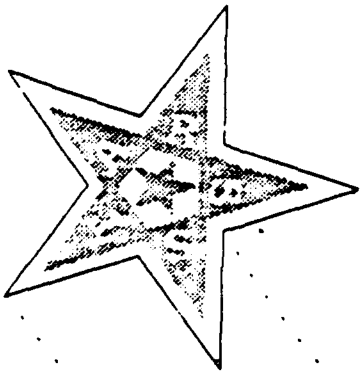
STARS '91 Proceedings

December 3 & 4, 1991



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 3-4 December 1991	3. REPORT TYPE AND DATES COVERED Proceedings

4. TITLE AND SUBTITLE

STARS '91 Proceedings 3-4 December, 1991

5 FUNDING NUMBERS

F19628-88-D-0031

6. AUTHOR(S)

Paramax Corporation

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Paramax Corporation
12010 Sunrise Valley Drive
Reston, VA 22091

8. PERFORMING ORGANIZATION REPORT NUMBER

04014

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of the Air Force
Headquarters Electronic Systmes Division
Hanscom, AFB MA 01731-5000

10. SPONSORING MONITORING AGENCY REPORT NUMBER

04014

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT

Distribution "A"

This document has been approved for public release and sale; its distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

These proceedings introduce you the concept of megaprogramming and describe the role STARS is playing in the transition to this new paradigm.

14 SUBJECT TERMS

15 NUMBER OF PAGES

51

16 PRICE CODE

17 SECURITY CLASSIFICATION OF REPORT unclassified	18 SECURITY CLASSIFICATION OF THIS PAGE unclassified	19 SECURITY CLASSIFICATION OF ABSTRACT unclassified	20 LIMITATION OF ABSTRACT SAR

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	-	Contract	**PR**	-	Project
G	-	Grant	**TA**	-	Task
PE	-	Program Element	**WU**	-	Work Unit Accession No

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD	-	See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE	-	See authorities.
NASA	-	See Handbook NHB 2200.2.
NTIS	-	Leave blank.

Block 12b. Distribution Code.

DOD	-	Leave blank.
DOE	-	Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA	-	Leave blank
NTIS	-	Leave blank

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

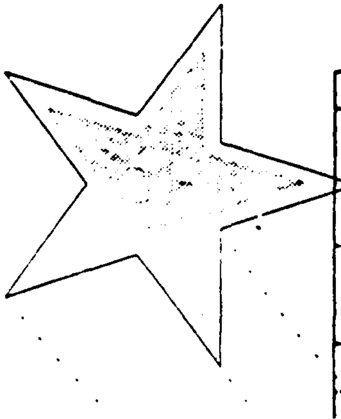
Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

STARS '91 Proceedings

December 3 & 4, 1991



STARS Technology Center
Suite 400
801 North Randolph Street
Arlington, Va. 22203

TIPS FOR READERS

The page numbers shown on the individual presentations run sequentially within each track and within the Plenary Sessions. This is reflected, as well, in the Contents. However, the page numbers shown in the *Authors Index* found at the back of these proceedings include a coded prefix (P, 1, 2, 3, or 4) to indicate the part of the proceedings in which the material can be found.

Contents

Plenary Sessions

Track 1 – Process Driven Development

Track 2 – Domain Specific Reuse

Contents (Continued)

Track 3 -- Technology Support

Contents (Continued)

Track 4 -- Technology Transition

Program Chair's Statement

The goal of the STARS program is to increase productivity, reliability, and quality of DoD application software. STARS is approaching this by synergistically integrating support for modern software development processes and modern reuse concepts within state-of-the-art software engineering environment technology. STARS is focused on accelerating a change in the way software is developed within the DoD. This change represents a shift to a megaprogramming paradigm.

The STARS program would like you to "Join us in the transition to megaprogramming." The conference program has been designed to introduce you to the concepts of megaprogramming and describe the role STARS is playing in the transition to this new paradigm. The plenary session will provide you a high level overview as well as some economic analysis of the potential benefits of megaprogramming.

The closing discussion by Dr. Barry Boehm will describe the relationship of megaprogramming and STARS to the DoD Software technology plan.

Three of the four track sessions will focus on the major elements of megaprogramming:

- Process-driven development,
- Domain-specific reuse, and
- Technology support.

The fourth track will concentrate on the STARS technology transition activities associated with accelerating the shift to this new way of doing business. The format of these track sessions is intended to provide ample opportunity for discussion and informal interchange.

The STARS program needs your help to move these concepts, processes, and technologies into widespread use. To this end, we are soliciting you to become part of the STARS Affiliates program. The STARS Director, John Foreman, will describe the program during the plenary session and there will be evening sessions for those of you considering becoming Technology Transition Affiliates to discuss your interests with members of the STARS staff.

My sincere thanks go to the members of the program committee who coordinated the selection of topics and the development of the presentations. Most of the credit for the program, however, must go to the individual presenters who put a great deal of personal effort into creating the 34 presentations you have in your proceedings.

I would also like to thank the conference chair, Don Harmon, the conference committee, and the publications staffs for a splendid job of organizing this conference. And finally, my special thanks to BGen Denis Brown for making time in his schedule to address us at STARS' 91.

Dick Drake
Program Chair, STARS '91

IBM Corporation
800 North Frederick Ave.
Gaithersburg, MD 20879
Internet: DDrake@AJPO.SEI.CMU.Edu

Conference Committee

Chair
Don Harmon
Unisys Defense Systems, Inc.

Program Chair
Dick Drake
IBM

Treasurer
Ray Grimes
IBM

Publicity Co-Chairs

Joel Trimble	R.C. King
SEI	Boeing

Exhibits Chair
Terry Courtwright
CEA

Proceedings Chair
Bob Kleinfeld
Unisys Defense Systems, Inc.

Graphic Arts/Publications Chair
Jean Gardner
Unisys Defense Systems, Inc.

Program Committee

John Foreman
DARPA STARS
Suite 400
801 North Randolph Street
Arlington, VA 22203
jtf@sei.cmu.edu

Dick Drake
IBM Corporation
Bldg 182/3H96
18100 Frederick Pike
Gaithersburg, MD 20879
ddrake@ajpo.sei.cmu.edu

Larry Frank
Boeing
Suite 400
801 North Randolph Street
Arlington, VA 22203
frank@stars.rosslyn.unisys.com

Jim Henslee
USAF ESD/AVS
Bldg 1704
Hanscom AFB, MA 01731
henslee@gw1.hanscom.af.mil

Teri Payton
Unisys Defense Systems, Inc.
12010 Sunrise Valley Drive
Reston, VA 22091
payton@stars.reston.unisys.com

STARS '91 Agenda
December 3-4, 1991

Day 1, Tuesday, December 3, 1991
Lobby Level

Grand Foyer	7:30am - 8:30am	Registration and Continental Breakfast	
Grand A	8:30am - 8:45am	STARS' 91 Overview	Don Harmon (Unisys)
	8:45am - 9:05am	STARS and Megaprogramming	Dr. Barry Boehm (DARPA)
	9:05am - 9:45am	STARS Vision, Mission, Strategy and Achievements	John Foreman (DARPA)
	9:45am - 10:15am	Break	
	10:15am - 10:45am	Economic Impact of STARS Supported Technology	Dr. Thomas P. Frazier (IDA)
	10:45am - 11:15am	Technology Transfer and Community Involvement	John Foreman (DARPA)
	11:15am - 11:45am	Questions and Answers	John Foreman (DARPA)
Junior Ballroom & Grand B	11:45am - 2:00pm	Demonstrations (buffet lunch available)	

Day 1, Tuesday, December 3, 1991

Time	Event and Location			
	Pavilion 22	Pavilion 23	Mezzanine 2	Mezzanine 3
2:00pm-2:45 pm	Process Driven Development Vision, Strategies and Achievements Dick Drake, IBM	Domain Specific Reuse: Vision, Strategies and Achievements Teri Payton, Unisys Defense Systems, Inc.	Technology Support: Vision, Strategy, and Achievements Larry Frank, Boeing	Software Technology Transition Process Priscilla Fowler, SEI
2:45pm-3:15pm	Break and informal discussions			
3:15pm-4:00pm	Process Concepts Dr. James E. King, Boeing	Reuse Concepts Maggie Davis, Boeing	Project Support Environment Services Reference Model Dr. Peter Feiler, SEI	STARS Technology Transition Strategies Joe Morin, SEI
4:00pm-4:30pm	Break and informal discussions			
4:30pm-5:15pm	Process Asset Library Jim Over, SEI	Integrating Reuse into a Life-Cycle Process Bonnie Danner, TRW Domain Analysis Process Model Dr. Ruben Prieto-Diaz, Reuse, Inc.	Stars Standards Portfolio Jim Hamilton, Boeing STARS Role in Standards Maturation Bob Ekman, IBM	Reuse Acquisition Issues Bob Bowes, DSD Laboratories

Day 1, Tuesday, December 3, 1991

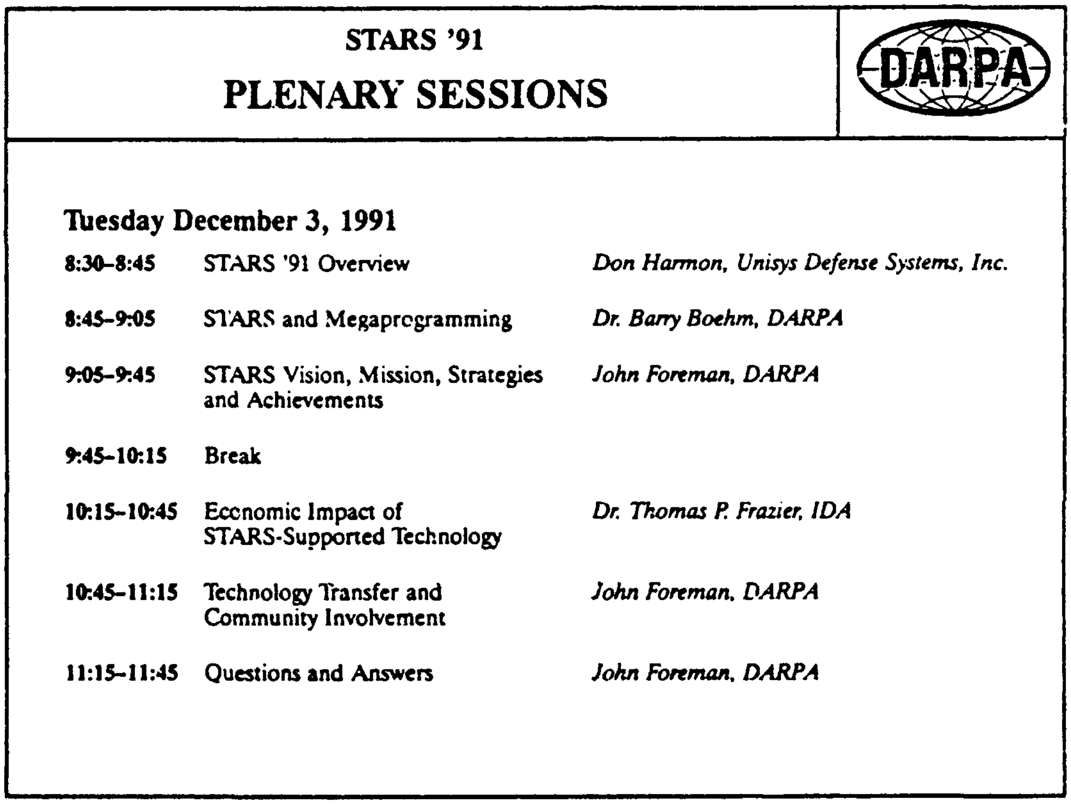
Time	Event and Location			
5:30pm-6:00pm	Invited Speaker: Denis M. Brown (DISA/Center for Information Management) Grand Ballroom A			
6:00pm-8:00pm	Demonstrations and Reception: Junior Ballroom and Grand Ballroom B			
	Pavilion 22	Pavilion 23	Mezzanine 2	Mezzanine 3
8:00pm-9:30pm	Community Involvement Working Group Process Driven Development Dick Drake, IBM	Community Involvement Working Group Domain Specific Reuse Teri Payton, Unisys Defense Systems, Inc.	SEMATECH: Software Methods and Tools Program Jeffrey Kantor and Claude Baudoin, SEMATECH Community Involvement Working Group Technology Support Larry Frank, Boeing	Community Involvement Working Group Technology Transition Joe Morin, SEI

Day 2, Wednesday, December 4, 1991

Time	Event and Location			
	Pavilion 22	Pavilion 23	Mezzanine 2	Mezzanine 3
7:30am-8:30am	Registration and Continental Breakfast			
8:30am-9:15am	Experiment in Process Definition and Representation Carol Klingler, TRW	STARS Asset Library Open Architecture Framework (ALOAF) Dick Creps, Unisys Defense Systems, Inc.	IBM STARS SEE Evolution Strategy Mary Catherine Ward, IBM	STARS Demonstration Projects Dan Burton, SEI
9:15am-9:45am	Break and informal discussions			
9:45am-10:30am	Enacting the Software Process William H. Ett, IBM	STARS Library Mechanisms: Comparisons and Experiences Marlene Hazle, MITRE	Unisys STARS SEE Evolution Strategy Dr. Thomas E. Shields, Unisys Defense Systems, Inc.	*Megaprogramming Adoption Risks and Strategy Discussion* Dr. Jerry Pixton, Unisys Defense Systems, Inc.
10:30am-11:00am	Break and informal discussions			
11:00am-11:45am	Process Measurement Hal Hart, TRW	ASSET Jim Moore, IBM CARDS Rosa Armstrong, EWA	Boeing STARS SEE Evolution Strategy John Neorr, Boeing	CARDS Reuse Blueprint Hans Polzer, Unisys Defense Systems, Inc.

Day 2, Wednesday, December 4, 1991

Time	Event and Location			
11:45am-1:45pm	Buffet lunch: Grand Ballroom B			
11:45am-2:30pm	Demonstrations: Junior Ballroom			
1:45pm-2:30pm	Pavilion 22	Pavilion 23	Mezzanine 2	Mezzanine 3
	Process Driven Development Feedback Session Bill Hodges, Boeing	Domain Specific Reuse Feedback Session Dave Ceoly, IBM	Technology Support Feedback Session Hans Polzer, Unisys Defense Systems, Inc.	Technology Transition Feedback Session Jim Henslee, USAF ESD
2:30pm-3:00pm	Break and informal discussions			
3:00pm-3:40pm	Grand Ballroom A			
	DoD Software Technology Plan and STARS Dr. Barry Boehm (DARPA)			
3:40pm-4:00pm	Closing Remarks John Foreman (DARPA)			

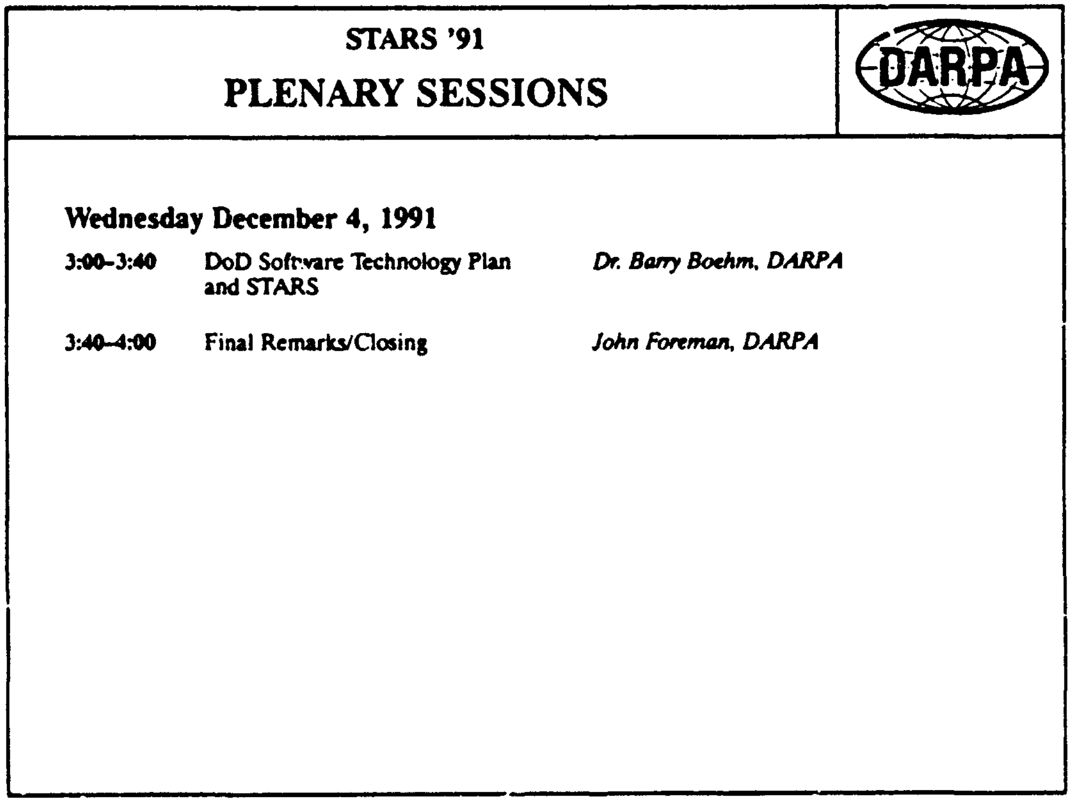


STARS '91

PLENARY SESSIONS

Tuesday December 3, 1991

8:30–8:45	STARS '91 Overview	*Don Harmon, Unisys Defense Systems, Inc.*
8:45–9:05	S1ARS and Megaprcgramming	*Dr. Barry Boehm, DARPA*
9:05–9:45	STARS Vision, Mission, Strategies and Achievements	*John Foreman, DARPA*
9:45–10:15	Break	
10:15–10:45	Eccnomic Impact of STARS-Supported Technology	*Dr. Thomas P. Frazier, IDA*
10:45–11:15	Technology Transfer and Community Involvement	*John Foreman, DARPA*
11:15–11:45	Questions and Answers	*John Foreman, DARPA*

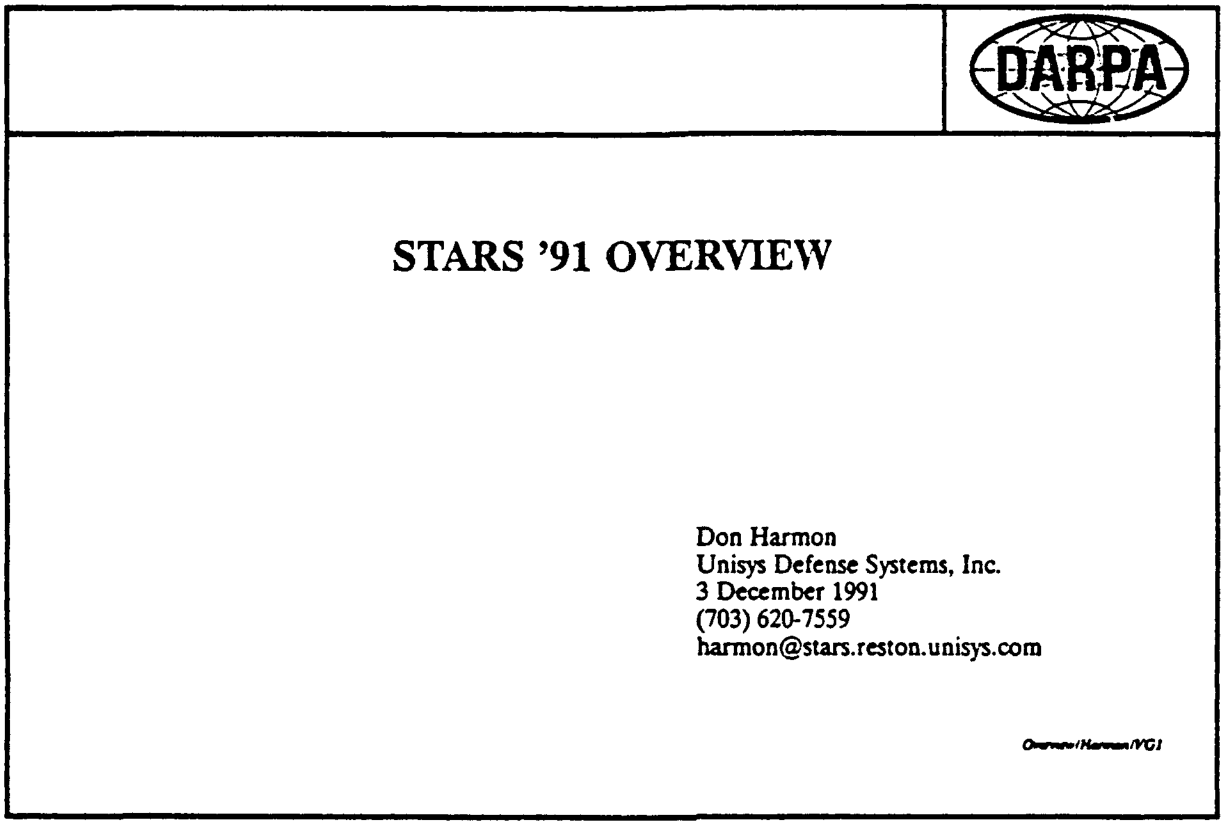


STARS '91

PLENARY SESSIONS

Wednesday December 4, 1991

3:00–3:40	DoD Software Technology Plan and STARS	*Dr. Barry Boehm, DARPA*
3:40–4:00	Final Remarks/Closing	*John Foreman, DARPA*



DARPA

STARS '91 OVERVIEW

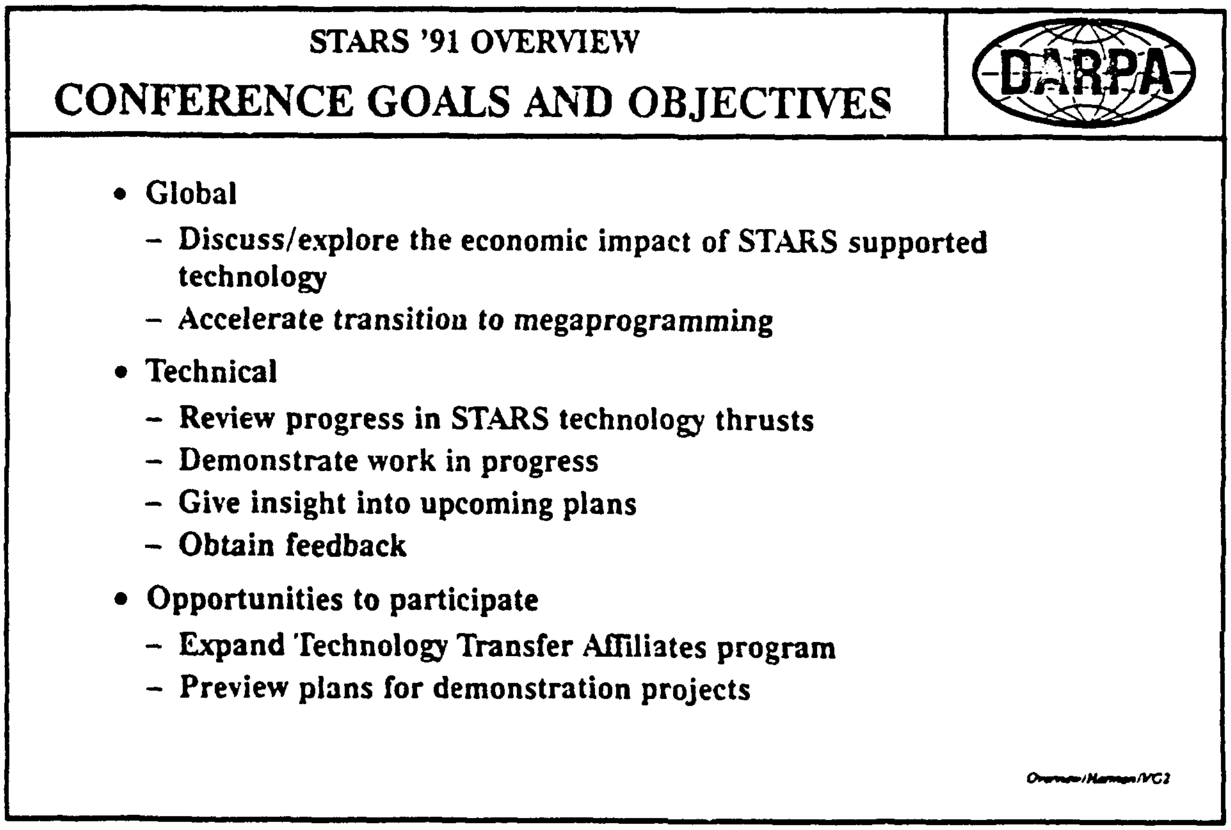
Don Harmon
Unisys Defense Systems, Inc.
3 December 1991
(703) 620-7559
harmon@stars.reston.unisys.com

Overview/Harmon/VG1

Good morning! Welcome to STARS'91. Let me spend the first few minutes explaining our program so you can get the most out of the next two days.

Our conference theme is "Join The Transition to Megaprogramming" and, consistent with this theme, we have selected some ambitious goals and objectives for our program. In STARS'91, we will provide you with detailed presentations and demonstrations of our results to date and obtain feedback from you. And we will tell you how your organization can join in the transition to megaprogramming.

If you look toward the screen you can see where we have summarized the STARS'91 goals.



STARS '91 OVERVIEW

CONFERENCE GOALS AND OBJECTIVES

DARPA

- Global
  - Discuss/explore the economic impact of STARS supported technology
  - Accelerate transitiou to megaprogramming
- Technical
  - Review progress in STARS technology thrusts
  - Demonstrate work in progress
  - Give insight into upcoming plans
  - Obtain feedback
- Opportunities to participate
  - Expand Technology Transfer Affiliates program
  - Preview plans for demonstration projects

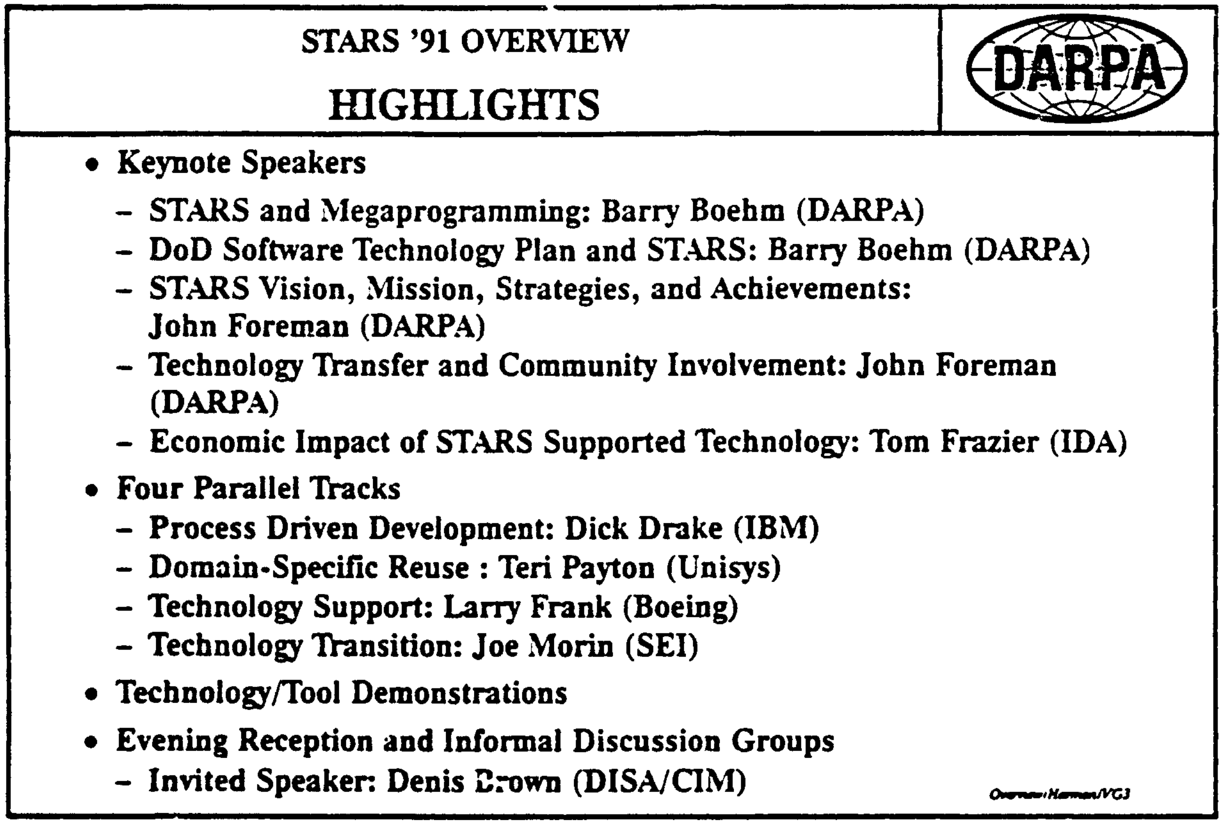
Overview/Harmon/VG2

We start with the global objectives of discussing the economic benefits of STARS supported technology and of accelerating the transition tc megaprogramming.

Our technical goals consist of reviewing STARS technical work, and secondly—getting your feedback on this work.

We also will present opportunities for you to participate with STARS. In part this will come from an expan:ion of our Technology Transfer Affiliates Program, and, in part, by previewing the plans for our Demonstration Projects.

Next, le! me go over th: highlights of our program.



STARS '91 OVERVIEW

HIGHLIGHTS

- Keynote Speakers
  - STARS and Megaprogramming: Barry Boehm (DARPA)
  - DoD Software Technology Plan and STARS: Barry Boehm (DARPA)
  - STARS Vision, Mission, Strategies, and Achievements: John Foreman (DARPA)
  - Technology Transfer and Community Involvement: John Foreman (DARPA)
  - Economic Impact of STARS Supported Technology: Tom Frazier (IDA)
- Four Parallel Tracks
  - Process Driven Development: Dick Drake (IBM)
  - Domain-Specific Reuse : Teri Payton (Unisys)
  - Technology Support: Larry Frank (Boeing)
  - Technology Transition: Joe Morin (SEI)
- Technology/Tool Demonstrations
- Evening Reception and Informal Discussion Groups
  - Invited Speaker: Denis Brown (DISA/CIM)

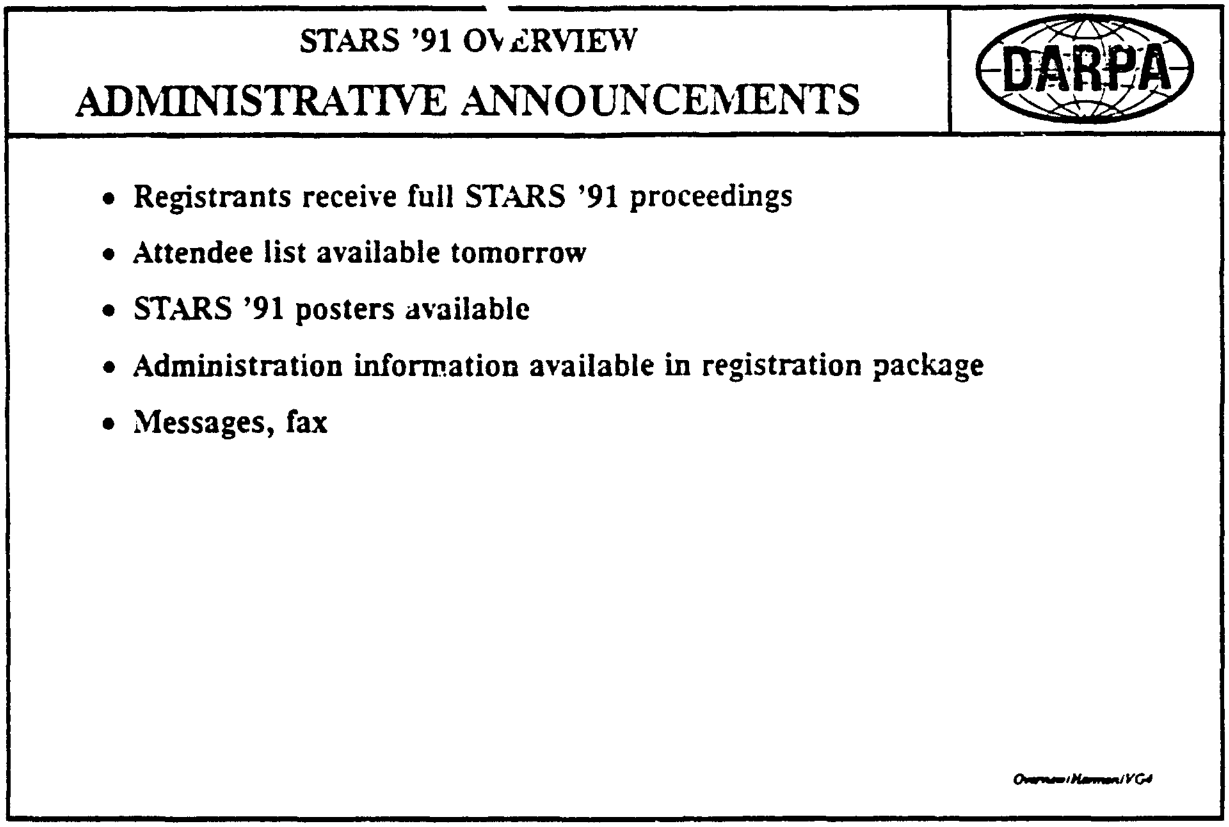
Overview/Harman/VG3

STARS'91 is fortunate to have several distinguished keynote speakers who will present the overall STARS strategy. These include Dr. Barry Poehm, Director of DARPA/SISTO, who will talk on "STARS and Megaprogramming". Dr. Boehm will also be on our program tomorrow to discuss the "DoD Software Technology Plan and STARS". John Foreman, the DARPA/STARS Program Director, will discuss the "STARS Vision, Mission, and Strategy and Achievements", as well as a second talk on "Technology Transfer and Community Involvement". We also have Dr. Thomas P. Frazier who will provide some insight into the "Economic Impact of STARS Supported Technology".

The main body of our program will feature four tracks. These will be led by the three STARS System Architects: Dick Drake of IBM, Teri Payton of Unisys, and Larry Frank of Boeing. The fourth Track, Technology Transfer, will be led by Joe Morin of the SEI.

We are fortunate to have Denis M. Brown (BGen USAF, Ret.), Director of DISA/CIM, as our invited guest speaker this evening. Finally, we have a reception planned for this evening to be followed by informal discussion groups. You will hear more about these from the Track Chairs.

Next, our administrative announcements.



STARS '91 OVERVIEW

ADMINISTRATIVE ANNOUNCEMENTS

DARPA

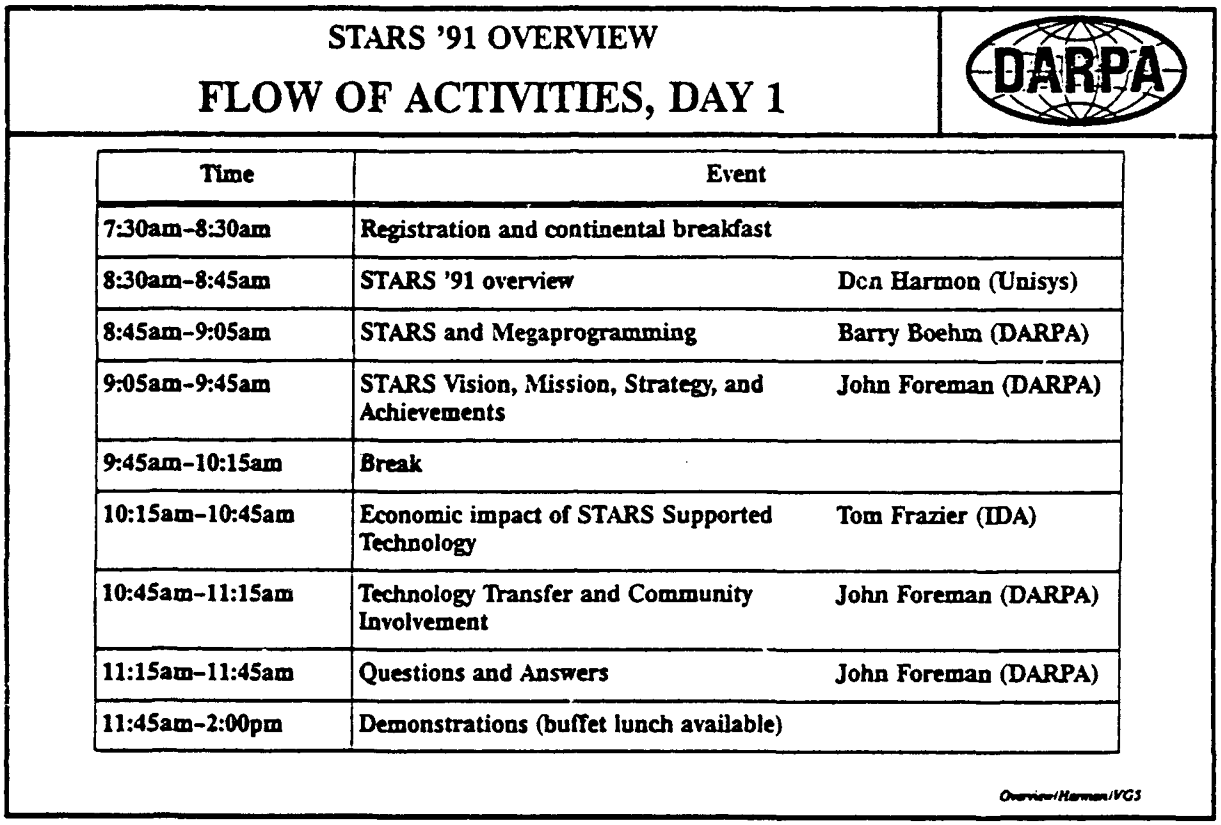
- Registrants receive full STARS '91 proceedings
- Attendee list available tomorrow
- STARS '91 posters available
- Administration information available in registration package
- Messages, fax

Overview/Harmon/VG4

Each of you should have a 3-ring binder containing the STARS'91 Proceedings. This binder contains copies of all briefings. Tomorrow we will have copies of the attendee lists available at the registration desk which can be added to the presentation materials.

STARS'91 posters are also available at the registration desk. Please help yourself.

If you want to mail material back to your home station, you can take advantage of our mailing service located in the lobby. You will also find a .r.essage center there for telephone calls.



STARS '91 OVERVIEW

FLOW OF ACTIVITIES, DAY 1

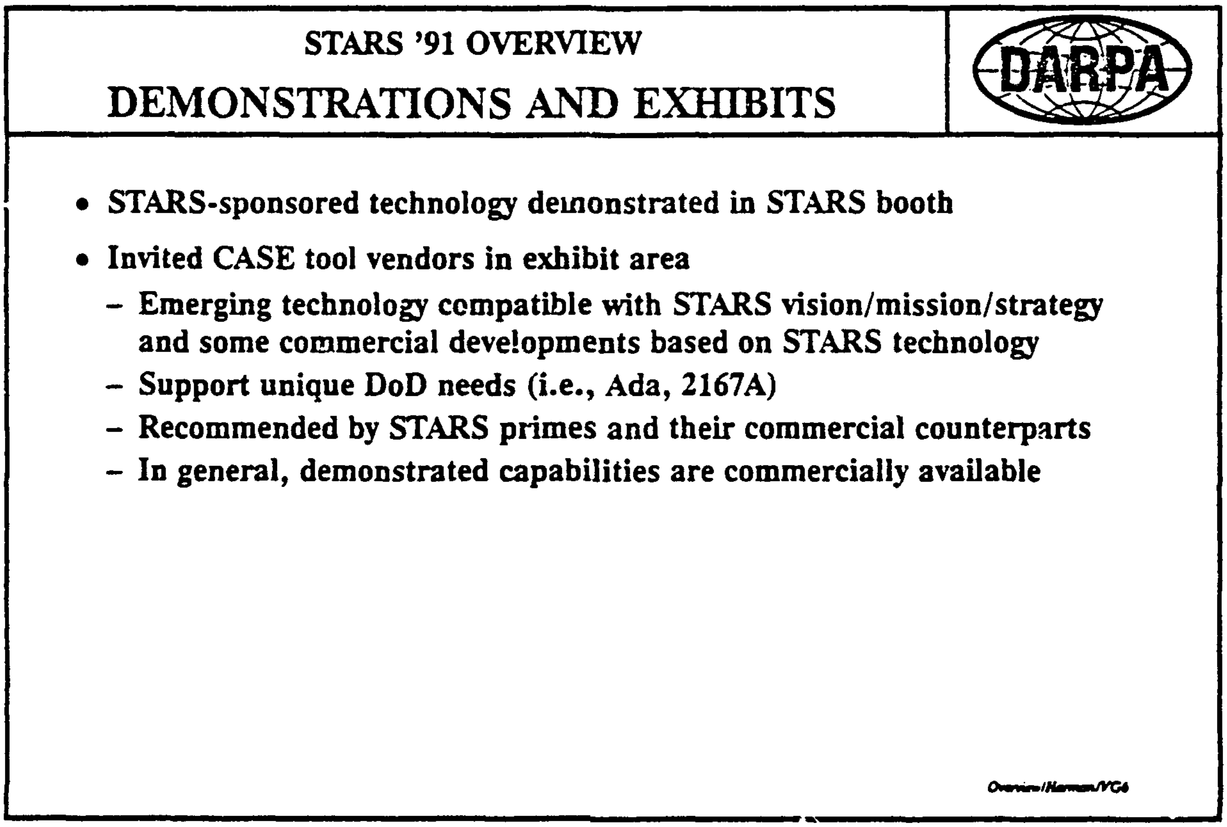
DARPA

Time	Event	
7:30am-8:30am	Registration and continental breakfast	
8:30am-8:45am	STARS '91 overview	Dcn Harmon (Unisys)
8:45am-9:05am	STARS and Megaprogramming	Barry Boehm (DARPA)
9:05am-9:45am	STARS Vision, Mission, Strategy, and Achievements	John Foreman (DARPA)
9:45am-10:15am	Break	
10:15am-10:45am	Economic impact of STARS Supported Technology	Tom Frazier (IDA)
10:45am-11:15am	Technology Transfer and Community Involvement	John Foreman (DARPA)
11:15am-11:45am	Questions and Answers	John Foreman (DARPA)
11:45am-2:00pm	Demonstrations (buffet lunch available)	

Overview/Harmon/VGS

At a glance, here are this morning's activities. Note that they all take place here in the Grand Ballroom. Our keynote speakers, Barry Boehm, John Foreman and Tom Frazier will provide the strategic themes that establish the context for STARS, tie the various pieces of the program together, and provide an overview of future plans. John Foreman will close this morning's sessions with a Q & A period.

Starting at 11:45 we have an extended lunch period. This provides you with an opportunity to visit the demonstrations and exhibits. These are setup in the Junior Ballroom which is adjacent to the dining area.



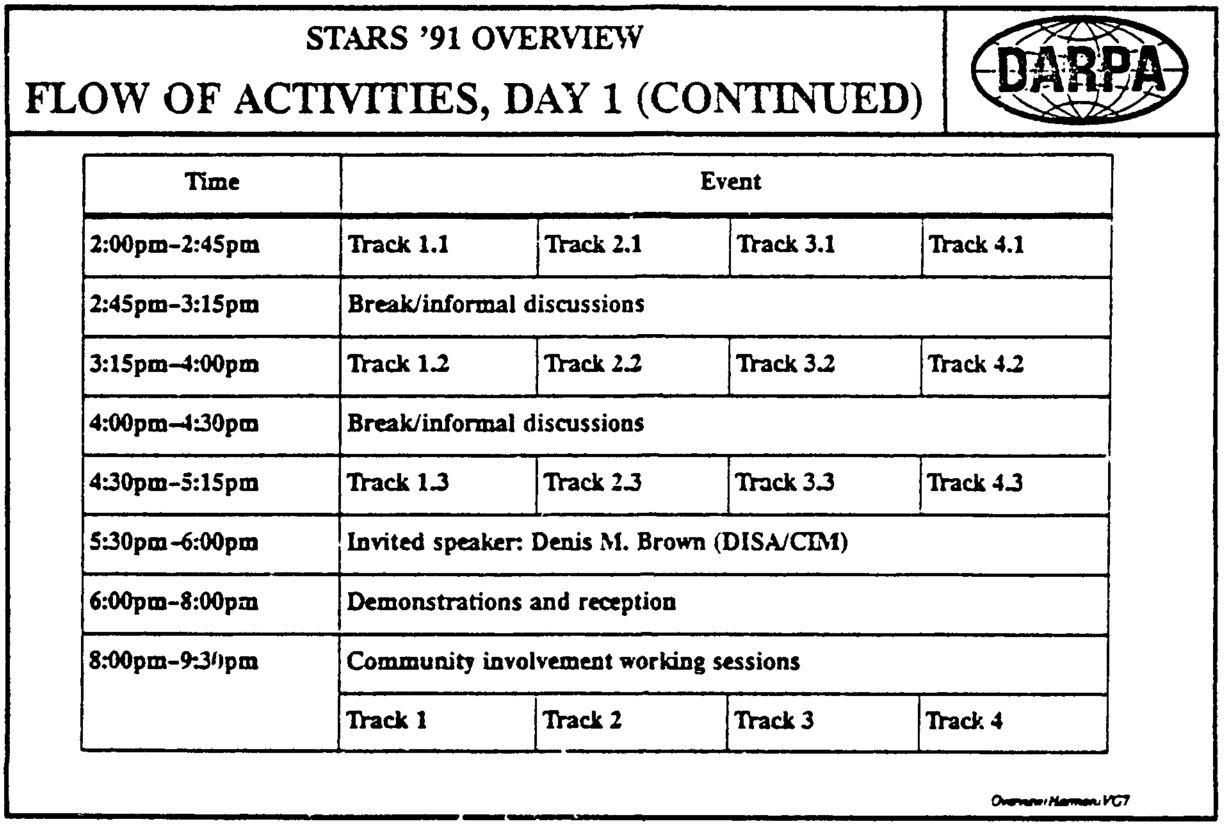
STARS '91 OVERVIEW

DEMONSTRATIONS AND EXHIBITS

- STARS-sponsored technology demonstrated in STARS booth
- Invited CASE tool vendors in exhibit area
  - Emerging technology compatible with STARS vision/mission/strategy and some commercial developments based on STARS technology
  - Support unique DoD needs (i.e., Ada, 2167A)
  - Recommended by STARS primes and their commercial counterparts
  - In general, demonstrated capabilities are commercially available

Overview/Harman/VG6

Next let me digress a minute and talk about the STARS'91 Exhibits and Demonstrations. These consist of a STARS booth in which STARS-sponsored technology is demonstrated. The STARS booth is located just inside the Lord Fairfax room where we will have our buffet lunch. In the adjoining Junior Ballroom, the invited CASE Tool vendors will hold their demonstrations. These companies have been selected because their products are consistent with the STARS vision, mission and strategy.



