: IRDS ATIS"; it was later rejected by ISO IRDS, and is no longer "active".

There is some overlap between ATIS and PCTE, but they are in no way competing or in any sense interchangeable specifications. PCTE is primarily interested in tool portability, and appears to have an operating system orientation. ATIS is specifically interested in tool integration services, and assumes that conforming environments will have solved the portability problem separately. Thus, ATIS should be complementary to PCTE, and should be able to use PCTE as a portable basis on which to provide tool integration services within a development environment.

For the most part, ATIS represents a higher level of services than does PCTE. However, it is not possible to layer ATIS on top of PCTE as the two specifications stand today. The primary reason for this is that the object orientation of ATIS, which results from efforts to best satisfy the requirements of Ease of Extensibility, Work Flow (Process) Control, Integrity Enforcement, Data Integration, Customizability, Reusability, and Security, cannot be implemented "on top of" PCTE if so doing allows direct access by tools to the PCTE interfaces without violating those requirements (particularly Integrity Enforcement and Ease of Extensibility). On the other hand, if ATIS is divided into two portions, a low-level object-oriented model and high-level IPSE model, and if the services provided by that low-level model are provided by PCTE as its evolution continues, then the fundamental conflict between ATIS and PCTE would seem to disappear, with only relatively minor specific details remaining in conflict. Those remaining conflicts should be reconcilable without major difficulty.

CIS and ATIS

presentation for:

2nd PCIS Workshop & IWCASE Update Mtg.
6 June 1991
Redondo Beach, CA

presented by:

Eric Black
Principal Architect
Atherton Technology
ericb@atherton.com

Copyright © 1991 Atherton Technology
30May91 edb

ATIS / CIS / Software BackPlane: Which is which?

- ATIS: A[therton] Tool Integration Services
  (an interface specification relating to those services
  of an IPSE pertaining to tool integration)

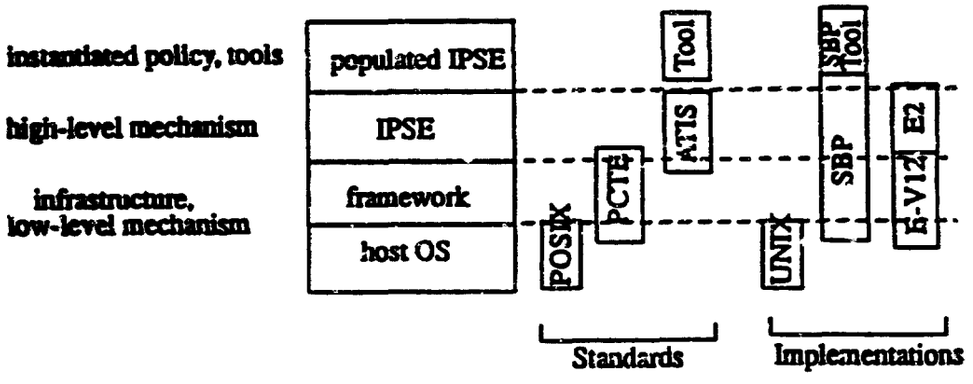
- CIS: CASE Integration Services committee
  (chartered to identify, select, develop, promote,
  and coordinate with other standards related to
  Integrated Project Support Environments

- Software BackPlane: a product, an implementation
  of an IPSE based on the ATIS interface specification

There's more to an IPSE than just tool portability and tool integration.

Copyright © 1991 Atherton Technology
30May91 edb

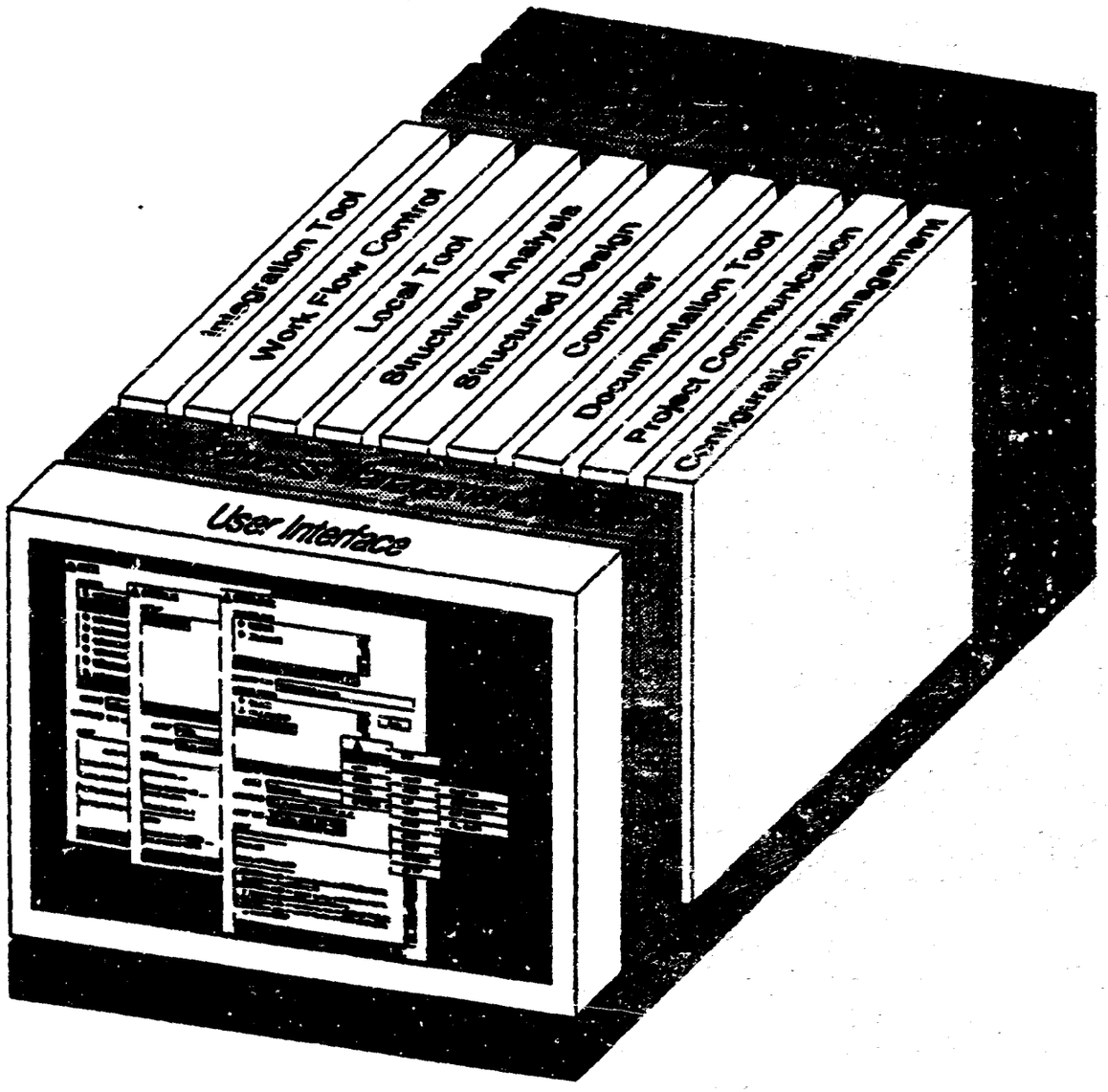
Areas of Overlap: framework, IPSE, Populated IPSE



Copyright © 1991 Atherton Technology
30May91 edb

ATHERTON

ATIS Mapping into the ECMA Reference Model



© 1991 Atherton Technology

3 Jun 91 / eG

Who are the users of an IPSE?

>> different points of view, different roles ⟹ different needs

- software designer/architect
- software developer
- documentation writer and editor
- project manager
- IPSE system administrator
- IPSE environment adapter / customizer
- tool integrator
- tool implementor/vendor
- software development process manager (SEI maven)
- maintenance manager
- configuration manager
- QA manager

Copyright © 1991 Atherton Technology
30May91 edb

IPSE Requirements

- Heterogeneous, Distributed, Persistent Repository
- Queryability
- Easy Extensibility
- Integrity Enforcement
- Multi-user Access
- Administrative Control
- Habit Preservation
- Availability

Copyright © 1991 Atherton Technology
30May91 edb

IPSE Requirements (cont'd)

- Integration:
  - data integration
  - control integration
  - presentation integration
- Reproducibility
- Traceability
- Customizability
- Full life-cycle support
- Evolution
- Reusability

Copyright © 1991 Atherton Technology
30May91 edb

ATIS Semantic Models

- Basic object-oriented data model: objects, messages
- Versioning model
- Configuration model
- Work flow control model
- Tool integration / client registration model
- Transaction model
- Security model
- Naming services model

Copyright © 1991 Atherton Technology
30May91 edb

Class-Driven Method Dispatch

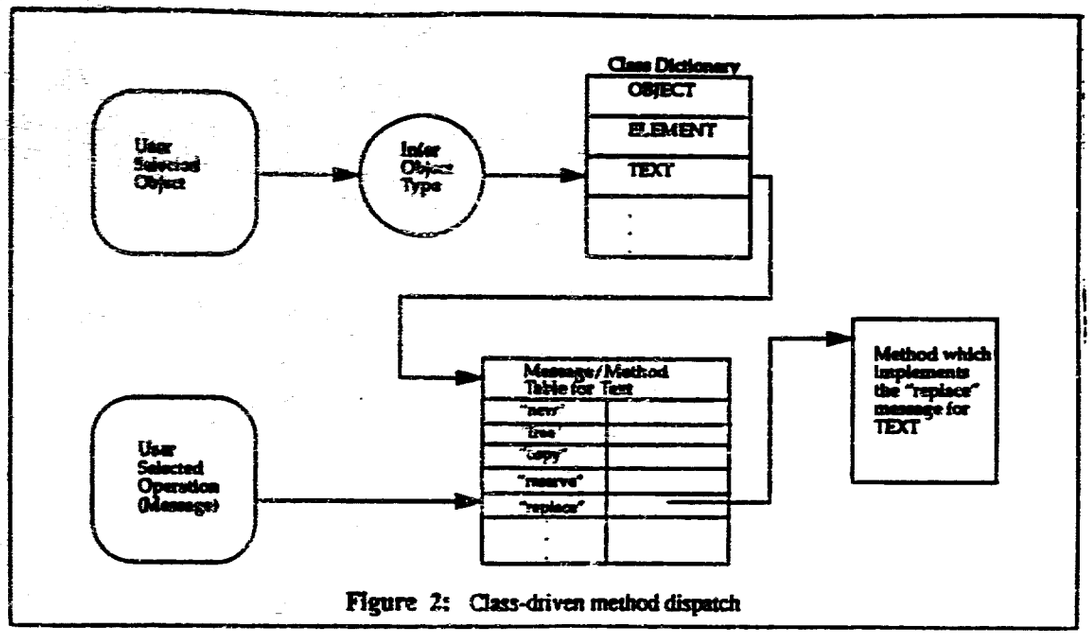


Figure 2: Class-driven method dispatch

Copyright © 1991 Atherton Technology
30May91 edh

Typical Method Structure

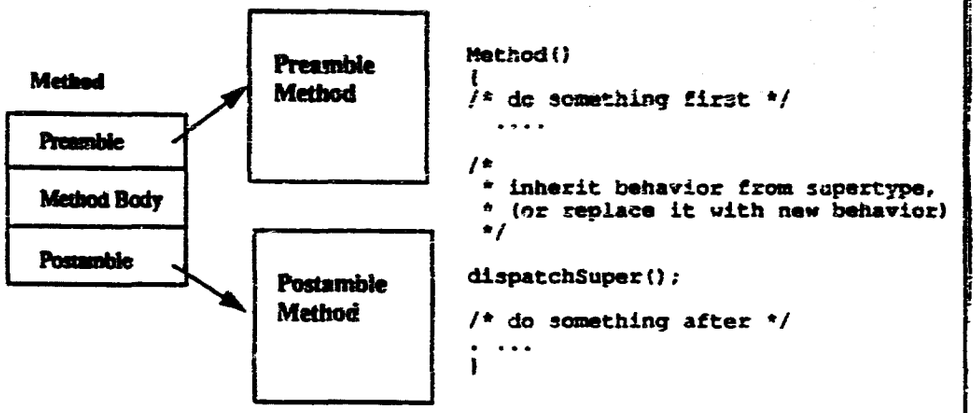


Figure 4: typical method structure

Copyright © 1991 Atherton Technology
30May91 edb

Partial Class Hierarchy

- ELEMENT
  - NAMED-ELEMENT
  - MESSAGE
  - CONTEXT
  - VERSION
    - TYPE
      - DATA-TYPE
      - ELEMENT-TYPE
    - AGGREGATE
      - BINARY
        - TEXT
    - TOOL
      - METHOD

Copyright © 1991 Atherton Technology
30May91 edb

ATIS and PCTE

- Most of ATIS is "higher level" than most of PCTE
  - ATIS is specifically directed toward tool integration into an environment (and other tools)
  - PCTE is a basis for an environment providing portability -- is not an environment itself
- There currently are areas of overlap:
  some conflict, mostly complementary
- Biggest difference is the lack of encapsulation of behavior (methods) in PCTE, which ATIS includes due to requirements of:
  ease of extensibility, work flow (process) control, integrity enforcement, data integration, customizability, reusability, and security

Copyright © 1991 Atherton Technology
30May91 edb

ATIS History

- May 1987: shipment of *Software BackPlane* Release 1.0 from Atherton, based on Entity-Relationship model
- October 1987: shipment of *Software BackPlane*Release 2.0 from Atherton, based on Object-Oriented model
- late 1987: Atherton proposes joint effort with Digital to standardize IPSE tool integration services
- January 1988: Atherton and Digital begin work defining the models and interfaces for CASE tool integration services, using *Software BackPlane* as a starting point; specification is called "Atherton Tool Integration Services"

Copyright © 1991 Atherton Technology
30May91 edh

ATIS History (cont'd)

- May 1988: Atherton and Digital host ATIS kick-off meeting in Dallas with workstation vendors, CASE tool vendors, CASE tool users in attendance; ATIS document titled "ATIS Phase 1" is distributed, and an informal consortium of hardware vendors, environment vendors, tool vendors, and end-users is formed

- July 1988: discussions and feedback are incorporated into ATIS document, yielding the new version "ATIS Phase 2"

- 1988-1989: consortium continues work refining the ATIS specification; the name is changed to "A Tool Integration Service"; the name "Common Application and Tool Integration Services" is used briefly but dropped

Copyright © 1991 Atherton Technology
30May91 edb

ATIS History (cont'd)

- Februrary 1990: independent CASE Integration Services (CIS) committee formed to take over & continue evolution of ATIS as a standard, and to identify, select, develop, promote, and coordinate with other standards related to IPSEs

- March 1990: special augmented version of ATIS titled "IRDS ATIS" submitted to ANSI IRDS; accepted as a proposed standard titled "ANSI X3H4 Working Draft: IRDS ATIS"; later rejected by ISO IRDS

Copyright © 1991 Atherton Technology
30May91 edb

ATIS History (cont'd)

- November 1990: CIS adopts a baseline version of the ATIS specification, titled "CIS Base Document V1.0", putting ATIS under formal configuration control; to achieve consensus, certain items under contention (such as security, transaction, and distribution models) are temporarily removed from the document (but identified as required in the finished specification)
- March 1991: CIS committee proposes formation of ANSI SPARC Technical Committee to carry out CIS's charter as a formal IPSE interface standardization effort
- July 1991: ANSI SPARC decision expected

Copyright © 1991 Atherton Technology
30May91 edb

Which ATIS is the *Real* ATIS?

- ATIS Phase 1 - *no*
- ATIS Phase 2 - *no*
- ANSI X3H4 Working Draft: IRDS ATIS - *no*
- ☛ **CIS Base Document V1.0** -*yes*

Copyright © 1991 Atherton Technology
30May91 edb

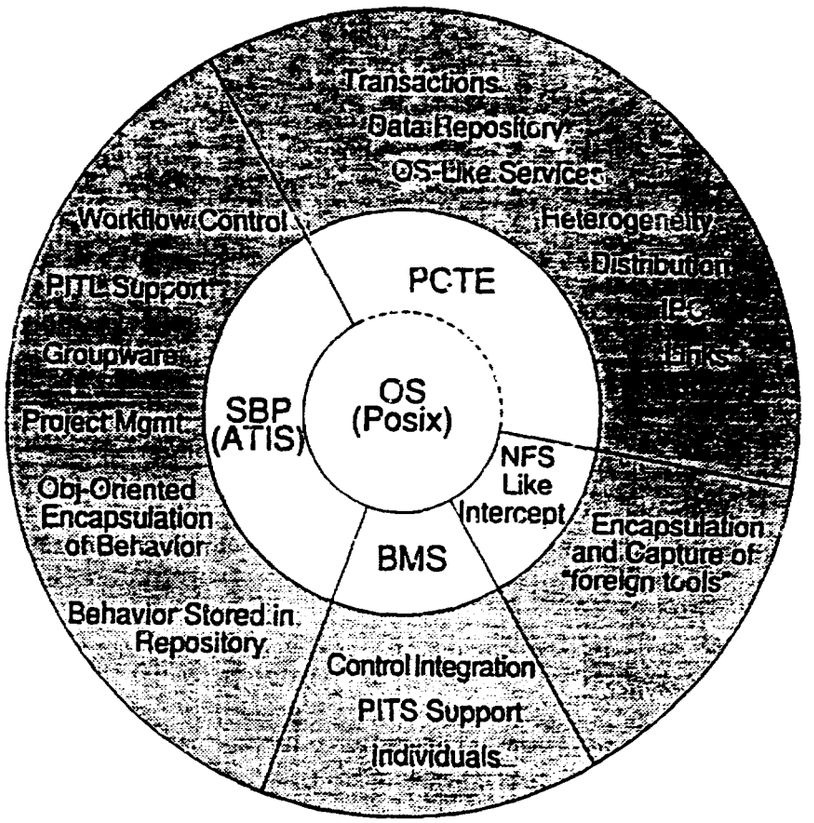
Why is ATIS Object-Oriented?

- provide easy incremental extensibility
  - ☛ easily add new types& operations by extending existing ones
- manage complexity
  - ☛ 450 E-R interfaces -> 12 O-O interfaces
- enforce integrity constraints
  - ☛ mutual constraints between objects not easily expressed except with code; where else to put the code?
- easier to provide triggers
  - ☛ "stack" preconditions, postconditions & code
- fine-grained access control permissions
  - ☛ besides RWED, have Version, Branch, etc.

Copyright © 1991 Atherton Technology
30May91 edb

Environment with both PITS and PITL support

Users



This is not an architecture!

Copyright © 1991 Atherton Technology
30May91 edb

PATRICIA ("TRICIA") OBERNDORF

PSESWG CO - CHAIRPERSON

NAVAL AIR DEVELOPMENT CENTER
CODE 7031
WARMINSTER, PA 10974-5000

(215)441-2737

tricia@nadc.navy.mil

NGCR PROGRAM PURPOSE

- PROVIDE COMPUTER RESOURCE STANDARDS CAPABLE OF MEETING NAVY MISSION CRITICAL REQUIREMENTS IN THE MID 1990'S AND BEYOND
  - HARDWARE AND SOFTWARE
  - SUCCESSOR TO CURRENT NAVY STANDARDS
  - COST EFFECTIVE

NGCR CONCEPT

NGCR IS NOT

A COMPUTER SYSTEM

A SOFTWARE SYSTEM

NGCR IS

HARDWARE AND SOFTWARE
INTERFACE AND PROTOCOL STANDARDS

STANDARDIZATION AREAS

MULTIPROCESSOR INTERCONNECTS

- BACKPLANE (1992)*
- HIGH SPEED DATA TRANSFER NETWORK (1994)
- HIGH PERFORMANCE BACKPLANE (1997)

MULTISYSTEM INTERCONNECTS

- SAFENET I / LOCAL AREA NETWORK (1990)*
- SAFENET II / LOCAL AREA NETWORK (1991)*
- HIGH PERFORMANCE LOCAL AREA NETWORK (1998)

SOFTWARE STANDARDIZATION AREAS

- OPERATING SYSTEM (1996*; 1998 MLS)
- DATA BASE MANAGEMENT SYSTEM (1998)
- PROGRAMMING SUPPORT ENVIRONMENT (1998)
- GRAPHICS LANGUAGE / INTERFACE (1998)

APPROACH

ALTERNATIVES FOR ACHIEVING A STANDARD

ORDER OF PRECEDENCE:

I SELECT EXISTING "STANDARD" / GROUP OF "STANDARDS"

- -MEETS ALL REQUIREMENTS

II SELECT "DEVELOPING" STANDARD

- -MEETS MOST, NOT ALL REQUIREMENTS
- -PARTICIPATE IN DEFINITION

III MOTIVATE STANDARDIZATION, "AD HOC" WIDELY USED COMMERCIAL PRODUCT

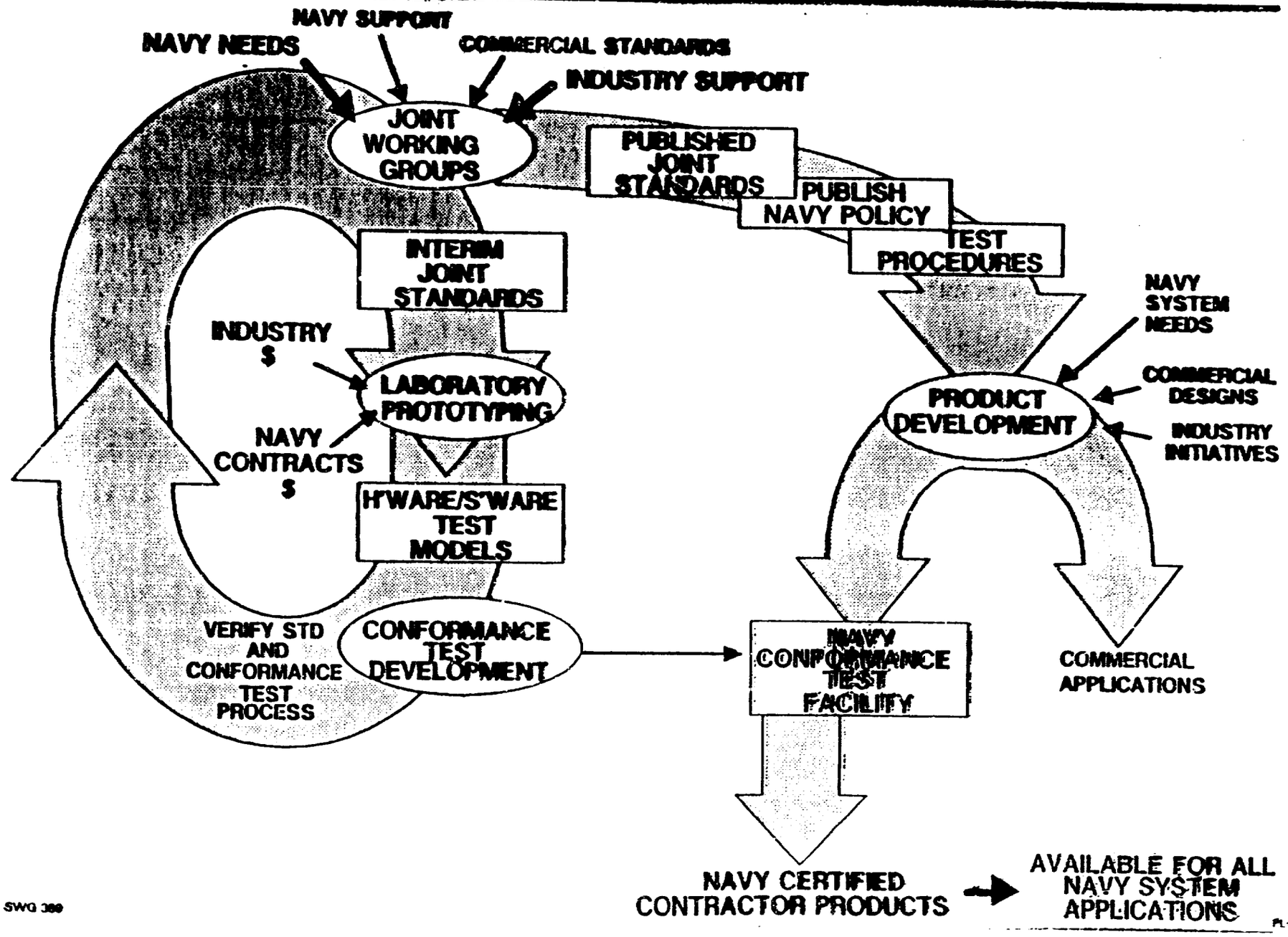
- -MEETS SOME REQUIREMENTS
- -PARTICIPATE IN DEFINITION

IV DEVELOP OWN STANDARD

NGCR APPROACH
OPEN SYSTEMS ENGINEERING ARCHITECTURE

- PROVIDES FRAMEWORK FOR SYSTEMS DESIGN
  - DOES NOT DEFINE OR STANDARDIZE ON A COMPUTER DESIGN
  - STANDARDIZES HARDWARE / SOFTWARE INTERFACES
  - PROVIDES FRAMEWORK FOR INDUSTRY IR&D INVESTMENT
- FOLLOWS STANDARDIZATION TRENDS IN COMMERCIAL SECTOR
- IMPLEMENTED THROUGH JOINT NAVY / INDUSTRY WORKING GROUPS
  - WIDELY USED NON-PROPRIETARY COMMERCIAL STANDARDS BASE

PROGRAM FLOW



SUMMARY

KEY FACTORS
- COMPETITION
- INTEROPERABILITY
- COMMONALITY
- LOGISTICS
- SUPPORTABILITY
- TECHNOLOGY
- COMMERCIAL BASE
- MODULAR DESIGNS

- REVOLUTION IN NAVY COMPUTER RESOURCES ACQUISITION STRATEGY OF PAST 30 YEARS TO REFLECT DOD MARKET SHARE AND LONG DEVELOPMENT CYCLE
  - NON-PROPRIETARY OPEN SYSTEMS ARCHITECTURE
  - CFE
- EMPHASIS ON COMPETITION, INTEROPERABILITY AND COMMONALITY
- RAPID, CONTINUOUS TRANSITION OF U.S. INDUSTRY STATE OF THE PRACTICE TO NAVY SYSTEMS
  - COMMERCIAL DESIGNS AND WIDELY ACCEPTED NON-PROPRIETARY COMMERCIAL STANDARDS
  - INDUSTRY DESIGN CREATIVITY AND TECHNOLOGIES

OBJECTIVE

To select **a set** of industry-based PSE interface standards

for use in the support of the development of Navy systems

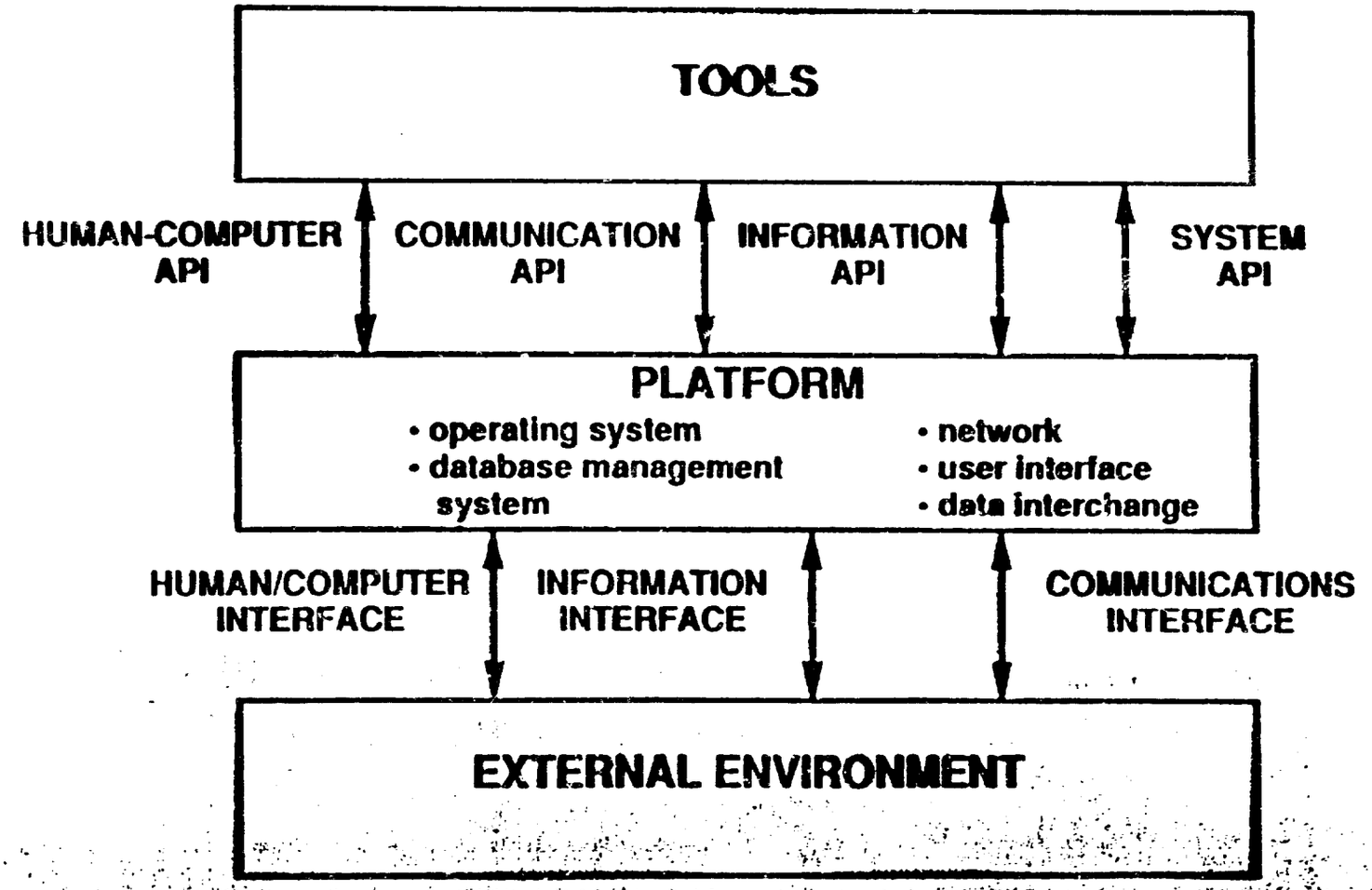
- joint industry/Navy/other working group
- subgroups as needed
- benefit from/coordinate with as many other related projects as possible (e.g., NIST, STARS)

HIGH-LEVEL REQUIREMENTS

- support the development and maintenance of the full range of Navy mission critical computer resources, particularly application software

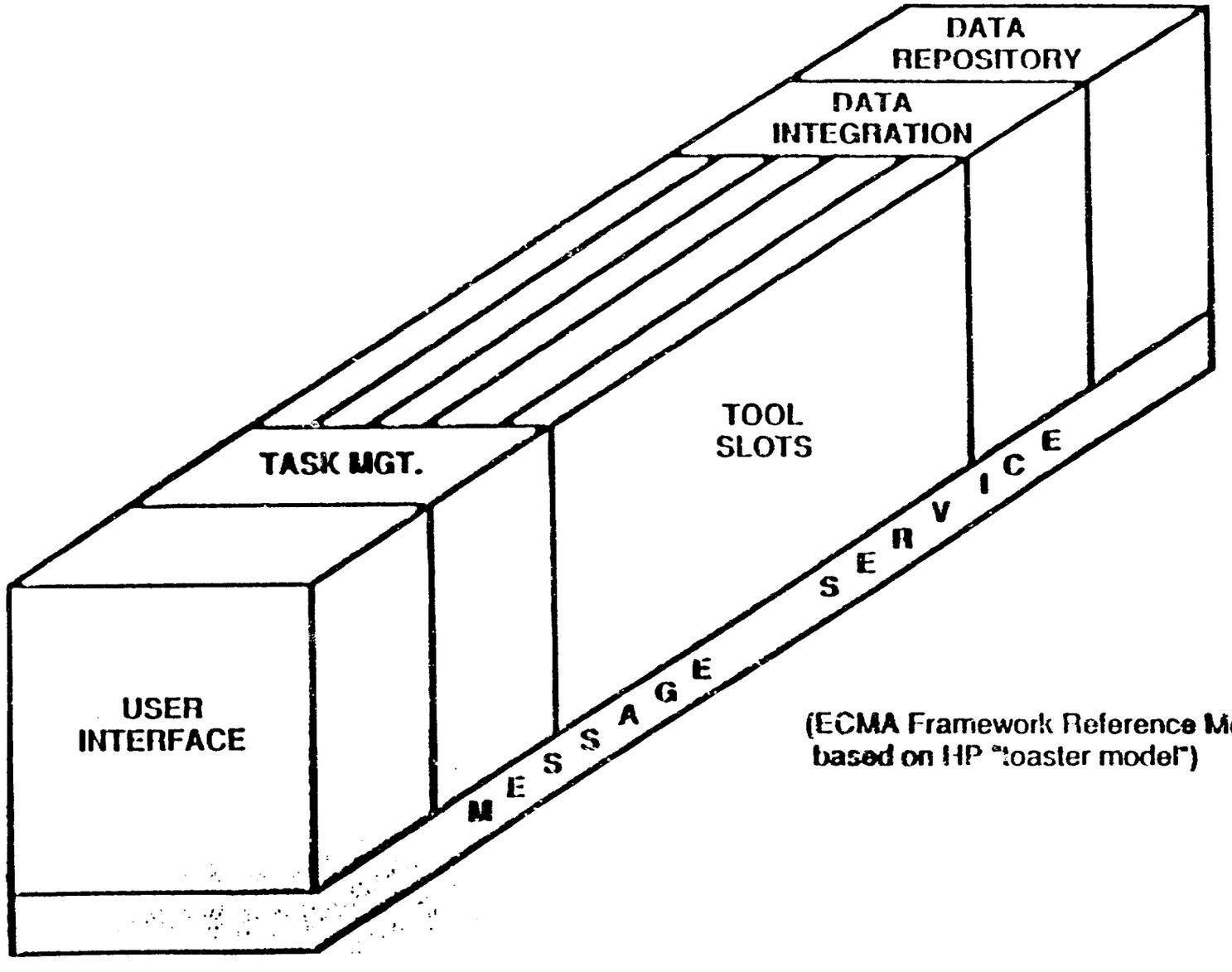
- facilitate the incorporation and use of COTS and NDI tools and products in the construction of Navy project PSEs

- facilitate the cost-effective introduction of new technology into Navy project PSEs

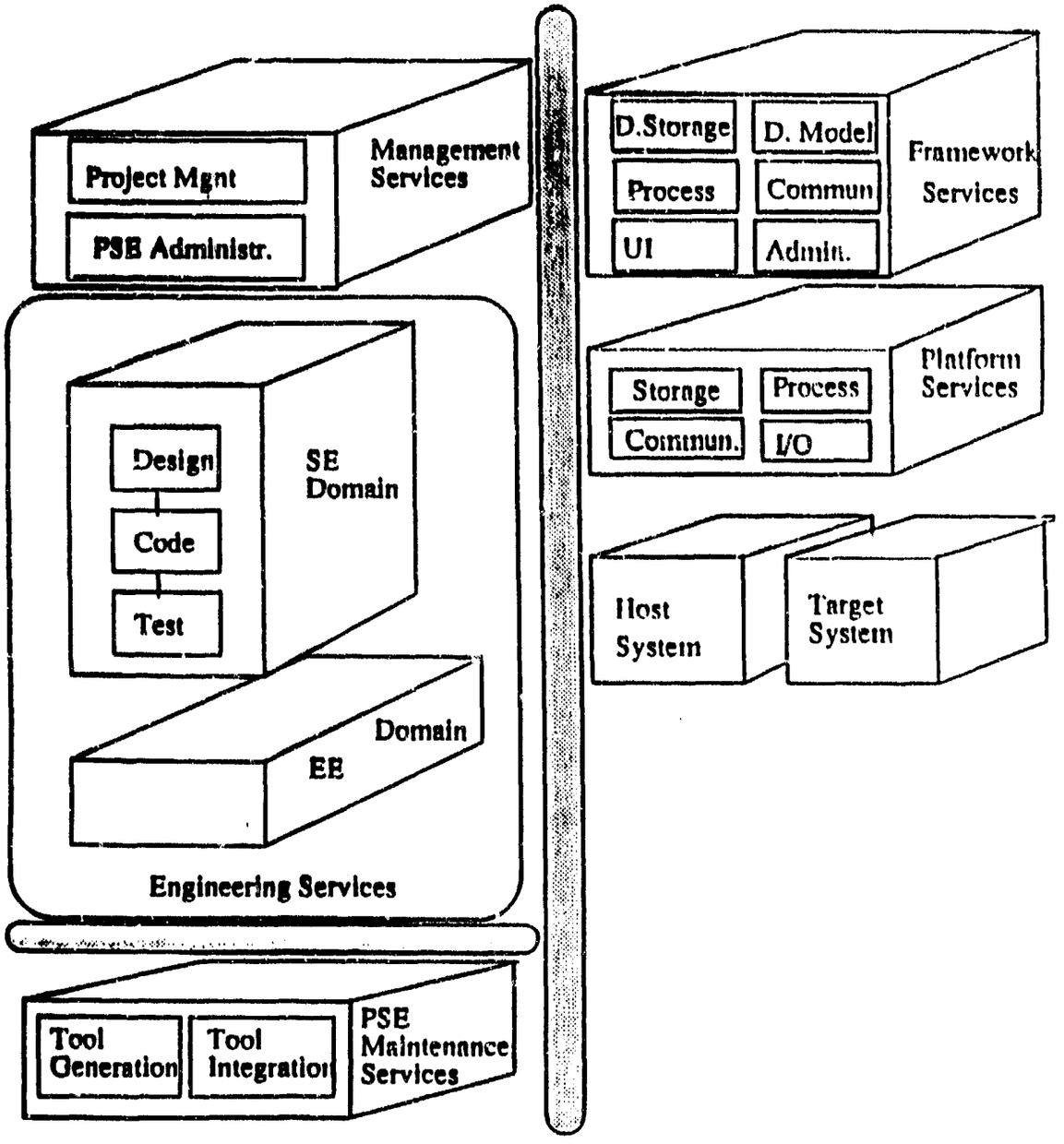


(adapted from IEEE P1003.0 Guide Reference Model)

lab01-3



(ECMA Framework Reference Model, based on HP "toaster model")


Project Mgnt
PSE Administr.
Management Services
D.Storage
D. Model
Process
Commun
UI
Admin.
Framework Services
Storage
Process
Commun.
I/O
Platform Services
Design
Code
Test
SE Domain
Domain
EE
Engineering Services
Host System
Target System
Tool Generation
Tool Integration
PSE Maintenance Services

PLANNED ACTIVITIES

- Planning meeting: 29-30 January 1991
  Baltimore
- First industry meeting: 26 - 28 February 1991
  NSWC-WO
- Quarterly meetings (Feb/May/Aug/Nov)
- Coordination with many other groups
- May meeting focus on STARS decisions
- Incremental "installments" on standard
- Completion ~1998

PSESWG

Milestones and Deliverables

Plan

Products		Start	Deliver
(AP) Plan of Action and Milestones		now	FEB 91
(AP) Incremental Standard Development Approaches White Paper	Draft 0.1	now	FEB 91
	Draft 1.0	FEB 91	NOV 91
(AT) PSE Technology Report	Draft 0.1	now	MAY 91
	Draft 0.2	MAY 91	AUG 91
	Draft 1.0	AUG 91	NOV 91
(RM) PSEI Reference Model	Draft 0.5	FEB 91	FEB 92
	Draft 1.0	FEB 92	NOV 92
(RM) Scope of Standardization Report			
	Draft 0.1	MAY 91	FEB 92
	Draft 1.0	FEB 92	NOV 92
(AT) Conference on PSEI Standards		now	AUG 92
(AP) Calendar of Related Events		MAY 91	Continuing
(AP) Annual Report			Annually in November
(All) Comprehensive Standardization Plan		NOV 91	JUL 93
(All) Published Military Standard Available for Use		JUL 93	SEP 98

‡

The COHESION Framework Program

Ed Cuoco
COHESION Framework Product Manager

Slide No. 1

Types of Products



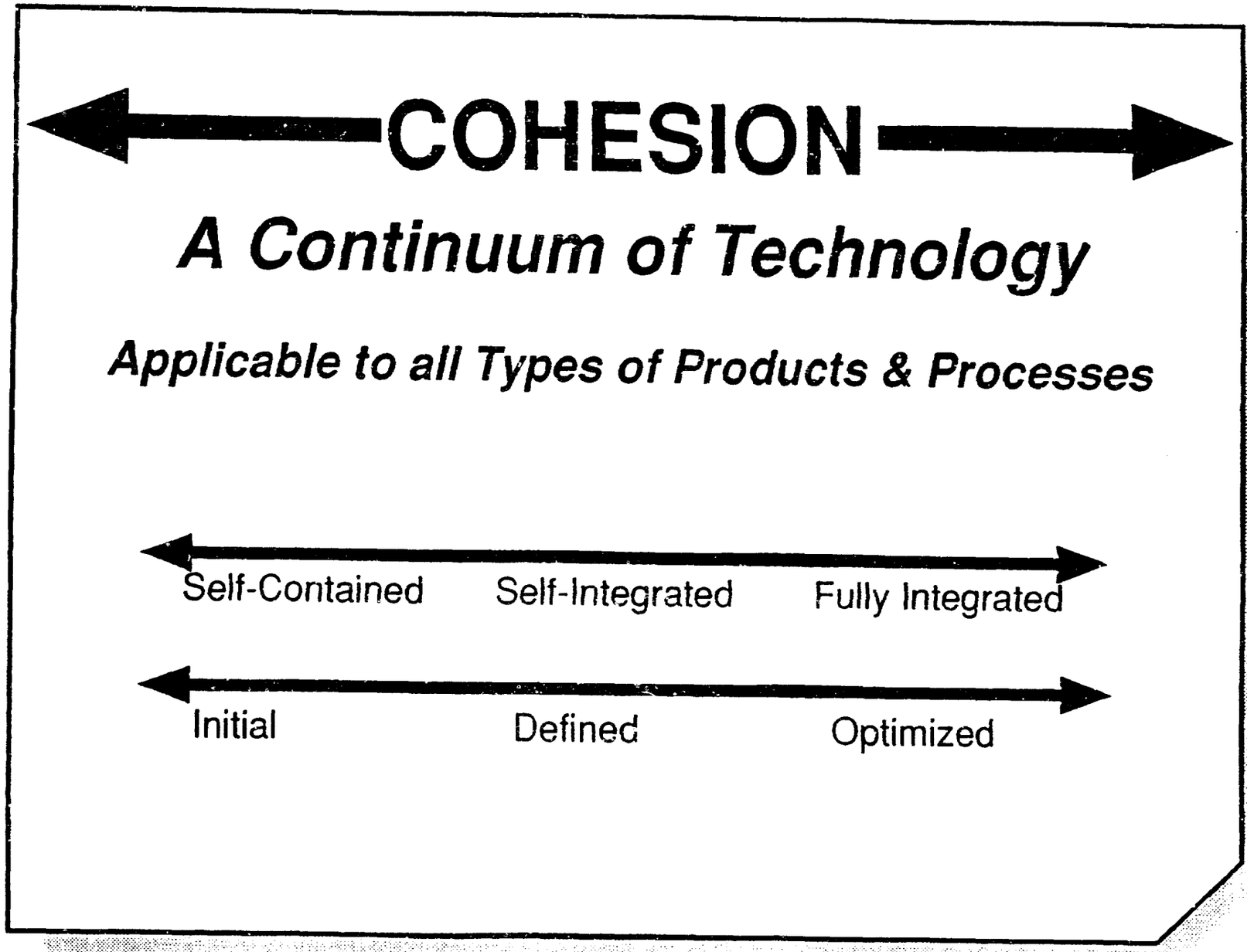
Self-Contained	Self-Integrated	Fully Integrated
- Spread sheet - Laboratory prototype	- Modeling tool - Accounts payable - Canned reports	- Enterprise-wide data integration in applications - Data warehouse - CASE environment

Range of Process



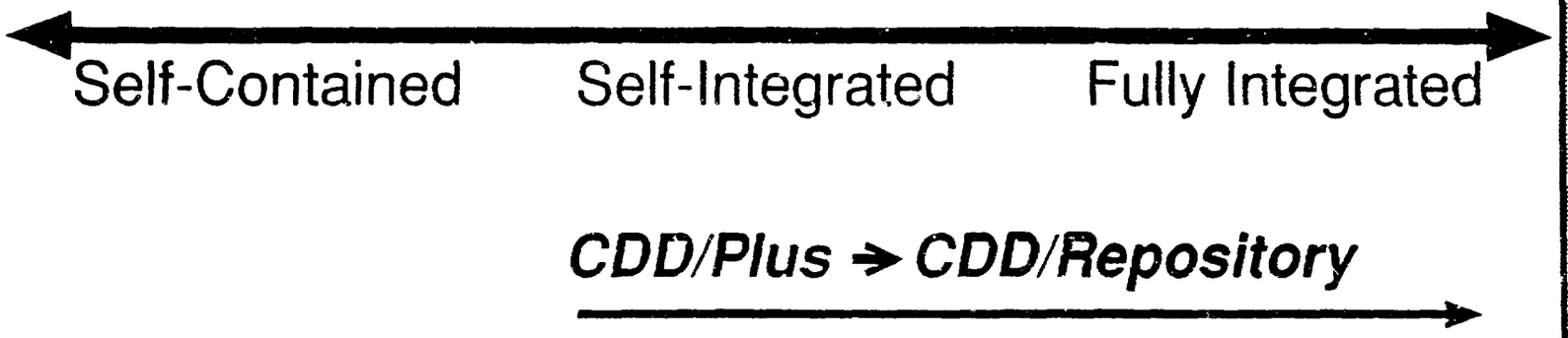
Initial*	Defined*	Optimizing*
Ad hoc, chaotic	**Process defined & institutionalized**	**Improvement fed back into process**

* based on technical report CMU/SEI-89-TR-1 from the Software Engineering Institute.


COHESION
A Continuum of Technology
Applicable to all Types of Products & Processes
Self-Contained
Self-Integrated
Fully Integrated
Initial
Defined
Optimized

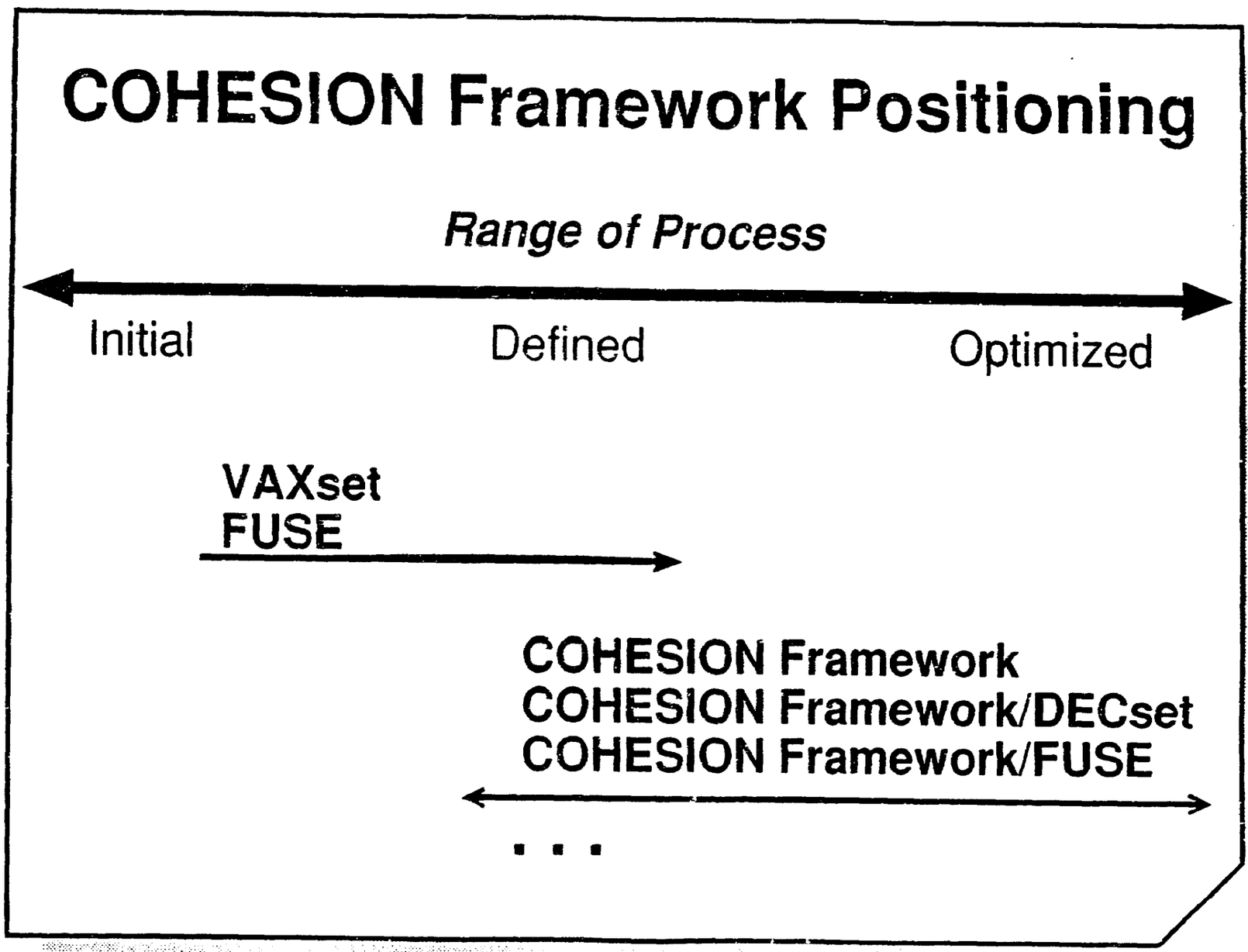
Repository Positioning

Types of Products



CDD/Plus → CDD/Repository

COHESION Framework Positioning



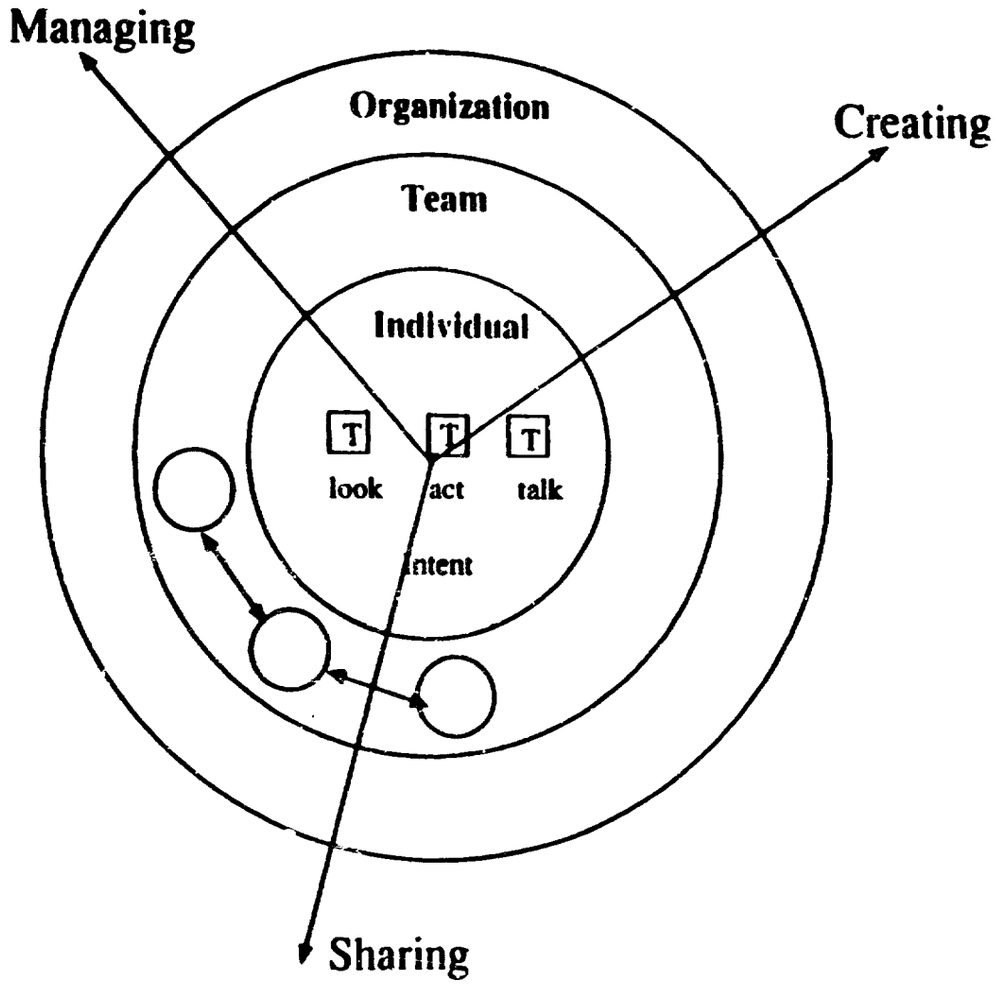
Why COHESION Framework?

- CDD/Repository provides underlying services for key functions required to build a project support environment for CASE.
- Framework builds upon DECwindows/Motif™ (presentation), CDD/Repository (data), ACAS (control) integration within the context of a project-oriented CASE solution.
- The Framework uses CDD/Repository for sharing project metadata.

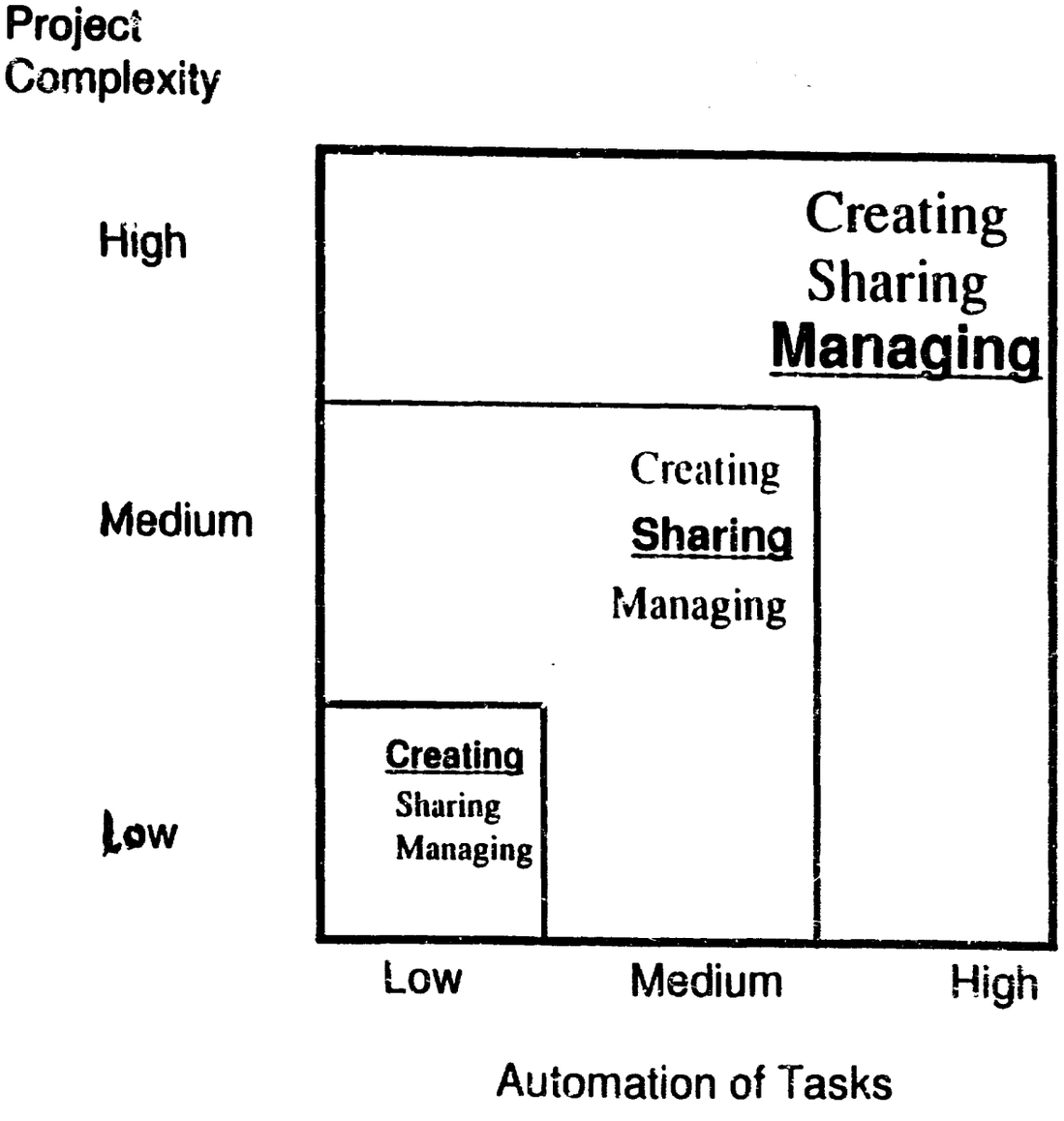
Program Goals

- Provide a common user environment for application development
- Provide consistent, reliable, storage for project data
- Automate policies and procedures
- Manage parallel development
- Allow tracking of changes
- Preserve investments in existing tools
- Support Digital platforms and later others
- Use this technology for our own development

Dimensions:


Managing
Organization
Creating
Team
Individual
T
T
T
look
act
talk
Intent
Sharing

Dimensions: complexity



The Customer's Problem :

For the individual :

* several tools and several tasks : need to remove road blocks
* need to maintain an efficient and effective work flow, data flow, communication flow
* need to maintain and enhance creativity

For the team :

* need to share work
* need to keep track of project schedules and deliverables
* like to have "coordination for free"

For the organization/enterprise :

* need to share resources/processes
* control, policy (on data, etc.)
* need to do more with less : increase business opportunities, competitive advantage

The solution:

For the individual :

* intuitive workflow support
* applying energy to make deliverables, not to understand how to make tools work
* hide the mechanisms : process and environment are set up for you

For the team :

* comprehensive support for multiple streams of development
* support of project methodology and management
* keep individual preferences in the way a person works, while maintaining a consistent project methodology
* new decision support functions

For the organization :

* tie project data to other enterprise processes
* support the enterprise methodology and management
* new business- level decision support functions

The Solution

Must address the individual, the team AND the organization

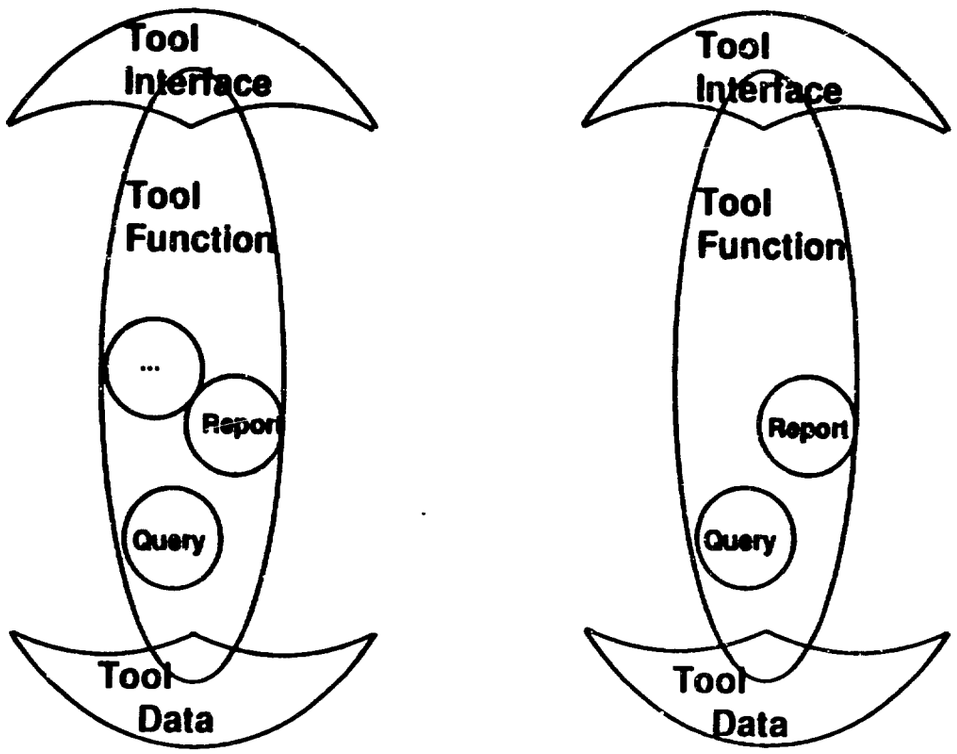
Plug and play : choose the best for you.

Function-based design and packaging to support the customer's work.

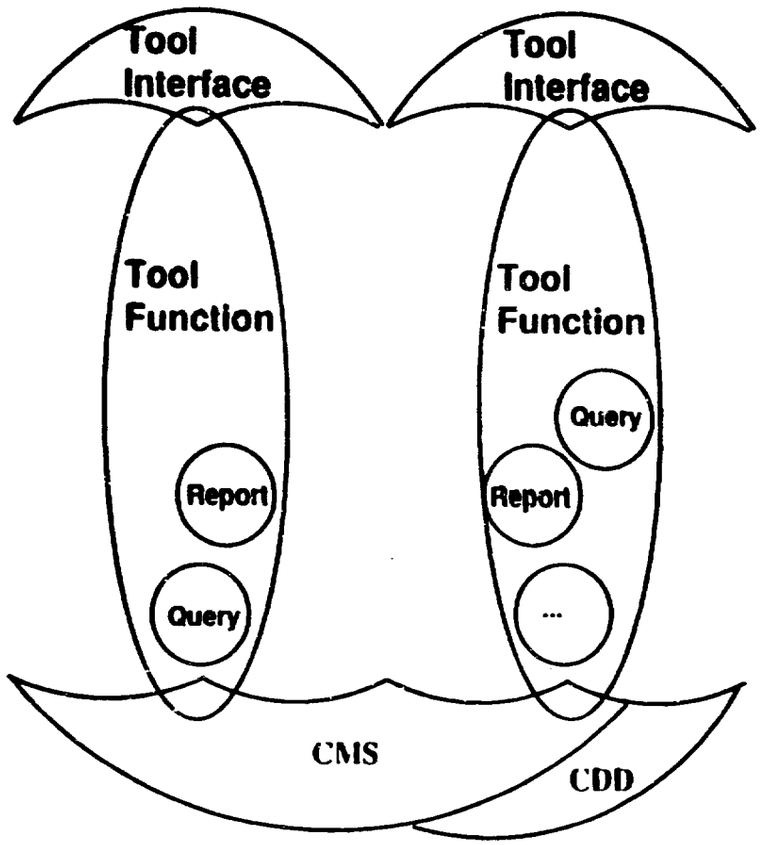
Services for the individual, the team, the organization.