STARS '91 OVERVIEW

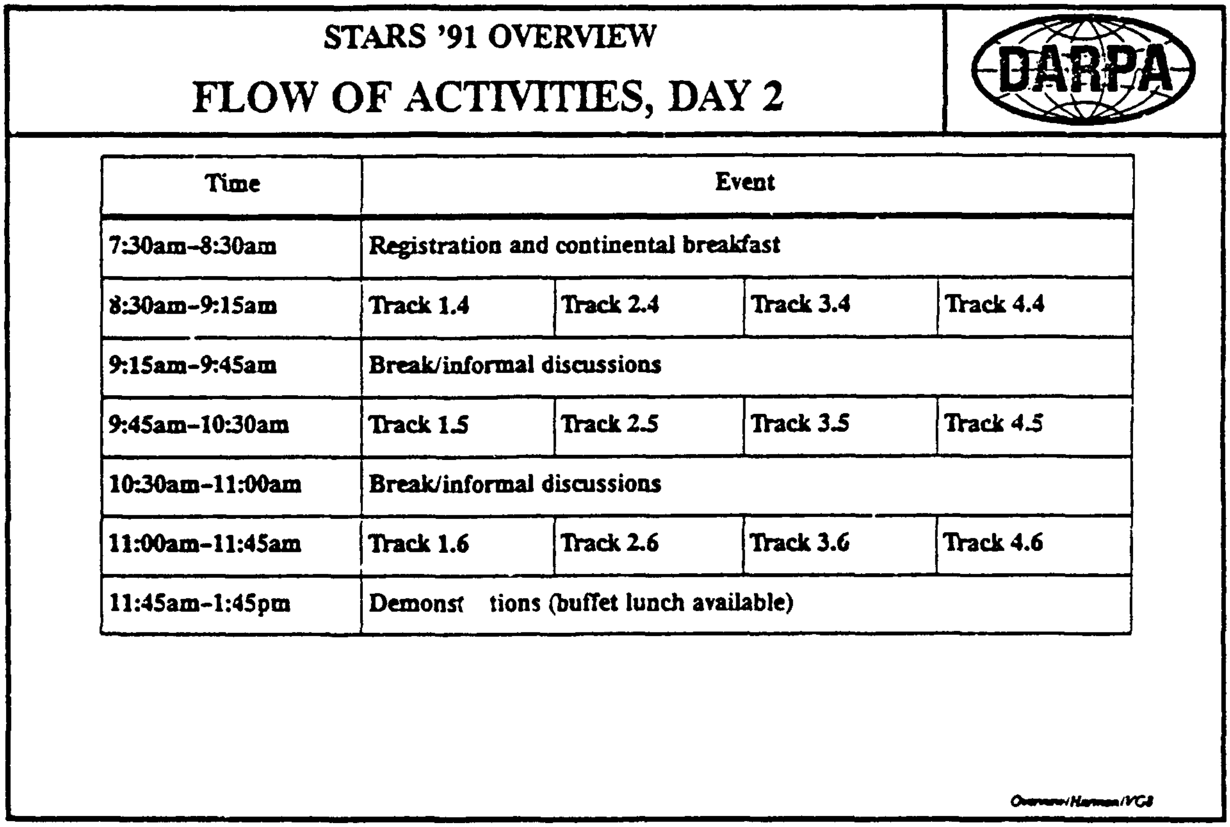
FLOW OF ACTIVITIES, DAY 1 (CONTINUED)

Time	Event			
2:00pm-2:45pm	Track 1.1	Track 2.1	Track 3.1	Track 4.1
2:45pm-3:15pm	Break/informal discussions			
3:15pm-4:00pm	Track 1.2	Track 2.2	Track 3.2	Track 4.2
4:00pm-4:30pm	Break/informal discussions			
4:30pm-5:15pm	Track 1.3	Track 2.3	Track 3.3	Track 4.3
5:30pm-6:00pm	Invited speaker: Denis M. Brown (DISA/CIM)			
6:00pm-8:00pm	Demonstrations and reception			
8:00pm-9:30pm	Community involvement working sessions			
	Track 1	Track 2	Track 3	Track 4

Overview/Harman/VG7

This afternoon we will break into the parallel tracks. Within each track there will be six sessions of 45 minutes followed by a thirty period of informal discussions and break. On this vu-graph the notation Track x.y is used, where x is numbered 1 to 4 to indicate one of the four tracks, and y indicates the session within the track. Generally, people will want to stay within a track, but the breaks are structured so that individuals can cross from one track to another. Track room assignments are shown on the signs outside this meeting area.

This evening we continue with the demonstrations and exhibits in the Junior Ballroom plus a reception has been scheduled. And later, starting at 8 pm we have informal working discussions. You will hear more about these in the Track sessions. Also don't forget, Denis M. Brown will speak at 5:30 in this room.



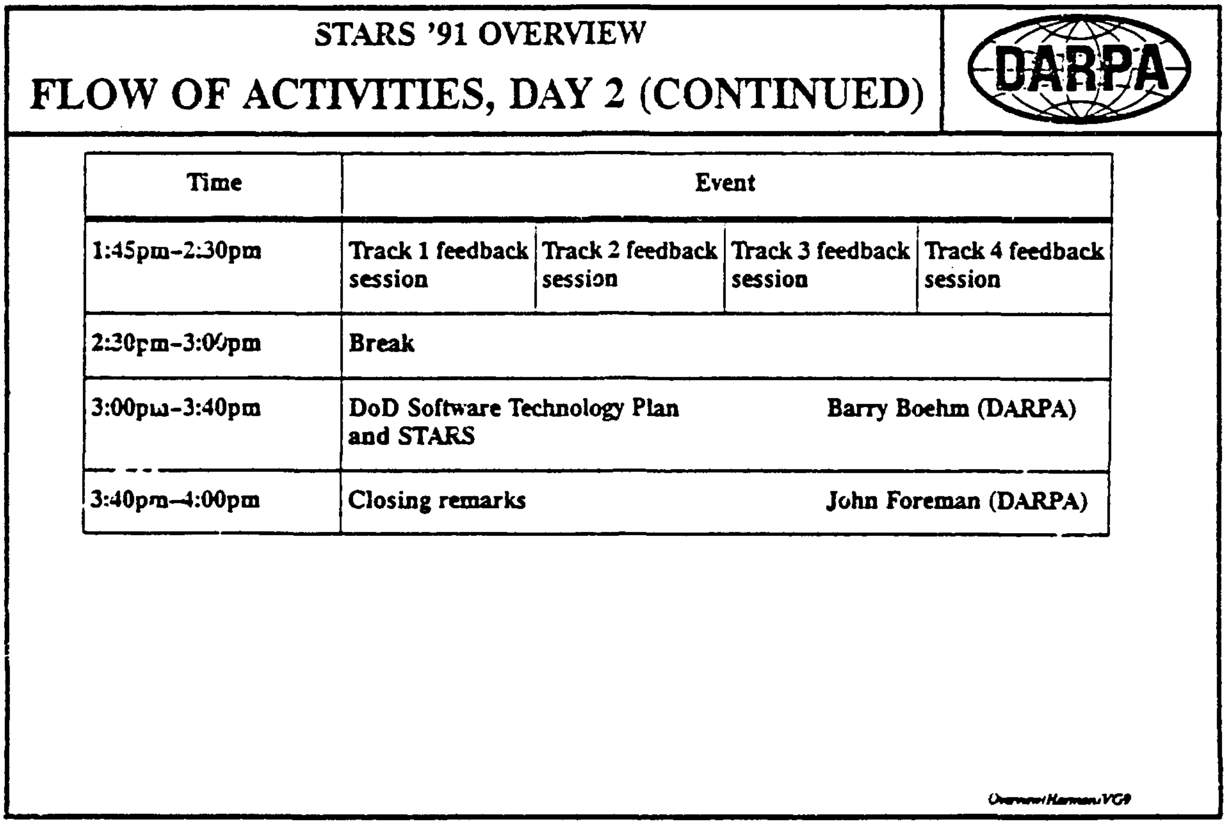
STARS '91 OVERVIEW

FLOW OF ACTIVITIES, DAY 2

Time	Event			
7:30am-8:30am	Registration and continental breakfast			
8:30am-9:15am	Track 1.4	Track 2.4	Track 3.4	Track 4.4
9:15am-9:45am	Break/informal discussions			
9:45am-10:30am	Track 1.5	Track 2.5	Track 3.5	Track 4.5
10:30am-11:00am	Break/informal discussions			
11:00am-11:45am	Track 1.6	Track 2.6	Track 3.6	Track 4.6
11:45am-1:45pm	Demonst tions (buffet lunch available)			

Overview/Harmon/VG8

Day 2 continues with the parallel Track sessions. The lunch and demo period will be the same as on the previous day.



STARS '91 OVERVIEW

FLOW OF ACTIVITIES, DAY 2 (CONTINUED)

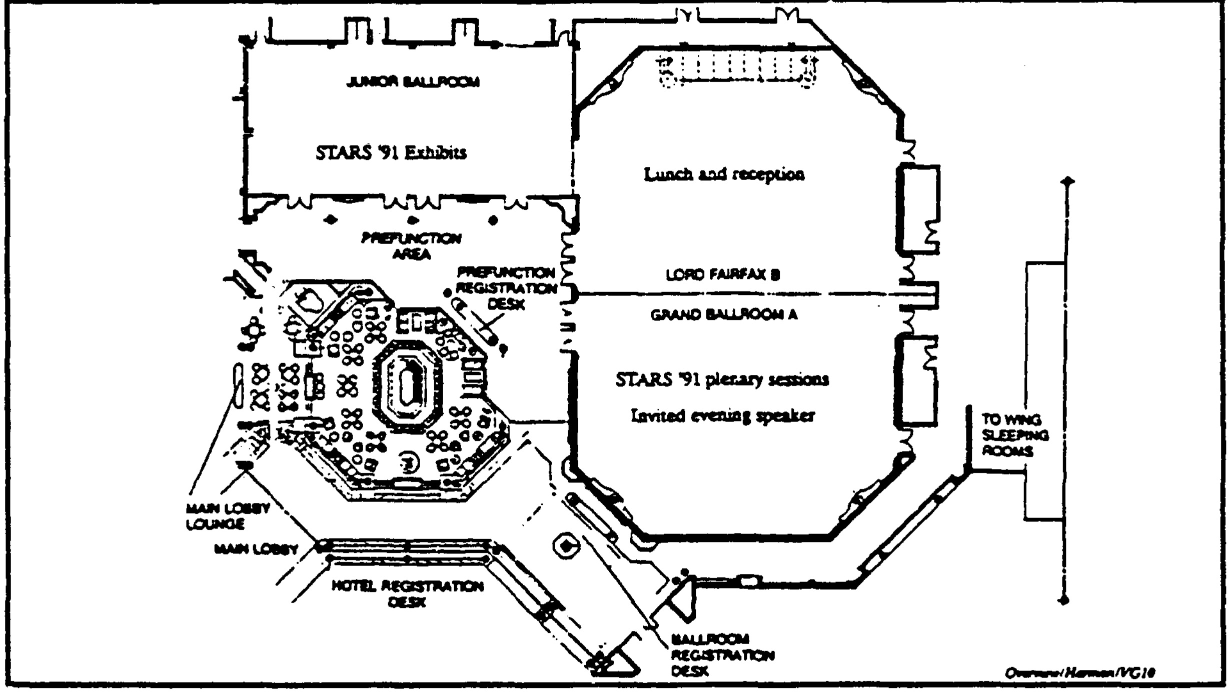
Time	Event			
1:45pm-2:30pm	Track 1 feedback session	Track 2 feedback session	Track 3 feedback session	Track 4 feedback session
2:30pm-3:00pm	Break			
3:00pm-3:40pm	DoD Software Technology Plan and STARS		Barry Boehm (DARPA)	
3:40pm-4:00pm	Closing remarks		John Foreman (DARPA)	

Overview/Harmon/VG9

After lunch, there will be a final Track Feedback session conducted by the STARS Program Managers. This is a general feedback session so attend the one for the Track you have been most involved with. And then starting at 3 pm, a final plenary session will be held. Be sure to stay. Dr Boehm will be talking about the DoD Software Technology Plan and STARS, and John Foreman will summarize significant issues raised during the Conference and point out where we need to go from here.

STARS '91 OVERVIEW

LOCATION OF EVENTS



STARS '91 Exhibits

Lunch and reception

STARS '91 plenary sessions

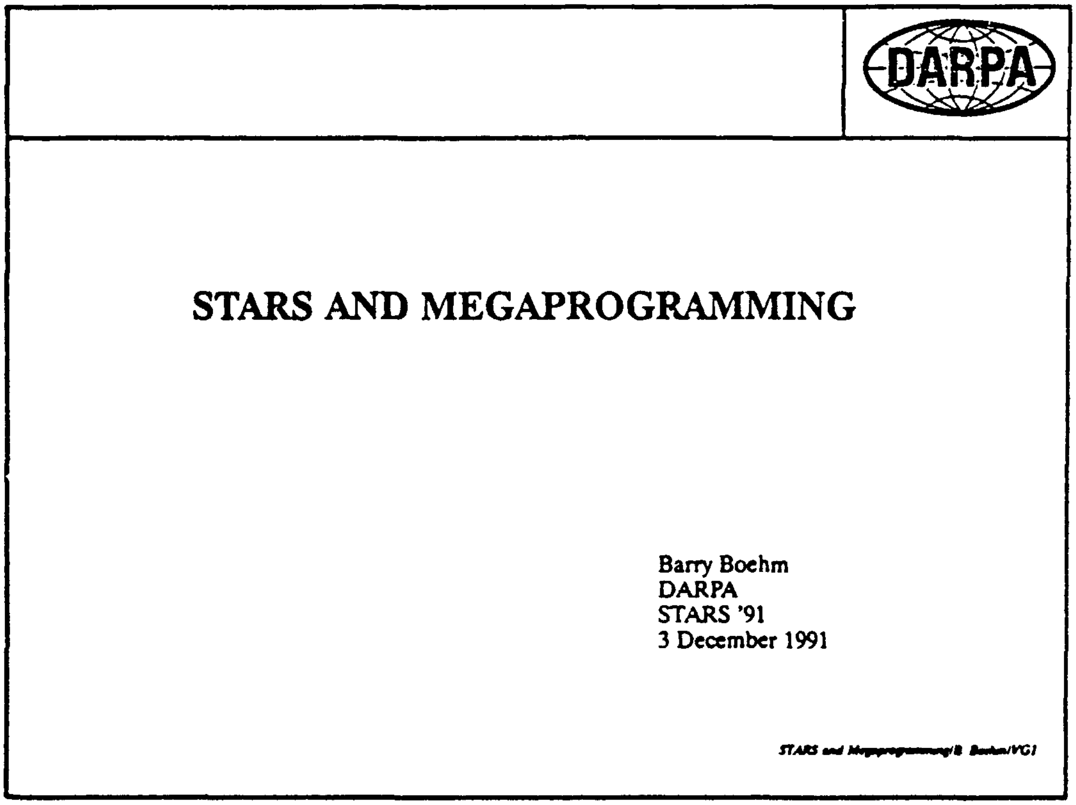
Invited evening speaker

Overview/Harmon/VG10

My final vugraph shows the location of the major meeting areas. To help you find the Track rooms, signs are located right outside this meeting area.

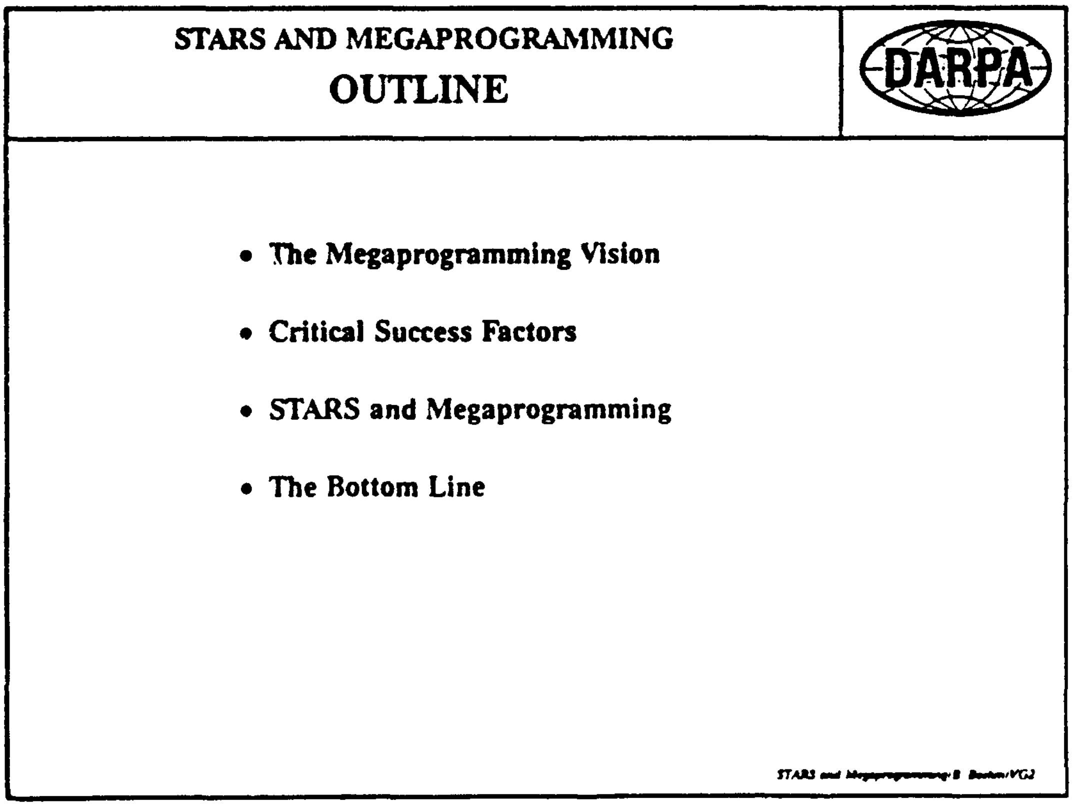
Now to begin our Program, I would like to introduce you to Dr. Barry Boehm who is the Director of DARPA/SISTO which is the U.S. government's largest computer-communication's research organization. He will speak to us on "STARS and Megaprogramming".

Dr. Boehm –



STARS AND MEGAPROGRAMMING

Barry Boehm
DARPA
STARS '91
3 December 1991

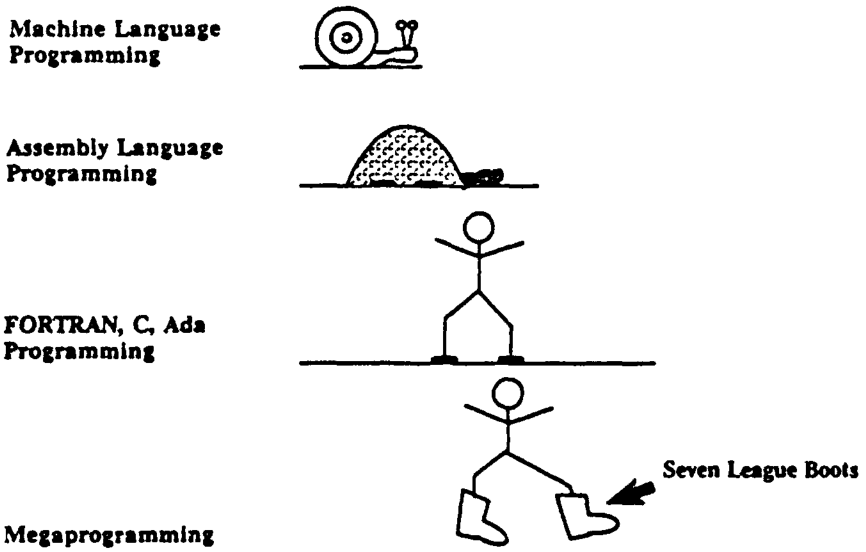


STARS AND MEGAPROGRAMMING
OUTLINE

- The Megaprogramming Vision
- Critical Success Factors
- STARS and Megaprogramming
- The Bottom Line

STARS AND MEGAPROGRAMMING

COMPOSING SW COMPONENTS RATHER THAN LINES OF CODE



Machine Language Programming

Assembly Language Programming

FORTRAN, C, Ada Programming

Megaprogramming

Seven League Boots

STARS and Megaprogramming/B. Boehm/VG3

STARS AND MEGAPROGRAMMING

MEGAPROGRAMMING BENEFITS

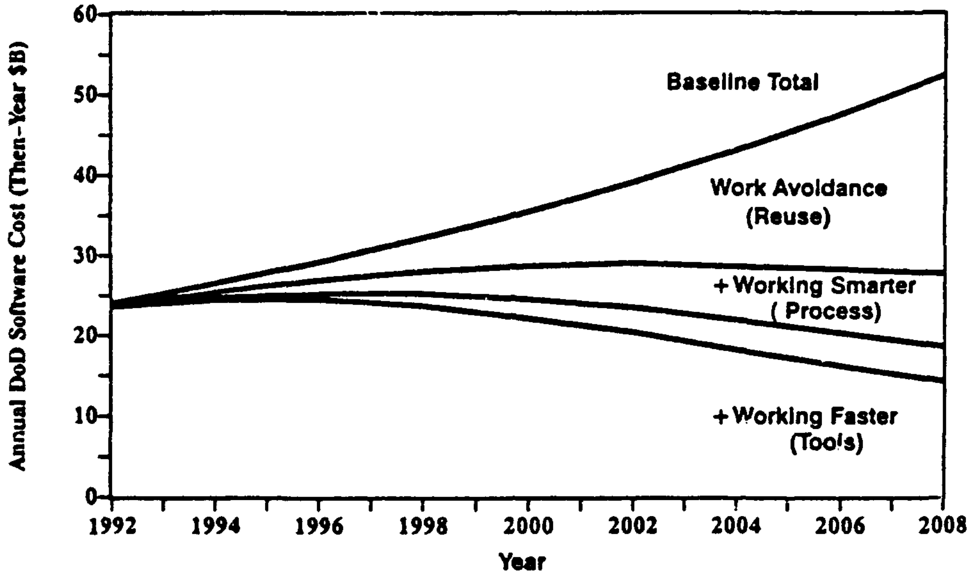
- Development productivity
  - Fewer components to assemble, integrate, test
  - Smaller teams; less overhead
  - Opportunities for application generators
- Maintenance productivity
  - Update (40%): open interfaces; alternative components
  - Adaptive (25%): open interfaces to infrastructure software and hardware
  - Corrective (25%): fewer residual errors; effects encapsulated
- Reliability, Availability, Security
  - Proven vs unproven components
- Portability, Interoperability
  - Well-defined open interfaces
- Operational Capability
  - Darwinian evolution of best components
  - Accommodates hardware technology improvements

STARS and Megaprogramming/B. Boehm/VG4

STARS AND MEGAPROGRAMMING

PROGRAM RESULTS BY SOURCE OF SAVINGS

DARPA



STARS and Megaprogramming/B. Boehm/VG5

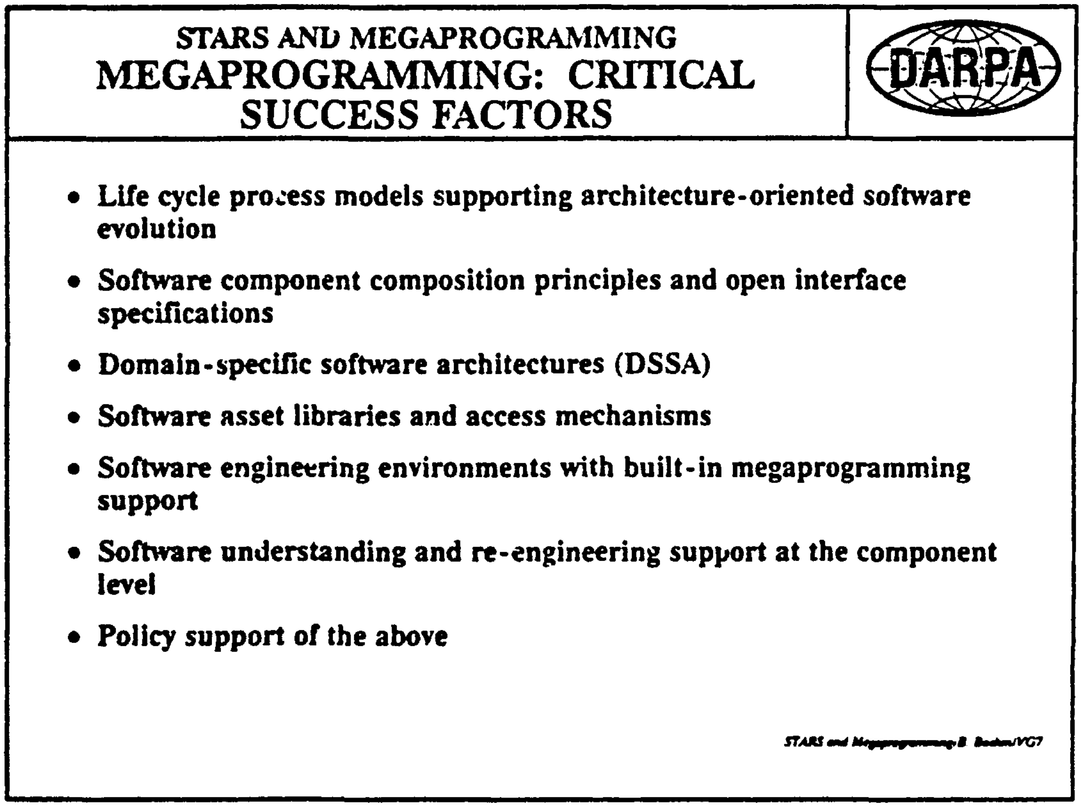
STARS AND MEGAPROGRAMMING

MEGAPROGRAMMING PARADIGM SHIFTS

DARPA

- Not programming as taught, practiced today
  - Requires higher level of thinking, engineering
- Evolution and opportunity-oriented rather than "requirements snapshot" oriented
  - Supports continuous system-level improvement
- Requires domain expertise, assets as well as programming expertise, assets
- Requires open-systems approach
- Evolutionary progress feasible
  - Doesn't depend on unpredictable breakthroughs

STARS and Megaprogramming/B. Boehm/VG6

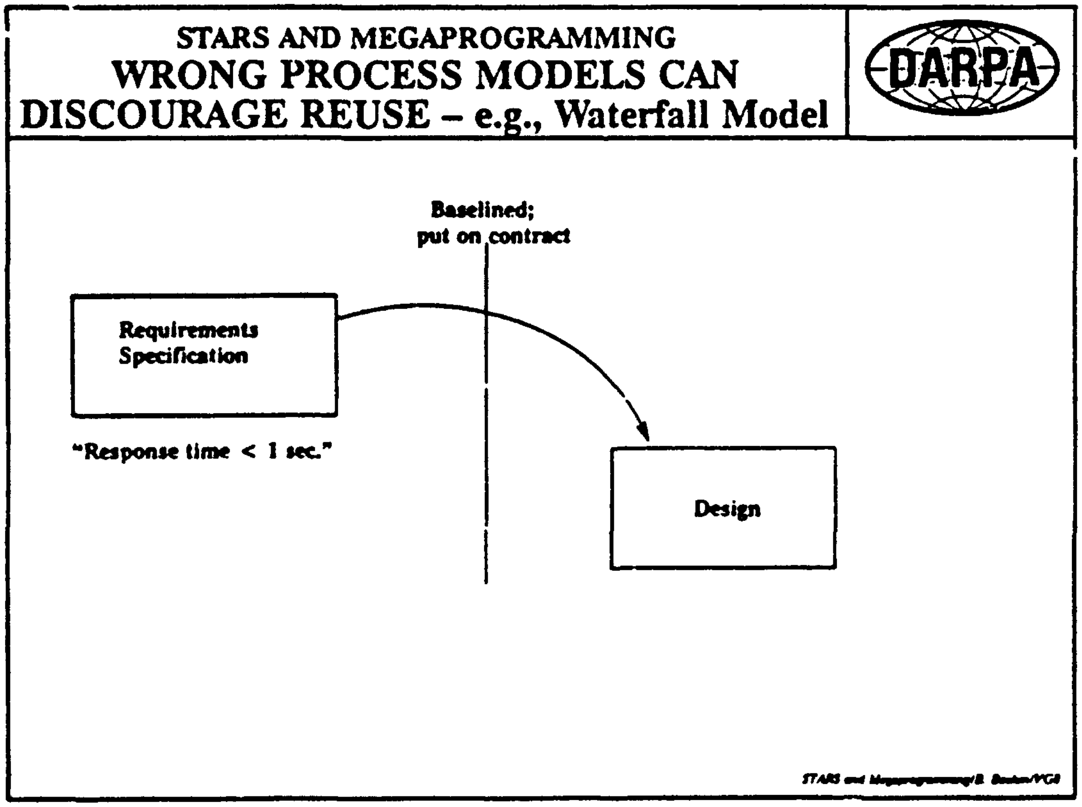


STARS AND MEGAPROGRAMMING

MEGAPROGRAMMING: CRITICAL SUCCESS FACTORS

- Life cycle pro:ess models supporting architecture-oriented software evolution
- Software component composition principles and open interface specifications
- Domain-specific software architectures (DSSA)
- Software asset libraries and access mechanisms
- Software engineering environments with built-in megaprogramming support
- Software understanding and re-engineering support at the component level
- Policy support of the above

STARS and Megaprogramming/B. Boehm/VG7



STARS AND MEGAPROGRAMMING

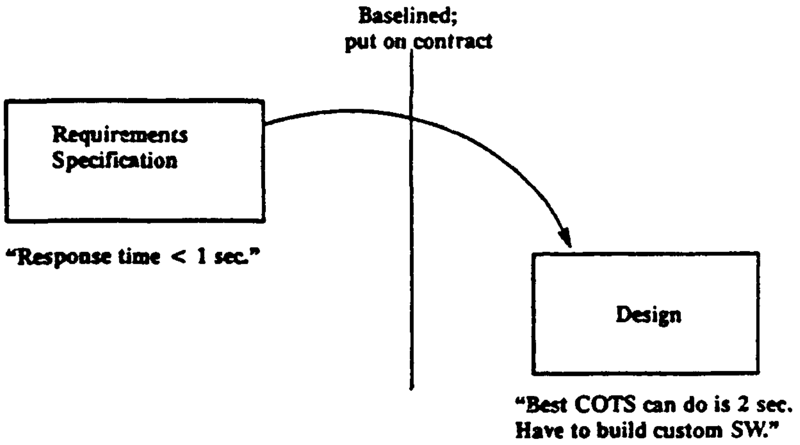
WRONG PROCESS MODELS CAN DISCOURAGE REUSE – e.g., Waterfall Model

STARS and Megaprogramming/B. Boehm/VG8

STARS AND MEGAPROGRAMMING

WRONG PROCESS MODELS CAN DISCOURAGE REUSE – e.g., Waterfall Model

DARPA


Baselined;
put on contract
Requirements Specification
"Response time < 1 sec."
Design
"Best COTS can do is 2 sec.
Have to build custom SW."

Tools and environments built to waterfall model will also discourage reuse.

STARS and Megaprogramming/B. Boehm/VG9

STARS AND MEGAPROGRAMMING

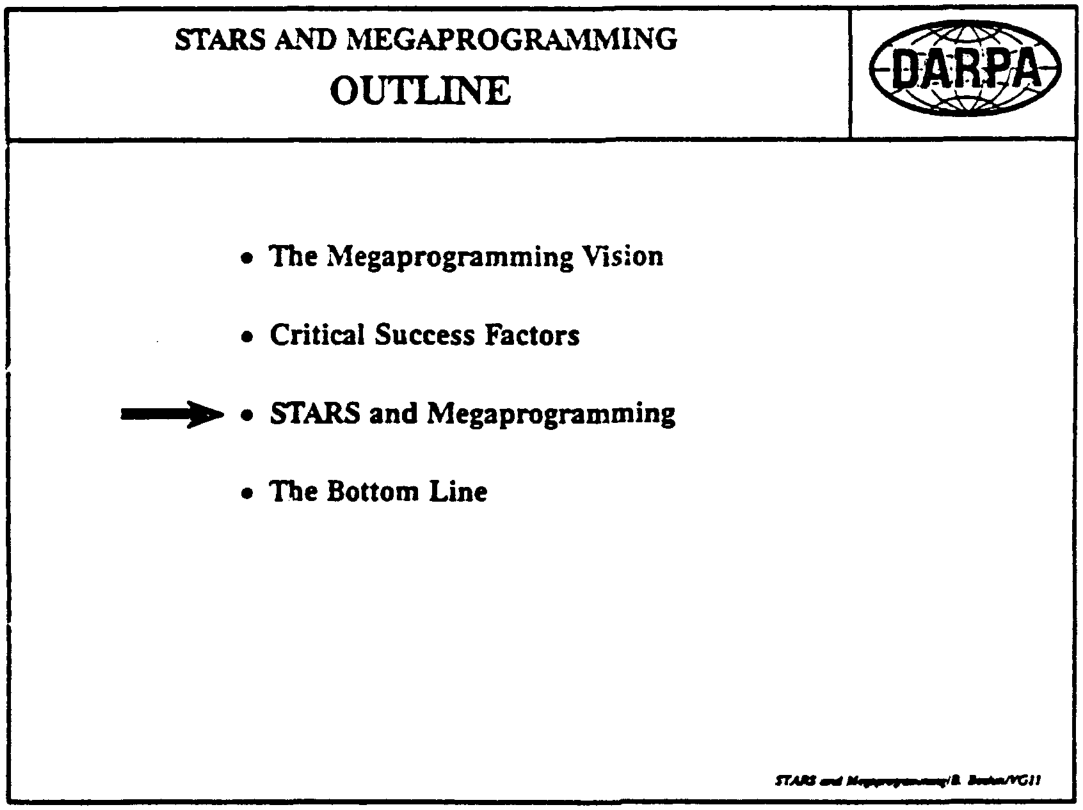
DSSA PROGRAM SUMMARY CHART

DARPA

Domain Specific Software Architecture

DOMAIN	CONTRACTORS	MILITARY
Avionics	IBM/UCI/UTex	Wright Labs
Command and Control	GTE Contel/USC-ISI GMU	CECOM
Vehicle Management Systems	TFS/Stanford	ARDEC
Guidance, Navigation & Control	Honeywell/UMd	ONR/NAWC
Distributed Intelligent Control	ORA/Cornell MSI	ARDEC
Prototyping Technology Insertion	TRW/Stanford	ONR

STARS and Megaprogramming/B. Boehm/VG10



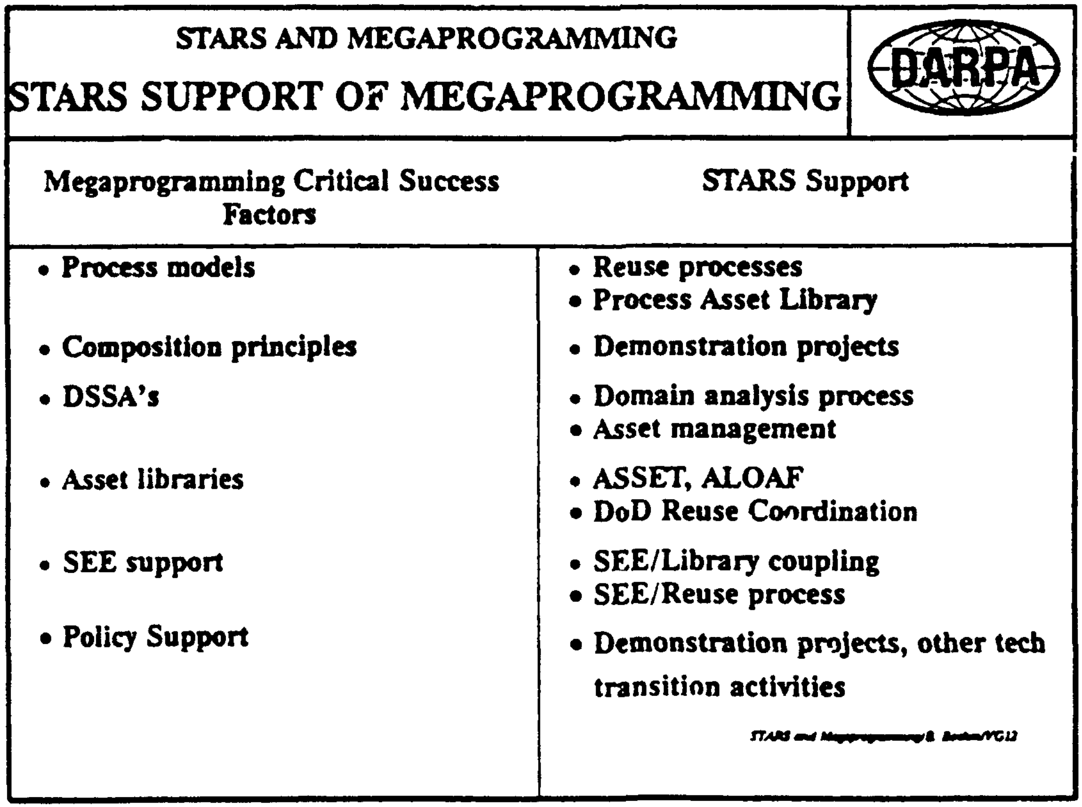
STARS AND MEGAPROGRAMMING

OUTLINE

DARPA

- The Megaprogramming Vision
- Critical Success Factors
- → STARS and Megaprogramming
- The Bottom Line

STARS and Megaprogramming/B. Boehm/VG11



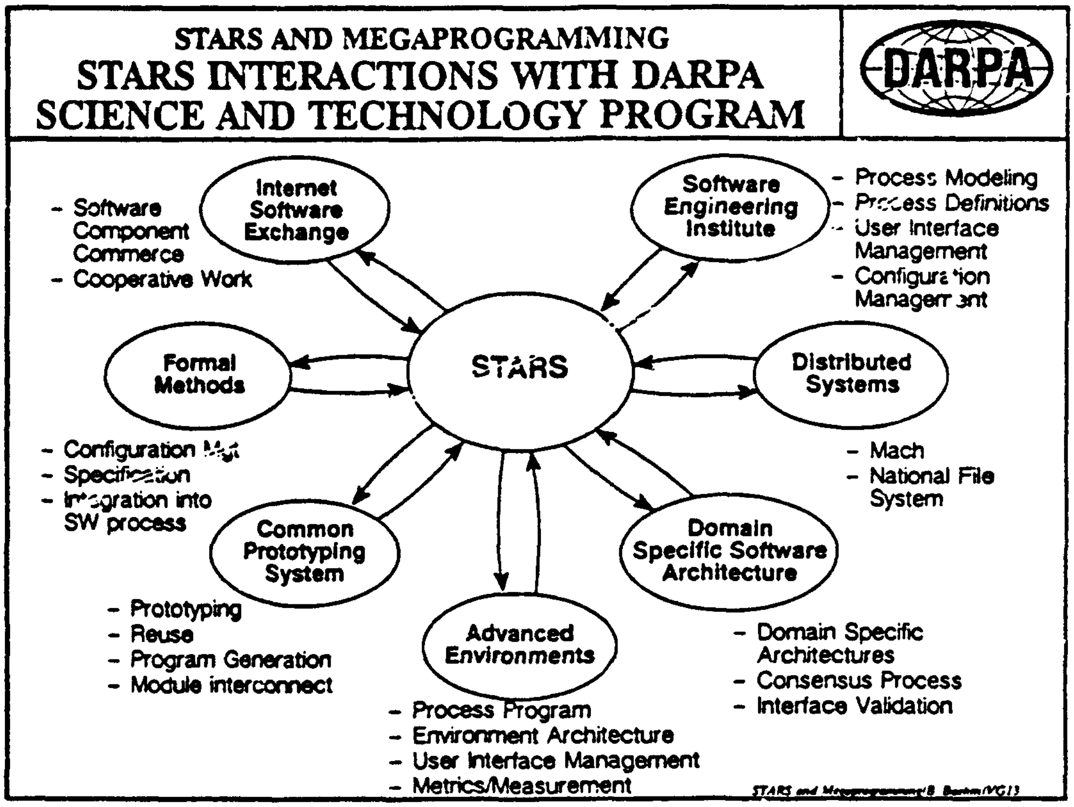
STARS AND MEGAPROGRAMMING

STARS SUPPORT OF MEGAPROGRAMMING

DARPA

Megaprogramming Critical Success Factors	STARS Support
• Process models	• Reuse processes • Process Asset Library
• Composition principles	• Demonstration projects
• DSSA's	• Domain analysis process • Asset management
• Asset libraries	• ASSET, ALOAF • DoD Reuse Coordination
• SEE support	• SEE/Library coupling • SEE/Reuse process
• Policy Support	• Demonstration projects, other tech transition activities

STARS and Megaprogramming/B. Boehm/VG12



STARS AND MEGAPROGRAMMING

STARS INTERACTIONS WITH DARPA SCIENCE AND TECHNOLOGY PROGRAM

DARPA
STARS
Internet Software Exchange
Software Engineering Institute
Formal Methods
Distributed Systems
Common Prototyping System
Advanced Environments
Domain Specific Software Architecture

- Software Component Commerce
- Cooperative Work

- Process Modeling
- Prc:ess Definitions
- User Interface Management
- Configure 'ion Managerr ənt

- Configuration Mgt
- Specific:tion
- In'egration into SW process

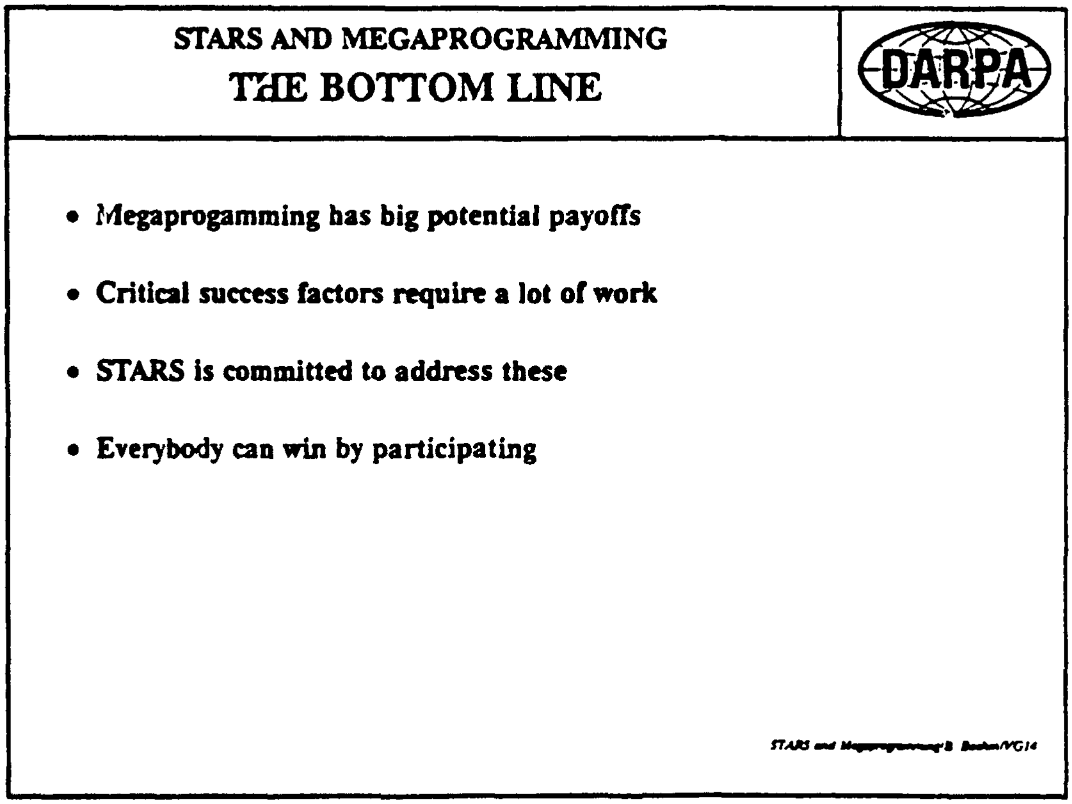
- Mach
- National File System

- Prototyping
- Reuse
- Program Generation
- Module interconnect

- Process Program
- Environment Architecture
- User Interface Management
- Metrics/Measurement

- Domain Specific Architectures
- Consensus Process
- Interface Validation

STARS and Megaprogramming/B Boehm/VG13



STARS AND MEGAPROGRAMMING

THE BOTTOM LINE

- Megaprogamming has big potential payoffs
- Critical success factors require a lot of work
- STARS is committed to address these
- Everybody can win by participating

STARS and Megaprogramming/B Boehm/VG14

STARS
VISION, MISSION, STRATEGY AND ACHIEVEMENTS

John Foreman
STARS Program Manager
3 December 1991
(703) 243–8655
jtf@sei.cmu.edu

VISION, MISSION, STRATEGY, ACHIEVEMENTS
OUTLINE

→ • **Introduce STARS**
- **Vision, Mission, Strategy**
- **Objectives/Approach**
- **Achievements**

VISION, MISSION, STRATEGY, ACHIEVEMENTS

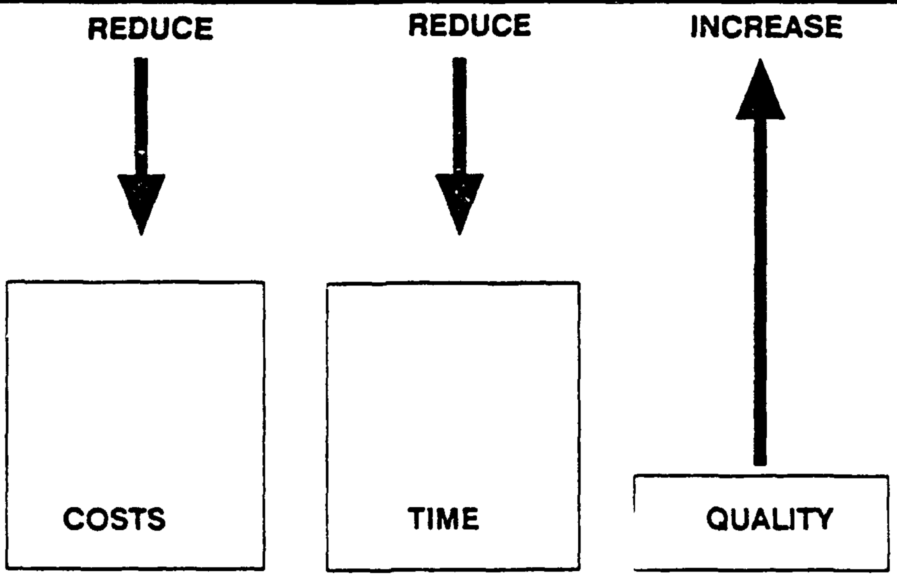
WHY STARS?

- Facilitate fundamental changes in DoD software development process/technologies
- Address issues beyond the context of any particular development program--neutral ground to develop new ways of doing business
- Influence commercial industry to accommodate DoD needs/directions
- Facilitate enabling technologies for market in DoD specific reusable components/architectures
- Address DoD software problem sooner
  - DoD need for application software exceeds available funding
  - Current software development paradigm inefficient

Vision, Mission, Strategy and Achievements/J. Foreman/VG3

VISION, MISSION, STRATEGY, ACHIEVEMENTS

STARS OVERALL GOALS

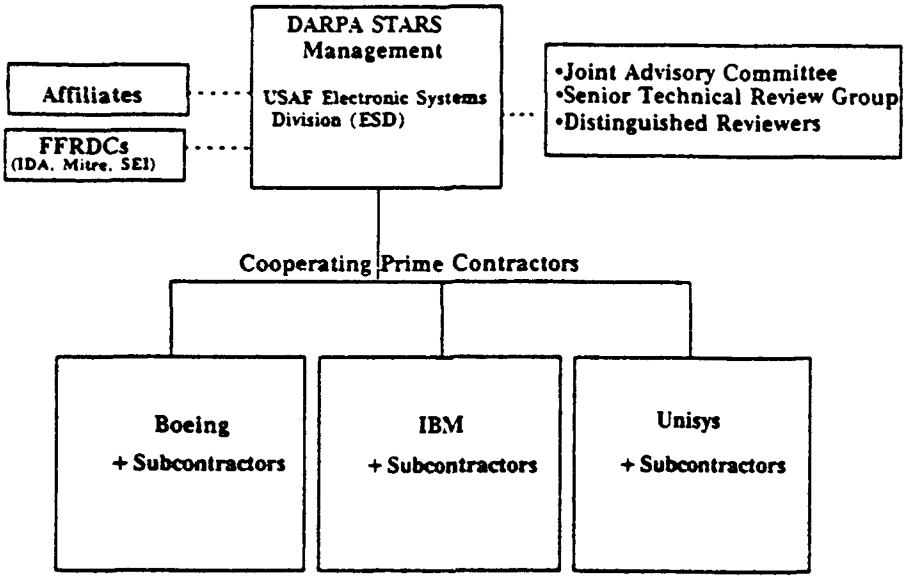


Vision, Mission, Strategy and Achievements/J. Foreman/VG4

VISION, MISSION, STRATEGY, ACHIEVEMENTS

STARS PROGRAM ORGANIZATION

DARPA



DARPA STARS Management

USAF Electronic Systems Division (ESD)

Affiliates

FFRDCs (IDA, Mitre, SEI)

•Joint Advisory Committee
•Senior Technical Review Group
•Distinguished Reviewers

Cooperating Prime Contractors

Boeing + Subcontractors

IBM + Subcontractors

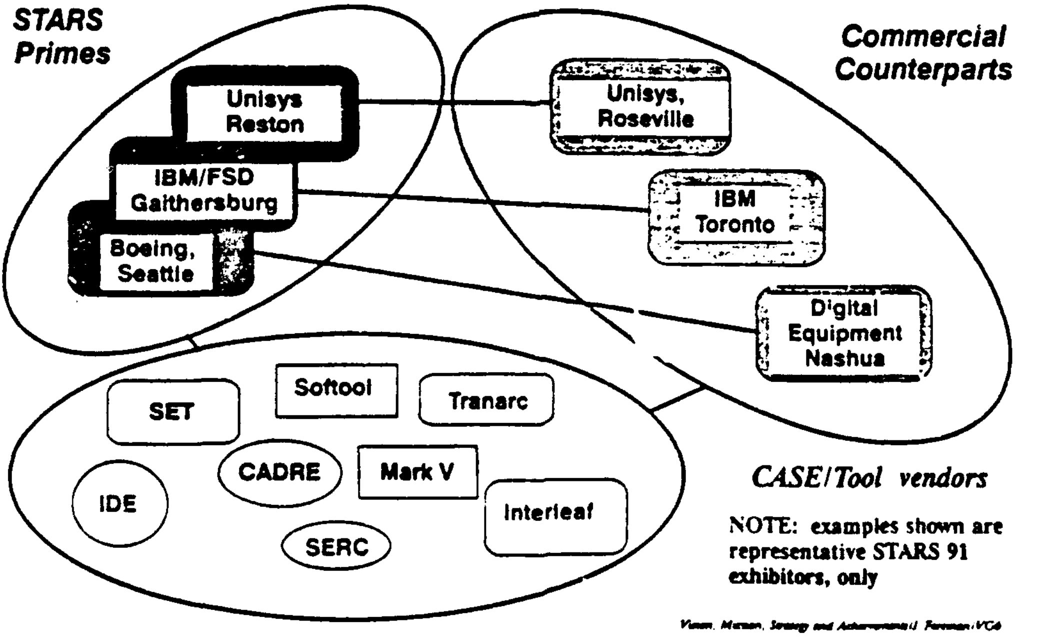
Unisys + Subcontractors

Vision, Mission, Strategy and Achievements/L Foreman/VC5

VISION, MISSION, STRATEGY, ACHIEVEMENTS

STARS ROLES AND RELATIONSHIPS

DARPA



STARS Primes

Unisys Reston

IBM/FSD Gaithersburg

Boeing, Seattle

Commercial Counterparts

Unisys, Roseville

IBM Toronto

Digital Equipment Nashua

SET

Softool

Tranarc

IDE

CADRE

Mark V

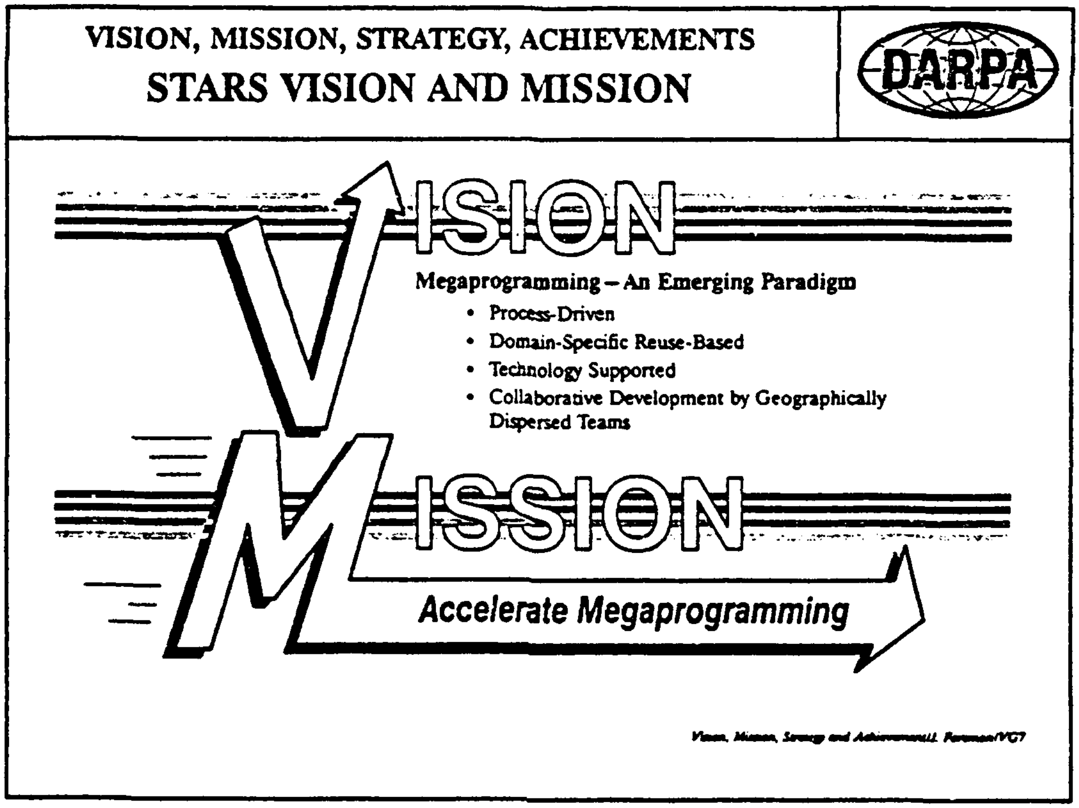
Interleaf

SERC

CASE/Tool vendors

NOTE: examples shown are representative STARS 91 exhibitors, only

Vision, Mission, Strategy and Achievements/L Foreman/VC6



VISION, MISSION, STRATEGY, ACHIEVEMENTS
STARS VISION AND MISSION

DARPA

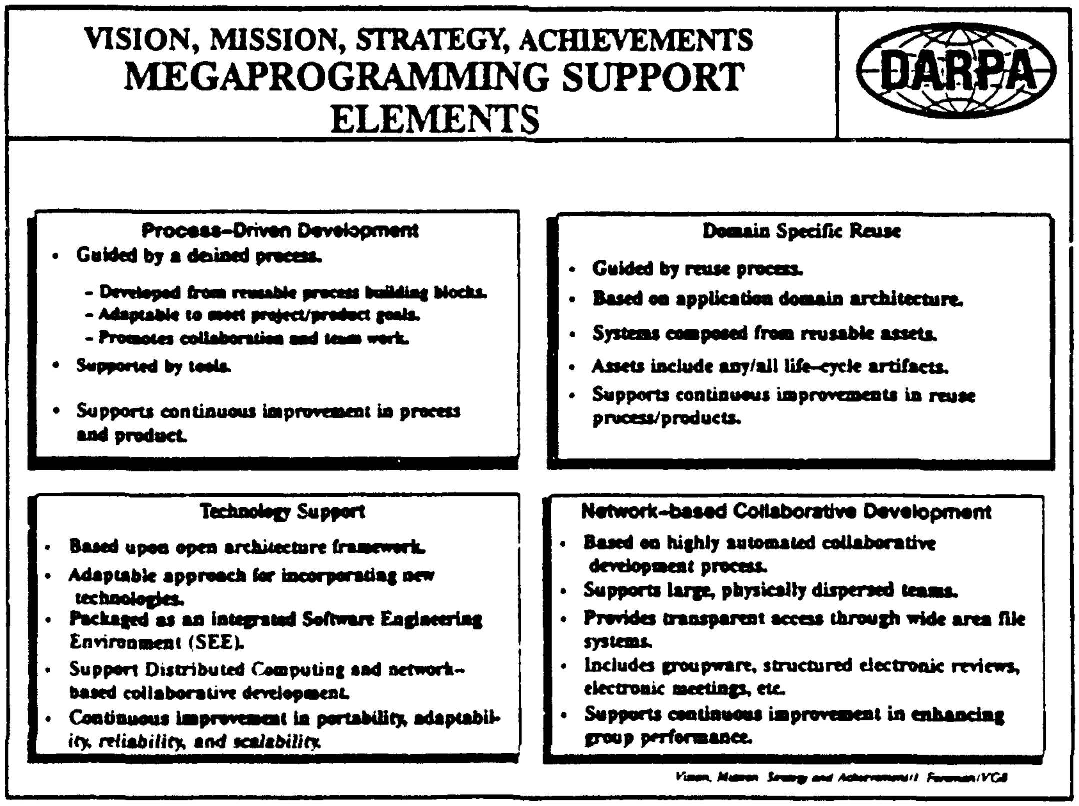
VISION
MISSION

Megaprogramming – An Emerging Paradigm

- Process-Driven
- Domain-Specific Reuse-Based
- Technology Supported
- Collaborative Development by Geographically Dispersed Teams

Accelerate Megaprogramming

Vision, Mission, Strategy and AchievementsLL Foreman/VG7



VISION, MISSION, STRATEGY, ACHIEVEMENTS
MEGAPROGRAMMING SUPPORT ELEMENTS

DARPA

Process–Driven Development

- **Guided by a defined process.**
  - **Developed from reusable process building blocks.**
  - **Adaptable to meet project/product goals.**
  - **Promotes collaboration and team work.**
- **Supported by tools.**
- **Supports continuous improvement in process and product.**

Domain Specific Reuse

- **Guided by reuse process.**
- **Based on application domain architecture.**
- **Systems composed from reusable assets.**
- **Assets include any/all life–cycle artifacts.**
- **Supports continuous improvements in reuse process/products.**

Technology Support

- **Based upon open architecture framework.**
- **Adaptable approach for incorporating new technologies.**
- **Packaged as an integrated Software Engineering Environment (SEE).**
- **Support Distributed Computing and network–based collaborative development.**
- **Continuous improvement in portability, adaptability, *reliability, and scalability.***

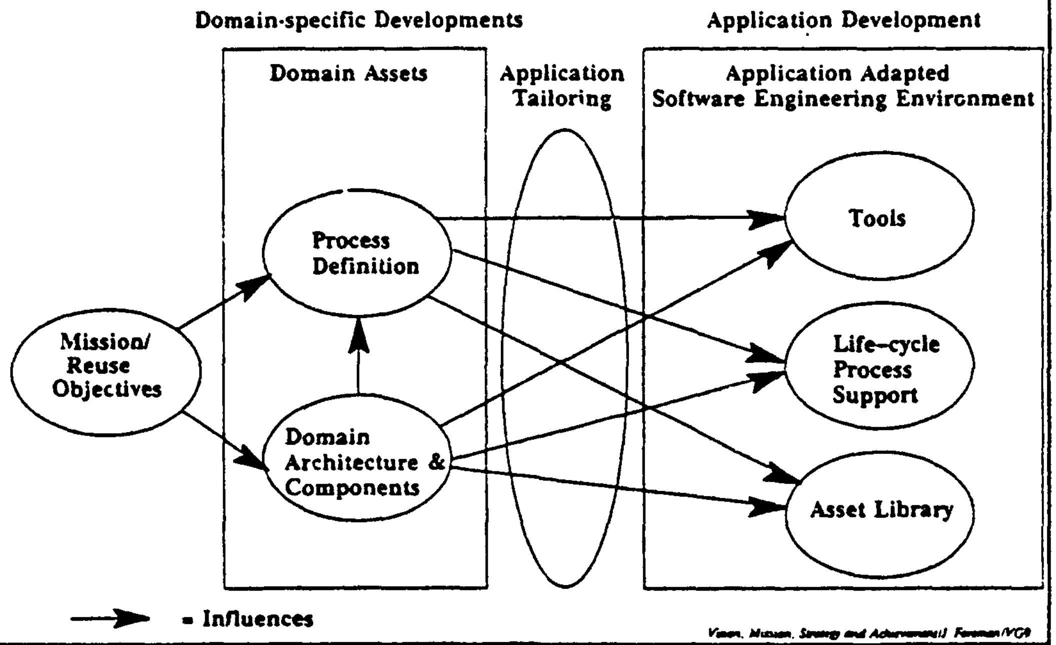
Network–based Collaborative Development

- **Based on highly automated collaborative development process.**
- **Supports large, physically dispersed teams.**
- **Provides transparent access through wide area file systems.**
- **Includes groupware, structured electronic reviews, electronic meetings, etc.**
- **Supports continuous improvement in enhancing group performance.**

Vision, Mission Strategy and Achievements// Foreman/VG8

VISION, MISSION, STRATEGY, ACHIEVEMENTS

MEGAPROGRAMMING CONCEPT



Domain-specific Developments

Application Development

Domain Assets

Application Tailoring

Application Adapted Software Engineering Environment

Mission/ Reuse Objectives

Process Definition

Domain Architecture & Components

Tools

Life-cycle Process Support

Asset Library

⟶ = Influences

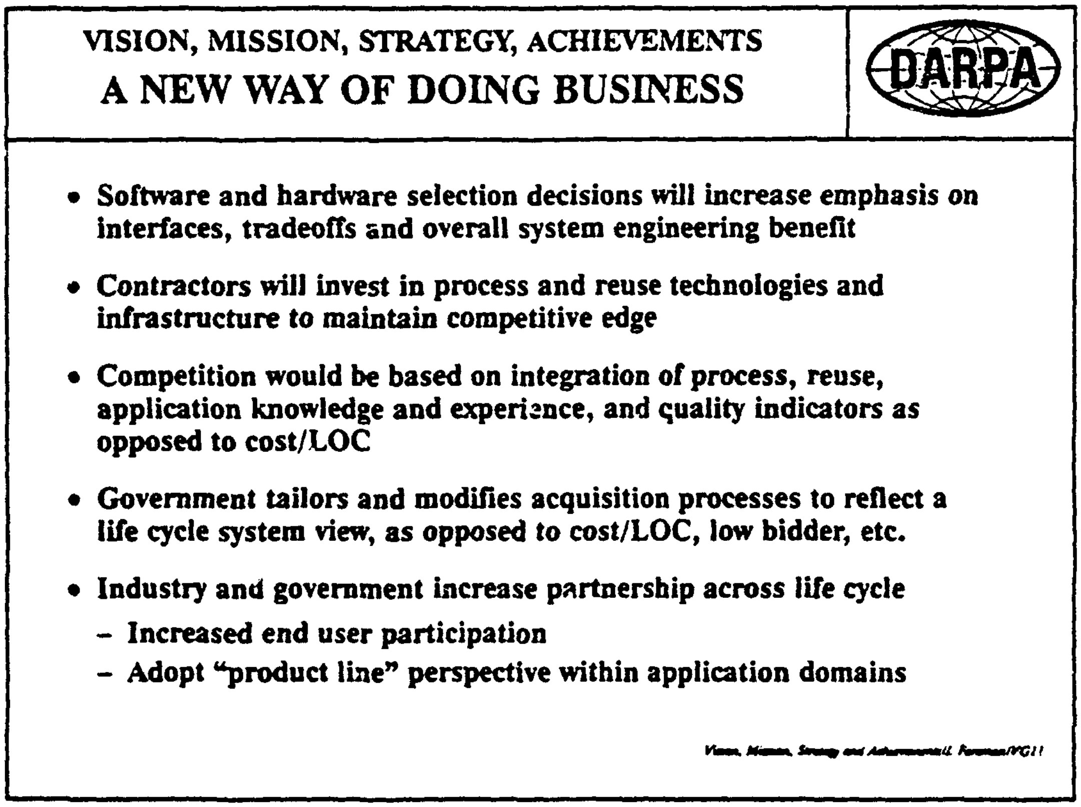
Vision, Mission, Strategy and Achievements// Foreman/VG9

VISION, MISSION, STRATEGY, ACHIEVEMENTS

WHAT PROBLEMS ARE WE ADDRESSING?

Current Problems	Megaprogramming Solution
Lack of common understanding of requirements between end-user and developer	Architecture-based rapid prototyping with end-user involvement
Difficulty in understanding and maintaining software developed by others	Well-defined architecture context, component interfaces & localization of behavior
Difficulties in scaling up to large developments by many people with diversified skills	Megaprogramming processes and technologies supporting architecture based reuse and collaborative development
New systems often treated as unprecedented	Building unprecedented from precedented components
Chaotic and inefficient development process	Defined processes, with continuous improvement

Vision, Mission, Strategy and Achievements// Foreman/VG10



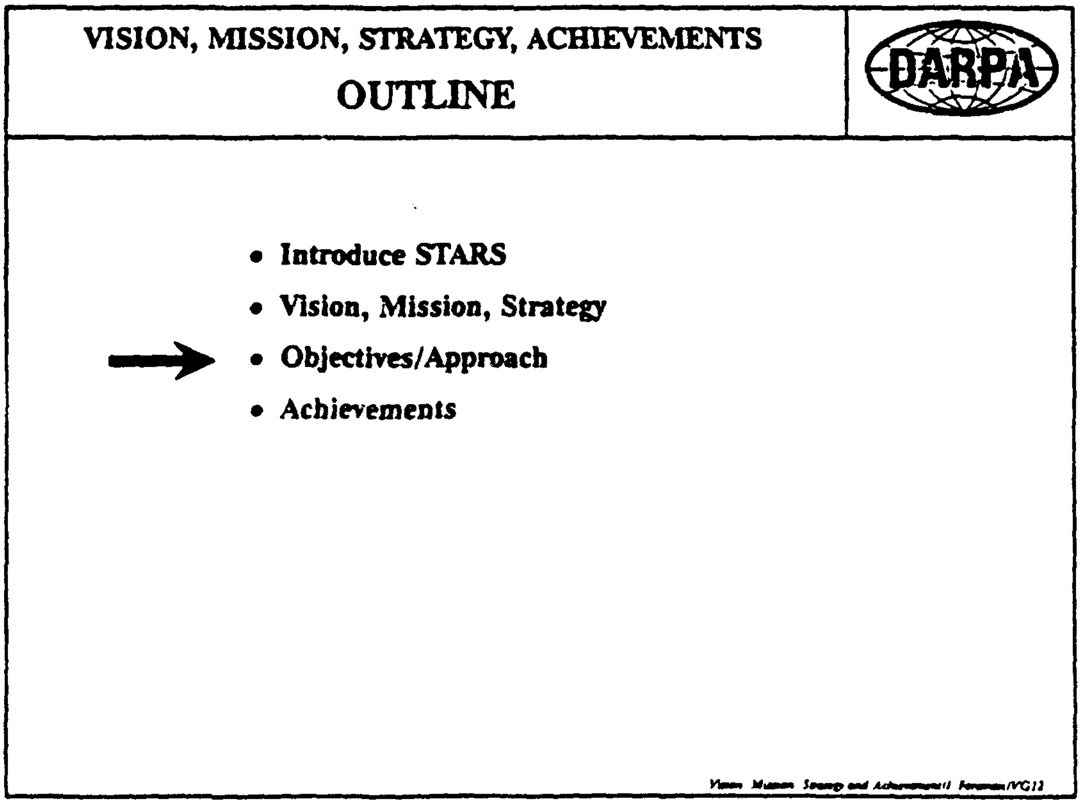
VISION, MISSION, STRATEGY, ACHIEVEMENTS

A NEW WAY OF DOING BUSINESS

DARPA

- Software and hardware selection decisions will increase emphasis on interfaces, tradeoffs and overall system engineering benefit
- Contractors will invest in process and reuse technologies and infrastructure to maintain competitive edge
- Competition would be based on integration of process, reuse, application knowledge and experience, and quality indicators as opposed to cost/LOC
- Government tailors and modifies acquisition processes to reflect a life cycle system view, as opposed to cost/LOC, low bidder, etc.
- Industry and government increase partnership across life cycle
  - Increased end user participation
  - Adopt "product line" perspective within application domains

Vision, Mission, Strategy and Achievements/L Foreman/VG11



VISION, MISSION, STRATEGY, ACHIEVEMENTS

OUTLINE

DARPA

- Introduce STARS
- Vision, Mission, Strategy
- → Objectives/Approach
- Achievements

Vision Mission Strategy and Achievements/L Foreman/VG12

VISION, MISSION, STRATEGY, ACHIEVEMENTS

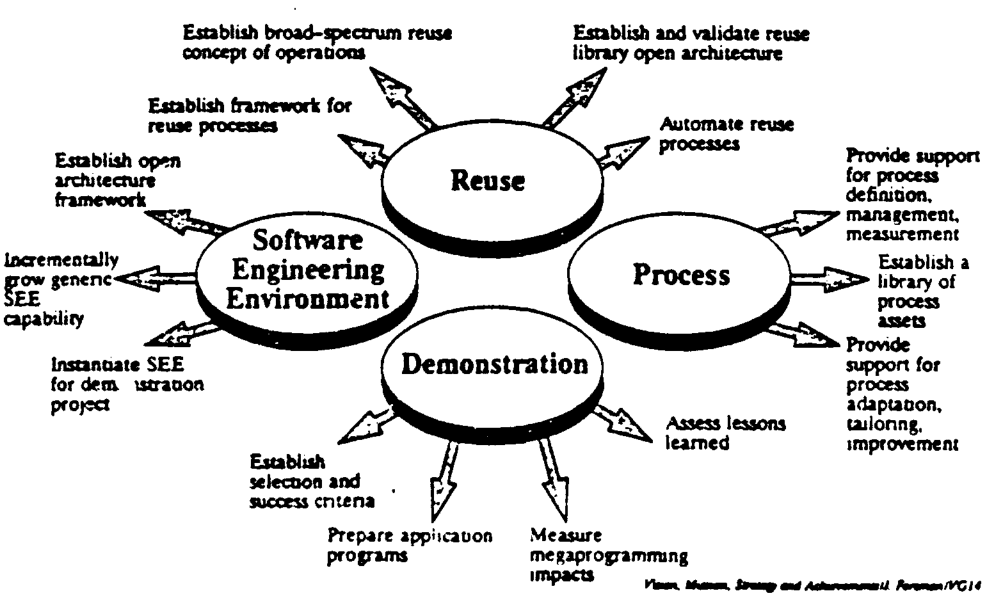
TOP LEVEL PROGRAM OBJECTIVES

- Demonstrate the benefits of megaprogramming in a familiar context [create motivation]
- Reduce megaprogramming adoption risks by providing transition guidance
- Develop and accelerate the availability of processes and technologies to support the use and continued evolution of megaprogramming

Vision, Mission, Strategy and Achievements/L Foreman/VG13

VISION, MISSION, STRATEGY, ACHIEVEMENTS

STARS ACTIVITIES

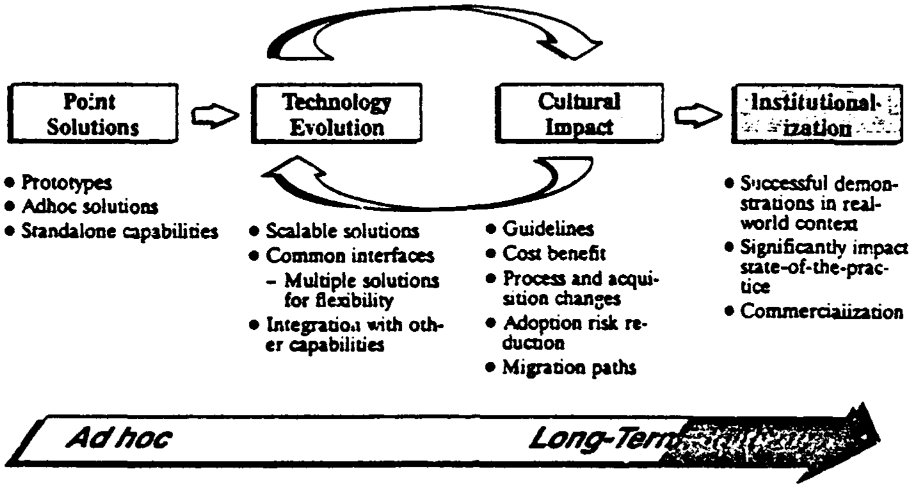


Reuse
Establish broad-spectrum reuse concept of operations
Establish and validate reuse library open architecture
Establish framework for reuse processes
Automate reuse processes
Software Engineering Environment
Establish open architecture framework
Incrementally grow generic SEE capability
Instantiate SEE for dem istration project
Process
Provide support for process definition, management, measurement
Establish a library of process assets
Provide support for process adaptation, tailoring, improvement
Demonstration
Establish selection and success criteria
Prepare application programs
Measure megaprogramming impacts
Assess lessons learned

Vision, Mission, Strategy and Achievements/L Foreman/VG14

VISION, MISSION, STRATEGY, ACHIEVEMENTS

STARS APPROACH



Point Solutions

- Prototypes
- Adhoc solutions
- Standalone capabilities

Technology Evolution

- Scalable solutions
- Common interfaces
  - Multiple solutions for flexibility
- Integration with other capabilities

Cultural Impact

- Guidelines
- Cost benefit
- Process and acquisition changes
- Adoption risk reduction
- Migration paths

Institutionalization

- Successful demonstrations in real-world context
- Significantly impact state-of-the-practice
- Commercialization

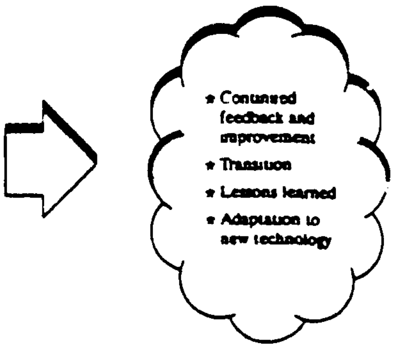
Vision, Mission, Strategy and Achievements/J. Foreman/VG13

VISION, MISSION, STRATEGY, ACHIEVEMENTS

OVERALL PROGRAM TIMELINE

Phase Activity	Architecture Definition and Risk Reduction 90-91	SEE Instantiation 92-93	Evaluation Maturation 94-96
SEE, reuse, process technology integration	Prototyping Early integration	Integration Tailoring for evaluation (process, reuse) Evaluation phase support plan	Improvement/feedback Support to application development
Demonstration and assessment	Criteria for evaluating STARS system Criteria for selecting application development projects	Continue system criteria development Select and prepare application program (SPO and contractor)	Monitor application development Obtain and assess lessons learned
External activities	Initiate reusable asset and architecture development Software process definition	Continue asset development Initial asset evaluation Software process definition	Application development

DARPA Activities After STARS



- ★ Continued feedback and improvement
- ★ Transition
- ★ Lessons learned
- ★ Adaptation to new technology

Vision, Mission, Strategy and Achievements/J. Foreman/VG14

VISION, MISSION, STRATEGY, ACHIEVEMENTS

OUTLINE

DARPA

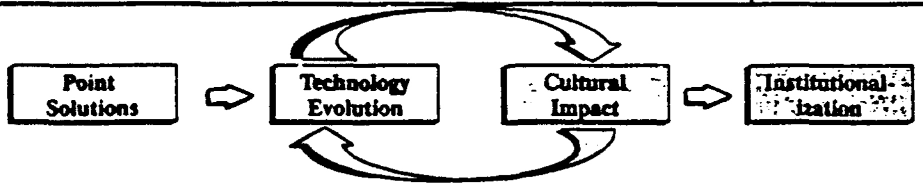
- Introduce STARS
- Vision, Mission, Strategy
- Objectives/Approach
- ➔ Achievements

Vision, Mission, Strategy and Achievements/J. Foreman/VG17

VISION, MISSION, STRATEGY, ACHIEVEMENTS

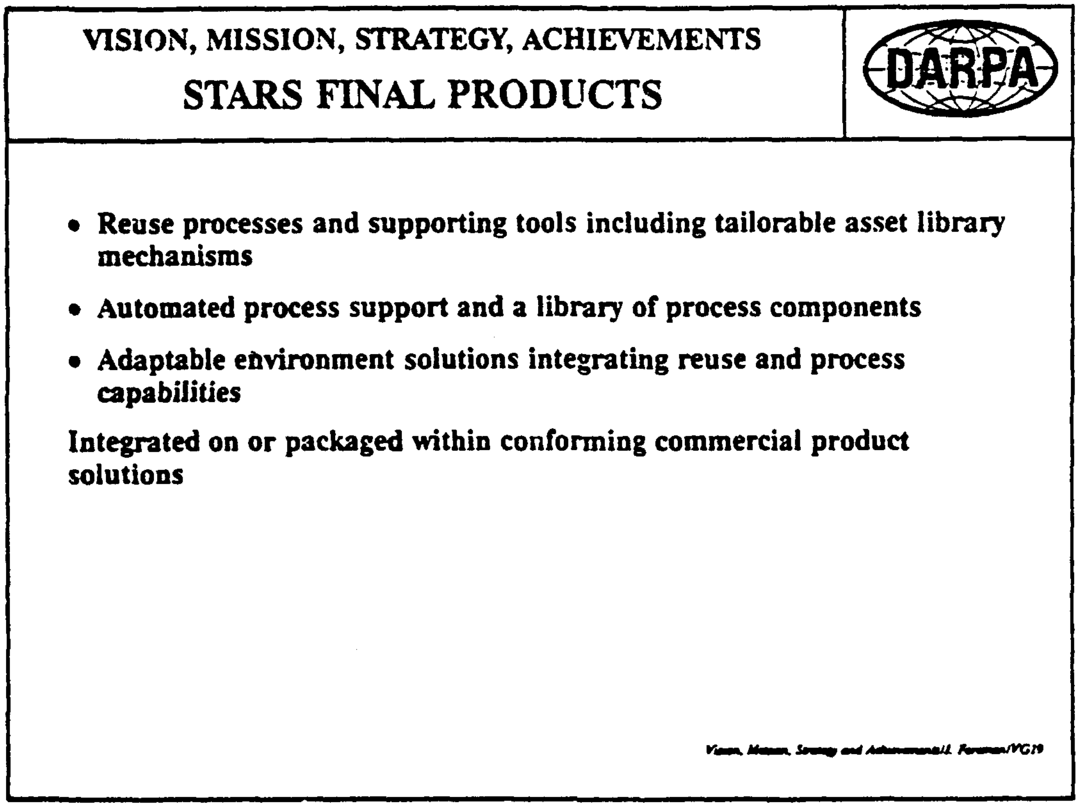
KEY ACHIEVEMENTS

DARPA



	Point Solutions	Technology Evolution	Cultural Impact	Institutionalization
REUSE	• Prototype reuse libraries	• 2nd generation reuse libraries • Asset library open arch • RIG for reuse library interoperability	• Reuse concept of operations • Trial usage: AF CARDS & Navy NTSC	
PROCESS	• Prototype process management capabilities	• Process definitions (e.g., domain analysis, cleanroom, software first)	• Process concept summary	
TECHNOLOGY SUPPORT	• Ada/X bindings • Universal Ada Test Language (UATL) • 3 SEE Testbeds • STARS Foundations	• Initial standards profile • SEE Framework convergence meetings		
GLOBAL			• *Impact on AF PRISM Program*	• Relationships w/commercial counterparts

Vision, Mission, Strategy and Achievements/J. Foreman/VG18



VISION, MISSION, STRATEGY, ACHIEVEMENTS

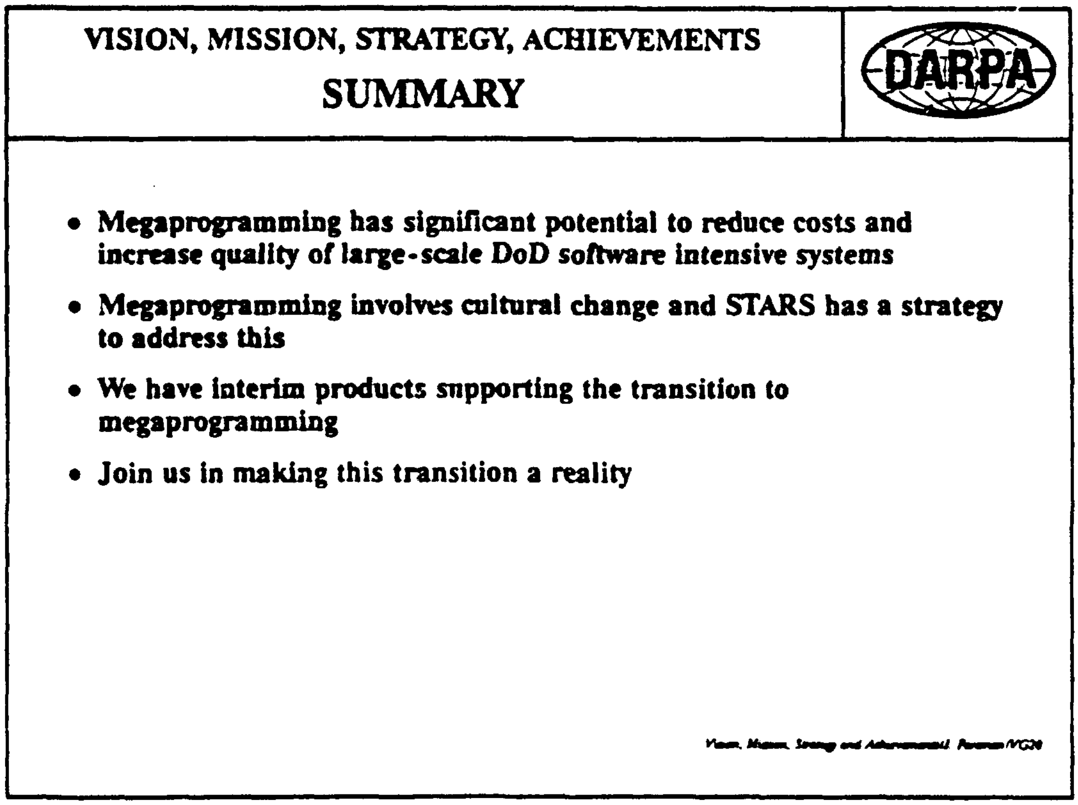
STARS FINAL PRODUCTS

DARPA

- Reuse processes and supporting tools including tailorable asset library mechanisms
- Automated process support and a library of process components
- Adaptable environment solutions integrating reuse and process capabilities

Integrated on or packaged within conforming commercial product solutions

Vision, Mission, Strategy and Achievements/J. Foreman/VG19



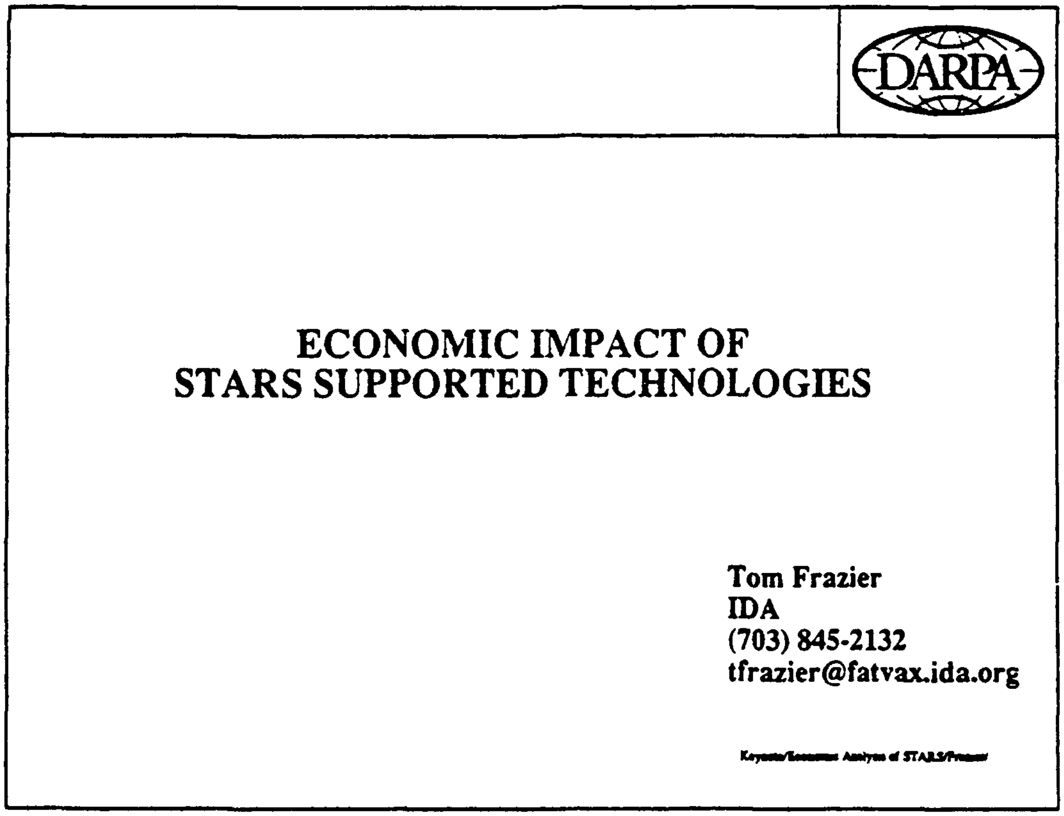
VISION, MISSION, STRATEGY, ACHIEVEMENTS

SUMMARY

DARPA

- Megaprogramming has significant potential to reduce costs and increase quality of large-scale DoD software intensive systems
- Megaprogramming involves cultural change and STARS has a strategy to address this
- We have interim products supporting the transition to megaprogramming
- Join us in making this transition a reality

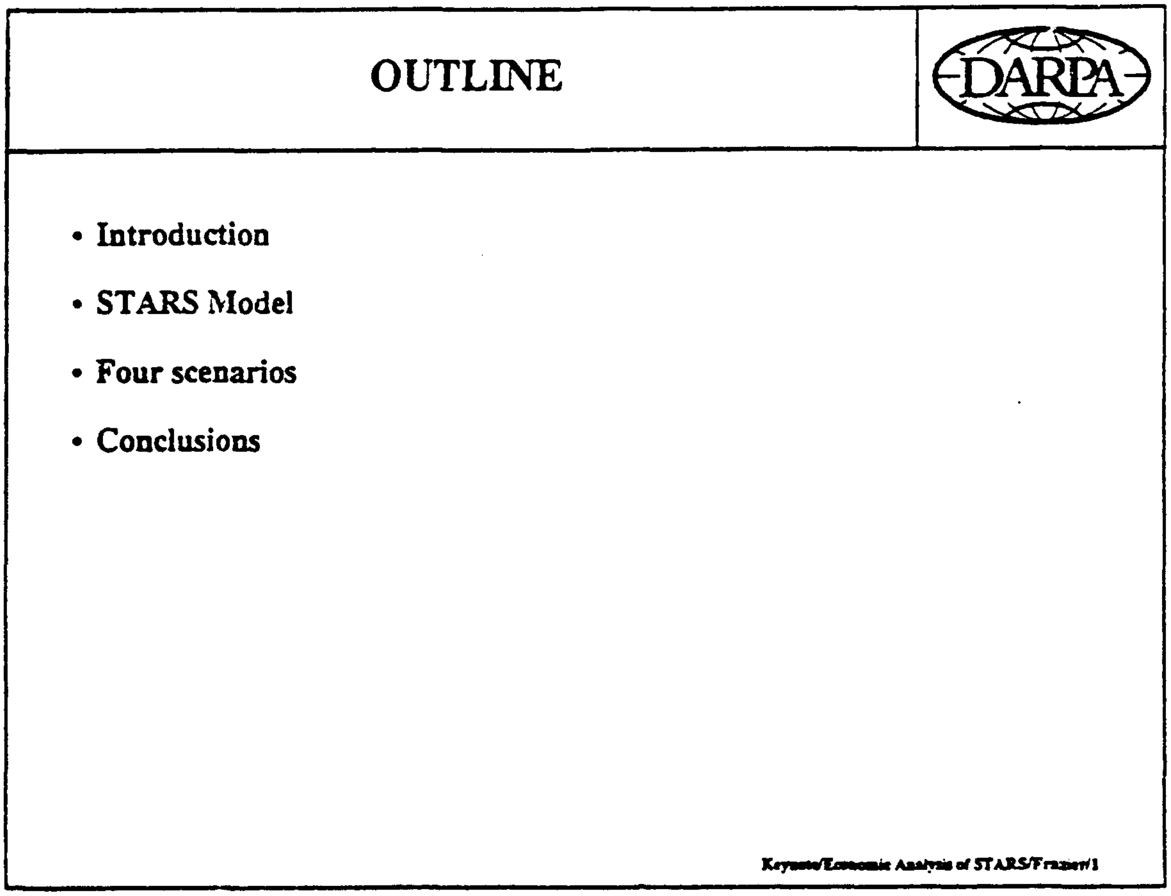
Vision, Mission, Strategy and Achievements/J. Foreman/VG20



ECONOMIC IMPACT OF STARS SUPPORTED TECHNOLOGIES

Tom Frazier
IDA
(703) 845-2132
tfrazier@fatvax.ida.org

Keynote/Economic Analysis of STARS/Frazier

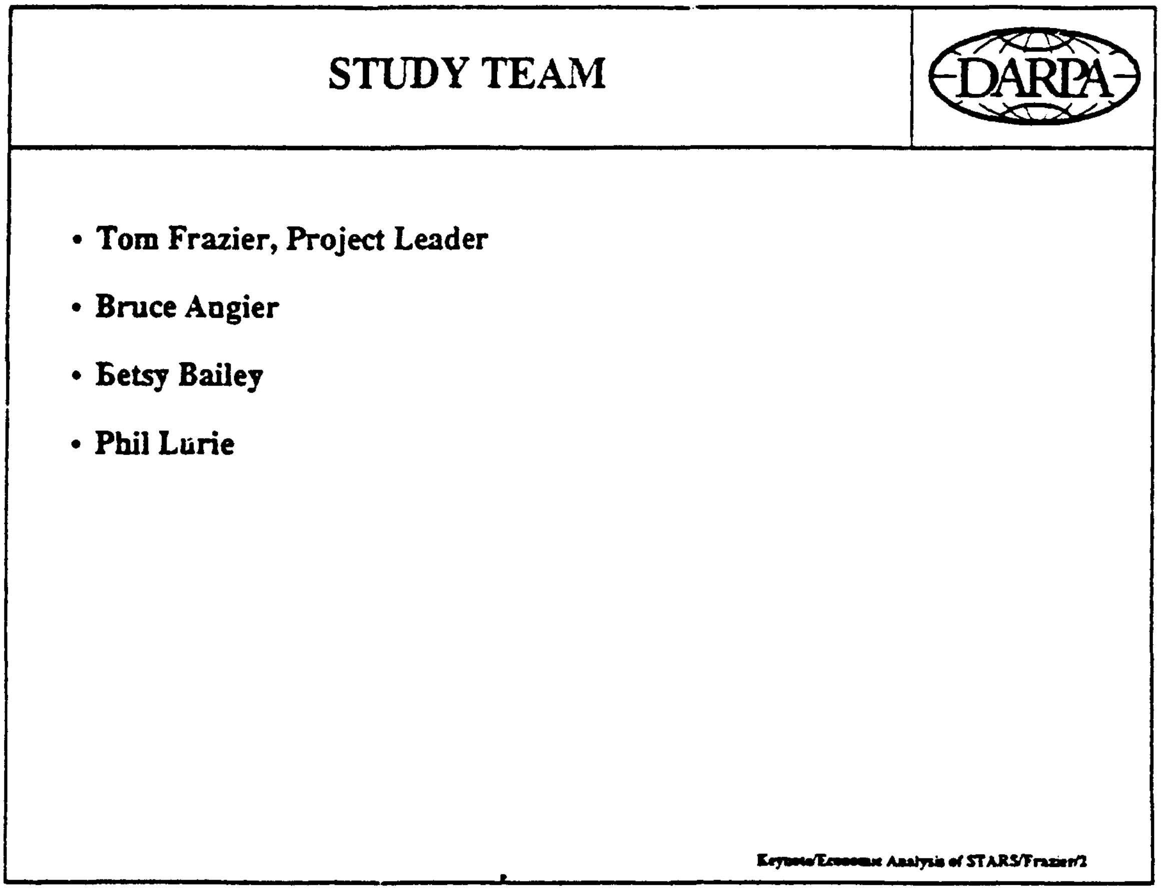


OUTLINE

- Introduction
- STARS Model
- Four scenarios
- Conclusions

Keynote/Economic Analysis of STARS/Frazier/1

This briefing is divided into four parts. The Introduction reviews our collabration with Barry Boehm on an economic analysis of DoD software costs. The work presented here is a more detailed extension of that analysis. The second part of the briefing presents a description of an automated model used to analyze the economic impact of megaprogramming technologies. The third part of the briefing details the results from applying this model to four scenarios. The final part of the briefing presents a set of conclusions drawn from our work.