The Architecture

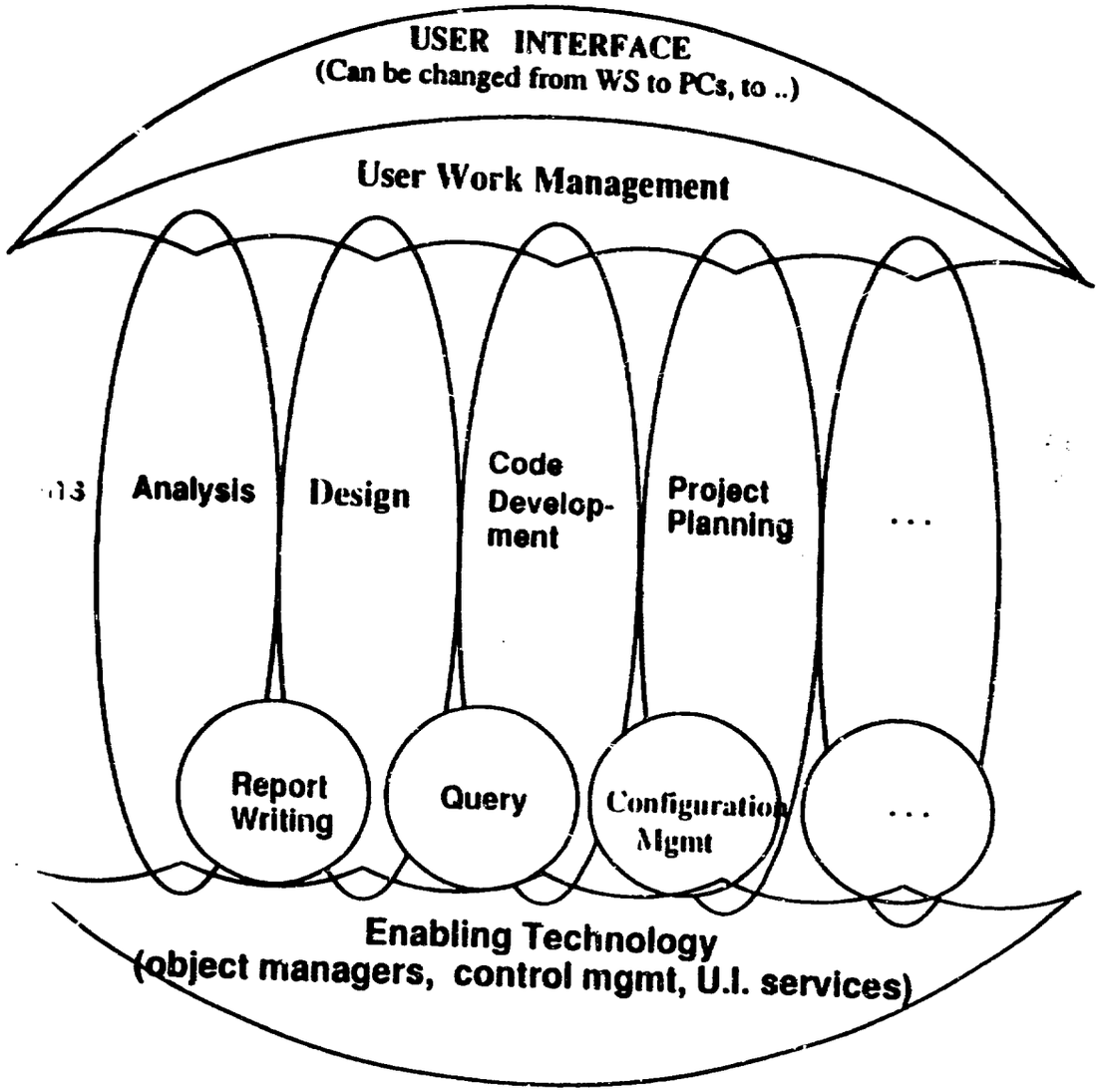
No Sharing


Tool
Interface
Tool
Function
...
Report
Query
Tool
Data
Tool
Interface
Tool
Function
Report
Query
Tool
Data

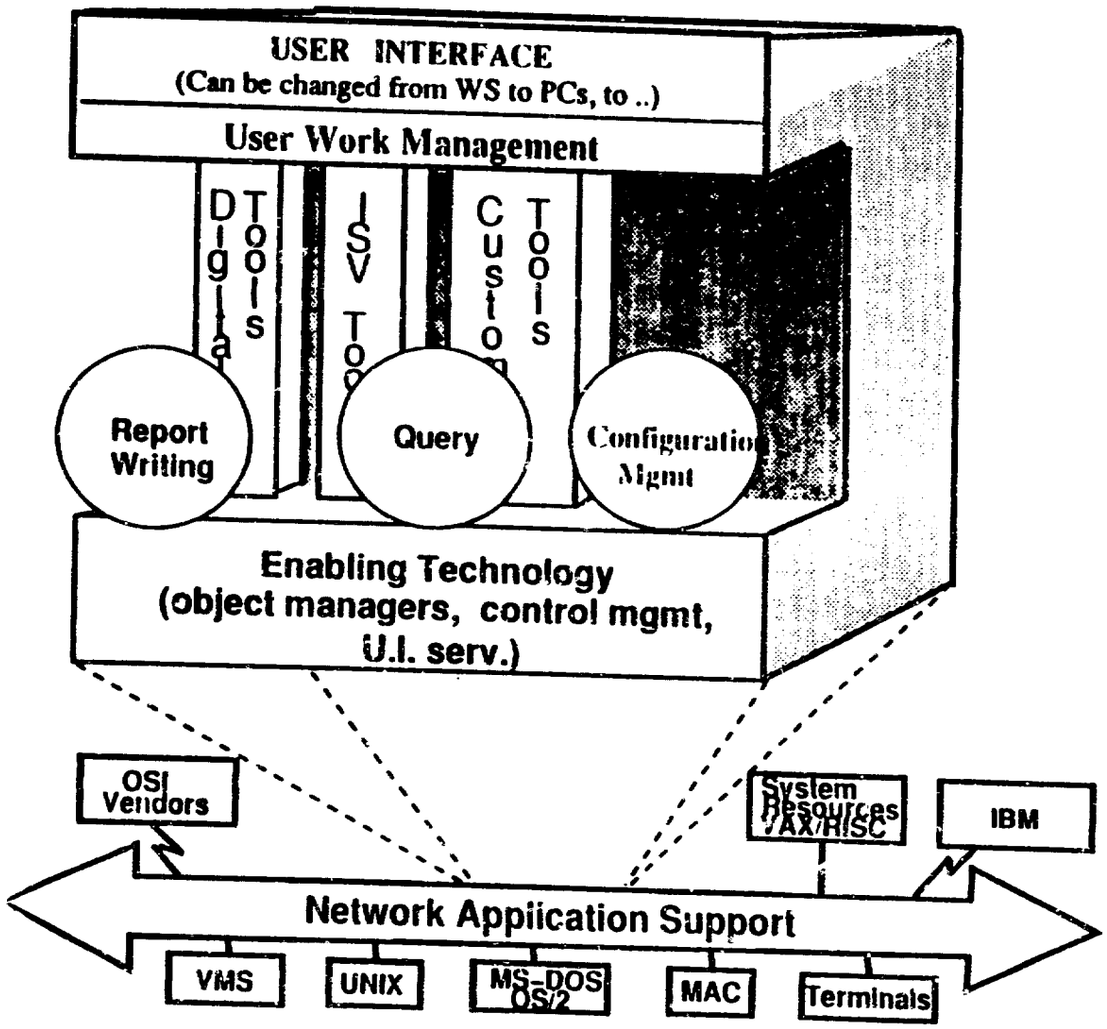
Some sharing /some point to point integration

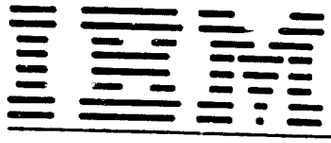


Structure of the Environment

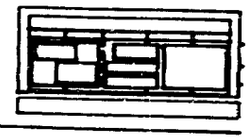


COHESION


USER INTERFACE
(Can be changed from WS to PCs, to ..)
User Work Management
Digital Tools
ISV Tools
Custom Tools
Report Writing
Query
Configuration Mgmt
Enabling Technology
(object managers, control mgmt, U.I. serv.)
OSI Vendors
System Resources VAX/RISC
IBM
Network Application Support
VMS
UNIX
MS-DOS OS/2
MAC
Terminals



BLUEPRINT FOR A
MORE PRODUCTIVE
FUTURE



AD/CYCLE

PLATFORM



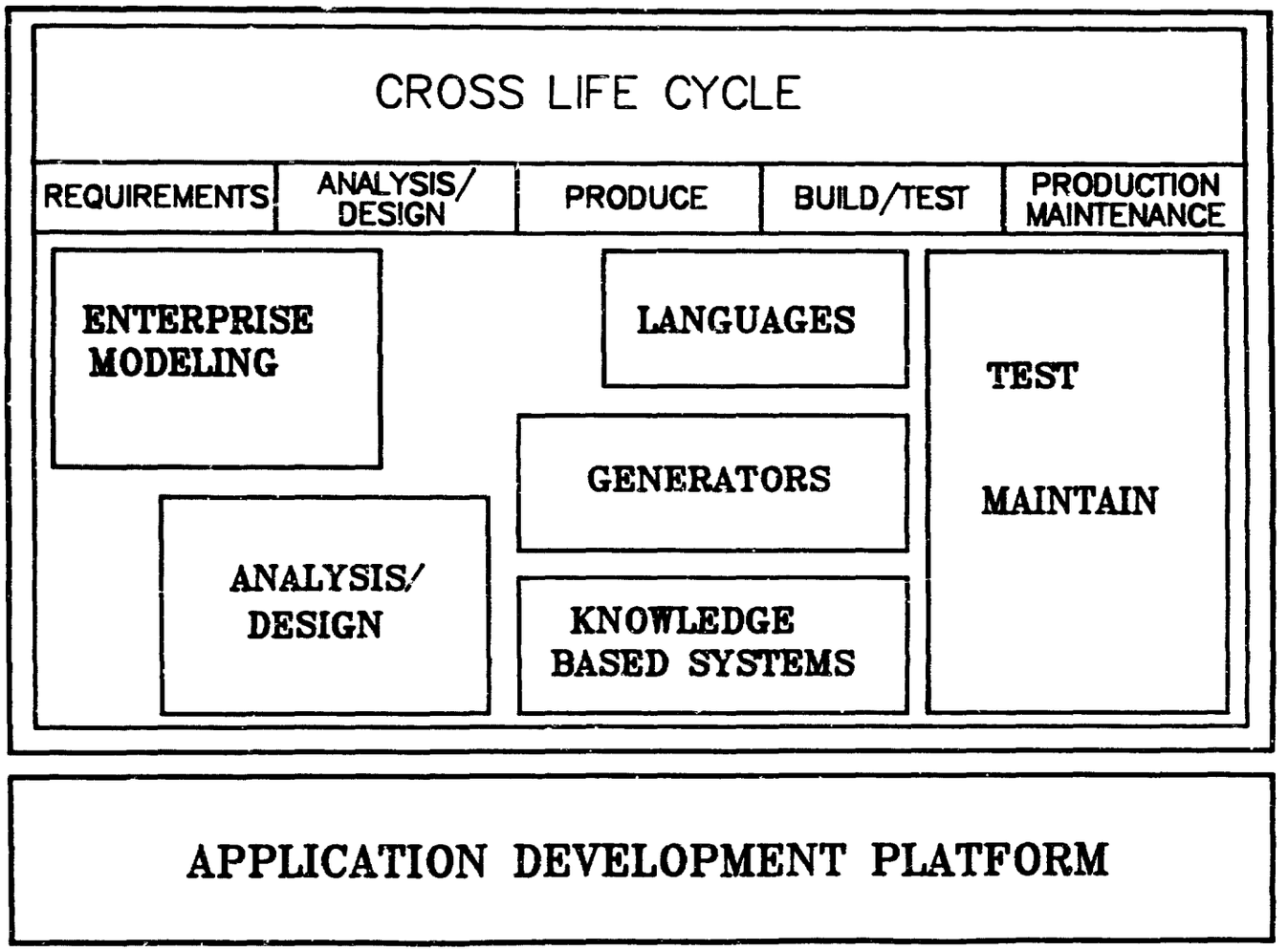
AD/CYCLE PLATFORM

OBJECTIVES

- DESCRIBE THE AD/CYCLE PLATFORM
  - CUA
  - PWS
  - INFORMATION MODEL
  - REPOSITORY/LIBRARY SERVICES
  - TOOL SERVICES
- EXPLAIN VALUE OF EACH COMPONENT
- DISCUSS AVAILABILITY OF COMPONENTS

TOOL INTEGRATION AND MIGRATION

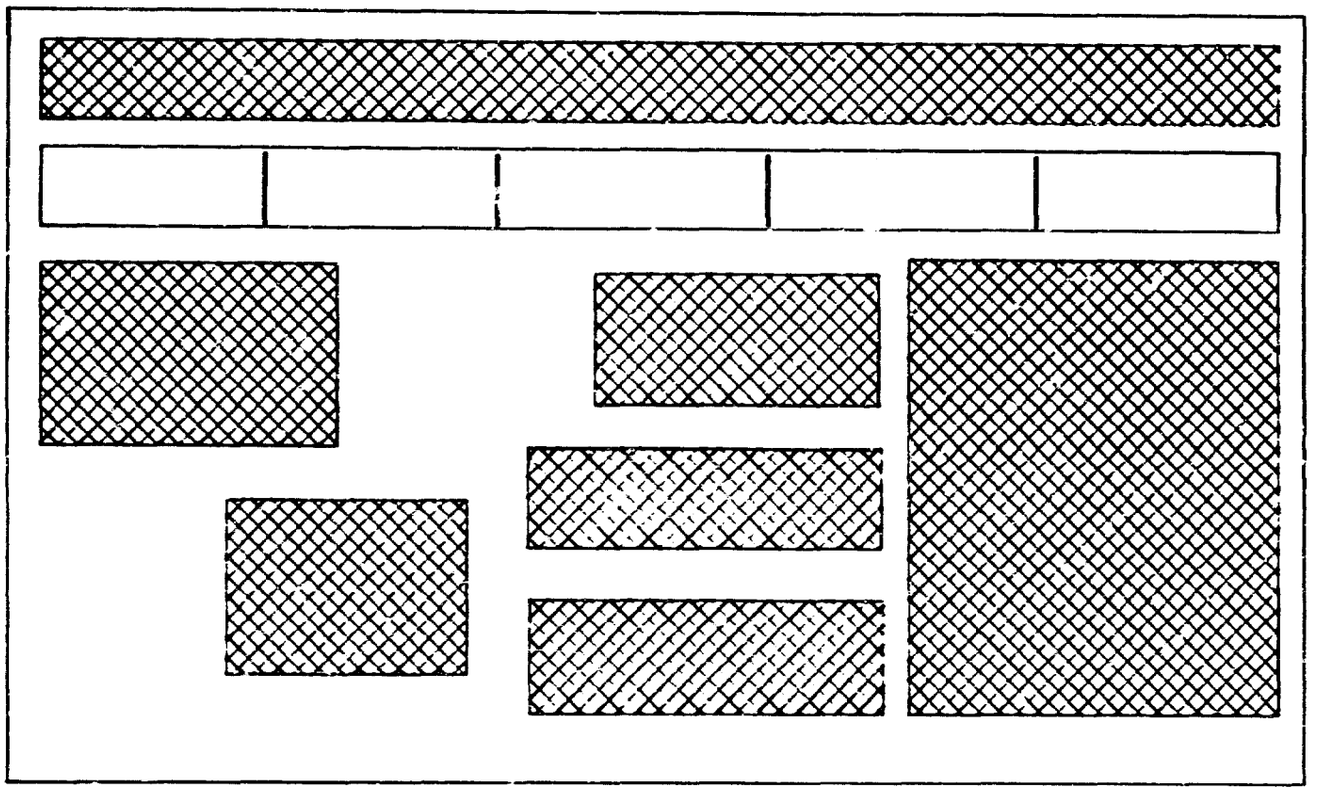
IBM AD/Cycle Model



© Copyright International Business Machines Corporation 1990

BLUEPRINT FOR A
MORE PRODUCTIVE
FUTURE

AD/CYCLE PLATFORM



AD PLATFORM

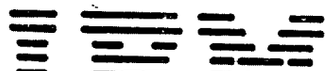
CUA

PWS SERVICES

TOOL SERVICES

AD INFORMATION MODEL

REPOSITORY/LIBRARY SERVICES

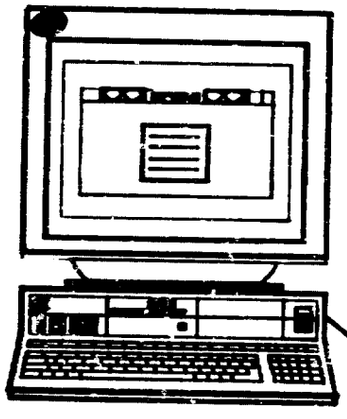


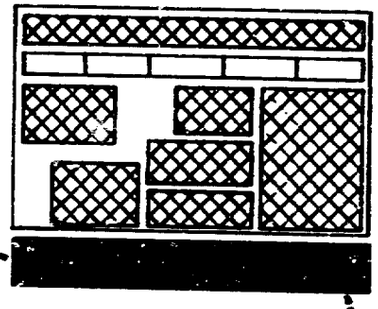
BLUEPRINT FOR A
MORE PRODUCTIVE
FUTURE



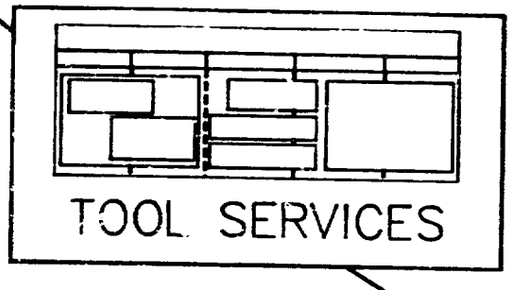
AD/CYCLE PLATFORM

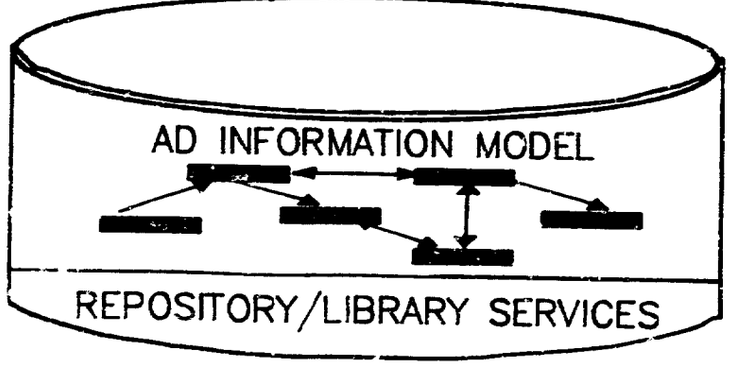
COMMON
USER
ACCESS

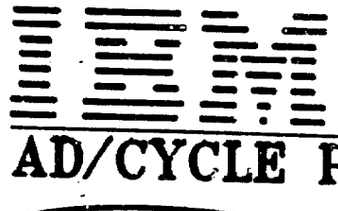




PWS
SERVICES



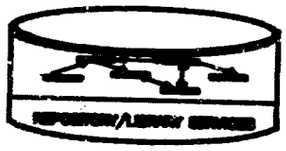




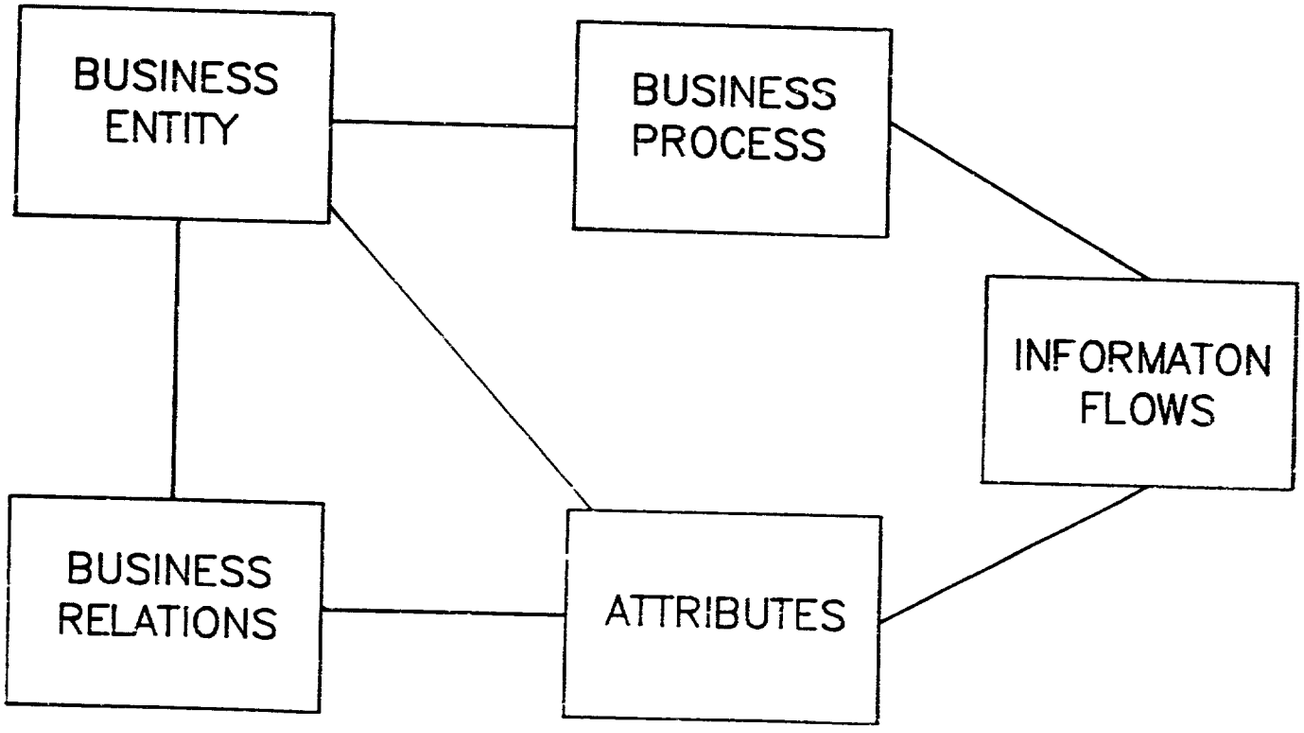
BLUEPRINT FOR A
MORE PRODUCTIVE
FUTURE



AD/CYCLE PLATFORM

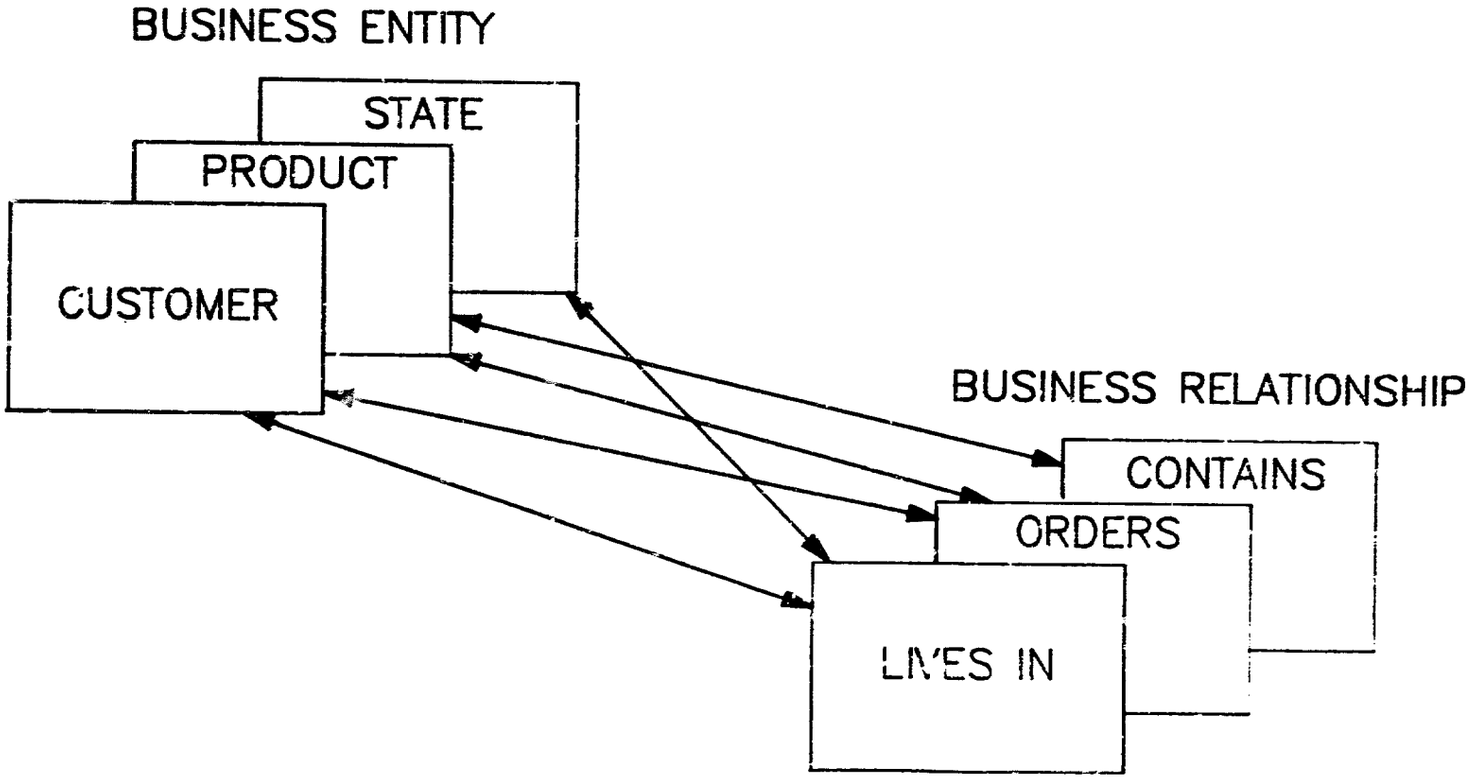


AD INFORMATION MODEL

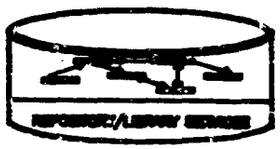


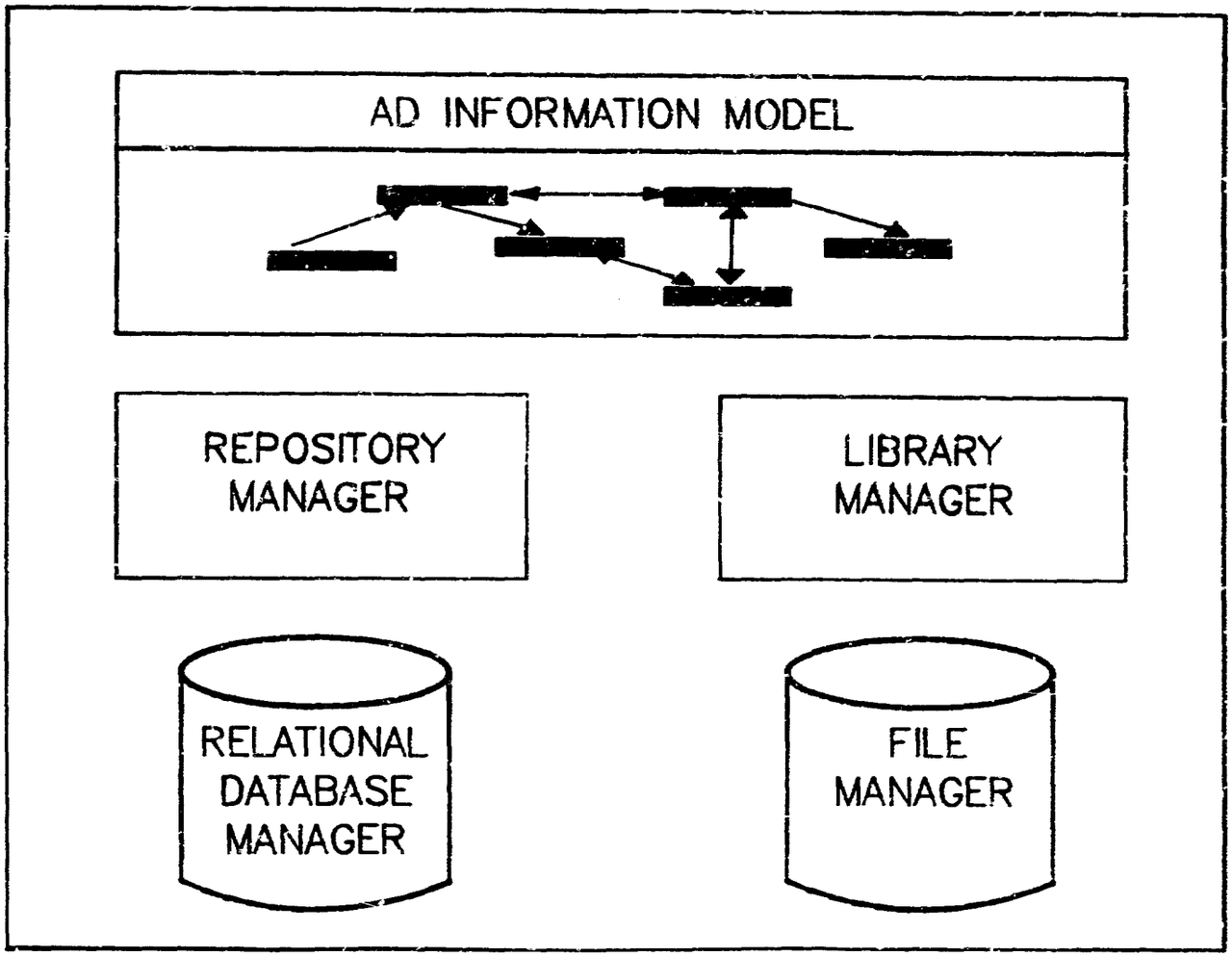
AD/CYCLE PLATFORM

INSTANCES DEFINED BY PARTICULAR BUSINESS



AD/CYCLE PLATFORM



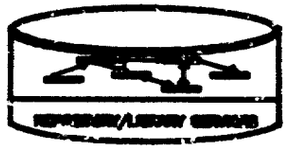




BLUEPRINT FOR A
MORE PRODUCTIVE
FUTURE



AD/CYCLE PLATFORM



REPOSITORY MANAGER

CONFORMANCE OF ENTITY/RELATIONSHIP ELEMENTS TO RULES
IS ASSERTED THROUGH CONSTRAINTS AND POLICIES

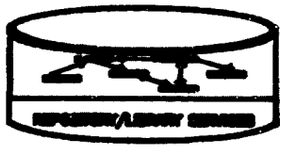
- INTEGRITY
  (eg PROCESS TYPE must be Data or Control)

- DERIVATION
  (eg RECORDLENGTH = Sum of FIELDLENGTHS)

- TRIGGER
  (eg LOG MESSAGE on Data Design Change)

- SECURITY
  (eg only VALIDATION ROUTINE can alter VALIDATED ATTRIBUTE)

AD/CYCLE PLATFORM



REPOSITORY MANAGER

PROVIDES ENTITY/RELATIONSHIP DATA HANDLING TO SUPPORT:

- SHARING OF DATA ELEMENTS
- CONSISTENT REFINEMENT OF APPLICATIONS ACROSS LIFE CYCLE
- IMPACT ANALYSIS
- DATA MODELING
- AUTOMATIC APPLICATION BUILD

AD/CYCLE PLATFORM



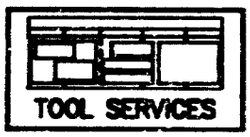
LIBRARY MANAGER

WILL PROVIDE:

- HIERARCHICAL VERSIONING OF APPLICATION COMPONENTS
- CONFIGURATION MANAGEMENT
- NAMING CONVENTIONS FOR ORGANIZING DATA
- AUTOMATIC/CONTROLLED BUILD FOR APPLICATION CONSTRUCTION

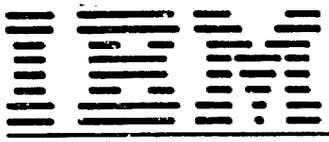


AD/CYCLE PLATFORM



TOOL SERVICES

- PROVIDES COMMON FUNCTION TO ACHIEVE INTEGRATION OF:
  - PRESENTATION
    (EG EXTENDED CUA CONTROL)
  - FUNCTION
    (EG INVOCATION OF TOOLS)
  - DATA
    (EG WORKSTATION ACCESS TO HOST DATA)
- EVOLUTION TO WORK PLACE MODEL AND PROCESS MANAGEMENT



BLUEPRINT FOR A
MORE PRODUCTIVE
FUTURE



AD/CYCLE PLATFORM

- CUA
  - GUIDELINES – GRAPHICAL MODEL
- PWS
  - OS/2 EE
    - – PRESENTATION MANAGER
    - – DBMS/FILE SERVICES
    - – COMMUNICATION SERVICES
- INFORMATION MODEL
  - BASE
    - – FRONT–END ENTITY/PROCESS
    - – DICTIONARY CONSTRUCTS
- TOOL SERVICES
  - WSP
    - – REGISTRATION/INVOCATION
    - – REMOTE DATA ACCESS
- REPOSITORY
  - RM/MVS
    - – ENTITY/RELATIONSHIP MODELING
    - – CENTRALIZED CONTROL/SHARING
- LIBRARY
  - SCLM
    - – AUTOMATIC/CONTROLLED BUILD
    - – VERSIONING



BLUEPRINT FOR A
MORE PRODUCTIVE
FUTURE



AD/CYCLE PLATFORM

9/89

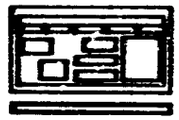
TODAY'S
BASE

OS/2 EE
CUA GRAPHICAL
INFORMATION MODEL BASE
REPOSITORY
LIBRARY
WORK STATION PLATFORM

CUA WORKPLACE INTRODUCTION
INFORMATION MODEL EXTENTIONS
REMOTE REPOSITORY ACCESS
LIBRARY/REPOSITORY CONNECTION
LIBRARY ENHANCEMENTS

CUA WORKPLACE
INFORMATION MODEL EVOLUTION
REPOSITORY VERSIONING
LIBRARY/REPOSITORY SYCHRONIZATION
VM REPOSITORY
OS/400 REPOSITORY/LIBRARY

TOOL INTEGRATION AND MIGRATION

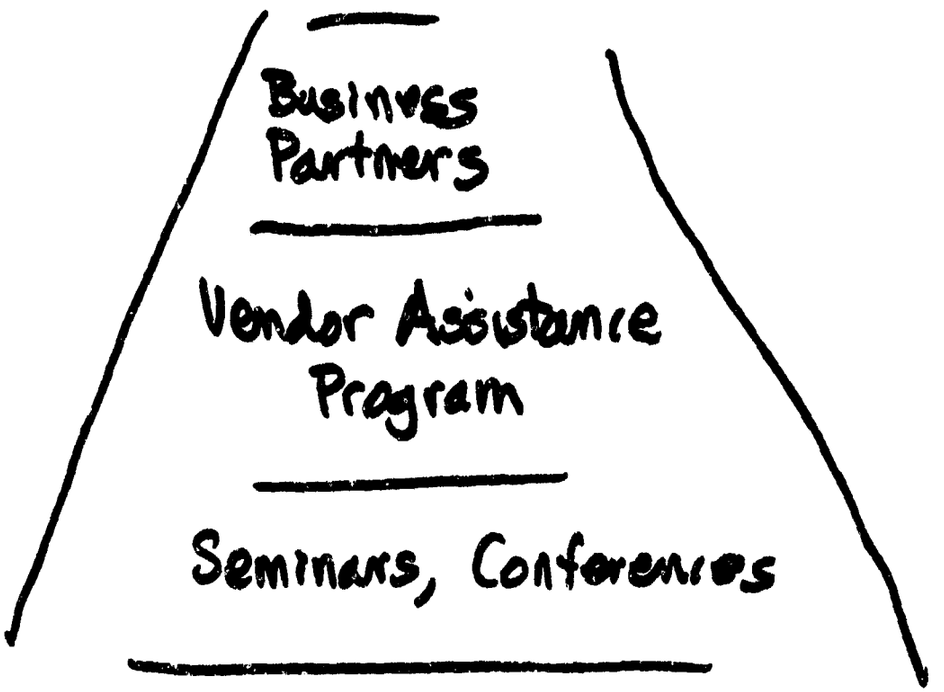


TOOL INTEGRATION CRITERIA

- ☐ THREE DISTINCT ASPECTS
  - ● PRESENTATION
    - APPEARANCE
    - INTERACTION
  - ● FUNCTION:
    - TOOL–TO–TOOL FLOWS
    - REGISTRATION/INVOCATION
    - COMMON SERVICES
    - REDUNDANT FUNCTION
  - ● DATA
    - LIBRARY/REPOSITORY USE
    - DESCRIPTIVE VS DATA ITEMS
    - SHARING BETWEEN TOOLS
- ☐ EVOLVES THROUGH TIME
  - ● TOOL CONFORMANCE FOLLOWS PLATFORM ENABLING

© Copyright International Business Machines Corporation 1990

Vendor Relationships

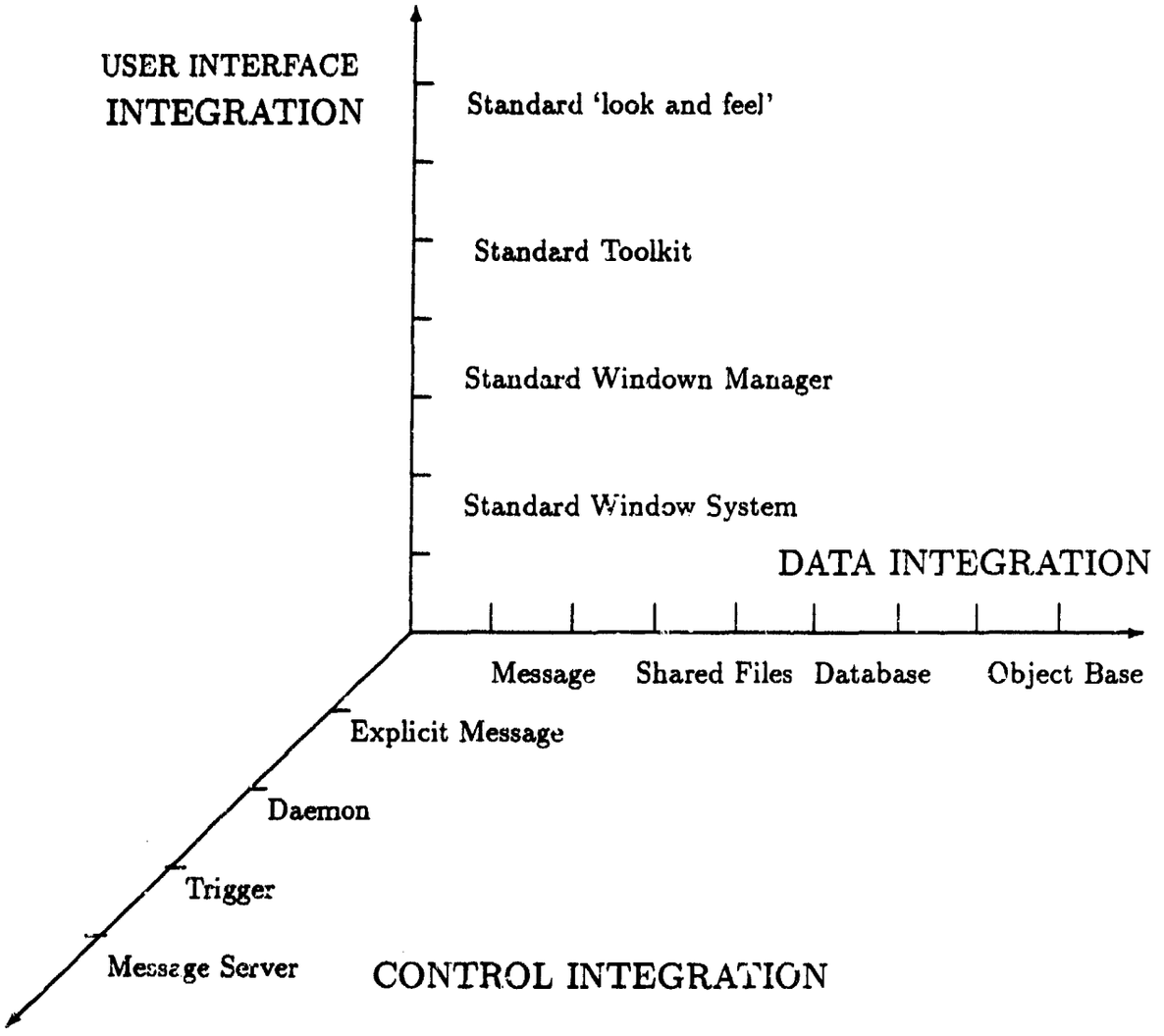


Outline

- Integration Technology
- The Broadcast Message Server
- The BMS on PCTE
- Tools in our Prototype
- Conclusions from our Experience

Toaster + Overlays

Integration Dimensions

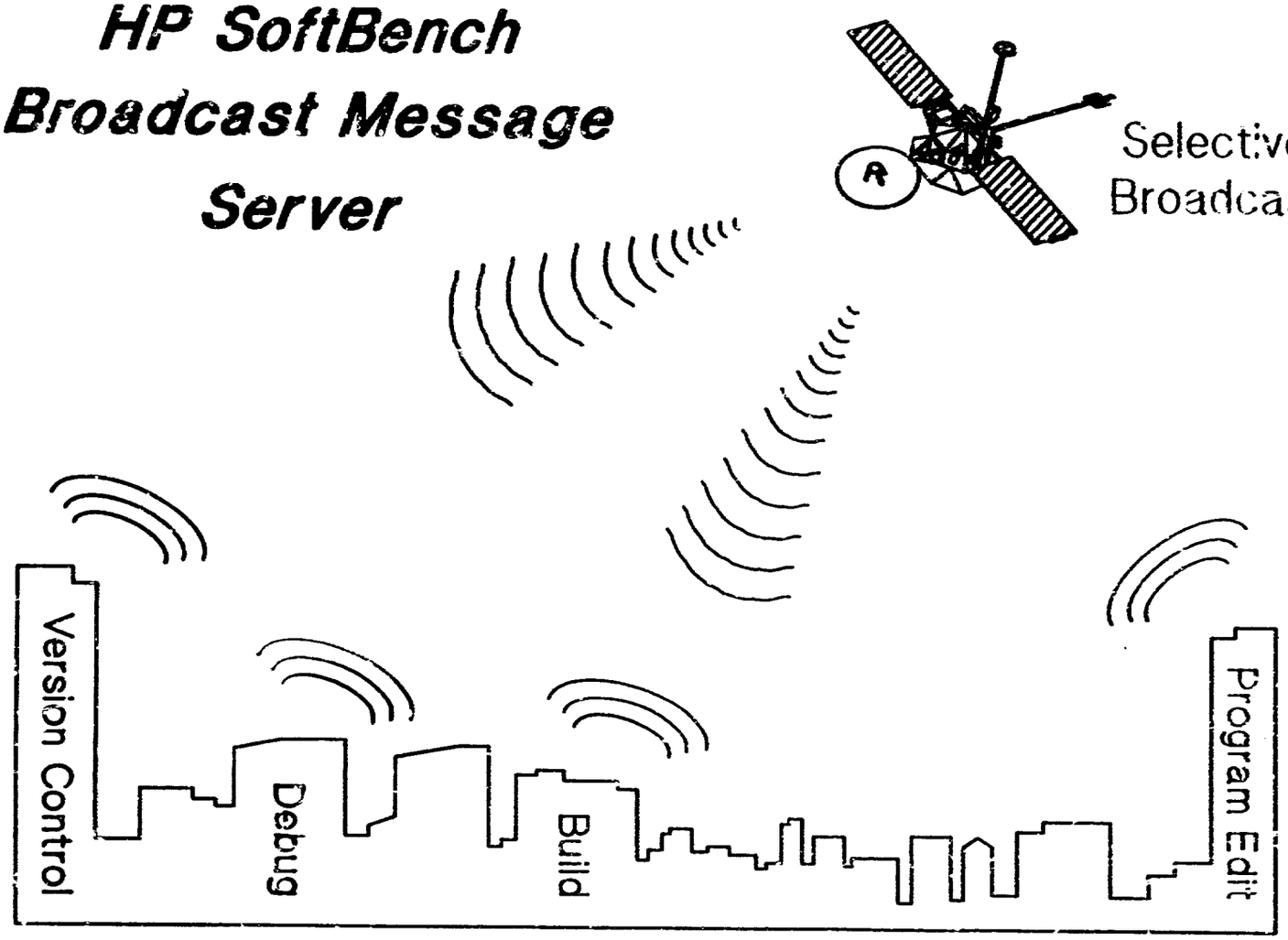


Enabling Standards and Technologies

- **PCTE (OMS) - addressing Data Integration.**
- **HP SoftBench BMS - addressing Control Integration.**
- **OSF/Motif - addressing User Interface Integration.**

The BMS Satellite

HP SoftBench Broadcast Message Server



HEWLETT PACKARD

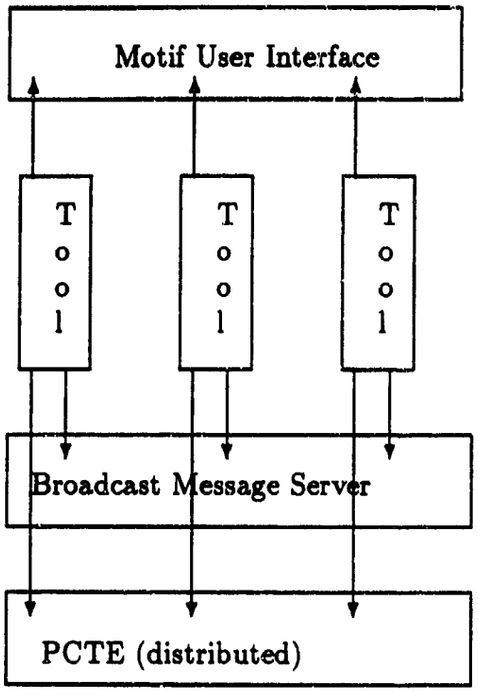
A BMS Environment

- Tools have to broadcast relevant messages;
- Tools have to respond and react to appropriate messages;
- Tools are automatically started when a request is made;
- The BMS is low cost:
  - Easy to implement;
  - Easy to modify tools to use it;
  - Evolutionary not revolutionary.

How does the PCTE BMS differ?

- BMS is implemented entirely upon PCTE (and thus portable) instead of UNIX;
- PCTE message queues for IPC instead of sockets.
- PCTE process control instead of UNIX fork and exec.
- Messages which contain the working schema and current reference object are supported.

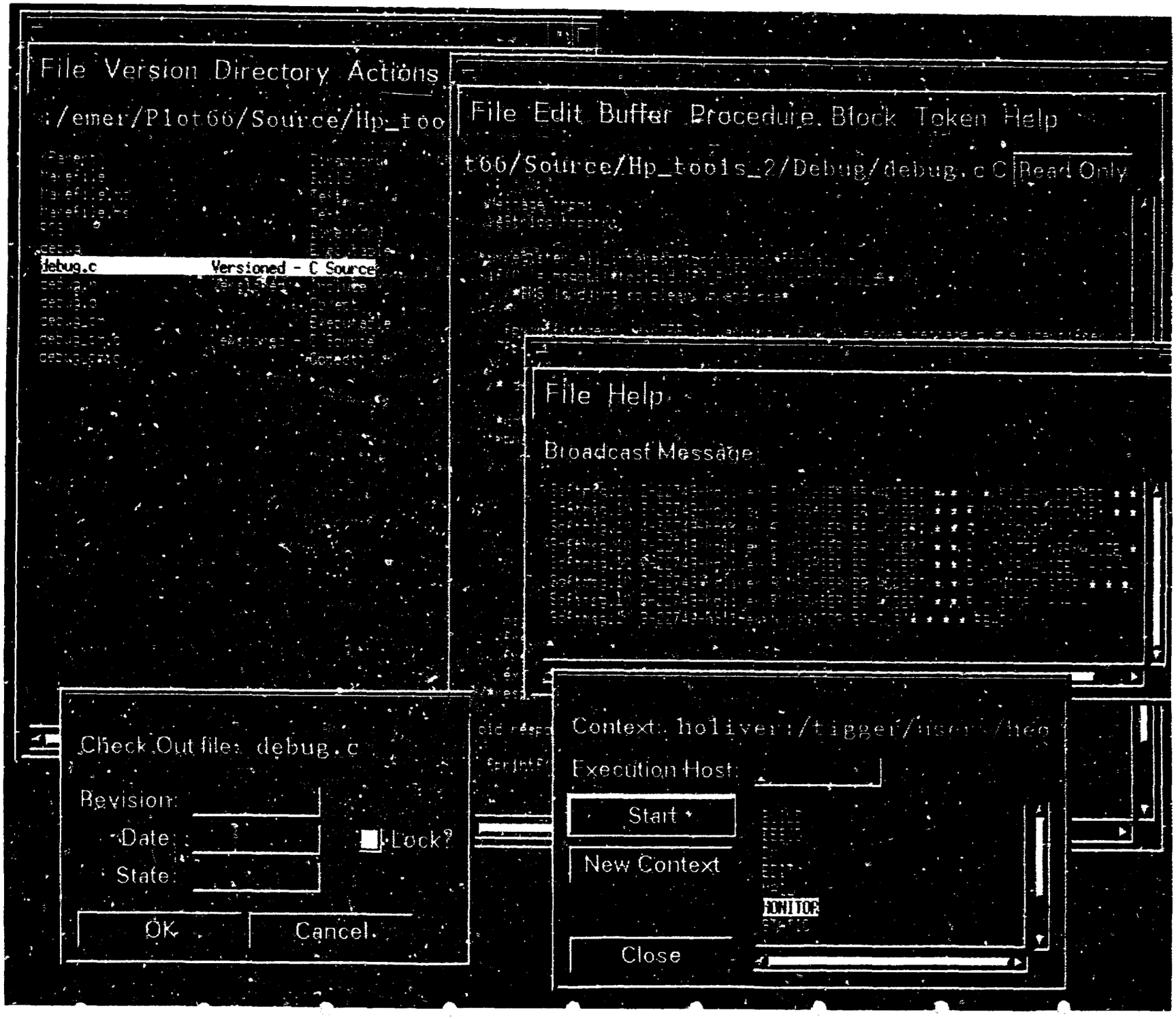
Implementation Architecture

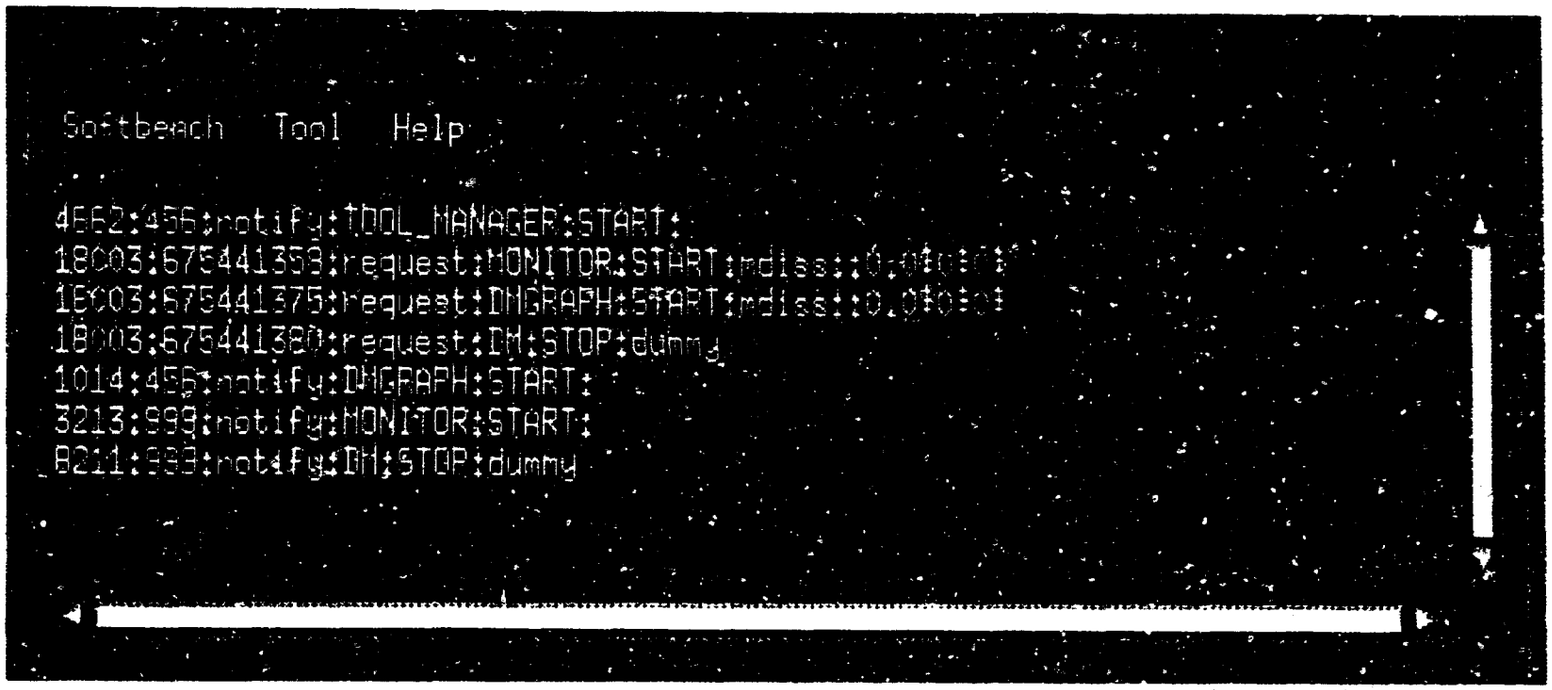


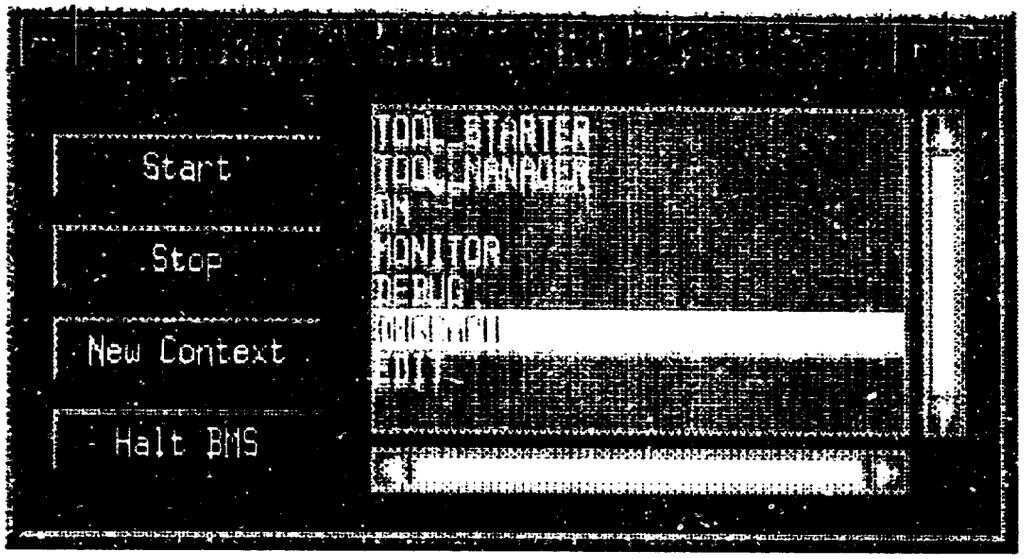
Our Prototype Tools

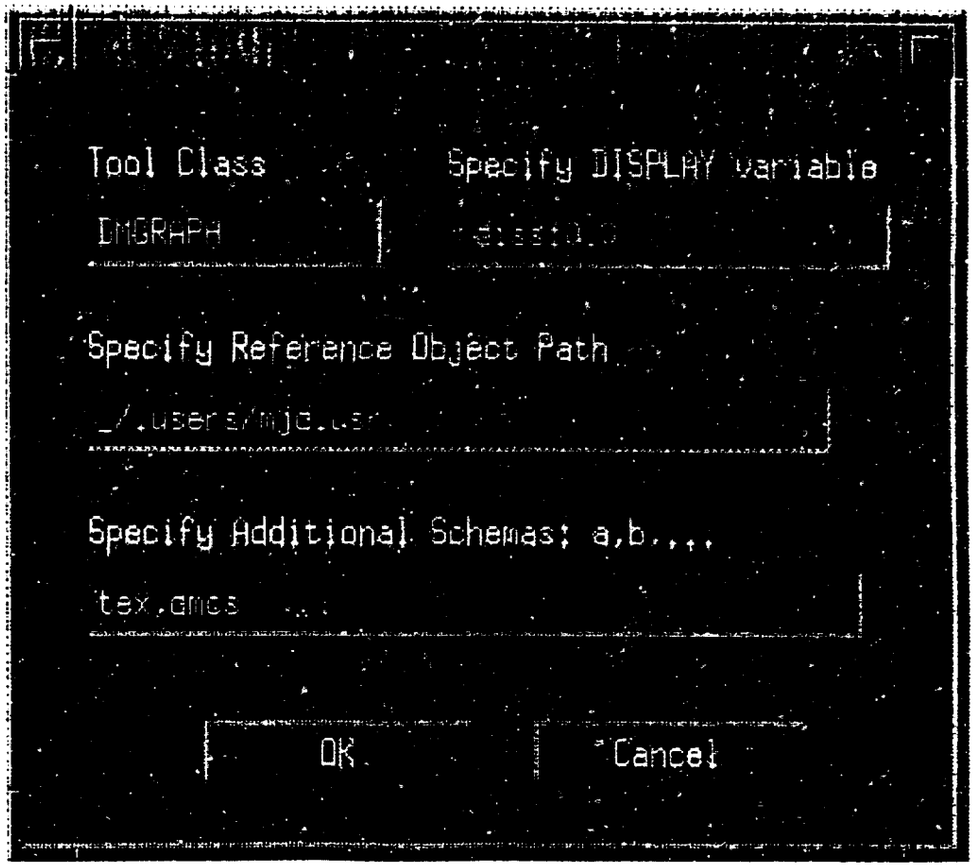
- **Tool Starter**
- **Monitor**
- **Development Manager**
- **Graphical Development Manager**
- **Editor ****

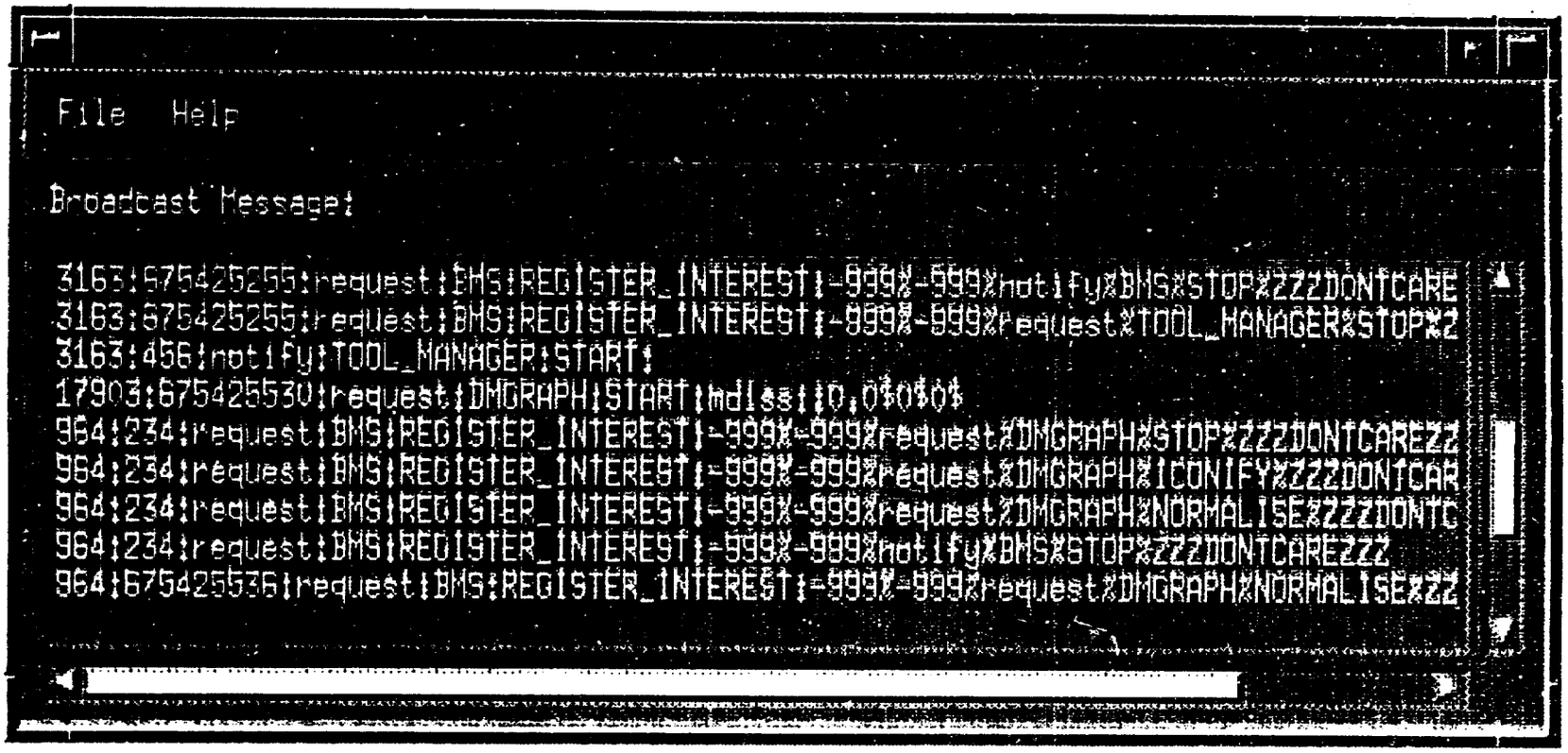
The Prototype Screendumps


File Version Directory Actions
:/emer/Plot66/Source/Hp_too
debug.c
Versioned - C Source
File Edit Buffer Procedure Block Token Help
t66/Source/Hp_tools_2/Debug/debug.c C
Read Only
File Help
Broadcast Message
Check Out file: debug.c
Revision:
Date:
Lock?
State:
OK
Cancel
Context: holiver:/tigger/user
Execution Host:
Start
New Context
Close