STUDY TEAM

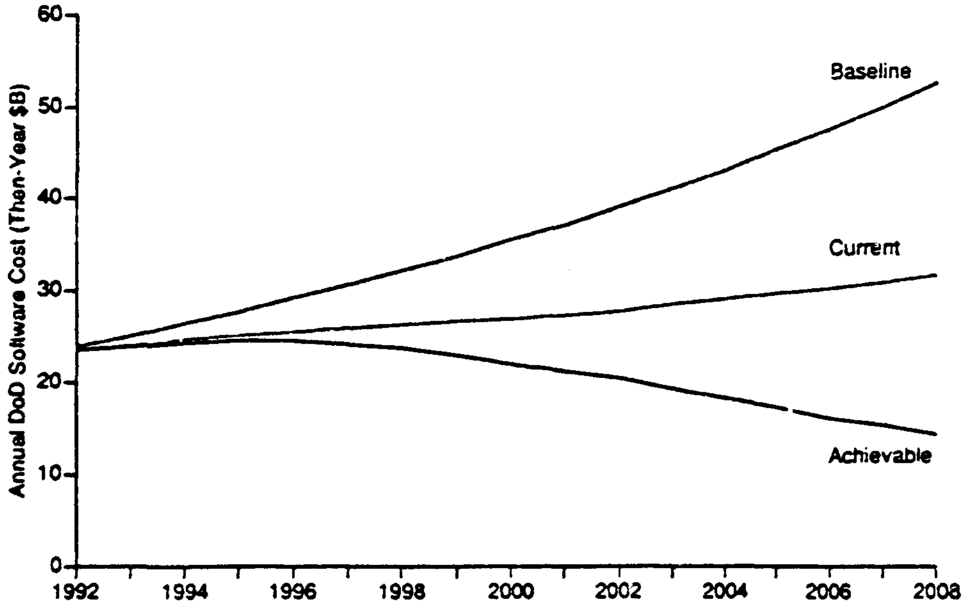
- Tom Frazier, Project Leader
- Bruce Angier
- Betsy Bailey
- Phil Lurie

Keynote/Economic Analysis of STARS/Frazier/2

Tom Frazier, Bruce Angier and Phil Lurie are Research Staff Members in the Cost Analysis and Research Division of the Institute for Defense Analyses (IDA). Betsy Bailey is an Adjunct Staff Member at IDA.

SWTP COST SAVINGS

DARPA



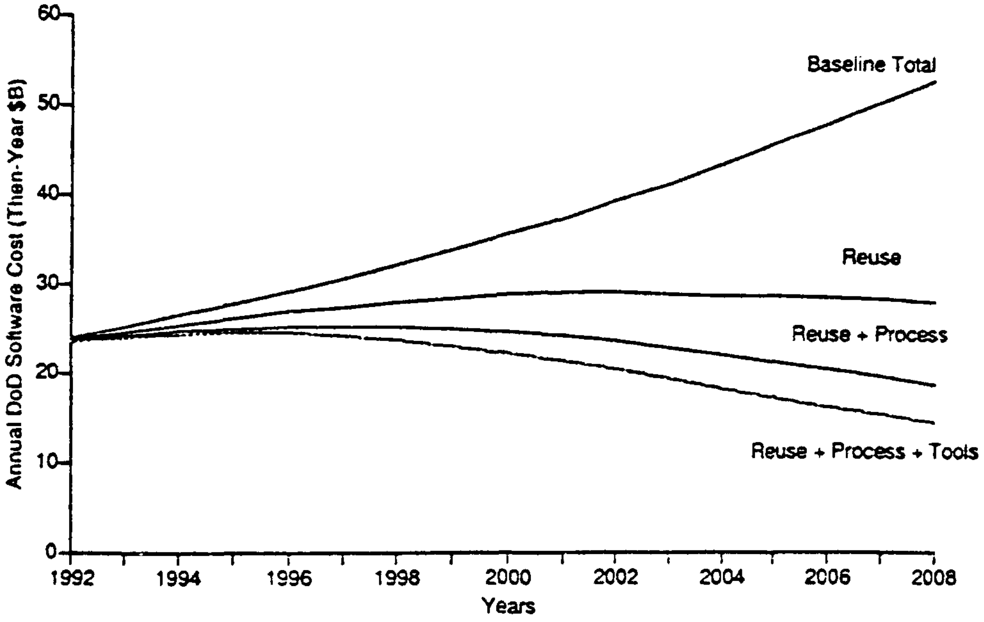
Keynote/Economic Analysis of STARS/Fraser/3

The figure above was taken from Chapter 10 of the Software Technology Plan (SWTP) and represents the results of an earlier economic analysis that we collaborated on with Barry Boehm. This earlier work helps to set the stage for the presentation today which describes a more detailed extension of that analysis. The top line (Baseline) represents the projected growth in DoD software expenditures in the absence of specific DoD technology investments. It assumes that the 1992 expenditures total $24B [EIA80, EIA85, EIA90, AVWK91] and that the annual growth in demand is 4% [EIA90] coupled with a 4% annual growth in productivity as a result of advances from the commercial sector in CASE technology [MARTIN83, LEVITAN88].

The middle and bottom lines (Current and Achievable) are projections based on two different investment scenarios in megaprogramming technologies. The Current scenario represents projections based on the currently planned investments outlined in the SWTP. The Achievable scenario represents further savings from increased levels of investment.

One of the goals stated for the Software Technology Plan is to achieve a factor of two reduction in software unit costs by the year 2000. One of the questions to be explored through the economic modeling discussed here is whether this ambitious goal can be realized.

SWTP COST SAVINGS BY SOURCE

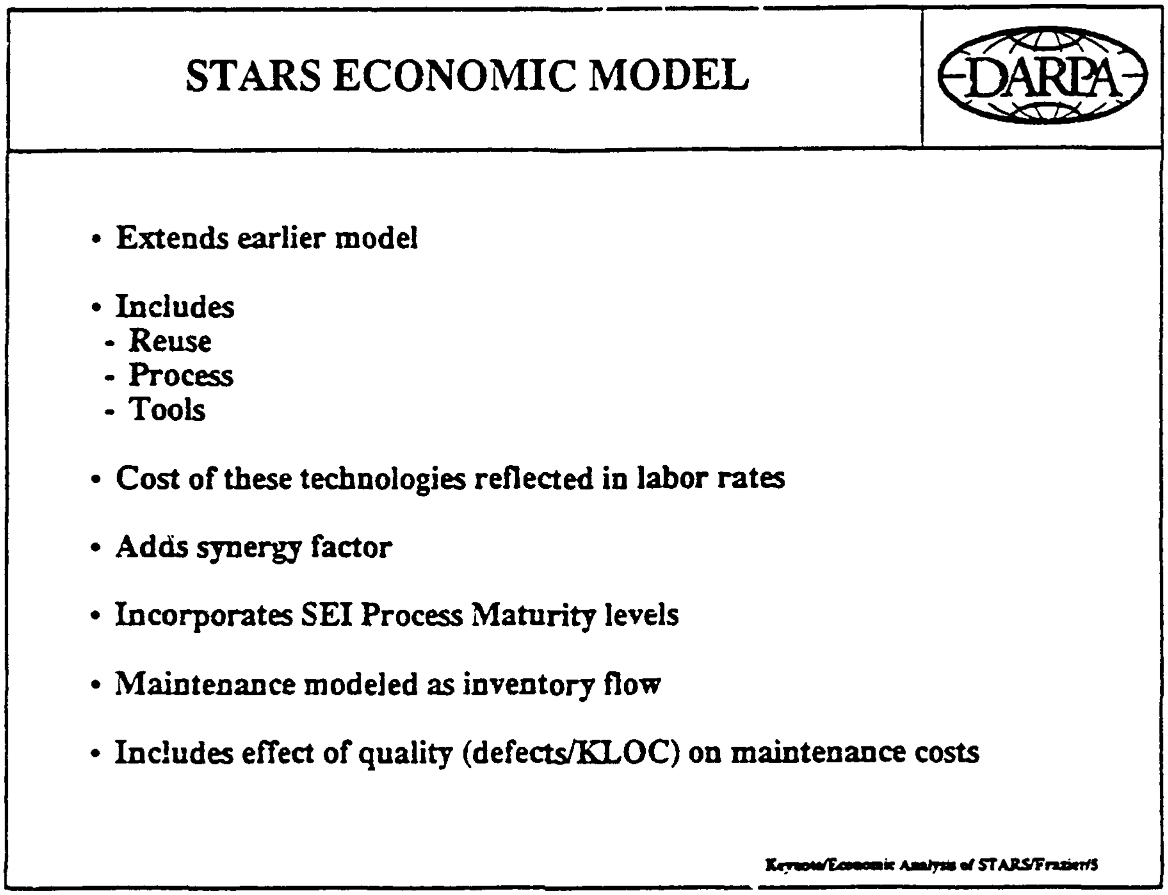


Keynote/Economic Analysis of STARS/Frazier/4

The analysis described in the SWTP assumes savings from three sources: reuse, process improvements, and tools or software engineering environments (SEEs). The figure above shows the contribution of each technology to the savings found under the Achievable investment scenario.

The term "reuse" is intended to apply generally to any form of work avoidance through software reuse, application generators, and commercial-off-the-shelf (COTS) software. Process improvements, which include prototyping and risk management, enable projects to avoid costly rework and work smarter. Improvements in SEE's result in better, more interoperable tools which allow software practitioners to work faster.

In the analysis described in the SWTP, for each of these technologies, there was a time series of parameter values for each year from 1992 to 2008 which represent the fraction of savings (FS) from the use of the technology and the fraction of time (FT) that the technology is actually used. The realized savings for a given year were arrived at by multiplying the FS and FT together and then subtracting the product from 1. This value was then multiplied by the baseline expenditures to yield a new projection which takes into account the savings resulting from the technology. For example, if the FS for reuse in a given year is .70 and the FT is .10, then the realized savings are (.70)(.10) = .07. Subtracting from 1 gives .93 which is then multiplied by the baseline costs to give the new projected cost.



STARS ECONOMIC MODEL

- Extends earlier model
- Includes
  - Reuse
  - Process
  - Tools
- Cost of these technologies reflected in labor rates
- Adds synergy factor
- Incorporates SEI Process Maturity levels
- Maintenance modeled as inventory flow
- Includes effect of quality (defects/KLOC) on maintenance costs

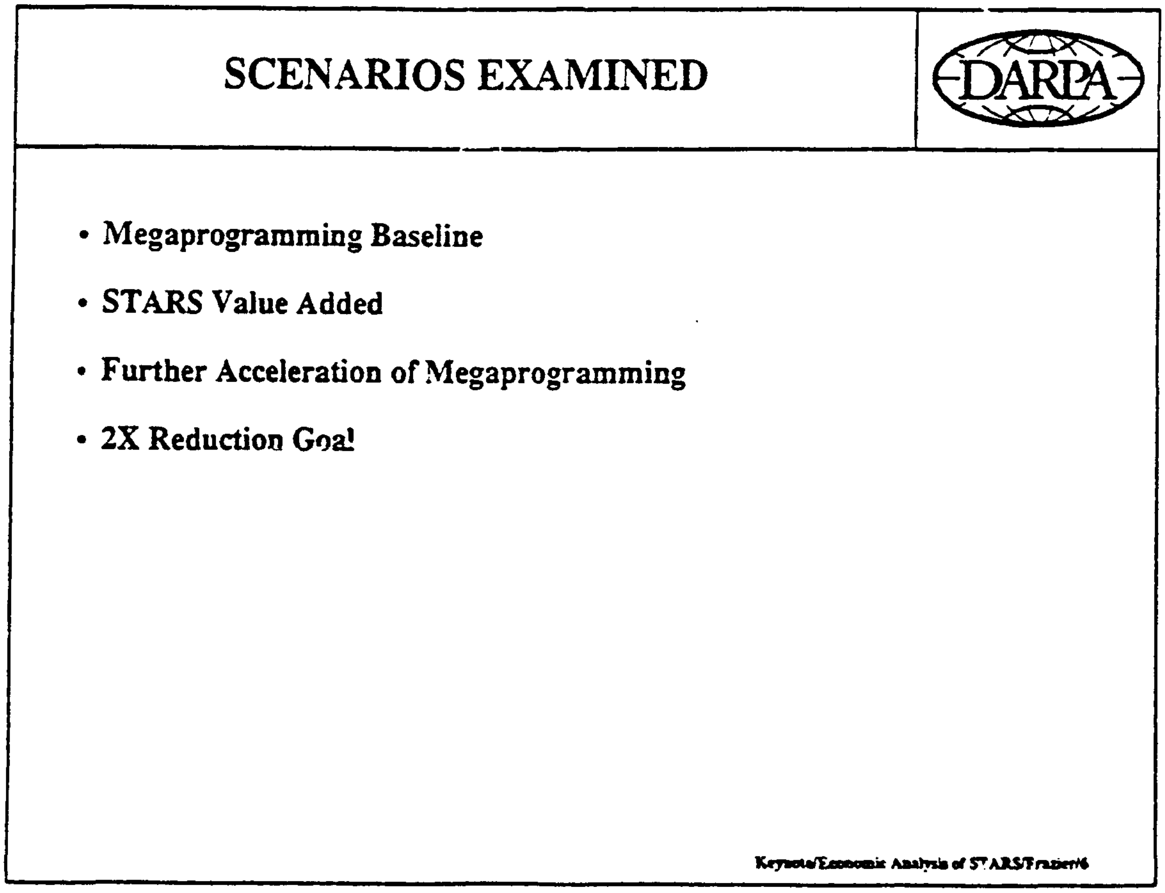
Keynote/Economic Analysis of STARS/Frazier/5

The extension to the earlier model examines savings from the same three technologies. It also reflects the costs incurred in implementing these technologies via increased labor rates.

We felt that any realistic attempt to model the economics associated with megaprogramming must consider the impact of process maturity. The model distinguishes between the five SEI Process Maturity levels. Though not explicit in the SEI framework, we believe there is a correlation between process maturity and the level of sophistication of the megaprogramming technologies. This correlation is reflected in the model by assuming that, for any given year, the FSs and FTs increase with process maturity level. Additional gains are realized at higher levels through the addition of a factor to account for synergy between technologies. While the SEI is primarily responsible for facilitating the movement of software organizations to higher levels, STARS is expected to be a facilitator as well by enhancing the technologies which underlie this movement. The analysis is conservative in that STARS is not given any credit for this facilitation.

A portion of the model deals with development costs and a portion with maintenance. Maintenance is modeled as an inventory flow process and will be described in more detail in a later vugraph.

The model assumes that quality increases with increases in SEI level. Variations in code quality (defects/KLOC) affect the amount of corrective maintenance required in the model and are reflected in maintenance costs.



SCENARIOS EXAMINED

- Megaprogramming Baseline
- STARS Value Added
- Further Acceleration of Megaprogramming
- 2X Reduction Goal

Keynote/Economic Analysis of STARS/Frazier/6

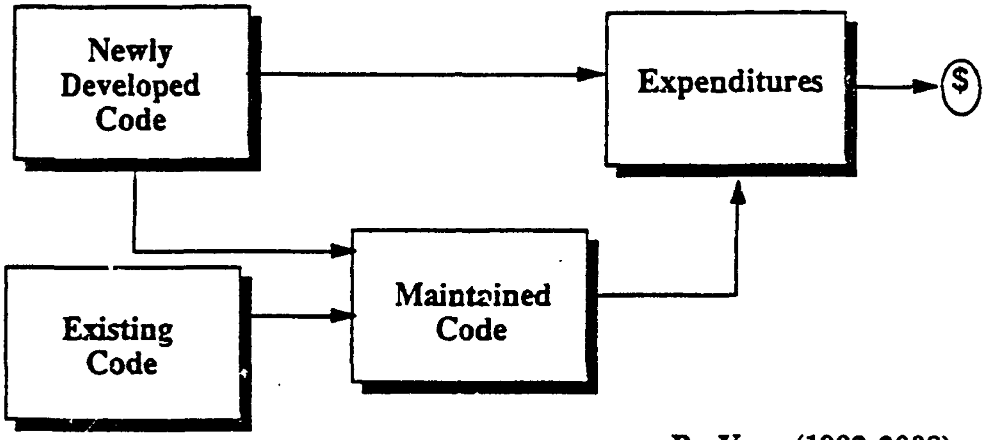
The results of modeling several different scenarios will be described. The scenarios are listed above. Each scenario will be discussed in detail in later vugraphs.

The Megaprogramming Baseline reflects our best estimate of the rate at which the FS and FT associated with reuse, process, and SEEs will change over time. Several studies have shown that the adoption of new technologies is extremely slow. Studies by IDA have shown that, on average, it takes 9 years between the introduction of a technology until it is used 50% of the time (FT = .50) and 18 years until its use approaches 100%. The STARS Added Value scenario reflects changes to the FTs and FSs to reflect acceleration in the adoption of these technologies and greater savings from their use. The Further Acceleration of Megaprogramming scenario looks at the impact of moving the adoption of these technologies up even faster than the rate we envision with STARS. Finally, the 2X reduction scenario looks at one combination of model parameters which do produce the stated goal outlined in the Software Technology Plan of reducing software unit costs by a factor of two. The feasibility of meeting these parameter values will be discussed.

For each of the scenarios, the bottom line values of interest are the total DoD software costs for each year included in the scenario and the Net Present Value of savings in the future. The latter discounts future constant dollar savings by 10% per year and, as such, represents a conservative measure of the economic value of megaprogramming technologies.

Prior to presenting the scenarios, the basic structure of the model is briefly described.

MODEL OVERVIEW



- By Year (1992-2008)
- By SEI Level

Keynote/Economic Analysis of STARS/Frazier/7

The model contains a development and a maintenance portion. Code developed in a given year is costed and tracked through the maintenance portion of the model. Both new code and maintained code is tracked by year and by SEI capability level.

The model is implemented as an Excel™ spreadsheet containing approximately 40,000 cells. It requires about 1.2MB of disk space.

™Excel is a registered trademark of Microsoft Corporation.

NEW DEVELOPMENT PORTION OF MODEL

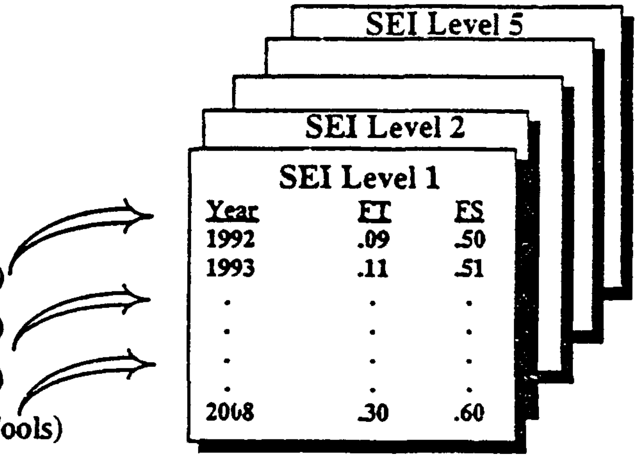
$$PM = \alpha(KDSI)^{\beta} * Reuse * Process * Tools * Synergy$$

Reuse = 1 - (FT * FS)

Process = 1 - (FT * FS)

Tools = 1 - (FT * FS)

Synergy = f (Process, Tools)



Year	FT	FS
1992	.09	.50
1993	.11	.51
.	.	.
.	.	.
.	.	.
.	.	.
2008	.30	.60

Keynote/Economic Analysis of STARS/Frazier/8

The structure of the Development portion of the model is shown above. A COCOMO-like equation is used to calculate effort and costs. Effort calculations are made for each SEI level for each year from 1992 through 2008.

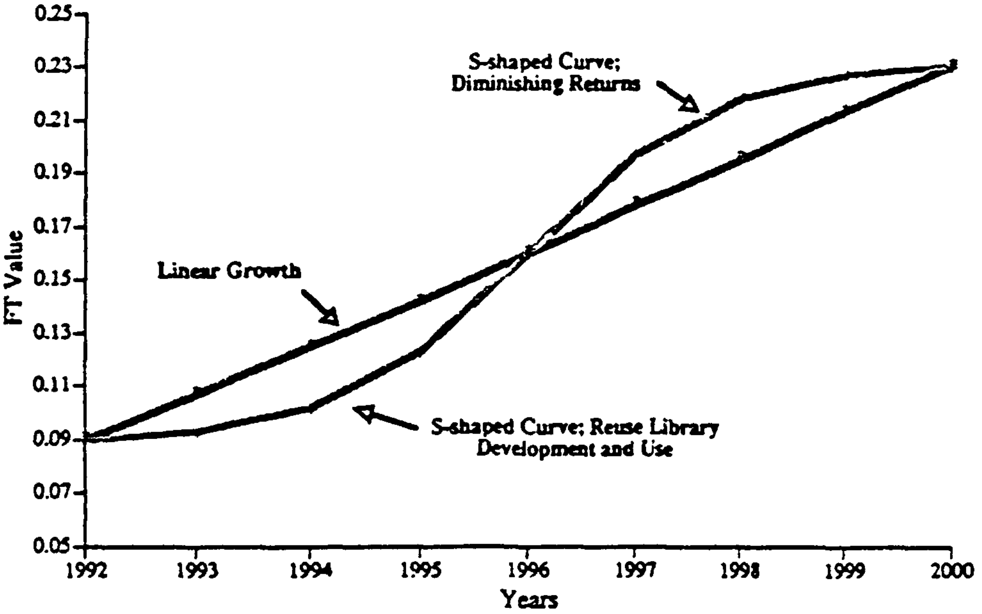
The alpha and beta terms vary across SEI levels as shown below. These coefficients generate software development productivities consistent with those reported in [PUTNAM91].

	α	β
SEI Level 1	4.75	1.06
2	2.80	1.05
3	1.50	1.04
4	1.20	1.03
5	1.00	1.02

The Reuse, Process, and Tool factors in the equation represent the product of FS and FS which is subtracted from 1. These operate as do the Effort Adjustment Factors (EAFs) in COCOMO. The contribution of each of the three factors can be further enhanced via the Synergy factor which is set to 1 for SEI Levels 1 and 2 and less than 1 for Levels 3, 4, and 5.

EXAMPLES OF TECHNOLOGY TRANSITION PATHS

DARPA

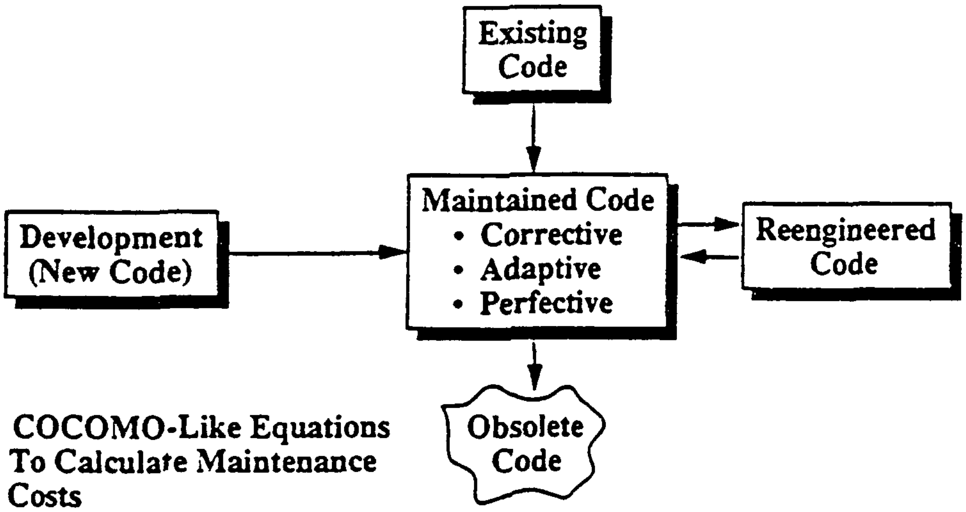


Keynote/Economic Analysis of STARS/Frazier/9

The incorporation of new technologies is represented by the FT parameters in the model. As described earlier, these parameters represent the fraction of software that is being developed using the megaprogramming technologies. The values of these parameters are key determinants of the dollar savings estimated using the model.

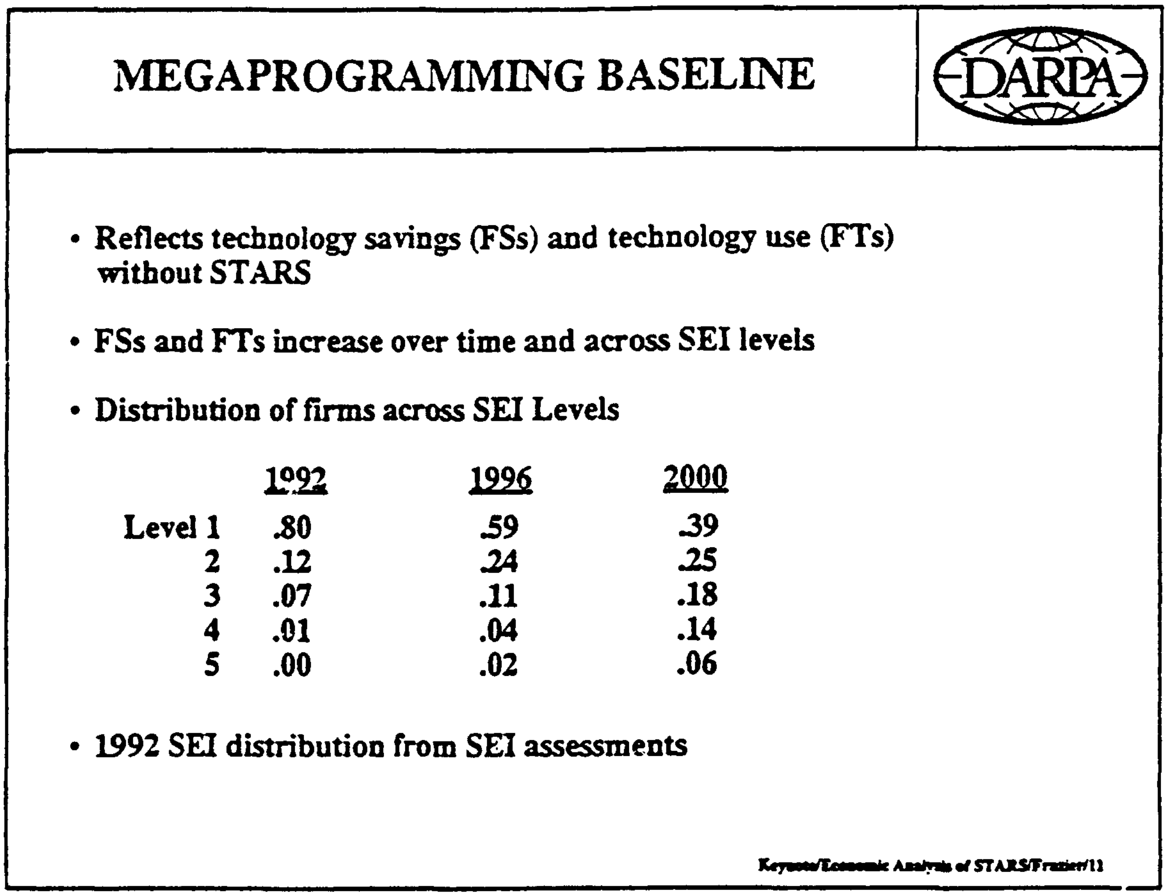
The figure above shows two ways that the FT parameters could be modeled. The straight line shows a fixed increase every year. While this is the simplest way to model changes in FT over time, it may not accurately reflect the path by which some technologies come into use. For example, software reuse may be associated with a substantial overhead at first as software libraries are developed and populated with components. This would result in a slow increase in the early years followed by an acceleration as libraries become more fully populated and finally a flattening out as projects reach a ceiling in terms of the proportion of software that can be developed from reusable components. This technology transition path is represented by the S-shaped curve in the figure. In the model, the FTs are linear across time for process and SEEs and S-shaped for reuse.

MAINTENANCE PORTION OF MODEL



- COCOMO-Like Equations To Calculate Maintenance Costs

The Maintenance portion is modeled as an inventory flow process. Maintenance begins with a stock of existing code to be maintained. Each year, newly developed code is added to the total stock of maintained cost. Some code is deleted from the total stock via obsolescence and some is reengineered. A portion of the stock of maintained code is changed as a result of corrective, adaptive, and perfective maintenance. Variations is code quality (defects/KLOC) affect the amount of corrective maintenance and are reflected in maintenance costs. The model assumes different defect densities and different maintenance productivities for different SEI levels.



MEGAPROGRAMMING BASELINE

- Reflects technology savings (FSs) and technology use (FTs) without STARS
- FSs and FTs increase over time and across SEI levels
- Distribution of firms across SEI Levels

	1992	1996	2000
Level 1	.80	.59	.39
2	.12	.24	.25
3	.07	.11	.18
4	.01	.04	.14
5	.00	.02	.06

- 1992 SEI distribution from SEI assessments

Keynote/Economic Analysis of STARS/Frazier/11

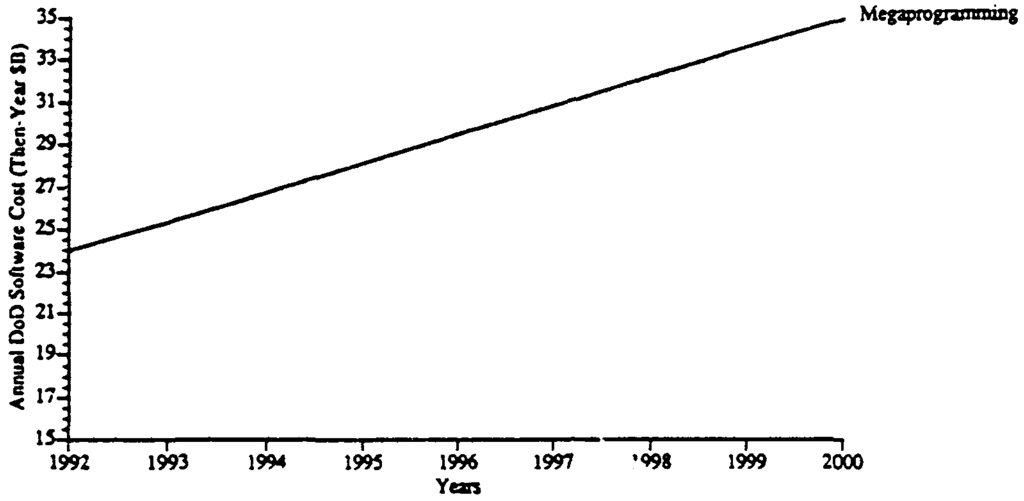
The first scenario represents the Megaprogramming Baseline. This scenario reflects our best estimates of the savings (FSs) resulting from megaprogramming technologies and the fraction of time (FT) they are likely to be used in the absence of STARS-related efforts to accelerate these technologies.

The distribution of projects across SEI levels is shown above. The starting values were taken from the results of the most recent SEI assessments [KITSON91]. The distribution changes annually with fewer projects in Level 1 over time and a greater proportion at the higher levels. The movement across levels represents our best estimate based on work by [PUTNAM91]. (The SEI currently has no such projections.)

In the Megaprogramming Baseline, the FS and FT values increase gradually over time and as one moves up the SEI levels.

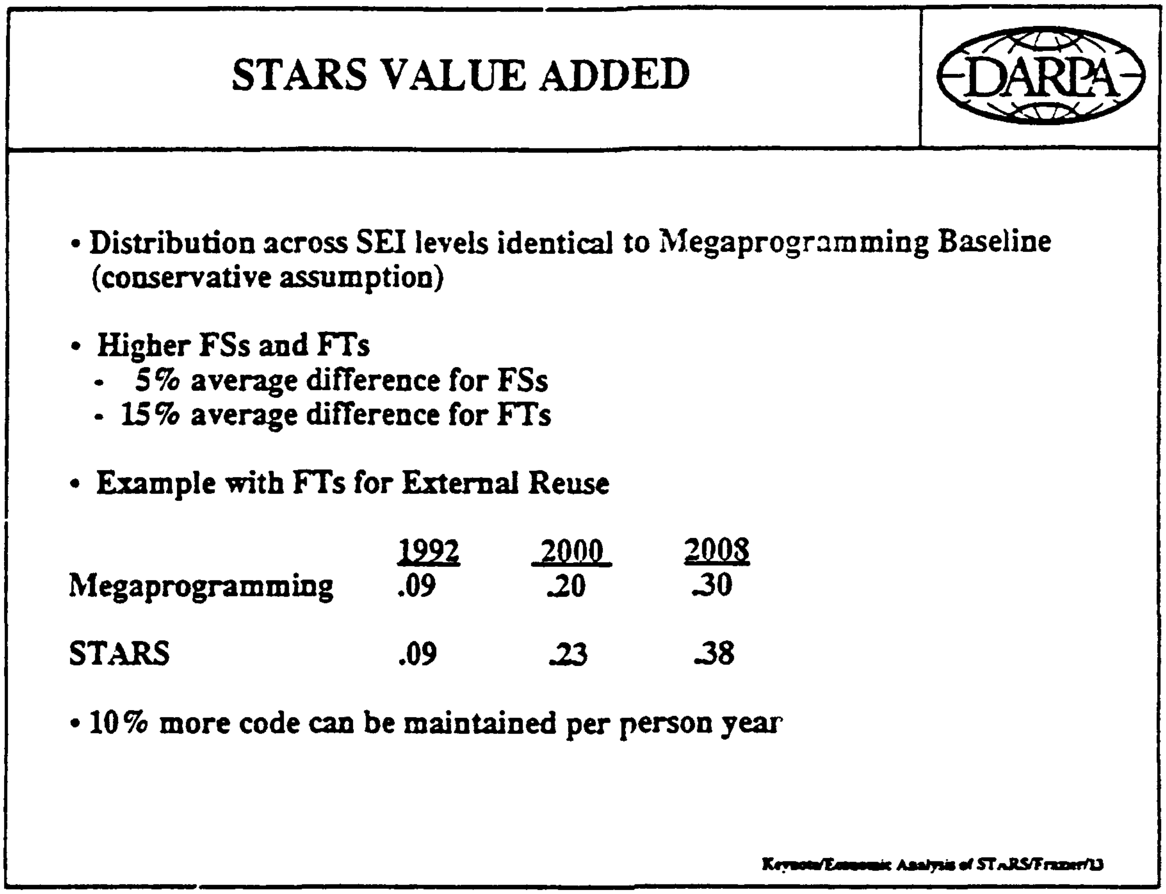
MEGAPROGRAMMING BASELINE: RESULTS

DARPA



Keynote/Economic Analysis of STARS/Frazier/12

The estimated total annual DoD software costs for the Megaprogramming scenario is presented in the chart shown above. We estimate by the year 2000, this scenario will result in annual costs of about $35 billion in then-year dollars. The results are presented in billions of then-year dollars. In constant 1992 dollars, the annual costs estimates remain fairly constant around $24 billion over the 1992-2000 time period.



STARS VALUE ADDED

- Distribution across SEI levels identical to Megaprogramming Baseline (conservative assumption)
- Higher FSs and FTs
  - 5% average difference for FSs
  - 15% average difference for FTs
- Example with FTs for External Reuse

	1992	2000	2008
Megaprogramming	.09	.20	.30
STARS	.09	.23	.38

- 10% more code can be maintained per person year

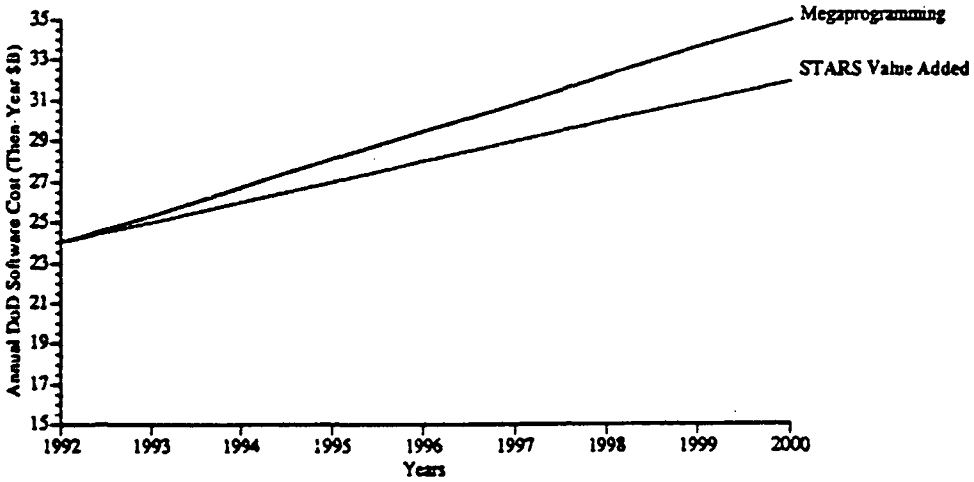
Keynote/Economic Analysis of STARS/Frazer/13

The distribution of projects across SEI levels is the same for the STARS Value-Added scenario as for the Megaprogramming Baseline. While the FSs and FTs begin with identical values for the first year (1992), the rate of increase is higher for the STARS Value-Added scenario with the result that, on average, the FS values are approximately 5% higher than for the Megaprogramming Baseline while the FT values are approximately 15% higher.

One additional difference between the two scenarios is a 10% increase for the STARS Value Added in the amount of code that can be maintained per person person year.

STARS VALUE ADDED: RESULTS

DARPA

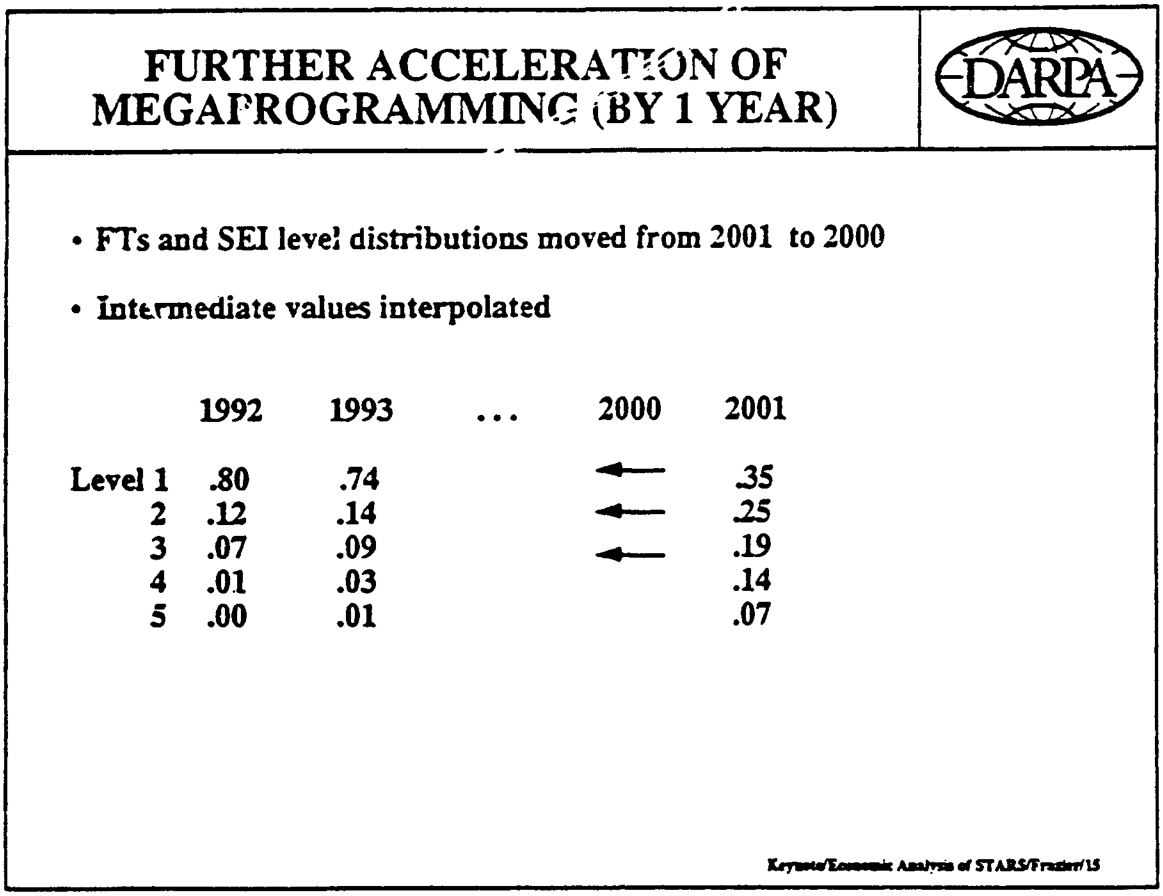


Scenario	NPV ($B)
STARS Value Added	$6.6

Keynote/Economic Analysis of STARS/Frazier/14

This chart presents the estimated additional savings DoD might realize due to the STARS program. Again, the estimates are presented in billions of then-year dollars. If we compute the value of the savings in today's dollars and also account for the time value of money by "discounting" the stream of savings, the result is a financial measure of merit called the Net Present Value (NPV). The Office of Management and Budget (OMB) guidelines specify a discount rate of 10%. This rate was used in all the reported NPV results. The NPV of the additional STARS savings is estimated to be $6.6 billion.

This is a conservative estimate of the potential savings from the STARS program for two reasons. First, many of the savings will only be realized in the first decade of the next century. For purposes of this study we ignore those savings. Second, we assume it will take some time for the full extent of the payoff from STARS technology infusion. Given OMB's 10% discount rate, savings that occur in the out-years are rather severely discounted.



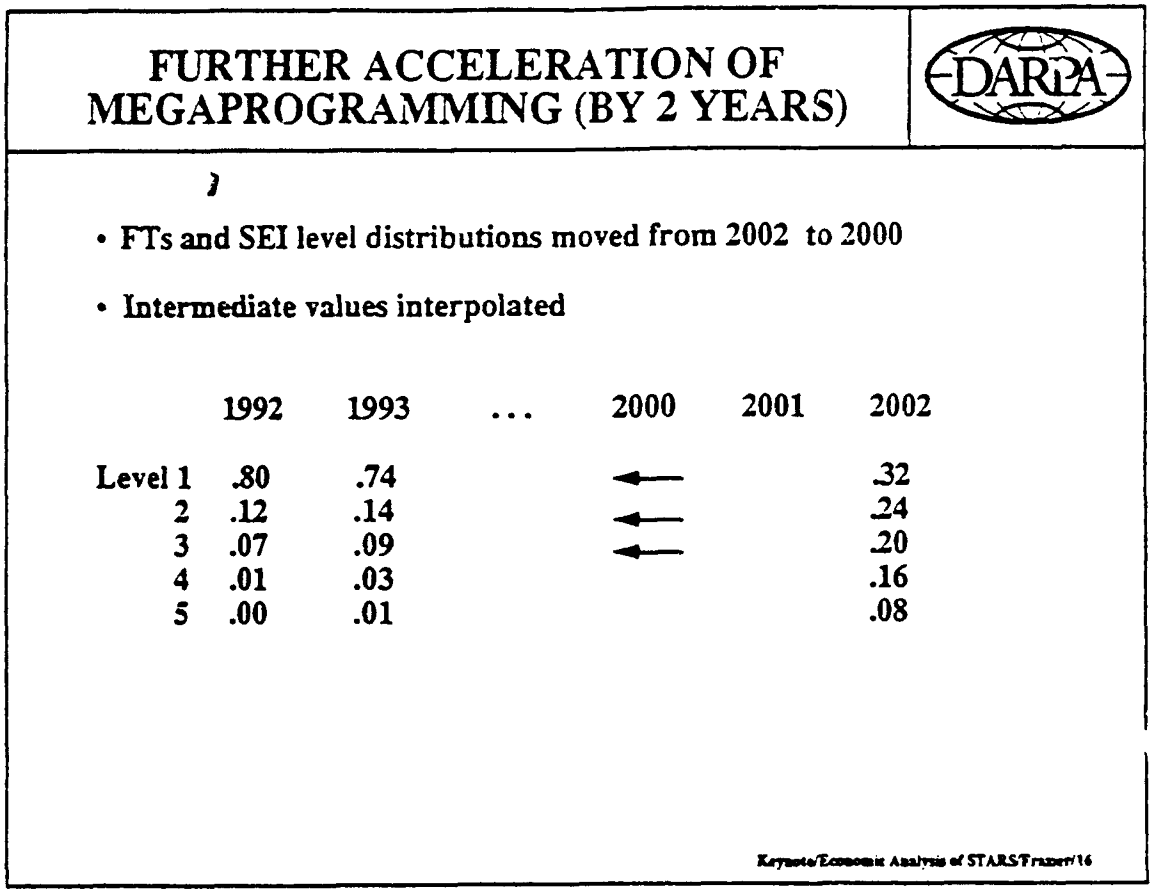
FURTHER ACCELERATION OF MEGAPROGRAMMING (BY 1 YEAR)

- FTs and SEI level distributions moved from 2001 to 2000
- Intermediate values interpolated

	1992	1993	...	2000	2001
Level 1	.80	.74		←	.35
2	.12	.14		←	.25
3	.07	.09		←	.19
4	.01	.03			.14
5	.00	.01			.07

Keynote/Economic Analysis of STARS/Frazier/15

The next scenario looks at the effects of accelerating technology transition by one year. The STARS Value Added scenario was choosen as the baseline for this analysis. The distribution of firms across SEI levels in the year 2001 was assigned to the year 2000. In addition, the FTs from the year 2001 were assigned to the year 2000. Values between 1992 and 2000 were interpolated.



FURTHER ACCELERATION OF MEGAPROGRAMMING (BY 2 YEARS)

- FTs and SEI level distributions moved from 2002 to 2000
- Intermediate values interpolated

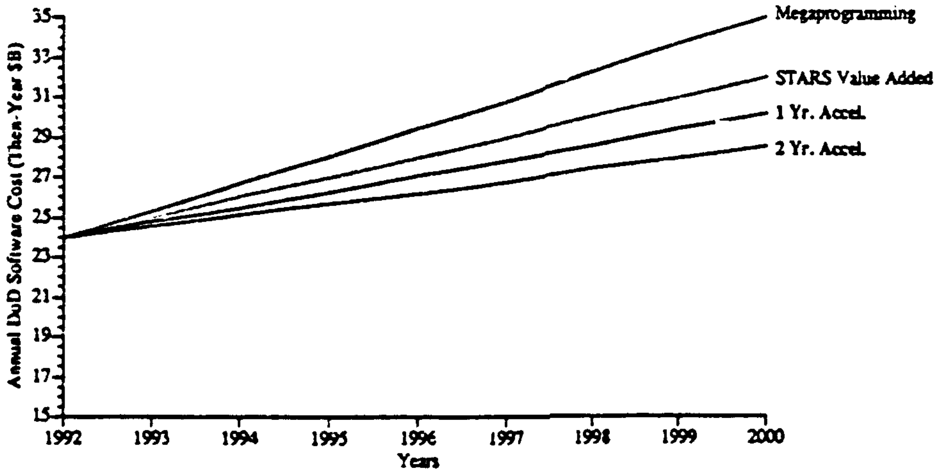
	1992	1993	...	2000	2001	2002
Level 1	.80	.74		←		.32
2	.12	.14		←		.24
3	.07	.09		←		.20
4	.01	.03				.16
5	.00	.01				.08

Keynote/Economic Analysis of STARS/Fraser/16

This scenario looked at the effects of accelerating technology transition by two years. The STARS Value Added scenario was again chosen as the baseline for this analysis. The distribution of firms across SEI levels in the year 2002 was assigned to the year 2000. In addition, the FTs from the year 2002 were assigned to the year 2000. Values between 1992 and 2002 were interpolated.

FURTHER ACCELERATION OF MEGAPROGRAMMING: RESULTS

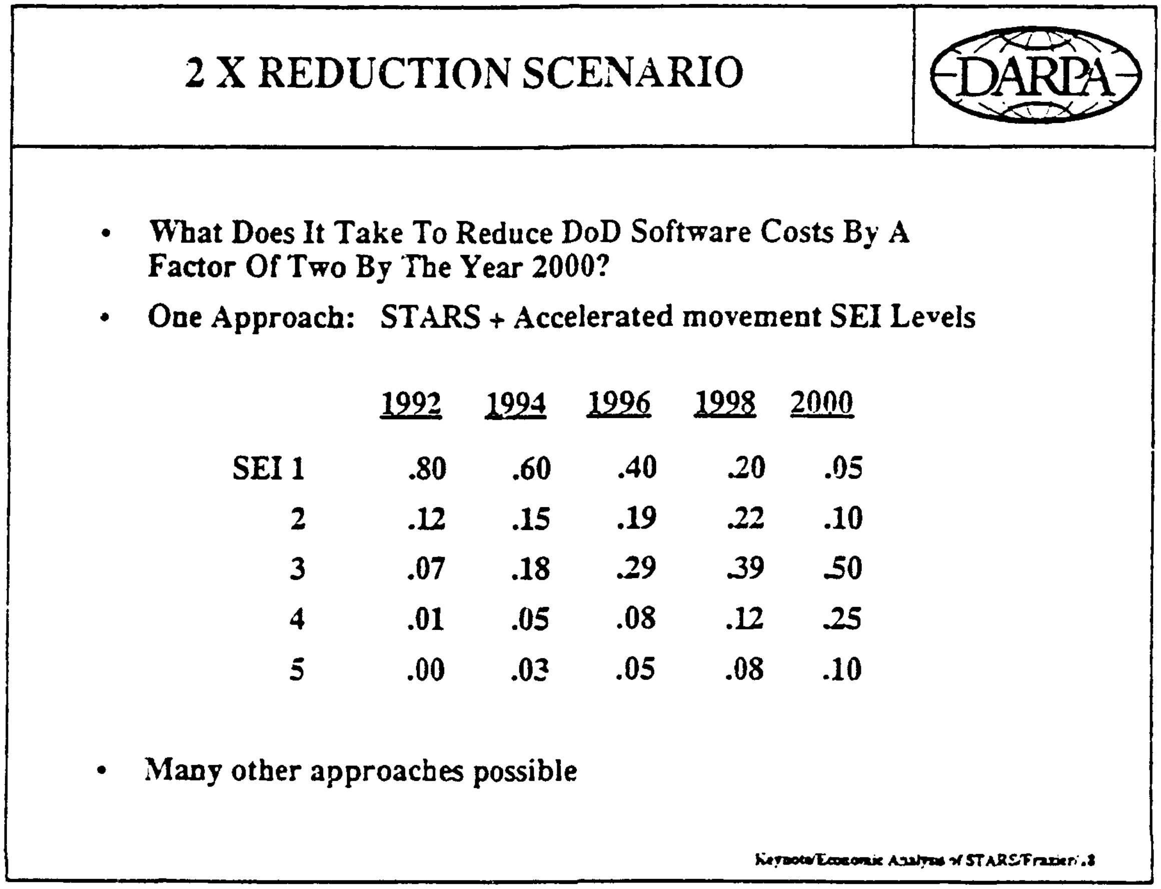
DARPA



Scenario	NPV ($B)
STARS Value Added	$6.6
1 Yr. Accel.	4.1
2 Yr. Accel.	2.5

Keynote/Economic Analysis of STARS/Frazer/17

The chart shows the estimated additional savings resulting from accelerating the pace of technology transition assumed in the STARS scenario by both one and two years. The NPV of these additional savings for one year acceleration is estimated to be $4.1 billion. The NPV of the incremental savings gained by speeding up software technology utilization by two years is estimated to be $2.4 billion. The total NPV of the combined savings due to the acceleration of new technologies is $6.5 billion, or about equal to the estimated savings due to STARS.



2 X REDUCTION SCENARIO

- What Does It Take To Reduce DoD Software Costs By A Factor Of Two By The Year 2000?
- One Approach: STARS + Accelerated movement SEI Levels

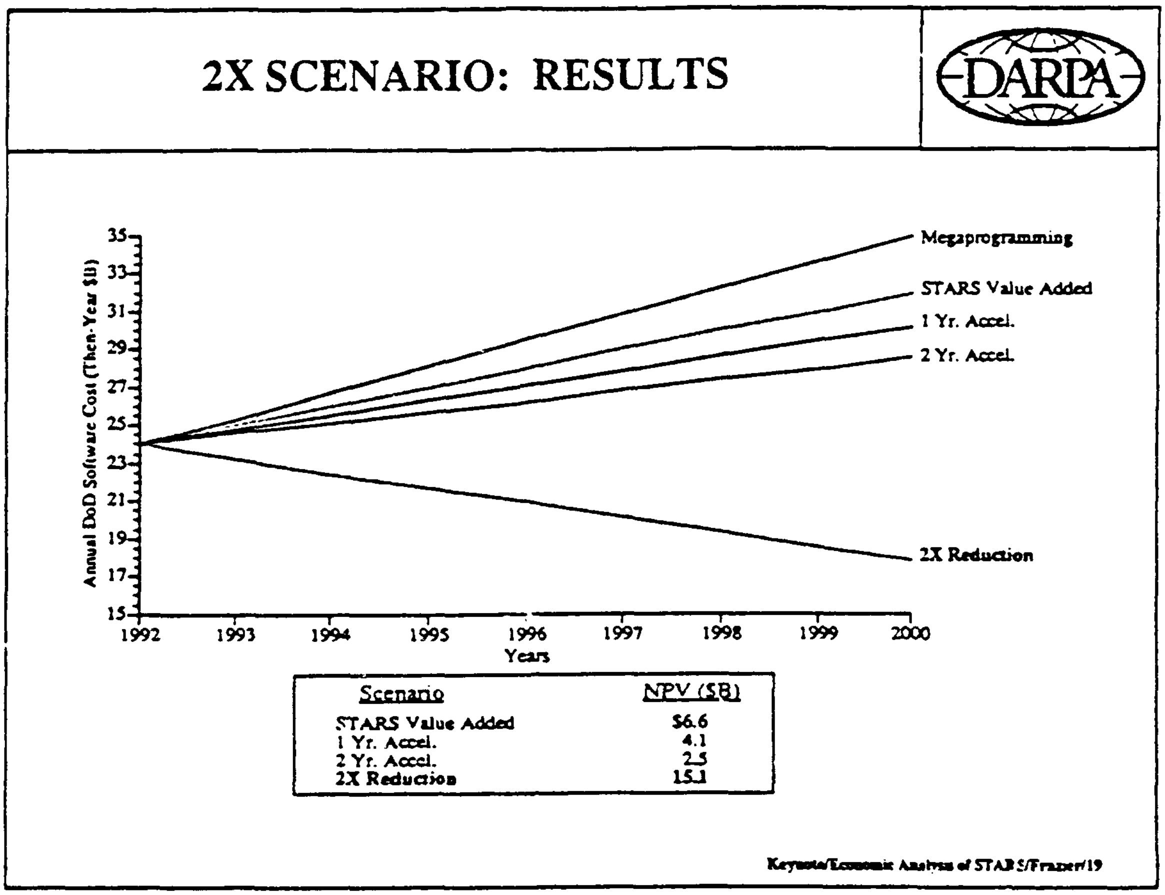
	1992	1994	1996	1998	2000
SEI 1	.80	.60	.40	.20	.05
2	.12	.15	.19	.22	.10
3	.07	.18	.29	.39	.50
4	.01	.05	.08	.12	.25
5	.00	.03	.05	.08	.10

- Many other approaches possible

Keynote/Economic Analysis of STARS/Frazier/.8

As noted earlier, one of the goals of the Software Technology Plan is to reduce DoD software costs by a factor of two by the year 2000. We tried various combinations of savings generated by STARS and savings generated by moving firms up SEI levels in order to ascertain if such a reduction was possible. Using the Megaprogramming Baseline expenditures of $24B (in constant 1992 dollars) for the year 2000, we found a combination of savings from STARS plus savings from moving firms across SEI levels that will result in expenditures of $12B in the year 2000.

Note that currently, 80% of software firms are in Level 1. This value has to decrease to 5% in just eight years to realize the 2X reduction goal. Is it possible to move from 80% of the firms in Level 1 to only 5% by the year 2000? It seems unlikely. However, without programs such as STARS it is virtually impossible.

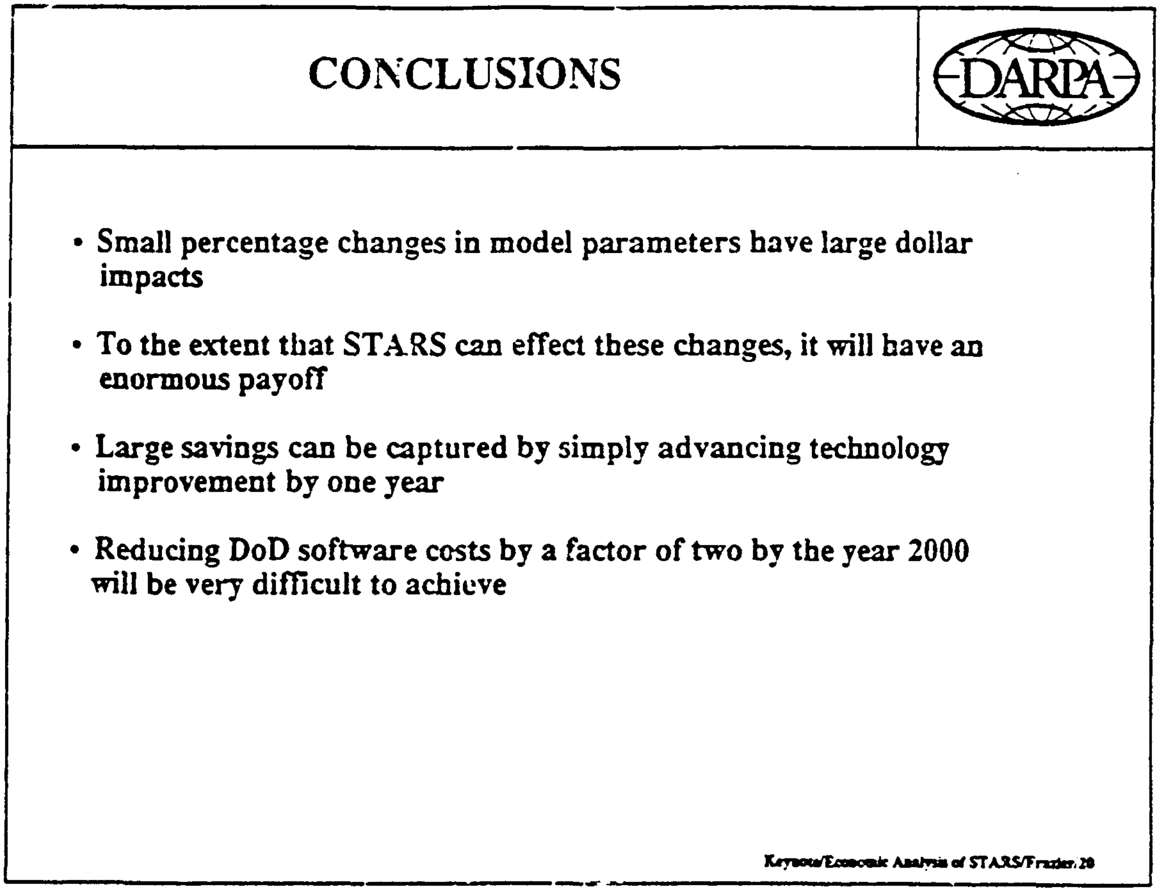


2X SCENARIO: RESULTS

Scenario	NPV ($B)
STARS Value Added	$6.6
1 Yr. Accel.	4.1
2 Yr. Accel.	2.5
2X Reduction	15.1

Keynote/Economic Analysis of STARS/Frazer/19

The results of the final scenario are presented graphically above. The estimated NPV of the additional savings from this scenario is $15.1 billion.



CONCLUSIONS

- Small percentage changes in model parameters have large dollar impacts
- To the extent that STARS can effect these changes, it will have an enormous payoff
- Large savings can be captured by simply advancing technology improvement by one year
- Reducing DoD software costs by a factor of two by the year 2000 will be very difficult to achieve

Keynote/Economic Analysis of STARS/Frazier.20

This research is on-going and, therefore, any conclusions we might draw about the particular dollar savings must be viewed as tentative. However, there are several conclusions that can be put forward.

First, the model is very sensitive to small changes in several key parameters. These include the distribution of firms across SEI levels, the values of FT, especially in the early years before their impact is dampened by discounting, and the amount of code that can be maintained by one person per year.

Second, to the extent that the STARS program can effect these changes, it will have a relatively large payoff. Even if the estimated discounted savings are cut in half, STARS is still extremely cost effective.

Third, the model suggests that a small acceleration in the introduction of new software technologies has a large payoff.

Finally, achieving the goal of reducing total DoD software costs by the year 2000 will require significant improvements in the way DoD develops and maintains software.

REFERENCES

[AVWK90] *Aviation Week & Space Technology*, March 18, 1991, Vol. 134, No. 11, p. 51.

[BASILI 81] Basili, V.R., and Freburger, K., "Programming Measurement and Estimation in the Software Engineering Laboratory," *Journal of Systems and Software*, February 1981.

[BOEHM81] Boehm, Barry W., *Software Engineering Economics*, Prentice Hall, Englewood, NJ, 1981.

[BOEHM 89] Boehm, B.W., and Royce, W.E., "Ada COCOMO," Proceedings Fourth COCOMO Users' Group Meeting, SEI, November 1989.

[BOEHM91a] Boehm, B.W., "Economic Analysis of Software Technology Investments," April 29, 1991, briefing charts presented at Mitre Software Engineering Economics Conference, April 29, 1991.

[BOEHM91b] Boehm, B.W., Frazier, T.P., Angier, B., Bailey, E.K., and Wilson, K.L., *A User's Guide For The Software Technology Economic Impact Model*, Institute for Defense Analyses, IDA Document D-971, October 1991.

[EIA80] "DoD Digital Data Processing Study: A Ten Year Forecast," Electronic Industries Association, mimeographed annotated briefing.

[EIA85] "DoD Computing Activities and Programs, 1985 Specific Market Study," Requirements Committee, Government Division, Electronic Industries Association, mimeographed annotated briefing.

[EIA90] *EIA Information System Forecast for the 1990's: the 2nd Annual Federal Information Systems Conference Final Report*, Requirements Committee, Government Division, Electronics Industry Association, May 30, 1990, Alexandria, VA

[EIA91] *EIA Information System Forecast for the 1990's: the 3rd Annual Federal Information Systems Conference Final Report*, Requirements Committee, Government Division, Electronics Industry Association, May 23, 1991, Alexandria, VA

[HELLER90] Heller, Gerald H. and Page, Gerald T., "Impact of a Process Improvement Program in a Production Software Environment: Are We any Better?", *Proceedings of the Fifteenth Annual Software Engineering Workshop*, (SEL-90-006, NASA Software Engineering Laboratory, Goddard Space Flight Center, November 1990.

[HUMPHREY89] Humphrey, W. S., Kitson, D. H., and Kasse, T. C., *The State of Software Engineering Practice: A Preliminary Report*, CMU/SEI-89-TR-1 or ESD-TR-89-01, Software Engineering Institute, February, 1989.

[JONES 86] Jones, T.C., *Programming Productivity*, McGraw Hill, 1986.

[KITSON91] Kitson, David and Masters, Stephen, "State of the Practice: A Software Process Maturity Prospective", (mimeo), Software Engineering Institute, August 1991.

[LEVITAN88] Levitan, K. B., Salasin, J., Frazier, T. P., and Angier, B. N., Final Report on the Status of Software Obsolescence in the DoD, IDA Paper P-2136, Institute for Defense Analyses, August 1988.

[MARTIN83] Martin, E., "The Context of STARS," *Computer*, Vol. 16, No. 3, March 1983, p. 14-17.

[OMB72] Office of Management and Budget, "Discount Rates to be Used in Evaluating Time-Distributed Costs and Benefits", Circular No. A-94, March 27, 1972.

[PUTNAM91] Putnam, L. H., "Linking the QSM Productivity Index with the SEI Maturity Levels and Projecting Transitions to Various Maturity Levels", © 1991, QSM, McLean, Va.

[RATIONAL89] Mimeographed materials, Rational Corp., dated December 1989.

[REIFER90] Reifer, D., "Reuse Metrics and Measurements", *Proceedings of the Fifteenth Annual Software Engineering Workshop*, (SEL-90-006, NASA Software Engineering Laboratory, Goddard Space Flight Center, November 1990.

[SEIDOWITZ89] Seidowitz, E. and Stark, M., "Ada in the SEL: Experience with Operational Ada Projects," *Proceedings of the Second NASA Ada Users' Symposium*, SEL-89-008, NASA/SEL, Goddard Space Flight Center, Greenbelt, MD, November, 1989.

[SHAFER91] Constructed from hourily labor rates obtained from William Shafer, IDA.

[SWTP91] Department of Defense Software Technology Plan, Volume II: Plan of Action, Prepared for Director of Defense Research and Engineering, Draft 6, October 31, 1991

[WILLIS90] Willis, Ronald R., "Case History and Lessons Learned in Software Process Improvement", Mimeo, April 1990.

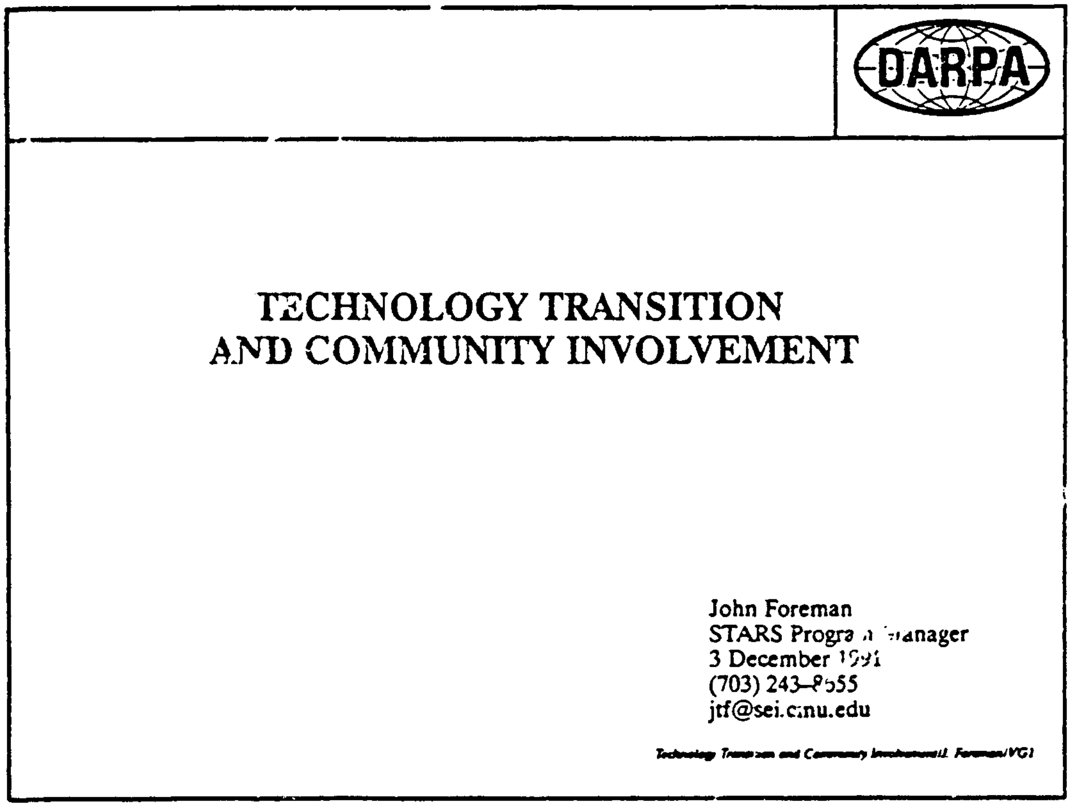
Parameter Type	Source
DoD Software Spending	EIA(all), AVWK91
Discount Rate	OMB72
SEI Distributions (starting)	HUMPHREY89, KITSON91
COCOMO parameters and productivity start values	Fit to material cited in PUTNAM90, also BOEHM89
Productivity Growth	MARTIN83, LEVITAN88
Reuse	
FT	SWTP91
FS	SEIDOWITZ89, REIFER90, BASILI81
Tools	
FT	SWTP91
FS	SWTP91, RATIONAL
Process	SWTP91, JONES86
General Maintenance	BOEHM81, HELLER90, PUTNAM91
Labor Rates	SHAFER91, WILLIS90

ACRONYMS

DoD	-	Department of Defense
STARS	-	Software Technology for Adaptable, Reliable Systems
SEI	-	Software Engineering Institute
IDA	-	Institute for Defense Analyses
FS	-	Fraction of Savings
FT	-	Fraction of Time
PM	-	Person Months
CASE	-	Computer Aided Software Engineering
COCOMO	-	Constructive Cost Model
LOC	-	Lines of Code
DARPA	-	defined on other presentations
KDSI	-	Thousands of Delivered Source Instructions
SWTP	-	Software Technology Plan

NOTES

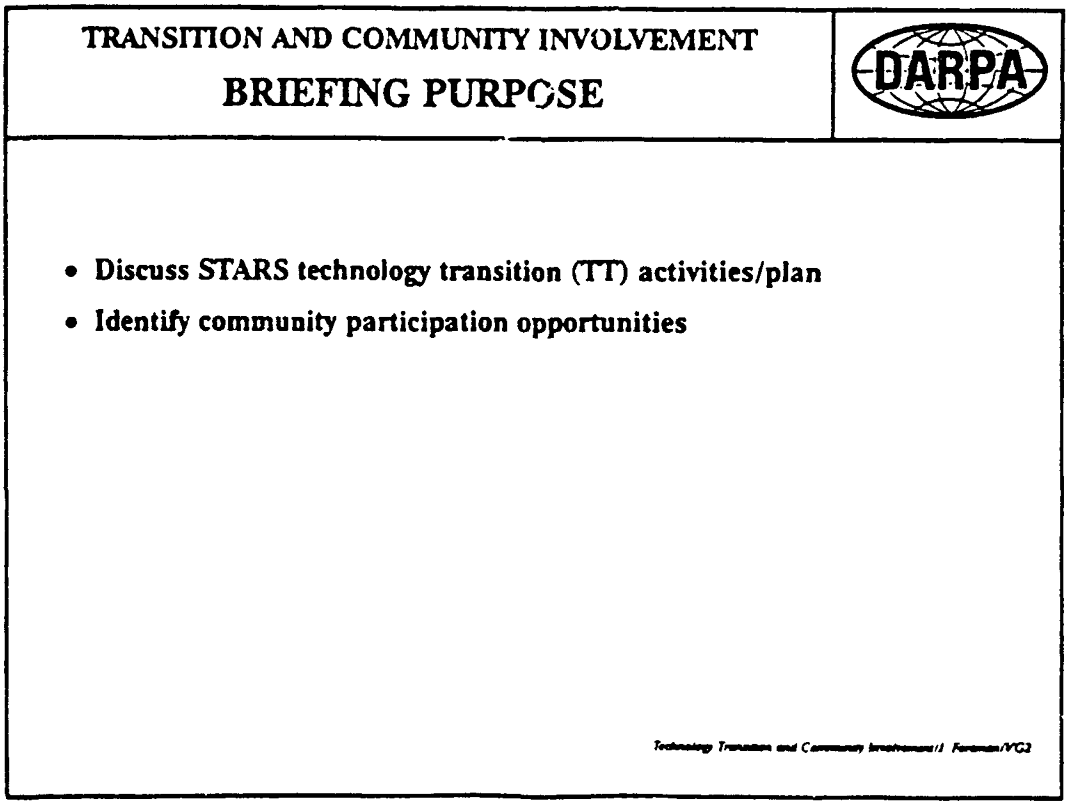
NOTES



TECHNOLOGY TRANSITION AND COMMUNITY INVOLVEMENT

John Foreman
STARS Program Manager
3 December 1991
(703) 243-8655
jtf@sei.cmu.edu

Technology Transition and Community Involvement/J. Foreman/VG1



TRANSITION AND COMMUNITY INVOLVEMENT

BRIEFING PURPOSE

- Discuss STARS technology transition (TT) activities/plan
- Identify community participation opportunities

Technology Transition and Community Involvement/J. Foreman/VG2

TRANSITION AND COMMUNITY INVOLVEMENT

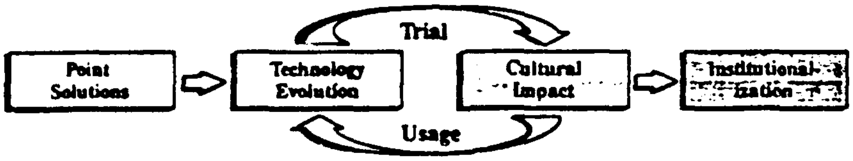
OUTLINE

- Megaprogramming and Culture Change
- Community Involvement
- Affiliates Program

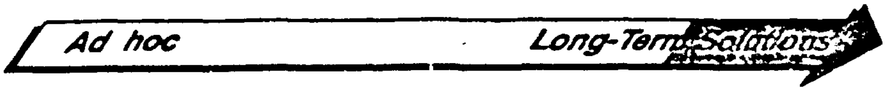
Technology Transition and Community Involvement/J. Foreman/VG1

TRANSITION AND COMMUNITY INVOLVEMENT

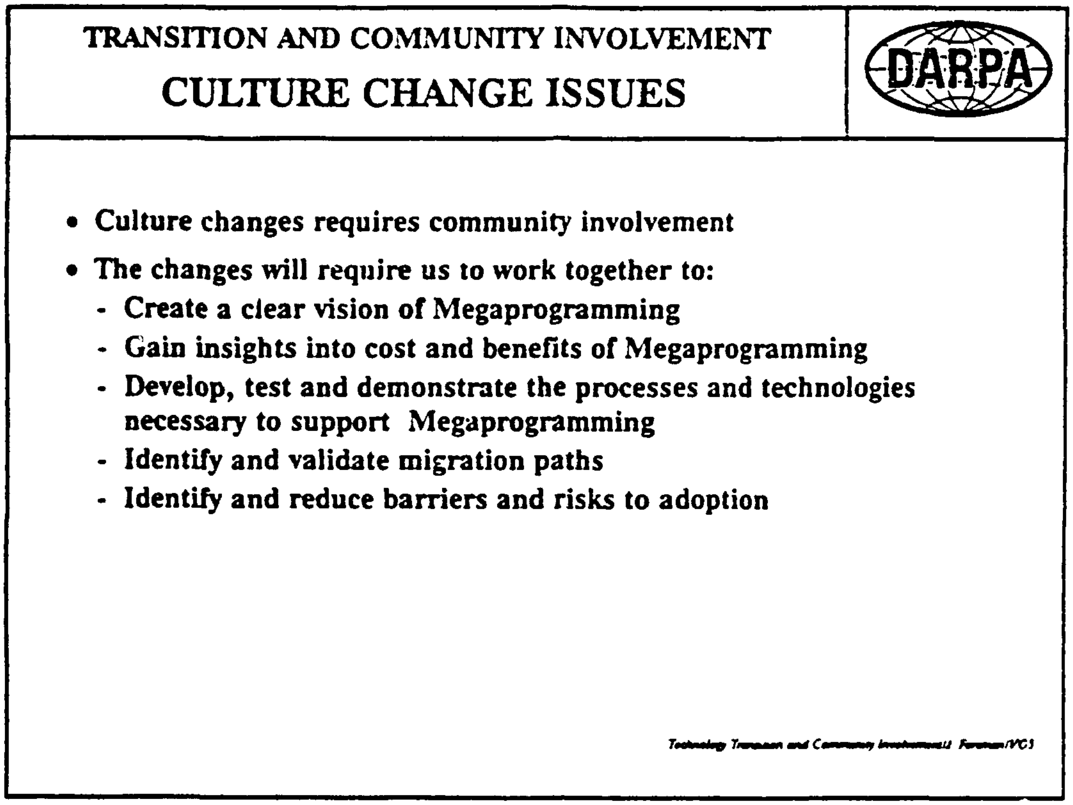
WORKING TOGETHER TOWARDS MEGAPROGRAMMING



- Work together to evolve point solutions to meet broad cultural and organizational needs
- Trial usage needed to evolve Megaprogramming processes and technologies
- Suppliers and customers need to work together to better understand and overcome the cultural impact and barriers to acceptance



Technology Transition and Community Involvement/J. Foreman/VG4

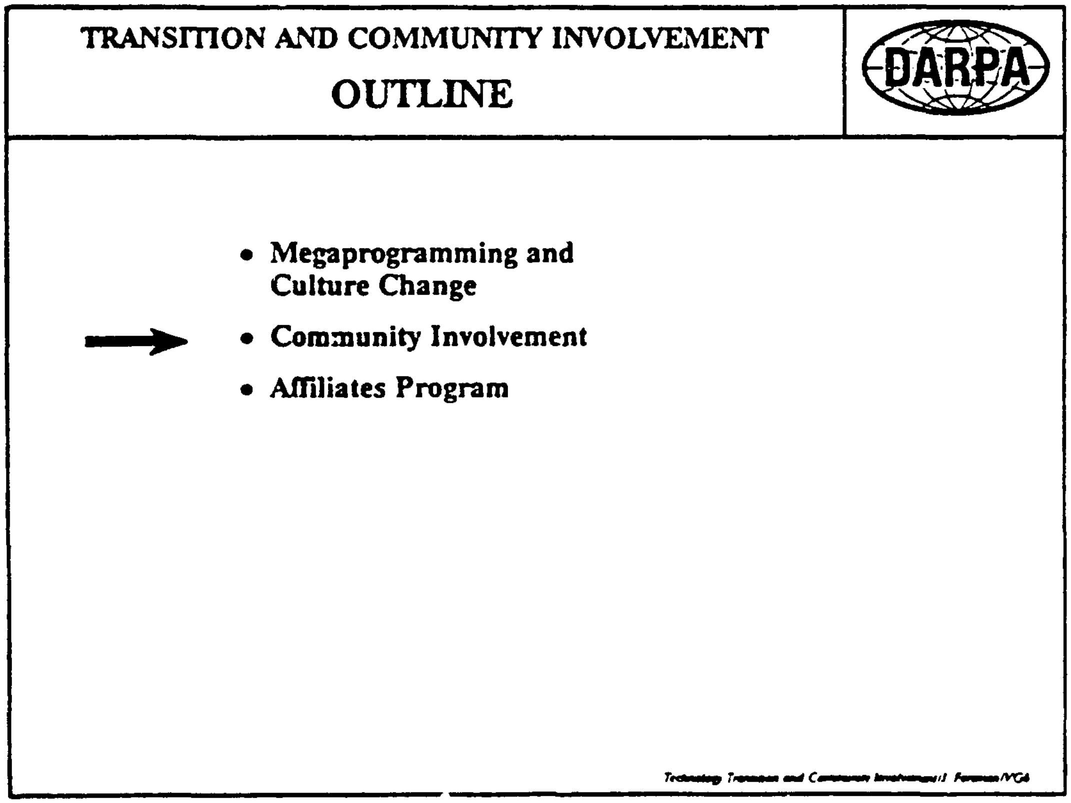


TRANSITION AND COMMUNITY INVOLVEMENT

CULTURE CHANGE ISSUES

- Culture changes requires community involvement
- The changes will require us to work together to:
  - Create a clear vision of Megaprogramming
  - Gain insights into cost and benefits of Megaprogramming
  - Develop, test and demonstrate the processes and technologies necessary to support Megaprogramming
  - Identify and validate migration paths
  - Identify and reduce barriers and risks to adoption

Technology Transition and Community Involvement/I Forum/VG3

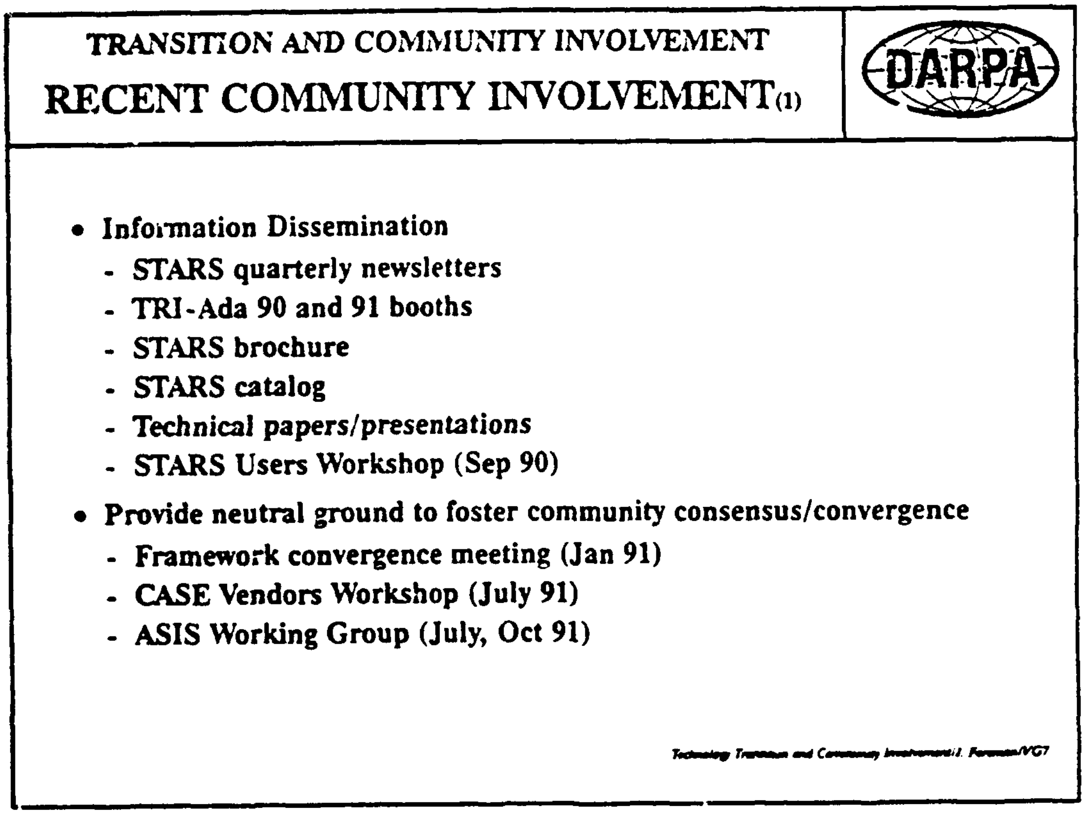


TRANSITION AND COMMUNITY INVOLVEMENT

OUTLINE

- Megaprogramming and Culture Change
- → Community Involvement
- Affiliates Program

Technology Transition and Community Involvement/I Forum/VG4



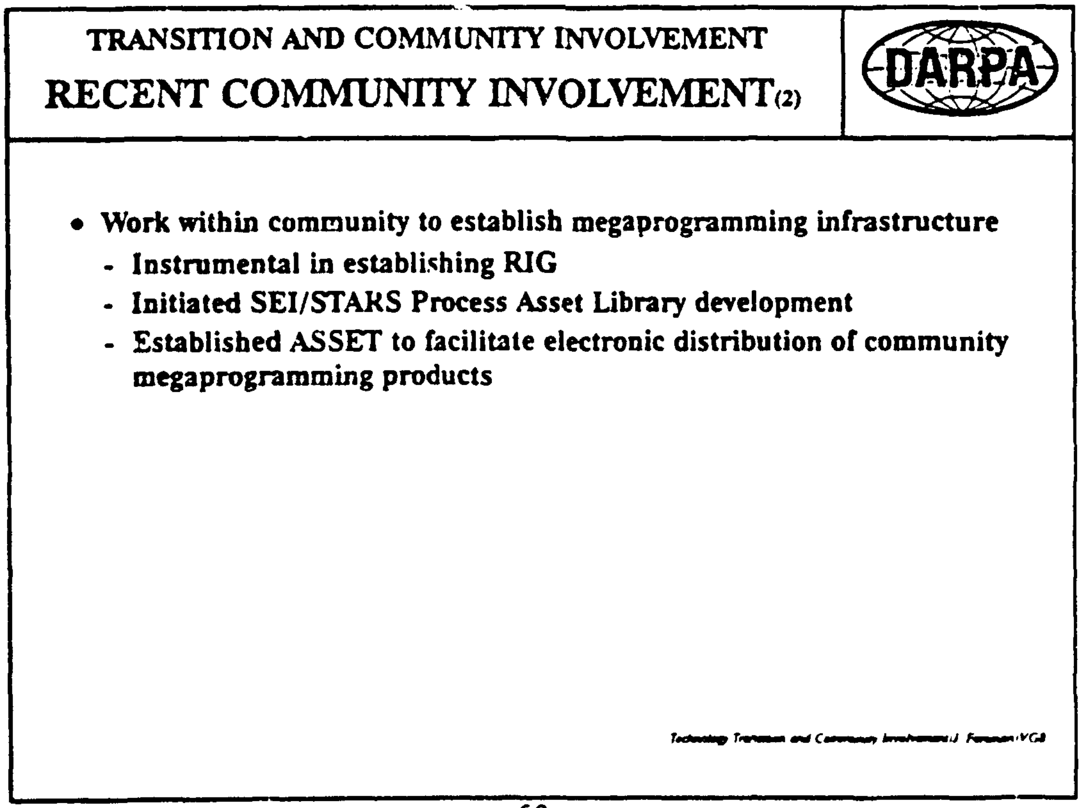
TRANSITION AND COMMUNITY INVOLVEMENT

RECENT COMMUNITY INVOLVEMENT(1)

DARPA

- Information Dissemination
  - STARS quarterly newsletters
  - TRI-Ada 90 and 91 booths
  - STARS brochure
  - STARS catalog
  - Technical papers/presentations
  - STARS Users Workshop (Sep 90)
- Provide neutral ground to foster community consensus/convergence
  - Framework convergence meeting (Jan 91)
  - CASE Vendors Workshop (July 91)
  - ASIS Working Group (July, Oct 91)

Technology Transition and Community Involvement/J. Foreman/VG7



TRANSITION AND COMMUNITY INVOLVEMENT

RECENT COMMUNITY INVOLVEMENT(2)

DARPA

- Work within community to establish megaprogramming infrastructure
  - Instrumental in establishing RIG
  - Initiated SEI/STARS Process Asset Library development
  - Established ASSET to facilitate electronic distribution of community megaprogramming products

Technology Transition and Community Involvement/J. Foreman/VG8

TRANSITION AND COMMUNITY INVOLVEMENT

CONTEXT FOR AFFILIATES PROGRAM

- STARS technology transition coordinator
- Packaging of interim products
- ASSET: Asset Source for Software Engineering Technology
- Commercialization
- Demonstration projects

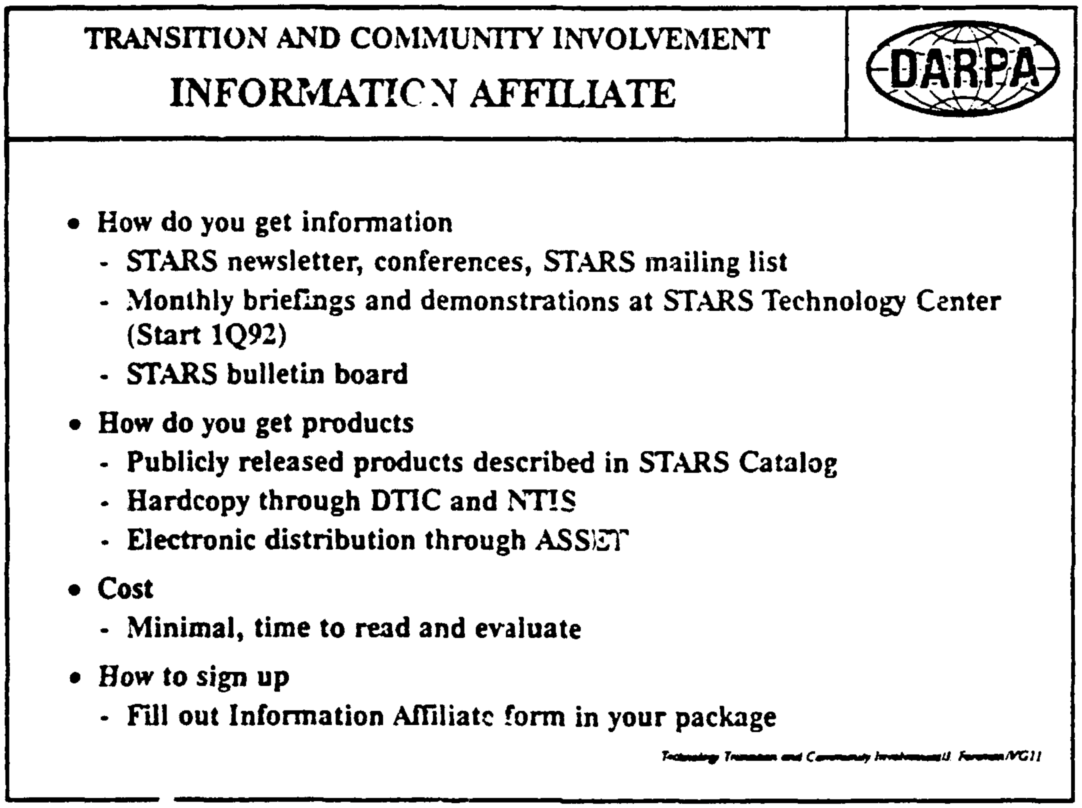
Technology Transition and Community Involvement/J. Foreman/VG9

TRANSITION AND COMMUNITY INVOLVEMENT

STARS AFFILIATES PROGRAM OVERVIEW

- STARS Information Affiliates
  - General community information dissemination
- STARS Technology Transfer Affiliates
  - Commitment of effort (STARS and Affiliates)
- STARS Prime Affiliates
  - Case-by-case basis between Affiliates and STARS prime(s)

Technology Transition and Community Involvement/J. Foreman/VG10



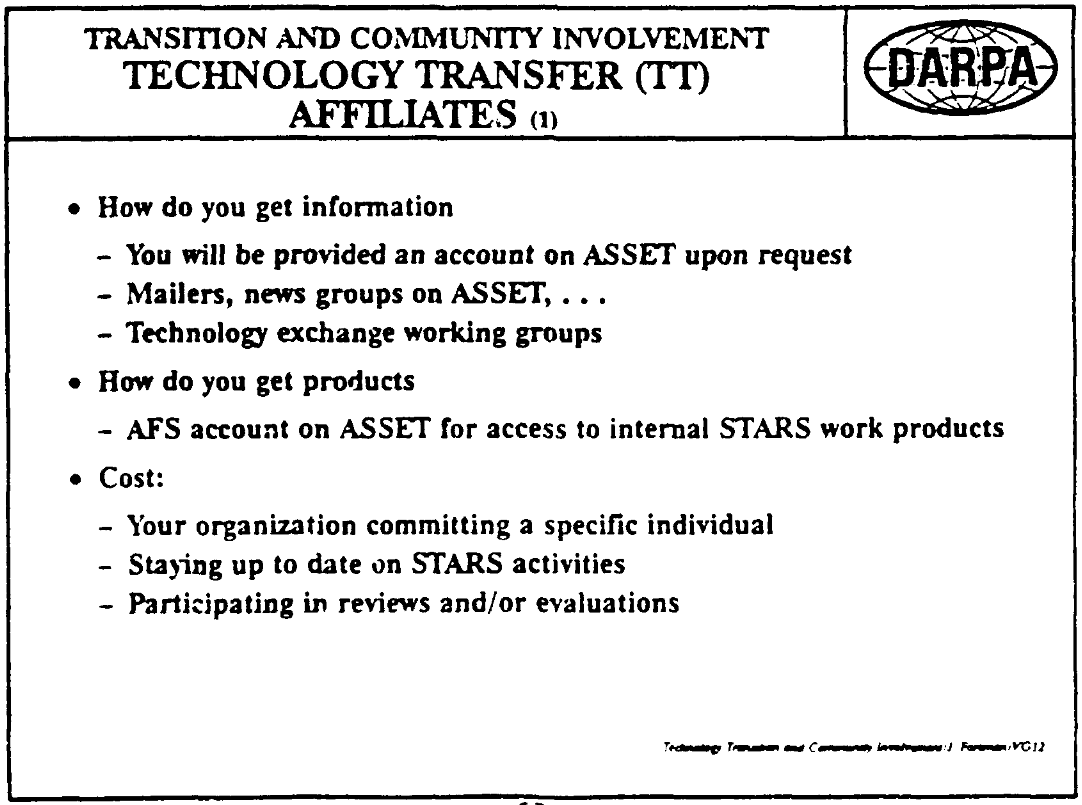
TRANSITION AND COMMUNITY INVOLVEMENT

INFORMATION AFFILIATE

DARPA

- How do you get information
  - STARS newsletter, conferences, STARS mailing list
  - Monthly briefings and demonstrations at STARS Technology Center (Start 1Q92)
  - STARS bulletin board
- How do you get products
  - Publicly released products described in STARS Catalog
  - Hardcopy through DTIC and NTIS
  - Electronic distribution through ASSET
- Cost
  - Minimal, time to read and evaluate
- How to sign up
  - Fill out Information Affiliate form in your package

Technology Transition and Community Involvement/J. Foreman/VG11



TRANSITION AND COMMUNITY INVOLVEMENT

TECHNOLOGY TRANSFER (TT) AFFILIATES (1)

DARPA

- How do you get information
  - You will be provided an account on ASSET upon request
  - Mailers, news groups on ASSET, . . .
  - Technology exchange working groups
- How do you get products
  - AFS account on ASSET for access to internal STARS work products
- Cost:
  - Your organization committing a specific individual
  - Staying up to date on STARS activities
  - Participating in reviews and/or evaluations

Technology Transition and Community Involvement/J. Foreman/VG12

TRANSITION AND COMMUNITY INVOLVEMENT

TECHNOLOGY TRANSFER (TT) AFFILIATES (2)

- How do you provide feedback
  - Providing review and feedback on STARS work products
  - Conducting alpha/beta test of products and providing lessons learned
  - Participating in joint technology experiments
  - Participating in TT Affiliates meetings
  - Future STARS Conferences become TT/Prime Affiliates Users meetings

Technology Transition and Community Involvement/J. Foreman/VG13

TRANSITION AND COMMUNITY INVOLVEMENT

TECHNOLOGY TRANSFER (TT) AFFILIATES (3)