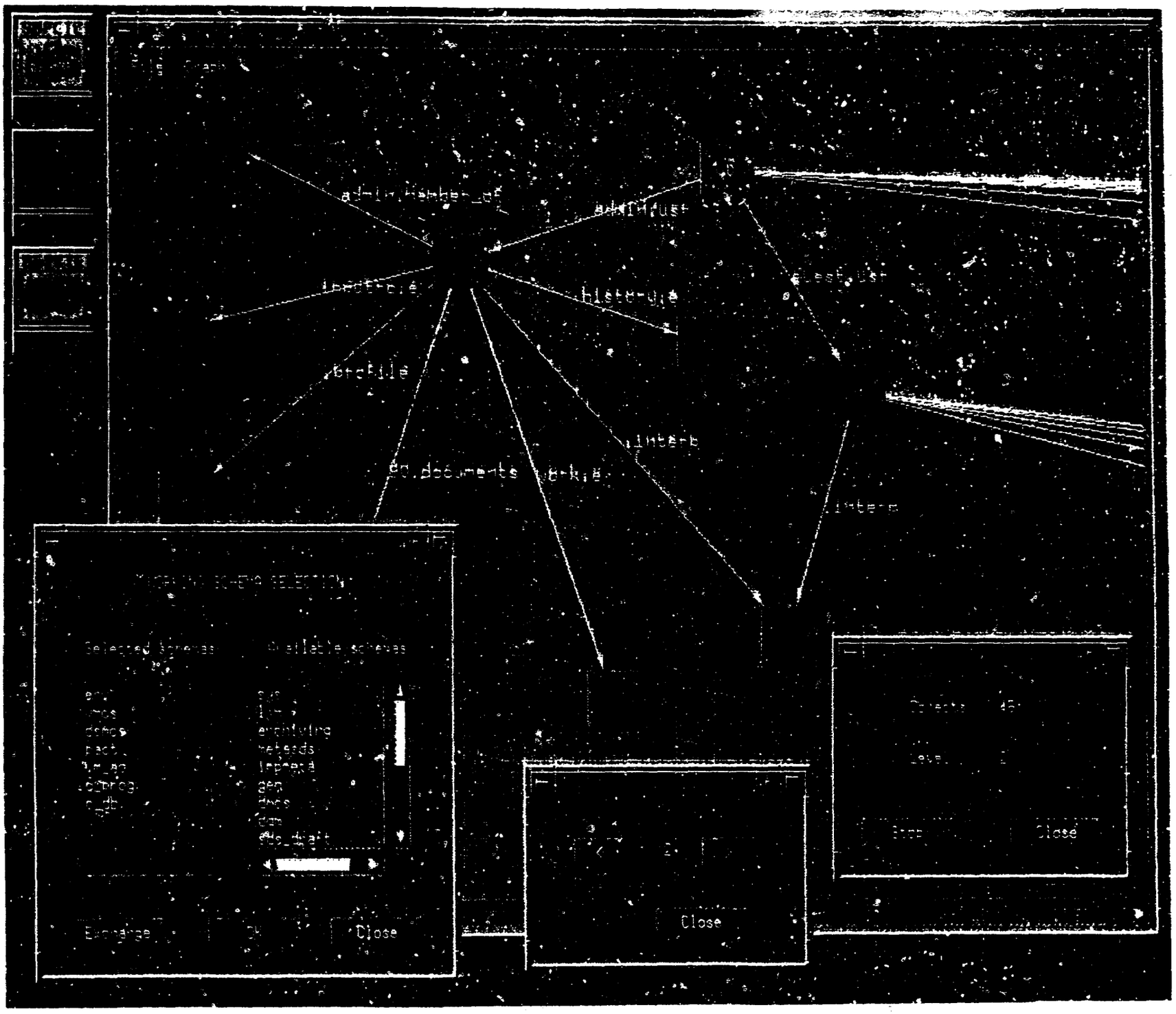

Softbench Tool Help
4662:456:notify:TOOL_MANAGER:START:
1EC03:675441375:request:DMGRAPH:START:mdiss::0.0:0:0:
1014:456:notify:DMGRAPH:START:
3213:999:notify:MONITOR:START:
8211:999:notify:DM:STOP:dummy

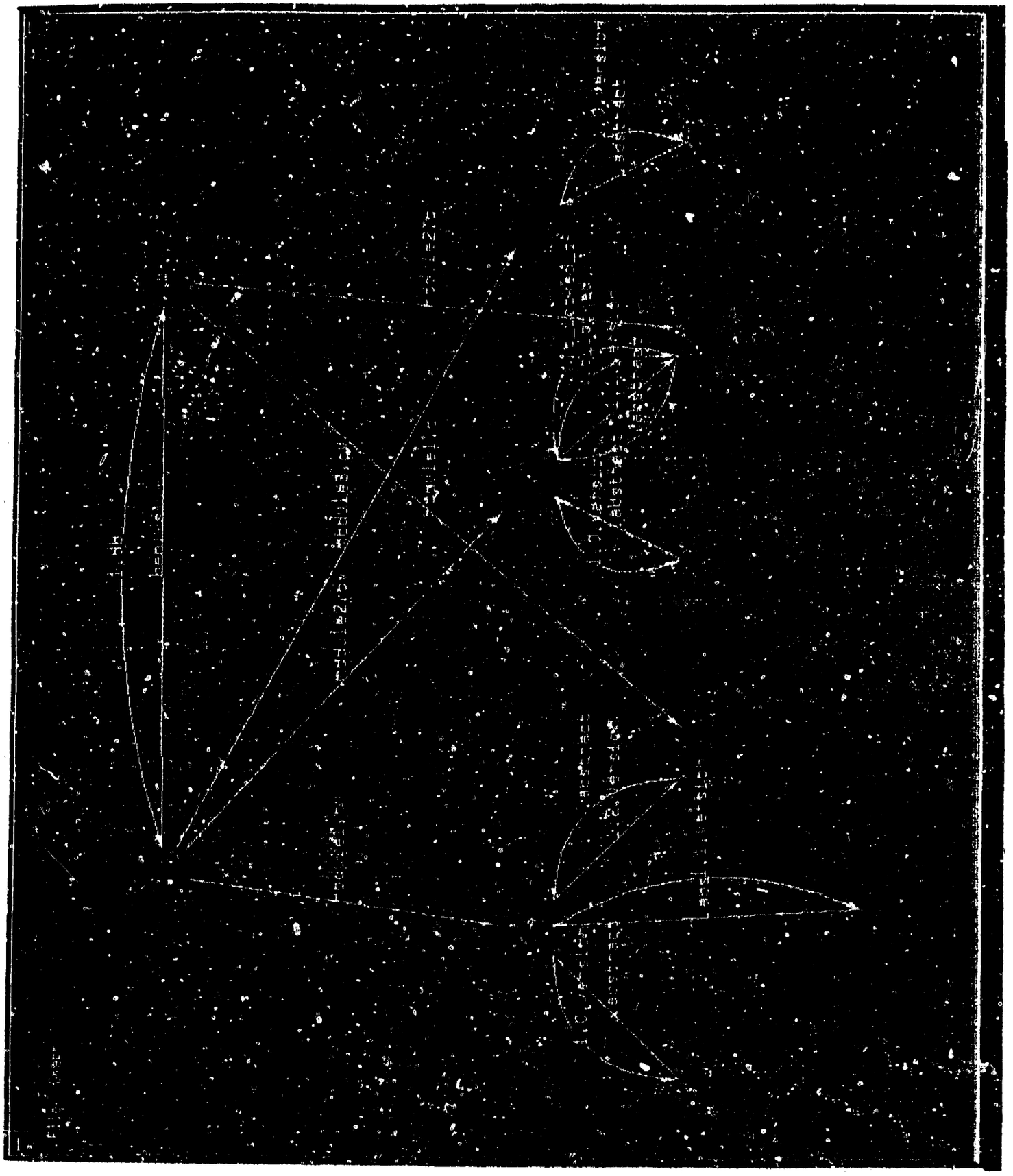

Start
Stop
New Context
Halt BMS
TOOL_STARTER
TOOL_MANAGER
DM
MONITOR
DEBUG
EDIT

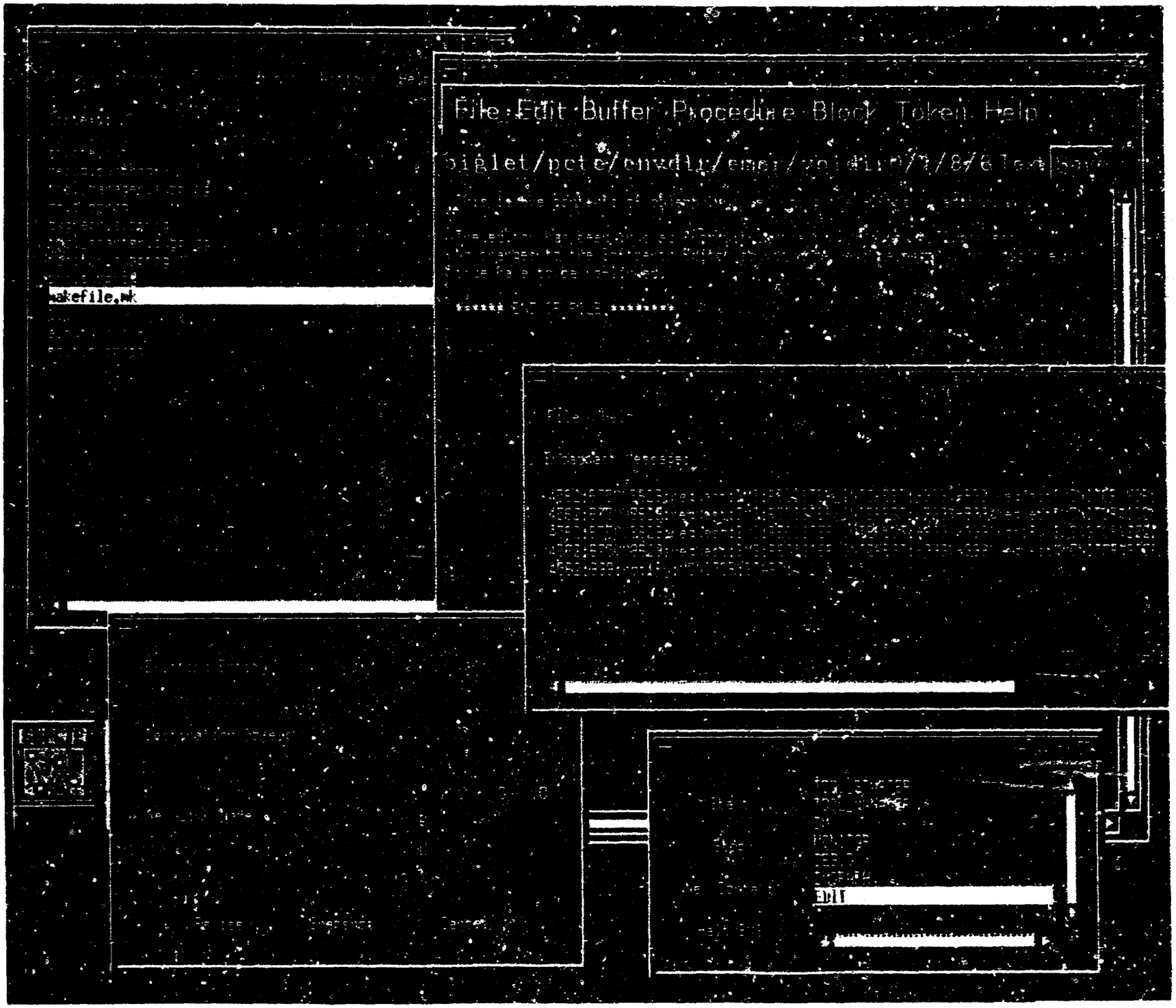

Tool Class
Specify DISPLAY Variable
DMGRAPH
Specify Reference Object Path
Specify Additional Schemas: a,b,...
tex.dmgs
OK
Cancel


File Help
Broadcast Message:
3163:675425255:request:BMS:REGISTER_INTEREST:-999%-999%notify%BMS%STOP%ZZZDONTCARE
3163:675425255:request:BMS:REGISTER_INTEREST:-999%-999%request%TOOL_MANAGER%STOP%Z
3163:456:notify:TOOL_MANAGER:START:
17903:675425530:request:DMGRAPH:START:mdlss:!0,0$0$0$
964:234:request:BMS:REGISTER_INTEREST:-999%-999%request%DMGRAPH%STOP%ZZZDONTCAREZZ
964:234:request:BMS:REGISTER_INTEREST:-999%-999%request%DMGRAPH%ICONIFY%ZZZDONTCAR
964:234:request:BMS:REGISTER_INTEREST:-999%-999%request%DMGRAPH%NORMALISE%ZZZDONTC
964:234:request:BMS:REGISTER_INTEREST:-999%-999%notify%BMS%STOP%ZZZDONTCAREZZZ
964:675425536:request:BMS:REGISTER_INTEREST:-999%-999%request%DMGRAPH%NORMALISE%ZZ


File Version Current Object Actions Help
Context: heo.user
1..
emerl.member_of
startup_msgs.e
bms.prog
demo.prog
code_dev.prog
backup_start.e
george.prog
startup.e
inputrc.e
demo_sqs.e
databases.e
Existing Entity
.U/inputrc.e
Destination Object
.U
New Link Name
inputrc.e
Revise
Snapshot
Cancel


Selected schemas
Available schemas
Close
Close
Close




File Edit Buffer Procedure Block Token Help
Makefile.mk
Edit

Our Experiences

- The provision of data, control and user interface integration from a PCTE and SoftBench combination is technically feasible

- We believe that tools can achieve tight integration by using data, control and user interface integration support from the framework.

- PCTE does provide help for the tool writer;

- PCTE is an effective portability platform;

- designing for PCTE and SoftBench is good engineering practice.

Summary

Frameworks can provide data, control and user interface integration.

We believe that a PCTE and SoftBench combination adds value to PCTE and to SoftBench.

This can be achieved in an evolutionary way.

Adding Control Integration to PCTE

Huw Oliver,
Hewlett-Packard Laboratories,
Bristol BS12 6QZ, England.
e-mail: heo@hplb.hpl.hp.com
Copyright ©1991, Hewlett-Packard Company

Abstract

The PCTE interfaces provide data-integration services. In a good Software Engineering Environment (SEE), however, more is needed: control integration to automatically start tools and share services. We report on our intermediate practical experience of adding control integration to PCTE. More precisely, we show how Broadcast Message Services can be layered on the PCTE platform thus forming a SEE framework that spans the tool integration dimensions.

Keywords: CASE, Software Engineering Environments, PCTE, SoftBench.

1 Introduction

The computer support environment provided for software engineering today typically consists of a set of standalone tools. These tools are monolithic. These tools do not usually cooperate. They cannot access each other's functionality. They have no access to each other's data (and would not be able to understand it if they could). Their user-interfaces differ widely.

The tools are monolithic in that they provide many of the services more naturally provided by the framework within which they operate or by other tools. For instance, some document processing tools today offer version control even though it is also provided by configuration management tools. Such tools provide so much because the tool providers have no way of composing tools from small modular pieces.

We are interested in how the framework can provide different types of composition or integration services. These integrating services would help tools to be smaller, more modular and built into the support environment as needed by the software engineer.

A complete support environment for software engineering will be a large, complex system. Neither the high level of financial resources nor the wide range of expertise required to provide all the elements of a support environment will be found within a single organisation. The use of open standards for these elements is an essential enabling factor for the production of quality SEE implementations.

We have looked at two technologies which provide elements of a support environment and investigated how they might be combined. The technologies are SoftBench [2] [4] and PCTE [7] [8] [9].

SoftBench is a product of Hewlett-Packard. It consists of an integration framework and an integrated set of tools.

PCTE stands for "a basis for a Portable Common Tools Environment". PCTE defines an interface to support CASE tools and development environments. PCTE itself does not provide any tools: it is a framework on which to build and integrate tools. The development of the interface has culminated in the ECMA PCTE abstract specification [9] which the ECMA general assembly adopted as an ECMA standard in December 1990.

We have undertaken a prototyping activity to show how these components can be combined. The goals of our prototyping activity are to investigate how to construct a support environment, to learn how to use it and to examine the benefits of working with it. We are using an implementation of version 1.5 [7] of the PCTE interface in our prototyping activities. We report here our intermediate technical results from constructing the prototype SEE framework in the HP research laboratories.

2 Integration Services in an SEE Framework

The ECMA CASE environment framework reference model [1] identifies and defines integration services that a framework may provide to support a SEE, and groups related integration services together. Figure 1 shows the overall structure of the reference model (this is an conceptual architecture not an implementation architecture).

The reference model (RM) can be used to categorise the services offered by an SEE framework. Although the ECMA RM activity was spawned from the ECMA PCTE Standards committee, the RM is completely independent of PCTE. The RM can be used to position standards proposals and commercial products, and helps to understand the relationships between different framework offerings.

This section quickly sketches the services required of an SEE framework in terms of those detailed in the RM.

The RM identifies three main aspects of tool integration:

- Data Integration (addressed by the data repository plus data integration services) is the sharing of data and descriptions of that data (schemas) between the users and tools of the support environment.
- Control Integration (addressed by the task management plus the message services) is the management of cooperation between independently developed tools to achieve a coordinated effect.
- User Interface Integration (addressed by the user interface services) is a common look and feel for tools.

2.1 Data Integration

The maintenance, management, and naming of data entities or objects and the relationships among them is the general purpose of the data repository services.

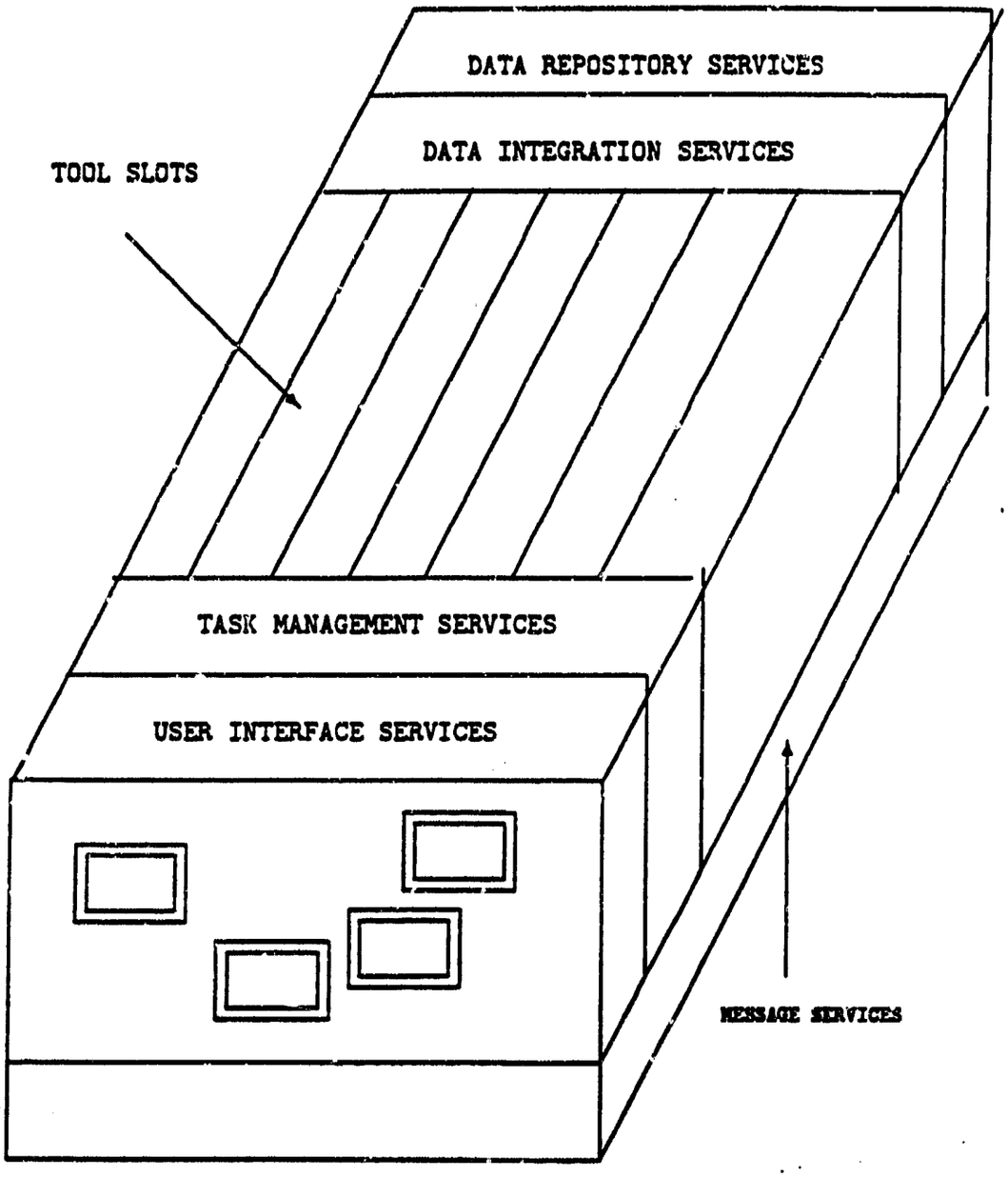


Figure 1: Reference Model Structure

Basic support for process execution and control is also addressed here along with a location service to support physical distribution of data and processes.

The data integration services enhance the data repository services by providing higher-level semantics and operations with which to handle the data stored in the repository.

2.2 Control Integration

A high level of control integration implies that a tool can invoke or stimulate another tool to perform some piece of the software process. Control integration is governed by the extent to which a tool makes it possible for other tools to invoke the functionality it provides, and the extent to which the tool calls other tools to communicate changed circumstances.

The message services aim to provide a standard communication service that can be used for inter-tool and inter-service communication.

2.3 User Interface Integration

User interface services are required by all applications. Efforts such as OSF/Motif provide generic services which are suitable for SEEs.

3 Enabling Technologies

The application of the RM to an interface definition will result in a detailed analysis of what SEE framework services are covered by that interface. We have carried out several such applications. Included among these are the application of the RM to PCTE and to SoftBench.

The following important points result from positioning PCTE 1.5 and the tool integration component of HP's SoftBench environment against the RM:

- PCTE covers the majority of the data integration facilities;
- SoftBench addresses control integration via its Broadcast Message Server.
- SoftBench addresses user interface integration via OSF/Motif.

SoftBench treats control integration as an orthogonal issue to data management. SoftBench can be used with many different repositories. We chose PCTE because of its wide coverage of data management facilities and because it is a standard tool portability platform.

From the point of view of integration technology, SoftBench and PCTE are complementary and add value to one another. This analysis encouraged us to investigate the combination of the SoftBench and PCTE integration technologies in practice. We next give an overview of each of SoftBench and PCTE and then describe our approach to combining them.

3.1 SoftBench

The SoftBench environment consists of a set of integration services and an extensible set of tools that communicate by sending and receiving messages. From the point

of view of an environment builder, SoftBench consists of the Broadcast Message Server(BMS), the Execution Manager (EM), the user interface, support for distribution, the set of tools and the Encapsulator. The BMS and the Execution Manager are described in further detail by Cagan [2]. Further information about SoftBench tools can be found in Gerety's description [4].

3.1.1 SoftBench Integration Services

1. SoftBench's Broadcast Message Server (BMS) enables executing SoftBench tools to cooperate in supporting a software engineer to carry out tasks. Executing tools in SoftBench send a message to the BMS when they: require a service; have performed an action that may be of importance to others; or have a failure to report. The BMS forwards this message to all the executing tools that have registered interest in a "message-pattern" that the message matches (so the message 'broadcast' is in fact selective). Messages can be sent to the BMS by tools so they can register and unregister interest in patterns.
2. The Execution Manager (EM) in SoftBench keeps track of the tools that are executing. The execution manager cooperates closely with the BMS so that when a request message is received by the BMS, the EM determines whether a new tool should be started to service that request or whether the request can be satisfactorally handled by a tool that is already running. SoftBench tools are grouped into classes. Differing criteria can be applied for differing classes of tools. A class is a set of tools that provide equivalent services. Example tool classes are EDIT, COMPILE, or DEBUG.
3. All SoftBench tools have a common look and feel which conforms to the OSF/Motif [3] standard.
4. SoftBench is designed to operate over a distributed network of workstations, and offers distributed computing support of three kinds. Firstly, SoftBench can start tools and support transparent communications between tools executing on remots hosts. Secondly, SoftBench tools are built on the network transparent X Window System which means that programs can run on one system and display visually on another. Thirdly, SoftBench supports access to remote data.

3.1.2 SoftBench Tools

The initial set of tools delivered with the SoftBench product concentrates on support for developing, versioning, and debugging C and C++ programs. An increasing number of Encapsulated tools are available to extend the core environment, for example tools for configuration management, documentation, structured analysis and structured design, and testing.

Some fundamental SoftBench tools of particular relevance to our work to date are the Tool Manager, the Message Monitor and the Development Manager.

- The Tool Manager presents a way for a user to directly invoke tools. While this is useful at the start of a work session the user will later take advantage of the BMS and EM support for control integration. The user will normally

be working within a particular tool (such as a debugger) and will be accessing the functionality of other tools from within that tool.

- The Message Monitor displays all messages that get sent in the environment.
- The Development Manager offers a view of the underlying file system, including an indication of the type of information held within the file (e.g. C source or build information). It also presents a set of operations available on those files (such as versioning). The set of operations made available dynamically matches the type of the file (e.g. it is not possible to even try to check-out a non-versioned file).

We see in section 4.2.3 how we have modified these tools to run on a combined SoftBench and PCTE framework.

The user sees tools working synchronously because cooperation between tools can be specified and the SoftBench system supports the execution of that cooperation. For example, should the user change the source code of a program while working in the static analysis tool, notification of those changes are automatically forwarded to any editor working on the source file for that code. The SoftBench user is also presented with seamless functionality (synergy) in that the services provided by one tool appear (to the user) to be available from several other tools also. For example, code can be recompiled through a user request to the debugger (which is automatically forwarded to the build tool via the BMS). The real benefit of SoftBench to a software developer is that it makes available these advantages of well-presented control integration.

SoftBench provides a further tool called the Encapsulator. This tool enables existing tools to be integrated into the SoftBench support environment without source code modification. It enables a wrapper to be developed for a tool so that its input and output is monitored. Suitable SoftBench messages can then be sent and acted upon by the encapsulated tool, and a SoftBench user interface can be developed so that the tool looks as well as behaves like a true SoftBench citizen (although this holds for a particular set of tools: those that can use standard input/standard output and that can be decoupled from any bitmapped screen handling they do).

3.2 PCTE Integration Services

A major contribution of PCTE is its Object Management System (OMS), designed to meet the data integration needs of CASE tools. The OMS provides the ability to model relationships between data objects, by supporting a variant of the entity-relationship-attribute data model. Object management facilities include typing, schemas and transactions to support data structuring and data sharing, and to maintain data integrity.

PCTE provides a complete interface for the tool writer, including process management and inter-process communication. PCTE provides synchronous and asynchronous calling of processes on local or remote hosts. The services provided are at a higher level of abstraction than those typically provided by the operating system. PCTE inter-process communication services are provided via the PCTE message queue. These services are closely modelled on the X/Open System V

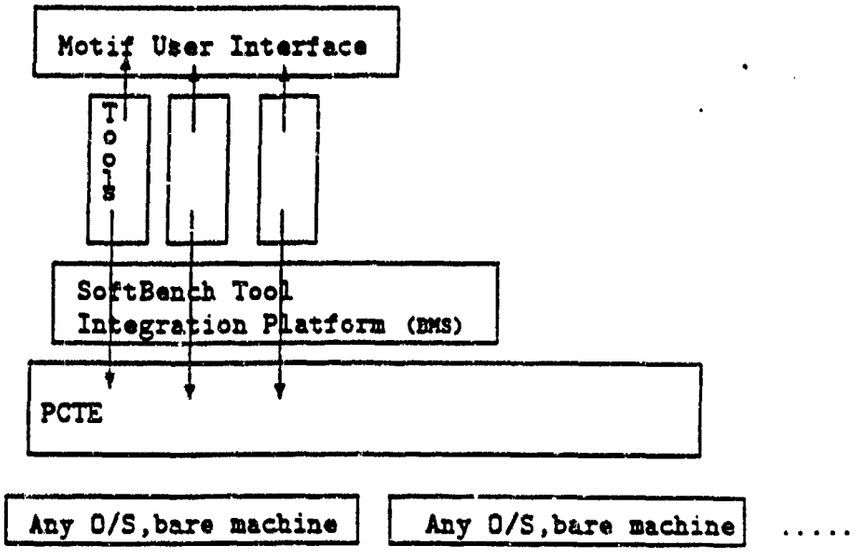


Figure 2: Prototype Architecture

UNIX[1] interfaces. These services are also at a higher level of abstraction than, say, socket based communication primitives.

The hardware architecture for a PCTE system is a network of bitmapped workstations connected by a high speed reliable LAN. PCTE is a distributed architecture, and all the object management and process management facilities are transparently distributed.

4 Prototyping Experience

In this section, we report on our intermediate results from building a prototype SEE framework.

4.1 Architecture

The architecture of the prototype is shown in figure 2. Because PCTE provides a complete interface for the tool writer and because the BMS control integration serives are at a higher level than the PCTE facilities, we have re-implemented the BMS on top of PCTE.

Each of the boxes shows one of the existing components from which we constructed the prototype. The arrows from the tools show which services were accessed by the tools. Thus the tools are linked in with and make calls to:

1. the Motif X Window libraries;
2. the BMS component of the SoftBench libraries;
3. the PCTE libraries.

[1]UNIX is a registered trademark of UNIX System Laboratories Inc.

It can be seen from the architecture that the SoftBench Tool Integration Platform only provides the BMS services. We are investigating extending this so that tools only access the PCTE services through this intermediate layer. This has the advantages of protecting the system from changes in successive PCTE versions and minimising the task of providing data integration in some way other than through the PCTE object base. It would also mean that existing SoftBench tools could be ported with a minimum of effort to the combined SoftBench/PCTE framework.

4.2 Description of the prototype

We are using the GIE Emeraude implementation of the PCTE 1.5 specifications known as Emeraude v12. It is a complete implementation of the PCTE 1.5 interfaces with additional Common Services (e.g. Metabase, Version Management).

PCTE's claim to provide a portability platform was verified by us when we ported several thousand lines of source code between workstations of different hardware from different manufacturers.

Figure 3 shows some of the elements of the prototype. The top box represents the BMS; the boxes in the second row represent class managers; the boxes in the bottom row represent instances of tools. All communication is via the BMS. We now describe these elements in more detail.

4.2.1 The BMS

The BMS runs as an PCTE process. The BMS communicates with all the tools through the PCTE inter-process communication mechanism of message queues. These replace the socket connections in the SoftBench BMS.

The BMS has an associated message queue whose whereabouts in the object base must be known by all tools. The message queue's location was (arbitrarily) chosen to be linked to the static context of the BMS (static context is the PCTE term for 'program'). Essentially the BMS maintains a 'pattern map' which is a map from tool identifiers to the set of message-patterns in which those tools have registered interest. It continuously reads from its message queue, suspending execution until a message arrives. The message will be forwarded to any interested tools or may cause the pattern map to be updated.

4.2.2 The Class Managers

Every tool belongs to a class. Each class defines the functionality which tools of that class will provide to other tools. This functionality is accessed by sending request messages to the tool. There is a class manager for each class. The class manager maintains a list of the running tools of its class and carries knowledge of whether there is a tool able to service any given request or whether a new tool needs to be started.

Each class manager runs as a PCTE process. They each have an associated message queue linked to their static context. Each class manager continuously reads from its message queue, suspending execution until a message arrives. Any request message will be forwarded to whichever tool is able to service it. All class managers are very similar except for the knowledge about when new tools should

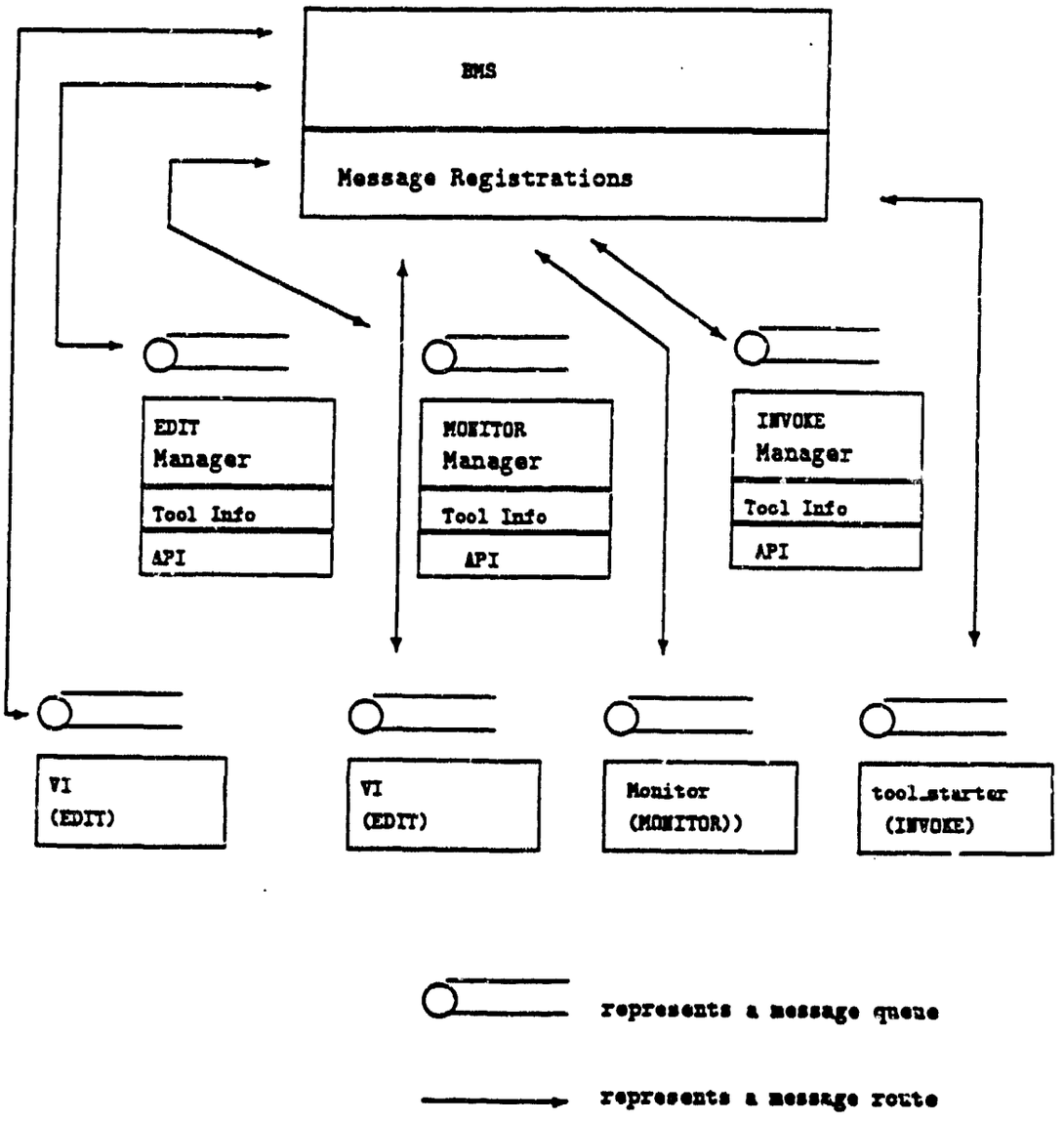


Figure 3: Current Prototype

be started up. This knowledge is more complicated in PCTE where objects do not have unique pathnames and where the context of a tool includes the working schema of that tool.

The amalgamation of all the class managers corresponds to the Execution Manager in SoftBench. By separating out the class manager processes we were able to make the decisions about whether to start new tools to handle requests specific to the class of the tool. In SoftBench the Execution Manager used the UNIX execution primitives. In our prototype these have been replaced with the PCTE execution primitives.

4.2.3 Tools

A number of simple tools have been put together for this prototype:

- The *INVOKE* tool corresponds to the SoftBench Tool Manager. It allows the user to select, start and stop tools of any of the available tool classes.
- The *MONITOR* tool corresponds to the SoftBench Message Monitor. It registers an interest in all kinds of messages and displays them. It provides a window onto the BMS activities.
- The *DM* tool corresponds to the SoftBench Development Manager. While the SoftBench development manager gives an interface to the UNIX file system, the DM tool gives a similar interface to the PCTE object base. This enables us to navigate around the object base. The tool includes some version management facilities using the common services provided with the Emeraude product.
- the DMGRAPH tool is a graphical interface tool to the PCTE object base. It navigates the object base via mouse selection of objects, displays the object graph to a user chosen depth, reorientates and manipulates the graphical representation and dynamically manipulates working schemas to provide views on the object base.
- The *EDIT* tool is for editing the contents of objects.

Each tool, like the class managers, runs as a PCTE process. They each have an associated message queue linked to their static context. Each tool continuously reads from its message queue (not suspending execution) until a message arrives. Any request message will be serviced in a tool specific way.

4.2.4 Additional PCTE features of interest

A PCTE installation will typically be distributed over a set of workstations connected by a local area network. The transparent distribution facilities provided by PCTE meant that we did not have to concern ourselves with distribution when designing the prototype. We believe that the SoftBench distribution facilities can be provided on top of PCTE with the added advantage of location transparent access to data.

ECMA PCTE implementations will provide more services than PCTE 1.5. One such service is the ability to respond to events such as access to particular objects in the object base. Adding such services to our existing control integrations services are of much interest and will provide further research directions.

We have not used the PCTE support for concurrency and integrity control and activities. We have not heavily used the schema management facilities.

5 Summary of What We have Learnt

Several points came out of our construction work with respect to PCTE:

- we found the PCTE interface useful in building the BMS and the prototype tools. It provided all the facilities we needed and many of the services were at a higher level than that provided by the operating system.
- PCTE is an effective portability platform;
- we found object identification somewhat confusing, having to switch between pathnames, internal references, external references and volume number, object number pairs. A clear notion of object surrogate would have simplified our task.
- documentation is needed to guide the tool writer through the many design decisions he needs to make. This should include a guide for data integration (how to use the schemas provided and how to write new ones, etc.), and a guide for control integration (how to use the interfaces exported by existing tools and what new message interface a tool should provide, etc.) ;
- a clear and well documented migration path from existing toolsets will be needed;
- The distribution facilities provided by PCTE meant that we did not have to concern ourselves with distribution issues when designing the prototype.

There are many software architecture decisions which should be made by tool writers even if PCTE is not used as the basis for the support environment (such as the production of appropriate schemas and the use of integrating service libraries which hide the underlying technology). These are generally good engineering practices but will protect investment in tools and will ease the transition to PCTE.

The prototyping work at HP Laboratories has proven the feasibility of adding a BMS to PCTE. We are starting a new prototyping phase to experiment with rehosting the SoftBench environment on PCTE.

References

[1] A. Earl. A Reference Model for Computer Assisted Software Engineering Environment Frameworks. Technical Report ECMA/TR/90/55, ECMA, 17 August 1990.

[2] M. R. Cagan. The HP SoftBench Environment: An Architecture for a New Generation of Software Tools. *Hewlett-Packard Journal*, 41(3):36–47, June 1990.

[3] A. O. Deininger and C. V. Fernandez. Making Computer Behaviour Consistent: The OSF/Motif Graphical User Interface. *Hewlett-Packard Journal*, 41(3):6–26, June 1990.

[4] C. Gerety. A New Generation of Software Development Tools. *Hewlett-Packard Journal*, 41(3):48–58, June 1990.

[5] F.Gallo G.Boudier and I.Thomas. Overview of PCTE and PCTE+. *ACM SIGPLAN Notices*, 24(2), February 1989.

[6] I.Thomas. PCTE Interfaces: Supporting Tools in Software Engineering Environments. *IEEE Software*, November 1989.

[7] Brussels Commission of the European Communities. PCTE Functional Specifications, Version 1.5, November 1988.

[8] Software Sciences Ltd. GIE Emeraude, Selenia Industrie Elettroniche Associate. PCTE+ Functional Specification, Issue 3, October 1988.

[9] Portable Common Tool Environment - Abstract Specification. Technical Report ECMA Standard 149, ECMA, Geneva, December 1990.

MIF Control Integration Reactive Integration (Implicit Invocation)

Goal

- Facilitate Evolution
  - Behavior Based
    - What component does
      - Its External Interface
    - Not its structure
      - No code reading
  - Consumer Based
    - New component specifies:
      - Activation Criteria
      - Response to that activation
    - Unmodified Components
- Cleaner Program Structure
  - Better Modularization
    - User Interface Separation
      - Multiple Views
      - Dynamic Update
  - Stronger Semantic Models
    - Constraints
      - Repairing
    - Derivation
      - Maintained
        - Cached
        - Dynamically Updated

Reactive Integration Key Ideas

Separate *notification*
from *detection*

Layer notifications on top of MIF interoperability

i.e. Data passing & control primitives

Admit a wide variety of detection mechanisms

Establish a *Dispatcher*

as an active agent
which couples modules
that register their interest in particular activities
with the modules that perform those activities.

Reactive Integration

Detection Mechanisms

Database Triggers	FSD, Workshop
Wrappers	
ADT	Chiron, OO
Active Values	Loops
Surrogates	
Objects	Zeitgeist
Modules	Triton
Intermodule Calls	MIF
Advised Functions	
Interpreters	
Attached Functions	Amadeus
Explicit Announcement	Field, Softbench

User Interface Dynamic Update

Abstract Interface

	Proposed	**Chiron**	**FSD**
Registration		**(with Dispatcher)**	
Interest		*Instance*	{*Relation*}
Instance ⊗ {*Operation* ∨ *Relation*} ∨ Event			
Notification Filter/Extractor			
Transaction Buffering Policy			
Notification		**(to Artist)**	
Filter Function		*None* (in Artist)	*None* (in Artist)
Extractor Function		*Op* + Parms + *Results*	*No Data* (2-State *Queries*)
Transaction Buffering		None	Combined
None			
Rollback			
Delayed			
Combined			

User Interface Dynamic Update

Abstract Interface

Proposed	**Chiron**	**FSD**
Display Rendering		
Buffering Policy	*None*	*SCS*
Clock		
Progress		
System Control State		
Invocation Policy	*None*	*None*
Policy Interpreter		
Deregistration		
Artist Shutdown	*None*	*Shut-down*
Composition (Intermediate Dispatcher)		
Temporary Registrar	*None*	*None*
Pass-through Registration		
Register-For-Artist parameter		

Reactive Integration Open Issues

Composability

Scalability

Managing Computational Support

- **Filtering**
- **Buffering**

Reactive Integration Plans

Develop Interface Specs

Construct Implementation

Integrate Diverse RIs

- FSD (including Forms Kit)
- Chiron
- Artifacts
- Triton

MIF Architecture

Software Bus

- Interface Description
  - Ports
- Interconnection
  - Channels
  - Configuration
- Communication
  - Operations
  - Values
  - Exceptions

MIF Architecture Components

Interface Description

Ports

Entry Points

Name

Signature

Stream

Signature

MIF Architecture Components

Interconnection

- Channels
  - Point to Point
  - 1-N
  - N-1
  - N-N (Broadcast)
- Configuration
  - Component identification
    - Name/version
  - Port/Channel Binding
  - Establishment
    - Static
    - Dynamic
      - Component startup
      - Component restart

MIF Architecture Components

Communication

- Operations
  - Procedure Invocation
    - Synchronous
    - Asynchronous
  - Read/Write
    - Buffered/Unbuffered
- Value Passing
  - Strategies
    - Interchange intermediary
      - Convert into and outof
    - Common Underlying Semantics (KCPL)
    - Shared Constructors/Destructors
    - Imposed Representation (Scorpion)
  - Remote Object Handles
    - Handle Lifetime
      - Session
      - Persistent
    - Handle Scope
      - Single Client
      - Global

Module Interconnect Formalism (MIF)

Software Bus Abstraction

- Common Interfacing Target
  - Replaces Bilateral Interfaces
- Run-time Agent
  - Effects Communication
  - Assists Execution

Communication

- Move Typed Values between Modules
  - Currently only Point-to-Point
  - Encoded Transmission
- Identify Operation to Perform on those Values
- Identify Target Module

Configuration Management (Inter-Module Binding)

- Static:
  - Fixed Configuration of Modules
  - Startup & Termination
- Dynamic:
  - Time Varying Interconnections
  - Server Connections
  - Server Registration
    - Identification by Name
      - Description

Reference Model

1. A *type model* to specify values communicated between modules.

2. A set of *control paradigms*.

3. A set of *communication promises* that the bus will ensure.

4. Linguistic means to define a module's *interface*.

5. Linguistic means to define the *representation* of values of each type that the bus may transfer.

6. Linguistic means to specify valid *communication patterns*; such a specification is termed a *configuration*.

7. *Language bindings* that map each language type into a bus representation and language control constructs into bus control paradigms; each such language is termed a *target language*.

8. A procedural means to cause *transfer of data/control*.

9. A procedural means for a module to *determine* the configuration and module *interfaces*.

10. A collection of analysis, display, and generational *tools* and runtime libraries.

Information Resource Dictionary Systems

Data Dictionary Systems have existed in the data processing industry for a number of years. These systems have generally been used to capture and store definitions of data and files. In recent years the scope of dictionaries has expended to include a variety of other information resources. As a result, these systems are now called Information Resource Dictionary Systems rather than Data Dictionary Systems to reflect their expanded scope. However, the reason for implementing these systems remains the same:

- Provide effective control over information resources;
- Assure adequate documentation;
- Provide standards for achieving consistency across applications; and
- Provide configuration management and traceability over time.

For example, an IRDS can help achieve standardization when gathering requirements for a new system which is to be built. The IRDS can provide the vocabulary to be used between the systems analyst and the end-user. Adherence to a commonly understood and generally agreed upon terminology will contribute substantially to assuring that the statement of requirements reflects the true needs of the end-user and the enterprise. Equally, for data elements which are shared by different users and perhaps different organizational components, availability of commonly agreed upon definitions of data elements will help in clearing up misunderstandings of terminology or, even more importantly, may prevent such misunderstandings from occurring.

An IRDS can provide a powerful tool for Information Resource Management. One of the problems that is encountered in IRM is the vast amount of data about information resources that is required to manage them, together with the very complex and numerous relationships that exist among them. This is precisely the sort of task that an IRDS can be made to do, provided that it has been conditioned to know how to deal not just with data entities or process entities, but with the entire gamut of information resource entities.

In order to accomplish these broader objectives, facilities will be needed that go well beyond those available in existing data dictionary systems. These include:

- Extensibility,
- Flexibility,
- Version Management, and
- Life cycle Phase control.

Extensibility

When dictionaries focused only on data, most installations could operate using the same dictionary structure. However, as the role of the dictionary system expands, it is more and more likely that different installations will want to store different information. Accordingly, IRDS's must allow each installation to extend the types of information that are stored in the IRDS.

Extended types defined by the user must be subject to the same controls as any types that are defined by the dictionary vendor. All facilities of the dictionary must apply equally to vendor supplied data and the extensions. There can be no distinction.

The architecture of an IRDS must be designed form the outset to be extensible. Not surprisingly, this was a major goal of the ANSI and FIPS IRDS standard.

Flexibility

An IRDS must be suitable for a wide variety of uses. The interfaces to the IRDS must be flexible enough to deal with the extended information types. Users must also be able to access IRDS data on their own. However, the controls inherent in the IRDS must be maintained.

Version Management

An IRDS must be able to maintain and manage multiple versions of an entity in a consistent way. Version maintenance cannot be left to each application or user, but must be an integral part of the IRDS.

Life cycle Phase control

As entities in the dictionary move through their life cycles they should be subject to different sets of constraints. When an entity represents a component of a product that is in the early stages of development, it may be subject to fairly loose control. However, when a system or building or piece of electronic equipment goes into "production" the controls must be very tight. The IRDS needs to be able to track entities through life cycle phases and to enforce different levels of control during different phases.

IRDS Standardization

Standards committees are not formed to do basic research. That is, they are not allowed to invent something out of thin air and standardize it. Instead, the job of a standards organization is to examine existing products in a relatively mature industry and to develop a standard based on those existing products. The development of the IRDS Standard was no different.

Two design objectives for the IRDS Standard were defined at the outset. First, the IRDS Standard must contain the major features and capabilities that are provided by existing commercial dictionary systems. At the same time, it must provide the power and flexibility needed to support the next generation of Information Resource Management.

The second objective was that the standard should be modular to allow it to be implemented in a wide variety of environments. If the standards required every conceivable feature to be implemented in order for a dictionary system to be declared conformant, it might be impossible for conformant implementations of the standard to run on smaller machines. Thus, the standard defines a set of minimal functionality, called the "Core Standard" or "Module I", and a set of optional "modules" that define additional functionality that can be added by the implementor when appropriate for the particular target environment. There are currently five optional modules.

In 1980, two organizations began independently to develop a standard for dictionary systems in the U.S. The National Bureau of Standards (NBS, now renamed NIST) began an effort to develop a Federal Information Processing Standard (FIPS) that was called the "Federal Information Processing Standard for Data Dictionary Systems". The American National Standards Institute began a similar effort with the approval by the American National Standards Committee for Information Systems (X3) of what is now called the Accredited Standards Committee (ASC) X3H4. Their charter was to develop an American National Standard for "Information Resource Dictionary Systems".

These efforts continued independently until 1983 when X3H4 voted to adopt the current draft version of the FIPS standard as its base document for all subsequent work. From 1983-1988, X3H4 and NIST worked together to develop the draft proposed American National Standard for Information Resource Dictionary Systems (dpANS IRDS).

During that time, dpANS IRDS was given two formal public reviews. The first one was in 1985 and it resulted in such extensive revisions that a second public review had to be performed in 1986. Only minor revisions were required as a result of the second review and, on October 19, 1988, the dpANS IRDS became an official American National Standard. This standard covers the functionality of an IRDS and defines Panel and Command Language interfaces to the IRDS.

On April 5, 1989, the National Institute of Standards and Technology announced the FIPS standard. Other than a minor change in the conformance statement, this standard is identical to the ANSI standard. This standard became effective for U.S. Government Agencies on September 25, 1989. However, there was an additional eighteen month "transition period" during which U.S. Government Agencies could continue to get waivers to allow them to purchase dictionary systems that were not compliant with the FIPS. Thus, the FIPS did not become mandatory until March 25, 1991.

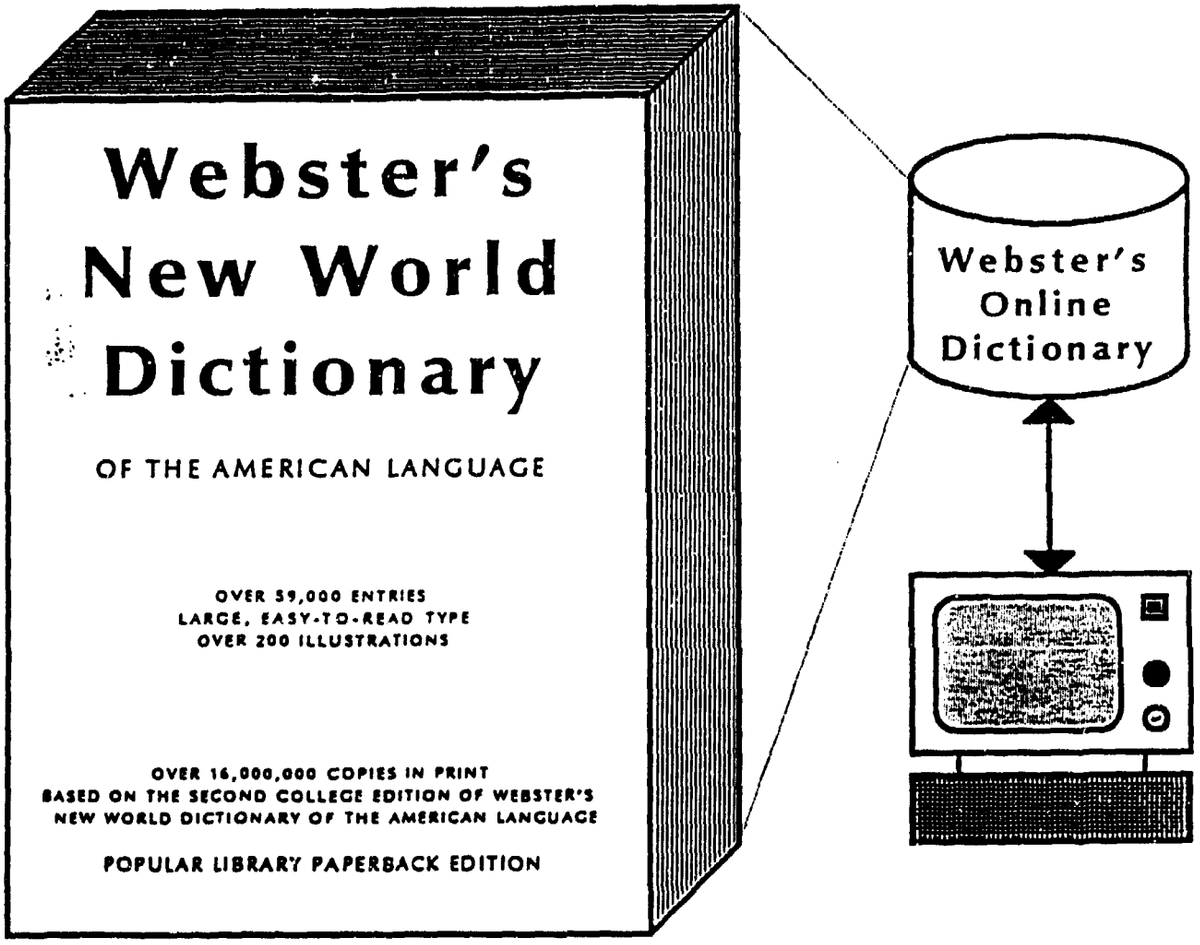
Information Resource Dictionary System

E. James Emerson
Repository Technologies, Inc.
3590 Hobson Road, Suite 204
Woodridge, IL 60517
(708) 515-0780

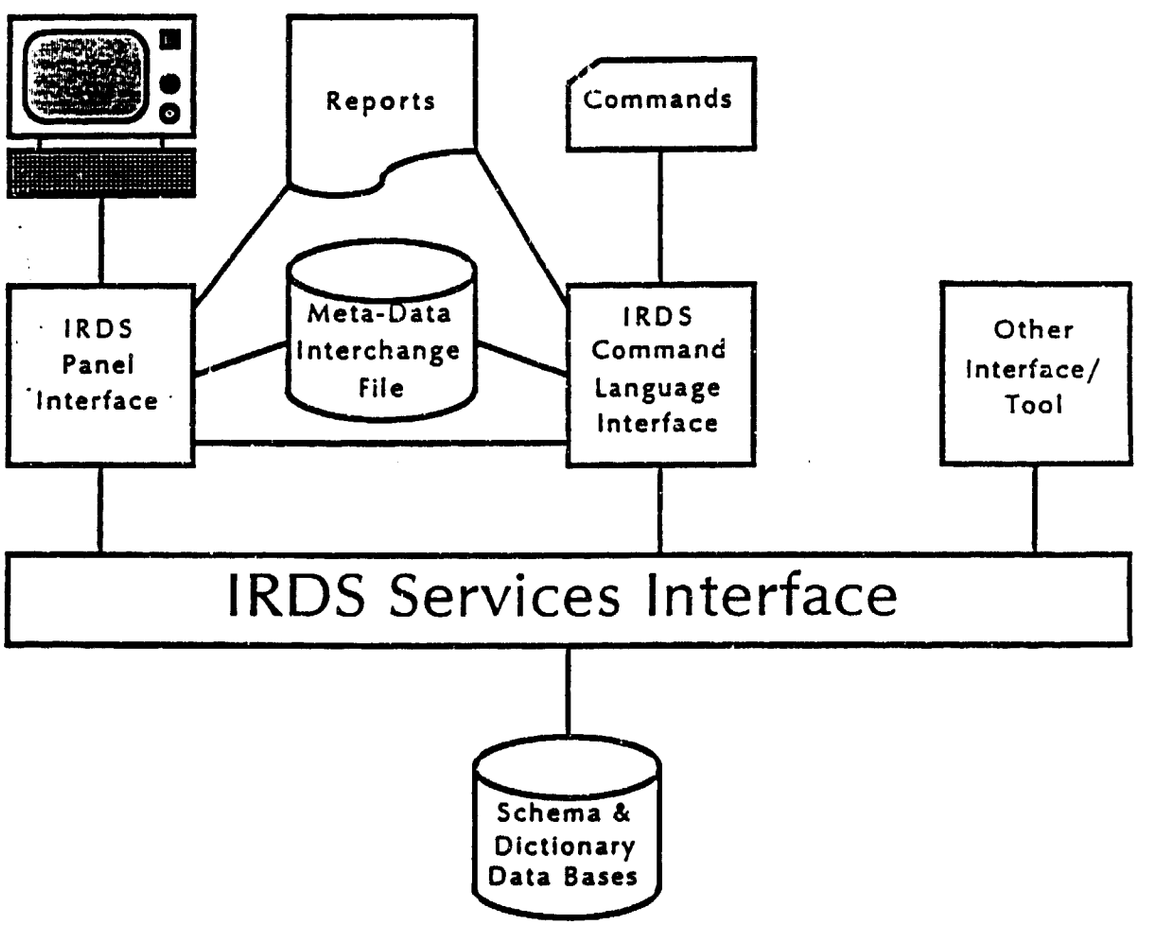
TOPICS

- Dictionary System Concepts
- History of the IRDS Standard
- Overview of the IRDS Standard

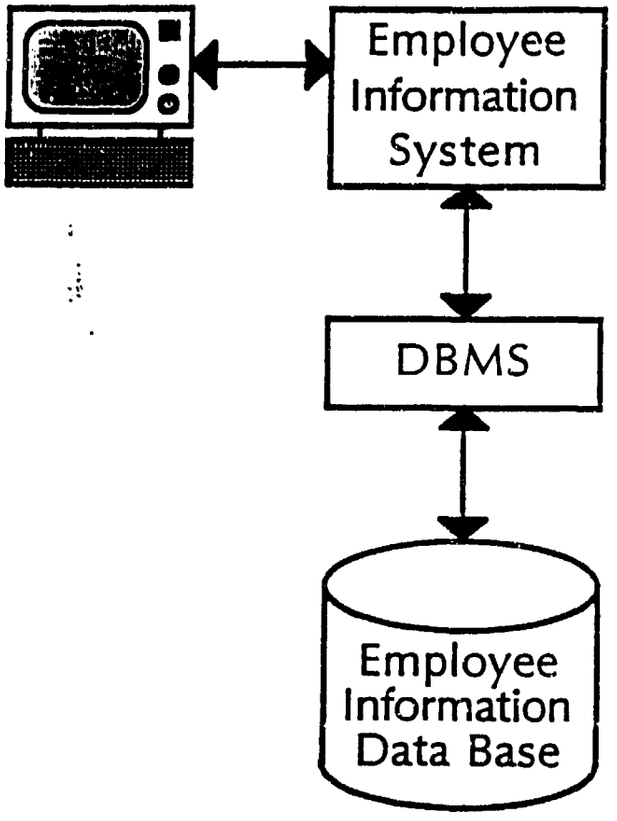
ONE TYPE OF DICTIONARY SYSTEM



ANOTHER TYPE OF DICTIONARY SYSTEM



APPLICATIONS AND DATA

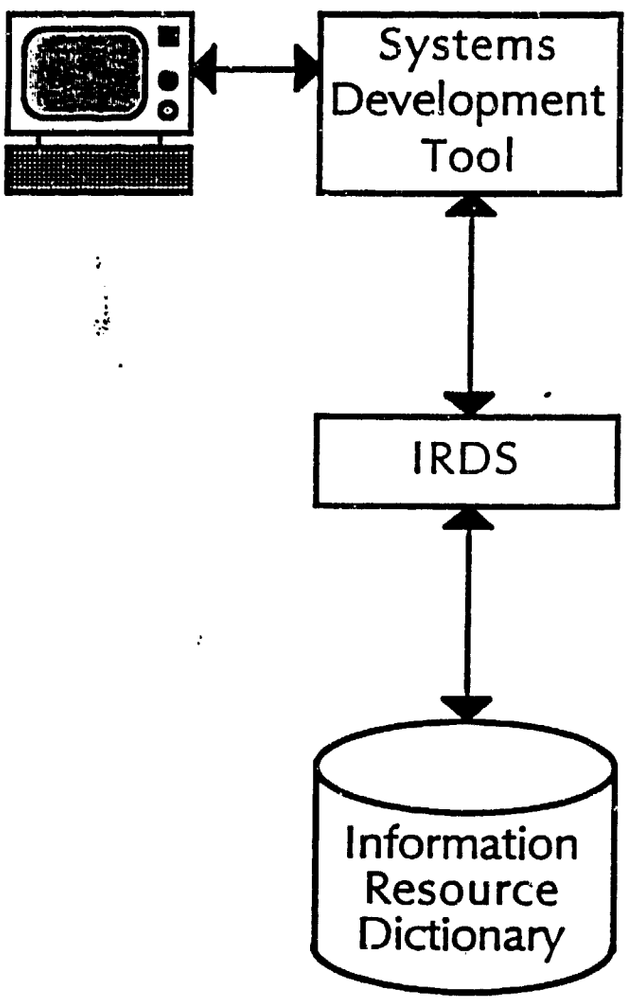


An application automates a business function for the corporation.

A DBMS manages the data that describes the corporation's business resources.

Each "record" describes a real employee (a corporate resource).

SYSTEM DEVELOPMENT TOOLS AND META-DATA



A systems development tool automates a systems development function for the data processing organization within the corporation.

An IRDS manages the meta-data that describes the corporation's information resources.

Each "record" describes a real information resource (e.g. program or data base) which are also corporate resources.

WHAT IS AN IRDS

An **IRDS** as a *special purpose* **DBMS** used by systems development and systems management tools for storage and maintenance of meta-data that describe a corporation's information resources.

WHY AN IRDS?

- Provide consistency of Meta-Data
- Facilitate reuse of data elements and programs
- Identify **all** the components that comprise an application system (Configuration Management)
- Determine the overall impact of a change to a particular component of an application system (Change Management)
- Provide a means to integrate application development and systems management tools from different vendors

HISTORY OF THE IRDS STANDARD

1980	Committee Formed
1983	Base Document Adopted
1984-1985	Vendor Workshops
1985-1986	Public Review
October 19, 1988	ANSI IRDS Standard Adopted ANSI X3.138-1988
April 5, 1989	NIST IRDS Standard Adopted FIPS PUB 156
September 25, 1989	NIST Standard Effective Date
March 25, 1991	End of Transition Period

ISO IRDS

- Base document adopted by IRDS Rapporteur Group in 1986
- Interest is in Services Interface only
- Was compliant with ANSI IRDS of a short period
- Data model changed from entity-relationship to object-association
- Attempt at cooperation delayed final approval of ANSI IRDS Standard
- IRDS Rapporteur Group is now moving towards pure SQL approach

IBM REPOSITORY MANAGER/MVS

- IBM has submitted an alternative Services Interface to ANSI and ISO
- Alternative is based on Entity-Relationship Tools Interface
- IBM may submit proposal for Object Service Tools interface in the future
- The Repository will become the de facto standard for SAA

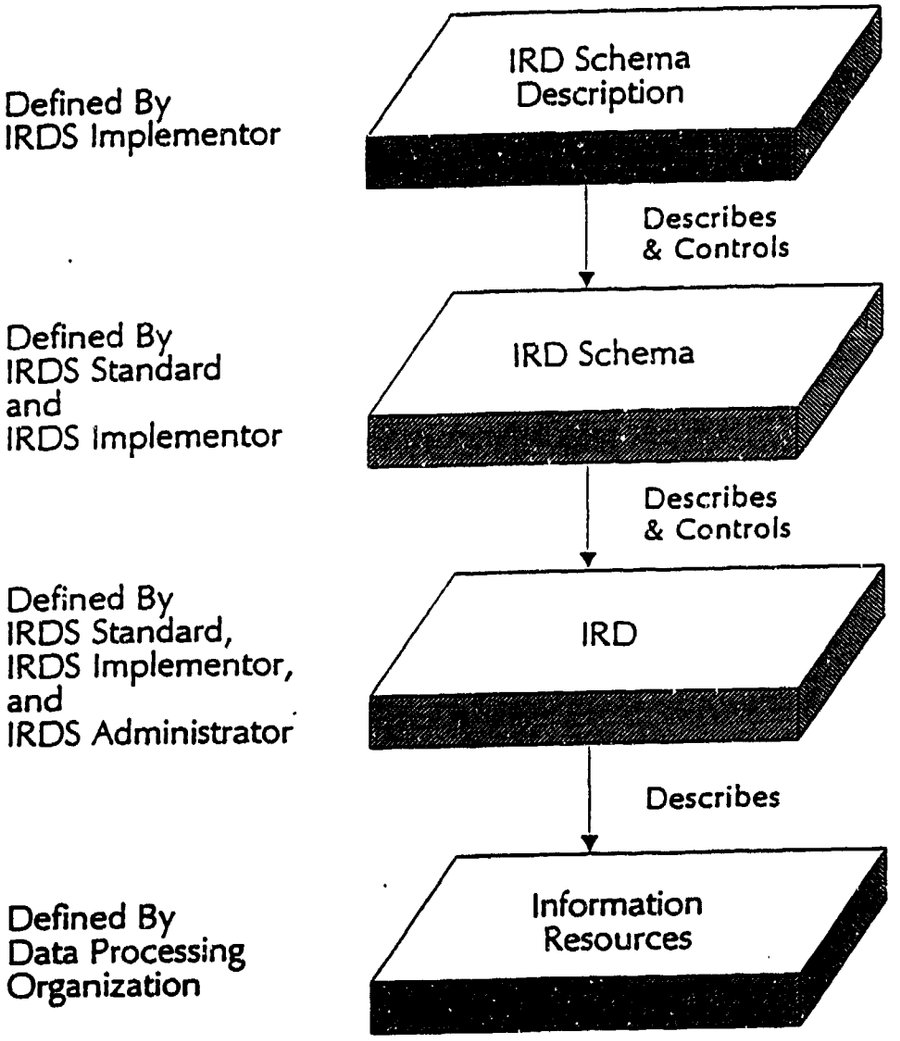
DEC ATIS

- DEC has submitted ATIS as a proposed IRDS interface to ANSI and ISO
- Within ANSI, proposal is for a next-generation standard
- Current standard is treated as a subset
- X3H4 adopted proposal as a base document for "IRDS Support of CASE"
- X3H4 recommended proposal as new work item within ISO

ANSI IRDS FEATURES

- Command Language Syntax and Semantics
- Menu-driven Panel Interface Semantics
- Fully Extensible
- Maintain Multiple Versions
- Life Cycle Phase Control
- Security
- IRD Views
- Impact of Change and Other Reports

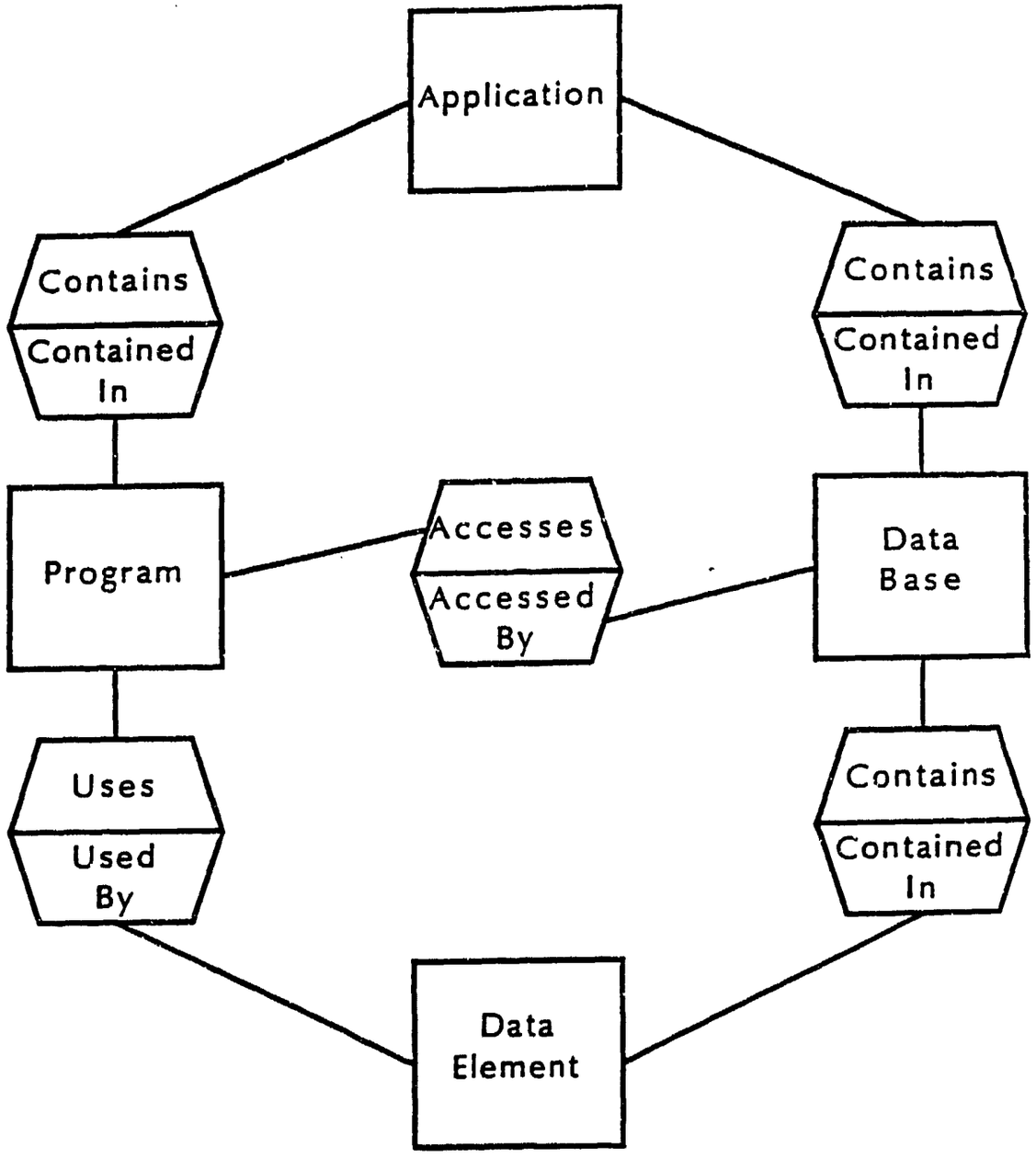
IRDS ARCHITECTURE



EXTENSIBILITY

- Standard specifies a "Starter Set" IRD Schema.
- Users are expected to extend this to satisfy their requirements.
- Extensions are defined by Entity Types, Relationship Types, and Attribute Types in the IRD Schema.
- There is no distinction between starter set and extensions. Starter set can be deleted if desired.

SOME COMPONENTS OF A TYPICAL IRDS SCHEMA



IRDS APPLICATIONS

- Maintain meta-data throughout the development life cycle, including operational phase
- Support data element standardization program
- Manage schema and sub-schema definition for a DBMS
- Support records, report, and forms management programs

FUTURE DIRECTIONS

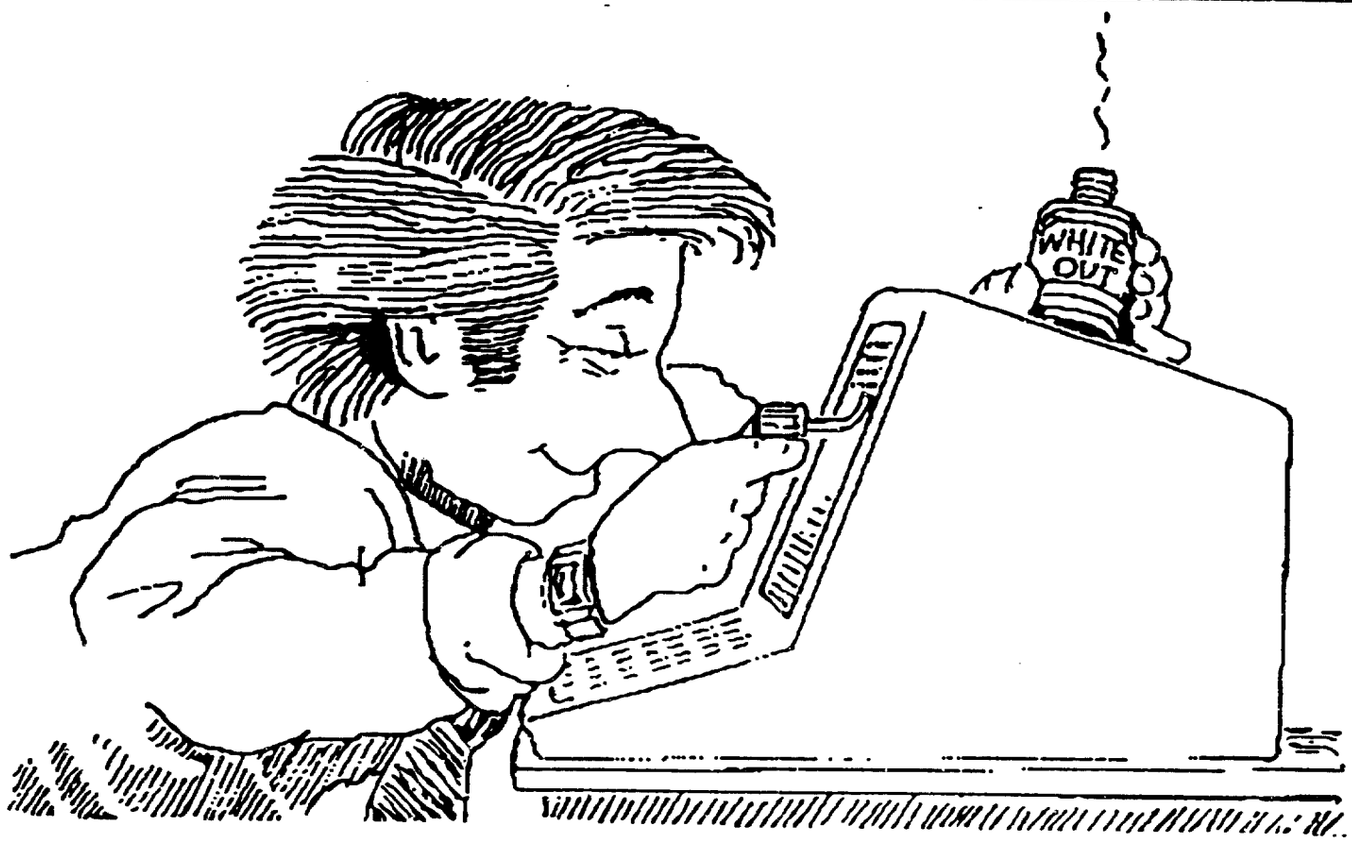
- Establish standards for the representation of meta-data in the IRDS
- Establish standard methodologies for Configuration Management and Change Management
- Vendors provide tools that maintain meta-data in the IRDS
- ANSI and/or ISO adopt a "next generation" standard

SECTION III

PCIS WORKSHOP

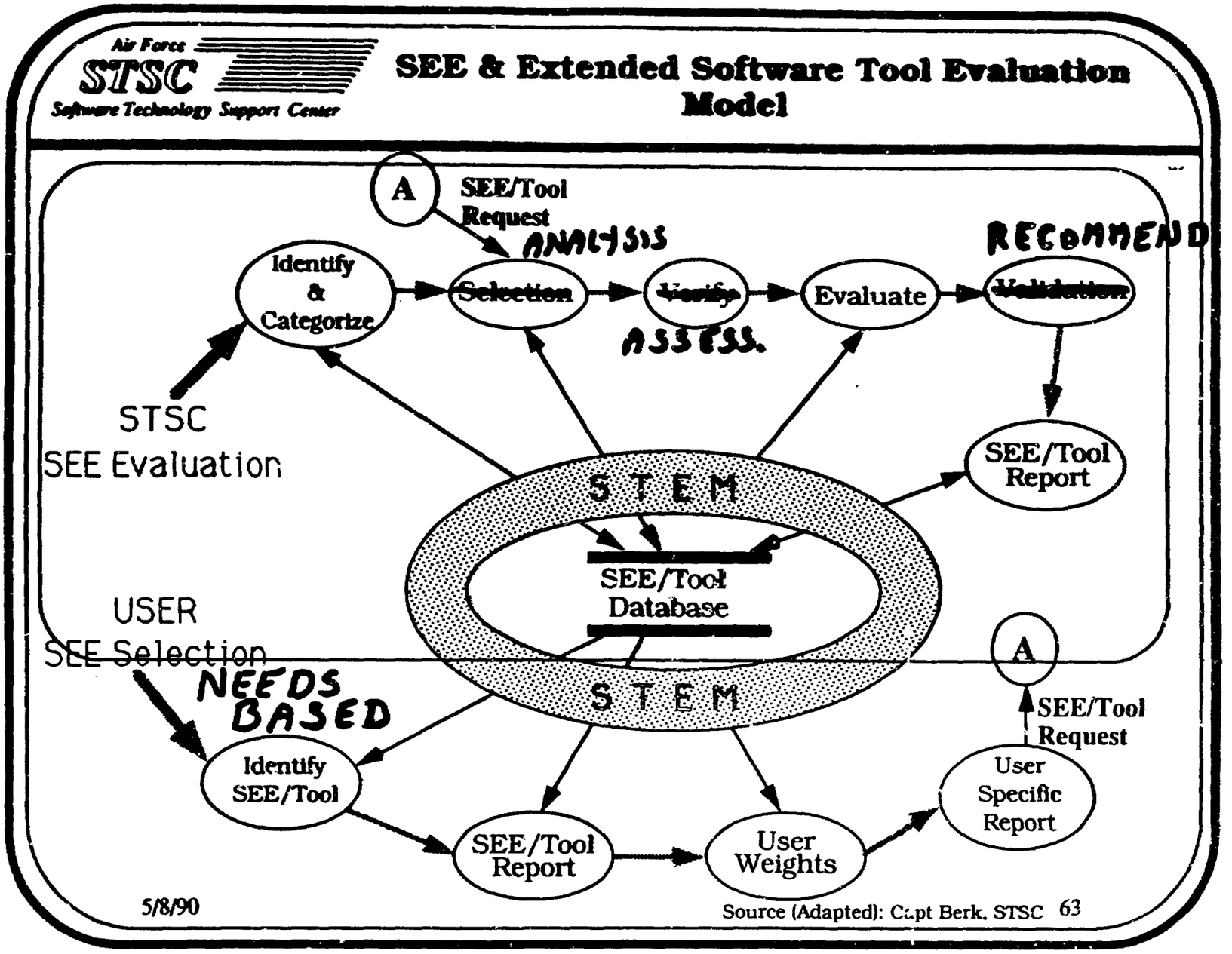
Air Force STSC
Software Technology Support Center

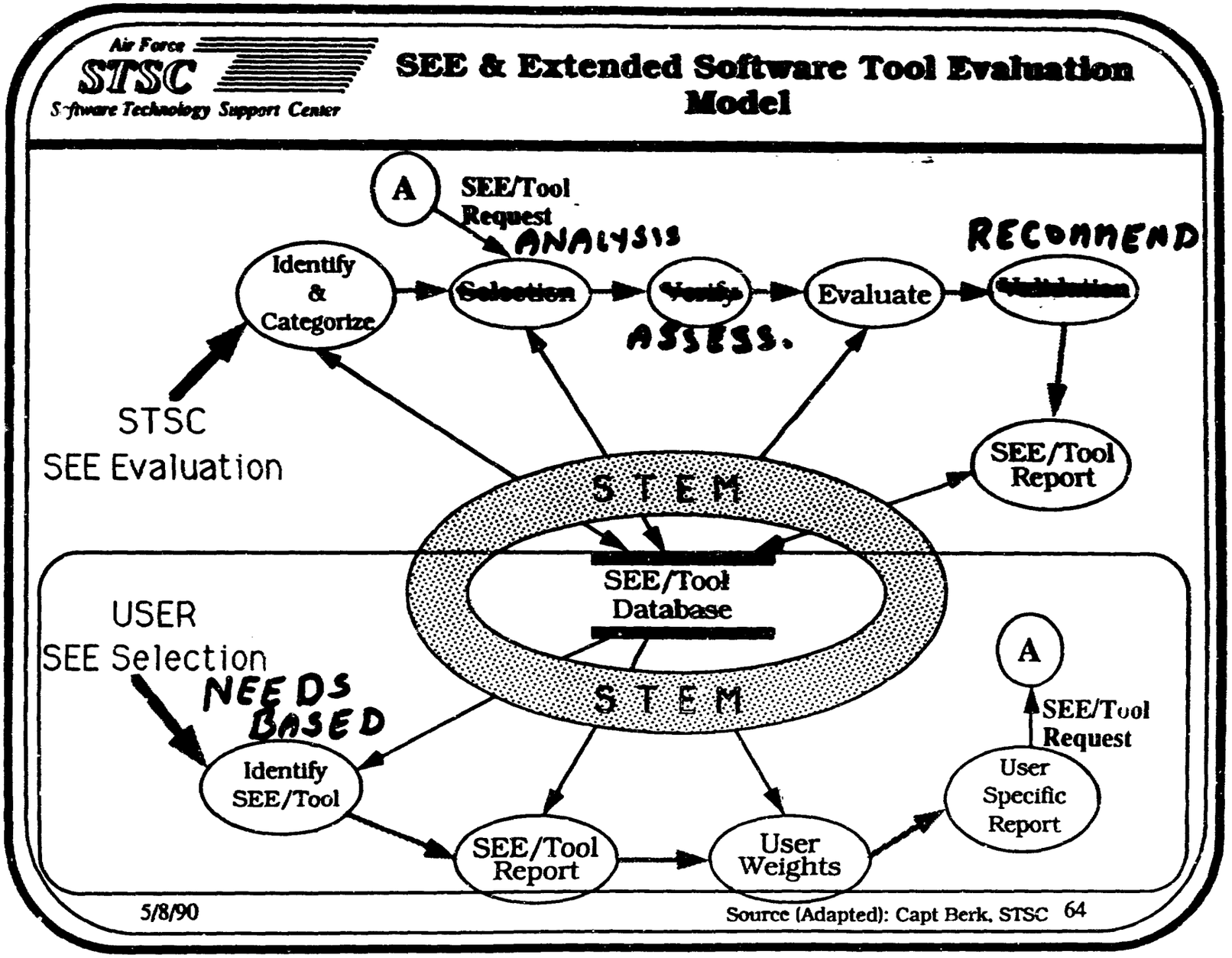
The Software Technology Support Center (STSC)



Mr. Bob Hanrahan, CSP, CDP
OO-ALC/TISAC
HILL AFB, UTAH 84056
AV 458-7703/8045
(801) 777-7703/8045
hanrahan@oodis01.af.mil

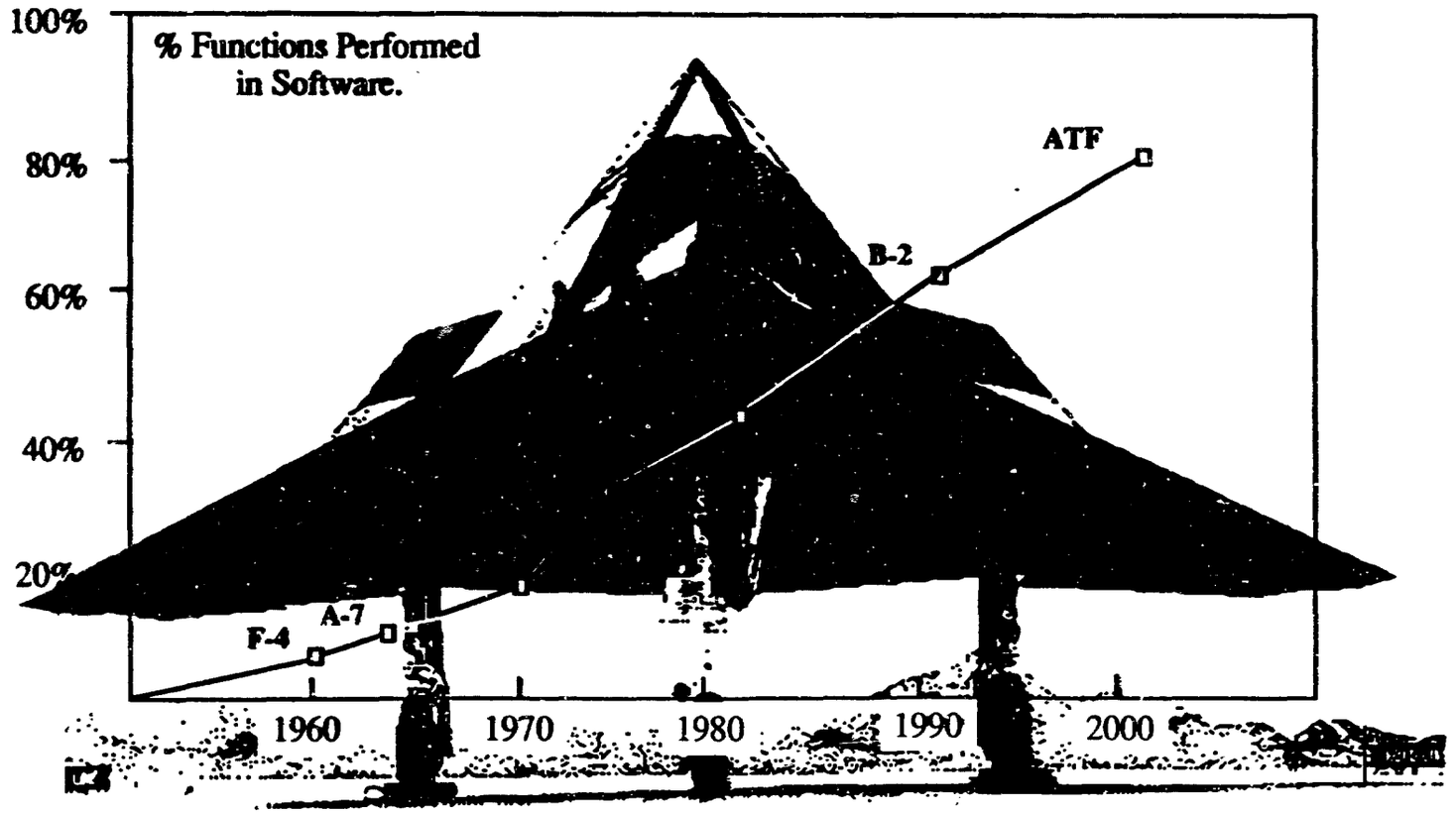
OO-ALC/TISAC [redacted]


Air Force
STSC
Software Technology Support Center
SEE & Extended Software Tool Evaluation Model
A
SEE/Tool Request
ANALYSIS
RECOMMEND
Identify & Categorize
Selection
Verify
Evaluate
Validation
ASSESS.
STSC SEE Evaluation
SEE/Tool Report
STEM
SEE/Tool Database
USER SEE Selection
NEEDS BASED
A
SEE/Tool Request
Identify SEE/Tool
SEE/Tool Report
User Weights
User Specific Report
5/8/90
Source (Adapted): Capt Berk, STSC
63

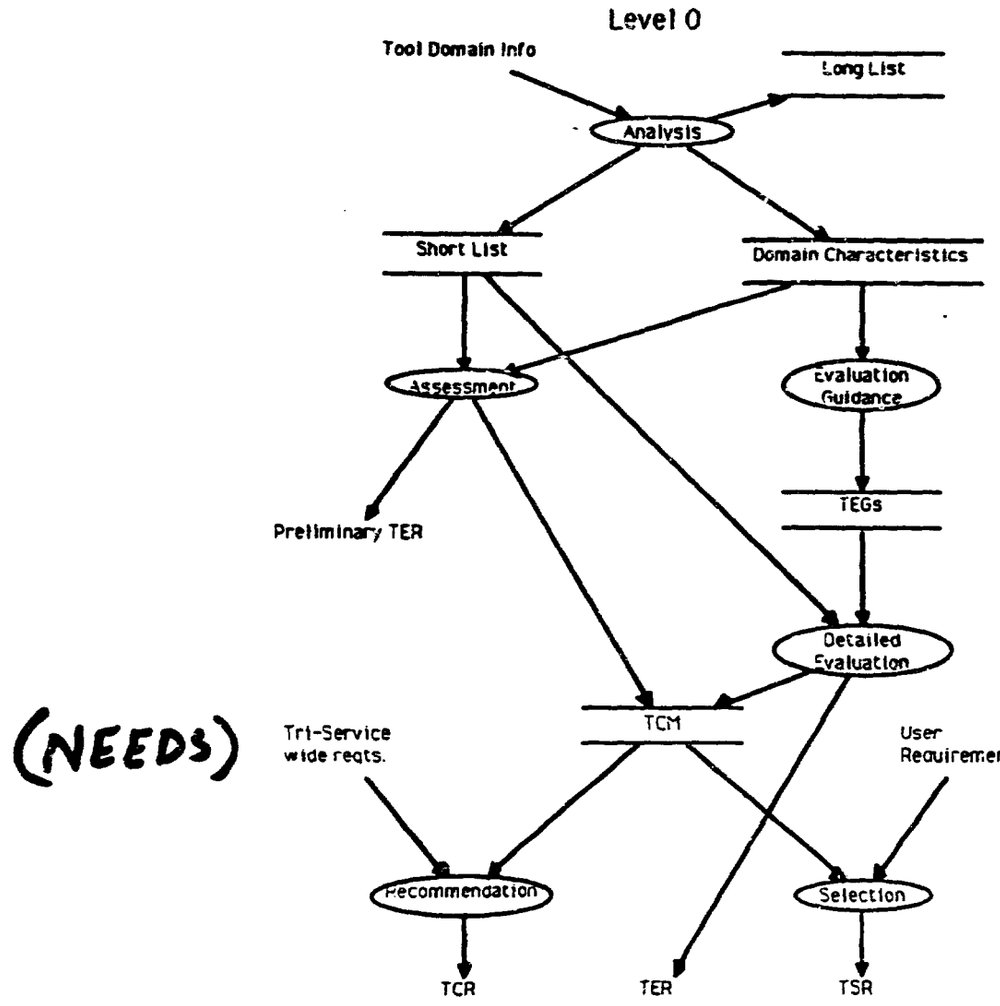

Air Force
STSC
Software Technology Support Center
SEE & Extended Software Tool Evaluation Model
A
SEE/Tool Request
ANALYSIS
RECOMMEND
Identify & Categorize
Selection
Verify
Evaluate
Validation
ASSESS.
STSC SEE Evaluation
SEE/Tool Report
STEM
SEE/Tool Database
STEM
USER SEE Selection
NEEDS BASED
A
SEE/Tool Request
Identify SEE/Tool
User Specific Report
SEE/Tool Report
User Weights
5/8/90
Source (Adapted): Capt Berk, STSC 64

Air Force
STSC
Software Technology Support Center

Weapon System Dependency



Air Force STSC
Software Technology Support Center



1

Air Force STSC Software Technology Support Center

ENVIRONMENTS TEAM MEMBERS

ROBERT (BOB) P. HANRAHAN, CSP, CDP

RON PETERSON, PhD

GREG JONES, PhD

ODEAN BOWLER

JAMES VAN BUREN

CAROL RIEPING

JOHN GROTZKY

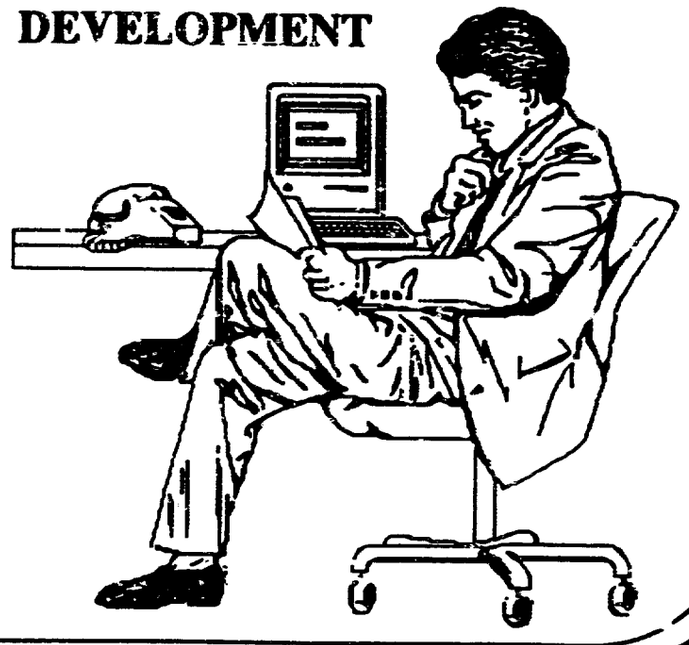
JUDI PETERSON

Air Force STSC
Software Technology Support Center

SOFTWARE ENGINEERING ENVIRONMENT (SEE) PROJECT

- **GOAL:**

DEVELOP STSC CAPABILITY FOR PROVIDING GUIDANCE ON CURRENT AND EMERGING SEE TECHNOLOGY TO SOFTWARE DEVELOPMENT AND SUPPORT ACTIVITIES



16 April 1991

Air Force STSC Software Technology Support Center

SEE PROJECT

- **OBJECTIVES**

EXTEND TEST AND EVALUATION PROCESSES

SEE "reference model" - SEE "framework"

SEE EVALUATION CAPABILITY - "mapping"

INTERIM CAPABILITY - "state of the practice"

ROADMAP - "state of the art"

"IDEAL" SEE - Fully integrated framework, process, methods, automation

518

Air Force STSC
Software Technology Support Center

SEE PROJECT

- **SCOPE**

  SEE "FRAMEWORKS"

  SEE COMPONENTS AND INTEGRATION

  PROCESS

  METHODOLOGY/METHODS

  RE-ENGINEERING (small scale efforts, as required)

Air Force STSC Software Technology Support Center

SEE PROJECT

- **ACTIVITIES**

  B-1 BOMBER Doc.

  ROME LABORATORIES - TECHNOLOGY TRANSITION PLAN

  JIAWG - TOOL EVALUATION WORKING GROUP

  - SOFTWARE ENGINEERING ENVIRONMENT DEMONSTRATION (SEED) NODE

  *. **SEE SECURITY** STEM

  F-16 A/B - PROCESS, METHODOLOGY, TOOLS

  NIST ISEE W.G.

  RETURN ON INVESTMENT (ROI) MODEL STUDY

Air Force STSC
Software Technology Support Center

Test and Evaluation Concept

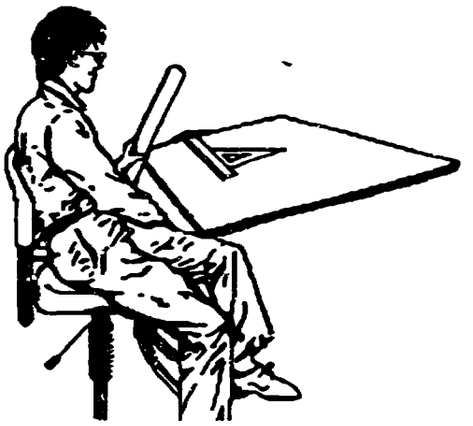
T&E Phase	STEM	Products
Analysis	*Essential Criteria* *(What is Needed?)*	Long and Short Lists of SEE "efforts"
Assessment	*Evaluation Criteria* *(What Do They Do ?)*	Paper Assesment of Short List "efforts"
Detailed Evaluation	*Test & Evaluation Procedures* *(How Well Do They Do It ?)*	Verifies Features Evaluates Quality Attributes
Recommendation	*Criteria Weighting* *(How Important is It ?)*	Comparisons of "efforts"

* MAPPING

Air Force STSC
Software Technology Support Center

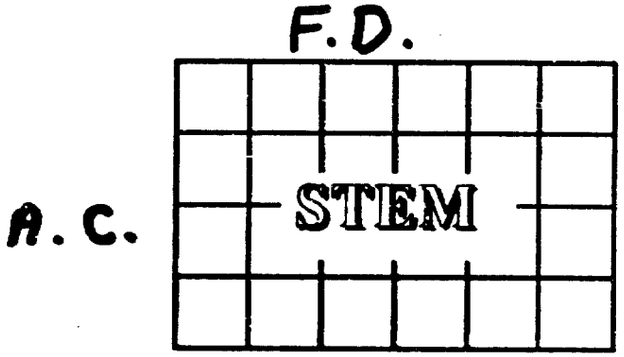
STEM
Definition

" **Specification of a model software engineering environment that is used to classify and evaluate (using criteria and validated metrics) both the environment framework and its associated software engineering tools".**



Air Force STSC
Software Technology Support Center

Software Tool Evaluation Model (STEM)



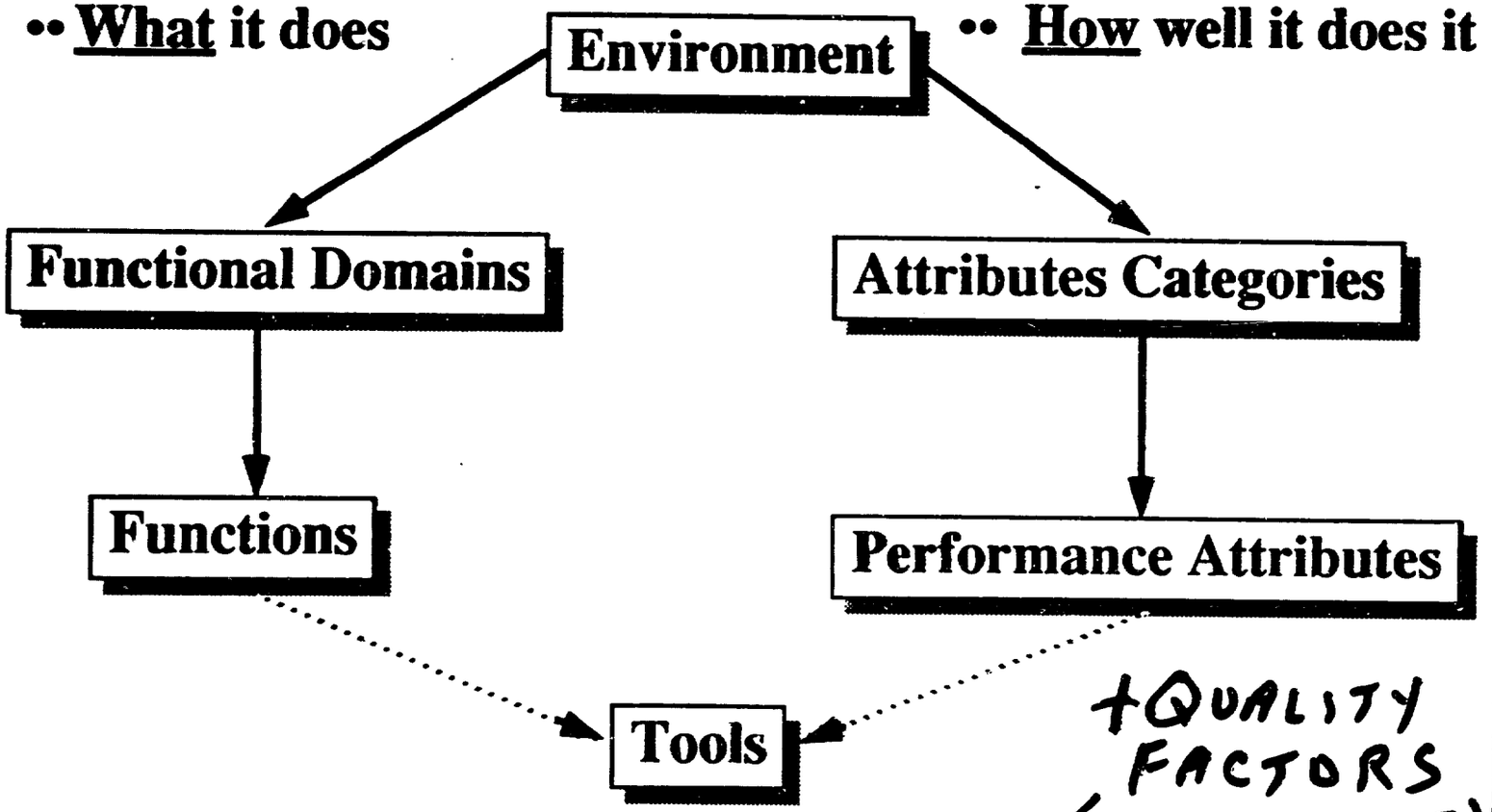
"FRAMEWORK"
YARDSTICK
TARGET ARCHITECTURE
SPECIFICATION

• ***SEE "effort" Categories***	*(What Type of "effort" is It ?)*
• ***Essential Criteria***	*(What is Needed?)*
• ***Evaluation Criteria***	*(What Do These "efforts" Do ?)*
• ***Criteria Weighting***	*(How Important are the Criteria?)*
• ***Test & Evaluation Procedures***	*(How Well Do These "efforts" Perform ?)*

Air Force STSC Software Technology Support Center

Initial STEM Structure

The software engineering environment can be subdivided two ways



Air Force STSC Software Technology Support Center

Initial STEM Structure (cont'd)

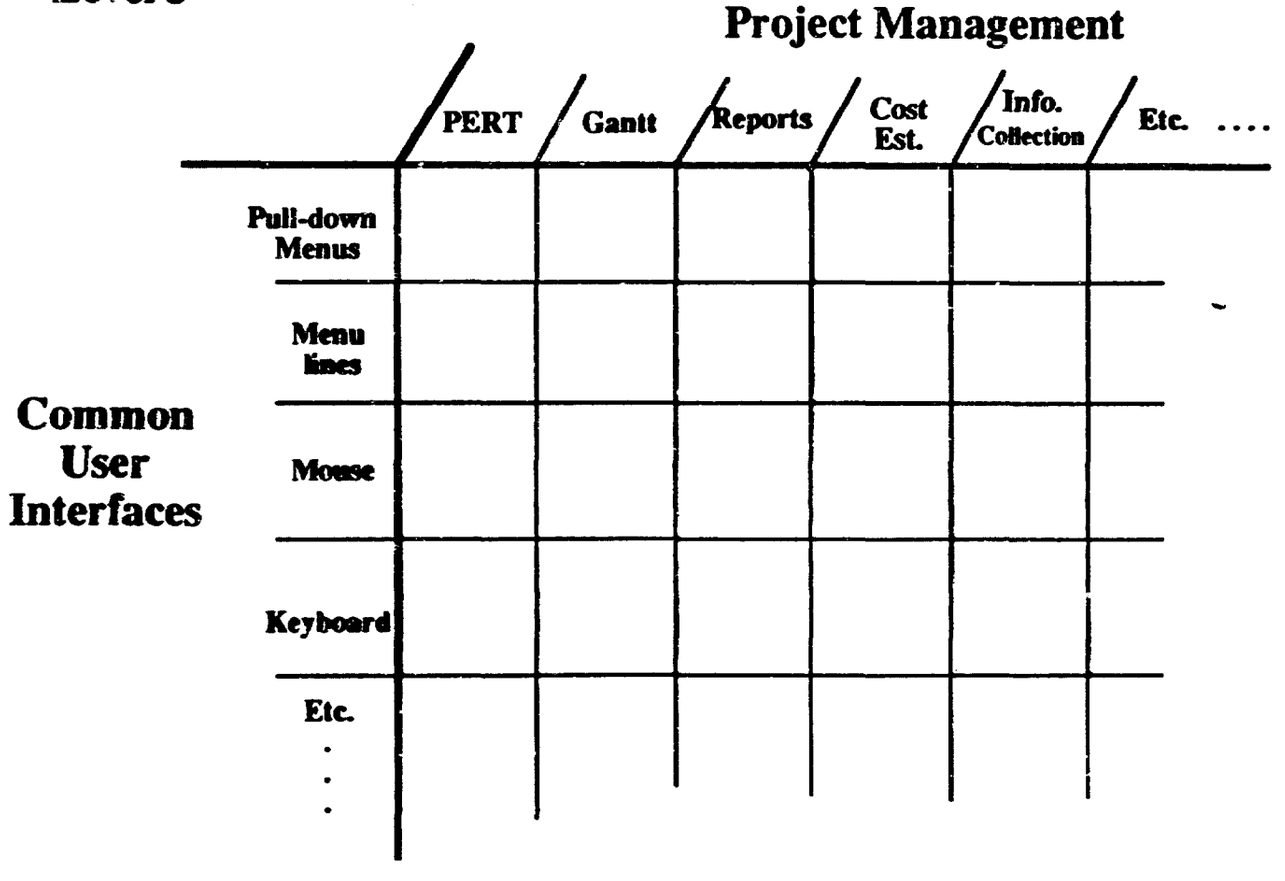
Level 2

		SW Mgmt				SW Req'ts		
		Proj Mgmt	CM	Env Mgmt	Metrics & Quality	Proto-type	Modeling	Req'ts Def.
Human Factors	Common User Interfaces							
	Availability of Training							
	Documentation for User							
	Role Division							
	On-line Help							
Mgmt Factors	Database Security							
	Metrics data gathering							
	Multi-User							
	Multi-project Support							
	Project Mgmt Support							

Air Force STSC Software Technology Support Center

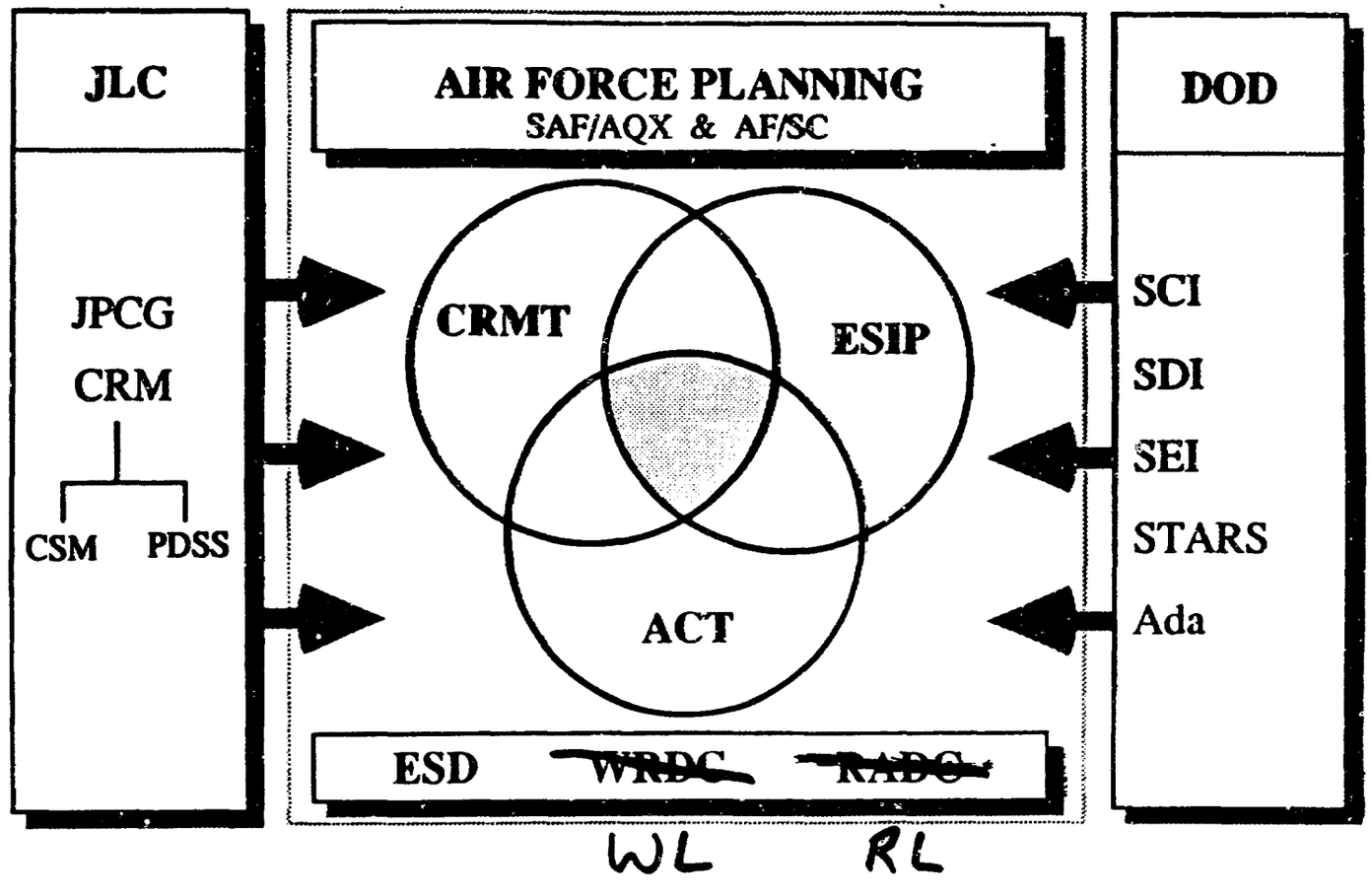
Initial STEM Structure (cont'd)

Level 3



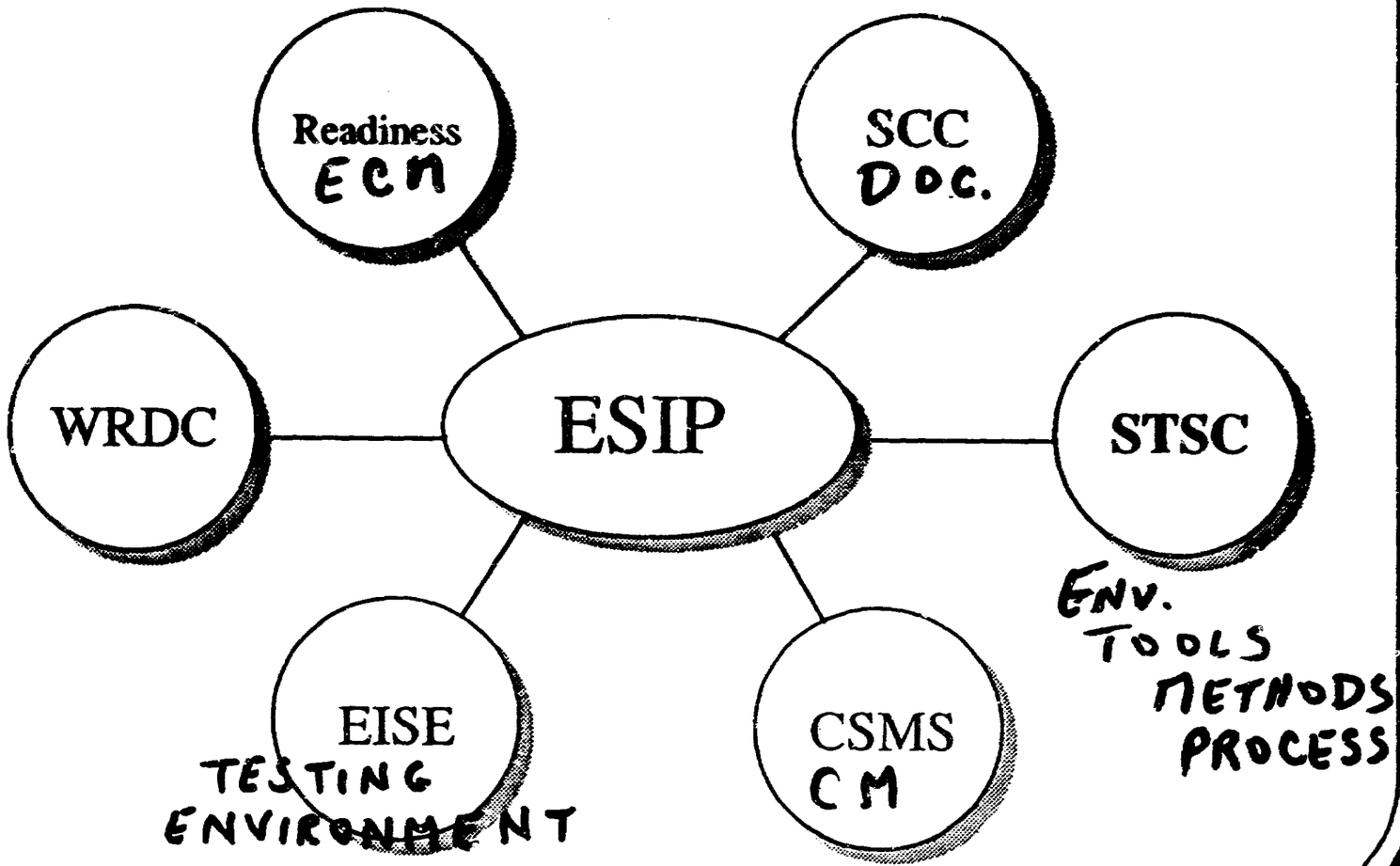
Air Force STSC Software Technology Support Center

Current Initiatives



5/8/90

ESIP


Readiness
ECN
SCC
Doc.
WRDC
ESIP
STSC
ENV.
TOOLS
METHODS
PROCESS
EISE
TESTING
ENVIRONMENT
CSMS
CM

Air Force STSC
Software Technology Support Center

Dissemination



Electronic Bulletin Board:

- **DDN:** Log into LOGDIS at OO-ALC on Hill AFB Computer and select 10.2.9.
- **Direct:** Dial (801) 777-7753 or AV 458-7753 using a modem set at 8/n/1,1200 or 2400 baud.



Newsletter (Crosstalk):

George A. Klipper
OO-ALC/TISAC — Tel: (801) 777-7703 or AV 458-7703
Hill AFB, Utah 84056 — Fax: (801) 777-8069 or AV 458-8069



Software Tool Evaluation Reports: SEE RPT OCT 91

STSC
OO-ALC/TISAC — Tel: (801) 777-7703 or AV 458-7703
Hill AFB, Utah 84056 — Fax: (801) 777-8069 or AV 458-8069

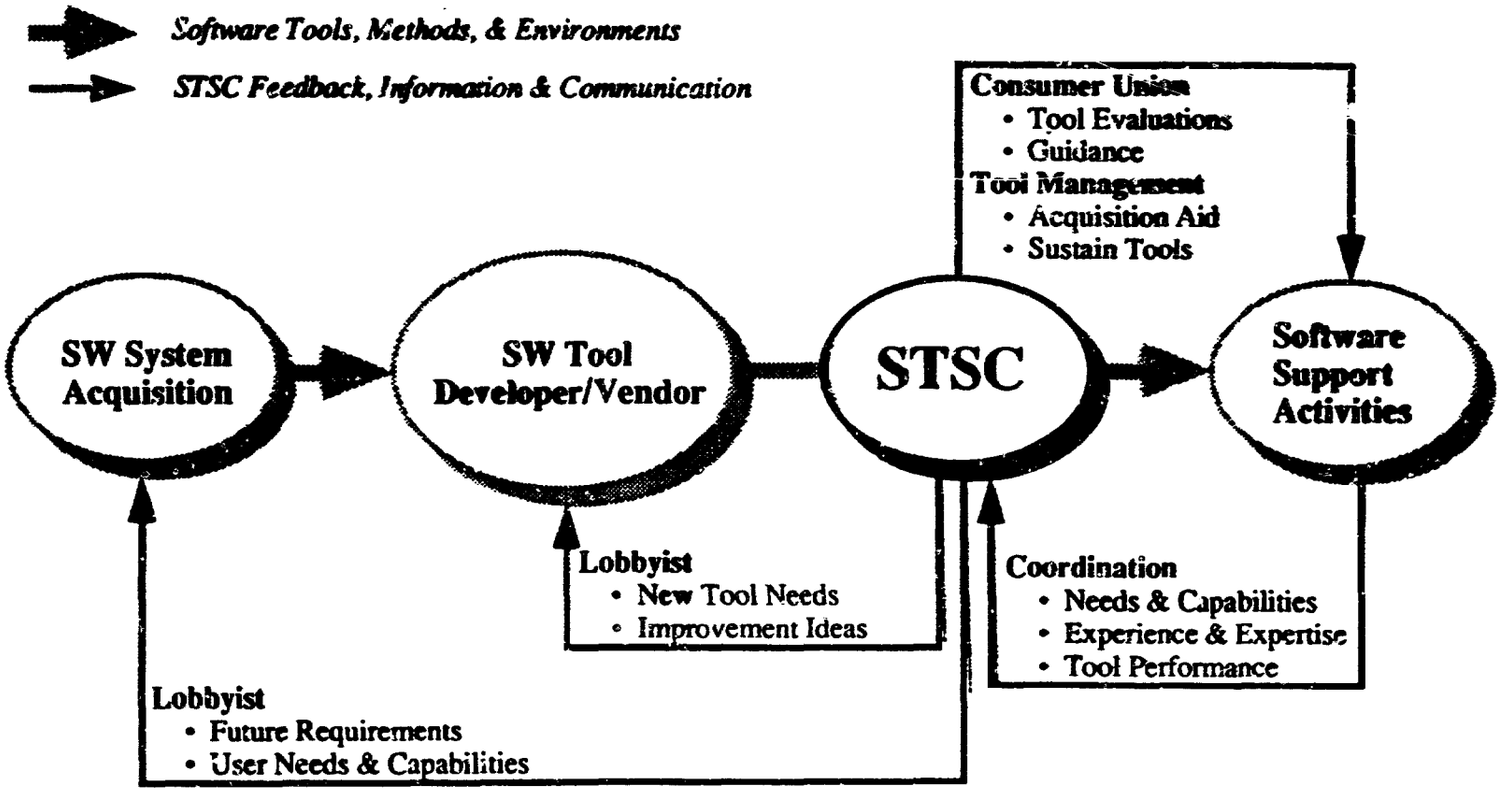


Software Conferences

- Annually (Spring). 15-19APR91
- Forum for Exchange of Ideas.