- 2 way technology transition
  - Broaden exposure for your technology to technically knowledgeable community
  - Publicize your products supporting megaprogramming in ASSET
  - Potential use of your domain specific assets on a demonstration project
- How to sign up
  - Number of TT Affiliates Limited
  - Participate in appropriate technology area sessions this evening
  - Submit TT Affiliates questionnaire and information forms

Technology Transition and Community Involvement/J. Foreman/VG14

TRANSITION AND COMMUNITY INVOLVEMENT

STARS PRIME AFFILIATES

DARPA

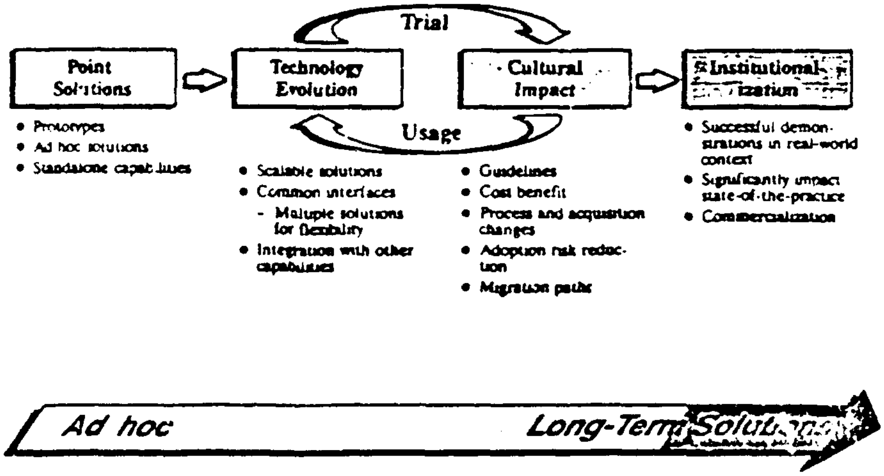
- Approach: extended TT affiliate that also works directly with STARS prime(s). Examples:
  - Co-development of SW engineering capabilities
  - Joint commercialization effort
  - Prime may provide access to SEE testbeds for selected integration experiments
- How to sign up
  - Negotiation with individual STARS primes on a case-by-case basis
  - Contact Boeing, Unisys, IBM, program managers

Technology Transition and Community Involvement/J. Foreman/VG15

TRANSITION AND COMMUNITY INVOLVEMENT

JOIN US IN TRANSITION TO MEGAPROGRAMMING

DARPA



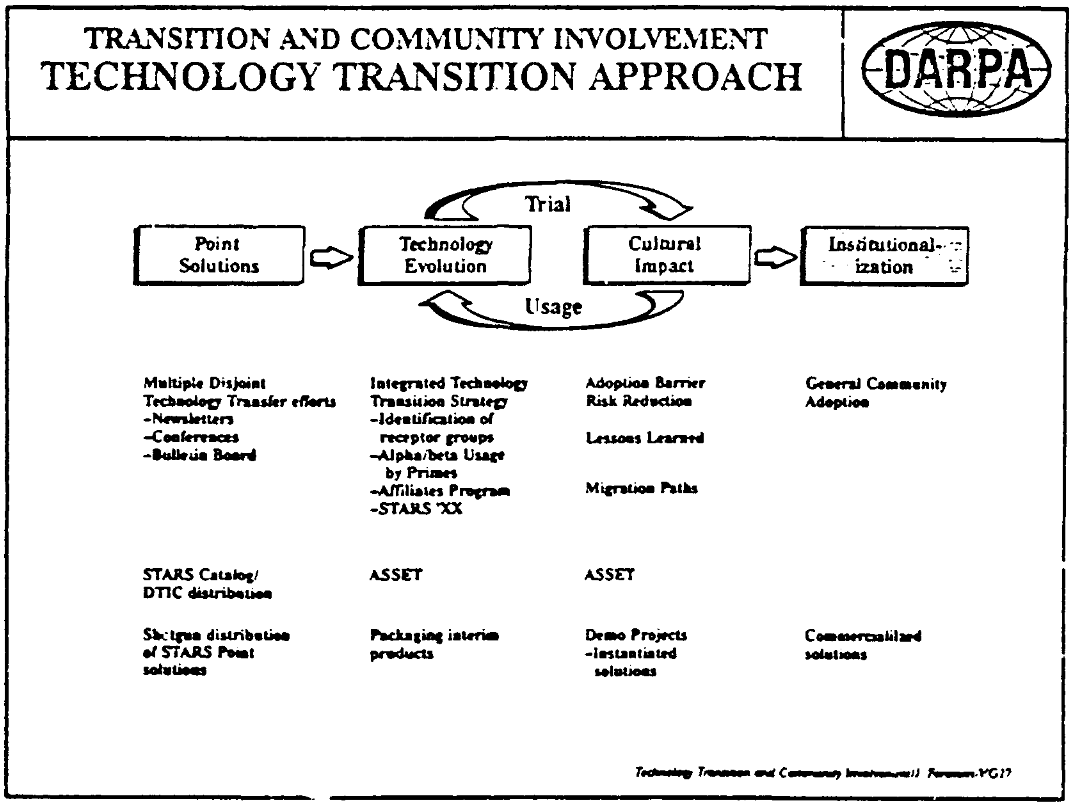
- Prototypes
- Ad hoc solutions
- Standalone capabilities

- Scalable solutions
- Common interfaces
  - Multiple solutions for flexibility
- Integration with other capabilities

- Guidelines
- Cost benefit
- Process and acquisition changes
- Adoption risk reduction
- Migration paths

- Successful demonstrations in real-world context
- Significantly impact state-of-the-practice
- Commercialization

Technology Transition and Community Involvement/J. Foreman/VG16



TRANSITION AND COMMUNITY INVOLVEMENT

TECHNOLOGY TRANSITION APPROACH

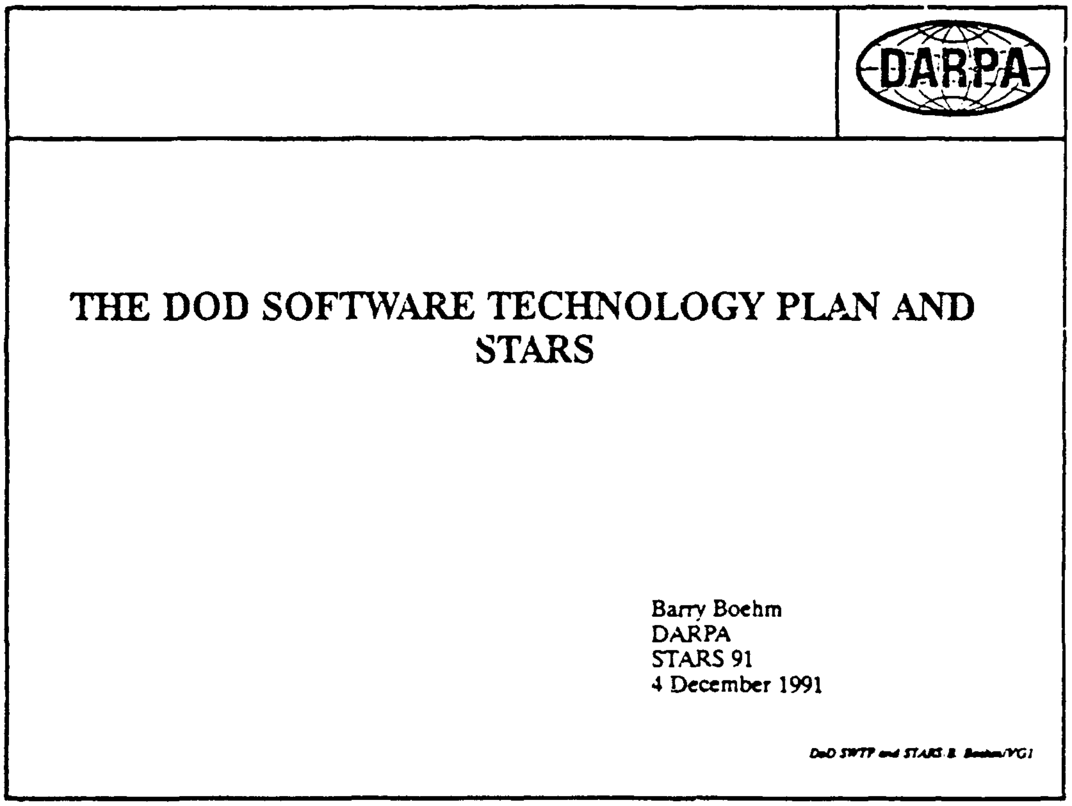
DARPA

Trial

Usage

Point Solutions	Technology Evolution	Cultural Impact	Institutional-ization
Multiple Disjoint Technology Transfer efforts -Newsletters -Conferences -Bulletin Board	Integrated Technology Transition Strategy -Identification of receptor groups -Alpha/beta Usage by Primes -Affiliates Program -STARS 'XX	Adoption Barrier Risk Reduction Lessons Learned Migration Paths	General Community Adoption
STARS Catalog/ DTIC distribution	ASSET	ASSET	
Shotgun distribution of STARS Point solutions	Packaging interim products	Demo Projects -Instantiated solutions	Commercialilized solutions

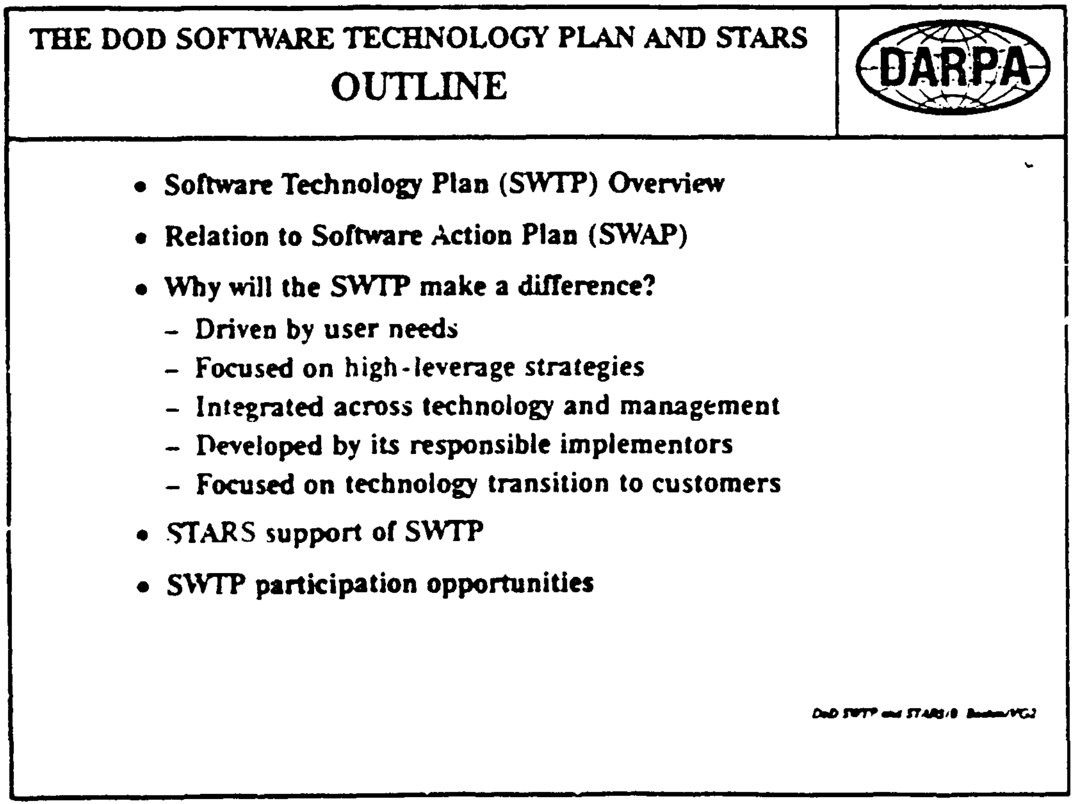
Technology Transition and Community Involvement [illegible] Forum-YG17



THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

Barry Boehm
DARPA
STARS 91
4 December 1991

DoD SWTP and STARS B. Boehm/VG1



THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS
OUTLINE

- Software Technology Plan (SWTP) Overview
- Relation to Software Action Plan (SWAP)
- Why will the SWTP make a difference?
  - Driven by user needs
  - Focused on high-leverage strategies
  - Integrated across technology and management
  - Developed by its responsible implementors
  - Focused on technology transition to customers
- STARS support of SWTP
- SWTP participation opportunities

DoD SWTP and STARS/B. Boehm/VG2

THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

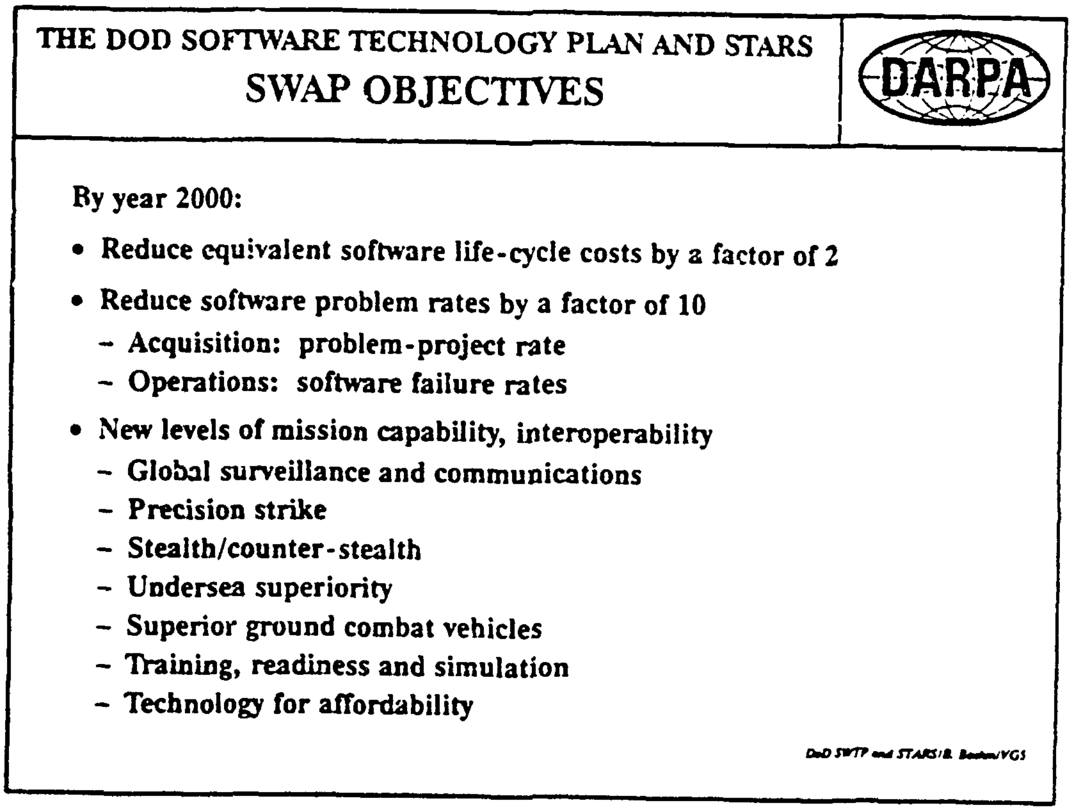
OVERVIEW OF DOD SOFTWARE TECHNOLOGY PLAN

- Part of DDR&E Software Action Plan
- Scope includes all DoD software technology base, FY 1992-2007
  - 6.1, 6.2, 6.3A Software Science and Technology Programs
- Being created by DoD software technology program managers
  - With extensive external review cycle
- Two investment program levels defined
  - Current program: flat out-year budgets
  - Achievable program: increase to cover technology opportunities
- ROI analysis performed to determine whether investment justified

THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

SWAP CONTEXT

- Software is key to smart, flexible DoD forces
  - Desert Storm: PGM's, Patriot, surveillance, logistics
- Software is difficult to acquire and support
  - USAF/ESD: 70% of problem projects due to software
- Software cost increasing from current $24-32B/year level



THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

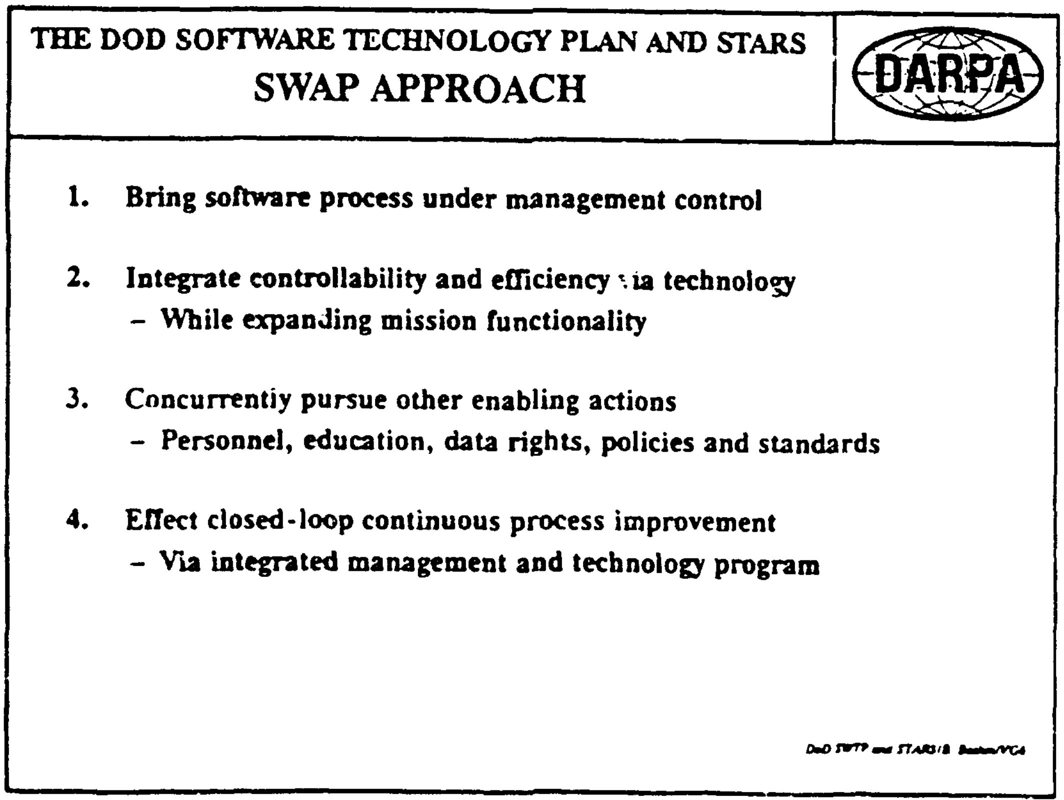
SWAP OBJECTIVES

DARPA

By year 2000:

- Reduce equivalent software life-cycle costs by a factor of 2
- Reduce software problem rates by a factor of 10
  - Acquisition: problem-project rate
  - Operations: software failure rates
- New levels of mission capability, interoperability
  - Global surveillance and communications
  - Precision strike
  - Stealth/counter-stealth
  - Undersea superiority
  - Superior ground combat vehicles
  - Training, readiness and simulation
  - Technology for affordability

DoD SWTP and STARS/B. Boehm/VG5



THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

SWAP APPROACH

DARPA

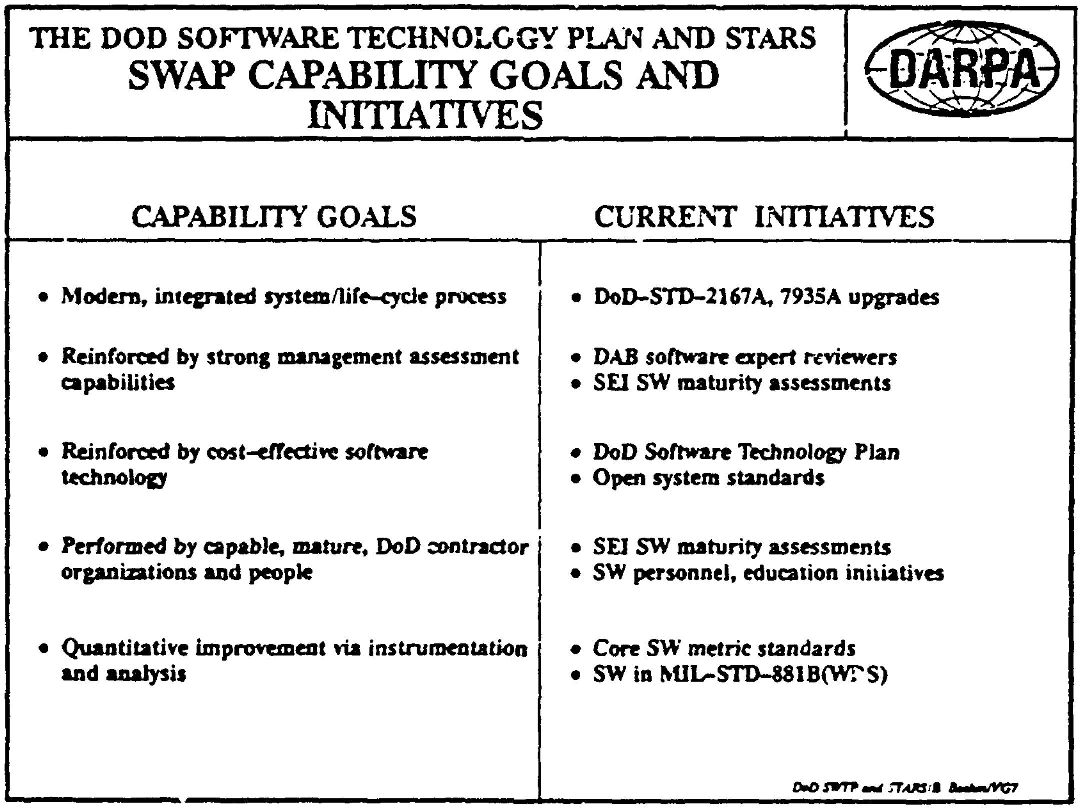
1. Bring software process under management control

2. Integrate controllability and efficiency via technology
   - While expanding mission functionality

3. Concurrentiy pursue other enabling actions
   - Personnel, education, data rights, policies and standards

4. Effect closed-loop continuous process improvement
   - Via integrated management and technology program

DoD SWTP and STARS/B. Boehm/VG6

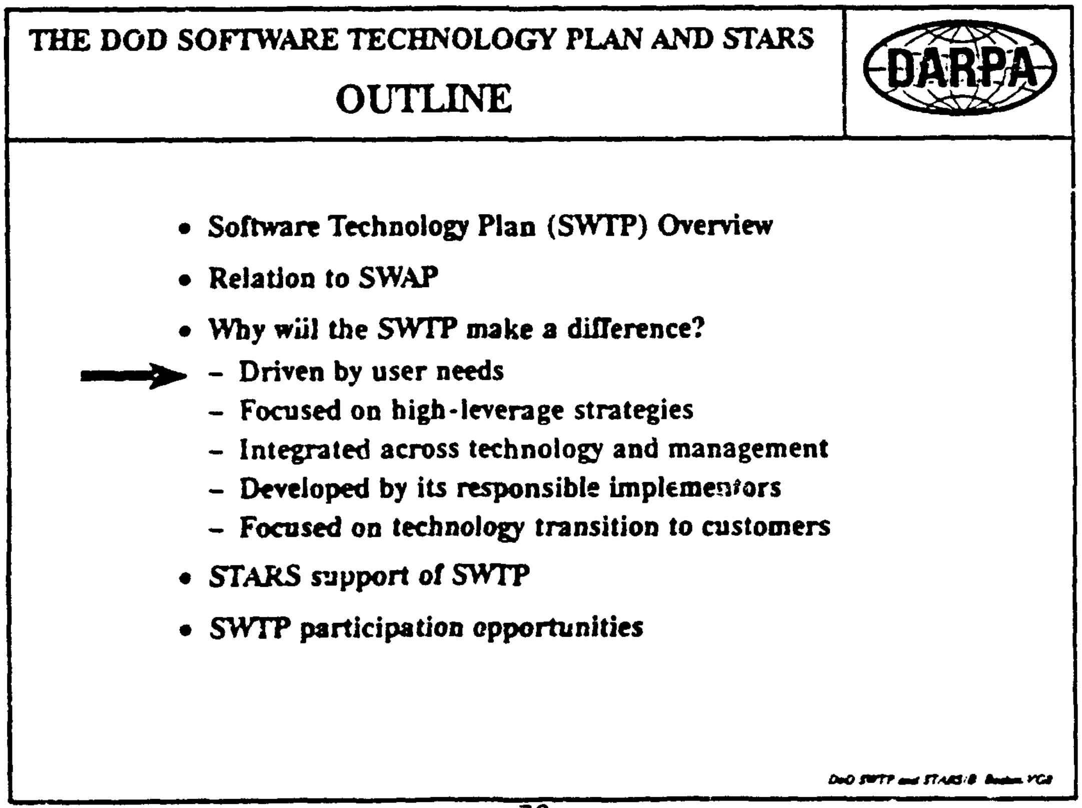


THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

SWAP CAPABILITY GOALS AND INITIATIVES

CAPABILITY GOALS	CURRENT INITIATIVES
• Modern, integrated system/life-cycle process	• DoD-STD-2167A, 7935A upgrades
• Reinforced by strong management assessment capabilities	• DAB software expert reviewers • SEI SW maturity assessments
• Reinforced by cost-effective software technology	• DoD Software Technology Plan • Open system standards
• Performed by capable, mature, DoD contractor organizations and people	• SEI SW maturity assessments • SW personnel, education initiatives
• Quantitative improvement via instrumentation and analysis	• Core SW metric standards • SW in MIL-STD-881B(WBS)

DoD SWTP and STARS:B Boehm/VG7

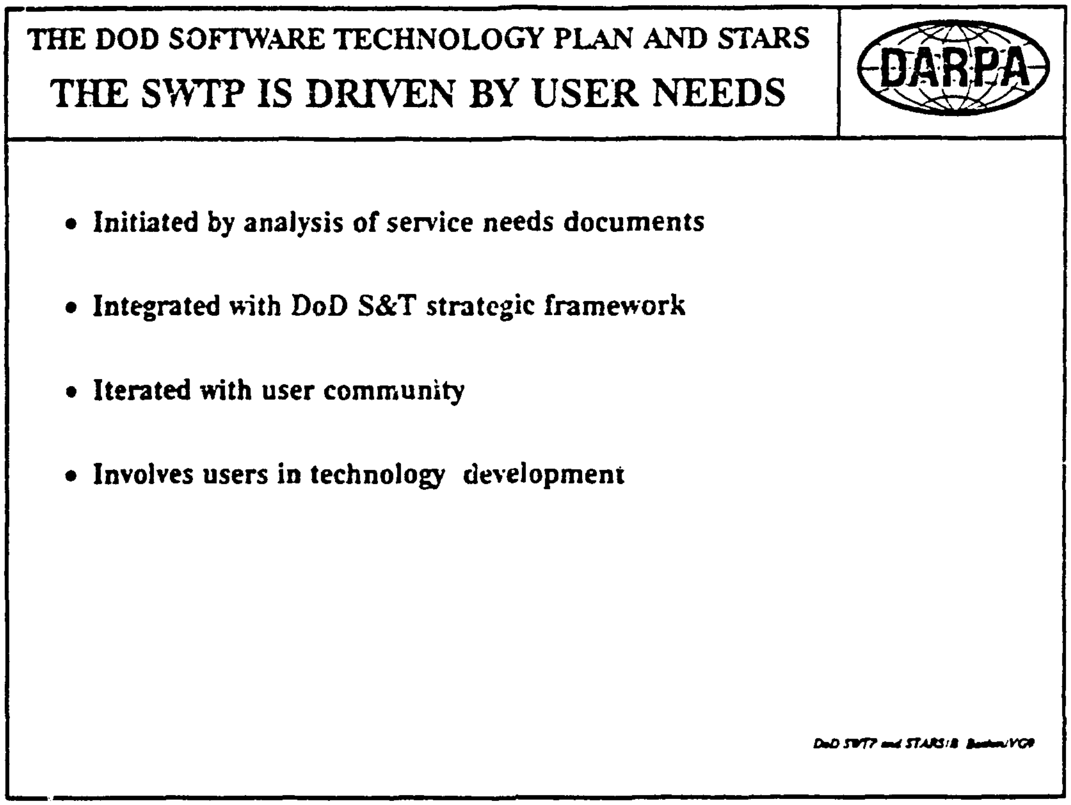


THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

OUTLINE

- Software Technology Plan (SWTP) Overview
- Relation to SWAP
- Why will the SWTP make a difference?
  - → Driven by user needs
  - Focused on high-leverage strategies
  - Integrated across technology and management
  - Developed by its responsible implementors
  - Focused on technology transition to customers
- STARS support of SWTP
- SWTP participation opportunities

DoD SWTP and STARS:B Boehm VG8

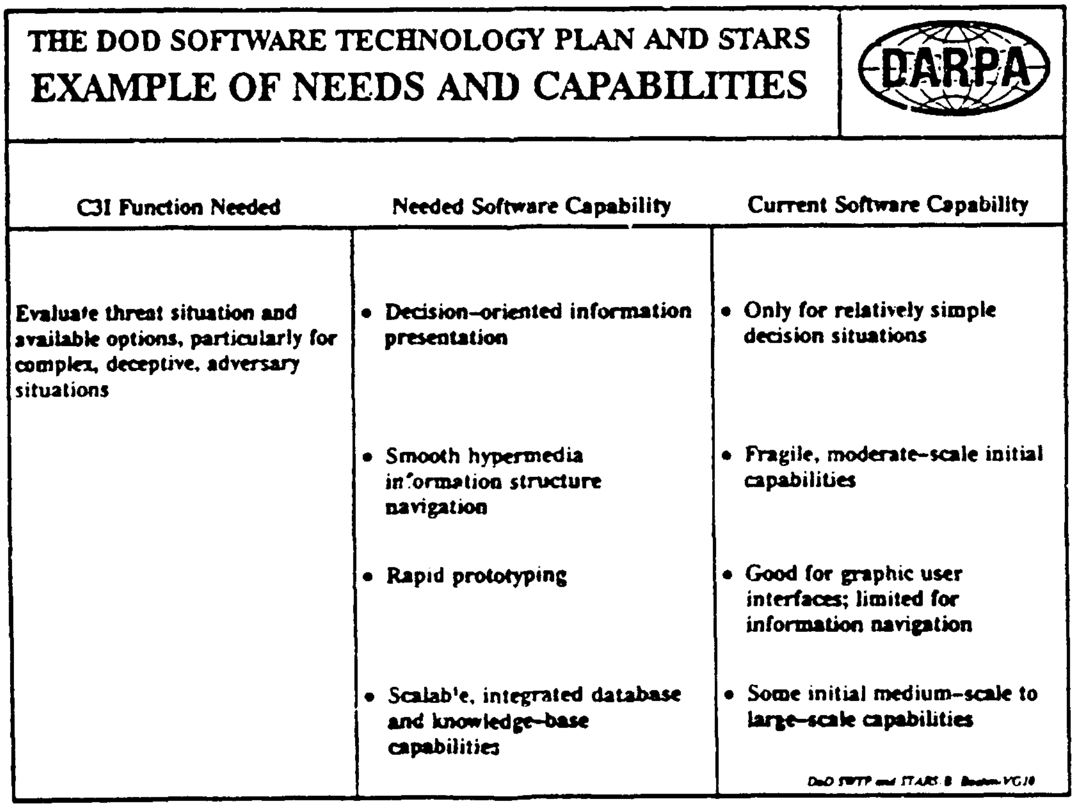


THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

THE SWTP IS DRIVEN BY USER NEEDS

- Initiated by analysis of service needs documents
- Integrated with DoD S&T strategic framework
- Iterated with user community
- Involves users in technology development

DoD SWTP and STARS/8 Boston/VG9

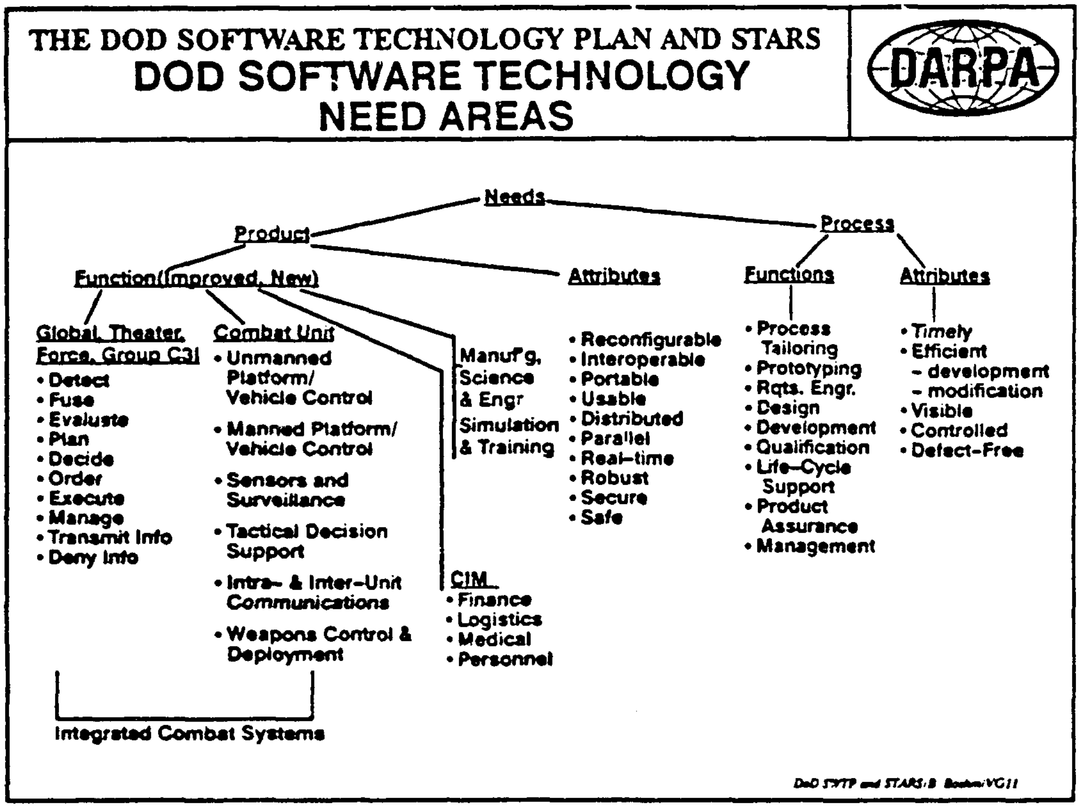


THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

EXAMPLE OF NEEDS AND CAPABILITIES

C3I Function Needed	Needed Software Capability	Current Software Capability
Evaluate threat situation and available options, particularly for complex, deceptive, adversary situations	• Decision–oriented information presentation	• Only for relatively simple decision situations
	• Smooth hypermedia in'ormation structure navigation	• Fragile, moderate–scale initial capabilities
	• Rapid prototyping	• Good for graphic user interfaces; limited for information navigation
	• Scalab'e, integrated database and knowledge–base capabilities	• Some initial medium–scale to large–scale capabilities

DoD SWTP and STARS 8 Boston-VG10



THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

DOD SOFTWARE TECHNOLOGY NEED AREAS

DARPA

Needs

Product

Function(Improved, New)

Global, Theater, Force, Group C3I
- Detect
- Fuse
- Evaluate
- Plan
- Decide
- Order
- Execute
- Manage
- Transmit Info
- Deny Info

Combat Unit
- Unmanned Platform/ Vehicle Control
- Manned Platform/ Vehicle Control
- Sensors and Surveillance
- Tactical Decision Support
- Intra- & Inter-Unit Communications
- Weapons Control & Deployment

Integrated Combat Systems

Manuf'g, Science & Engr Simulation & Training

CIM
- Finance
- Logistics
- Medical
- Personnel

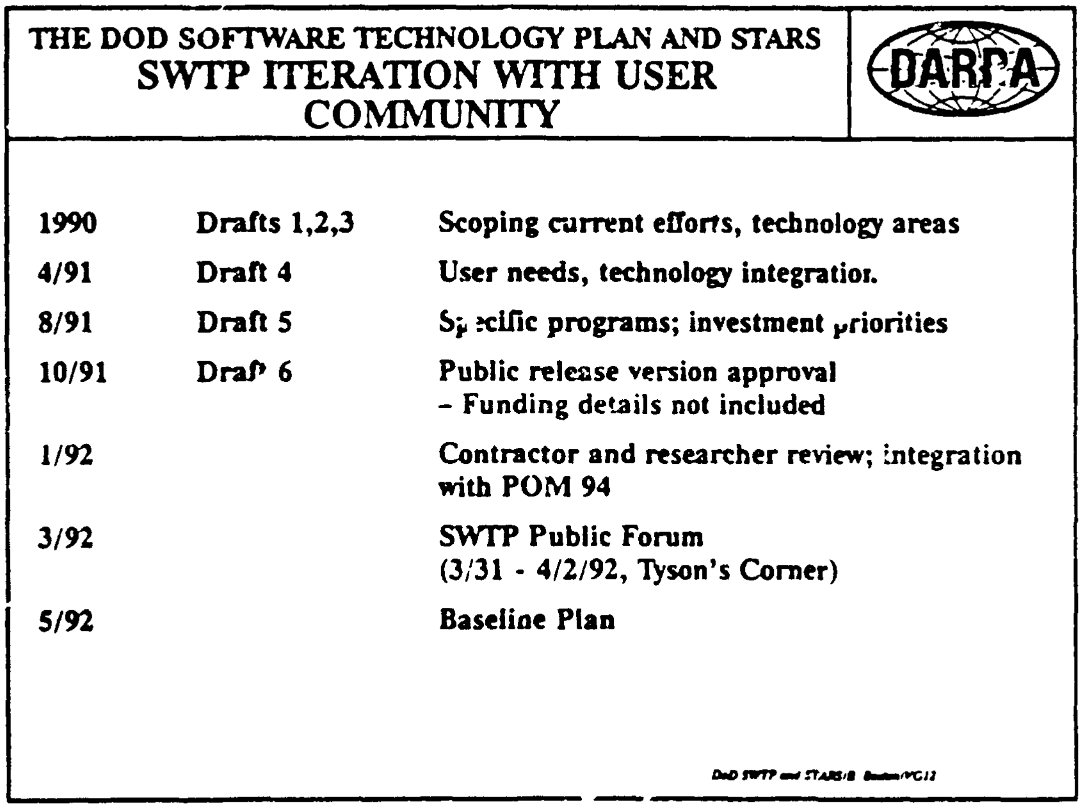
Attributes
- Reconfigurable
- Interoperable
- Portable
- Usable
- Distributed
- Parallel
- Real-time
- Robust
- Secure
- Safe

Process

Functions
- Process Tailoring
- Prototyping
- Rqts. Engr.
- Design
- Development
- Qualification
- Life-Cycle Support
- Product Assurance
- Management

Attributes
- Timely
- Efficient
  - development
  - modification
- Visible
- Controlled
- Defect-Free

DoD SWTP and STARS/8 Boehm/VG11



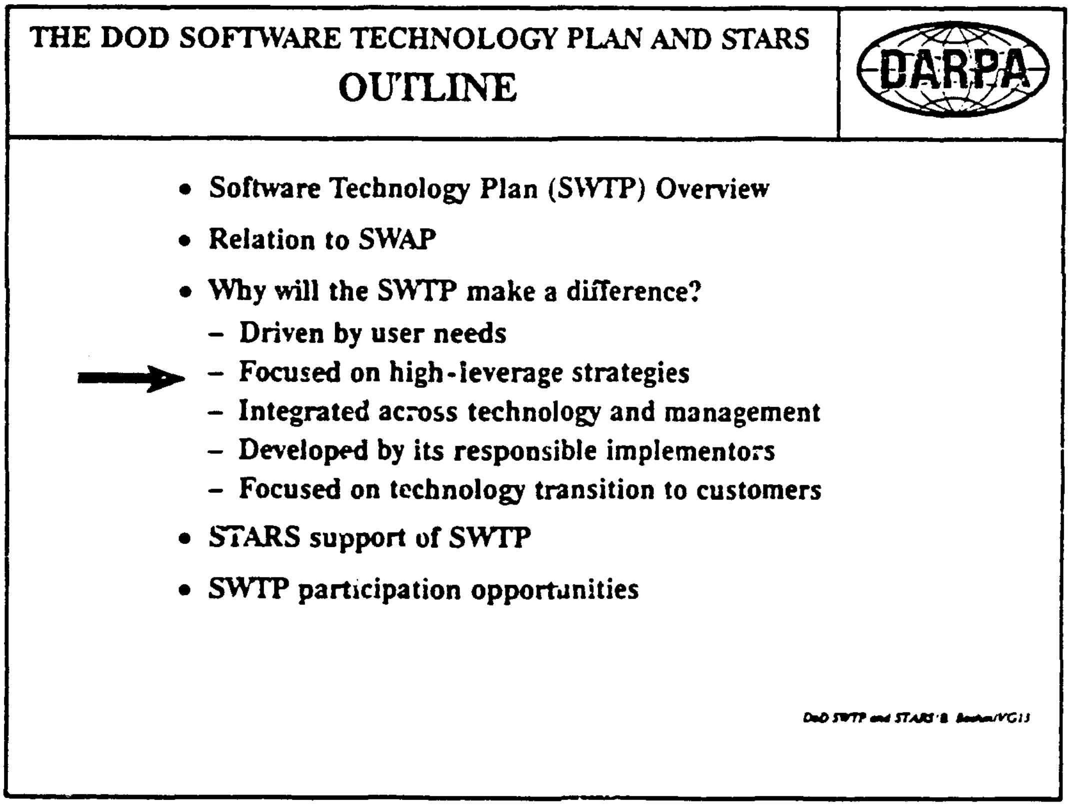
THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

SWTP ITERATION WITH USER COMMUNITY

DARPA

1990	Drafts 1,2,3	Scoping current efforts, technology areas
4/91	Draft 4	User needs, technology integratior.
8/91	Draft 5	Sṗ ecific programs; investment priorities
10/91	Draf' 6	Public release version approval - Funding details not included
1/92		Contractor and researcher review; integration with POM 94
3/92		SWTP Public Forum (3/31 - 4/2/92, Tyson's Corner)
5/92		Baseline Plan

DoD SWTP and STARS/8 Boehm/VG12

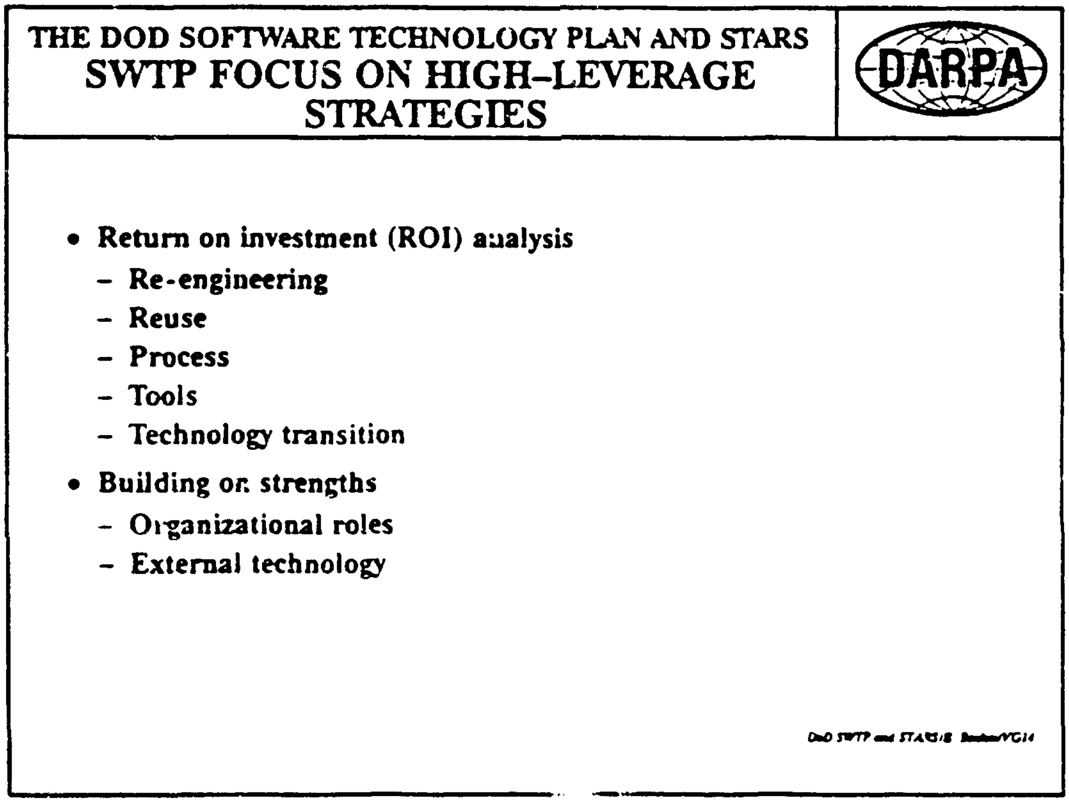


THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

OUTLINE

- Software Technology Plan (SWTP) Overview
- Relation to SWAP
- Why will the SWTP make a difference?
  - Driven by user needs
  - → Focused on high-leverage strategies
  - Integrated across technology and management
  - Developed by its responsible implementors
  - Focused on technology transition to customers
- STARS support of SWTP
- SWTP participation opportunities

DoD SWTP and STARS'8 Boehm/VG13



THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

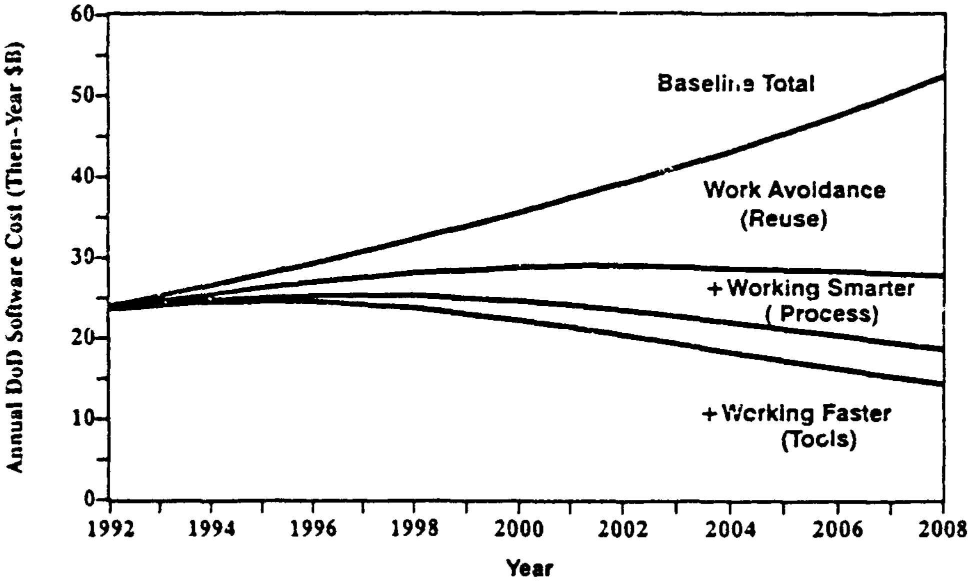
SWTP FOCUS ON HIGH-LEVERAGE STRATEGIES

- Return on investment (ROI) analysis
  - Re-engineering
  - Reuse
  - Process
  - Tools
  - Technology transition
- Building on strengths
  - Organizational roles
  - External technology