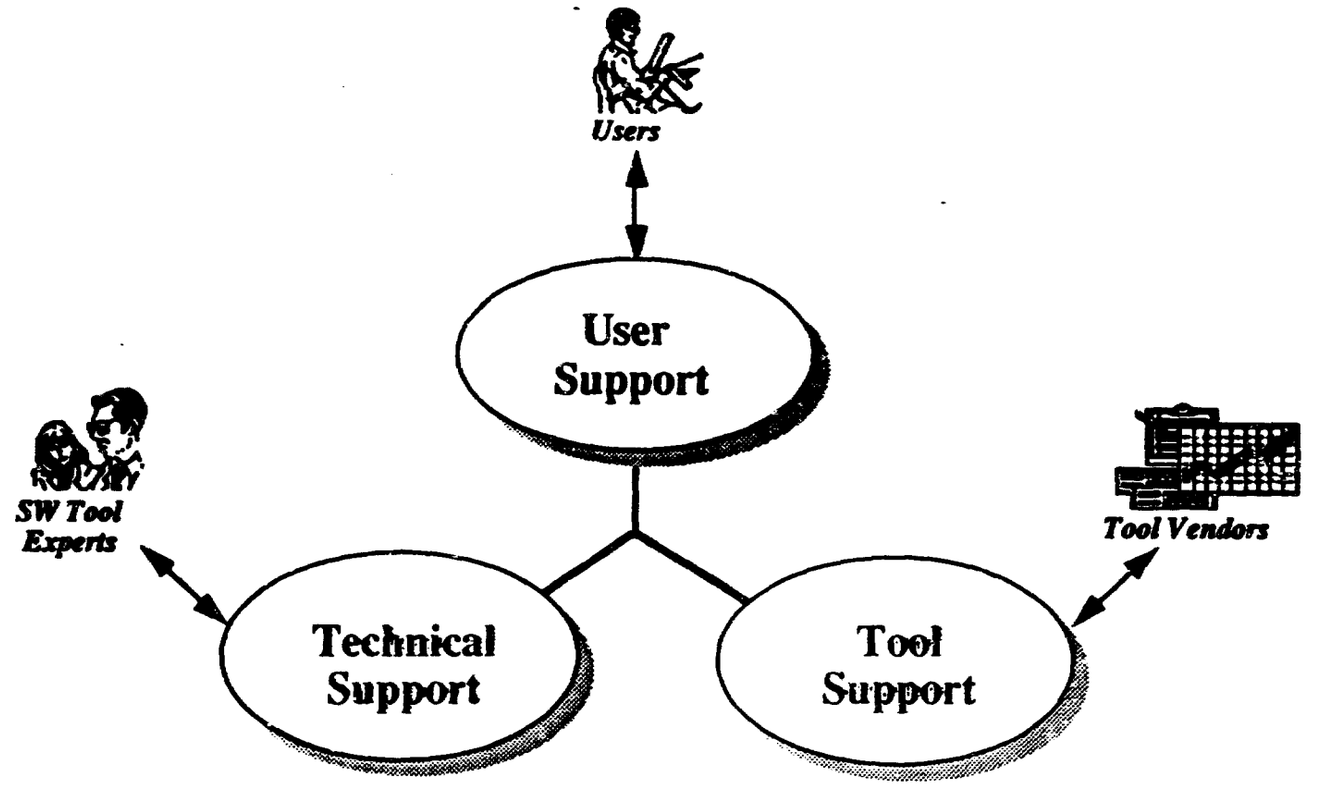
Air Force
STSC
Software Technology Support Center

STSC's Role



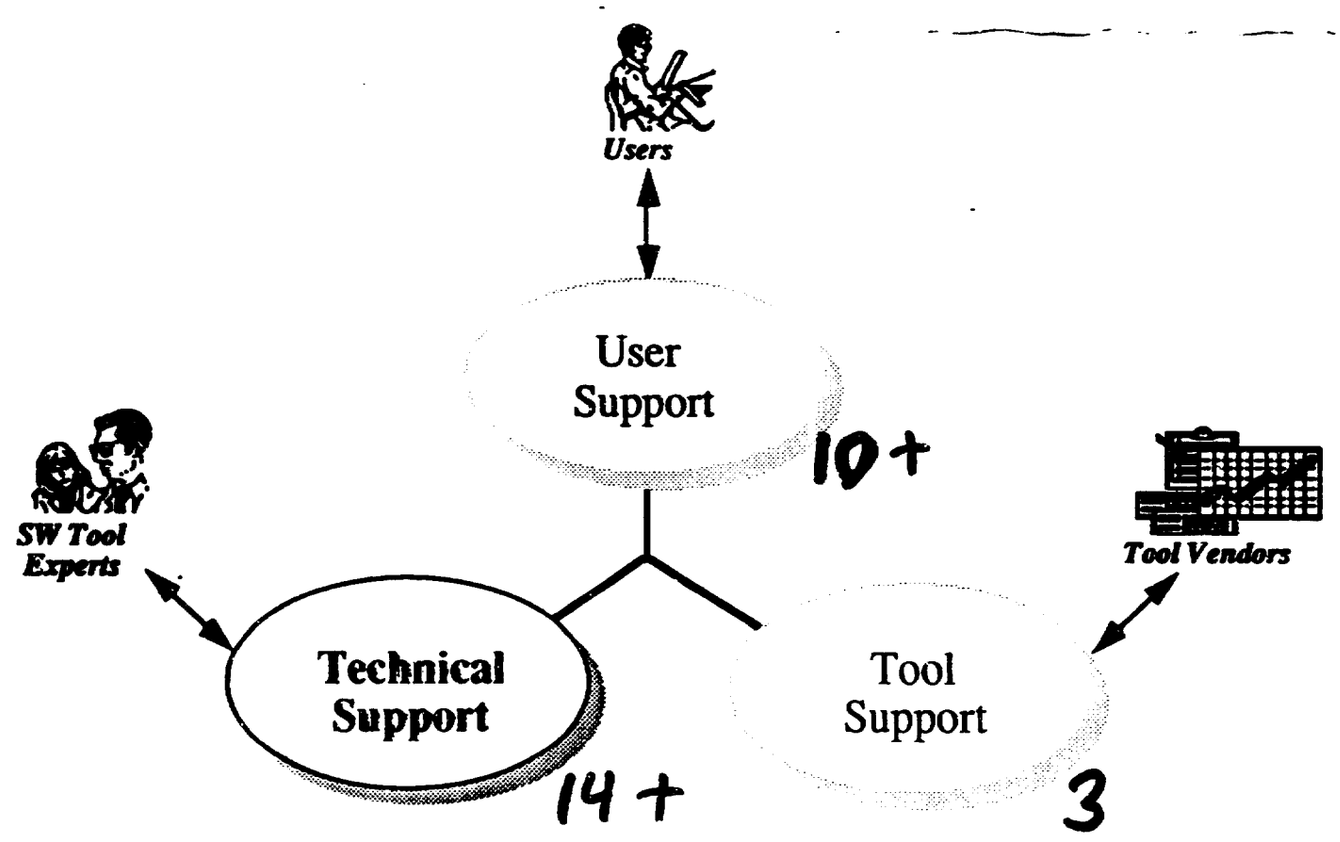
Air Force
STSC
Software Technology Support Center

STSC Functions



Air Force STSC Software Technology Support Center

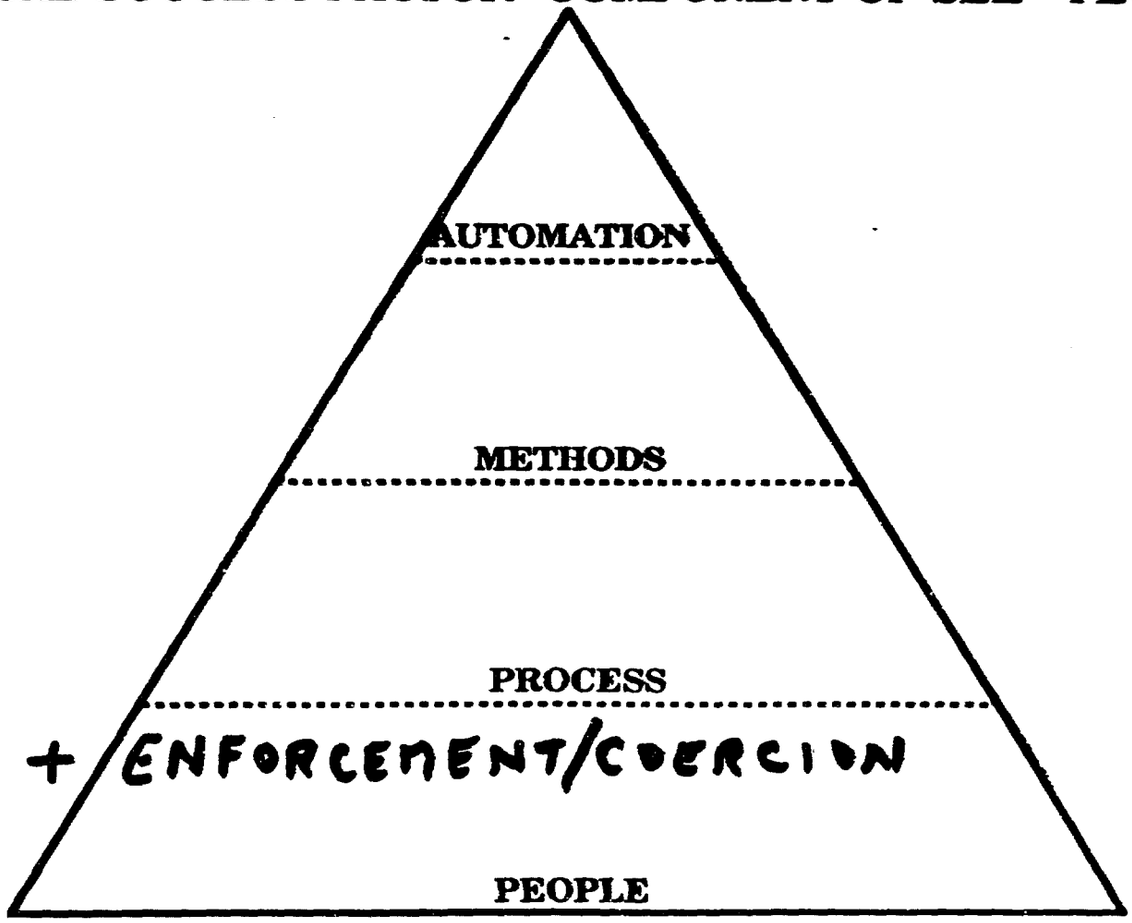
Technical Support



Air Force STSC Software Technology Support Center

SEE MODEL - EXTENDED

"CRITICAL SUCCESS FACTOR" COMPONENT OF SEE - PEOPLE



5/8/90

Source: Adapted from Charette, 1986

57

Air Force STSC Software Technology Support Center

SEE TAXONOMY - EXAMPLE ONLY

Environment Category	Capabilities Req'd	Imp't'nt	Useful	Repos	L-C Sup	Lang	etc.
SEE	X	X		X			
S.W. ENG. "Environment"	X			X			
INTEGRATED "environment"	X						
"environment" Tools/procedures							

Note: All definitions and criteria to be determined

SOFTWARE ENGINEERING REQUIREMENTS FOR THE STRATEGIC DEFENSE SYSTEM

JACKIE CRISTINA
U. S. ARMY STRATEGIC DEFENSE COMMAND
TECHNOLOGY BRANCH
CSSD-SA-BT
(205) 955-3861

UNCLASSIFIED

OUTLINE

- PURPOSE
- HISTORICAL PERSPECTIVE
- THE SDS ARCHITECTURE
- PROBLEM
- APPROACH TO INTEGRATION, SIMULATION, AND MAINTENANCE

PURPOSE

- SOFTWARE NEEDS OF STRATEGIC DEFENSE INITIATIVE
- USASDC APPROACH TO FULFILLING NEEDS FOR STRATEGIC DEFENSE SYSTEM DEVELOPMENT

HISTORICAL PERSPECTIVE

DEFENSIVE TECHNOLOGIES STUDY TEAM ("THE FLETCHER PANEL"),

VOL V - BATTLE MANAGEMENT, COMMUNICATIONS, AND DATA PROCESSING, DTD OCT 83

FINDINGS:

- BATTLE MANAGEMENT (AND SYSTEM) SOFTWARE MORE COMPLEX THAN SAFEGUARD
- GEOGRAPHICALLY DISTRIBUTED COMPUTING RESOURCES, INCLUDING SPACE BASED DRIVE COMPLEXITY
- OPERATIONS IN HOSTILE ENVIRONMENTS
- CAPABLE OF COPING WITH TEMPORARY SYSTEM FAULTS

RECOMMENDATIONS:

- COMPLETE SOFTWARE DEVELOPMENT ENVIRONMENTS THAT ALLOW:
  - Δ FORMAL REQUIREMENTS AND SPECIFICATIONS LANGUAGES
  - Δ SYSTEM AUDIT TRAILS WITH ANALYSIS TOOLS
- DEVELOPMENT OF REALTIME SOFTWARE WITH PERFORMANCE SPECIFIED AND VERIFIED AT ALL LEVELS OF SYSTEM

UNCLASSIFIED

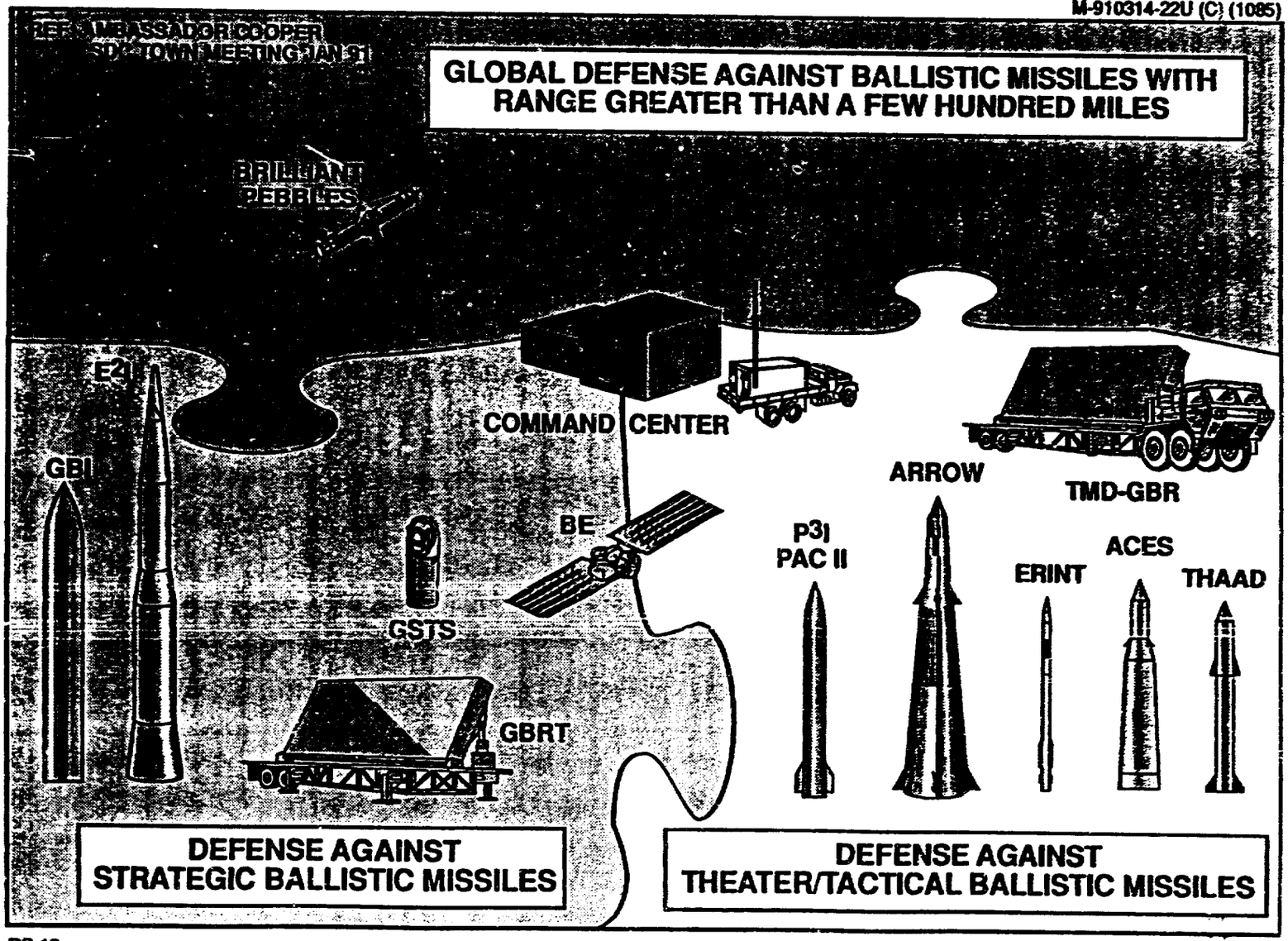
HISTORICAL PERSPECTIVE

EASTPORT STUDY GROUP,

SUMMER STUDY 1985-A REPORT TO THE DIRECTOR, STRATEGIC DEFENSE INITIATIVE ORGANIZATION DECEMBER 1985

"... THE ANTICIPATED COMPLEXITY OF THE BATTLE MANAGEMENT SOFTWARE AND NECESSITY TO TEST, SIMULATE, MODIFY, AND EVOLVE THE SYSTEM MAKE BATTLE MANAGEMENT AND COMMAND, CONTROL, AND COMMUNICATION (BM/C3) THE PARAMOUNT STRATEGIC DEFENSE PROBLEM."

GLOBAL PROTECTION AGAINST LIMITED STRIKES (GPALS) (U)

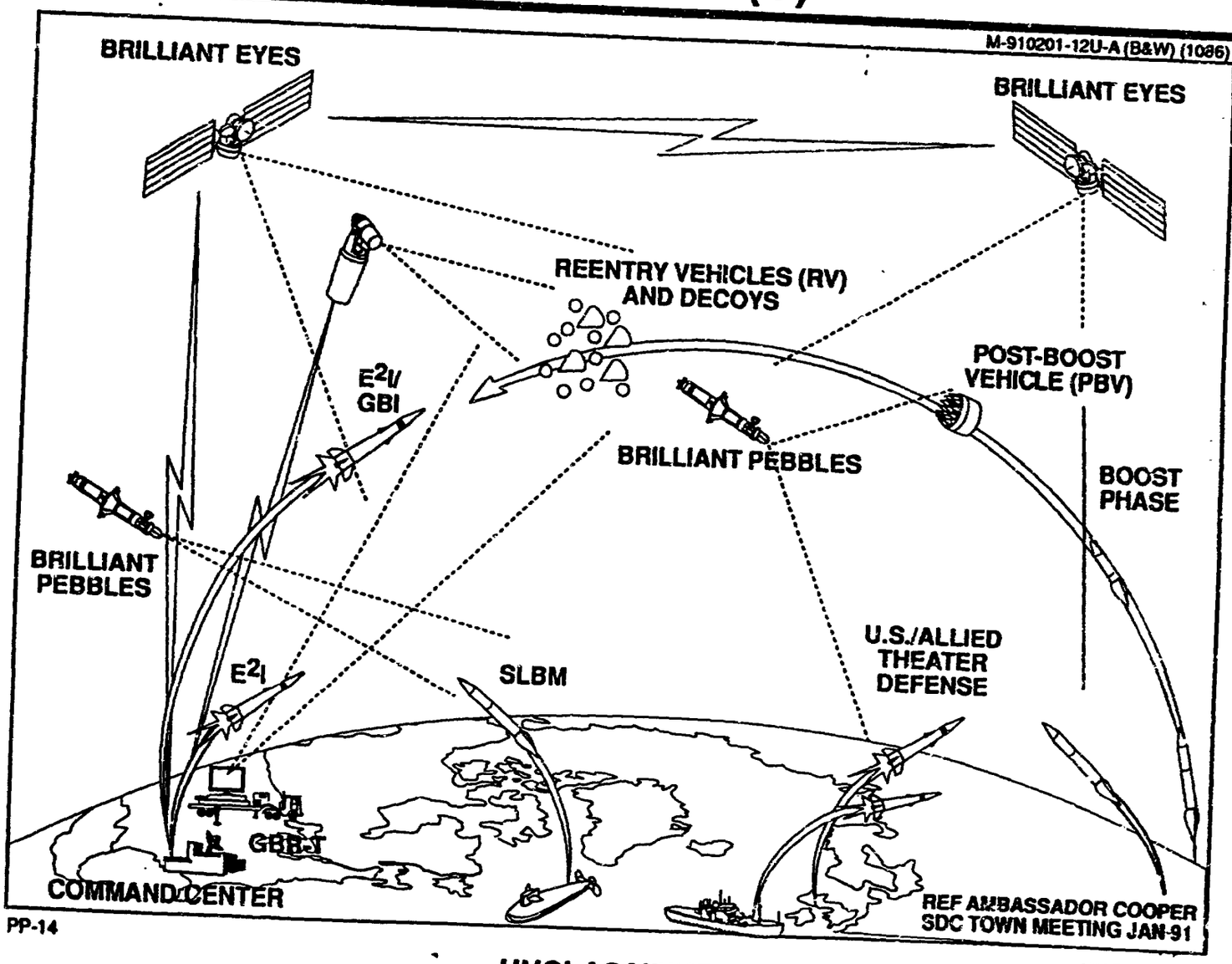


PP-13

UNCLASSIFIED

UNCLASSIFIED

GPALS ELEMENTS–STRATEGIC AND THEATER (U)



UNCLASSIFIED

UNCLASSIFIED

SDS SOFTWARE SIZING

CURRENT THINKING

BM/C3 SYSTEM	**9.485 MLOC**
ELEMENTS	
ELEMENT OPERATIONS	**TBD**
ELEMENT BATTLE MGT	**TBD**

SOURCE: USASDC SOFTWARE SIZING, MATTR STUDY, SEPT 90

THE PROBLEM

SOFTWARE FOR STRATEGIC DEFENSE SYSTEM

- HIGHLY RELIABLE
- GREAT COMPLEXITY
- LARGE VOLUME
- SCHEDULES
- TRUSTED AND SECURE ENVIRONMENT

INTEGRATION PROCESS

SYSTEM COMPLEXITY

- MULTI USERS
- DISTRIBUTED
- MULTI CONTRACTORS
- REALTIME
- MULTI AGENCIES
- OPERATIONAL AND TECHNICAL INTERFACES
- MAINTENANCE AND SUPPORTABILITY



REASONS FOR A SOFTWARE ENGINEERING ENVIRONMENT



- GOOD ENGINEERING PRACTICES
- FORMAL STRUCTURES
- EVOLVING SYSTEM
- CHANGING OF REQUIREMENTS
- ADA DEVELOPMENT FOR A SYSTEM
- REUSE
- MAINTENANCE OF A SYSTEM
- TRACEABILITY OF REQUIREMENTS
- INTERFACE CONTROL

USASDC APPROACH

DISTRIBUTED COMPUTING DESIGN SYSTEM (DCDS)

- MATURE PRODUCT
- GOVERNMENT OWNED AND CONTROLLED
- OPERATION AND MAINTENANCE MODE
- SELECTED AS THE SOFTWARE DEVELOPMENT ENVIRONMENT
- PROVIDED AT NO COST TO USERS

STRATEGIC DEFENSE DEVELOPMENT SYSTEM (SDDS)

- BUILDS ON CONCEPTS AND METHODOLOGIES OF DCDS
- SELECTED AS THE SDS SOFTWARE ENGINEERING ENVIRONMENT
- COMPLETED 32 MONTHS OF AN 80 MONTHS CONTRACT
- COMPLETED CRITICAL DESIGN REVIEW AND INTO AN ALPHA SYSTEM DEVELOPMENT
- GOVERNMENT OWNED AND CONTROLLED

SUMMARY

- **USASDC APPROACH TO SYSTEM INTEGRATION:**
  - Δ IMPLEMENT A SOFTWARE POLICY
  - Δ USE OF ADA AS THE SOFTWARE DEVELOPMENT LANGUAGE
  - Δ REQUIRE DELIVERY OF THE SYSTEM INTO AN USASDC SOFTWARE ENGINEERING ENVIRONMENT

- **HOW TO INFORMATION ABOUT DCDS/SDDS:**

JACKIE CRISTINA
CSSD-SA-BT
U. S. ARMY STRATEGIC DEFENSE COMMAND
P. O. BOX 1500
HUNTSVILLE, ALABAMA 35807
PHONE (205) 955-3861

UNCLASSIFIED

Needs of PCTE Programme

A personal view of

Hugh Davis
ICL Secure Systems

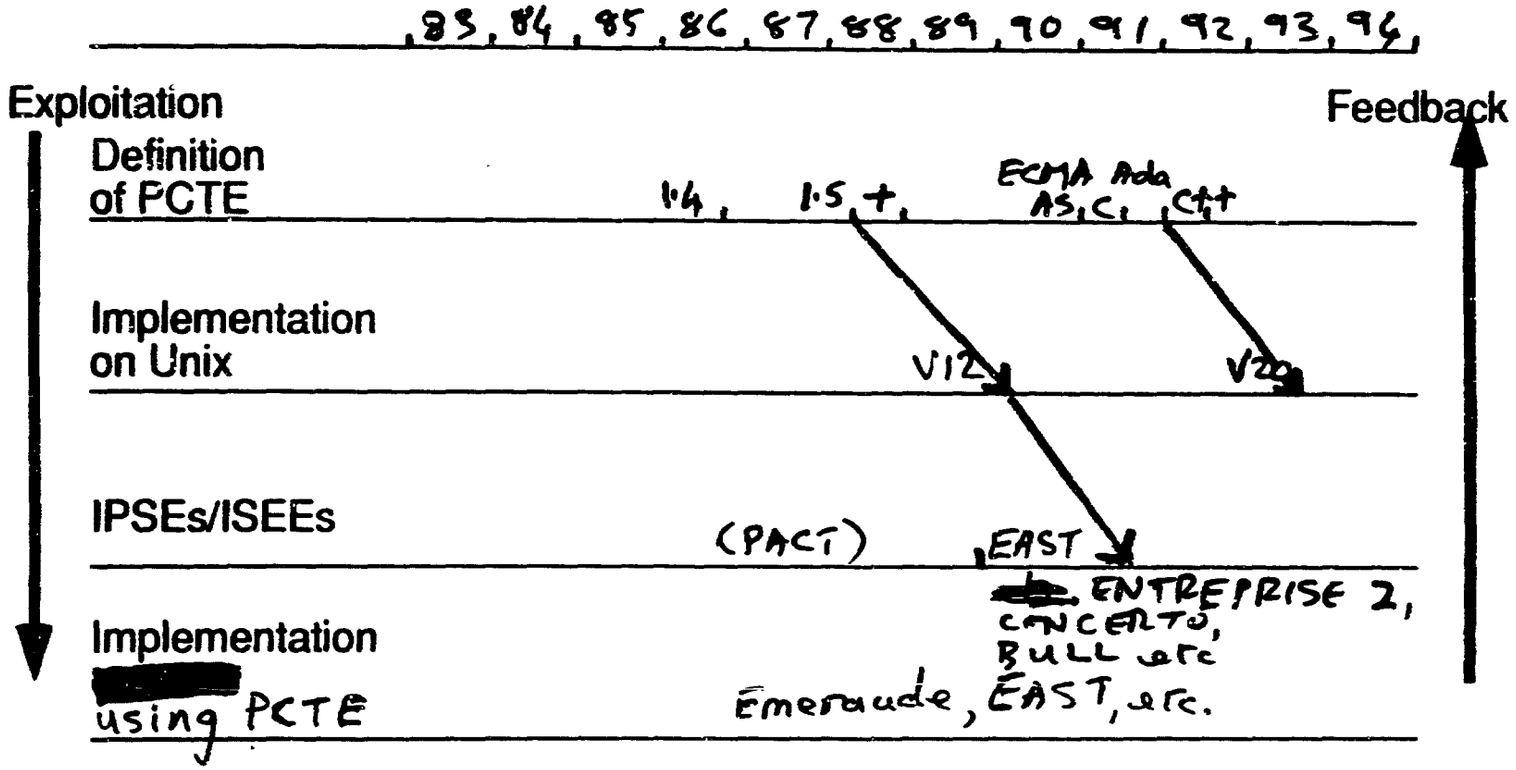
Second PCIS Workshop
3rd-7th June 1991

ICL

Contents

- PCTE in Perspective
- Needs of PCTE Programme
- PCTE-CAIS Convergence
- Vision for the Future

PCTE in Perspective

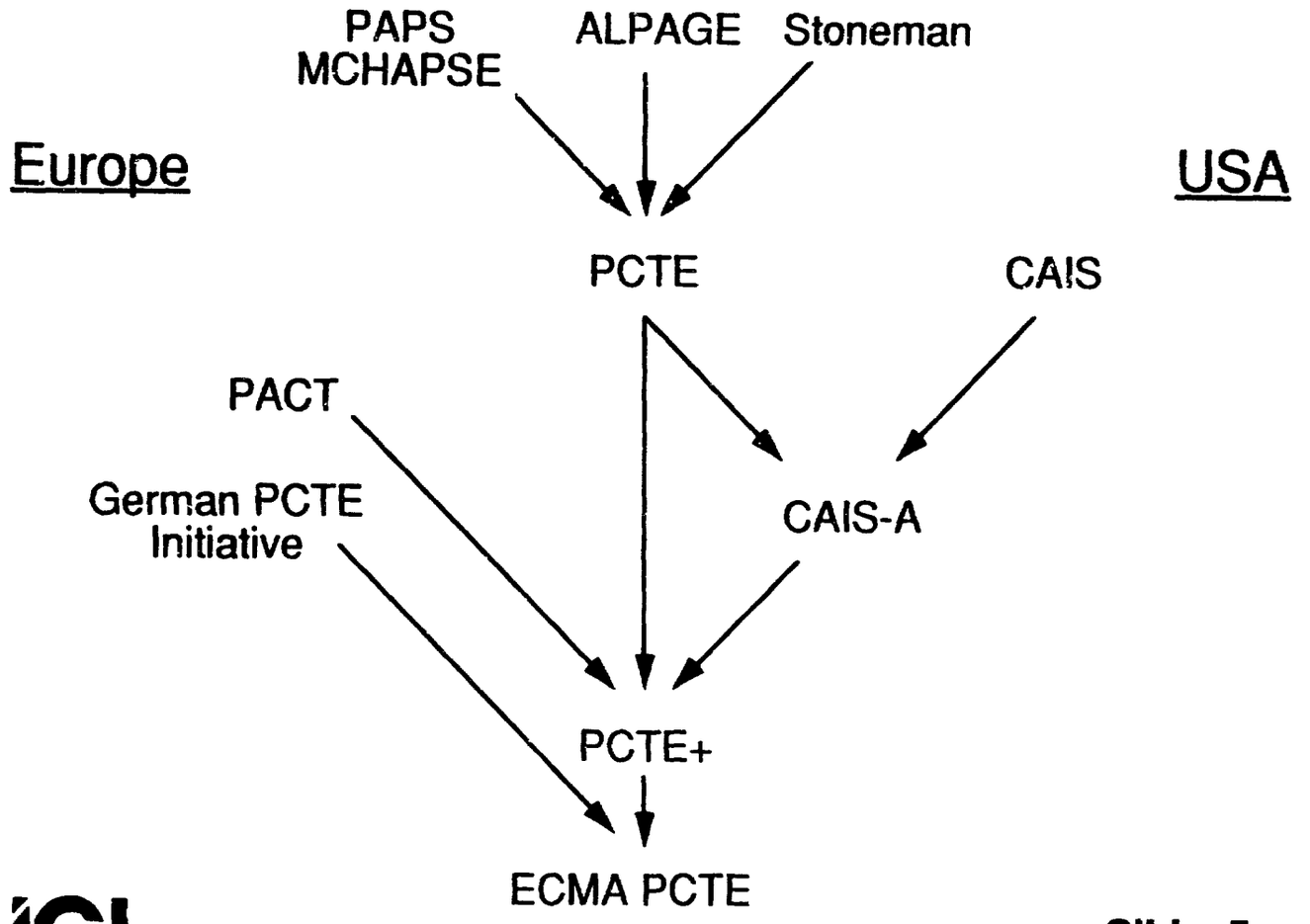


ICL

Needs of PCTE Programme

- Stability of definition, integrity of concepts and controlled change for
  - implementors
  - tool writers
  - environment builders
- Feedback from users
  - as above +
  - application writers
- Definition of PCIS aims and process
- Definition of TC33 role in PCIS
- Working method
  - PCIS results (*on PCTE*) developed as ECMA TC33-TGEP changes

Pedigree


PAPS
MCHAPSE
ALPAGE
Stoneman
Europe
USA
PCTE
CAIS
PACT
German PCTE
Initiative
CAIS-A
PCTE+
ECMA PCTE

PCIS Opportunities

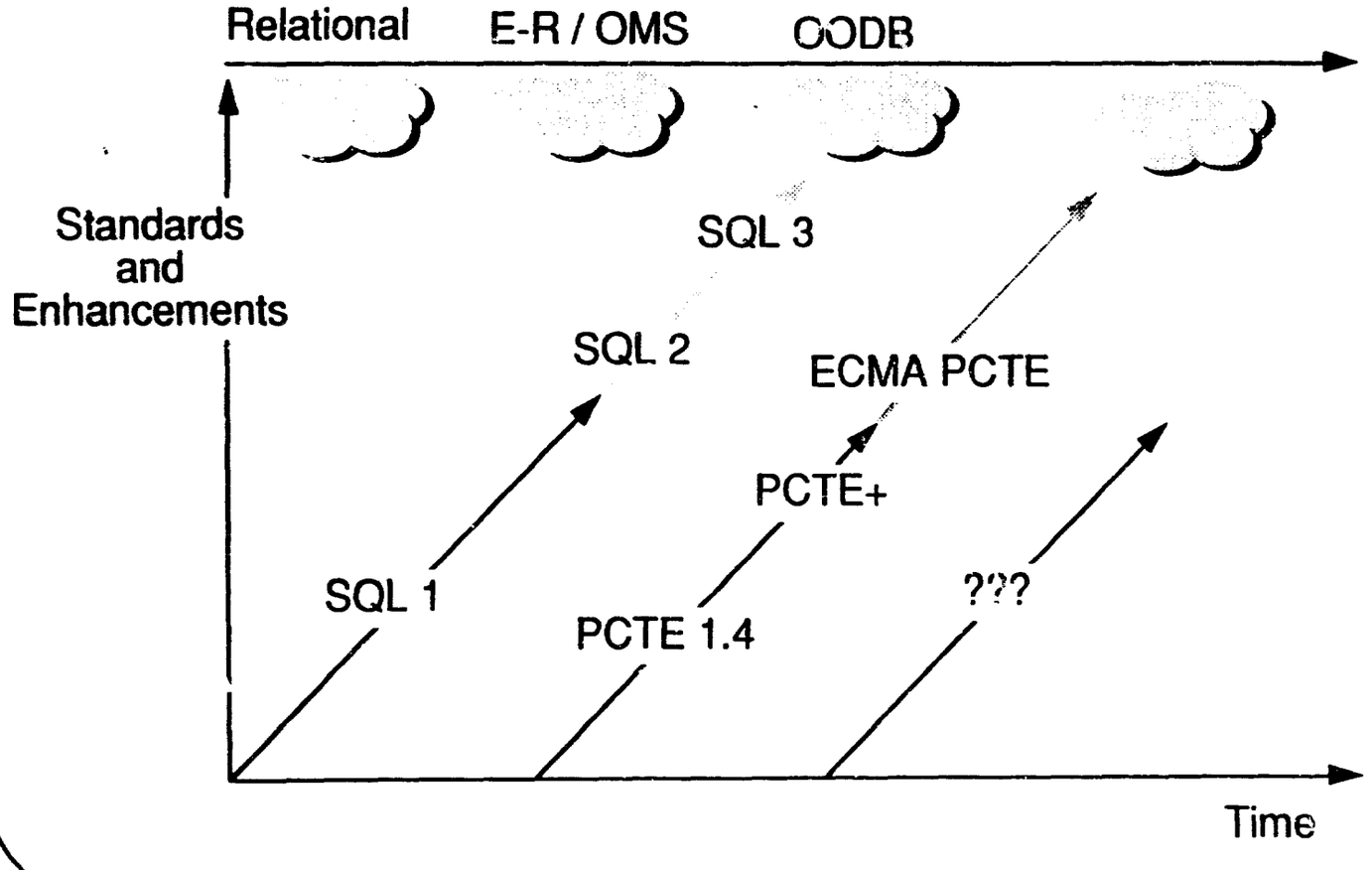
System Engineering progress
- Garmisch 6x
- Stoneman 80

"Military and Civil" market ≠>
- Equal value
- Mutually exclusive
- Opposed

No niche for interface def.

PCIS must work with existing "non-military" programmes
- PCTE
- ?

Database Evolution



Stoneman 9X

- Requirements
  - Process support
  - General purpose with SE specialization
  - Object-oriented
  - Knowledge-oriented
  - ...
- Approach
  - Application of SE techniques
    - Requirements capture and analysis
    - Semi-formal, abstract, structured definition
  - Requirements (from NRAC)
  - Design (from ECMA Reference Model with improvements)
    - Service descriptions
    - Structure of services
    - Glossary

Reference Model Improvements

- Service descriptions
  - Separate precise semi-formal statement from commentary
  - Rationalize "dimensions", e.g.
    - Summary
    - State (type definitions)
    - Operations
    - Related services
- Structure of services
  - Define and separate data, presentation and control integration
  - Views, e.g.
    - User (process integration)
    - ...
- Glossary

SEMATECH's Advanced Development Environment (ADE)

or

Software Engineering Environments: The Needs of Manufacturing Users

A Presentation to the PCIS Workshop
June 4, 1991

SEMATECH's Advanced Development Environment

Claude Baudoin

SEMATECH is a consortium of 14 U.S. companies which are either semiconductor manufacturers, or computer companies with their own chip-making operations. The consortium performs pre-competitive research on advanced manufacturing techniques and equipment. It is funded by DARPA for 50% of its $200M yearly budget, and by its member companies for the other half.

CIM (Computer-Integrated Manufacturing) is increasingly being recognized as one of the major challenges for effective manufacturing. SEMATECH has created a CIM Architecture which defines several layers:

- applications
- a generic semiconductor manufacturing model
- a development environment
- an execution environment

The Advanced Development Environment (ADE) project is a key part of how we intend to improve the quality and accelerate the emergence of new, model-driven manufacturing applications. The project's deliverables are created by a working group of member company representatives, created in December 1990, which has:

- defined the requirements for a CASE framework,
- performed an extensive evaluation of several commercial frameworks,
- started writing an "Analysis and Design Guide," a set of life cycle guidelines for CIM applications based on object-oriented techniques,
- developed a plan and lined up resources for education and technology transfer

In the framework area, SEMATECH's CIM Architecture concept of an "integrated model repository" containing various models which can drive the execution of CIM applications means that the presence of a data integration component is considered a key requirement. This and other aspects of the program relevant to the PCIS Workshop will be presented.

SEMATECH Advanced Development Environment

OUTLINE

- Program Context
- Vision, Goals and Strategies
- High-Level Environment Needs
- Roles, Activities and Scenarios
- Requirements

SEMATECH

A Consortium of American Semiconductor Manufacturers
Working with Government and Academia
to Sponsor and Conduct Research
Aimed at Assuring World Leadership
in Semiconductor Manufacturing Technology

SEMATECH

SEMATECH
INNOVATION FOR AMERICA'S FUTURE

A partnership among the American government, the American people, and fourteen semiconductor manufacturers

AMD
AT&T
Digital Equipment
Harris
Hewlett-Packard
Intel
IBM
LSI Logic
Micron
Motorola
National Semiconductor
NCR
Rockwell
Texas Instruments

SEMATECH

CIM APPLICATIONS ARCHITECTURE (CAA)

Framework for building and maintaining applications

CAA provides:

- **Methodological approach to development and maintenance**
- **Mechanism for defining and using manufacturing models**

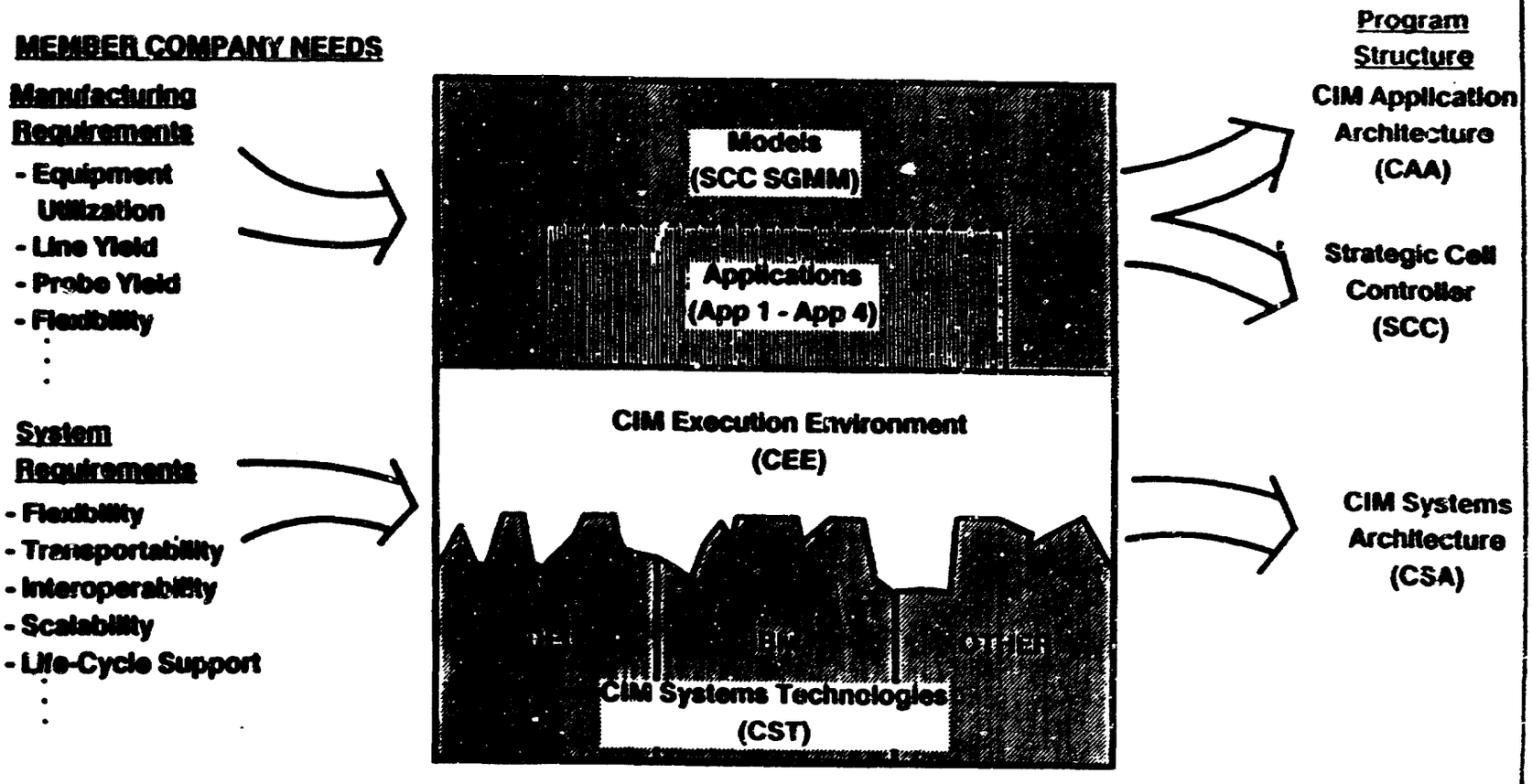
CAA includes:

- **Advanced Development Environment (ADE)**
- **CASE tools and Formal Development Methods**
- **Hierarchy of Integrated Models (IM)**
- **Integrated Model Repository (IMR)**

Execution environment is dependent on the modeling environment for its operation. As the models change, the application changes.

SEMATECH

ARCHITECTURAL VISION FOR FUTURE MANUFACTURING SYSTEMS



SEMATECH

SEMATECH Advanced Development Environment

VISION

Establish end-user computing for manufacturing applications (analogy with use of a spreadsheet program).

Build applications with unprecedented quality, and 2 to 10 times faster than with traditional methods.

Embed routine decision making in the software, freeing end users to tend to higher-level manufacuring decisions.

Fundamental shift in the role of system designers: instead of being the bottleneck in the implementation of the user's application requirements, they get out of the way but focus on building the tools that allow the domain experts to easily specify their own applications.

Preserve competition between CASE tool suppliers through "open framework" approach.

SEMATECH Advanced Development Environment

GOAL

To reduce the time, cost and skills required to develop and maintain manufacturing systems by creating an integrated environment which consists of a rich set of tools allowing multiple customers to direct all stages of an application's usage, from requirements to retirement.

Key phrases:
- **reduce [system] skills**
- **integrated environment**
- **multiple customers**
- **all stages**

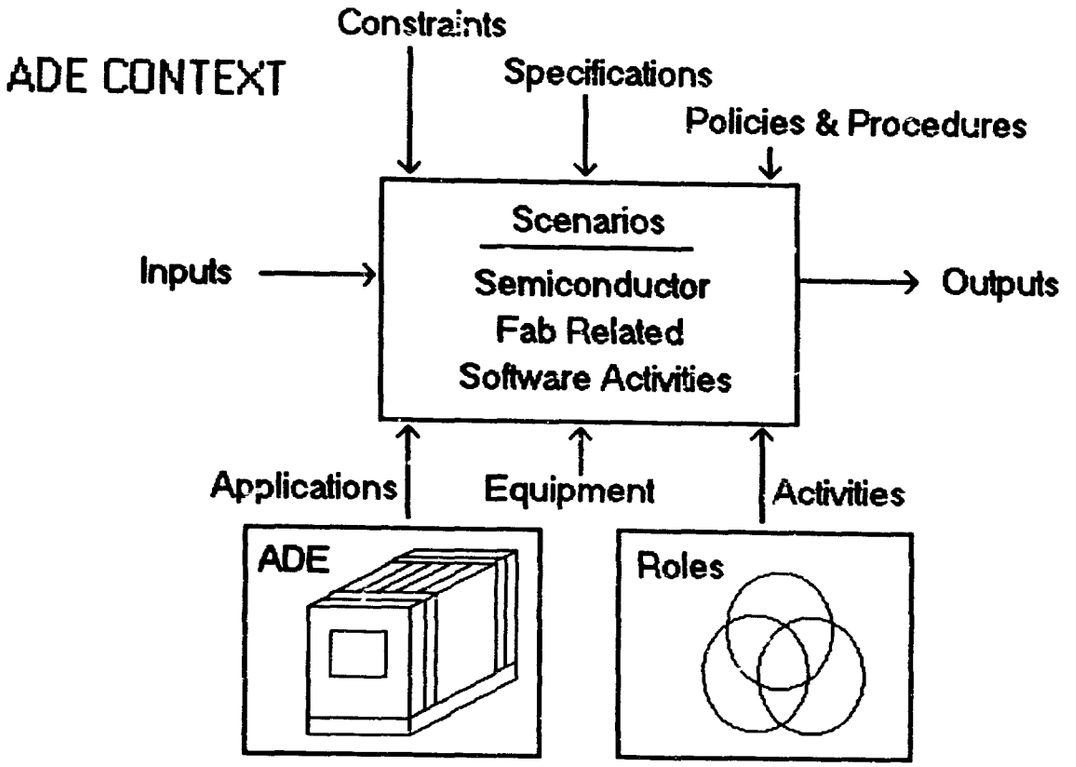
CONCLUSION

- Many needs are the same as those of other types of users
- A few critical needs are driven by specific manufacturing needs:
  - access to repository at run-time
  - reuse of analysis and design models, with their repository implications
  - language and operating system preferences
- We have started discussing the potential synergy with:
  - other DARPA-sponsored projects (STARS)
  - other manufacturing organizations (e.g. Ford)
- All the necessary technology seems to exist, although from foreign sources (e.g. PCTE)
- We need commercial support of our needs now
- Intriguing possibilities to be explored in manufacturing world:
  - extend the repository to manufacturing information
  - extend process management to cover the manufacturing process

SEMATECH Advanced Development Environment

Overview of Framework Requirements

- Mask file system under OO or ER view of information
- Distributed
- Transaction control
- Concurrent access control
- Explicit metamodel
- Extensible metamodel (including at run time)
- Version management
- Run-time access to the repository
- Commercial status
- OSF/1 migration plans
- Ease of use
- Cost
- etc.


ADE CONTEXT
Constraints
Specifications
Policies & Procedures
Scenarios
Semiconductor
Fab Related
Software Activities
Inputs
Outputs
Applications
Equipment
Activities
ADE
Roles

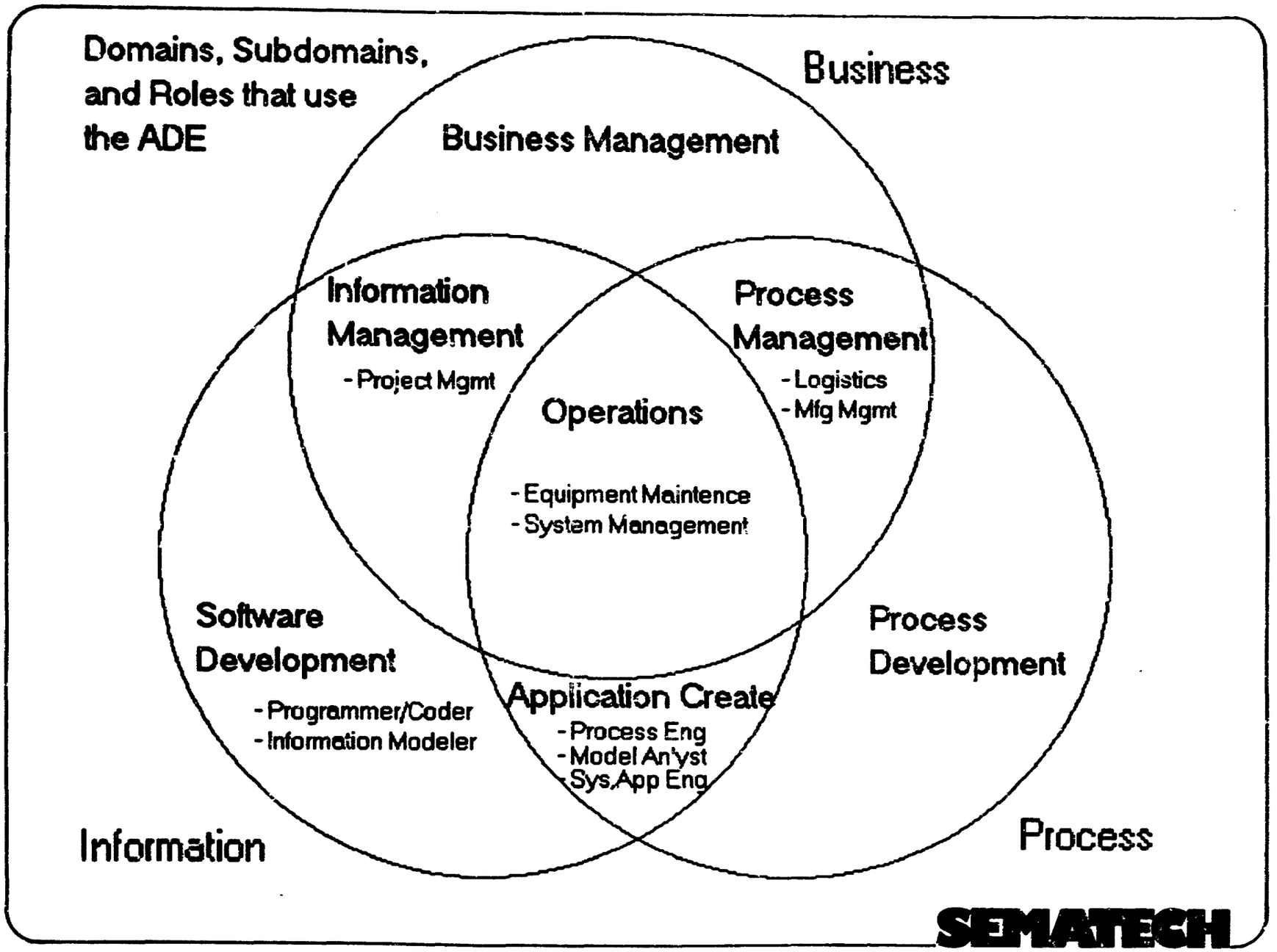
SEMATECH

Scenario Content

Scenario Title
Purpose of Scenario:
Example of Scenario:

Role 1	**Activity 1** Description of Activity 1 **Activity 2** Description of Activity 2	**Objects associated with Activity 1 & 2** List of Objects
Role 2	**Activity 3** Description of Activity 3 **Activity 4** Description of Activity 4	**Objects associated with Activity 3 & 4** List of Objects

SEMATECH


Domains, Subdomains, and Roles that use the ADE
Business
Business Management
Information Management
- Project Mgmt
Process Management
- Logistics
- Mfg Mgmt
Operations
- Equipment Maintence
- System Management
Software Development
- Programmer/Coder
- Information Modeler
Application Create
- Process Eng
- Model An'yst
- Sys,App Eng
Process Development
Information
Process
SEMATECH

SEMATECH Advanced Development Environment

High-Level Environment Needs

- enable migration to a process-driven, technology-supported paradigm
- explicitly recognize the roles and activities of the users and the objects they manipulate
- include process management, project management and configuration management
- support an "Integrated Model Repository," therefore provide fine-grained data integration (use control integration as a stopgap measure only)
- facilitate reuse of analysis and design objects, not just code
- extensibility, e.g. add new schemas for objects created by newly integrated tools
- support object-oriented methods, tools and languages (C++ and Smalltalk, not Ada) (impact on encapsulation strategy: needs access to class inheritance structure)
- provide for access to objects in repository at execution time

STRATEGIES

Cooperate with, influence, and leverage the knowledge of member companies; transfer our results to them.

Cooperate with, influence and leverage the knowledge of suppliers of CASE tools, in order to foster a strong U.S. capability in this domain and to facilitate continued competition and improvement in this marketplace.

Cooperate with, influence and participate in national and international standards efforts in the software engineering domain.

Periodically reevaluate the project's mission and steer our efforts to fulfill it.

Measure our achievements according to quality (as determined by our customers) and timeliness.

Environments — An Industry View

Introduction

PCIS 1 — Environment Users

Environment-Related MCC Projects

Summary Recommendations

Leigh Power
MCC, June, 1991

Introduction — Purpose

Q: "What's MCC doing in environments?" [Colket]

A: nothing. [Power]

A: a lot! [Power]

Summary of *some* environment-related MCC projects.

Relate MCC business experience to that of PCIS 1.

576

Goal:

- **Broaden (further) context of design environments.**
- **Flavor of what is commercially viable.**
- **Personal view of PCIS role for MCC.**

Industry Requirements — Environment Users (PCIS 1)

Focus on relevant, recognized, industrial problems:

- business problems >> technology
- systems (not just software) design
- support: process, teams, horizontal tools, change

... in the *context* of existing business assets:

- can't discard current systems
- heterogeneous, federated, secure subsystems
- customization

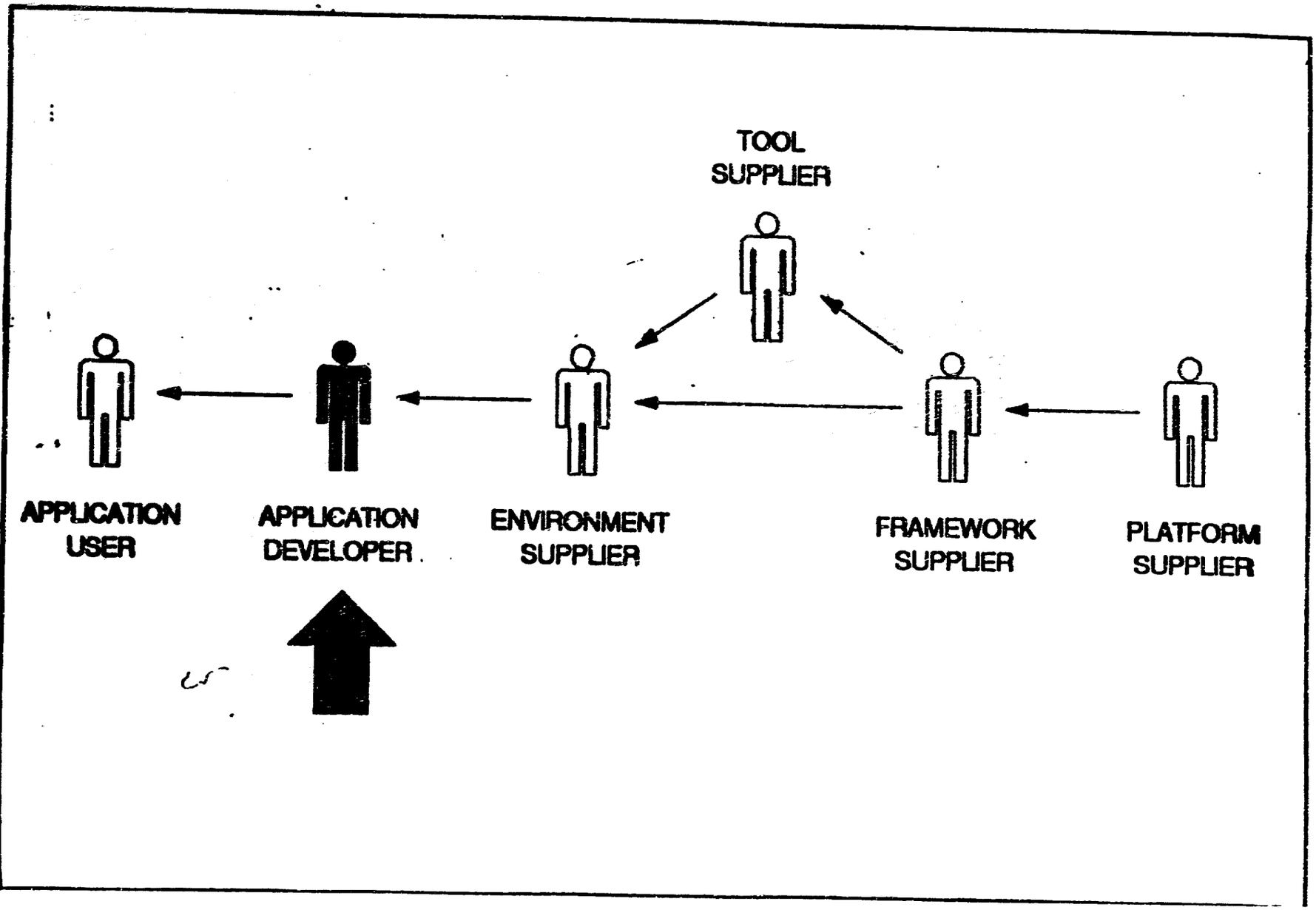
... using an evolutionary *process*.

- public process; commercial incentives
- environment evolution; stabilize PTIs
- user role: market pull (and direction)

Similar experience at MCC.

577

Environments Industry "Food Chain"



578

Some Environment-Related MCC Projects

EI (Enterprise Integration) Roy Smith (CAD) 512-338-3738 rsmith2@mcc.com
business(es) as an integrated, electronic whole

LDL (Logic Data Language) Carlo Zaniolo (STP) 512-338-3442 zaniolo@mcc.com
example of enterprise modeling (static and dynamic)

CT (Coordination Technology) Baldav Singh (STP) 512-338-3701 singh@mcc.com
coordination of people, tasks, organizations

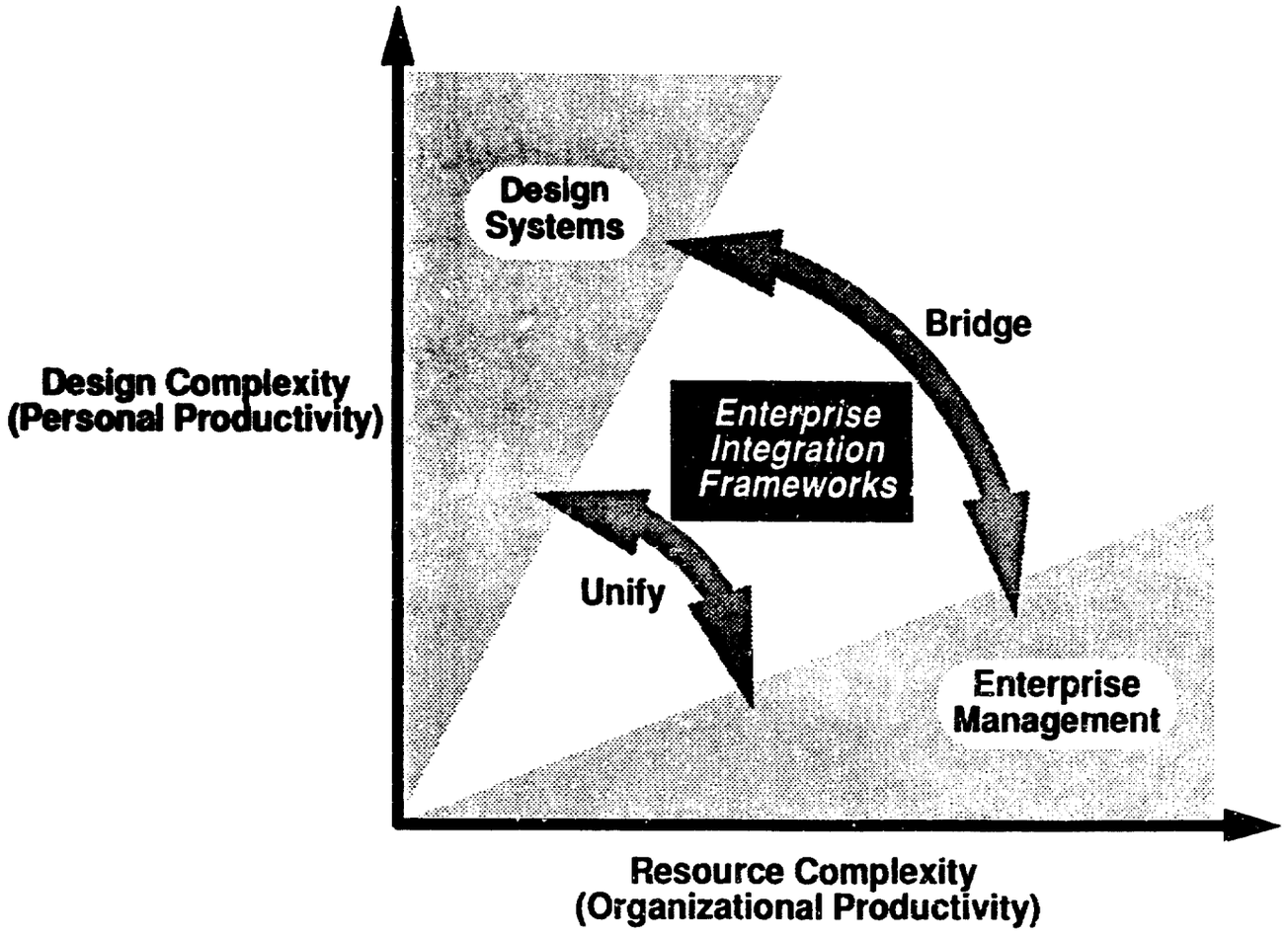
Carnot Phil Cannata (ACT) 512-338-3376 cannata@mcc.com
heterogeneous information management

ESP (Extensible Software Platform) Rob Smith (ESL) 512-338-3452 rjsmith@mcc.com
heterogeneous, parallel, compute engines

CFI (CAD Framework Initiative) Tom Rhyne (CFL) 512-338-3521 rhyne@mcc.com
model for evolutionary standards process

Enterprise Integration Conceptual Architecture

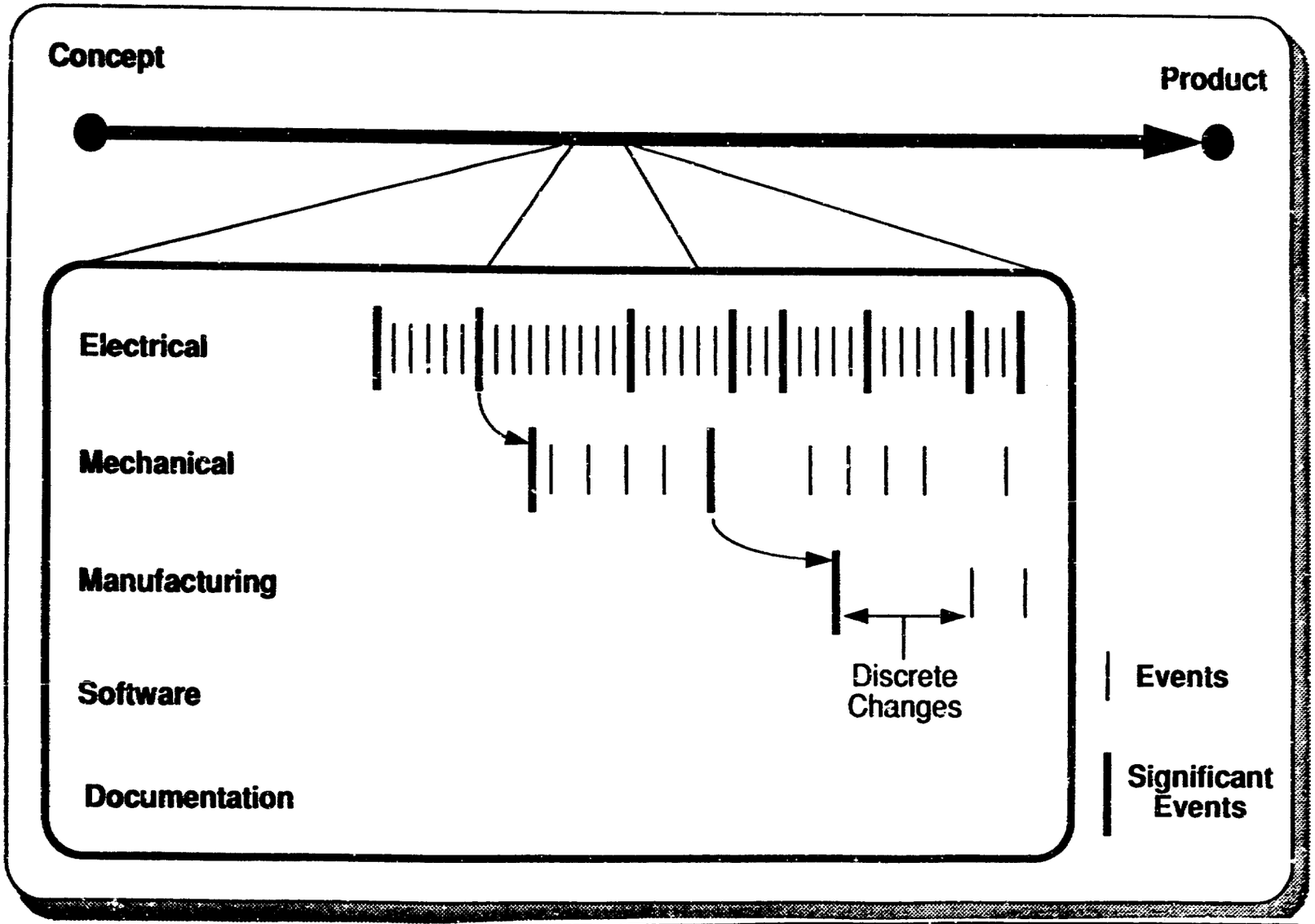
Filling in the Gap



© Microelectronics and Computer Technology Corporation

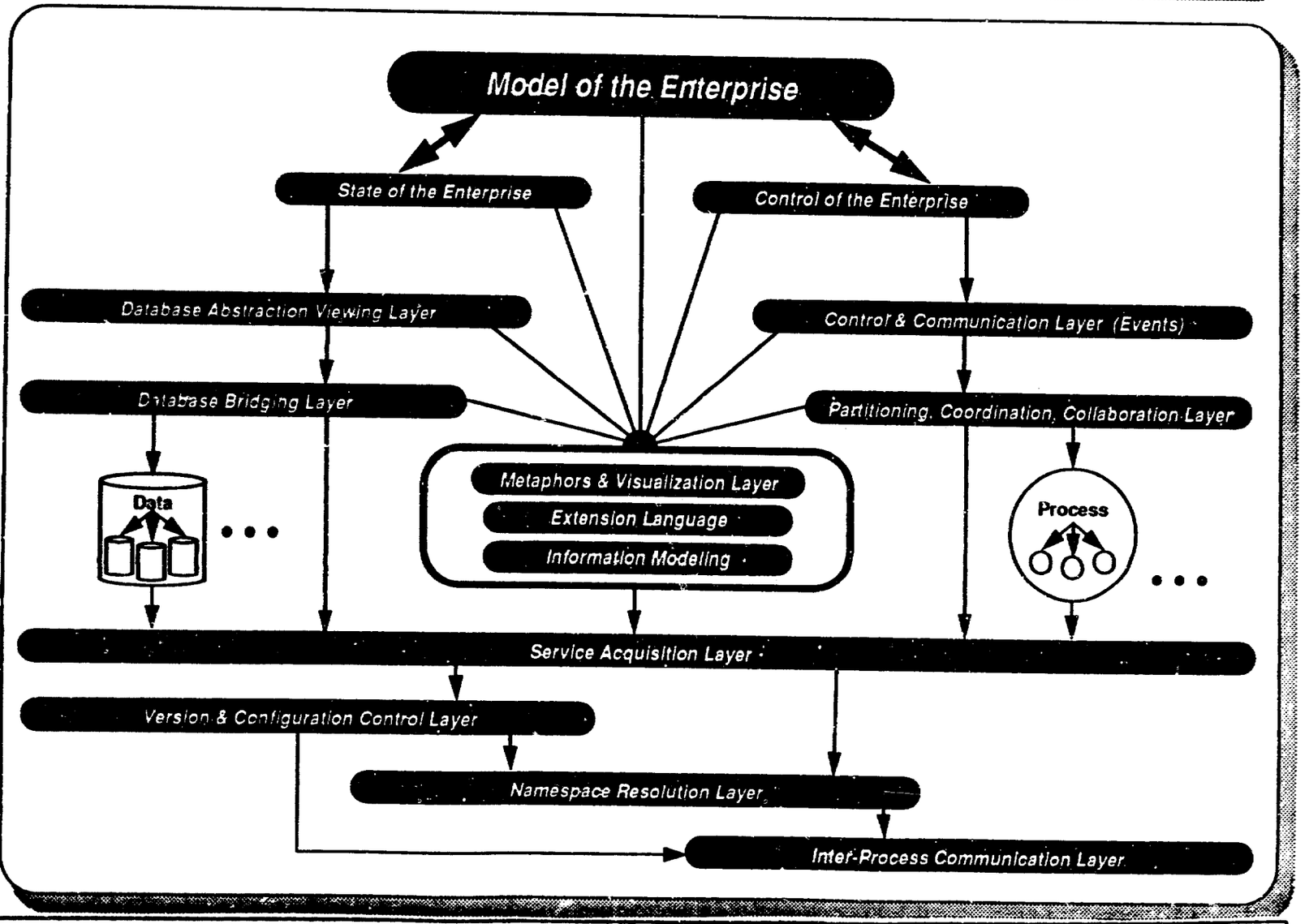
Enterprise Integration Conceptual Architecture

Fundamental Concepts - Change Control & Events



Enterprise Integration Conceptual Architecture

Integrating the Enterprise

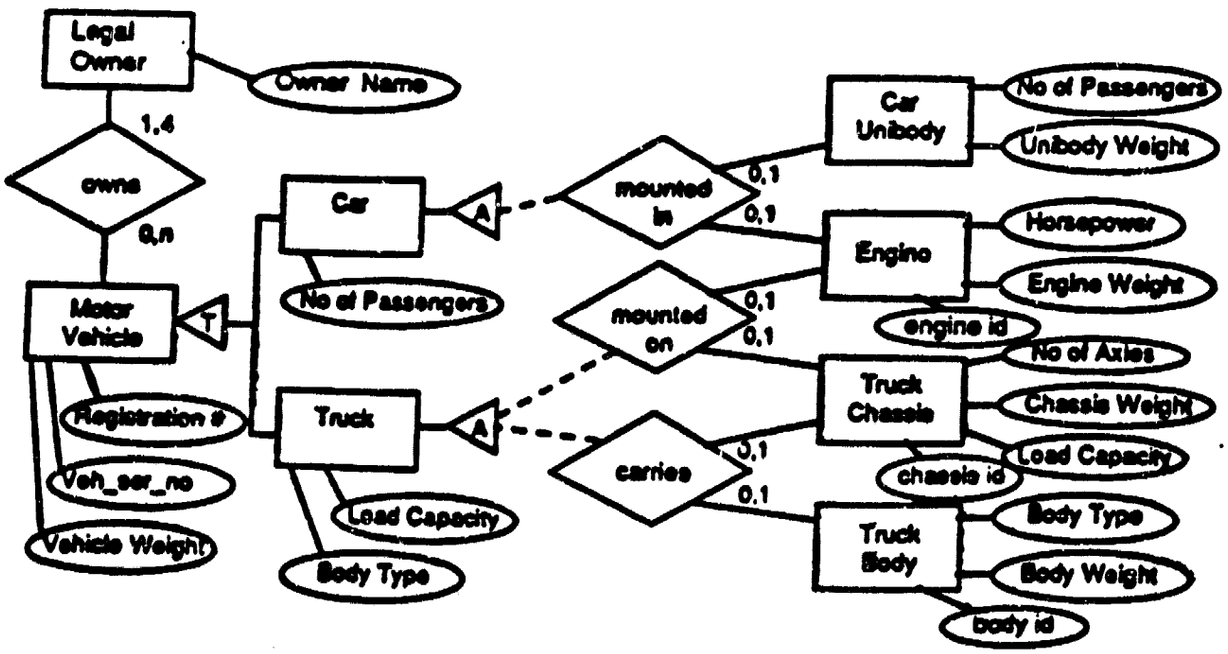


Enterprise Modeling
(Together with Pacific Bell Comp.)

Specifying in a consistent way, the details of an information system required for the support of some business. The POS methodology maps these dimensions into a set of E-R descriptions from which they are mapped into *LDL*.

Example: The N695 Motor Vehicle Registration Authority

- Objects: Cars, Trucks, Engines, etc.
- Processes: Transferring ownership from one legal owner to another.
- Constraints: "*A Vehicle must be owned by n legal owners* $0 < n \leq 4$"



Research Goal:

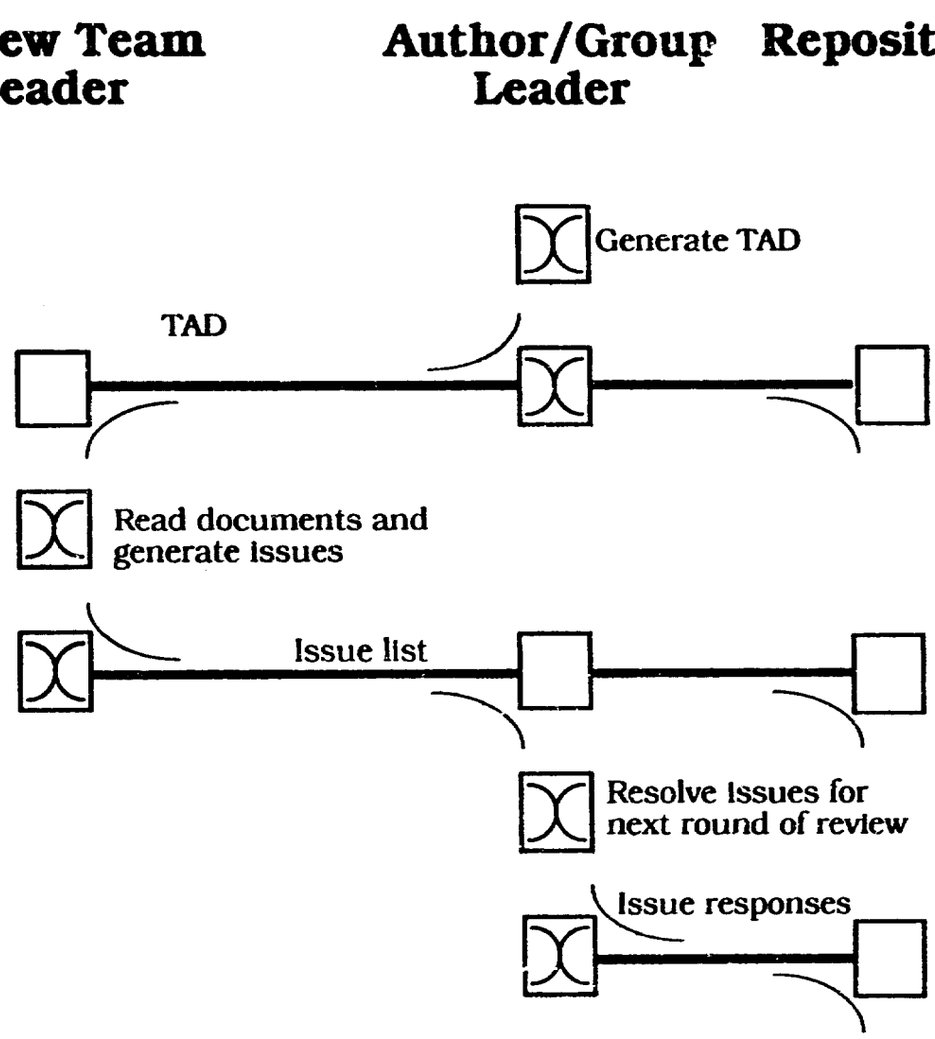
- Develop a software system to support coordination among people, tasks, and resources in an organization. The system will
  - encompass a formal model of coordination
  - support incremental project planning, execution, status monitoring and evolution
- Research focus is on coordination in general, using the software development process as an example.

Research Approach:

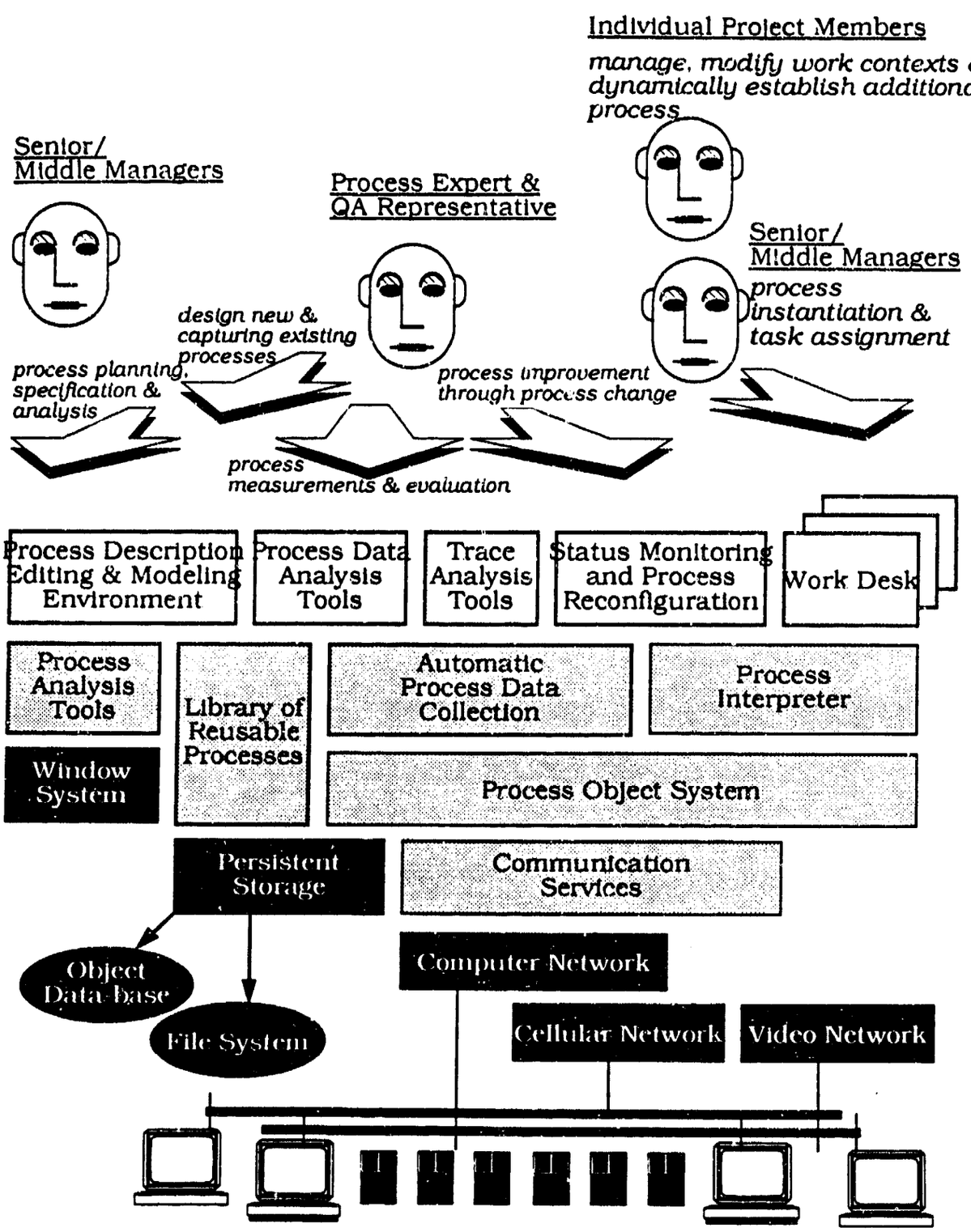
- *Hard automation* — inflexible execution of routine processes — is inappropriate because the work organizations are
  - not static but highly dynamic systems
  - evolve through continuous change
- *Flexible, adaptable and integrated automation* is required —
  - Flexible in human involvement, process enforcement and dynamic change
  - Adaptable to particular organizational practices
  - Integrated with existing tools & methods

584

Formal Technical Review Process



Block Architecture of Coordination Env.

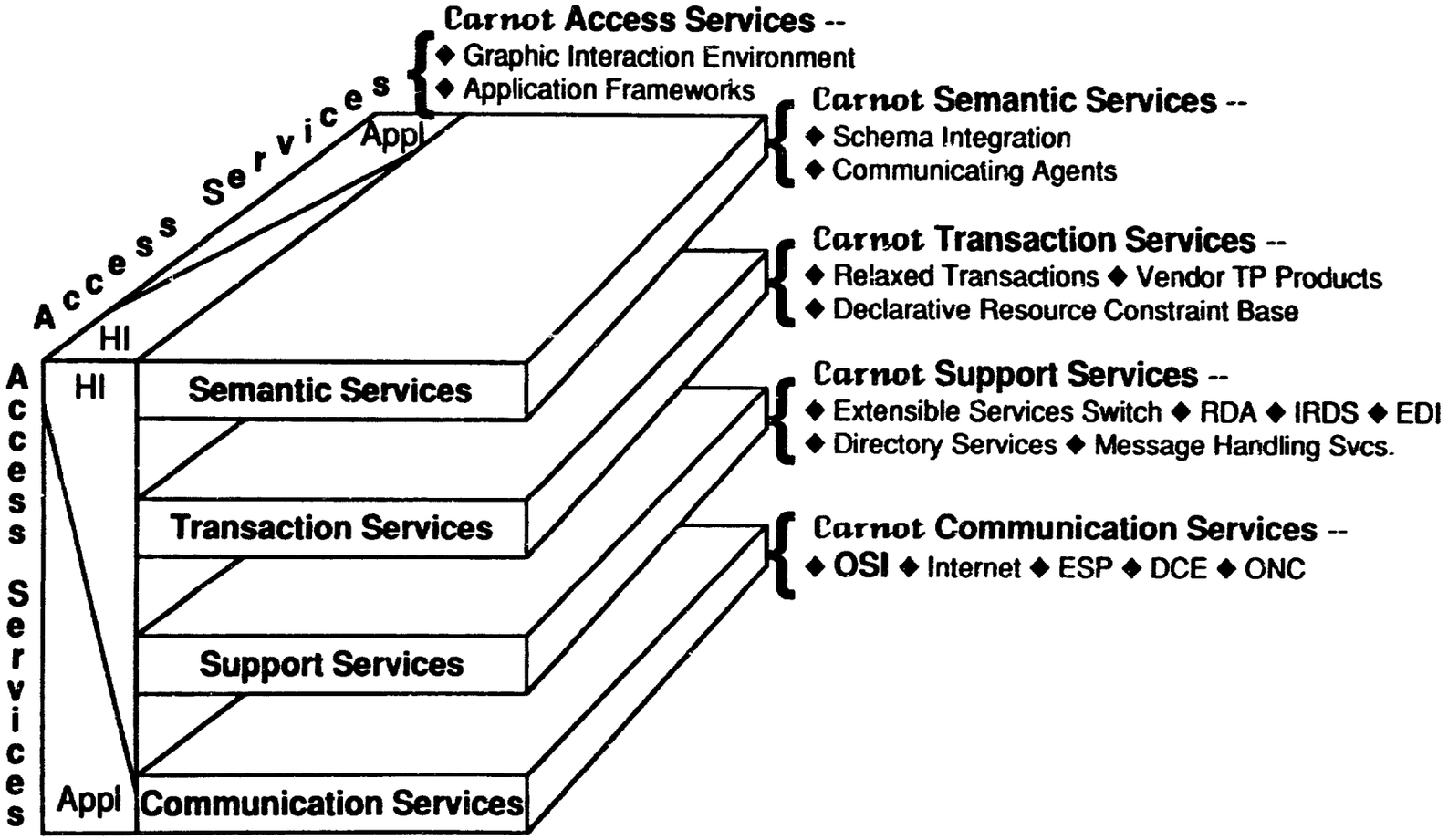


Carnot Project Goals for Enterprise Information Integration

- Demonstrate Open, Standard Implementations of Interoperability Architectures such as DAIS, OSCA, and MIA, Using Third-Party Software Extensively.
- Facilitate Corporate Logical Data Modeling (CLDM) Efforts.
- Intercept OSI World
- Prototype a Complete System (from Human Interface down to Network Communications).
- Unleash Highly Distributed Environments (Enterprise-Wide).
- Provide Significant, Positive Financial Impact for Carnot Members

© Microelectronics and Computer Technology Corporation

Carnot Architecture



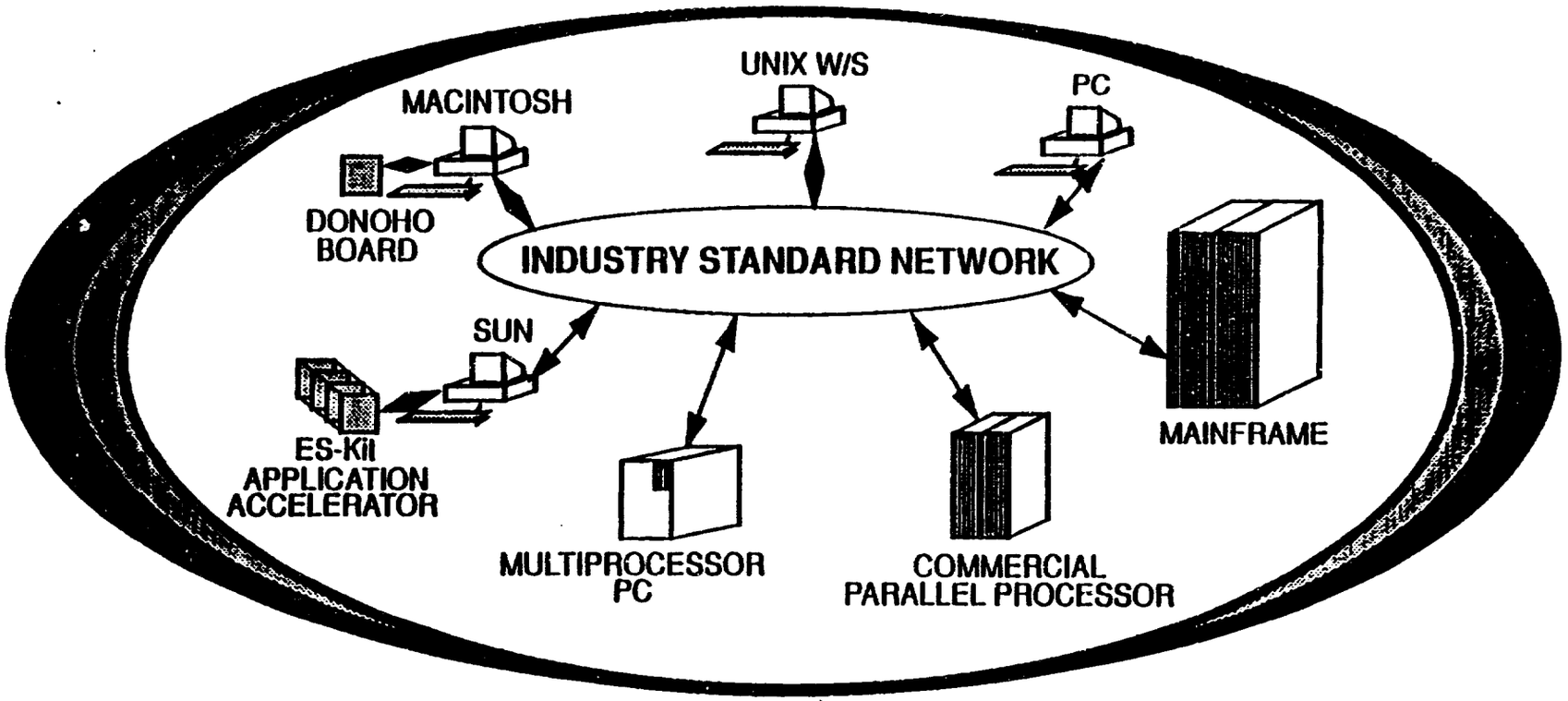
MCC Carnot Project

MCC/Advanced Computing Technology Program

Rev. 5.2

© Microelectronics and Computer Technology Corporation

Extensible Software Platform



ESP BASED APPLICATIONS WILL EXECUTE USING ALL AVAILABLE PROCESSORS IN PARALLEL

BASED UPON INDUSTRY STANDARD C++ LANGUAGE and TCP/IP NETWORKING

PRESERVES USER CHOICE OF APPROPRIATE PLATFORM

PRESERVES USER CHOICE OF NON-ESP BASED APPLICATIONS

PROVIDES DEVELOPER ACCESS TO SCALEABLE MULTIPLE HARDWARE AND O/S ENVIRONMENTS

FOR OBJECT - ORIENTED C++ APPLICATIONS

© Microelectronics and Computer Technology Corporation

Distributed Execution: What makes it *possible* ?

☑ *overloaded* method invocation

```
RempteBase::operator->()
employee* John_Doe;
new_salary = John_Doe->raise(more);
```

☑ *extended* object creation

```
John_Doe = new {remote} employee();
```

☑ *futures* for multi-threaded execution

```
future f;
f += Tax->compute_tax(salary);
f += Finance->compute_benefit(salary);
```

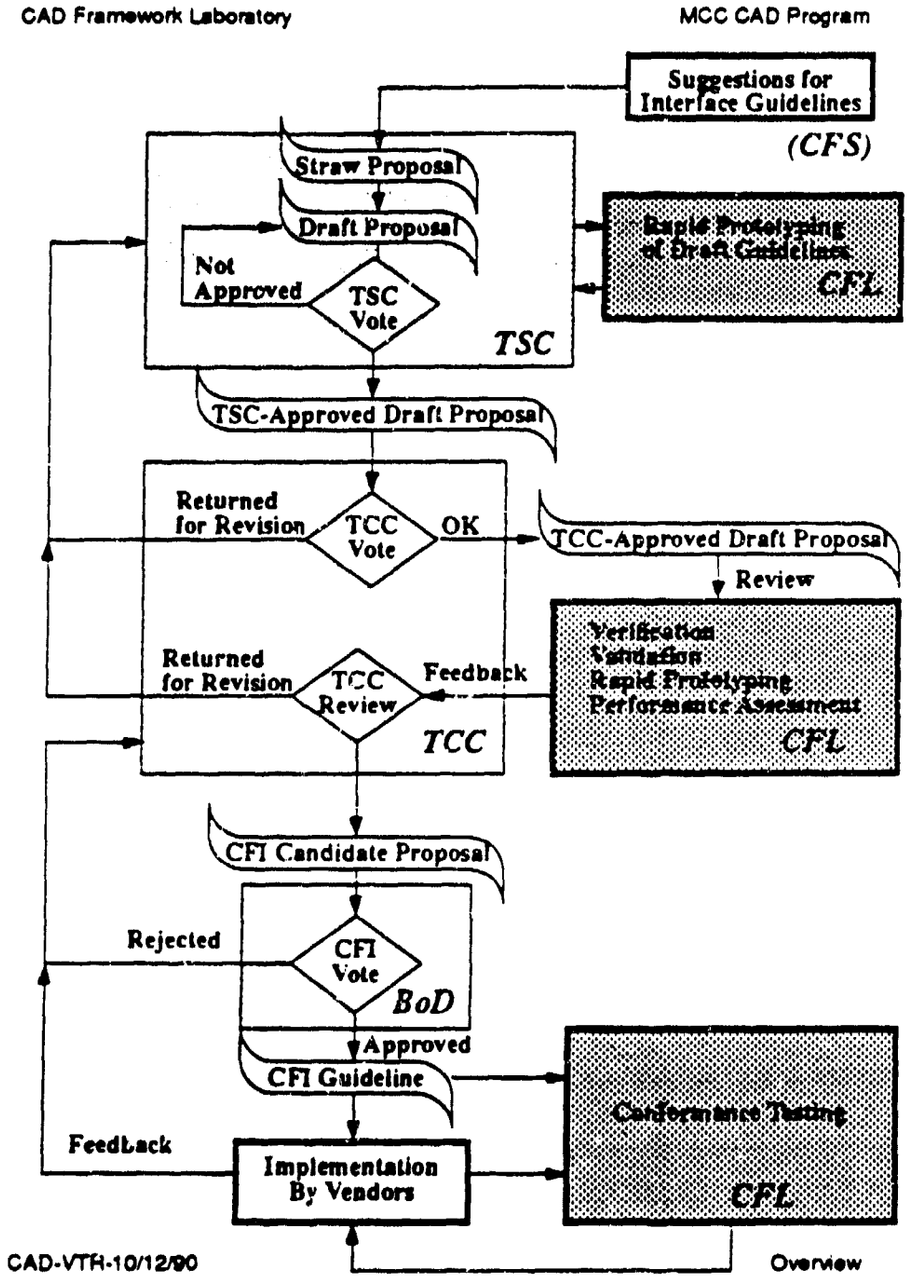


What is the CFI?

- *A Not-for-profit Corporation with Open Membership*
- *Supported by 38 Companies at Initial Meeting in May of 1988*
- *Pro Tem Steering Committee Elected at That Initial Meeting to Launch the CFI*
- *Committed to Development, Publication, and Adoption of Interface Standards for CAD Systems*

Mission: "to define interface standards that facilitate integration of design automation tools for the benefit of end users and vendors worldwide."

591


CAD Framework Laboratory
MCC CAD Program
Suggestions for Interface Guidelines
(CFS)
Straw Proposal
Draft Proposal
Not Approved
TSC Vote
TSC
Rapid Prototyping of Draft Guidelines
CFL
TSC-Approved Draft Proposal
Returned for Revision
TCC Vote
OK
TCC-Approved Draft Proposal
Review
Verification
Validation
Rapid Prototyping
Performance Assessment
CFL
Returned for Revision
TCC Review
Feedback
TCC
CFI Candidate Proposal
Rejected
CFI Vote
BoD
Approved
CFI Guideline
Conformance Testing
CFL
Feedback
Implementation By Vendors
CAD-VTH-10/12/90
Overview

Environment Users Presentation

by

T.R.J. Shorrock

D.S. Robinson

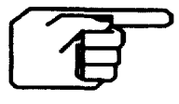
Content

What Aerospace has currently got.

What Aerospace needs in the future.

What Aerospace is doing about their needs.

Content



What Aerospace has currently got.

What Aerospace needs in the future.

What Aerospace is doing about their needs.

What Aerospace has currently got.

Product Management

- Item Types
- Version Management
  - integrated with tool invocation
  - multi-level versions
- Variant Management
- Configuration Management

PCIS - Environment Users

What Aerospace has currently got.

Organisation Management

- Domains
  - private workspace
  - restricted command sets to create roles
  - holds ownership of items
- Items shared between domains via Team Working

PCIS - Environment Users

What Aerospace has currently got.

Integrated Team Working

- Support for distributed teams
- Item publication to other domains
- Task Management between domains
  - Integrated with Version, Variants & Configurations
- Change Control and Review Process via electronic forms

PCIS - Environment Users

European Aerospace User Requirements

1 - PCIS should raise Level of Abstraction and provide Common User Services.

2 - PCIS should ensure the Framework fulfils todays Minimum User Requirements.

PCIS - Environment Users

Content

- What Aerospace has currently got.
- What Aerospace needs in the future.
- What Aerospace is doing about their needs.

What Aerospace needs in the future

- Traceability through levels of refinement
- Standard for tools: plug/unplug and retain access to data
- Standard for data: allow complete updates of environment
- To be supported over very long timescales

European Aerospace User Requirements

3 - PCIS should encourage Continuous User Input.

- Additional User Requirements/Experiences.
- Don't go away for 10 years!

Content

What Aerospace has currently got.

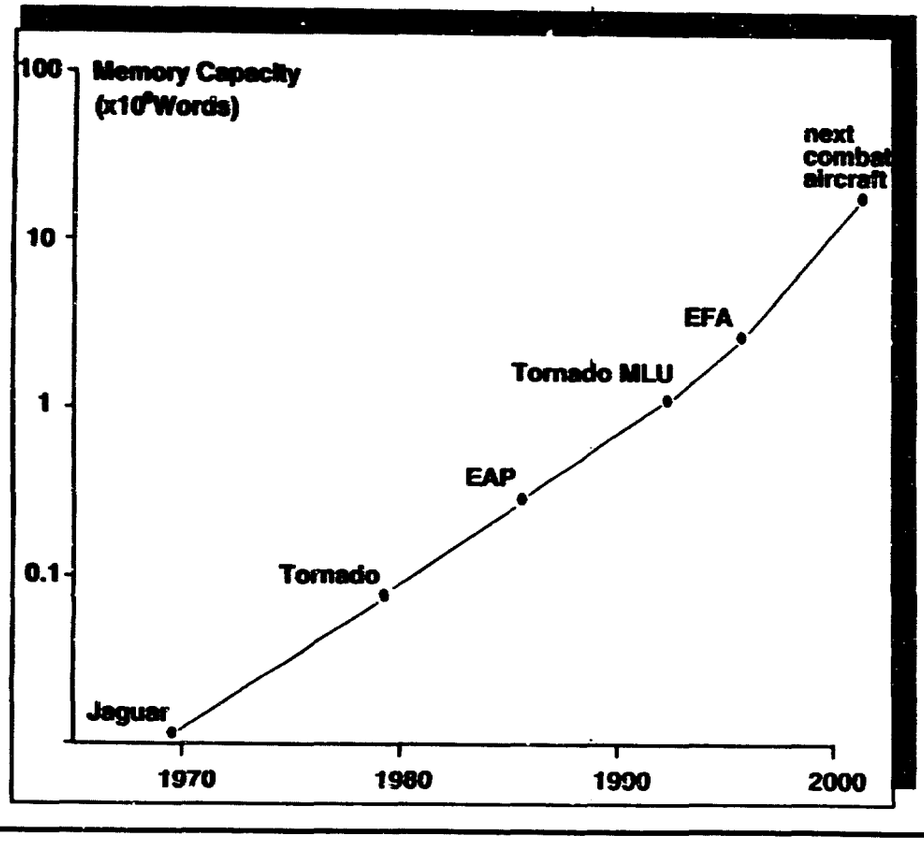
What Aerospace needs in the future.



What Aerospace is doing about their needs.

What Aerospace is doing about their needs

The Problem



- Software Growth
- Increasing System Complexity
- Improved Quality
- Improved Dependability
- International Collaboration
- Fixed-Price Contracting

PCIS - Environment Users

What Aerospace is doing - AIMS

Aerospace Intelligent Management and Development Environment for Embedded Systems

The Consortium:

Aerospatiale	**(CISI, CERT)**
Alenia	**(TXT)**
British Aerospace	**(SD-Scicon)**
[MBB]	
[CASA]	

PCIS - Environment Users

What is AIMS?

AIMS

- Industrial Research Project
- EUREKA Project
- User Driven
- Application Domain is ECS Development & Maintenance.

Aerospatiale, Alenia, British Aerospace

What Aerospace is doing - AIMS

Strategic Objectives

- Improve Productivity of the ECS Development Process.
- Stabilise the Time Schedule of ECS Development.
- Provide for the Effective Cooperation of the European Aerospace Industry.
- Improve Distinctly ECS Quality.

PCIS - Environment Users

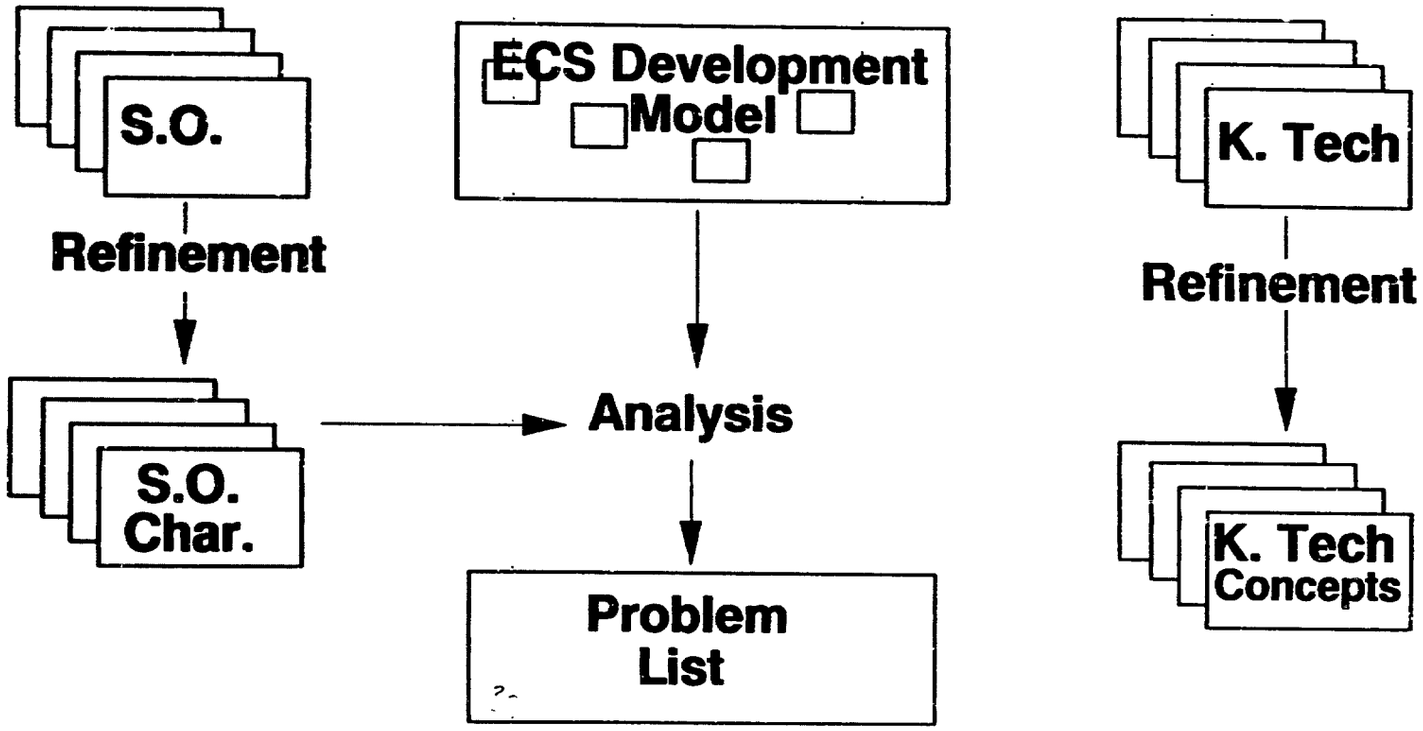
What Aerospace is doing - AIMS

Key Technologies

- Formal Methods
- Prototyping
- HCI
- Collaborative Working
- Reuse
- Expert Assistance

PCIS - Environment Users

What Aerospace is doing - AIMS



PCIS - Environment Users

What Aerospace is doing - AIMS

Evaluation Approach

- Detailed analysis
  - Identify Problem
  - Selection of Solution
- Detailed Assessment
  - Quantative Evidence

PCIS - Environment Users

European Aerospace Evaluations

- Timescale - 21 Months
- Cost/Benefit of the Key Technology Concepts solving specific problems

PCIS - Environment Users

Process Model Focus

- ECS Development Model is a basis for:
  - Communication
  - Problem Analysis
  - Solution Evaluation
  - Improvement Measurement
- A Process Model could be used for Requirements Definition.

European Aerospace User Requirements

4 - PCIS should investigate how to bridge the communication gap.

- Process Models
- Information Models

What Aerospace is doing - AIMS

Architecture

- **Method Integration**

Exploitation

- **Adopt it.**
- **Influence it.**
- **Make it happen.**

PCIS - Environment Users

European Aerospace User Requirements

1 - PCIS should raise Level of Abstraction and provide Common User Services.

2 - PCIS should ensure the Framework fulfils todays Minimum User Requirements.

3 - PCIS should encourage Continuous User Input.

4 - PCIS Should investigate how to bridge the Communication Gap.

PCIS - Environment Users

Report of

The Environment Users Session

Los Angeles PCIS Workshop

June 7, 1991

Participants:

Tim Lindquist	ASU	Chair
Vic Stenning	Anshar	Co-Chair
Kevin Hackett	SofTech	Recorder
Dave Robinson	SD-Scicon	
John Leary	Martin Marietta Corp.	
Leigh Power	MCC	
Judy Kerner	Aerospace Corp.	
Daniel LaBass	Mitre	
Claudio Costa	Alenia	
Don Hartman	Hal Computers	
Claude Baudoin	SEMATECH	
Maria Fischaleck	IABG	
Frank Belz	TRW	
Robert Rankin	DRA RSRE	
Hugh Davis	ICL	
Gary McKee	McKee Consulting	
Tom Baker	Boeing	
Audrey Canning	ERA Technology Ltd.	
Tim Shorrock	British Aerospace	
Gary Pritchett	SofTech	
Andrew Sound	Naval Weapons Center	
Yawar Ali	BNR	
Andy Rudmik	SPS	

Review of Environment Users Session

1. Review of Session Objectives

2. Overview of the London Workshop and Resulting Requirements

3. Individual Requirements Presentations:

   Vic Stenning : Process driven requirements, evolution and lego interfaces

   John Leary : Role-based requirements analysis

   Judy Kerner : Needs for PCIS Administrative Services

   Claude Baudoin : The Requirements process; technology push vs. user pull

   Robert Rankin (for Morron) Stability, Systems Engineering and Benefits

   Audrey Canning : Identifying Methods and Tools

4. Mass Confusion, DISCUSSIONS

5. Formulate Areas of Requirements

6. Breakouts to Refine Requirements

Resulting User Requirements

1. Requirements on the PCIS Process
   Introducing Change
   Activities and Products

2. Requirements on the PCIS Products

Business Drivers

The programme will be successful only if it stimulates considerable investment (and therefore buy-in) in PCIS results from the user as well as vendor communities

Users are aiming for quality, productivity and cost improvements - with acceptable risks

These are achieved via improvements to the process

An integral part of this is environment improvement

The PCIS program must view itself as introducing change

Introducing Change

The PCIS programme should identify the specific user problems that it aims to address, and should show how its products can be deployed by users to overcome those problems

The PCIS programme must involve continual user participation, and must respond to user feed-back

The PCIS requirements process should examine other environment requirements efforts to provide input into PCIS requirements definition and the PCIS introduction process

Examples:

Euro-fighter detailed requirements,

MOD Study Input

PCIS should exploit established and emerging standards (de facto and de jure); it should employ such standards wherever possible, rather than defining its own new solutions

The PCIS programme should address the means whereby the desirability of environment improvement can be demonstrated to strategic decision makers

PCIS products must be designed for incremental adoption as part of evolutionary environment improvement; it should be possible to adopt a selected sub-set of the products

The PCIS programme should explicitly address the problem of reluctance to change, and take steps to increase the likelihood of actual use

The PCIS programme should provide a 'road map' to assist users in exploitation of its results; this should cover (among other things)

- identification,
- justification,
- initiation, and
- management of environment improvements

PCIS Activities & Products for Achieving Change

Activities	Products
Survey	Guides/Maps to Standards
Analysis	Documented PCIS benefits
Evaluation	Comparison Guides for Standards
Develop	PCIS-conforming vendor products
Prototype	Public Demos
Surveys	Industry Case Studies
Selection	PCIS-supported application domains
Analysis	Domain-specific user problem model
Modeling	Domain-specific process description

* Exploit established work including on-going PCIS work.

* An alternative set of products should provide comparable benefits.

PCIS Laboratories

"A potential PCIS process mechanism"

A set of laboratories modeled on CFI/CFL

- Commercially viable standards developed by vendors
- Vendors build (integrate) laboratory prototypes and demos
- User industry members identify target problems

An umbrella organization to coordinate PCIS laboratories

- Provide strategic planning
- Establish domain-specific labs

Requirements on the PCIS Products

The PCIS shall cover a systems engineering approach, including software, hardware and manufacturing.

The PCIS should support horizontal tools and horizontal services (eg. integration services and related stds.)

User Interface Management
Configuration Management
Presentation, data and control integration

"If a choice must be made, PCIS should focus on horizontal rather than PTI level services."

PCIS should allow tools to exploit emerging and existing standards. PCIS should not be limited by a specific set of standards, but rather support environments of tools that use appropriate standards.

PCIS shall provide a full range of administrative services, such as those typically associated with the super-user or system manager. This should include:

- user and group creation, deletion and mgmt;
- resource and device creation, deletion, and mgmt;
- access control mgmt;
- environment instance creation, deletion and repair.

PCIS shall not preclude that the development environment may also be the target environment.

Traceability and Fine-Grained Data: PCIS should provide for efficient representation and manipulation of fine-grained data and for traceability (connectivity) among fine grained items.

Examples:
- hypertext
- source statements being traceable to requirements

Federation: PCIS should support co-operation and integrity maintenance among multiple heterogeneous data repositories. For example, tools may have their own data repositories that could co-operate through federation.

Conclusions

1. Mindset must change to adopt an approach to PCIS that is user needs-pull oriented rather than technology-push.
2. Focus on change management, including clear demonstration of benefits
3. Build on established standards wherever possible, rather than defining new ones.
4. Provide for incremental and evolutionary adoption and use, not big bang.
5. Currently, users need a federation of heterogeneous repositories. Allow for, but don't require, a single homogeneous repository.
6. Although tool portability remains a requirement, interoperability and incremental adoption should be the primary focus.
7. Promote open integrated systems by providing for facilities, such as fine-grained and traceable information.

Report of The Environment Users Session

Dr. Timothy Lindquist

0. Introduction.

The Environment Users Working Group consisted of 23 participants representing a broad spectrum of commercial, military and academic interests. The group also included both European and US concerns. The working group met the entire day on Wednesday of the workshop, and for six hours on Thursday. The results of the working group were summarized in a plenary session of the PCIS meeting. The overheads for that presentation are included in the workshop report (as amended to reflect further working group discussions.) The report also includes those overheads of the individual presentations from the session.

The Users Working Group report contains the following sections:

0. Introduction

1. Description of the Sessions,

2. Individual Presentations,

3. Summarization of Discussions,

4. Requirements:
   - Requirements on the PCIS Process,
   - Requirements on the PCIS Product,

5. Conclusions,

6. Participants,

7. Overheads of Individual Presentations.

1. Description of the Sessions.

The Wednesday session of the Users Group began with introductions, and a presentation by the chair on the scope and objectives of the working group. The group was to examine ISPE requirements from theperspective of the Application Developer. Output from the workshop is to aid in the requirements validation activity of the PCIS program. The working group resolved that there are many classes of users that fall in the category of application developers. Organizations that develop applications represent one form of user. Individual users represent

another. Each has their own set of needs that may in many instances conflict. An orthogonal view can also be taken, in which users can be seen to range from manufacturers to IPSE tool vendors. Each of these may have separate distinct needs.

Vic Stenning reviewed the results of the Environment Users Session from the London meeting, using the overheads from the Users Working Group report from that meeting. (These are included in the London Workshop report.) The presentation included user input to the PCIS process, as discussed at the London meeting.

The floor of the working group was then opened for individual presentations relating to user requirements. These presentations are described in the next section of this report.

As a result of the presentations and general discussions, the group had collected roughly 20 requirements. These were divided into four areas:

- Technical requirements,
- Identification and Justification of benefit,
- Managing Change, and
- PCIS scope.

Participants divided into four groups to work on specific requirements. Roughly one half a day was spent in the break-out groups, for the purpose of formulating statements of the requirements and their rationale.

The group reconvened Thursday evening to review the results of the break-outs. Tim Lindquist presented the results of the working group in Friday morning's plenary session.

2. Individual Presentations.

Participants were invited to present their list of potential requirements. Five presentations were giving covering various aspects of PCIS development and PCIS products.

Vic Stenning: Process driven requirements, evolution and lego interfaces.

> A potential role for PCIS is to facilitate environment improvement via process driven evolution. PCIS should adopt a business strategy in developing PCIS that builds on existing standards. The most important PCIS products would provide a LEGO interface promoting of existing standards and promoting open integrated environments. PCIS should not be another PTI.

John Leary: Role-based requirements analysis.

> Developing requirements representative of environment users should be done using a role-based process. Identify a list of users roles, map a workflow among roles and identify problems from the roles a workflow.

Judy Kerner: Needs for PCIS Administrative Services.

> PCIS should provide administrative services for managing users, groups, and other super-user functions.

Claude Baudoin: The Requirements process; technology push/user-pull.

> SEMATECH has taken a process, roles and activities approach to identifying user needs. These include business management, information management, operations, process management, software development, application creation and process development.

Robert Rankin (for Morron) Stability, Systems Engineering and Benefits.

> In the personal view of MW Morron, users currently need stability of the PCTE interfaces and attention to a systems engineering approach (as opposed to a more restricted software engineering only approach).

Audrey Canning: Identifying Methods and Tools.

> The MOD environment study identified many needs that should be examined by PCIS. The process used to identify the needs would apply as well.

3. Summarization of Discussions.

A majority of the users discussions focused on the needs of the PCIS process itself. The group addressed the question, what should be done to carry out the PCIS program so that it provides maximum benefits to its users?

The group felt that the most important issues for the program lie in identifying problems PCIS can profitably address, justifying the solution approach, and introducing change to the users. To provide more concrete input to PCIS, the group formulated a list of activities and products that would carry out this business approach to PCIS. The suggestion is just that, a suggestion. The group felt that any equivalent approach deemed appropriate would be suffice. The working groups particular approach is centered on the PCIS laboratory concept in which vendors are introduced early in the development cycle. Doing so provides for early analysis and a head start in arriving at production quality implementations.

Business Drivers

The program will be successful only if it stimulates considerable investment (and therefore buy-in) from its vendor and user communities. Vendors will buy-in and users will adopt a product, only if they see the potential benefit to their respective organizations. Users are aiming for quality, productivity and cost improvements. These must be present within the constraints of acceptable risk, and must take into account the natural tendency to resist change. Quality environment improvements are achieved via improvements to the process. The environment and the process by which the environment is used are closely related. The PCIS program must view itself as introducing change from this perspective. PCIS must pay attention to the hurdles that must be overcome to introduce change.

Feeling that PCIS should be user need driven as opposed to technology pushed, formulated a set of requirements for the PCIS products. These represent the needs of application developers, both as individual users and as organizational users. In this area, focal themes seemed to emphasize openness, integration services, traceability through levels of refinement, employing existing standards technology, and services to support cooperation among tools.

4. Requirements.

4.1 Requirements on the PCIS Process (Introducing Change).

The PCIS program should identify the specific user problems that it aims to address, and should show how its products can be deployed by users to overcome those problems.

The PCIS program must involve continual user participation, and must respond to user feedback.

The PCIS requirements process should examine other environment needs identification efforts to provide further input into PCIS requirements definition and the PCIS introduction process.

Examples:

Euro-fighter detailed requirements,

MOD Study Input

PCIS should exploit established and emerging standards (de facto and de jour); it should employ such standards wherever possible, rather than defining its own new solutions.

The PCIS program should address the means whereby the desirability of environment improvement can be demonstrated to strategic decision makers.

PCIS products must be designed for incremental adoption as part of evolutionary environment improvement. It should be possible to adopt a selected sub-set of the products.

The PCIS program should explicitly address the problem of reluctance to change, and take steps to increase the likelihood of actual use.

The PCIS program should provide a 'road map' to assist users in exploitation of its results; this should cover (among other things)

- identification,
- justification,
- initiation, and
- management of environment improvements

Sample PCIS Activities & Products for Achieving Change

This section presents more detail on one manner that the PCIS process can respond to these requirements. The approach is based on the notion of PCIS laboratories and it identifies specific activities (products) that should be carried out (developed.)

Activities	Products
Survey	Guides/Maps to Standards
Analysis	Documented PCIS benefits
Evaluation	Comparison Guides for Standards
Develop	PCIS-conforming vendor products
Prototype	Public Demos
Surveys	Industry Case Studies
Selection	PCIS-supported application domains
Analysis	Domain-specific user problem model
Modeling	Domain-specific process description

Guides and maps to standards are needed to help user organizations plan for future investments and introduction of PCIS-compliant environments. The comparison guide for emerging standards is for both vendor and user planning. It encourages standards convergence. The case studies, vendor public demonstrations and domain specific activity encourage adoption and investment.

The domain-specific process descriptions form the basis for project management and organizational process improvement. Public models are needed in this area to encourage introduction and process improvement.

The example put forward here is to organize a set of laboratories, modeled on CFI/CFL, along with an umbrella organization to oversee their coordination. It is assumed that the PCIS programme will produce a set of results, which specifically address user requirements and which exploit existing products and standards. The role of the laboratories in this process are outlined below.

PCIS Laboratories: A potential PCIS process mechanism

A set of laboratories modeled on CFI/CFL

- Commercially viable standards
- Vendors build (integrate) laboratory prototypes and demos
- User industry members identify target problems

An umbrella organization to coordinate PCIS laboratories

- Provide strategic planning
- Establish domain-specific labs

4.2. Requirements on the PCIS Products.

The PCIS shall cover a systems engineering approach, including software, hardware and manufacturing.

The PCIS should support horizontal tools and horizontal services (e.g. integration services and related standards.)

- o User Interface Management
- o Configuration Management
- o Presentation, data and control integration

If a choice must be made, PCIS should focus on horizontal rather than PTI level services.

PCIS should allow tools to exploit emerging and existing standards.
PCIS should not be limited by a specific set of standards, but rather support environments of tools that use appropriate standards.

PCIS shall provide a full range of administrative services, such as those typically associated with the super-user or system manager. This should include:

- user and group creation, deletion and management;
- resource and device creation, deletion, and management;
- access control management;
- environment instance creation, deletion and repair.

PCIS shall not preclude that the development environment may also be the target environment.

Traceability and Fine-Grained Data: PCIS should provide for efficient representation and manipulation of fine-grained data and for traceability (connectivity) among fine grained items.

Examples:

- hypertext
- source statements being traceable to requirements

Federation: PCIS should support co-operation and integrity maintenance among multiple heterogeneous data repositories. For example, tools may have their own data repositories that could co-operate through federation.

The PCIS process shall develop domain specific information and process models. Domains such as DOD 2167A and Aerospace should be addressed. These shall be addressed in the PCIS products. Such models are intended to promote interoperability.

The PCIS shall develop and support domain specific information interchange formats and protocols. These formats should support interoperability in heterogeneous environments.

5. Conclusions.

5.1. The mindset must change to adopt an approach to PCIS that is user needs-pull oriented rather than technology-push.

5.2. Focus on change management, including clear demonstration of benefits.

5.3. Build on established standards wherever possible, rather than defining new ones.

5.4. Provide for incremental and evolutionary adoption and use, not big bang.

5.5. Currently, users need a federation of heterogeneous repositories. Users need cooperation among tools. Allow for, but don't require, a single homogeneous repository.

5.6. Although tool portability remains a requirement, there are other overriding concerns, such as interoperability and incremental adoption.

5.7. Promote open integrated systems by providing for facilities, such as fine-grained and traceable information.

6. Participants.

Tim Lindquist	Arizona State Univ.	Chair
Vic Stenning	Anshar	Co-Chair
Kevin Hackett	SofTech, Inc.	Recorder
Dave Robinson	SD-Scicon	
John Leary	Martin Marietta Corp.	
Leigh Power	MCC	
Judy Kerner	Aerospace Corp.	
Daniel LaBass	Mitre	
Claudio Costa	Alenia	
Don Hartman	Hal Computers	

Claude Baudoin	SEMATECH
Maria Fischaleck	IABG
Frank Belz	TRW
Robert Rankin	DRA RSRE
Hugh Davis	ICL
Gary McKee	McKee Consulting
Tom Baker	Boeing
Audrey Canning	ERA Technology Ltd.
Tim Shorrock	British Aerospace
Gary Pritchett	SofTech, Inc.
Andrew Sound	Naval Weapons Center
Yawar Ali	BNR
Andy Rudmik	Software Productivity Solutions

PROCESS DRIVEN REQUIREMENTS, EVALUATION AND LEGO INTERFACES

VIC STENNING

ANSHAR

role of environment

to provide effective support
for an effective process

a role for PCIS

to facilitate environment improvement

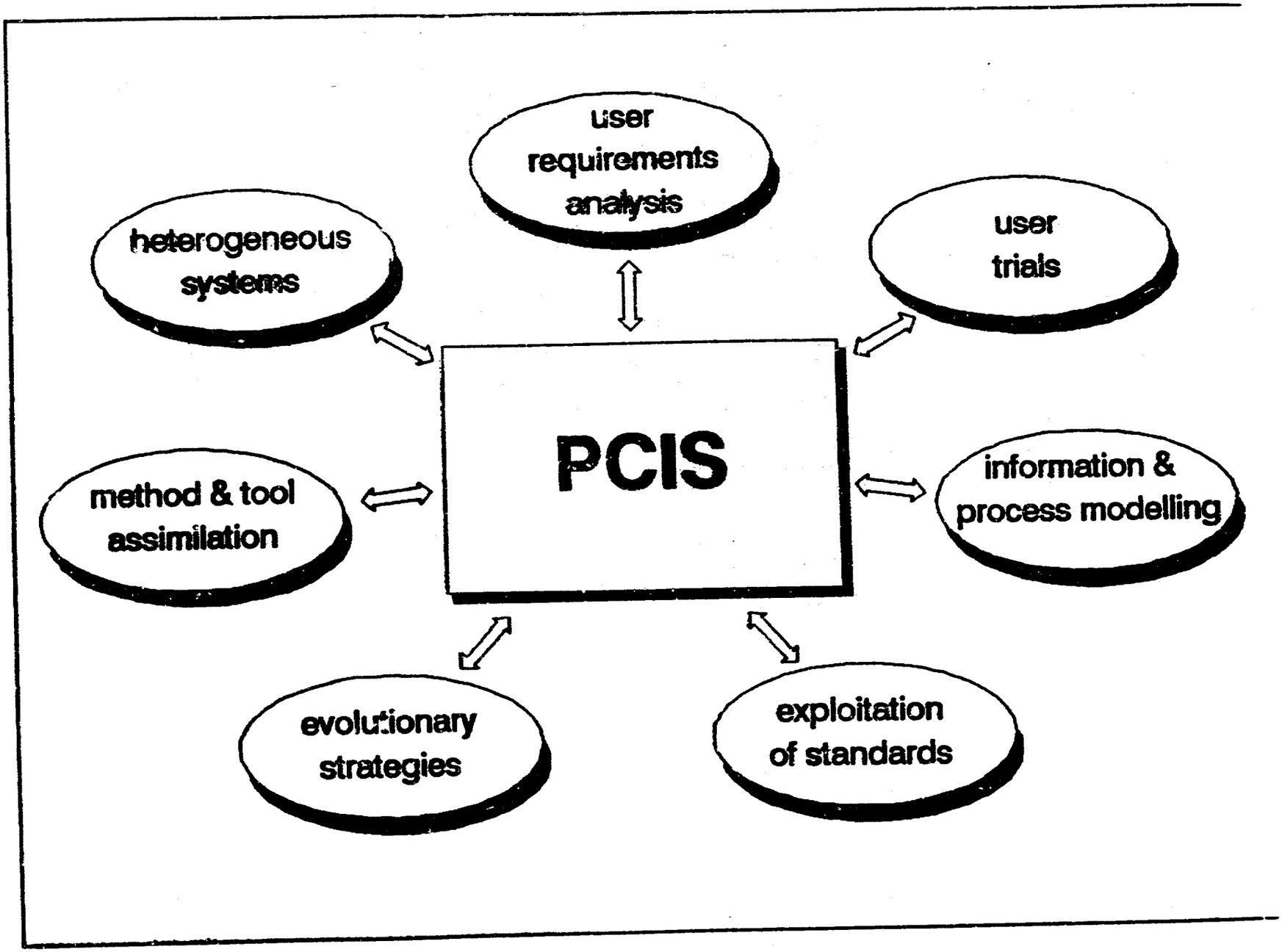
- process driven
- evolutionary

ANSHAR

not another PTI

644

ANSHAR


user requirements analysis
heterogeneous systems
user trials
PCIS
method & tool assimilation
information & process modelling
evolutionary strategies
exploitation of standards

ANSHAR

products

- 'LEGO' INTERFACE ??
- "Michelin Guide"
- demonstrators
- [standards / guidelines / procedures ...]