DoD SWTP and STARS'8 Boehm/VG14

STARS AND MEGAPROGRAMMING

PROGRAM RESULTS BY SOURCE OF SAVINGS



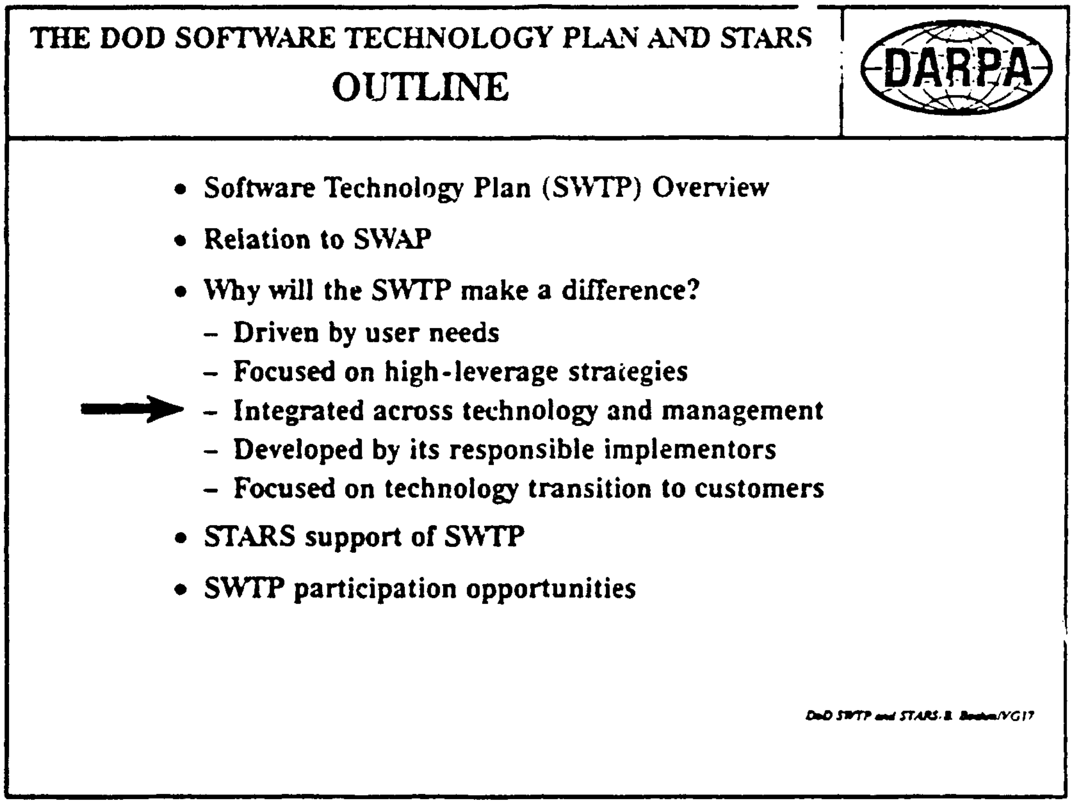
DoD SWTP and STARS/B. Boehm/VG15

STARS AND MEGAPROGRAMMING

BUILDING ON STRENGTHS: EXTEκNAL TECHNOLOGY

- Where possible, get DoD technology commercialized
- Or, get commercial technology DoD-ized
  - Accommodating Ada, embedded real-time, high assurance, high performance software
- This makes the "Iron Law of Software Maintenance" affordable
  - "For every $1 you spend on Software Product Development, you will spend at least $2 on its maintenance"
  - Commercialization spreads maintenance costs over much larger user base than DoD

DoD SWTP and STARS/B. Boehm/VG16

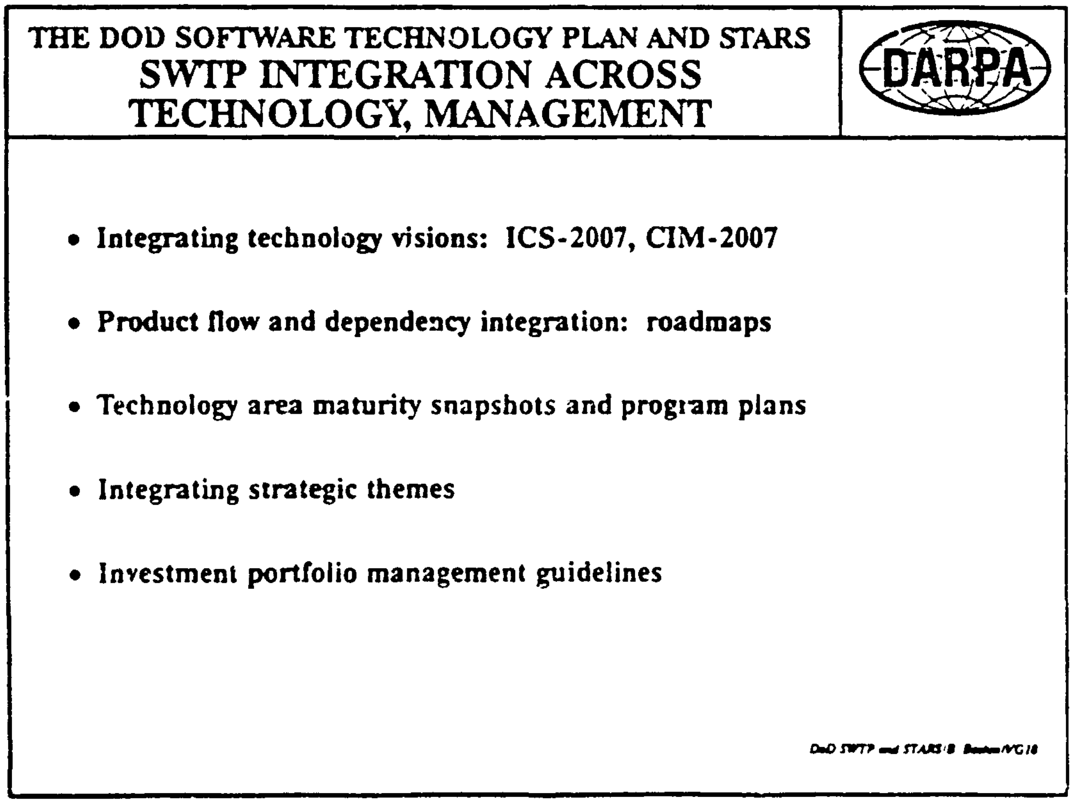


THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

OUTLINE

- Software Technology Plan (SWTP) Overview
- Relation to SWAP
- Why will the SWTP make a difference?
  - Driven by user needs
  - Focused on high-leverage straiegies
  - → Integrated across technology and management
  - Developed by its responsible implementors
  - Focused on technology transition to customers
- STARS support of SWTP
- SWTP participation opportunities

DoD SWTP and STARS/B Boehm/VG17



THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

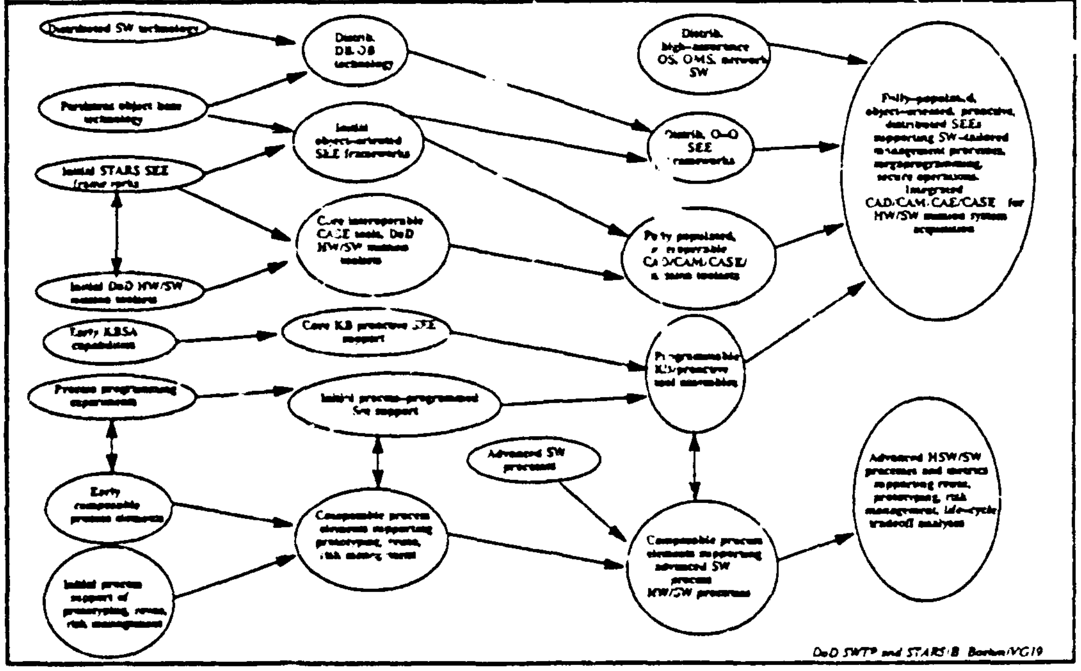
SWTP INTEGRATION ACROSS TECHNOLOGY, MANAGEMENT

- Integrating technology visions: ICS-2007, CIM-2007
- Product flow and dependency integration: roadmaps
- Technology area maturity snapshots and progıam plans
- Integrating strategic themes
- Investment portfolio management guidelines

DoD SWTP and STARS/B Boehm/VG18

THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

PORTION OF SWTP PROCESS TECHNOLOGY ROADMAP


Distributed SW technology
Distrib. DB, OS technology
Persistent object base technology
Initial object-oriented SEE frameworks
Initial STARS SEE frameworks
Core interoperable CASE tools, DoD HW/SW mission toolsets
Initial DoD HW/SW mission toolsets
Early KBSA capabilities
Core KB process SEE support
Process programming experiments
Initial process-programmed SW support
Early composable process elements
Initial process support of prototyping, reuse, risk management
Composable process elements supporting prototyping, reuse, risk mgmt, [illegible]
Advanced SW processes
Distrib. high-assurance OS, OMS, network SW
Distrib. O-O SEE frameworks
Partly populated, interoperable CAD/CAM/CASE/mission toolsets
Programmable KB process tool assembly
Composable process elements supporting advanced SW process HW/SW processes
Fully-populated, object-oriented, proactive, distributed SEEs supporting SW-tailored management processes, megaprogramming, secure operations. Integrated CAD/CAM/CAE/CASE for HW/SW mission system acquisition
Advanced HSW/SW processes and metrics supporting reuse, prototyping, risk management, life-cycle tradeoff analyses

DoD SWTP and STARS/B Boehm/VG19

THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

SWTP INTEGRATING STRATEGIC THEMES

- Megaprogramming
- High-level re-engineering
- Process support and technology/management synergy
- Commercial technology leverage
- Integrating artificial intelligence and software engineering

DoD SWTP and STARS/B Boehm/VG20

THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

OUTLINE

- Software Technology Plan (SWTP) Overview
- Relation to SWAP
- Why will the SWTP make a difference?
  - Driven by usei needs
  - Focused on high-leverage strategies
  - Integrated across technology and management
  - → Developed by its responsible implementors
  - Focused on technology transition to customers
- STARS support of SWTP
- SWTP participation opportunities

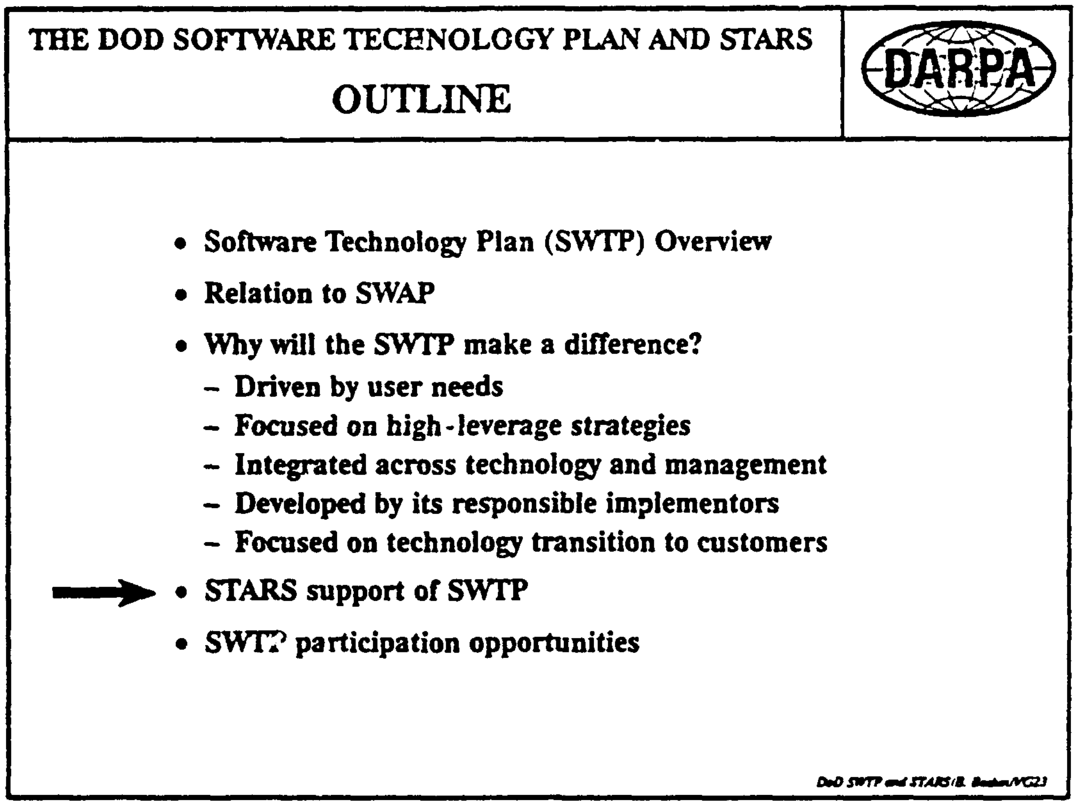
DoD SWTP and STARS/B Boehm/VG21

THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

SWTP TECHNOLOGY TRANSITION INITIATIVES

- User involvement in technology development
- Joint government/industry/university projects
- Process maturity assessments
- Mid-life cost-effectiveness reviews
- Stimulate technology advocates and receptors
- Closed-loop IR&D process
- Annual DoD Software Technology Conference
- Open systems

DoD SWTP and STARS/B Boehm/VG22

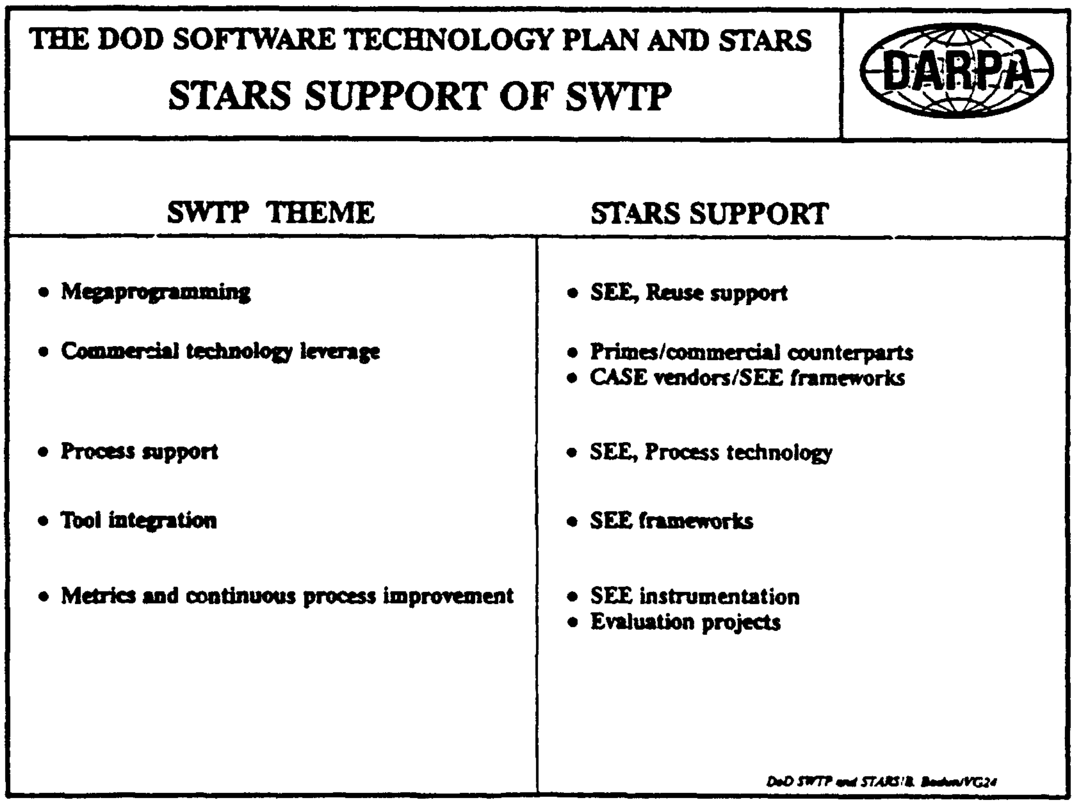


THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

OUTLINE

- Software Technology Plan (SWTP) Overview
- Relation to SWAP
- Why will the SWTP make a difference?
  - Driven by user needs
  - Focused on high-leverage strategies
  - Integrated across technology and management
  - Developed by its responsible implementors
  - Focused on technology transition to customers
- STARS support of SWTP
- SWTP participation opportunities

DoD SWTP and STARS/B. Boehm/VG23

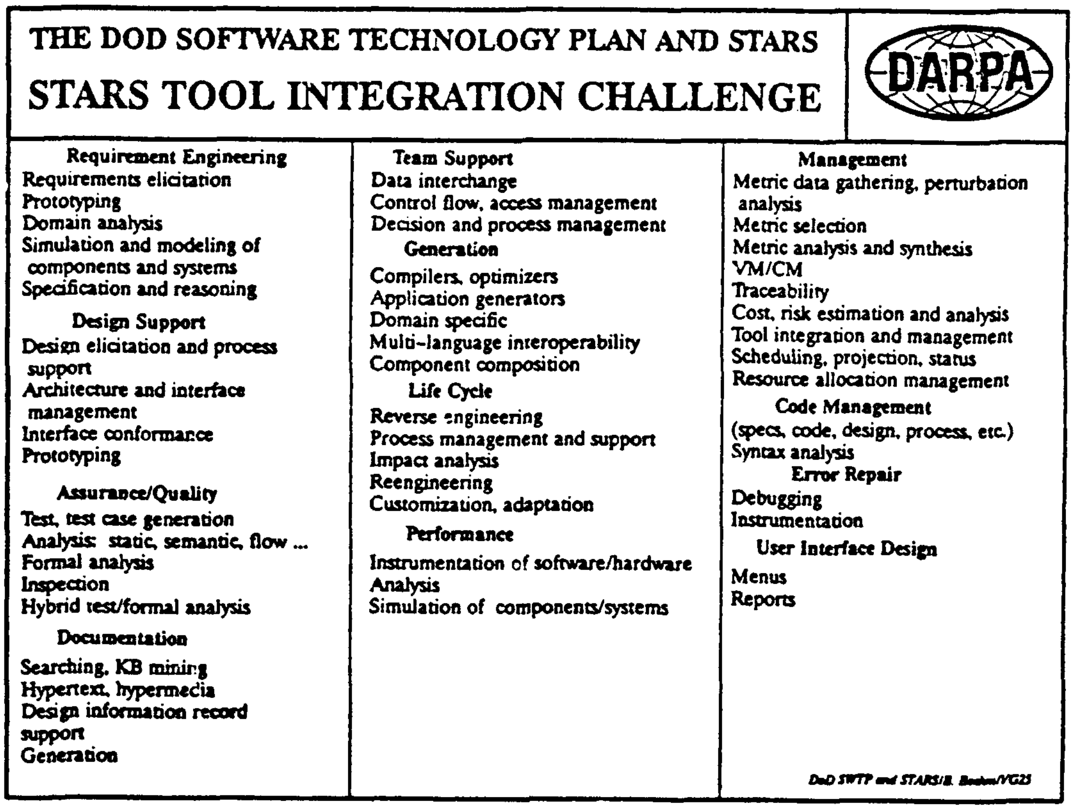


THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

STARS SUPPORT OF SWTP

SWTP THEME	STARS SUPPORT
• Megaprogramming	• SEE, Reuse support
• Commercial technology leverage	• Primes/commercial counterparts • CASE vendors/SEE frameworks
• Process support	• SEE, Process technology
• Tool integration	• SEE frameworks
• Metrics and continuous process improvement	• SEE instrumentation • Evaluation projects

DoD SWTP and STARS/B. Boehm/VG24



THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

STARS TOOL INTEGRATION CHALLENGE

DARPA

Requirement Engineering
Requirements elicitation
Prototyping
Domain analysis
Simulation and modeling of components and systems
Specification and reasoning

Design Support
Design elicitation and process support
Architecture and interface management
Interface conformance
Prototyping

Assurance/Quality
Test, test case generation
Analysis: static, semantic, flow ...
Formal analysis
Inspection
Hybrid test/formal analysis

Documentation
Searching, KB mining
Hypertext, hypermedia
Design information record support
Generation

Team Support
Data interchange
Control flow, access management
Decision and process management

Generation
Compilers, optimizers
Application generators
Domain specific
Multi-language interoperability
Component composition

Life Cycle
Reverse engineering
Process management and support
Impact analysis
Reengineering
Customization, adaptation

Performance
Instrumentation of software/hardware
Analysis
Simulation of components/systems

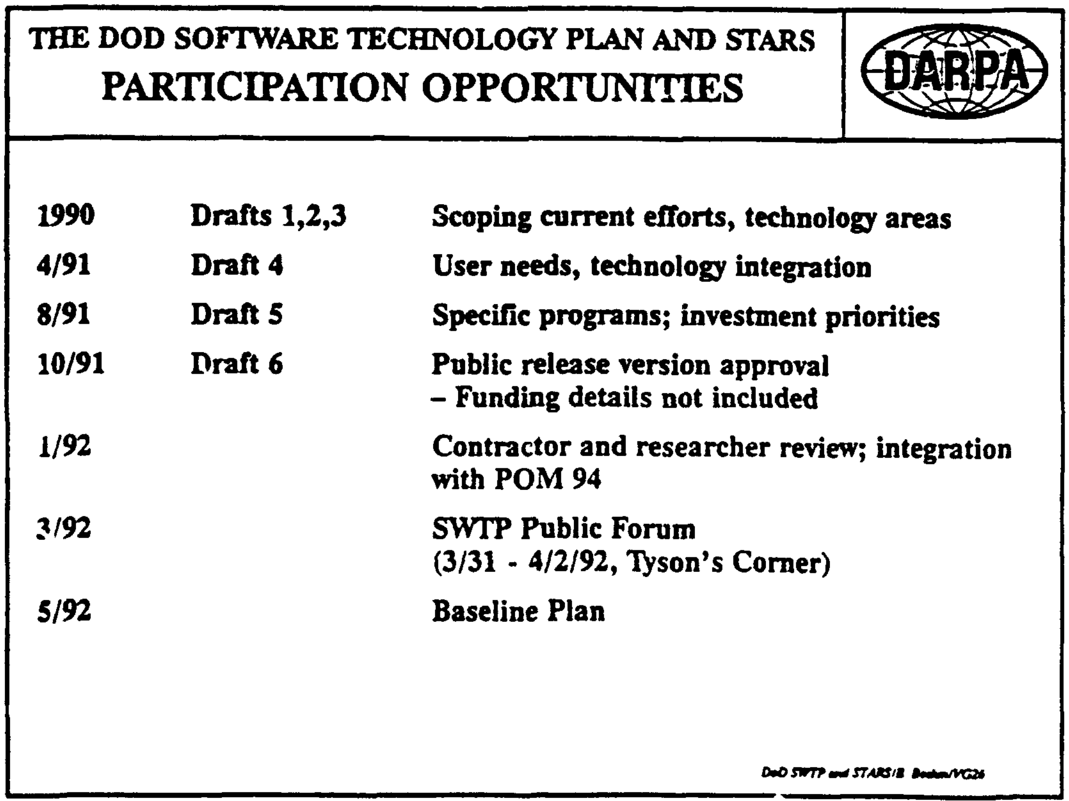
Management
Metric data gathering, perturbation analysis
Metric selection
Metric analysis and synthesis
VM/CM
Traceability
Cost, risk estimation and analysis
Tool integration and management
Scheduling, projection, status
Resource allocation management

Code Management
(specs, code, design, process, etc.)
Syntax analysis

Error Repair
Debugging
Instrumentation

User Interface Design
Menus
Reports

DoD SWTP and STARS/B. Boehm/VG25



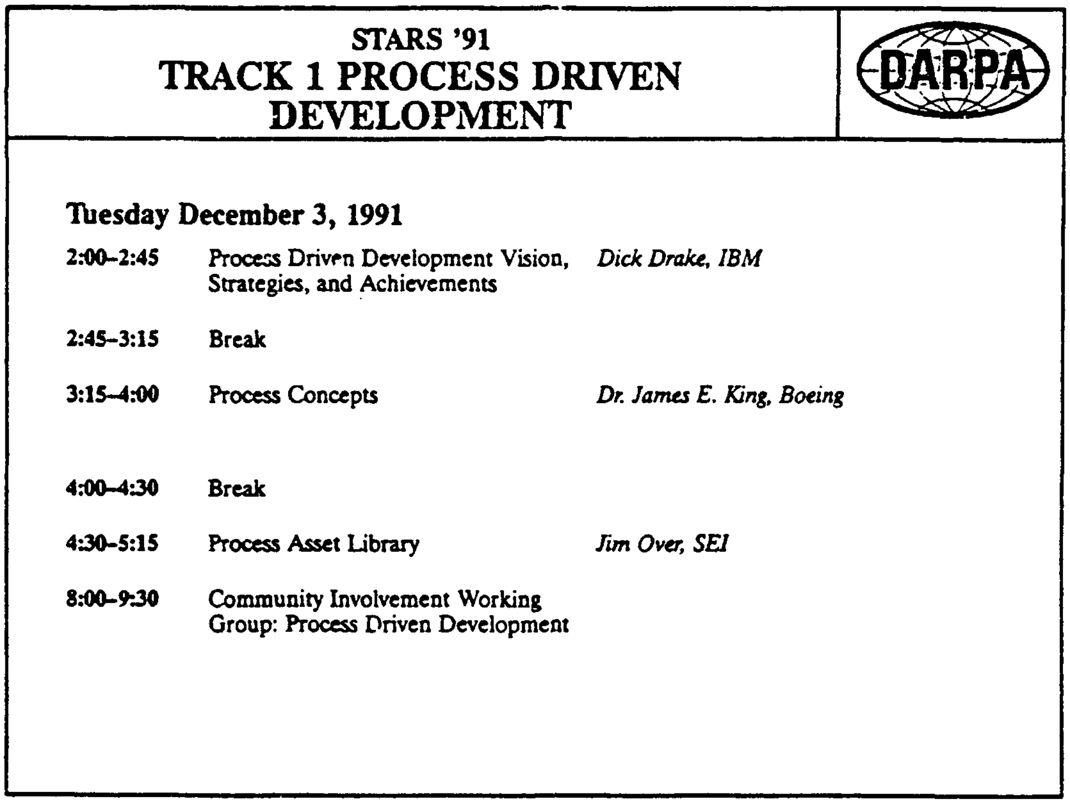
THE DOD SOFTWARE TECHNOLOGY PLAN AND STARS

PARTICIPATION OPPORTUNITIES

DARPA

1990	**Drafts 1,2,3**	**Scoping current efforts, technology areas**
4/91	**Draft 4**	**User needs, technology integration**
8/91	**Draft 5**	**Specific programs; investment priorities**
10/91	**Draft 6**	**Public release version approval** **– Funding details not included**
1/92		**Contractor and researcher review; integration with POM 94**
3/92		**SWTP Public Forum** **(3/31 - 4/2/92, Tyson's Corner)**
5/92		**Baseline Plan**

DoD SWTP and STARS/B. Boehm/VG26



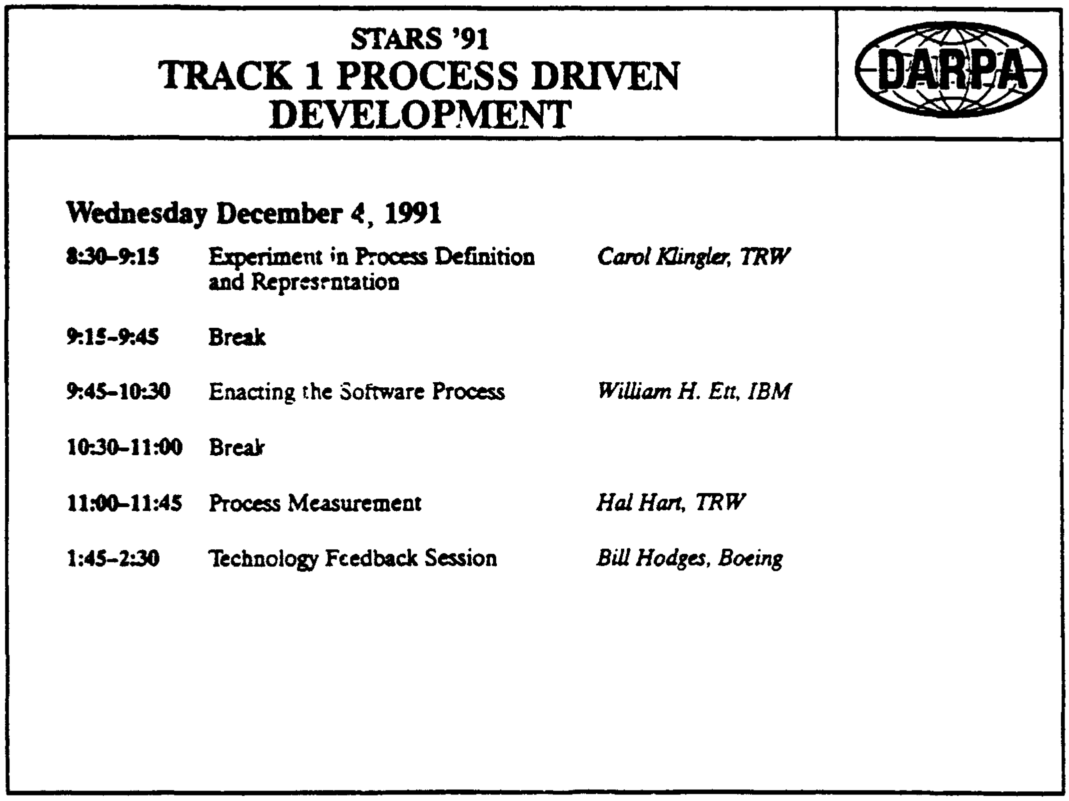
STARS '91

TRACK 1 PROCESS DRIVEN DEVELOPMENT

DARPA

Tuesday December 3, 1991

2:00–2:45	Process Driven Development Vision, Strategies, and Achievements	*Dick Drake, IBM*
2:45–3:15	Break	
3:15–4:00	Process Concepts	*Dr. James E. King, Boeing*
4:00–4:30	Break	
4:30–5:15	Process Asset Library	*Jim Over, SEI*
8:00–9:30	Community Involvement Working Group: Process Driven Development	



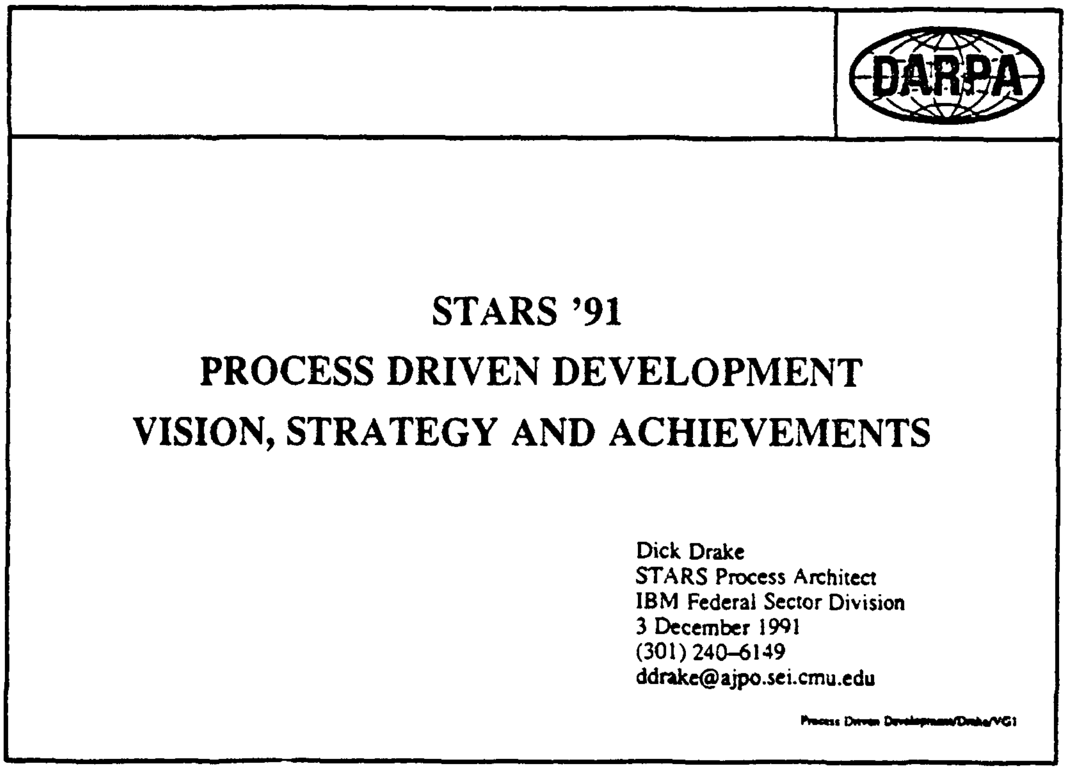
STARS '91

TRACK 1 PROCESS DRIVEN DEVELOPMENT

DARPA

Wednesday December 4, 1991

8:30–9:15	Experiment in Process Definition and Representation	*Carol Klingler, TRW*
9:15–9:45	Break	
9:45–10:30	Enacting the Software Process	*William H. Ett, IBM*
10:30–11:00	Break	
11:00–11:45	Process Measurement	*Hal Hart, TRW*
1:45–2:30	Technology Feedback Session	*Bill Hodges, Boeing*

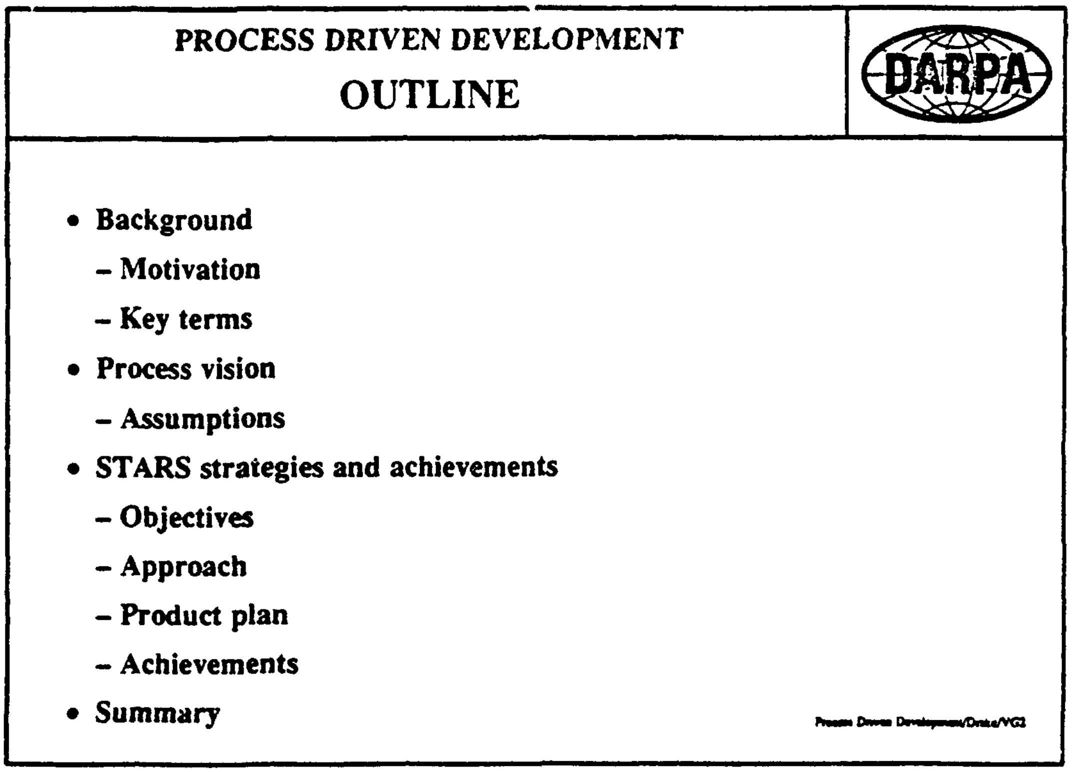


DARPA

STARS '91
PROCESS DRIVEN DEVELOPMENT
VISION, STRATEGY AND ACHIEVEMENTS

Dick Drake
STARS Process Architect
IBM Federal Sector Division
3 December 1991
(301) 240-6149
ddrake@ajpo.sei.cmu.edu

Process Driven Development/Drake/VG1

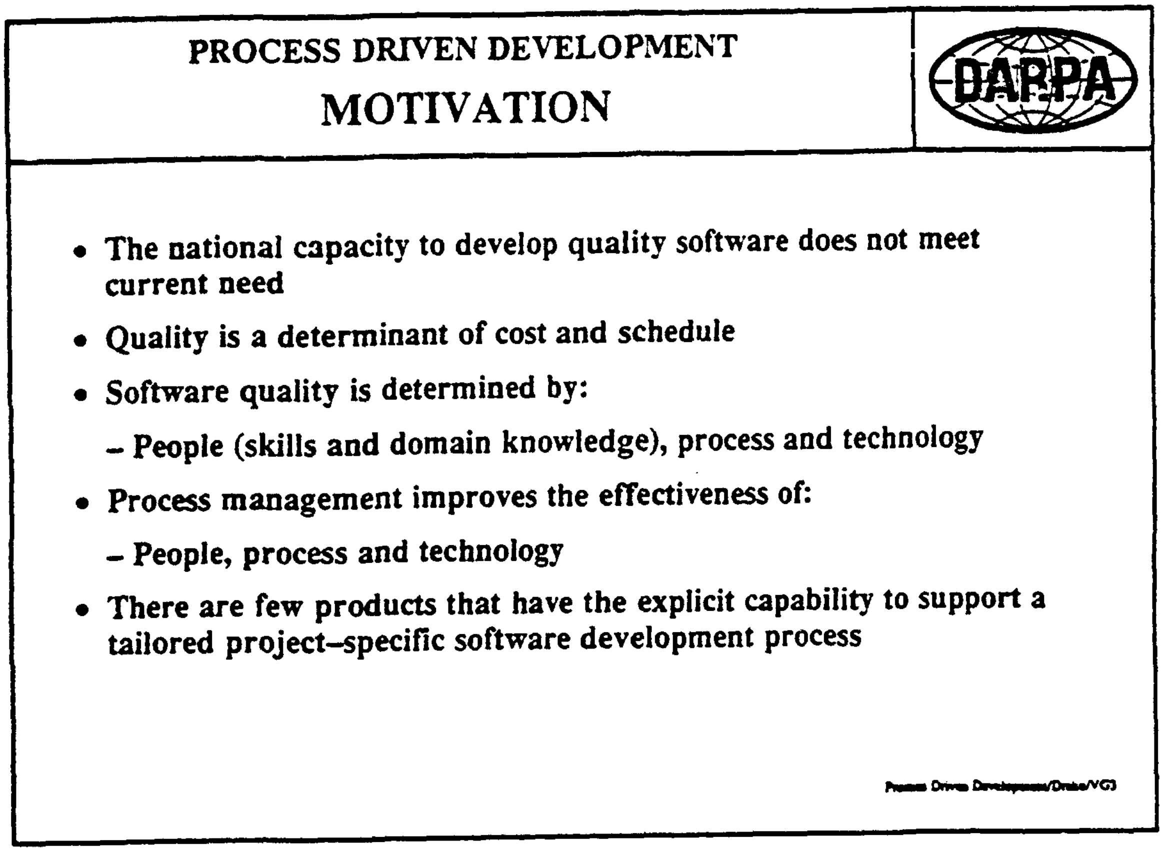


PROCESS DRIVEN DEVELOPMENT

OUTLINE

- Background
  - Motivation
  - Key terms
- Process vision
  - Assumptions
- STARS strategies and achievements
  - Objectives
  - Approach
  - Product plan
  - Achievements
- Summary

Process Driven Development/Drake/VG2



PROCESS DRIVEN DEVELOPMENT

MOTIVATION

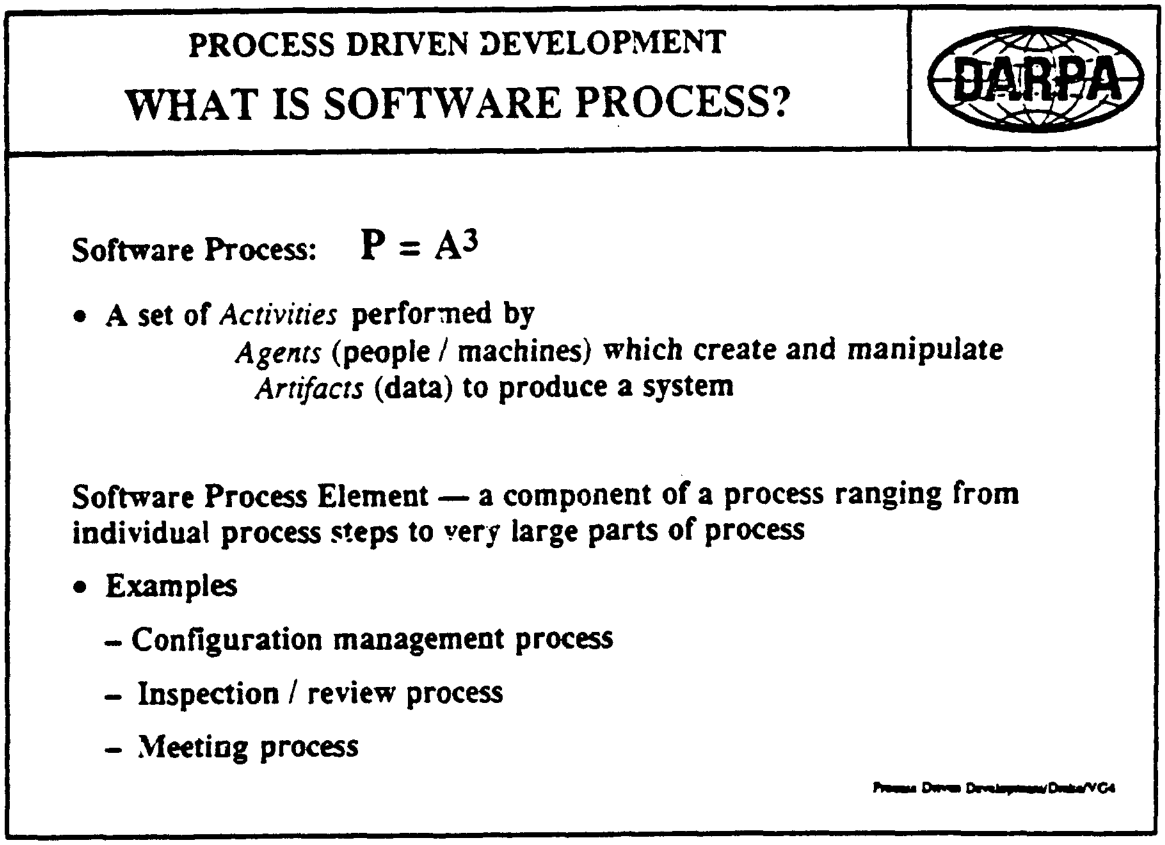
DARPA

- The national capacity to develop quality software does not meet current need
- Quality is a determinant of cost and schedule
- Software quality is determined by:
  - People (skills and domain knowledge), process and technology
- Process management improves the effectiveness of:
  - People, process and technology
- There are few products that have the explicit capability to support a tailored project–specific software development process

Process Driven Development/Drake/VG3

Why is software process important? The national capacity to develop quality software products in a reliable, predictable manner, does not meet the current need in the United States. Software product quality is a key determinant of both software cost and schedule. Lack of attention to the quality of a software system during its complete development cycle will almost certainly result in increases in cost and schedule over the long run.

Software quality is determined by people (their skills and their knowledge of the application domain), process and technology used to produce the software product. Process management has been shown to improve the effectiveness of the people, process and technology used to produce software. However, within the domain of software development, there are few products that have the explicit capability to support a tailored, project–specific software development process. The STARS mission is to accelerate the availability of processes and technologies to support process driven development.



PROCESS DRIVEN DEVELOPMENT

WHAT IS SOFTWARE PROCESS?

DARPA

Software Process: $P = A^3$

- A set of *Activities* performed by
  Agents (people / machines) which create and manipulate
  Artifacts (data) to produce a system

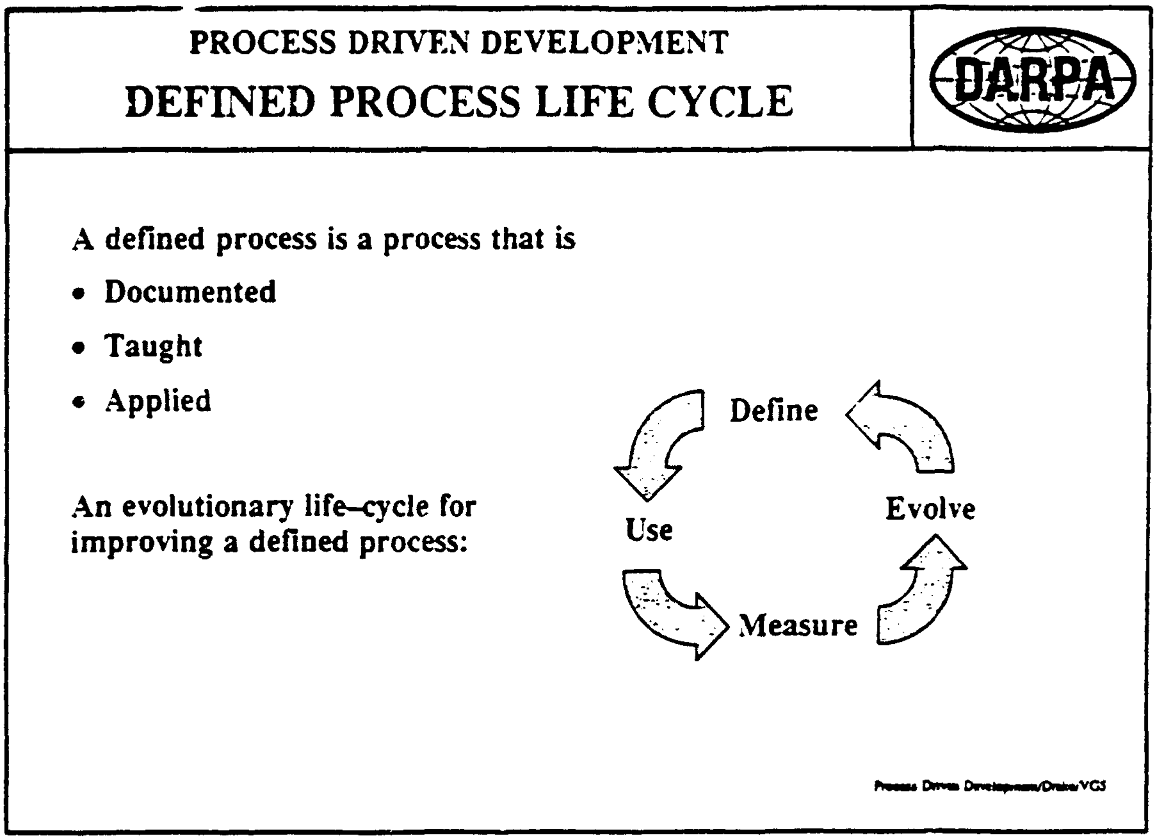
Software Process Element — a component of a process ranging from individual process steps to very large parts of process

- Examples
  - Configuration management process
  - Inspection / review process
  - Meeting process

Process Driven Development/[illegible]/VC4

First let's define a few key terms and concepts which will be important to an understanding of how STARS is supporting the definition and automation of software process. One simple way to define software process is to look at it as a set of Activities, performed by Agents (people or machines), which create and manipulate Artifacts (data, work products) to produce a system. It is important to remember that process is always there whether we carefully define it or not. Problems occur when processes are poorly defined, misunderstood and inconsistently applied.

Software processes are made up of software process elements. A process element is a component of a process ranging from individual process steps to very large parts of a process. For example within a configuration management process you might expect to find a number of process elements for conducting inspections or reviews. Likewise you would probably find meeting processes incorporated within a review processes.



PROCESS DRIVEN DEVELOPMENT

DEFINED PROCESS LIFE CYCLE

DARPA

A defined process is a process that is

- Documented
- Taught
- Applied

An evolutionary life–cycle for improving a defined process:

Define

Evolve

Use

Measure

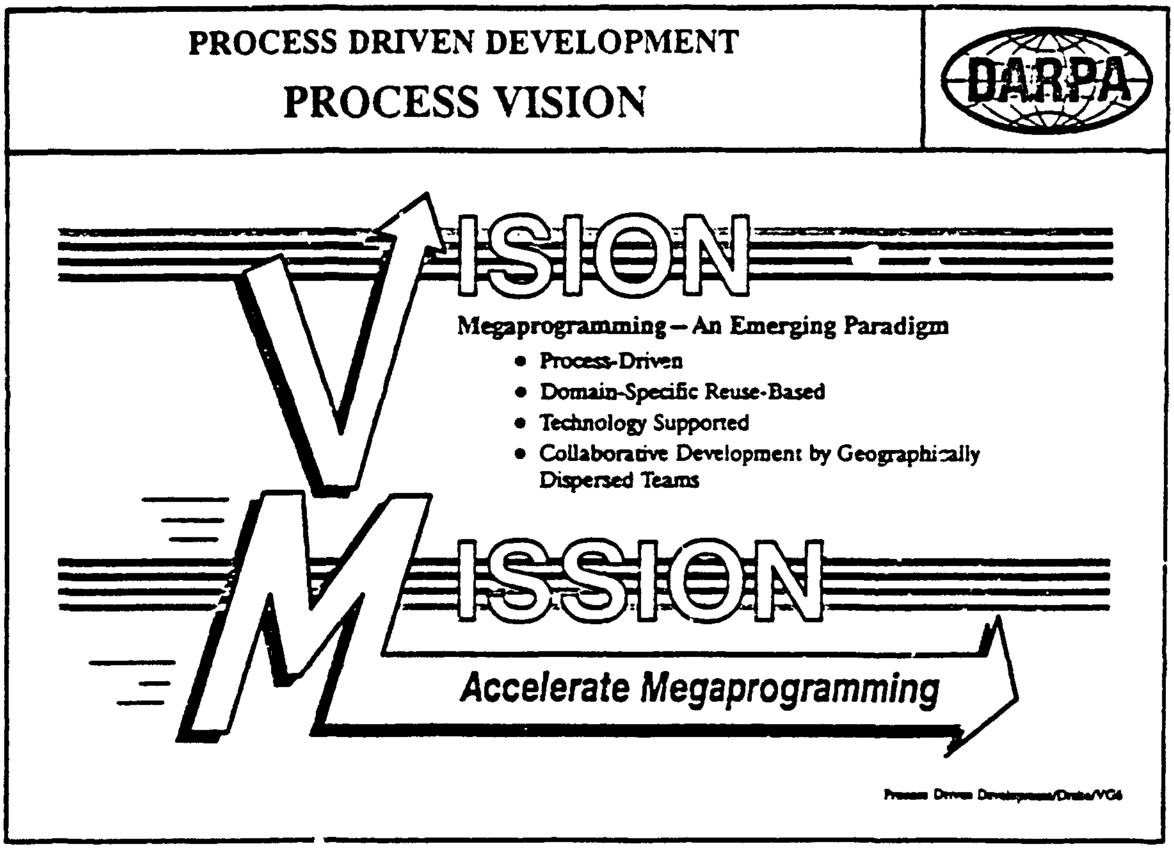
Process Driven Development/Drake/VC3

One of the most important concepts with respect to process driven development is the notion of a defined process. To qualify as a defined process it is not sufficient to have the process described in a notebook which can be found on the desk of all project personnel. A defined process must be documented, it must be taught to the people expected to apply it and it must be applied as documented and taught.

Process, however, is not a static thing. Processes are constantly changing and a process which is not changing is probably obsolete. Therefore, one must view process in terms of an evolutionary life–cycle for improving a defined process. A simple way to think of this life cycle is:

- *Define* – Establish the organizations process, adapt it to the specific project and product requirements and train people in its use.
- *Use* – Apply the process as defined
- *Measure* – Constantly monitor and measure the process as it is being performed
- *Evolve* – Continually evolve and improve the process based on the measurements and experience gained.

At each iteration through this cycle the definition would be refined.



PROCESS DRIVEN DEVELOPMENT

PROCESS VISION

VISION

Megaprogramming – An Emerging Paradigm

- Process-Driven
- Domain-Specific Reuse-Based
- Technology Supported
- Collaborative Development by Geographically Dispersed Teams

MISSION

Accelerate Megaprogramming

Process Driven Development/Drabe/VG4

Megaprogramming is an emerging paradigm which will dramatically change the way we produce software. A change of this proportion will take a long time to pervade the industry. A key ekment of this emerging paradigm is process driven development. The following pages will define what is meant by process driven and provide some underlying assumptions about how it can be applied.

PROCESS DRIVEN DEVELOPMENT

PROCESS - DRIVEN VISION

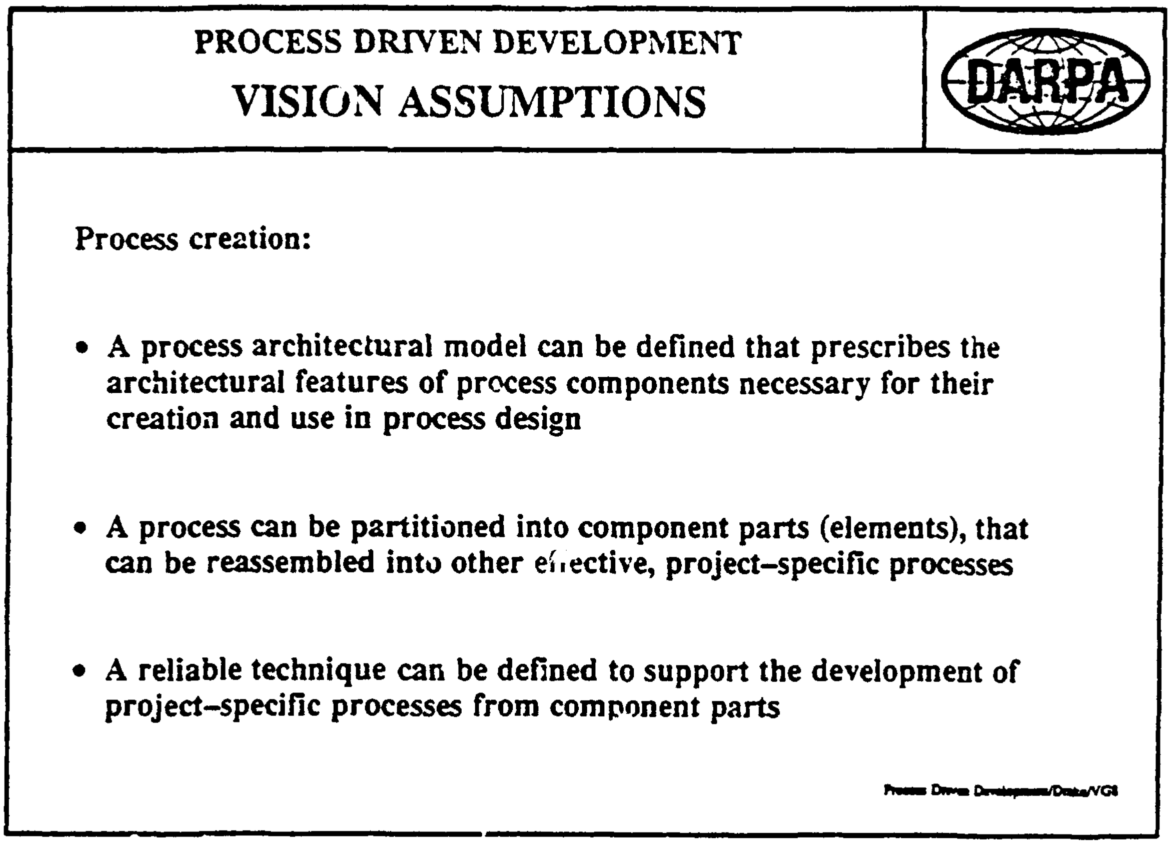
- Organizational process is established and then adapted and tailored to meet project and product goals
- Software development is guided by a defined process
- Environment and tools are integrated to support a defined process
- Defined process promotes collaboration and teamwork by making activities, roles, and dependencies visible
- Process management discipline supports continuous improvement of the defined process through measurement and feedback

Process Driven Development/Drake/VG7

Process driven development begins when an organization establishes the processes necessary to support their objectives. These processes are then adapted and tailored to meet the needs of the specific project and product to be built. The software development activities will be guided by this defined process (documented, taught and applied). A software development environment and its tools would be established and integrated based on the tailored process. This implies that you understand your process before you select your tools to carry out the process.

The use of the defined process will promote collaboration and teamwork by making activities, roles (taken on by agents) and dependencies visible to all project personnel. The discipline associated with a defined process will result in continuous process improvement through measurement and feedback (Define, Use, Measure, Evolve).

So in summary, process driven development implies that you have a defined process which is tailored to the problem and which is continually improved through measurement and feedback.



PROCESS DRIVEN DEVELOPMENT

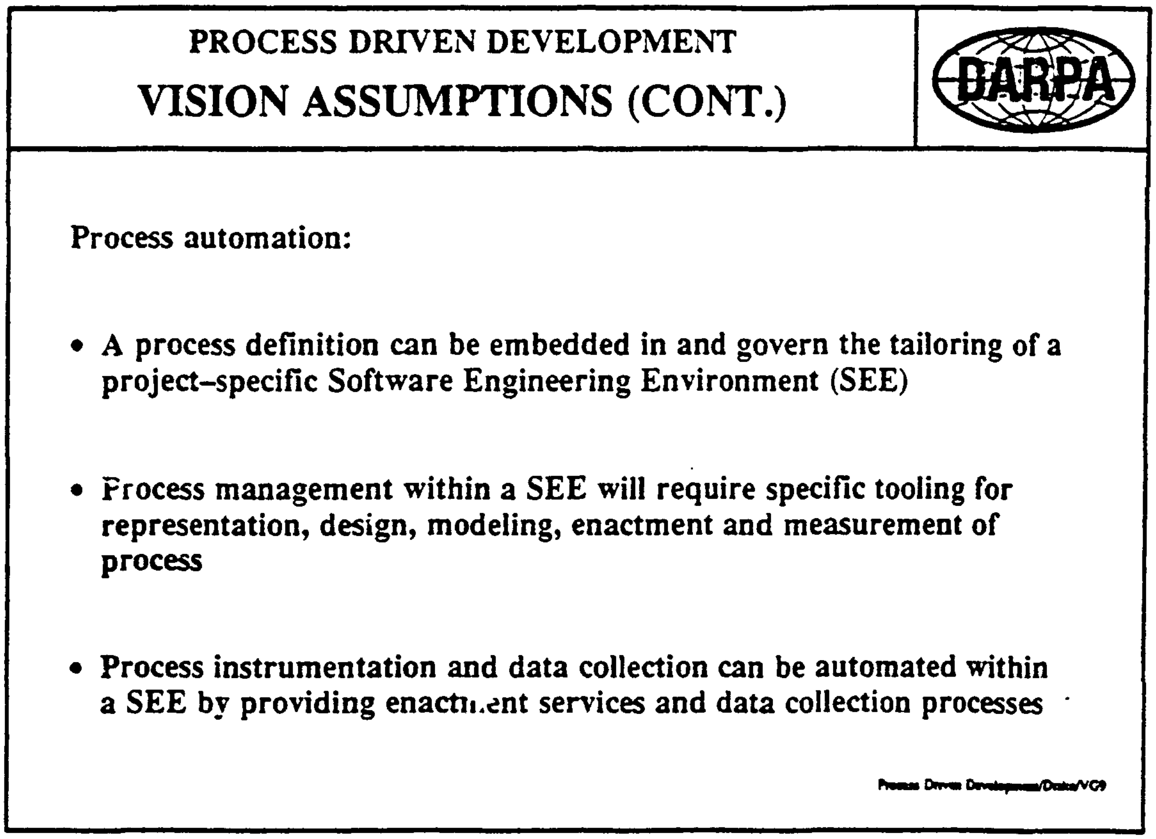
VISION ASSUMPTIONS

Process creation:

- A process architectural model can be defined that prescribes the architectural features of process components necessary for their creation and use in process design
- A process can be partitioned into component parts (elements), that can be reassembled into other effective, project-specific processes
- A reliable technique can be defined to support the development of project-specific processes from component parts

Process Driven Development/Data/VG8

There are several important assumptions underlying process driven development. Processes will be developed somewhat like software itself. A process architecture will be created in order to support the use of process elements in the construction of the process. Process will be constructed by assembling existing component parts to support project and product specific requirements. In other words, the processes will be constructed using libraries of existing process elements. Since processes need to be constructed, tailored and refined, a "process life-cycle process" will be developed to guide this process evolution.



PROCESS DRIVEN DEVELOPMENT

VISION ASSUMPTIONS (CONT.)

Process automation:

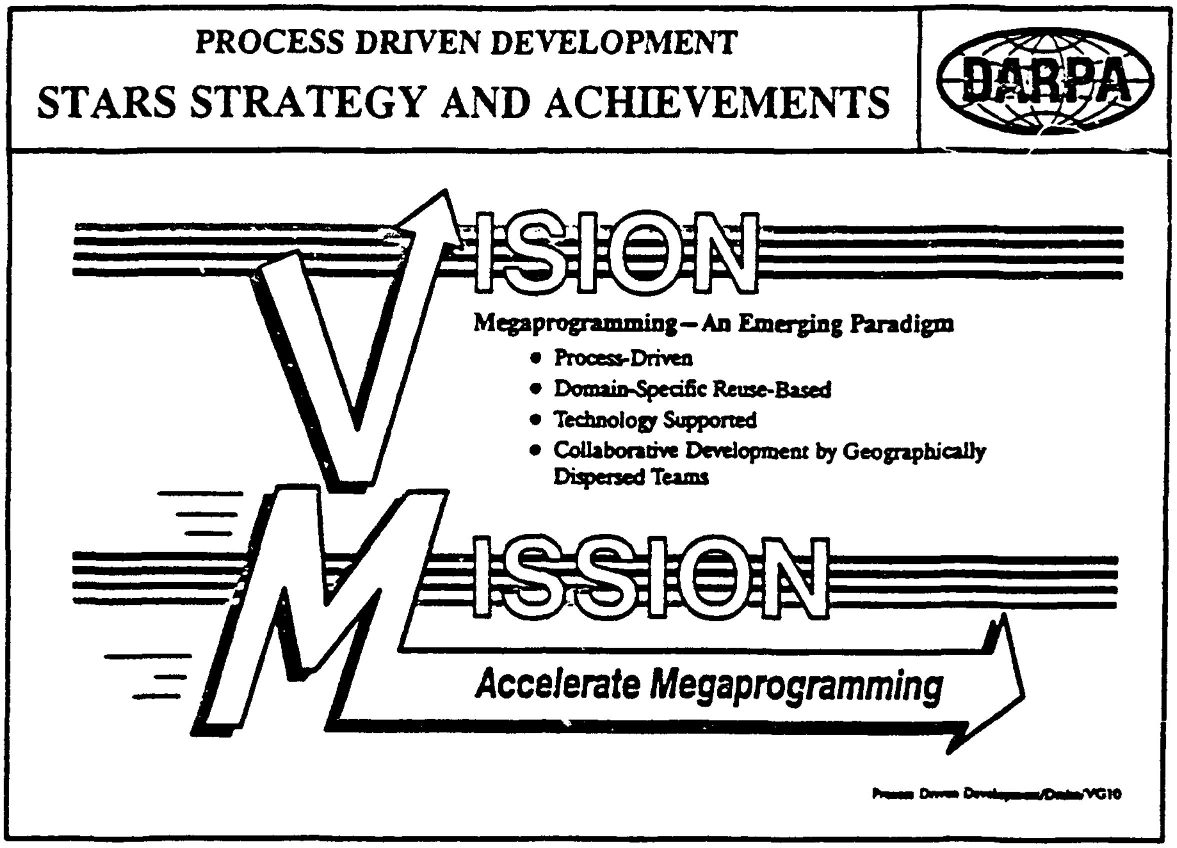
- A process definition can be embedded in and govern the tailoring of a project–specific Software Engineering Environment (SEE)
- Process management within a SEE will require specific tooling for representation, design, modeling, enactment and measurement of process
- Process instrumentation and data collection can be automated within a SEE by providing enactment services and data collection processes

Process Driven Development/Drake/VG9

Process driven development will require significant automation. The assumptions related to automation include the notion that process can be defined in a formal enough way to be embedded within the software engineering environment. This implies that the process is central to the tailoring of the environment. The resulting environment will contain knowledge of the process and thereby it will be able to offer many opportunities for automation.

The management of large complex software applications will require process support. In the future software development environments will contain support capabilities for representing, designing, modeling, simulating, measuring, enacting, dynamically changing and evolving software process.

The key to process improvement is measurement and feedback. Software engineering environments can be expected to contain support for instrumentation and data collection. These capabilities will have a great potential for automation in an environment which has an embedded process definition incorporated within it.



PROCESS DRIVEN DEVELOPMENT

STARS STRATEGY AND ACHIEVEMENTS

DARPA

VISION

Megaprogramming – An Emerging Paradigm

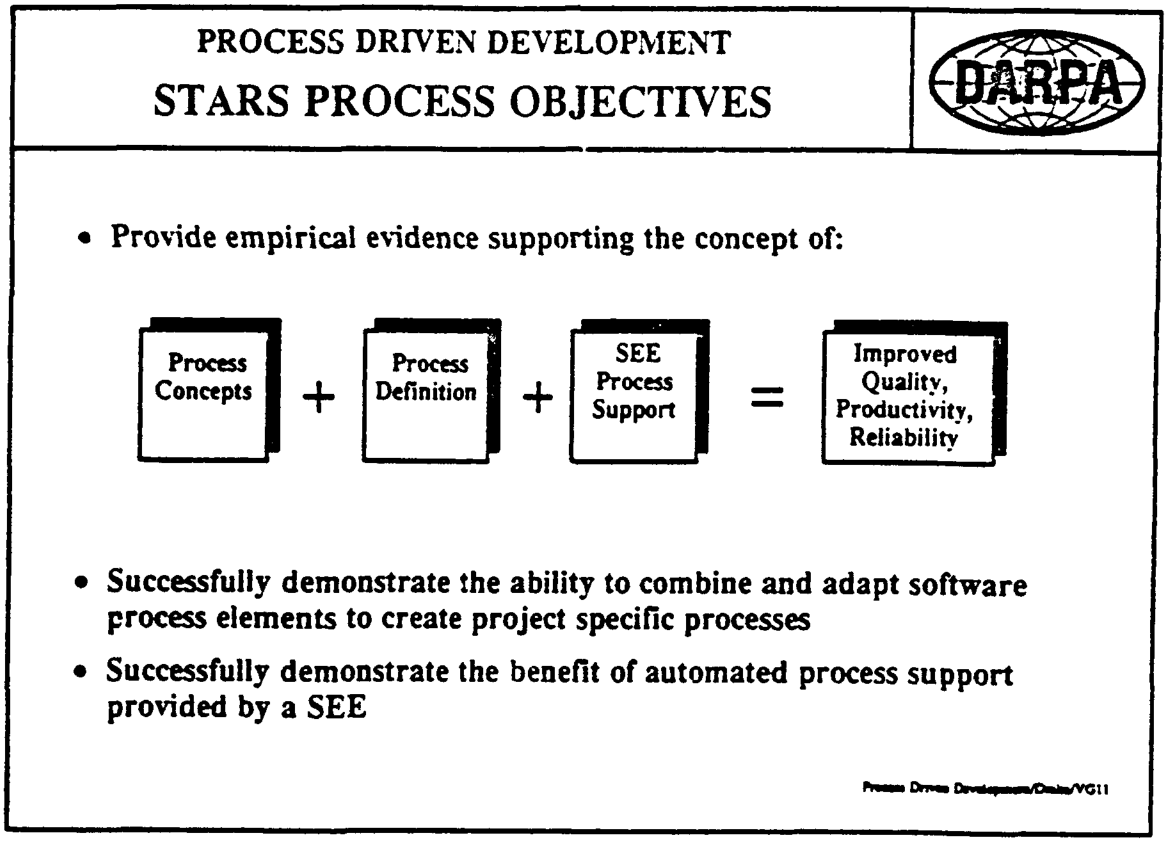
- Process-Driven
- Domain-Specific Reuse-Based
- Technology Supported
- Collaborative Development by Geographically Dispersed Teams

MISSION

Accelerate Megaprogramming

Process Driven Development/Drake/VG10

The previous pages have provided context and definition to the Megaprogramming notion of process driven development. The remainder of this presentation will concentrate on the STARS activities in support of the STARS mission to accelerate the transition to the process driven development aspects of Megaprogramming.



PROCESS DRIVEN DEVELOPMENT

STARS PROCESS OBJECTIVES

DARPA

- Provide empirical evidence supporting the concept of:

Process Concepts + Process Definition + SEE Process Support = Improved Quality, Productivity, Reliability

- **Successfully demonstrate the ability to combine and adapt software process elements to create project specific processes**
- **Successfully demonstrate the benefit of automated process support provided by a SEE**

Process Driven Development/[illegible]/VG11

STARS has an objective to demonstrate the value of process driven development. This includes providing evidence supporting the following concept:

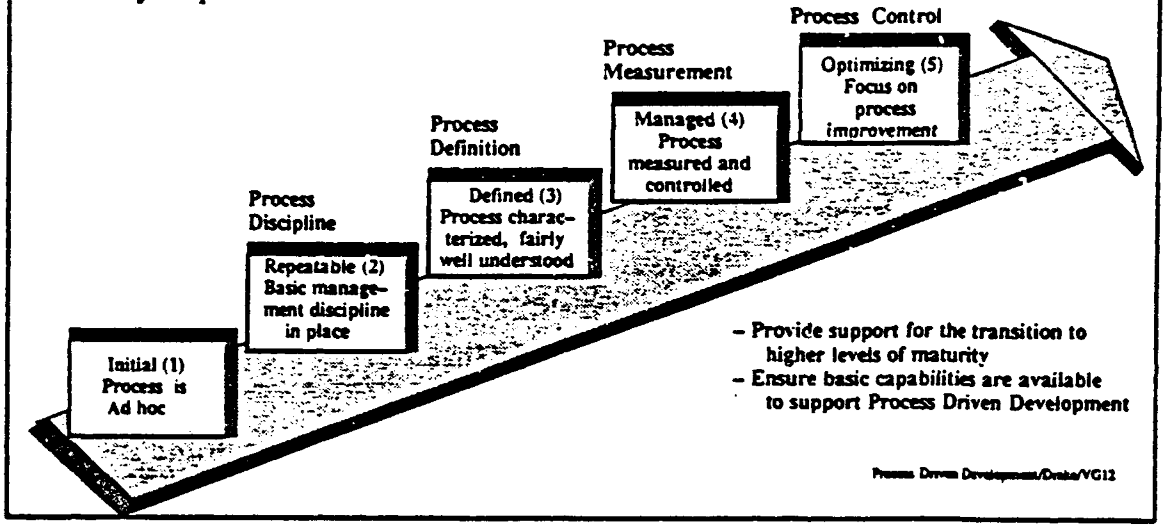
- If you start with a good set of process concepts, then create a process definition to support the project and product goals, and support this with automated process support within the environment,
- Then the results will be improved quality, productivity and reliability.

STARS will demonstrate the ability to combine and adapt software process elements in order to create a project specific process. STARS will also demonstrate the benefits that automation for the process within the SEE. The remaining presentations within this track will elaborate on this concept.

PROCESS DRIVEN DEVELOPMENT

STARS PROCESS OBJECTIVES (CONT.)

- Provide process and technology support to assist software development organizations in their progression up the SEI Capability Maturity Model (CMM) for defining, measuring, and evolving the project's life cycle processes


Process Control
Optimizing (5) Focus on process improvement
Process Measurement
Managed (4) Process measured and controlled
Process Definition
Defined (3) Process characterized, fairly well understood
Process Discipline
Repeatable (2) Basic management discipline in place
Initial (1) Process is Ad hoc

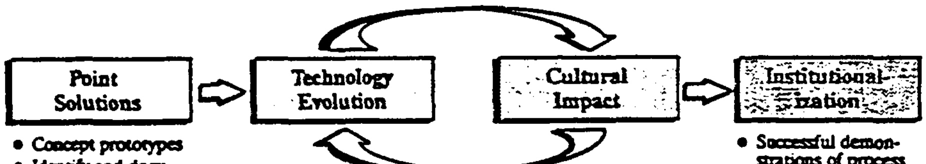
- Provide support for the transition to higher levels of maturity
- Ensure basic capabilities are available to support Process Driven Development

Process Driven Development/Drake/VG12

STARS will provide process and technology support to assist software development organizations in their progression up the SEI Capability Maturity Model (CMM). This includes both support for the transition to process driven development such as concepts, guidelines and techniques as well as basic capabilities and tools needed for process driven development.

PROCESS DRIVEN DEVELOPMENT

STARS PROCESS APPROACH



Point Solutions

- Concept prototypes
- Identify and document experience-tested software processes
- Process modeling experiments
- Process enaction experiments

Technology Evolution

- Library of fundamental process assets
- Capability to compose a software process from reusable process assets
- Capability to guide and support the performance and measurement of a defined process

Cultural Impact

- Guidelines and techniques f:r process definition and measurement
- Guidelines for tailoring process to specific development standards and acquisition policy
- Identify and publicize cost benefit models for process-driven development

Institutionalization

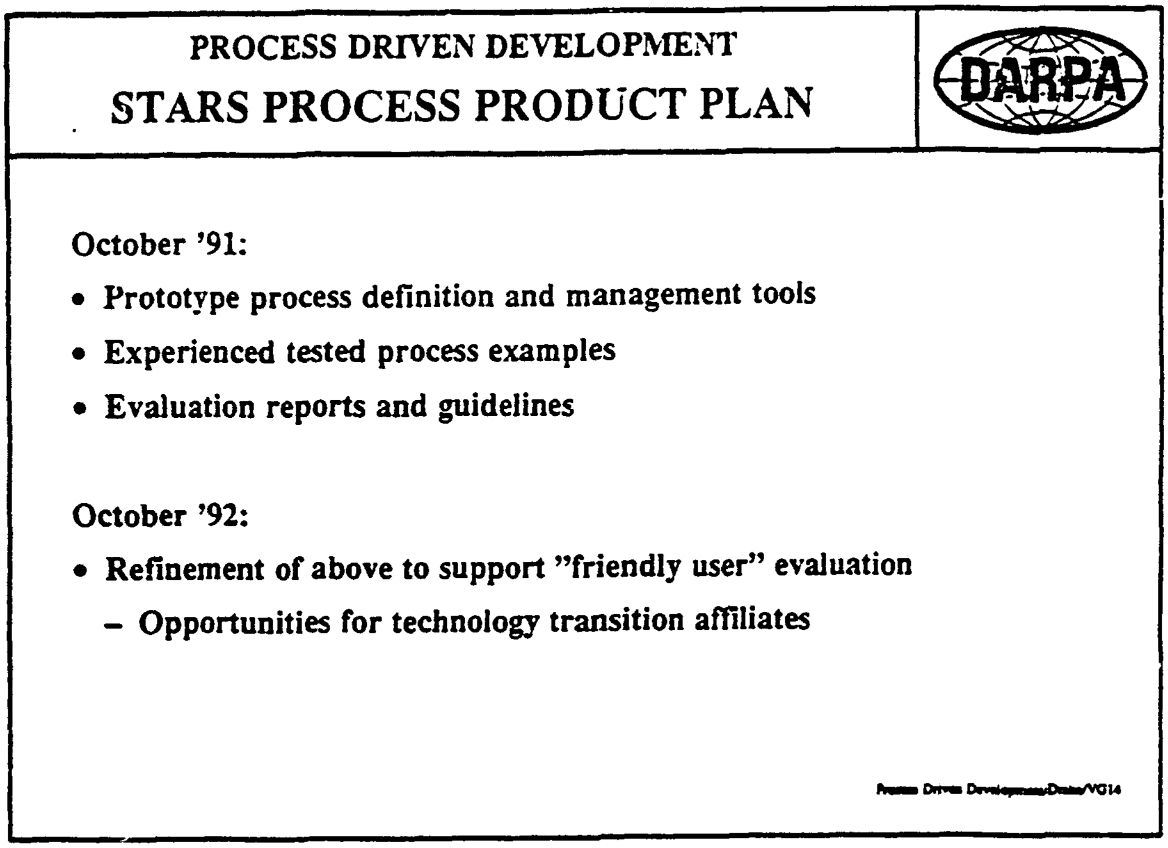
- Successful demonstrations of process definition and management on DoD projects
- Commercial technology for process defin.tion
- Integrated SEE support for process management

Process Driven Development/Dates/VG13

The STARS approach to accelerating the shift to Megaprogramming involves starting with point solutions and then evolving those solutions to support software in the large. This involves maturing and integrating the capabilities within the software engineering environment. At the same time as the capabilities are being matured, STARS will address cultural issues of how to apply these concepts and how to transition organizations to a process driven approach. STARS will focus its efforts in three key areas:

- Process definition and representation
- Process Asset Library
- Process enactment and measurement

STARS will attempt to begin the institutionalization of the process driven concepts by demonstrating the benefits on real DoD projects. The STARS prime contractors will work with their commercial counterparts and the CASE vendor community to commercialize the process support capabilities and integrate them within software engineering environments.



PROCESS DRIVEN DEVELOPMENT

STARS PROCESS PRODUCT PLAN

DARPA

October '91:

- Prototype process definition and management tools
- Experienced tested process examples
- Evaluation reports and guidelines

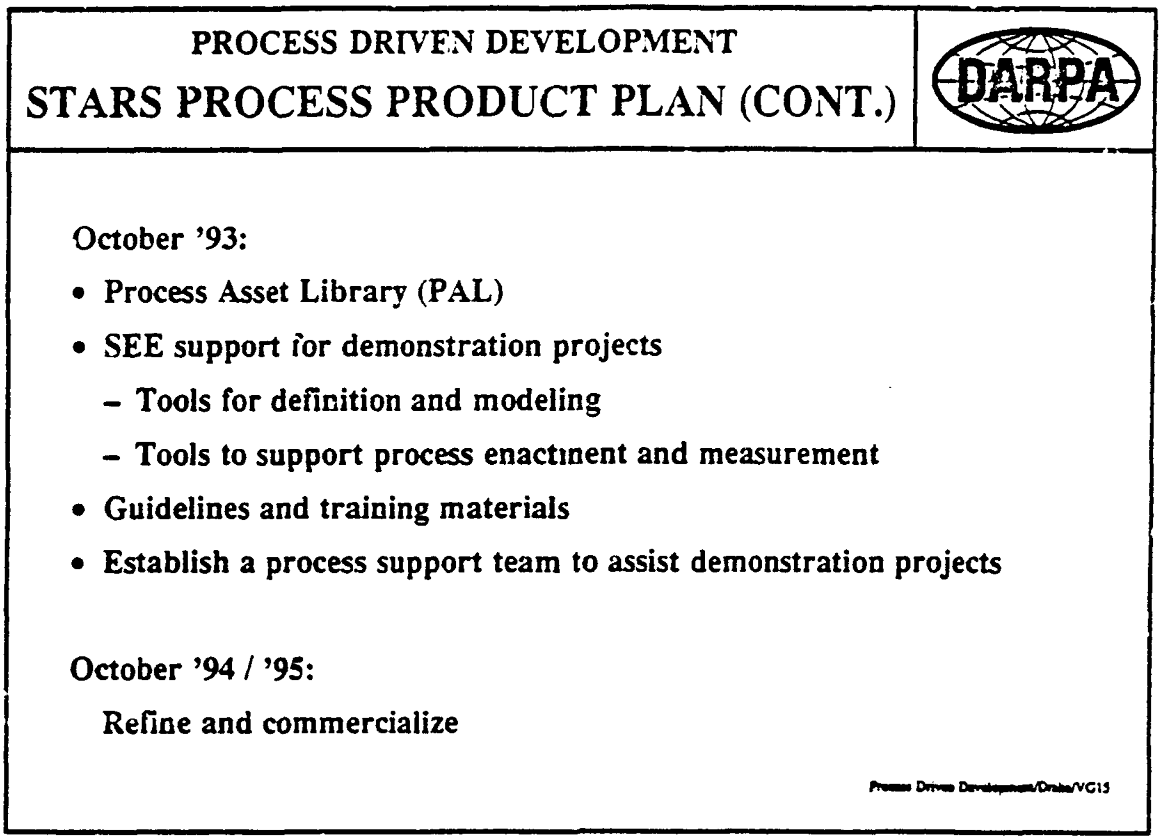
October '92:

- Refinement of above to support "friendly user" evaluation
  - Opportunities for technology transition affiliates

Process Driven Development/Drate/VG14

Over the last year STARS process efforts have been focused on experimenting with and evaluating a number of point solutions. This includes prototypes of a number of different process definition and management tools. STARS in conjunction with the SEI has gathered from industry sources a .nany experience tested processes and has defined a several modern processes. STARS has also published concepts, guidelines and lessons learned reports. The specific items are described later in this presentation and further elaborated in other presentations within this track.

Over the next year STARS will mature these capabilities and begin integrating them in software engineering environments. By the fourth quarter of 1992, these capabilities will be applied in a number of test projects (alpha test cases) in order to gain experience to help refi.e the capabilities. This may offer opportunities for interested Technology Transition affiliates to work closely with the STARS program.



PROCESS DRIVEN DEVELOPMENT

STARS PROCESS PRODUCT PLAN (CONT.)

DARPA

October '93:

- Process Asset Library (PAL)
- SEE support for demonstration projects
  - Tools for definition and modeling
  - Tools to support process enactment and measurement
- Guidelines and training materials
- Establish a process support team to assist demonstration projects

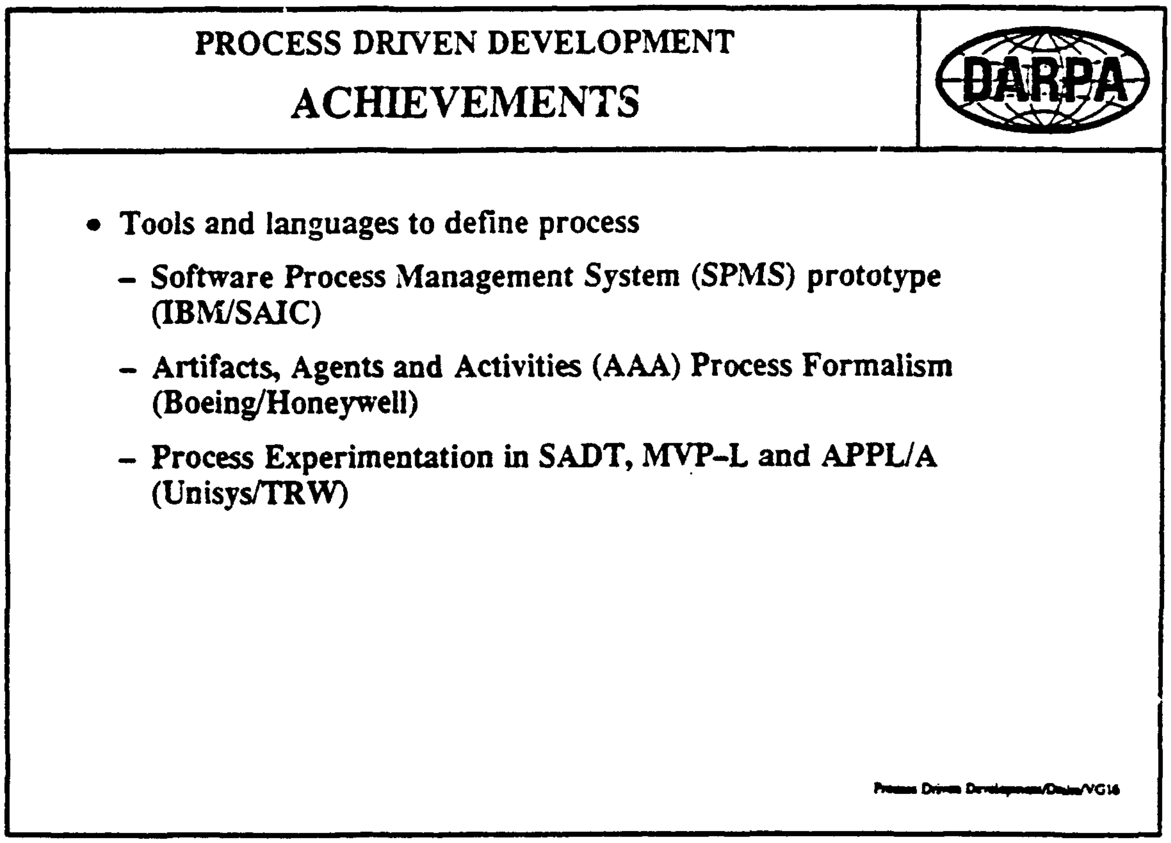
October '94 / '95:

Refine and commercialize

Process Driven Development/Drake/VG15

The STARS demonstration projects will begin in October of 1993. At that time STARS will have capabilities available to support these projects including a Process Asset Library and process definition, modeling, enactment and measurement tools. Guidelines and training material will be available to support these capabilities. A process support team will be formed to provide ongoing support for the demonstration projects.

During 1994 and 1995, STARS will work with the demonstration projects, the STARS affiliates and the CASE vendor community to enhance and refine the capabilities.



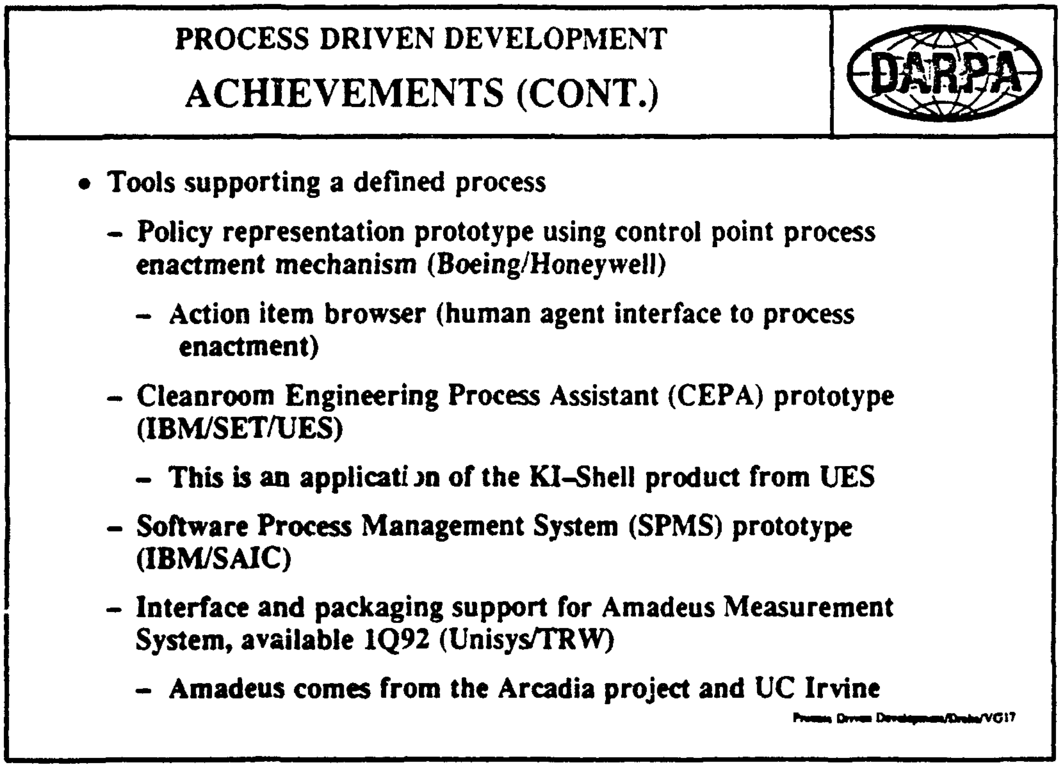
PROCESS DRIVEN DEVELOPMENT

ACHIEVEMENTS

- Tools and languages to define process
  - Software Process Management System (SPMS) prototype (IBM/SAIC)
  - Artifacts, Agents and Activities (AAA) Process Formalism (Boeing/Honeywell)
  - Process Experimentation in SADT, MVP-L and APPL/A (Unisys/TRW)

Process Driven Development/[illegible]/VG16

Further information on the documents, tools and processes listed on these charts can be found either through the STARS Catalog or through participation in the STARS Technology Transfer Affiliates program. Many of the capabilities are available for demonstration.



PROCESS DRIVEN DEVELOPMENT

ACHIEVEMENTS (CONT.)

- Tools supporting a defined process
  - Policy representation prototype using control point process enactment mechanism (Boeing/Honeywell)
    - Action item browser (human agent interface to process enactment)
  - Cleanroom Engineering Process Assistant (CEPA) prototype (IBM/SET/UES)
    - This is an applicati ɔn of the KI–Shell product from UES
  - Software Process Management System (SPMS) prototype (IBM/SAIC)
  - Interface and packaging support for Amadeus Measurement System, available 1Q92 (Unisys/TRW)
    - Amadeus comes from the Arcadia project and UC Irvine

Process Driven Development/Drake/VG17

PROCESS DRIVEN DEVELOPMENT

ACHIEVEMENTS (CONT.)

- Processes
  - Cleanroom engineering software process (IBM/SET)
  - Composite Process Model (Unisys/TRW)
  - Risk–reduction reasoning–based development paradigm tailored to Navy C2 systems (Unisys/TRW)
  - Software–first system development process (IBM)
  - SEI/STARS joint effort to acquire experience tested processes
  - Domain analysis process (IBM/SAIC)
  - Asset certification process (Unisys)
  - IEEE P 1074 process component set (SAIC)

PROCESS DRIVEN DEVELOPMENT

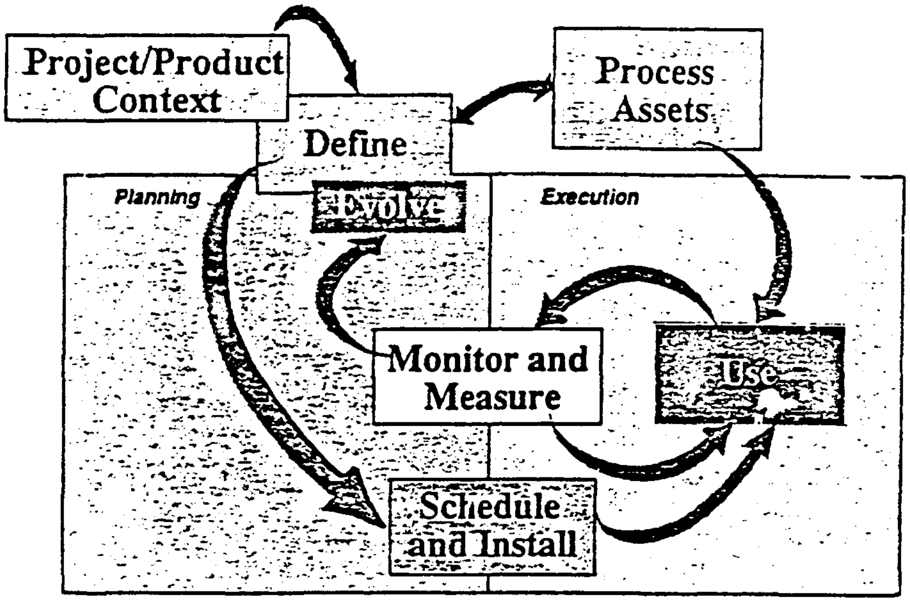
ACHIEVEMENTS (CONT.)

- Concepts / guidelines / lessons learned
  - Process Concep'ᵗˢ Summary (Unisys/Boeing/IBM/Others)
  - Software Process Tools and Techniques Evaluation Report (IBM)
  - Process Concepts Scenarios (Boeing)
  - Process Definition Advisory Group (PDAG) Workshop Report (SEI)
  - Process Programming Language Experimentation Report (Unisy.. TRW)
  - Process Programming Experiment: Initial Lessons Learned (Unisys/TRW/University of Maryland)

Process Driven Development/[illegible]/VG19

PROCESS DRIVEN DEVELOPMENT