ANSHAR

ROLE-BASED REQUIREMENTS ANALYSIS

JOHN LEARY

MARTIN MARIETTA CORPORATION

"AD HOC" ½ HYPOTHESES

THE REAL ENVIRONMENT ISSUE IS

(1) " WHY BOTHER "

PCIS USER REQUIREMENTS MUST BE BASED ON

(2) CLEARLY ARTICULATED PROBLEMS

- anecdotes
- experiments
- history

THE "ELEPHANT" LOOKS DIFFERENT

(3) TO EACH DIFFERENT USER

"AD HOC" 2/2 PROPOSAL
for an
ENV. USER W.G. activity

(1) SUGGEST AN 'ENCOMPASSING' LIST OF ROLES

(2) ELABORATE A 'NOTION' FOR EACH ROLE

(3) MAP WORKFLOW BETWEEN ROLE

(4) IDENTIFY PROBLEMS IN
- CARRYING OUT ROLES
- REALIZING EFFICIENT WORK FLOW

NEEDS FOR PCIS ADMINISTRATIVE SERVICES

JUDY KERNER

AEROSPACE CORPORATION

JUDY KERNER
5 JUNE 1991

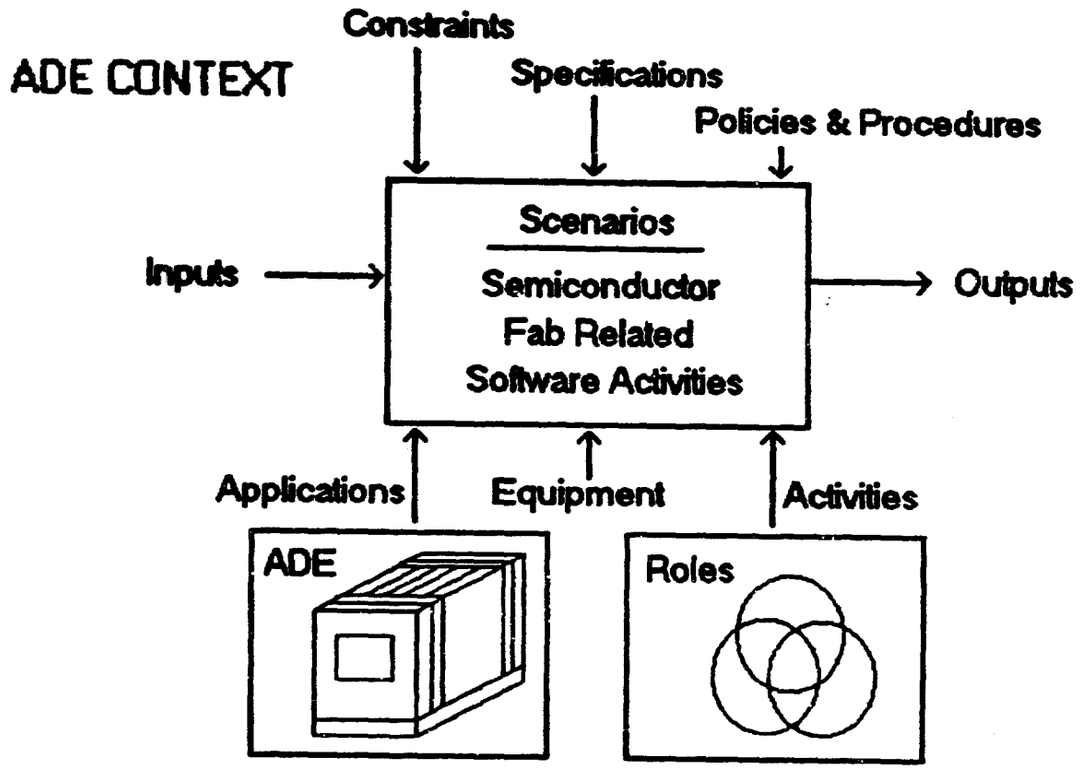
REQUIREMENT FOR ADMINISTRATIVE SERVICES

- PCIS MUST PROVIDE ADMINISTRATIVE SERVICES, SUCH AS:
  - START UP / SHUT DOWN / REPAIR ENVIRONMENT INSTANCES
  - ADD / DELETE USERS & GROUPS
  - ASSOCIATE / RE-ASSOCIATE USERS / GROUPS
  - OTHER "SUPER-USER" FUNCTIONS
- RATIONALE:
  - CAN'T ASSUME ENVIRONMENTS ARE ALL UP & RUNNING, CONFIGURED APPROPRIATELY - COMPLETENESS REQUIRES THESE SERVICES
  - CLAIM HAS BEEN MADE THAT THESE SERVICES ARE IMPLEMENTATION-DEPENDENT, BUT SOME HOST-INDEPENDENT ADMINISTRATIVE SERVICES ARE NOT ONLY POSSIBLE, BUT CRITICAL

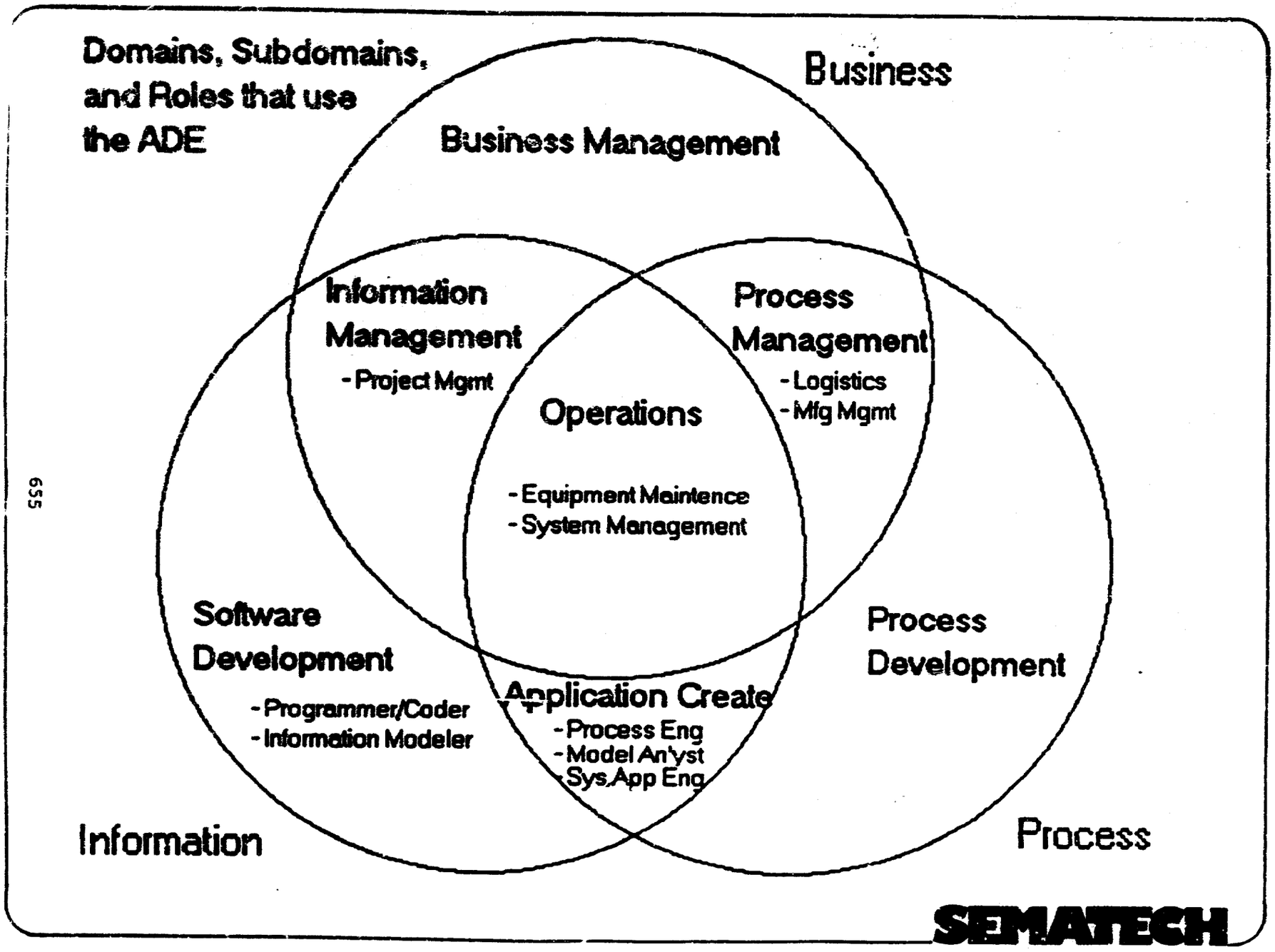
THE REQUIREMENTS PROCESS: TECHNOLOGY PUSH VERSUS USER PULL

CLAUDE BAUDOIN

SEMATECH


ADE CONTEXT
Constraints
Specifications
Policies & Procedures
Scenarios
Semiconductor
Fab Related
Software Activities
Inputs
Outputs
Applications
Equipment
Activities
ADE
Roles

SEMATECH


Domains, Subdomains, and Roles that use the ADE
Business
Business Management
Information Management
- Project Mgmt
Process Management
- Logistics
- Mfg Mgmt
Operations
- Equipment Maintence
- System Management
Software Development
- Programmer/Coder
- Information Modeler
Application Create
- Process Eng
- Model Anlyst
Sys App Eng
Process Development
Information
Process
SEMATECH

Scenario Content

Scenario Title
Purpose of Scenario:
Example of Scenario:

Role 1	**Activity 1** Description of Activity 1 **Activity 2** Description of Activity 2	**Objects associated with Activity 1 & 2** List of Objects
Role 2	**Activity 3** Description of Activity 3 **Activity 4** Description of Activity 4	**Objects associated with Activity 3 & 4** List of Objects

SEMATECH

STABILITY, SYSTEMS ENGINEERING AND BENEFITS

ROBERT RANKIN
for
M. MORRON

DRA RSRE

Tool User Needs of PCIS

A Personal View

M.W. Morron
BNR Europe Ltd

Stability

* Do not redefine any interfaces
* Allow five years for experience
* If needed, identify complementary interfaces
* Original PCIS plan would inevitably perpetuate itself

Systems Engineering

* Need common repository for hardware and software design tools
* Need higher level system tools
* Need for common project management and configuration management

Realise Benefits

* Many companies have invested heavily in PCTE over 8 years

* Need stability for more tool investment

* Need return benefits

Summary

* **Stability**

* **Address Systems Engineering**

* **Realise Benefit**

IDENTIFYING METHODS AND TOOLS

AUDREY CANNING

ERA TECHNOLOGY LTD.

TO RECOMMEND

SETS OF S/W ENGINEERIN

METHODS AND TOOLS

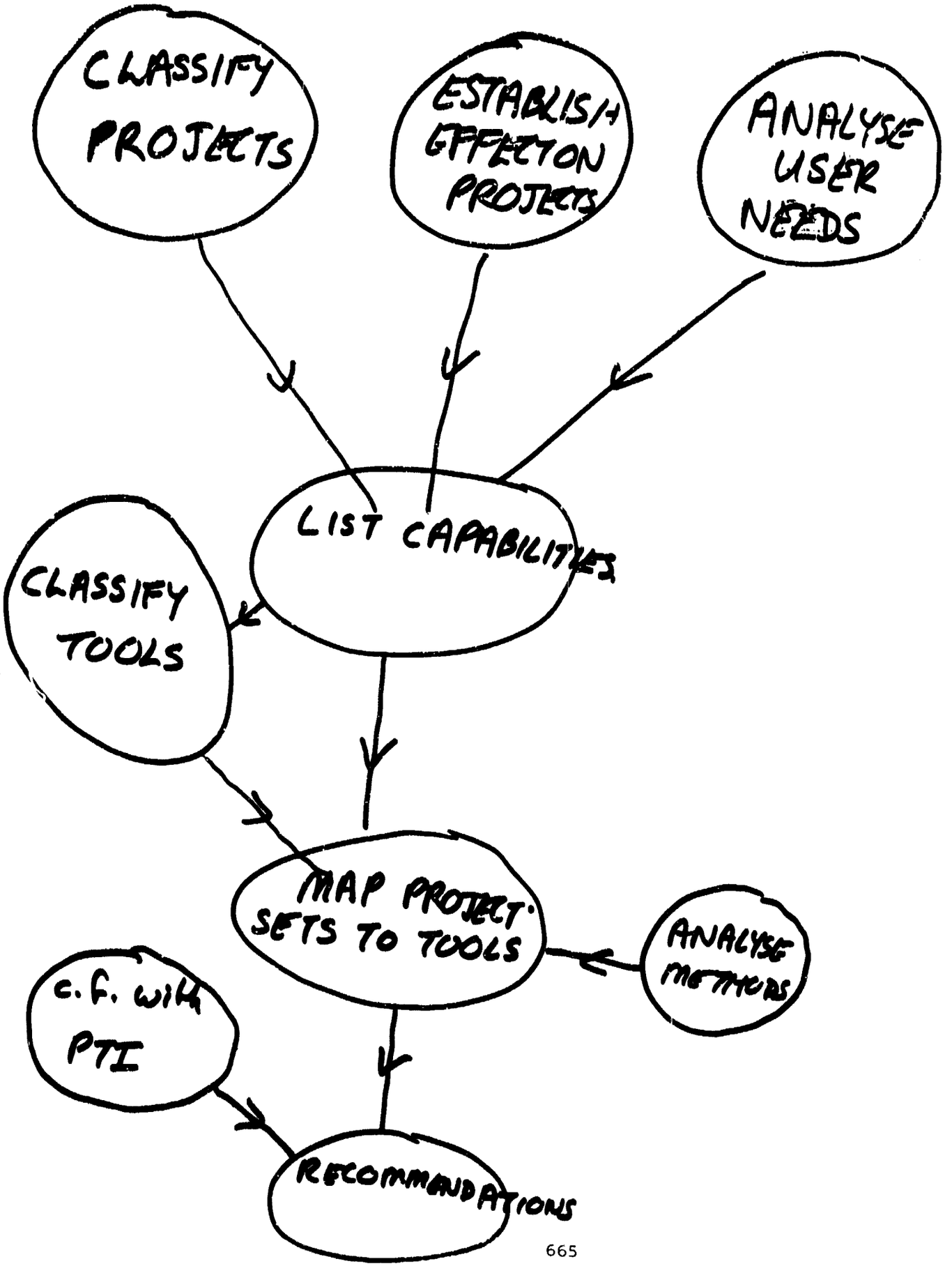
WHICH WOULD SATISFY

THE REQUIREMENTS OF

DEFENCE PROJECTS OF

VARYING SIZE AND TYPES

To develop PSE Policy for Production of s/w Intensive projects


CLASSIFY PROJECTS
ESTABLISH EFFECTON PROJECTS
ANALYSE USER NEEDS
LIST CAPABILITIES
CLASSIFY TOOLS
MAP PROJECT SETS TO TOOLS
ANALYSE METHODS
c.f. with PTI
RECOMMENDATIONS

Classify projects
analyse user needs

- project
- pse
- pti

traceability

many → many mapping.

List capabilities - many dimensional
no best implementation - all offer user capabilities
'ECMA Ref model.'

Classify tools - what are they - no specs, little info

Map project sets & tools - [illegible] with granularity
2 levels - management
- system behaviour

Sheraton at Redondo Beach
HOTEL

300 NORTH HARBOR DRIVE REDONDO BEACH CALIFORNIA 90277 2552
PHONE (213) 318 8888 FAX (213) 376 1930

- Tools grouped by function
- Tools avoid functional overlap
- No single PSE or combinations of PSE entities can satisfy all capabilities
- Tools based on specified methods

- MOD not standardise on particular methods / method sets. - premature
- Use standards to improve support available
- PTI can form suitable basis for this - ∴ PSE policy to go forward based on PTI.

- All projects ideally want all capabilities (fine grained analysis)

- Re-evaluation at intervals through life

- Integration very important - but not available

- Requirements of user - mapped to PSE + PTI

multistandards	safety critical
k.b. PSE	formal methods
international collab	documentation
security	AI systems
multisite	real time target tech
multi-language	lifecycle paradigms
diversity of dev. methods	tool interfaces
	effect of PTI

- List of requirements available - here.

PCIS REQUIREMENTS VALIDATION
SECOND WORKSHOP REPORT

Environment Suppliers Working Group

Attendees	6/05/91 AM	6/05/91 PM	6/06/91 PM
G. Clow (SofTech)	X	X	X
R. Minot (GIE Emeraude)	X	X	X
C. Hitchon (SofTech)	X	X	X
Y. Ali (BNR)	X		
E. Black (Atherton)	X	X	X
G. Boudier (GIE Emeraude)	X	X	X
C. Bremeau (GIE Emeraude)	X	X	X
K. Bruso (Unisys)	X	X	
J. Dawes (ICL)	X		
R. Ekman (IBM)	X	X	
K. Hayter (DRA)	X		
M. Horton (Unisys)	X	X	X
M. Kavianpour (IBM)	X		
C. Nolan (DEC)	X	X	X
H. Oliver (HP)	X	X	
G Memmi (Bull Hn)	X	X	X
J. Printz	X	X	
W. D. Song	X	X	

Includes many Platform Suppliers

- Even before that group disbanded.
- Platform Suppliers are Environment Suppliers.

Background Discussions

Discussion: Object-Orientation as major WS 1 issue.

- Need to refine WS1 consensus to integrate OO and ER.

- OO is an "emerged" technology.

- ER model is a complementary and proven approach in existing SEE data models.

Presentation: Chris Nolan, ATIS/PCTE merger experiment.

- PCTE metaschema extended with function attachment.
- Some OO interfaces and PCTE interfaces specified.
- Prospects for such a merger are encouraging.

Discussion: Partitioning as major WS 1 issue.

- Modularity of interface sets, incremental exploitation and future incremental extensibility.

Discussion: Interface Set Goals - Portability and Integration.

- Implies completeness, but not all interfaces for applications in general, and not monolithic interface set.

Requirements for Object Orientation

- Substantial discussion, and several trial proposals.

- Goal: Refine WS1 "4.8 Object Orientation" initial resolutions.

- Goal: Incorporate proven ER model concepts and basic object-oriented concepts seen as essential to tool integration and extensibility.

Requirement 4.8 1a) PCIS shall support mechanisms for the definition of abstract data types with operations.

Vote: for: 12, against: 1, abstentions: 1

WS1 Requirement 4.8 1b) PCIS shall support extensions of abstract data types with operations by subtyping (inheriting attribute types, relationship types, and operations).

Vote: for: 12, against: 1, abstentions: 1

Rationale:

- "Should" to "shall" to give more force.

- The phrase "plus methods" redundant with "abstract data type", included for clarity.

- "Inheriting..." emphasizes the intent of subtyping.

- The dissenting voter: the requirements did not go far enough; a detailed object-oriented model should be required.

- The abstaining voter: the requirements may already be too detailed.

Requirements for Object Orientation, continued

WS1 Requirement 4.8 1c, 1d and 2a through 2f should be deleted.

Vote: for 12, against: 0, abstentions: 0

Rationale:

- Unnecessarily specific requirements impinging on PCIS definition and on PCIS implementations.

- Little support for these requirements in the first workshop.

NRAC page 37 item (h).

Could be interpreted as incompatible with an object-oriented approach:

h) the NSIS shall separate the descriptions of data and instances of data from the tools that operate upon them.

Reword:

h) The PCIS shall separate the descriptions of data (including operations defined on the data) and the instances of data from the tools that manipulate them.

Vote: for: 6, against: 0, abstentions: 0

Rationale:

- Clarifies the intent that descriptions should reside in the model.
- Not in conflict with OO.

Use of "Ada" in the NRAC:

- PCIS definition should avoid language dependencies (except in bindings).

- Some occurrences of "Ada" in the NRAC need to be generalized.

Item (i) on page 37 of the NRAC:

i) The NSIS shall support Ada program libraries.

Rephrase as in EURAC to clarify that the intent is to provide a concrete example of one application domain:

i) The data facilities shall be sufficient to support, at least, Ada program libraries.

Vote: for: 6, against: 0, abstentions: 1

Alternative: move the requirement into the rationale:

Vote: for: 4, against: 1, abstentions 2

Rationale:

For - Requirements are already sufficiently strong to address the application to Ada program libraries.

Against - Stating applications is an appropriate approach to requirements specification.

Use of "Ada" in the NRAC, continued:

The editors of the IRAC should:

1) Analyze occurrences of "Ada". Retain language-specific references only when there is an underlying language issue. Refer to the EURAC in this process (since it has undergone this process). (Sections 2.5F and 2.11A are examples where references to Ada can simply be removed.)

2) Group together all language-specific requirements which are retained. (For example sections 2.14 contains such requirements.)

3) Investigate whether there are C-specific requirements which should be added.

Vote: for: 7, against: 0, abstentions: 0

Rationale:

- The PCIS abstract specification (and hence its requirements) should be language independent, where possible.

- Yet language capabilities should not be compromised.

Input/Output Requirements

Another occurrence of "Ada":

4.4E Ada Input and Output. The NSIS shall provide that the input and output facilities of the Ada language (as defined in Chapter 14 of ANSI/MIL-STD1815A) can be used in reading and writing attributes whose values correspond to Ada external file. The facilities of Section 6 shall then apply.

Rephrase as language- independent generalization:

4.4E Binding language features applying to file contents shall apply to entities having file contents.

Vote: for: 11 against: 2, abstentions: 0

Rationale:

- Vote addressed the statement of the generalization of the requirement, not desirability of the requirement itself.

Discussion opened the larger question of whether the PCIS should provide IO interfaces.

- If so, how should these be related to language-defined IO interfaces or run-time interfaces.

Input/Output Requirements, continued

Alternative:

1) PCIS shall provide basic IO operations on bulk data attributes (i.e. contents).

2) PCIS shall define the effect on the object base of standard IO facilities of a given language.

3) It is desirable that compilers support PCIS semantics in implementations of the standard language IO facilities.

Vote: for: 5, against: 0, abstentions: 0

Rationale:

- Requirement (3) is viewed as a design requirement for PCIS. That is PCIS should facilitate the implementation of such IO facilities.

- Language-independent standard IO facilities aid portability and provide a basis for the implementation of language-dependent IO facilities which interface with the PCIS object model.

- Permits e.g. security requirements and object base effects for file access.

- Requirement (2): compiling systems may be consistent in what effects IO facilities have on the object base.

- Requirement (1): offers the possibility of consistency of content structures across compiling systems.

2.6 Partitioning

Replacement 2.6 (NRAC & WS1) with refinements:

1) PCIS shall be partitioned.

2) Each partition consists of a set of services and an interface set, where the services and interface set are self-consistent.

3) Any given service may be a component of one and only one partition.

4) PCIS shall specify the interdependence of any pair of partitions.

5) PCIS shall define levels of conformance for implementations in terms of the provision of specific partitions.

Vote: for: 5, against: 0 abstentions: 0

Rationale:

- The desirability of partitioning is emphasized.

- Incremental exploitation (implementation, use by tools, installation) is facilitated.

- Modularity is enhanced.

- Services and interfaces are distinguished. In the object-oriented module, a single set of interfaces may provide access to many services.

- Redundancy of functionality is undesirable -- e.g. confusing and wasteful.

PCIS REQUIREMENTS VALIDATION
SECOND WORKSHOP REPORT
(Environment Suppliers Working Group)

DRAFT

Attendees	6/05/91 AM	6/05/91 PM	6/06/91 PM
G. Clow (SofTech)	X	X	X
R. Minot (GIE Emeraude)	X	X	X
C. Hitchon (SofTech)	X	X	X
Y. Ali (BNR)	X		
E. Black (Atherton)	X	X	X
G. Boudier (GIE Emeraude)	X	X	X
C. Bremeau (GIE Emeraude)	X	X	X
K. Bruso (Unisys)	X	X	
J. Dawes (ICL)	X		
R. Ekman (IBM)	X	X	
K. Hayter (DRA)	X		
M. Horton (Unisys)	X	X	X
M. Kavianpour (IBM)	X		
C. Nolan (DEC)	X	X	X
H. Oliver (HP)	X	X	
G. Memmi (Bull Hn)	X	X	X
J. Printz	X	X	
W. D. Song	X	X	

Introduction:

The Platform Suppliers Group was small and disbanded, in part because platform suppliers were interested in the environment suppliers group. Members of the disbanded group migrated to the Environment Suppliers Group. It was observed that platform suppliers are generally also environment suppliers.

Presentations:

Chris Nolan gave a presentation on an experimental ATIS/PCTE merger in which PCTE provided the underlying data model management services. The prospects for the success of such a merger were found to be encouraging.

Discussion: Object-Orientation.

The first workshop identified an object-oriented capability as a requirement. However, the ER model is a complementary and proven approach in existing SEE data models.

Discussion: Partitioning.

The first workshop identified partitioning of the interface set as a useful approach for modularity of interface sets, incremental implementation and future incremental extensibility.

Discussion: Interface Set Goals - Portability and Integration.

Portability implies completeness in the sense that all important services for tools must be provided. It was observed that such a requirement is far more limited than providing all interfaces for applications in general.

Tool integration requires the selection of existing standards already in use or planned for use. PCIS should avoid the implementation of that which is already available. Extensibility is required to support tool composition.

Requirements for Object Orientation:

Several general requirements were proposed concerning PCIS support for an object-oriented data model.

General Object-Oriented Requirement Alternatives:

1. PCIS shall have an object model.

   Vote: None.

   Rationale:

   Such a requirement was seen as too vague, particularly the use of "object model".

2. PCIS shall be expressed through its object model.

   Vote: None.

   Rationale:

   Again, "object model" is vague and could be seen as constraining the standard to an "object oriented" description.

3. PCIS shall provide a data model which supports modeling of objects with types, type inheritance, attribute types, relationship types and operations applicable to instances of those types.

   Vote: for: 14, against, 0: abstentions: 0

   Rationale:

   - The requirement incorporates the proven ER model concepts of existing standards as well as basic object-oriented concepts seen as essential to tool integration and extensibility.

   - Although there was some disagreement with imposing a particular data model, it was also noted that some data model must be observed in the IRAC in order to adequately express the requirements. However, this does not require a particular data model for the PCIS.

Object-Oriented Issues:

The following were points of discussion. The issues are already addressed by NRAC and no motions for revision arose.

- Should inheritance be required on types other than object types (i.e., relationship types and attribute types)? Are relationships "first class" objects in the sense that attributes and operations can be associated with relationship types?

- Is multiple inheritance required?

- Are multiple implementations of methods permitted (as in abstract data types)?

4.8 Object Orientation

The alternative detailed requirements formulated in the first PCIS workshop were discussed. The first two requirements in alternative 1 were rephrased and then voted upon:

1a. PCIS shall support mechanisms for the definition of abstract data types with operations.

Vote: for: 12, against: 1, abstentions: 1

1b. PCIS shall support extensions of abstract data types with operations by subtyping (inheriting attribute types, relationship types, and operations).

Vote: for: 12, against: 1, abstentions: 1

Rationale:

- Rephrasing changed "should" to "shall" to give more force to the requirement.

- The phrase "plus methods" was replaced by "with operations", expressing essentially the same requirement without the implication of a commitment to a strictly object-oriented model. Although both phrases are redundant with "abstract data type", the later phrase was included for clarity.

- The phrase "Abstract Data Types" was changed to lower case to indicate that the general concept of abstract data types is referenced.

- The phrase "inheriting attribute types, relationship types and operations" was added to (1b) to emphasize the intent of subtyping.

- The dissenting voter felt that the requirement did not go far enough; that they do not really say what the data model must be, and that a detailed object-oriented model should be required.

- The absenting voter felt that the requirement may already be too detailed.

The requirements 1c, 1d and 2a through 2f were considered and their deletion was voted upon as a group:

Vote: for: 12, against: 0, abstentions: 0

Rationale:

- These requirements were generally seen as unnecessarily specific and tending to require object-oriented implementation techniques.

- There was relatively little support for these requirements in the first workshop.

NRAC page 3'' item (h).

It was pointed out that the following requirement in the NRAC could be interpreted as incompatible with an object-oriented approach:

h) the NSIS shall separate the descriptions of data and instances of data from the tools that operate upon them.

The following rewording was suggested:

h) The PCIS shall separate the descriptions of data (including operations on the data) and the instances of data from the tools that manipulate them.

Vote: for: 6, against: 0, abstentions: 0

Rationale:

The rewording clarifies the intent that the descriptions of data and operations should reside in the model and not be embedded within the tools which use the model. The rewording is not in conflict with an object-oriented model.

Use of "Ada" in the NRAC:

It was observed that a goal of the IRAC and the PCIS definition is to avoid language dependencies (except in the language bindings) and that some occurrences of "Ada" in the NRAC may need to be generalized.

Item (i) on page 37 of the NRAC is:

i) The NSIS shall support Ada program libraries.

A rephrasing of this statement in the EURAC was proposed to clarify that the intent is that the database facilities should be sufficient to support the implementation of Ada program libraries:

i) The data facilities shall be sufficient to support, at least, Ada program libraries.

Vote: for: 6, against: 0, abstentions: 1

Rationale:

- Specific mention of languages should be confined to areas where language binding is discussed. However, in this case no language dependency is described. The requirement merely uses the implementation of Ada program libraries as an example of capabilities that must be provided through the PCIS.

An alternative proposal was to move the requirement into the rationale.

Vote: for: 4, against: 1, abstentions: 2

Rationale:

For: The requirement itself is real. That is, the data model requirements are already sufficiently strong to address the application to Ada program libraries. This application could simply be mentioned in the rationale for data modeling requirements,

Against: Stating concrete requirements with examples is an appropriate approach to requirements specification.

Further discussion of the issue of the use of "Ada" in the NRAC lead to the following recommendations to the editors of the IRAC:

1. Analyze occurrences of "Ada". Retain language-specific references only when there is an underlying language issue. Refer to the EURAC in this process (since it has undergone this process). (Sections 2.5F and 2.11A are examples where references to Ada can simply be removed.)

2. Group together all language-specific requirements which are retained. (For example, section 2.14 contains such requirements.)

3. Investigate whether there are C-specific requirements which should be added.

Vote: for: 7, against: 0, abstentions: 0

Rationale:

These recommendations follow directly from the desire for language-independence in the PCIS abstract specification. They also point out that the investigation of potential language-dependent issues (such as implementation problems) needs to begin in the formulation of requirements and continue through the definition phases.

Input/Output Requirements.

Another occurrence of "Ada" in the NRAC is requirement 4.4E on page 54:

> **4.4E Ada Input and Output.** The NSIS shall provide that the input and output facilities of the Ada language (as defined in Chapter 14 of ANSI/MIL-STD-1815A) can be used in reading and writing attributes whose values correspond to Ada external files. The facilities of Section 6 shall then apply.

An alternative phrasing of requirement was proposed as a language-independent generalization of NRAC requirement.

> **4.4E** Binding language features applying to file contents shall apply to entities that have file contents.

Vote: for: 11, against: 2, abstentions: 0

This vote addressed the statement of the generalization of the requirement rather than the desirability of the requirement itself.

The rephrasing has actually opened the larger question of whether the PCIS should provide IO interfaces, and if so, how these should be related to language-defined IO interfaces or run-time interfaces which may be provided as part of the platform on which the language is used. During this discussion the following IO related requirements were formulated and voted upon. These requirements replace and refine 4.4E above:

1. PCIS shall provide basic IO operations on bulk data attributes (i.e., contents).

2. PCIS shall define the effect on the object base of standard IO facilities of a given language.

3. It is desirable that compilers support PCIS semantics in implementations of the standard language IO facilities.

Vote: for: 5, against: 0, abstentions: 0

Rationale:

- Requirement (3) is viewed as a design requirement for PCIS. That is, PCIS should facilitate the implementation of such IO facilities.

- Language-independent standard IO facilities are an aid to portability. They also provide a basis for the implementation of language-dependent IO facilities which interface with the PCIS object model.

- Both PCTE and CAIS provide IO interfaces. A key rationale for these interfaces is that additional semantics regarding the effects of IO operations on entities having contents are defined by both standards. Also, the management of security requirements regarding file access is necessary. This may present implementation problems or require the copying of files (e.g., export/import) prior to and/or following access.

- Requirement (2) assures that compiling systems will be consistent in what effects IO facilities have on the object base. Requirement (1) offers the possibility of consistency of content structures across compiling systems.

- PCTE provides primitive IO operations which allow the implementation and substitution of run-time IO interfaces which are integrated with PCTE object management. This solution is often viable when the desired compiling system support mentioned in (3) is not available.

- It is unlikely that compiler suppliers will have much interest in customizing language IO operations to interface with PCIS. However, on some platforms there may be hooks into the operating system for reporting IO operations or it may be possible to substitute PCIS integrated IO interfaces for those normally called using the linker. There is no assurance that the necessary facilities or hooks will always be available.

Partitioning:

Partitioning of the PCIS interface set was discussed. The general feeling was that partitioning is desirable. The following requirements were proposed as replacements for those in section 2.6 proposed in the first workshop and those included in the NRAC.

1. PCIS shall be partitioned.

2. Each partition consists of a set of services and an interface set, where the services and interface set are self-consistent.

3. Any given service may be a component of one and only one partition.

4. PCIS shall specify the interdependence of any pair of partitions.

5. PCIS shall define levels of conformance for implementations in terms of the provision of specific partitions.

Vote: for: 5, against: 0, abstentions: 0

Rationale:

- The desirability of partitioning is emphasized. Although requirements for both CAIS-A and PCTE mention partitioning, no true partitioning occurs in these standards. Problems with monolithic interface sets were acknowledged. The entire interface set should not be forced upon applications which require only some small portion of the services available.

- Services and interfaces are distinguished. In the object-oriented model, a single set of interfaces may provide access to many services.

- Redundancy of functionality is undesirable. Redundancy tends to confuse and lead to wasteful multiplicity. Thus, multiple partitions should not provide the same services.

- The interdependencies must be specified by the PCIS in order that the benefits of partitioning can actually be exploited to minimize development costs and other resource requirements. The ability to partition the services must be designed into the specification of the services and their interrelationships.

- The services which cannot act independently are naturally grouped in the same partition.

Wording Suggestion:

It was suggested that the IRAC use terms such as "valid" or "invalid" in place of "legal" and "illegal".

Vote: none.

ATIS

A

Tools

Integration

Standard

History

- Original work done in a CAD environment
- Atherton Technology founded to develop the same ideas for CASE
- In partnership with Digital, concepts extended to:
  - Eliminate UNIX dependencies
  - Enhance relationship support
  - Modularize and generalize facilities
  - Add dictionary concepts
- CIS committee started to gather industry ideas

Goals

- Provide services that can support the entire application development life cycle

  Scope includes traditional CASE repositories and traditional data dictionaries

- Address the problem of growing numbers of disparate repository & dictionary standards

What is ATIS ?

- ATIS consists of the following:
  - O-O interface defining services for data integration
  - Set of Abstract Data Types exporting services useful to integrate CASE
  - An extensible type hierarchy
  - Inheritance of behavior
  - A set of object models
  - Does not prescribe an implementation

Structure

1. ATIS description broken out into "models"
2. Each model describes a concept:
   A. Base Object Model
   B. Versioning
   C. Configuration Management
   D. Tools Integration
   E. Work Flow Control
   F. Dictionary Extensions

Base Object Model

Provides:

- objects, properties, methods
- a way of sending messages
- utility functions
- a simple type hierarchy

Refining Operations

Allows for the behavior of the system to be tailored to the site.

- Vendor operations must kept separate from site customization, so updates to the system leave site changes intact

.....

ATIS *preambles* and *postambles* allow customizations, and keep them distinct from vendor supplied behavior

.....

Version Model

- Introduces versions and variants
- Can create variants of variants; can merge variants
- Reserve/Replace mechanism controls concurrent creation of new versions
- Allows versioning of types

Configuration Model

- Introduces *configurations*
- A configuration represents all elements of a particular version of a system
- Configurations change when their contents change
- Configurations are changed via reserve/replace
- *Contexts* represent views of the repository
- *Aggregates* represent elements external to the repository

Tool Integration

- Allows new tools to extend the system
- Allows tools to make themselves known to ATIS
- Makes it possible for ATIS to invoke the tool's implementation of a method

Work Flow Model

- Controls configurations as they move through the system development and approval process
- Documents and enforces policy on stages of development

ATIS Object Model

0 Element
- 1 Named_Element
  - 2 Branch
  - 2 Context
  - 2 *IRD-VIEW*
  - 2 Database
  - 2 *NAMES*
  - 2 Partition
  - 2 Role
  - 2 User
  - 2 Version
    - 3 Aggregate
      - 4 Binary
        - 5 Binary Tool
        - 5 Text
          - 6 Text Tool
      - 4 Collection
    - 3 Message
    - 3 Tool
      - 4 Method
- 1 Relationship
  - 2 Depends-on
    - 3 *CONTAINS*
    - 3 *MEMBER-OF*
    - 3 *USES*

Meta Schema

0 Element

1 Named_Element

2 Version

3 Type

4 Data_Type

4 Element_Type
5 RELATION_TYPE

4 PROPERTY_TYPE

CIS

- CASE Integration Services

- "The CIS committee will develop or adopt a set of standard interfaces that promote the integration of software engineering tools"

- An ad hoc group of interested CASE industry vendors and users

- Started by Digital and Atherton to consider ATIS

- Charter strictly limits activities to address integration issues

- CIS assumes the presence of portability services

- Now it has been proposed as an ANSI technical committee

Other Standards

- **IRDS:**
  ATIS has been submitted to ANSI and ISO committees

- **CFI:**
  ATIS can be applicable to CAD

- **OMG:**
  ATIS could be implemented using the ORB

- **CDIF:**
  could be used by tools within an ATIS compliant environment

- **POSIX:**
  ATIS should be implementable on top of a POSIX compliant system

ATIS vs. PCTE

- ATIS in some areas is a
  - subset of PCTE (SEC,EXE,ACT)
  - superset in other (OMS, SMS)
- Overlap in the base type hierarchy (e.g. file concept) and some common concepts (versioning, configuration)
- PCTE does not provide facilities to define the behavior of the object types
- ATIS is less descriptive than PCTE in the relationships between types

==> *merge to get the best of both worlds*

ATIS on PCTE

Offer an ATIS interface, built using the PCTE facilities.

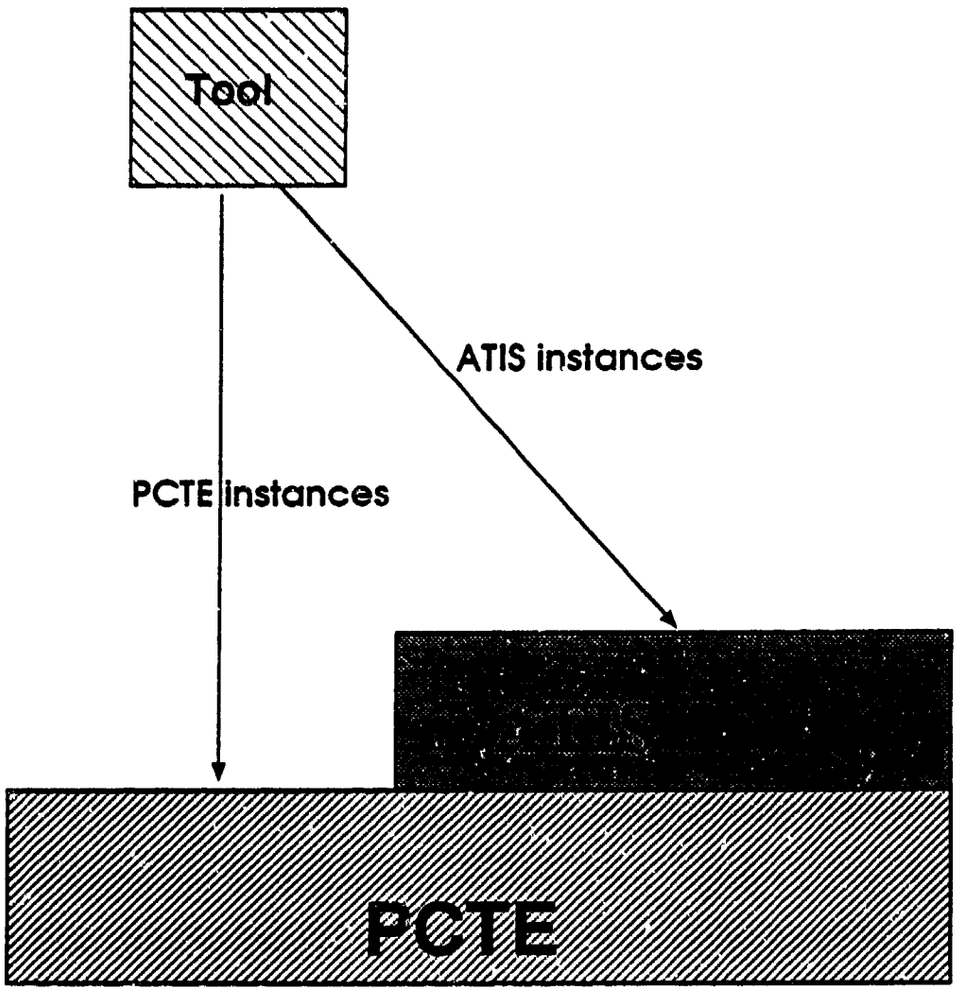
Pros:

- Get distribution, security, concurrency 'for free'
- Create a portable implementation across multiple platforms

Cons:

- Instances must be created and accessed from the same interface: PCTE or ATIS
- Does not solve the 'data model' problem: one repository, two data models

ATIS on PCTE



ATIS and PCTE

- PACE => PCTE
  ATIS
  Common
  Environment

- Merge the implementation: introduce messages and methods for PCTE types

- Merge the data model: where scopes are the same bring two object types into one

ATIS and PCTE

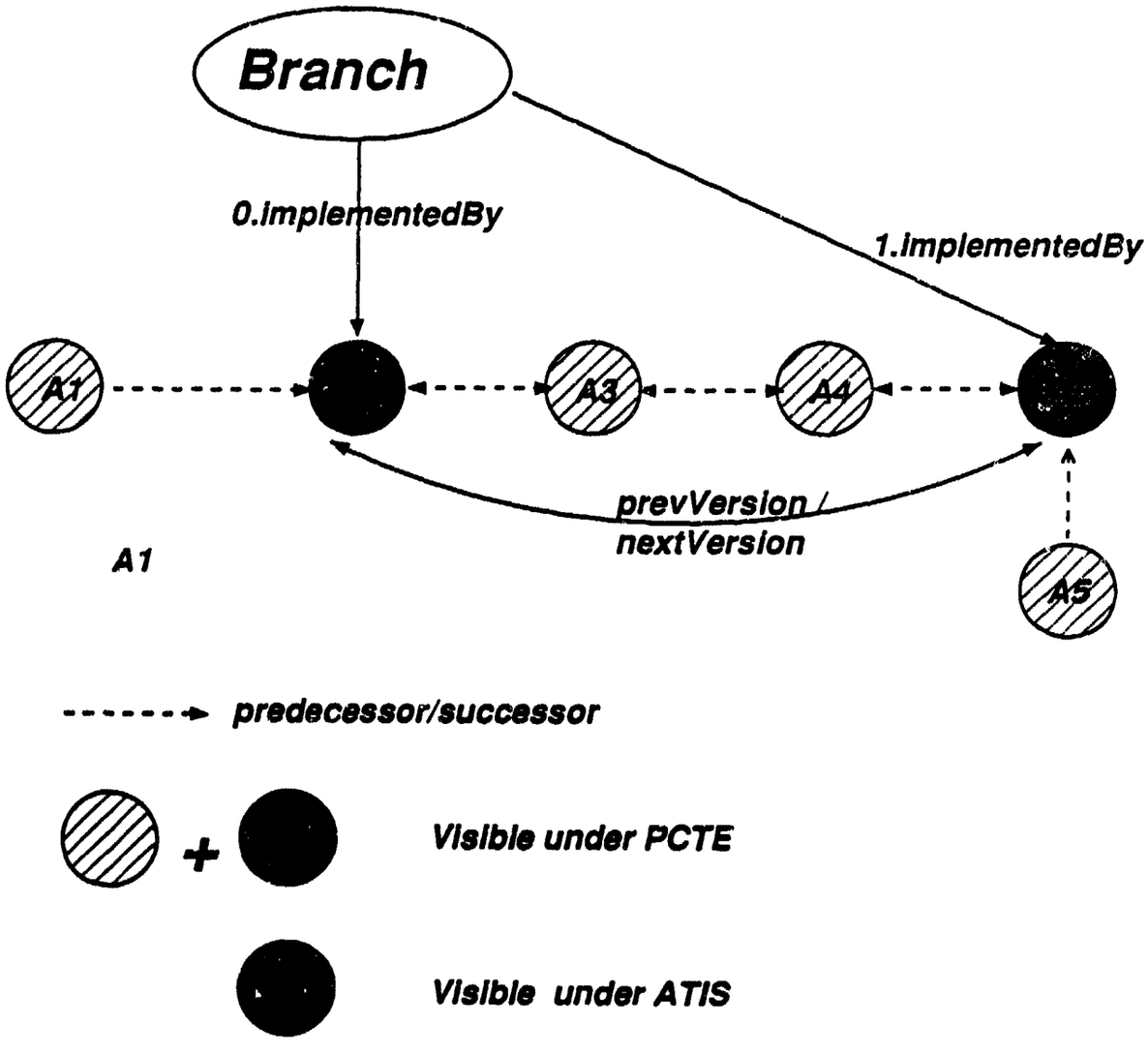
- Use PCTE links and SDSs to achieve different *views* of the common model

  ATIS has a restricted view of the PCTE object base

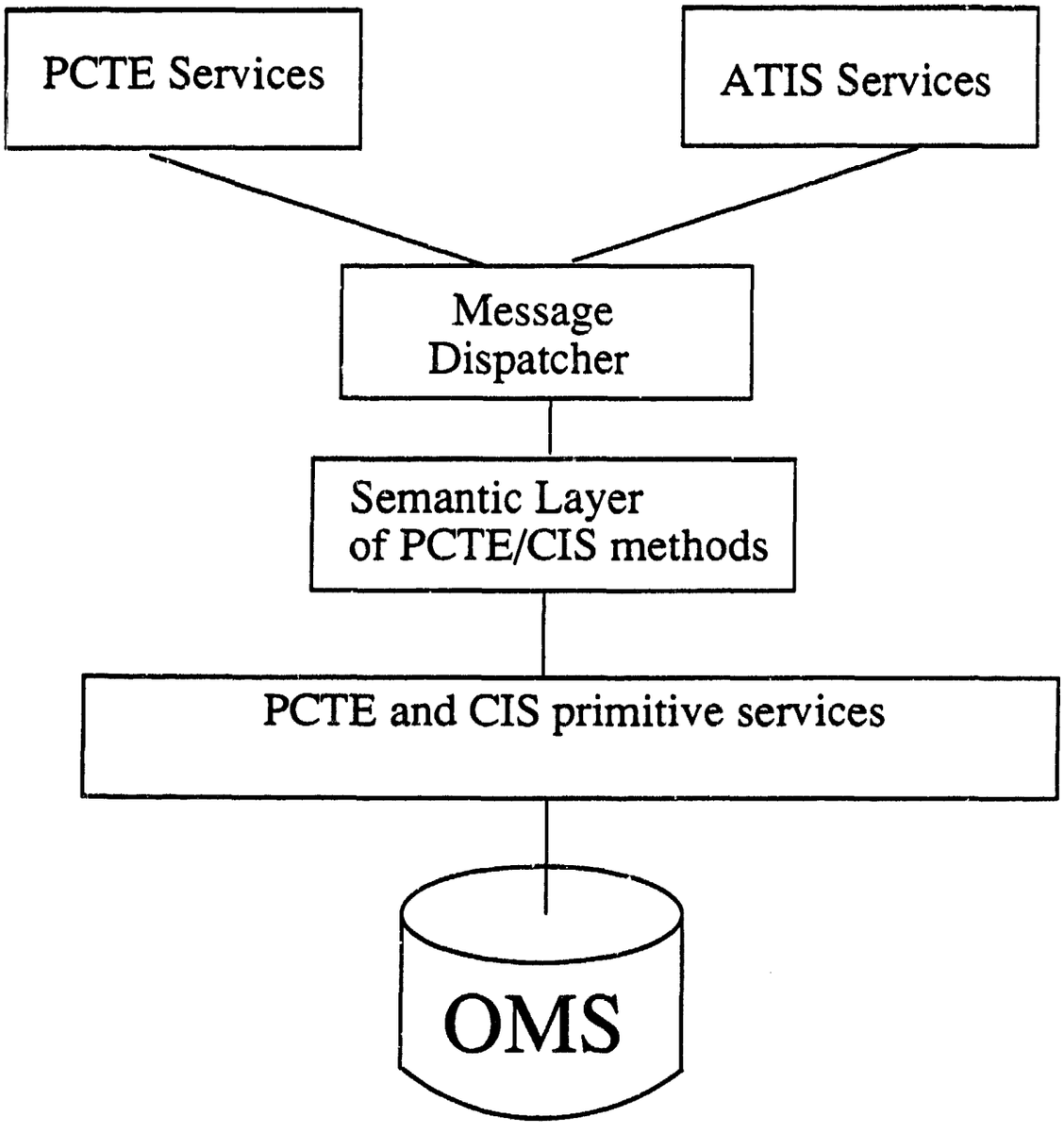
- Two interfaces and a merged data model

==> access any instance from both the interfaces

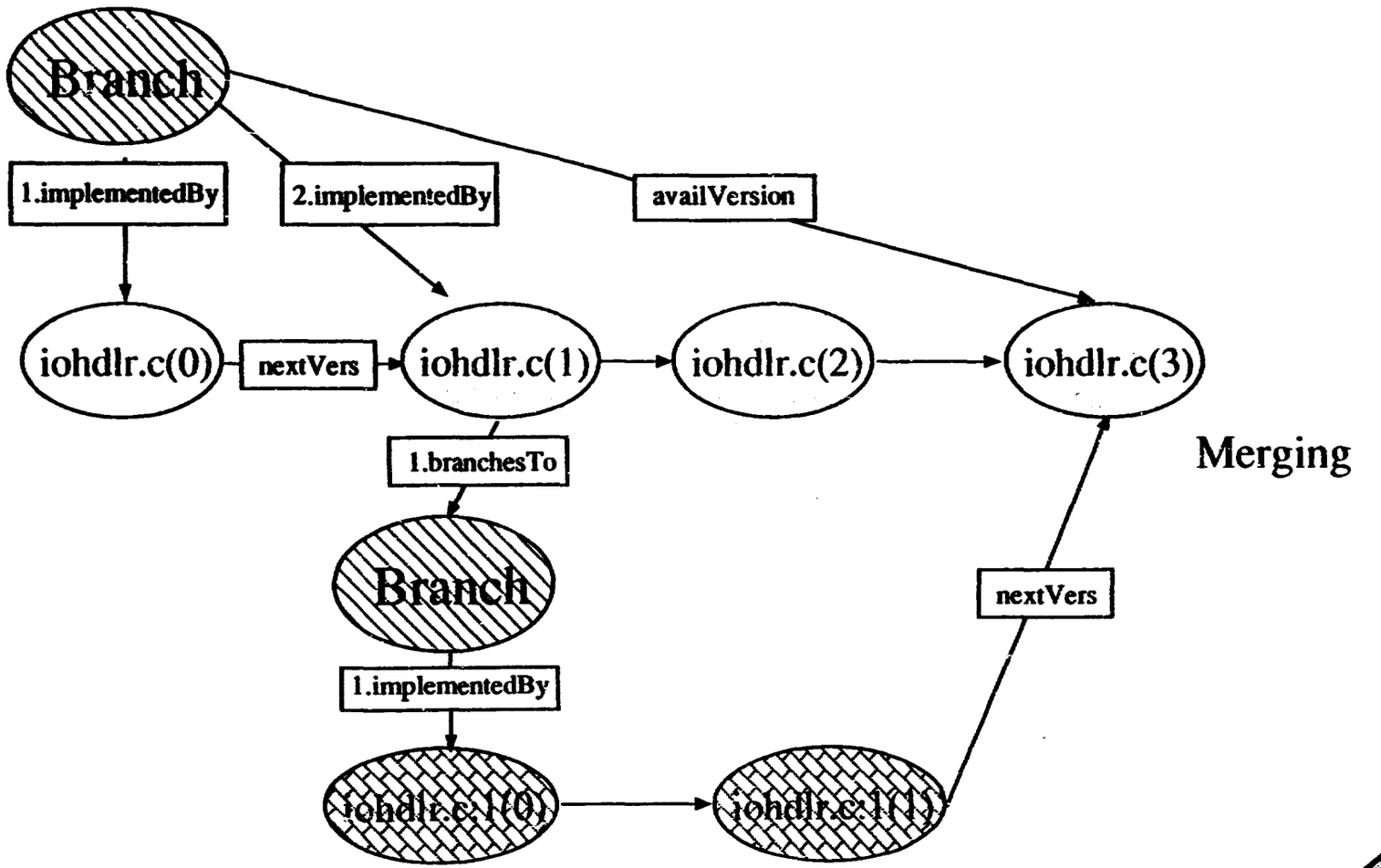
PACE Example

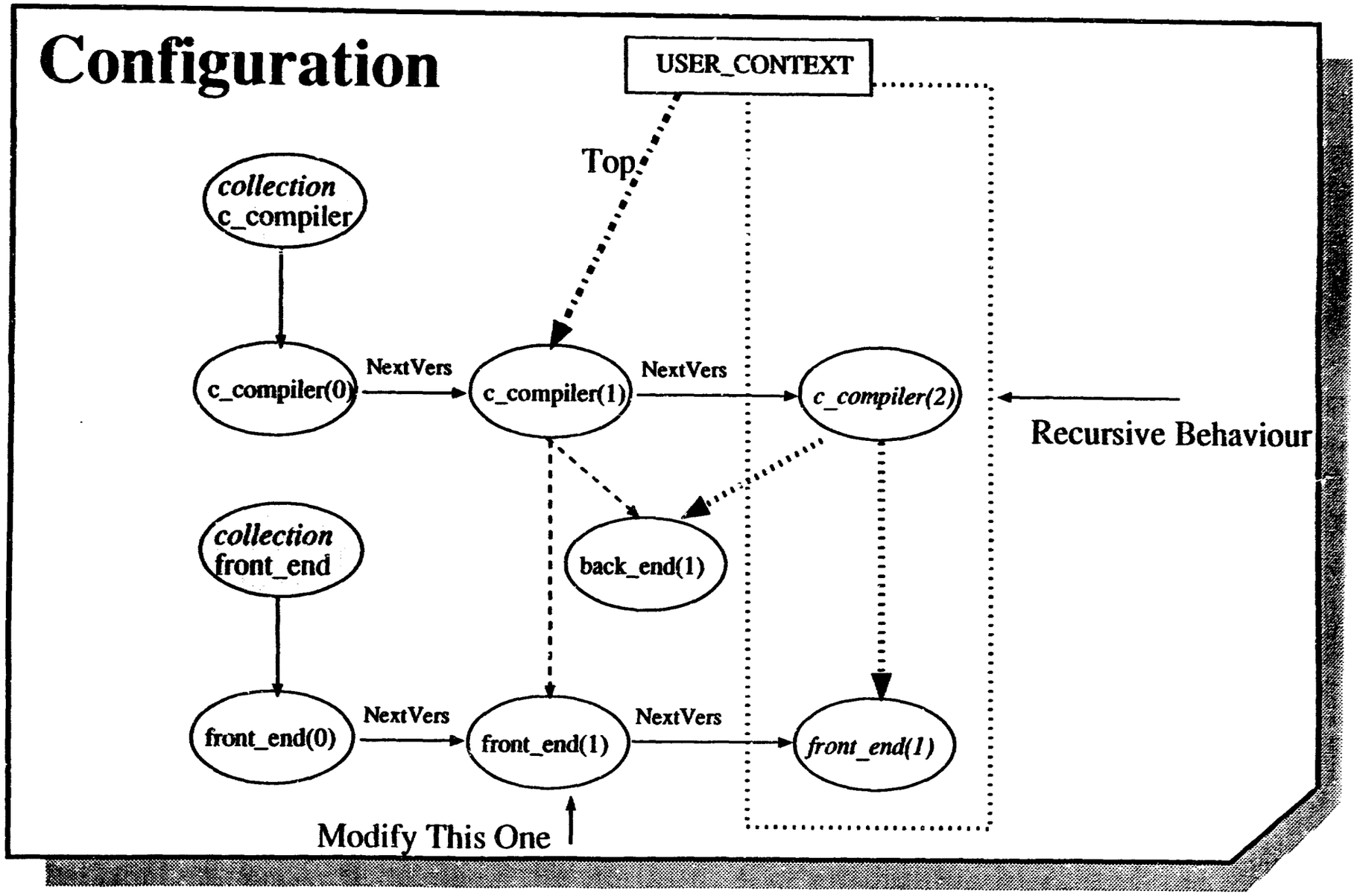


Merging PCTE and ATIS



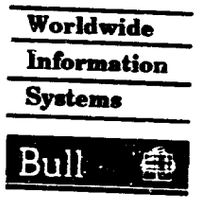
Versioning


Branch
1.implementedBy
2.implementedBy
availVersion
iohdlr.c(0)
nextVers
iohdlr.c(1)
iohdlr.c(2)
iohdlr.c(3)
1.branchesTo
Merging
Branch
nextVers
1.implementedBy
iohdlr.c:1(0)
iohdlr.c:1(1)


Configuration
USER_CONTEXT
Top
collection
c_compiler
c_compiler(0)
NextVers
c_compiler(1)
NextVers
c_compiler(2)
Recursive Behaviour
collection
front_end
back_end(1)
front_end(0)
NextVers
front_end(1)
NextVers
front_end(1)
Modify This One

PCIS Working Group Needs of Platform Providers

Burt Rubenstein
Director, Groupe Software Technology



June 1991

PCIS - Needs of Platform Providers

- No New Needs (amplify those in NRAC and 1st workshop)
- Software Development is Crucial - software sells hardware
- Platform provider is also tool provider, environment provider and environment user
- Portability and Connectivity to various platforms, not just Unix
- Need to support various domains
  - "Business" software development as well as technical
- Support of evolving Hardware environments
  - e.g., Distributed Heterogeneous Workgroups
- Need ability to attract ISV's - practical and wide-spread standards

Copyright © 1991 Groupe Bull

PCIS - Needs of Platform Providers (2)

- Platform Vendor - Interconnectivity is mandatory (operational)
  - Interoperability (portability) is desirable (to attract tools)
- Layer on top of other standards
  - no arbitrary new standards [OSF, POSIX, XPG, ACE]
- Need for (independent) certification of tools and environments
- Need for Roadmap - phased implementation and delivery
  - Customers can't wait - want evolution and migration (protect investment)

Copyright © 1991 Groupe Bull

PCIS - Needs of Platform Providers
[15-20 min Tues, 6/4 ≈ 4pm]

Not new needs - just amplify those in NRAC and 1st workshop
Software Development is crucial to platform providers - software sells hardware
Platform provider is also tool provider, environment provider and environment user
Portability and Connectivity to various platforms, not just Unix
Need to support various domains - "Business" software development as well as technical
Support of evolving Hardware environments - e.g., Distributed Heterogeneous Workgroups → IBM
Need ability to attract ISV's - practical and wide-spread standards
Platform Vendor - Interconnectivity is mandatory (operational)
    - Interoperability (portability) is desirable (to attract tools)
Layer on top of other standards - no arbitrary new standards [OSF, POSIX, XPG, ACE]
Need for (independent) certification of tools and environments
Need for Roadmap - phased implementation and delivery – Layered Evolution in the Implementation
    - Customers can't wait - want evolution and migration (protect investment)

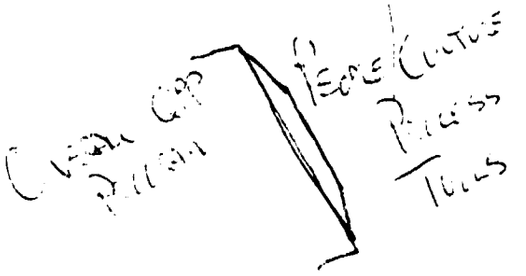
PCTE < Emerande
EUREKA & ESPRIT projects

Hyper-Web < ON PCTE
DO WE NEED TO REHOST?] PROVIDES ENVIRONMENT FOR SW WEB
PARTIALLY SOLVES CONTROL & VI INTEGRATION
& SUB-OBJECT LINKAGES

AtlantIS project · Internal → Product PHASE 1 Microsoft
EARLY 95

COMPLIMENTARY STANDARDS - e.g., LARGE-GRAIN VS. SMALL-GRAIN OBJECTS

EUROPE - PCTE
COMMERCIAL - esp. U.S. → AD/CYCLE
U.S. NON-DEFENSE FEDERAL - ANSI IRDS



Needs of Tool Builders

Attendees

Jorge Anderson
Francois Audras
Kurt Beitz
Jean Philippe Bourguignon
Bob Borowski
Herm Fischer (chair)
Hans Keus (co-chair)
Bob Munck
Hirogi Ochii
Erhardt Ploedereder
Clyde Roby
Bert Rubenstein
Nick Wybolt

Activities

Presentations

Nick Wybolt, Cadre

General Work

Review of Recommendations from Heathrow Sessions

Development of Success Factors – Avoidance Items

Identification of New Needs

Discussion of Control Integration

Consideration of commitment to external standards

Unfinished Heathrow Business

PCIS Implementation, all of it versus parts of it

Nick's Cadre Presentation

Herm's summary

Integration Technology

- Must be from platform supplier
- Must be dependably present

Baby Steps

- UI look and feel
- Pilot projects first

Data Integration

- Concerned about performance (vs. granularity)
- Granularity
- Non-pervasive repositories
- Tools give UIDs on private data to public relationship managers

Heathrow Issues

T1 What should PCIS Provide

T6 Configuration Management

T7 Tool communication

T10 Object Orientation

T11 Interoperability and language binding

T12 Networking

T13 User Interface

T15 Data Exchange between environments

T16 Licensing and accounting

T17 Net based help

T18 Future Work

Subjective Evaluation by Tool Vendors (or Environment Vendors)

What should PCIS (program, product) look like?

We've been talking to ourselves since 1981

People are writing and starting to use frameworks

People (tool writers) are complaining about frameworks

We should evaluate and do pilot projects on PTIs/frameworks:

- PCTE, PCTE+, CAIS
- Atherton, Cohesion, EAST, Enterprise 2, ESF, SoftBench

Build a matrix of likes and dislikes

produces "assessment" of strengths & weaknesses of PTI/framework technologies

Pick PTI/framework technology and PCIS should define a next generation of *that* technology

General Importance

Success	Avoidance
Corporate Backing	
Ubiquity Availability	Unique Schema Control
Platform Support (framework bought from hardware supplier)	Platform supplier has private framework architecture
Large selling market Commercial acceptance	Small market DoD too small (may be a stigma)
Compatibility with existing standards at release time	Dependence on standards recently demised at release
Low Cost per seat	High cost per seat
	Impact to computer operations Impact to project schedules
Incremental Acquisition	Monolithic acquisition & use
Performance	Workarounds
>>1 implementation with good performance	only 1 implementation

Tool Supplier Issues

Success	Avoidance
Reduce Tool Cost, Risk	Customer needs PCIS installed (PC Excel situation)
Integration – internally (betw. supplier's tools) is a must – externally (betw. diff. suppliers) is nice	Customer has to integrate (schema?)
Early availability prior to host/platform release	PCIS lags platform releases Tools are late to be ported

Modular Framework
Non-monolithic Framework

Requirement

PCIS shall facilitate, to a maximum degree, the decoupling of functional areas so that these functional parts can be used separately

Comments

Group "minimum configurations" – "levels of PCIS compatibility"

Avoid "random" subsets

Need modular non-monolithic ("polylithic") framework

Functional "parts" should have all of their components

PCIS can be defined modularly

The PCIS implementation can be built modularly using separate parts

Levels of Conformance

Validation suites

Tool works on several implementations

Verification that tool is PCIS - clean

Work-arounds are identified

Define compliance in parts

Performance evaluation benchmarks

Benchmarks are part of the PCIS specification

Minimal Capability to Integrate Multiple Object Mgrs.

Cohabitating Object Managers

Different Granuarity Object Managers sharing or cooperating

E.g., Ontos-PCIS installed on an Emeraude-PCIS

UID Assignments

Names and Navigation across Object Managers

Subtyping and methods integration between cohabitating Object Managers

- lower granularity specializes higher gran. methods

Ability to install low glevel readers/writers or handler for private data

Control Coordination at object level and across granularity boundary

PCIS and External Standards

They change

Do they apply at the right level

Do they replace us

Want as few as possible

- Define areas which need standards identified
- Identify single or multiple standards in the area

Needs of the Tool Builders
Wednesday A.M.

Introduction

Attendance

	Wed. AM	Wed. PM	Thurs. PM
Jorge Anderson	X	X	
Francois Audras	X	X	
K. Beitz	X	X	
J.P. Bourguignon	X		
Bob Borowski	X	X	
H. Fischer	X	X	
H. Kreuz	X	X	
B. Munck	X		
Hirogi Ochii	X		
E. Ploedereder		X	
C. Roby	X	X	
Bert Rubenstein	X		
N. Wybolt	X		

What the First PCIS Workshop Tool Builders' WG did

What should we do this time?
- Generate Additional requirements for IRAC with Rationale
- Review what we did last time as starting point

What is the IRAC supposed to be specifying?
- should we produce PTI, POM, PCM(public control manager) or PPM(public process manager)?
- last time, a large focus was on user requirements
- still don't have a mission statement
- consumers need control over their data structure and contents
  - tool swapability
  - consumer specified processes
  - process persitence
- consumers need to tool - address installation
- data sharability
- control exchange

tool out of toaster model
- more complicated process
- depending upon which implementation

consumer will tell tool builder what their class structure is like
- different vendors have different structures
- will be populated in different ways

need for powerful tools and procedures for those people who do installation of tools, because they are swamped

are we confusing basic system administration with installation of the tool

Activities
- Nominate New Tool Builder Needs
  2nd Map Builder Needs to User Needs
- List Success Factors
- List Avoidance Items

most tools should be run out of the box
problems occur with "pushing the envelope" of the tool
some problems occur with tailored installation

integration of tools within framework

toaster model marketed as frameworks
- Atherton
- HP Softbench
- DECs Cohesion
- EAST
- Enterprise II
- Eureka SW Factory
- EuroFighter IPSE

size of tool also makes a difference
e.g. Verdix Ada Compilation System needs almost one full time person

bigness of tool, or pervasiveness throughout the environment?
implies higher level things in the structure

small tool, e.g. pretty printer
will inherit from higher levels
implication of some possible changes

even different implementations of PCTE have different schemas
tools developed for one may not work for others

IPSEs have seen built without frameworks for years through "glueware"
easy to model the integration of tools, point-to-point
add framework
must now hope that this integration is still easily modified

users coming back with additional requirements
spcified data structure and contents
want to articulate our own process; not within tool
point-to-point could hide process

are 'process' a user WANT or NEED? is this a fad?

AD/Cycle calls out process modelling
probably first to do so in commercial product

Pure Commercial (Not military, not aero)
- not as aware of process modelling, perhaps
- they are more interested/concerned in puttnig forth best practices
- more *de rigeur* to do their internal processes, though not necessarily formalized

meta-level issue: what are we producing?
- a PTI (Public Tool Interface)
- a POM (Public Object Manager)
- a PCM (Public Control Manager)
- a PPM (Public Process Manager)

swapability issue
- people buying tools
- want multiple vendors integrated at all times in their environment
- tools yanked in and out helter/skelter?
- editors usually done so
- other tools usually not so swappable
- usually working with multiple vendors, e.g. both verdix and telesoft, over year-by-year replacability
- more encompassing tools are harder to replace/swap
- less encompassing, narrow tools are easy to replace/swap
- combinability issue different from swappability
  - combination should not be assumed because of swapability

charge of the day - gather info to write IRAC.

N _____ tk

Integration Framework Structure
- runs on Unix, OS/2, VMS

Observations of IPSE
- politics and economics more important than techy

IPSE Reqmts and Attribution
- IPSE should map onto organization
- can be introduced incrementally
  - presentation integration w/ common look&feel
  - put in place, pilot project use starts small, grows;
  - (softbench starts small and can grow)
- someone services the entire solution
  - not only at installation but over time
  - (people don't want to pay for a lot of "glue")
  - (framework provided by tool vendors)

Questions of Tool Integration Issues
- doubts of a common repository; more distributed data
- user interfaces - what kind?
- adaptable to new technologies
- who does and who supports the integration?

Capabilities provided by Framework - user/tool vendor concerns
- object storage and object relationship mgmt
  - fine-level linking would be nice, but not sure of how
  - capability will by used (need or want)
- requirement: tools need to export internal UIDs
- true control integration
- security/workflow - policy issue
  - integrity requirements across the board
  - - "don't lose my data"
- common user interface
  - toolkit to build one around products
- licensing capabilities
  - license server has to be shipped with the products now
  - since it is not common across platforms
  - also, broker different kinds of services

integrated link and message management framework
- combine link management and message switching in one box
- links between UIDs of objects and tools

emerging Standards
- Substrate Technology (substrata is operating system)
- Integration agent, CUI, OMS components of a CASE Framework should not be optional products (users don't want to pay pay for glue)
- Ubiquity = Standard (if everyone has it, its a standard)

SUMMARY (by HF)

Integrated Technology	- From Platform Vendor
	- Dependably Present
Baby Steps	- UI Look & Feel
	- Pilots First
Data Integration	- Concerned About Performance
	- Granularity
	- Non-Pervasive Repositories
	- Tools Give UIDs on Private Data/Public
	Relationship Mgmt

BREAK

Issues to revisit from 1st PCIS Workshop

T6	CM
T7	Tool Communication
T10	O-O
T11	Interoperability & Language Binding
T12	Net
T13	UI
T15	Data exchange between environments
T16	Licensing & accounting
T17	Net based help

T1: What should PCIS provide?

do we need an OMS?

NW: YES

course grained granularity versus fine grained?

NW: we just want to get a blob and manipulated it. we want control over the fine grained data. i would like a fine grained repository for both query access *and* for fast access. we need the ability to scale.

we need some words like FRAC 4.4 dealing with degrees of granularity and definitions of granularity. presentation integration?

NW: the war is over. we'll just choose motif or open look. adopt one or two of the style guides and run. take help as an example.
HK: in london, some tool vendors were reluctant to adopt any tool interface, because of the market. point 2 was that motif wasn't at a high enough level.
NW: once i get past the window dressing, i've got the control.
HF: x is too low level. look-and-feel is not alike at all, and if i take current tool building kits (EAST, Atherton), i get different look-and-feel.
BM: why should PCIS choose an existing standard?
HF: i think PCIS is in deep yogurt if it does. but if the users want to plug and play, we have to do something about this issue.
BM: if we've chosen std A, and if in 1998 stds A and B are equally widely used, tools will need to be built to use both.
NW: if we pick one, and design for it, and the other becomes the standard, we end up in deep trouble. we have to design for both.
All: so do we.
BM: it does the PCIS effort no good to choose a GUI standard. we can only hurt ourselves -- if we choose right, so what, and if we choose wrong, we get hurt.
HF: so PCIS needs some kind of style to standardize on some emerging standard. we don't necessarily want X.

LI: so, how much should PCIS commit to? if PCIS commits itself to too much in any area, PCIS will have to track the tech in that area and keep up with it.

HF: I still like what i wrote about technology tracking. If it gets embedded in silicon it can be a standard; if four or five standards emerge, nick will have to write tools that target all of those stds.

NW: but i don't want to standardize on a least common denominator. when i want to do something more sophisticated than motif allows, that forces me down to the Xlib level, and i don't want to have to do more of that than i do know.

CM

NW: essentially, i disagree that there is no consensus. will PCIS be publishing anything about CM reference models and versioning systems.

HF: i have been involved in some discussions about whether CM is toast or coils in the toaster model.

NW: to me, CM can be very tool specific.

HK: but i may want to use whatever CM system is provided by the framework/env i am using

BM: is there a single basic versioning capability that everyone will be happy with?

NW: are there still religious issues about the ordering of version numbering, etc.

HF: YES. absolutely. some people want long transactions, and some people want change sets, and each is based on a different paradigm, and that should be okay.

JPB: it would still help to have some basic elements to allow tools to build CM systems into a framework/environment.

HF: but there are still several basic paradigms onto which to base CM systems. if you're designing your CM as a piece of bread, that's fine.

HK: i would like to have versioning on both elementary and composite objects.

Tool communication

HF: do we need to get down to defining point-to-point message passing, and a concrete

O-O

Interoperability and Language Binding

NW: see T11. I violently agree with this. this has caused a lot of problems.

HF: are you willing to give up performance and re-usability for this.

Network

NW: the IRAC should be written with heterogeneity in mind.

BM: i want to be able to think about laptop distribution. i want to be able to undock my pc from one system and dock it elsewhere

Data exchange between environments

BM: we've got CEF, ultimately stolen from CAIS-A.
HF: but CEF is not a common external format in the manner of CDIF
BM: also, if we're running on heterogeneous machines, and we have some internal form, do we need a different external format.
HF: in that case you're talking about bits that you might not be able to take the information (in archive format) from environment to environment.
NW: is this data exchange between different PCTE implementations, or between PCTE and ATIS (e.g.)
HF: i expect that the answer is that it is between different ones (PCTE and ATIS)

Licencing & Accounting

Net Based Help

NW: I'm a bit concerned about these systems saying thou shalt have help
HF: and that thou shalt do it my way. if the tool builder is a good help/ platform citizen, the tool must conform its help to that of the platform.
BM: what's the point? what should PCIS do about help.
HF: if we're providing this help, we're not going to be very good citizens.
HK: why can't we worry about just some basic help message handling.

Internationalization

HF: There is much more of a push, even in the U.S., for internationalization.

Wednesday P.M.

Success/Avoidance
Tabled for know due to lack of input/Nick

Separatability

EP: When i use the OMS service, i only want to have to pay the price of using the OMS service. I don't want have to pay the price for process control.

HF: Should subsets of PCIS be allowed which do not contain all functionality of the PCIS?

BB: I think it should be possible, and that the PCIS documentation should define levels of compatibility so that framework implementations can say that they conform to the PCIS to such-and-such a degree.

HK: It seems like it should be possible to take parts of the environment out and replace them later with other parts. This is does not currently conform to the toaster model.

EP: This issue was also addressed at the last Workshop by the environment users group.

Control Integration

HF: We need to worry about what happens with an externally defined schema. (By externally defined schema, this means within the POM.) There may be different compilers, linkers, etc. which are going to have their own schemas, and they won't know the schema at run time.

EP: The current NRAC require the schema can be defined at run time.

HF: But this is a requirement on the framework. We're talking install-time binding. Its a requirement on tools, not on the framework.

EP: How do i/Am i expected to provide my tools schema to some other tool that wants to integrate with my tools. It doesn't necessarily know anything about the POM it is working with.

CR: As part of the installation process, would there be an encapsulator to conform to the POM?

HF: It's going to need to be there, but who's going to build it?

BB: An interesting part of this discussion is attempting to pacify/help out the users of the next 5 years so that they don't ignore PCIS altogether. What sort of political mandate will their be to use PCIS?

HF: Language is much easier to legislate than something where all the parts come from different places. It does no good unless you do something about mandating the parts, mandating the tools.

PCIS - How mandated

1) PCIS - Parts or all
2) Schemas
   - data
   - object
   - control
3) Process
4) Tasks

Mandating PCIS is not as difficult to mandate as the rest. What good is having something mandated if no one knows how its going to be used.

EP: The primary problem with current PTIs is that the user must take your tool and integrate the tool schema with the site schema.

HK: EAST as a consortium is defining a common schema. IEPG-TA13 is also doing the same thing. It is a very coarse-grained schema.

CR: the SWG on APSE is also defining a schema, but at an even coarser grain than the previous tool example.

EP: Even if Ubiquity exists, it is still not interesting if there is not some agreement on the schema.

We do not want to be compatible with obsolete standards. The answer is that we do not want to be compatible with only obsolete standards.

Does PCIS want to choose standards? How?

EP: This does not seem like a good idea.

BB: It causes PCIS to follow up on all emerging standards.

HK: Perhaps one solution is to say: in this area we choose this one, and in this area we choose several.

What should PCIS be? What should the programme be? The product?

EP; After all of this discussion, i'm still not sure what PCIS is supposed to be.

HF: I've been in meetings talking about requirements for 10 years, and we're still talking about requirements. It is very interesting to go out and see some of the early efforts in this area. For instance, Nick ripped cohesion apart.

We need to decide exactly what PCIS is. It is multiple choice, now. Some say environment framework, some say PTI.

People are currently both writing a lot and complaining about frameworks.

HK: We have identified requirements. Now we are coming back to the question of what exactly is PCIS, because we have been asked to come up with requirements for a product, and we are unclear on what exactly that is supposed to be.

We want to:

1) Evaluate the current state of the art (pilot projects or frameworks)
2) Matrix of likes and dislikes (this can be subjective)
3) Assessment of Strengths and weaknesses of these technologies
4) Pick technology (PTI or Framework) and PCIS should define a next generation of that technology.

What should PCIS provide?

For OMS, See FRAC section 4.4. Perhaps this section will be taken as a starting point; there needs to be definitions for granularity and for degrees of granularity.

CM

PCIS should be built to allow the building of different types of configuration management systems.

Need versioning on both elementary and composite objects.

Tool communication

Need to define separate layers.

The protocol needs to be abstract (OSI-like), as opposed to concrete (the factorialial expansion when new communication protocols are added)

O-O

Interoperability and Language Binding
Higher class structures constricts the ability for re-use of tools.

Network
Heterogeneous network: IRAC should be prepared with heterogeneity in mind. (PCE or Distrib. ???)

Data exchange between environments

Licencing & Accounting

Net Based Help

Internationalization
both icons and text

Separability
PCIS should facilitate to the maximum degree the decoupling of functional areas such that functional parts can be used separately.

At least look at the decoupling of the functional areas that are to be contained in a PTI forming an implementation of PCIS.

Group minimum configuration levels of PCIS compatibility. Avoid random subsets.

Modular non-monolithic ("polylithic"?) framework.

Functional parts should have all of their components.

The PCIS implementation can be built under modularity..

Success	Avoidance
General	
Needs to reduce my workload.	Implementing PCIS does not make sense if there is not a large market for it.
Needs to reduce my risk.	
Availability	
Compatible with existing standards	Difficult copyright/License issues
	Schema control should not put one vendor at the mercy of another.
Platform Support - Framwork provided by supplier	
Corporate Backing.	Platform supplier "forcing" use of a particular framework arch
Ubiquity	Unique
	Impact to Compute Center Operations
	Impact to Project Schedules
Low Cost/Seat	High Cost/Seat
Tool Supplier Issues	
Integration	Customer needs PCIS installed
- internally is a must	
- externally it is nice, and only more important if customers are asking for it.	.
Sufficient market	PCIS' Small Market

PCIS & External Standards

1) They change
2) Do they apply at the correct level
3) Do they replace PCIS
4) Choose one \
   Make one > want as few as possible
   Use all /

   we don't want make one because making one effectively adds one more to the list, aggrevating the choices.

Define the areas in which PCIS needs to make choices about standards.

These choices may be single or multiple, depending on the area.
The preference is to be able to choose the minimum number of standards.

What is PCIS (programme or product)?
A subjective evaluation of tool/environment by users and vendors.

PCIS - NOW MANDATED

1) PCIS - PARTS OR ALL

2) SCHEMAS
   DATA
   OBJECT
   CONTROL

3) PROCESS

4) TOOLS

PCIS & EXTERNAL STANDARDS

1) THEY CHANGE

2) DO THEY APPLY AT RIGHT LEVEL

3) DO THEY REPLACE US

4) CHOOSE ONE
~~APPLY ONE~~
USE ALL
} { WANT AS FEW AS POSSIBLE

→ DEFINE AREAS WHICH NEED STDS. IDENTIFIED
IDENTIFY SINGLE OR MULTIPLE STDS IN AREA
CHOOSE

PCIS shall facilitate, to a maximum degree the decoupling of functional areas so that these functional parts can be used separately.

~~Comments:~~

Group "minimum configurations" = levels of PCIS compatibility ~~?~~ ~ avoid random subsets

Modular non-monolithic ("polylithic") framework

Functional "parts" should have all of their components.

The PCIS implementation can be built modularly using ~~the services you need~~ separate parts

PCIS can be defined modularly.

SUBJECTING EVALUATION BY {PROGRAM / PRODUCT} USERS + VENDORS

WHAT SHOULD PCIS LOOK LIKE

1. We've been talking to ourselves since 1981
2. People are {using PTI's / writing funky} ←
3. People are complaining about —
4. ~~Show~~ We should evaluate
   - pilot projects on {PTI's / folks} — PTE, PTET, CALL — Alternation, Collocation, EAAT, Eint 2, LESF, Soft Bench
5. Matrix of likes & dislikes

⇓

ASSESSMENT OF STRENGTHS & WEAKNESSES OF {PTI / FOLK} TECHNOLOGIES

6. PICK {PTI / FOLK} TECHNOLOGY & PCIS SHOULD DEFINE A NEXT GENERATION OF THAT TECHNOLOGY

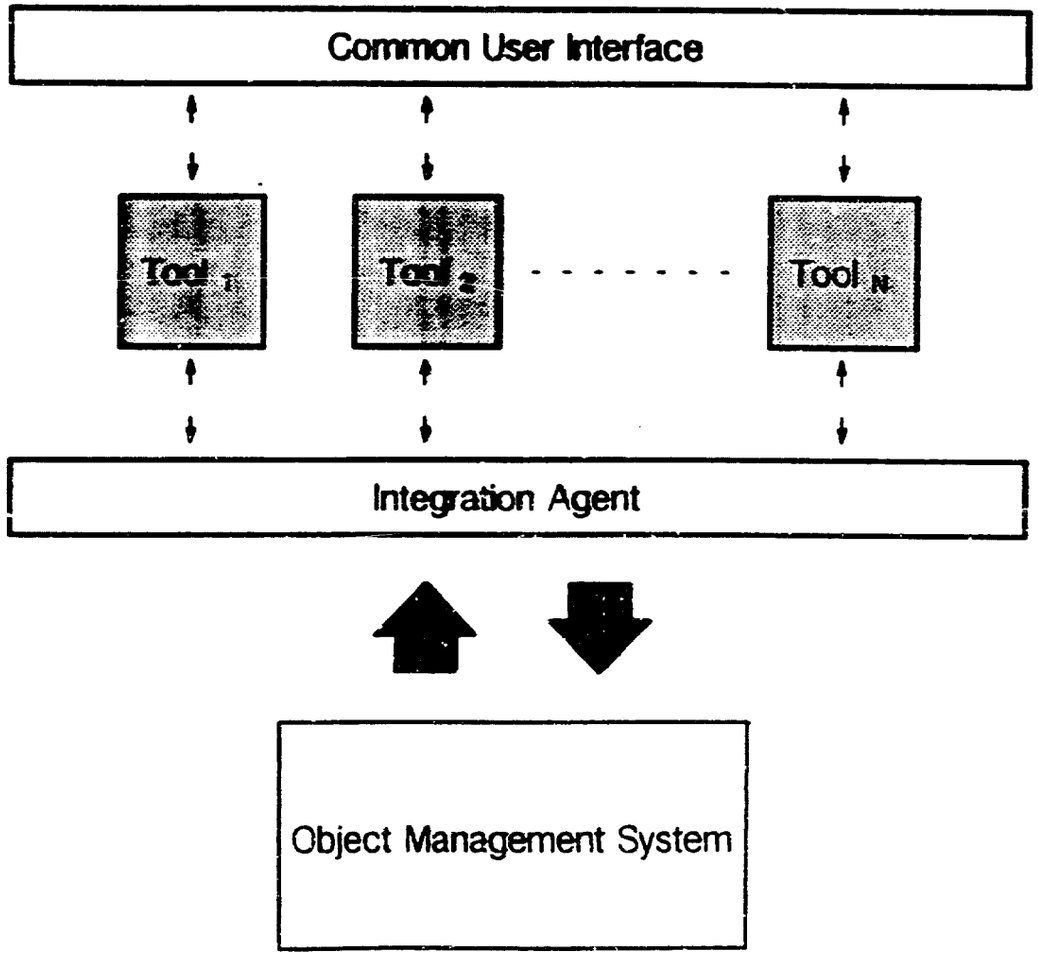
PCIS-NEEDS OF THE TOOL BUILDERS

NICHOLAS WYBOLT
CADRE TECHNOLOGIES

5/JUN/91

Copyright © 1991 Cadre Technologies Inc.

Integration Framework Structure



Copyright © 1991 Cadre Technologies Inc.

Introduction

IPSE

Observations:

- **The *Holy Grail* of most software project managers**
- **There are a number of ways to build an IPSE**
- **Tool integration is *necessary*, but not *sufficient*, to have an IPSE. The IPSE has to live in the context of, and be accepted by an organization**

Politics and economics are at least as important as technology.

Copyright © 1991 Cadre Technologies Inc.

Introduction

IPSE requirements & attributes:

- Cover all phases of development and support
- Support or connect with multiple disciplines
- Cover many activities
- Deal with multiple vendors and platforms
- Adapt to new and different technologies
- Adapt to existing organizational culture and work flows
- Can be introduced incrementally
- Someone services the entire solution
- Fast and inexpensive

Copyright © 1991 Cadre Technologies Inc.

Tool Integration Issues

Questions:

1. What system development phases and activities can be covered?
2. What tool choices does the IPSE provide?
3. Where does the data reside?
4. What kind of user interfaces are there?
5. How do the tools communicate?
6. How adaptable is the IPSE to new technology?
7. Who does the integration? Who supports it?

Copyright © 1991 Cadre Technologies Inc.

Frameworks – The Computing Environment

Capabilities Provided by Frameworks

User / Tool vendor concerns:

- Data exchange
- Object storage management
- Object relationship management
- Tool execution & inter-tool communication
- Security
- Distribution
- User interface
- Licensing

Copyright © 1991 Cadre Technologies Inc.

Tool Integration issues

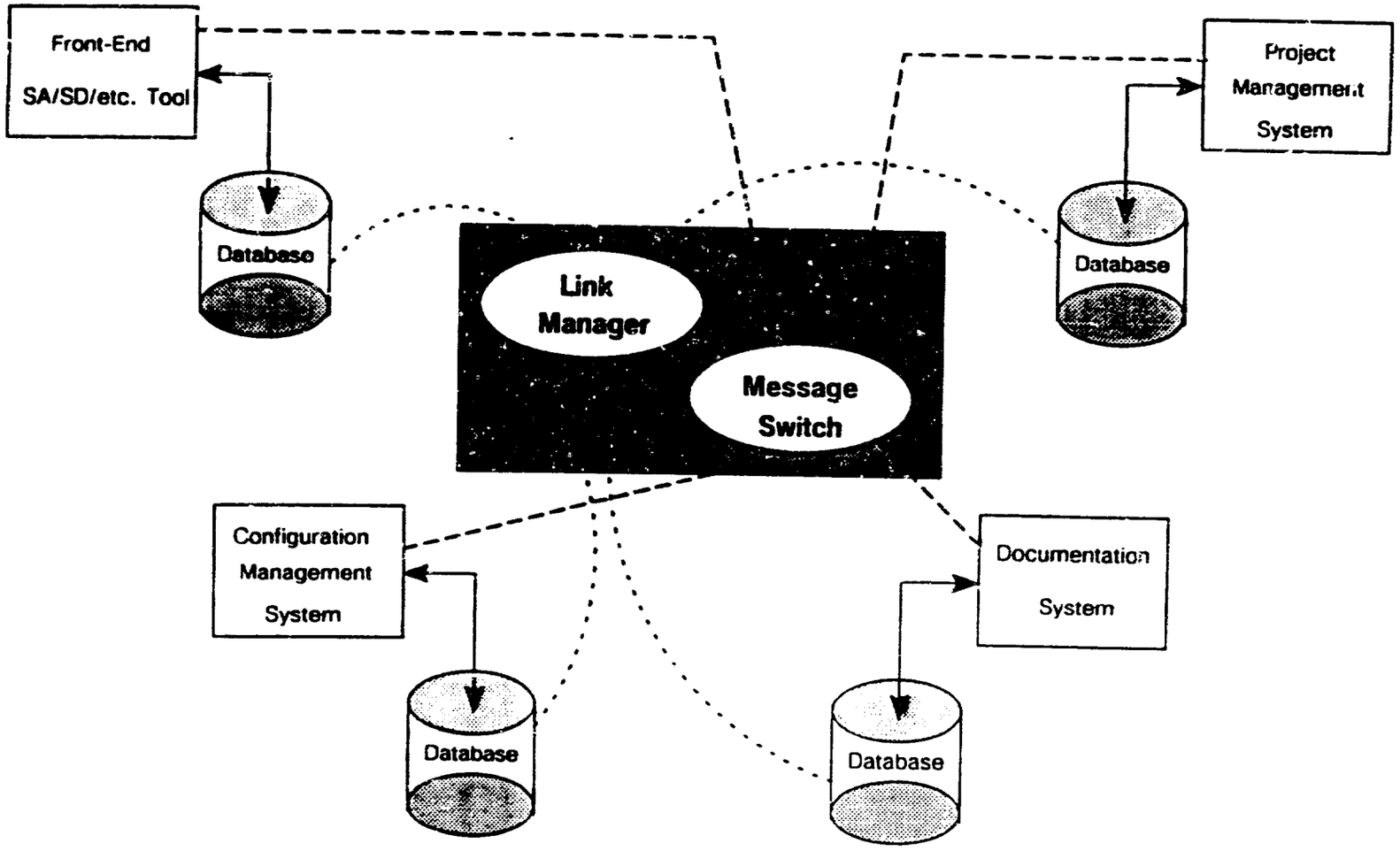
Different Agendas

Users and tool vendors judge a framework by the extent to which capabilities are provided.

Framework providers are concerned with the extent of capability that they must provide.

Copyright © 1991 Cadre Technologies Inc.

Integrated Link & Message Management Framework



Copyright © 1991 Cadre Technologies Inc.

The World In Which We Live

Styles

Presentation

Control

Data

Semantic

GUI's

OSF/Motif

OPEN LOOK

SAA CUA

"Standards"

ATIS

ECMA PCTE

CAIS-A

IRDS [ANSI + ISO]

EIA CDIF

VHDL

Repository (Data Model)

Programs

CIS

CFI

OMG

PCIS

IRDS 2

AD/Cycle

Cohesion

Products

SoftBench

NSE

Atherton BackPlane

CDD/Repository

Repository Manager/MVS

Mentor Falcon

Port-/Interoper-Ability

POSIX (IEEE 1003.1)

X/Open

ONC NFS

OSF/DEcorum

OMG Object Request Broker

757

Copyright © 1991 Cadre Technologies Inc.

Emerging Standards

Substrate Technology:

- Basic layer of technology or service that is expected to be in the computing environment where products are to be implemented
- One view is that the substrate *is* the operating system
- The *integration agent*, *common user interface*, and *object management system* components of the *CASE Framework* should not be optional products
- Platform vendors should own & supply substrate

Ubiquity = Standard

FINAL ATTENDANCE LIST

FOR

SECOND

PCIS

WORKSHOP

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Ackerman, Barry J. *NIST/PCIS/IWCASE*
Cadre Technologies Inc.
222 Richmond Street
Providence, RI 02903
PHONE: (401) 351-5950; FAX: (401) 351-7380
EMAIL:

Ali, Yawar *NIST/PCIS/IWCASE 2-c*
Bell-Northern Research Ltd.
P.O. Box 3511, Station C
Ottawa, Ontario
CANADA K1Y4H7
PHONE: 613/763-7741; FAX: 613/763-3305
EMAIL:

Andersen, Jorgen B. *NIST/PCIS a,d*
Honeywell
P.O. Box 21111
Phoenix, AZ 85036
PHONE: 602-436-1712; FAX: 602-436-2252
EMAIL:

Anderson, John *NIST 3*
Boeing Defense and Space Group
P.O. Box 3999, M/S 87-37
Seattle, WA 98124-2499
PHONE: (206) 773-4319; FAX:
EMAIL: anderson@stars.boeing.com

Audras, Francois *NIST/PCIS 1, a*
SYSECA
315 bureaux de la Colline
92213 St Cloud
FRANCE
PHONE: 33 1 49 11 70 00; FAX: 33 1 49 11 76 45
EMAIL: Francois.Audras@iv.bull.fr

LEGEND:

NIST ISEE Working Groups:
1. Object Management
2. Process and Task Management
3. Interface & Platform Services
4. User Interface

PCIS Working Groups:
a. Needs of the Tool Builders
b. Needs of the Platform Supplier
c. Needs of the Environment Supplier
d. Needs of the Environment Users

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Bagwill, Robert H. *NIST*
NIST
Bldg. 225, Room B266
Gaithersburg, MD 20899
PHONE: 301-975-3282; FAX: 301-590-0932
EMAIL: rbagwill@nist.gov

Baker, Thomas G. *NIST/PCIS//IWCASE 1,2,3,4-a*
Boeing
M/S 6H-03
P.O. Box 24346
Seattle, WA 98124
PHONE: 206-234-6234; FAX: 206-234-0256
EMAIL: tom@astars.boeing.com

Balzer, Robert *NIST/PCIS 3-c*
UCS-Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90291-6695
PHONE: 213-822-1511, ext. 249; FAX: 213-823-6714
EMAIL: balzer@isi.edu

Baudoin, Claude R. *PCIS d*
National Semiconductor, M/S 10-145
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
PHONE: (408) 721-6678; FAX: (408) 732-9654
EMAIL: BAUDOIN%SCVXCL.DNET@GPO.NSC.COM

Beitz, Kirk *NIST/PCIS/IWCASE 3,4-a*
Intermetrics, Inc.
733 Concord Ave.
Cambridge, MA 02138
PHONE: 617-661-1840 ext 4525; FAX:
EMAIL: johndoe@inmet.inmet.com

LEGEND:

NIST ISEE Working Groups:
1. Object Management
2. Process and Task Management
3. Interface & Platform Services
4. User Interface

PCIS Working Groups:
a. Needs of the Tool Builders
b. Needs of the Platform Supplier
c. Needs of the Environment Supplier
d. Needs of the Environment Users

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Belz, Frank — *PCIS d*
TRW R2/2062
One Space Park
Redondo Beach, CA 90278
PHONE: 213-812-0854; FAX: 213-812-7147
EMAIL: belz@trwarcadia.sdd.trw.com

Black, Eric — *PCIS c*
Atherton Technology
1333 Bordeaux Dr.
Sunnyvale, CA 94089
PHONE: (408) 734-9822; FAX: (408) 744-1607
EMAIL: ericb@atherton.com

Bladen, Jim — *PCIS a*
Telesoft
5959 Cornerstone Court West
San Diego, CA 92121
PHONE: 619-457-2700; FAX: 619-452-2117
EMAIL:

Borowski, Bob — *NIST/PCIS/IWCASE 2-a*
Protocol, A Division of Zycad
500 International Drive
Mount Olive, NI 07828
PHONE: 201-347-7900; FAX: 201-347-0303 fax
EMAIL:

Boswell, Murrah — *NIST/PCIS 1,2,3,4-a,b,c,d*
Navajo Technologies, Inc
Navajo Nation
Leupp, AZ 80035
PHONE: 602/686-6391; FAX: 602/686-6227
EMAIL:

Boudier, Gerard — *NIST/PCIS c*
Gie Emeraude c/o Bull 58F
68 route de Versailles
78430 Louveciennes
FRANCE
PHONE: 33 1 39 02 40 93; FAX: 33 1 39 02 42 06
EMAIL: Gerard.Boudier@lv.bull.fr

LEGEND:

NIST ISEE Working Groups:
1. *Object Management*
2. *Process and Task Management*
3. *Interface & Platform Services*
4. *User Interface*

PCIS Working Groups:
a. *Needs of the Tool Builders*
b. *Needs of the Platform Supplier*
c. *Needs of the Environment Supplier*
d. *Needs of the Environment Users*

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Bourguignon, Jean Philippe ***PCIS*** ***a,c***
SFGL
14 rue de la Ferme
92100 Boulo Gne
FRANCE
PHONE: 33 1 47 61 05 20 FAX: 33 1 47 61 92 15
EMAIL: Jean.Philippe.Bourguignon@sfgl.rlgl.fr (?)

Bremeau Christian ***NIST/PCIS*** ***c***
GIE Emeraude
c/o Bull (bat 58F)
68 route de Versailles
78430 Louveciennes
FRANCE
PHONE: 33 1 39 02 40 56; FAX: 33 1 39 02 42 06
EMAIL: Christian.Bremeau@lv.bull.fr

Bruso, Kelsey ***NIST/PCIS*** ***3-b,c***
Unisys Corporation
P.O. Box 64942
MS: 4762
St. Paul, MN 55164-0942
PHONE: 612-635-3478; FAX: 612-635-3899
EMAIL:

Canning, A ***PCIS*** ***d***
ERA Technology LTD
Cleeve Road
Leatherhead, Surrey
ENGLAND
PHONE: 0 372 374151; FAX: 0 372 374496
EMAIL:

Carlson, Susan ***PCIS*** ***d***
AdaIC
% IIT Research Institute
4600 Forbes Blvd.
Lanham, MD 20706
PHONE: 703/685-1477, 800/Ada-ic11; FAX: 703/685-7019
EMAIL: carlsons@ajpo.sei.cmu.edu

LEGEND:

NIST ISEE Working Groups:
1. Object Management
2. Process and Task Management
3. Interface & Platform Services
4. User Interface

PCIS Working Groups:
a. Needs of the Tool Builders
b. Needs of the Platform Supplier
c. Needs of the Environment Supplier
d. Needs of the Environment Users

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Carney, David — *NIST/PCIS* *1,c*
Institute for Defense Analyses
1801 North Beauregard Street
Alexandria, VA 22311-1772
PHONE: 703-845-6648; FAX: 703-845-6848
EMAIL: carney@ida.org

Cerino, Deborah — *NIST/PCIS* *1-d*
Rome Laboratory
RL/COEE
Griffiss Air Force Base, NY 13441
PHONE: 315/330-2054 ; FAX: 315/330-3911
EMAIL: cerino@softvax.radc.af.mil

Clow, Geoff — *NIST/PCIS/IWCASE* *2,c*
Softech, Inc.
Suite 100
10875 Rancho Bernardo Road
San Diego, CA 92127
PHONE: 619-451-9320; FAX: 619-451-9327
EMAIL: clow@nosc.mil

Colket, Currie — *PCIS* *3,d*
AJPO
The Pentagon Room 3E114
Washington, D.C. 20301-3081
PHONE: 703-614-0209; FAX: 703-685-7019
EMAIL: colket@ajpo.sei.cmu.edu

Costa, Claudio — *PCIS* *d*
Alenia Spa
Via Tiburtina KM 12.4
POB 7083-00131 Roma
ITALY
PHONE: +39 643602520; FAX: +39 64131091
EMAIL:

LEGEND:

NIST ISEE Working Groups:
1. *Object Management*
2. *Process and Task Management*
3. *Interface & Platform Services*
4. *User Interface*

PCIS Working Groups:
a. *Needs of the Tool Builders*
b. *Needs of the Platform Supplier*
c. *Needs of the Environment Supplier*
d. *Needs of the Environment Users*

8/7/91 5:01 PM

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Cox, Gary — ***PCIS*** ***a,b***
IBM Corp.
525 Lascade
Colorado Springs, CO 80903
PHONE: 719/520-3028; FAX:
EMAIL:

Cristina, Jackie — ***NIST/PCIS/IWCASE*** ***1,3-a,b,c,d***
US Army
Attn: CSSD-SA-BT/Jackie Cristina
P.O. Box 1500
Huntsville, AL 35807-3801
PHONE: 205-955-3861; FAX: 205-955-3994
EMAIL:

Cuoco, Ed — ***NIST/PCIS/IWCASE*** ***2,4-a,b***
Digital Equipment Corp.
110 Spit Brook Rd. ZK02-1/M11
Nashua, NH 03062
PHONE: 603-881-2338; FAX: 603-881-0120
EMAIL: cuoco@tle.dec.com

Cureton, W.H. — ***NIST/PCIS/IWCASE*** ***1-b***
Sun Microsystems
PAL1 424
2550 Garcia Ave.
Mountain View, CA 94043
PHONE: 415-336-2822; FAX:
EMAIL: bill.cureton@corp.sun.com

Davis, Hugh — ***NIST/PCIS*** ***3,d***
ICL
Eskdale Road
Winnersh
Wokingham
Berkshire RG11 5TT
ENGLAND
PHONE: +44 734 693131; FAX: +44 734 697636 fax
EMAIL: hfd@win.icl.co.uk

LEGEND:

NIST ISEE Working Groups:
1. *Object Management*
2. *Process and Task Management*
3. *Interface & Platform Services*
4. *User Interface*

PCIS Working Groups:
a. Needs of the Tool Builders
b. Needs of the Platform Supplier
c. Needs of the Environment Supplier
d. Needs of the Environment Users

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Dawes, John — *PCIS* *a,b,c,d*
ICL
Eskdale Road, Winnersh
Wokingham
Berkshire RG11 5TT
UNITED KINGDOM
PHONE: +44 734 693131; FAX: +44 734 697636 fax
EMAIL: sjd@win.icl.co.uk

Denson, David — *NIST*
Texas Instruments
2750 Prosperity Ave., Suite 100
Fairfax, VA 22031
PHONE: 703-849-1450; FAX: 703-560-5803
EMAIL:

Drake, Dick — *NIST* *(Did Not Attend)*
IBM
800 N. Frederick Ave.
M/S 18213F11
Gaithersburg, MD 20879
PHONE: 301-240-6149; FAX:
EMAIL: ddrake@ajpo.sei.cmu.edu

Dutton, Jim — *NIST/PCIS* *4-c*
International Software Systems
9430 Research Blvd. Echelon III Suite 250
Austin, TX 78729
PHONE: 512-338-5729; FAX: 512-338-5757
EMAIL: uucp: issi!dutton

Earl, Anthony — *NIST/PCIS* *2*
Hewlett-Packard
HP Labs. Filton Road, Stoke Gifford
Bristol BS12 6QZ
ENGLAND
PHONE: +44 0 272 799910; FAX: +44 0 272 790554
EMAIL: ane@hplb.hpl.hp.com

LEGEND:

NIST ISEE Working Groups:
1. *Object Management*
2. *Process and Task Management*
3. *Interface & Platform Services*
4. *User Interface*

PCIS Working Groups:
a. *Needs of the Tool Builders*
b. *Needs of the Platform Supplier*
c. *Needs of the Environment Supplier*
d. *Needs of the Environment Users*

8/7/91 5:01 PM

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Ecimouic, Dusan — *NIST/PCIS* *1,2*
425 Market Street
San Francisco, CA
PHONE: 415/545-2671; FAX: 415/545-2813
EMAIL:

Ekman, Robert W. — *NIST/PCIS* *1c*
IBM STARS
800 N. Frederick Ave.
Gaithersburg, MD 20879
PHONE: 301-240-6431; FAX:
EMAIL: ekmanb@rckvm1.vnet.ibm.com

Ett, William T — *NIST/PCIS* *2*
19724 Drop Forge Lane
Gaithersburg, MD 2087
PHONE: 301/245-6322; FAX: 301/240-6873
EMAIL: ettb@wmavm7.iinus1.ibm.com

Fischaleck, Maria — *PCIS* *a,d*
IABG
Einsteinstr 20
8012 Ottobrunn
GERMANY
PHONE: +49 89 6088-3560; FAX: +49 89 6088 3418
EMAIL: uucp: uuido.uucp!iabgsz!sztsun04!fischal

Fischer, Herm — *NIST/PCIS/IWCASE* *1,2,3-a,d*
Mark V Systems Limited
16400 Ventura Blvd., Suite 303
Encino, CA 91436
PHONE: 818-995-7671; FAX: 818-995-4267
EMAIL: hermix!fischer@rand.org

Gibbons, Mark — *NIST/PCIS* *1,2-d*
British Telecommunications
BT Laboratories
Martlesham Heath
IPSWICH IP5 7RE
ENGLAND
PHONE: +44 473 642056; FAX:
EMAIL: mgibbons@axion.bt.co.uk

LEGEND:

NIST ISEE Working Groups:
1. Object Management
2. Process and Task Management
3. Interface & Platform Services
4. User Interface

PCIS Working Groups:
a. Needs of the Tool Builders
b. Needs of the Platform Supplier
c. Needs of the Environment Supplier
d. Needs of the Environment Users

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Goldman, Neil M. *NIST/PCIS*
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
PHONE: 213-822-1511 x247; FAX: 213-823-6714
EMAIL: goldman@isi.edu

Gutzmann, Kurt *NIST/PCIS 1,2,3,4-a,b,c,d*
Softech
1300 Hercules, Suite 105
Houston, TX 77058
PHONE: 713-480-1994; FAX: 713-480-1994
EMAIL: gutzmann@ajpo.sei.cmu.edu

Haddad, Ranwa *NIST/PCIS/IWCASE 1,2,3,4-d*
Aerospace Corporation
M/S M8/166
P.O. Box 92957
Los Angeles, CA 90009-2957
PHONE: 213-336-3438; FAX: 213-336-3538
EMAIL: haddad@aerospace.aero.org

Hanrahan, Robert P. *NIST/PCIS/IWCASE 3-c*
US Air Force STSC
OOALC/TISAC
B100 Bay G
Hill AFB, UT 84056
PHONE: 801-777-8045; FAX: 801-777-8069
EMAIL: hanrahan@oodis01.af.mil

Harkleroad, Lt. Joseph E. *NIST/PCIS*
Naval Post Graduate School
Monterey, CA
PHONE: 408-646-2056; FAX: 4 08-646-2760
EMAIL:

Hart, Hal *NIST/PCIS/IWCASE 2-d*
TRW R2/2062
One Space Park
Redondo Beach, CA 90278
PHONE: 213-812-0661; FAX: 213-812-7147
EMAIL: halhart@ajpo.sei.cmu.edu, stars@trwarcadia.sdd.trw.com

LEGEND:

NIST ISEE Working Groups:
1. Object Management
2. Process and Task Management
3. Interface & Platform Services
4. User Interface

PCIS Working Groups:
a. Needs of the Tool Builders
b. Needs of the Platform Supplier
c. Needs of the Environment Supplier
d. Needs of the Environment Users

8/7/91 5:01 PM

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Hartman, Don — *NIST/PCIS* — *1,4-a,b,c,d*
Hal Computer Systems
8920 Business Park Drive
Austin, TX 78759
PHONE: 512-794-2855; FAX: 512-794-8737
EMAIL: halaus!dwh@hal.com

Hayter, Ken — *PCIS* — *c*
Defence Research Agency (UK)
RSRE
St. Andrews Rd.
Malvern
Worcestershire
WR14 3 PS
UNITED KINGDOM
PHONE: +44 684 89 5836; FAX: +44 684 89 4540
EMAIL: KWH%hermes.mod.UK@relay.MOD.UK

Horton, Michael — *NIST/PCIS/IWCASE* — *1-c*
Unisys Corp.
P.O. Box 517
Paoli, PA 19301
PHONE: 215-648-2527; FAX: 213-648-2288
EMAIL: horton@prc.unisys.com

Jenkins-Bnafa, Jovita — *NIST/PCIS* — *1,4-a,d*
TRW
1 Space Park R2/2044
Redondo Beach, CA 90278
PHONE: 213/812-1795; FAX: 213/812-7147
EMAIL:

Kavianpour, Mansour — *NIST/PCIS* — *2-a,c*
IBM Canada Ltd.
895 Don Mills Road
North York, Ontario
M3C 1V7
CANADA
PHONE: 416/448-2379; FAX: 416/448-2114
EMAIL: mansour@torolabs.vnet.ibm.com (?)

LEGEND:

NIST ISEE Working Groups:
1. *Object Management*
2. *Process and Task Management*
3. *Interface & Platform Services*
4. *User Interface*

PCIS Working Groups:
a. *Needs of the Tool Builders*
b. *Needs of the Platform Supplier*
c. *Needs of the Environment Supplier*
d. *Needs of the Environment Users*

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Keith, Sharon — *NIST/PCIS/IWCASE*
NOSC
271 Catalina Blvd.
San Diego, CA 92152
PHONE: 619/553-6858; FAX: 619/553-5799
EMAIL: keith@nosc.mil

Kerner, Judy — *NIST/PCIS* *1,3,4-a,d*
The Aerospace Corp.
MS M8/117
P.O. Box 92957
Los Angeles, CA 90009
PHONE: 213-336-3131; FAX: 213-336-3538
EMAIL: kerner@arecibo.aero.org

Keus, Hans E. — *PCIS* *a,c*
Westmount Technology
P.O. Box 5063
2600 GB Delft
THE NETHERLANDS
PHONE: +31 15 610815; FAX: +31 15 565701
EMAIL: uucp: hp4nl!wmt!hake

King, James — *NIST/IWCASE* *2*
Boeing Defense and Space Group
P.O. Box 3999, MS 87-37
Seattle, WA 98124-2499
PHONE: 206-773-4316; FAX: 206-773-4946
EMAIL: jk@stars.boeing.com

La Basse, Daniel T — *NIST/PCIS* *1,3*
407 Shirleen
Seabrook, TX 77586
PHONE: 713/333-0947; FAX: 713/333-2813
EMAIL:

Leary, John — *NIST/PCIS/IWCASE* *4-d*
Martin Marietta Corp.
4795 Meadow Wood La.
Chantilly, VA 22021
PHONE: 703-802-5604; FAX: 703-802-5464
EMAIL: learyj@ajpo.sei.cmu.edu

LEGEND:

NIST ISEE Working Groups:
1. Object Management
2. Process and Task Management
3. Interface & Platform Services
4. User Interface

PCIS Working Groups:
a. Needs of the Tool Builders
b. Needs of the Platform Supplier
c. Needs of the Environment Supplier
d. Needs of the Environment Users

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Leon, Jeff — *NIST/PCIS/IWCASE* — *1,3,4-a,d*
Hal Computer Systems
8920 Business Park Drive
Suite 300
Austin, TX 78759
PHONE: 512-794-2855; FAX: 512-794-8737
EMAIL:

Lindquist, Timothy E. — *PCIS* — *1-d*
Computer Science and Engineering Dept.
Arizona State University
Tempe, AZ 85287-5406
PHONE: 602-965-2783; FAX: 602-965-2751
EMAIL: lindquis@asuvax.eas.asu.edu

Longden, D. — *PCIS* — *4-d*
UK Mod
Room 441 Turnstile House
98 High Holborn
Longdon WC1V 6LL
ENGLAND
PHONE: +44 71 430 5331; FAX: +44 71 430 5650
EMAIL:

Matsubara, Tomoo — *NIST/PCIS* — *NUMBERLETTER*
6-81 ONOE-MACHI,
NAKA-KU, YOKOHAMA
JAPAN
PHONE: +81-45-681-2111; FAX: +81-45-681-4099
EMAIL: Matsu@aqu.hitachi-sk.co.jp

McKee, Gary — *NIST/PCIS* — *a, b, c,d*
McKee Consulting
PO Box 3009
Littleton, CO 80151
PHONE: 303/795-7287; FAX:
EMAIL: gmckee@ajpo.sei.cmu.edu
EXPRESS MAIL: 111 West Fremont Ave., Littleton, CO 80120-4242

LEGEND:

NIST ISEE Working Groups:
1. *Object Management*
2. *Process and Task Management*
3. *Interface & Platform Services*
4. *User Interface*

PCIS Working Groups:
a. *Needs of the Tool Builders*
b. *Needs of the Platform Supplier*
c. *Needs of the Environment Supplier*
d. *Needs of the Environment Users*

8/7/91 5:01 PM

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

McQuage, Neil — *NIST/PCIS/IWCASE 1-c*
Boeing Defense and Space Group
P.O. Box 3999, MS 87-37
Seattle, WA 98124-2499
PHONE: 206-773-4310; FAX: 206-773-4946
EMAIL: mcquage@stars.boeing.com

Memmi, G. — *NIST/PCIS/IWCASE 1,4-c*
Bull
300 Concord Rd.
MS MA30/8214
Billerica, MA 01821
PHONE: 508-294-2617; FAX: 508-294-4361=
EMAIL: memmi@pws.bull.com

Milligan, James — *NIST/PCIS*
Rome Laboratory
RL/COEE
Griffiss Air Force Base, NY 13441
PHONE: 315/330-2054; FAX: 315/330-3911
EMAIL: milligan@softvax.radc.af.mil

Minot, Regis — *NIST/PCIS c*
Gie Emeraude c/o Bull
68 route de Versailles
78430 Louveciennes
FRANCE
PHONE: (+33) 1.39.02. 51.16; FAX: (+33) 1.39.02. 42.06
EMAIL: Regis.Minot@lv.bull.fr

Morin, Joseph — *NIST 2,3-a,c*
Software Engineering Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213-3890
PHONE: 412-268-8594; FAX: 412-268-5758
EMAIL: jfm@sei.cmu.edu

LEGEND:

NIST ISEE Working Groups:
1. Object Management
2. Process and Task Management
3. Interface & Platform Services
4. User Interface

PCIS Working Groups:
a. Needs of the Tool Builders
b. Needs of the Platform Supplier
c. Needs of the Environment Supplier
d. Needs of the Environment Users

8/7/91 5:01 PM

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Morron, M.W. *NIST/PCIS/IWCASE 4 (Did Not Attend)*
BNR Europe, Ltd.
London Road
Harlow, Essex
CM17 9NA
UNITED KINGDOM
PHONE: +44 279 429531; FAX: +44 279 441551
EMAIL: mrm@stl.stc.co.uk

Mulholland, Sandra *NIST/PCIS*
Rockwell
MS 124-211
400 Collins Road NE
Cedar Rapids, IA 52498
PHONE: 319-395-4047 FAX:
EMAIL: smulholl@ajpo.sei.cmu.edu

Munck, Robert G. *NIST/PCIS 1-b*
Unisys
12010 Sunrise Valley Drive
Reston, VA 22091
PHONE: 703/620-7991; FAX 703/620-7916
EMAIL: munck@STARS.Reston.Unisys.COM

Nolan, Chris J. *NIST/PCIS/IWCASE*
Technical Director, SDT Italy,
Digital Equipment Corporation,
Piazza XX Settembre, 1
21100 VARESE
ITALY
PHONE: +39 332 298460; FAX: +39 332 240 290
EMAIL: nolan@varese.dec.com

Oberndorf, Patricia *NIST/IWCASE*
Naval Air Development Center
Code 7031
Warminster, PA 18974-5000
PHONE: 215-441-2737; FAX: 215-441-3225
EMAIL: tricia@nadc.navy.mil

LEGEND:

NIST ISEE Working Groups:
1. *Object Management*
2. *Process and Task Management*
3. *Interface & Platform Services*
4. *User Interface*

PCIS Working Groups:
a. *Needs of the Tool Builders*
b. *Needs of the Platform Supplier*
c. *Needs of the Environment Supplier*
d. *Needs of the Environment Users*

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Ochii, Hirzji — *NIST/PCIS 1, 3*
Hitachi Software Engineering Co., Ltd
6-81 Onoe-Cho Naka-Ku Yokohama City
Kanagawa,
JAPAN
PHONE: 0-45-824-2111; FAX: 0-45-824-0566
EMAIL:

Oliver, Huw — *PCIS*
Hewlett-Packard
HP Labs, Filton Road, Stoke Gifford
Bristol BS12 6QZ
ENGLAND
PHONE: +44 0 272 799910; FAX: +44 0 272 790554
EMAIL: heo@hplb.hpl.hp.com

Penedo, Maria — *NIST/PCIS/IWCASE 1*
TRW R2/2062
One Space Park,
Redondo Beach, CA 90278
PHONE: 213-812-0595; FAX: 213-812-7147
EMAIL: penedo@trwarcadia.sdd.trw.com

Peterson, Judi — *NIST/PCIS/IWCASE 1-d*
TRW
Mail Station: HAFB/100
1104 Country Hills Dr.
Ogden, UT 84403
PHONE: 801-773-8938; FAX: 801-777-8069
EMAIL: judip@oodis01.af.mil

Peterson, Ron — *NIST/PCIS 2-c*
TRW
Mail Station: HAFB/100
1104 Country Hills Dr.
Ogden, UT 84403
PHONE: 801-773-8938; FAX: 801-777-8069
EMAIL: ronp@oodis01.af.mil

LEGEND:

NIST ISEE Working Groups:
1. Object Management
2. Process and Task Management
3. Interface & Platform Services
4. User Interface

PCIS Working Groups:
a. Needs of the Tool Builders
b. Needs of the Platform Supplier
c. Needs of the Environment Supplier
d. Needs of the Environment Users

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Ploedereder, Erhard — *NIST/PCIS* *1-a*
Tartan Inc.
300 Oxford Drive - Level 4
Monroeville, PA 15146
PHONE: 412-856-3600; FAX: 412-856-3636
EMAIL: ploedere@tartan.com

Power, Leigh — *NIST/PCIS/IWCASE* *2-d*
MCC
3500 West Balcones Center Drive
Austin, TX 78750
PHONE: 512/338-3480; FAX: 512/338-3899
EMAIL: power@mcc.com

Printz, Jacques — *PCIS* *a,b,c,d*
CR2A
CGI Group
19, avenue Dubonnet
92411 Coubevoie Cedex
FRANCE
PHONE: 33 1 47 68 97 97; FAX: 33 1 47 68 87 81
EMAIL: printzj@ajpo.sei.cmu.edu

Pritchett, Gary — *NIST/PCIS/IWCASE* *1-a,b,c,d*
Softech, Inc.
Suite 100
10875 Rancho Bernardo Road
San Diego, CA 92127
PHONE: 619-451-9318; FAX: 619-451-9327
EMAIL: gpritchett@nosc.mil

Randall, Charlie — *NIST/PCIS/IWCASE* *1,2,4-a,c,d*
GHG Corp.
1300 Hercules
Suite 111
Houston, TX 77058
PHONE: 713-488-8806; FAX:
EMAIL:

LEGEND:

NIST ISEE Working Groups:
1. Object Management
2. Process and Task Management
3. Interface & Platform Services
4. User Interface

PCIS Working Groups:
a. Needs of the Tool Builders
b. Needs of the Platform Supplier
c. Needs of the Environment Supplier
d. Needs of the Environment Users

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Rankin, Robert — *PCIS d*
Defense Research Agency (DRA)
DRA-RSRE
St. Andrews Rd.
Malvern Works,
WR14 3PS
UNITED KINGDOM
PHONE: +44 684 89 58 16; FAX: +44 684 89 45 40
EMAIL: rankin%uk.mod.hermes@relay.mod.uk

Robinson, Dave — *PCIS d*
SD-Scicon UK Ltd
Pembroke House
Pembroke Broadway,
Camberley,
Surrey, GU15 3XD
ENGLAND
PHONE: 44-276-686200; FAX: 44-276-66435
EMAIL:

Roby, Clyde — *NIST/PCIS/IWCASE 1,2,3,4-d*
Institute for Defense Analyses
1801 N. Beauregard St.
Alexandria, VA 22311-1772
PHONE: 703-845-6666; FAX: 703-845-6848
EMAIL: roby@ida.org

Rubenstein, Burt — *PCIS*
Groupe Bull
300 Concord Rd.
MA30-821A
Billerica, MA 01821
PHONE: 508-294-4165; FAX: 508-294-4361
EMAIL: b.rubenstein@bull.com

Rudmik, Andy — *PCIS a*
Software Productivity Solutions, Inc.
P.O.Box 361697
Melbourne, FL 32936
PHONE: (407)984-3370; FAX:
EMAIL: axr@SPS.COM

LEGEND:

NIST ISEE Working Groups:
1. Object Management
2. Process and Task Management
3. Interface & Platform Services
4. User Interface

PCIS Working Groups:
a. Needs of the Tool Builders
b. Needs of the Platform Supplier
c. Needs of the Environment Supplier
d. Needs of the Environment Users

8/7/91 5:01 PM

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Ryan, Tom, Sr. — *NIST/PCIS* — *1,2,3,4-a,b,c,d*
Navajo Technologies
Navajo Nation
Leupp, AZ 80035
PHONE: 602/686-6391; FAX: 602/686-6391
EMAIL:

Scacchi, Walt — *NIST/PCIS/IWCASE* — *1,2-a,c*
Computer Science Dept.
University of Southern California
Los Angeles, CA 90089-0782
PHONE: 213-740-4782; FAX: 213-740-7285
EMAIL: scacchi@pollux.usc.edu

Schmiedekamp, Carl — *NIST/PCIS/IWCASE*
Naval Air Development Center
Code 7033
Warminster, PA 18974
PHONE: 215-441-1779; FAX: 215-441-3225
EMAIL: carls@nadc.navy.mil

Sharon, David — *IWCASE*
CASE ASSOCIATES, Inc.
15686 S. Bradley RD
Oregon City, OR 97045
PHONE: 503/656-0986; FAX: (503) 656-3207
EMAIL:

Shere, Kenneth — *NIST/PCIS/IWCASE* — *2-d*
Avtec Systems, Inc.
10530 Rosehaven Street
Suite 300
Fairfax, VA 22030
PHONE: 703-273-2211; FAX: 703-273-1313
EMAIL:

LEGEND:

NIST ISEE Working Groups:
1. Object Management
2. Process and Task Management
3. Interface & Platform Services
4. User Interface

PCIS Working Groups:
a. Needs of the Tool Builders
b. Needs of the Platform Supplier
c. Needs of the Environment Supplier
d. Needs of the Environment Users

8/7/91 5:01 PM

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Shorrock, Tim — *PCIS d*
British Aerospace Military Aircraft
Software Technology Dept.
Warton Aerodrome
Preston,
Lancs. PR4 1AX
ENGLAND
PHONE: 44-772-854630; FAX: 44-772-855216
EMAIL:

Solomond, John — *PCIS*
Ada Joint Program Office
The Pentagon Room 3E114
Washington, D.C. 20301-3081
PHONE: 703-614-0209; FAX: 703-685-7019
EMAIL: solomond@AJPO.SEI.CMU.EDU

Song, Wha Dal — *NIST/PCIS/IWCASE 1-c*
Arizona State University
1050 S. Stanley Pl. #123
Tempe, AZ 85281
PHONE: 602-967-8348; FAX:
EMAIL:

Sound, Andrew — *PCIS*
Naval Weapons Center
Code 3922
Naval Weapons Center
China Lake, CA 93555-6001
PHONE: 619-939-1286; FAX: 619-939-4754
EMAIL: andrew@sol.nwc.navy.mil

Stenning, Vic — *PCIS*
Anshar Limited
Thriftswood
Stevens Hill
Yateley
Camberley, Surrey GU17 7AY
UNITED KINGDOM
PHONE: +44 252 874 736; FAX: +44 252 874 736
EMAIL:

LEGEND:

NIST ISEE Working Groups:
1. *Object Management*
2. *Process and Task Management*
3. *Interface & Platform Services*
4. *User Interface*

PCIS Working Groups:
a. *Needs of the Tool Builders*
b. *Needs of the Platform Supplier*
c. *Needs of the Environment Supplier*
d. *Needs of the Environment Users*

8/7/91 5:01 PM

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Strelich, Tom *NIST*
General Research Corp.
5383 Hollister Ave.
Santa Barbara, CA 93105
PHONE: 805/964-7724 x327; FAX:
EMAIL:

Tachiqami, Akira *NIST/PCIS 3*
890 Kashimada Saiwai Kawasaki
Kanaqawa 211
JAPAN
PHONE: 044 540-1111; FAX: 044 549 1189
EMAIL:

Thomas, Ian *NIST/PCIS*
1266 Kifer Road
Sunnyvale, CA 94043

PHONE: 408/746-5157; FAX: 408/746-5539
EMAIL: thomas@hp-ses.sde.hp.com

Tily, C.N.J. *PCIS 3-d*
UK MoD
Room 418 Turnstile House
98 High Holborn
London WC1V 6LL
ENGLAND
PHONE: +44 71 430 5789; FAX: +44 71 430-5650
EMAIL:

Toyonag, Masato *NIST/PCIS*
6-81 ONOECHO NAKA
YOKOHAMA
JAPAN
PHONE: +81-045-681-2111; FAX: +81-045-681-2656
EMAIL: matsu@aqu.hitachi-sk.co.jp

LEGEND:

NIST ISEE Working Groups:
1. Object Management
2. Process and Task Management
3. Interface & Platform Services
4. User Interface

PCIS Working Groups:
a. Needs of the Tool Builders
b. Needs of the Platform Supplier
c. Needs of the Environment Supplier
d. Needs of the Environment Users

NIST/PCIS/IWCASE ATTENDANCE 3-7 JUNE 1991

Attending Which Workshop(s) & Working Group(s)

Warner, Don A. — *PCIS* c
Atherton Technology
5020 Campus Blvd.
New Port Beach, CA 92660
PHONE: 714-955-3665; FAX: 714-955-3762
EMAIL:

Wong, William — *NIST/PCIS/IWCASE*
NIST
Bldg. 225, Room B266
Gaithersburg, MD 20899
PHONE: 301-975-3341; FAX: 301-590-0932
EMAIL: wong@swe.ncsl.nist.gov

Wybolt, Nicholas — *NIST/PCIS/IWCASE* *1,3-a*
Cadre Technologies Inc.
222 Richmond Street
Providence, RI 02903
PHONE: 401-351-5950; FAX: 401-351-7380
EMAIL:

Zelkowitz, Marv — *NIST*
NIST/NCSL
Bldg. 225, Room B266
Gaithersburg, MD 20899
PHONE: 301-405-2690; FAX: 301-405-6707
EMAIL: mvz@cs.UMD.EDU

LEGEND:

NIST ISEE Working Groups:
1. *Object Management*
2. *Process and Task Management*
3. *Interface & Platform Services*
4. *User Interface*

PCIS Working Groups:
a. *Needs of the Tool Builders*
b. *Needs of the Platform Supplier*
c. *Needs of the Environment Supplier*
d. *Needs of the Environment Users*

8/7/91 5:01 PM

This document contains blank pages that were not filmed

This Document Contains Page/s Reproduced From Best Available Copy

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

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

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1991	Final

4. TITLE AND SUBTITLE
Proceedings of the Second Portable Common Interface Set (PCIS) Workshop: Interface Technology Analyses (ITA2)

5. FUNDING NUMBERS
MDA 903 89 C 0003

Task T-D5-496

6. AUTHOR(S)
Clyde G. Roby

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses (IDA)
1801 N. Beauregard St.
Alexandria, VA 22311-1772

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Document D-1047

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Ada Joint Program Office
DISA
500 North Washington Street, Suite 200
Falls Church, VA 22046

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, unlimited distribution: ~~18 March 1994~~.

12b. DISTRIBUTION CODE
2A

13. ABSTRACT (Maximum 200 words)

This document contains information, in the form of written reports and slide presentations, presented at the second PCIS (Portable Common Interface Set) Workshop, held in conjunction with the NIST ISEE (Integrated Software Engineering Environment) Workshop and the IWCASE (International Workshop on Computer-aided Software Engineering) Workshop, 3-7 June 1991 in Redondo Beach, California. This document was prepared for the Ada Joint Program Office (AJPO) in support of the NATO Special Working Group (SWG) on Ada Programming Support Environments (APSEs) Project. Several NATO nations formed this Special Working Group so that their Ministries of Defense, including the US DoD, might achieve the economies projected by using standard APSEs. The SWG is tasked to develop an interface set, known as PCIS, based upon current and emerging interface technologies. In 1990, the AJPO requested IDA support in the PCIS Programme by producing the IRAC (International Requirements and Design Criteria) for PCIS. During 1991, workshops were held to define the requirements and assess interface technologies in order to produce the IRAC and the PCIS requirements documents.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Interface Technology; Portable Common Interface Set (PCIS); Software Engineering Environment (SEE); Computer-aided Software Engineering (CASE)	808
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR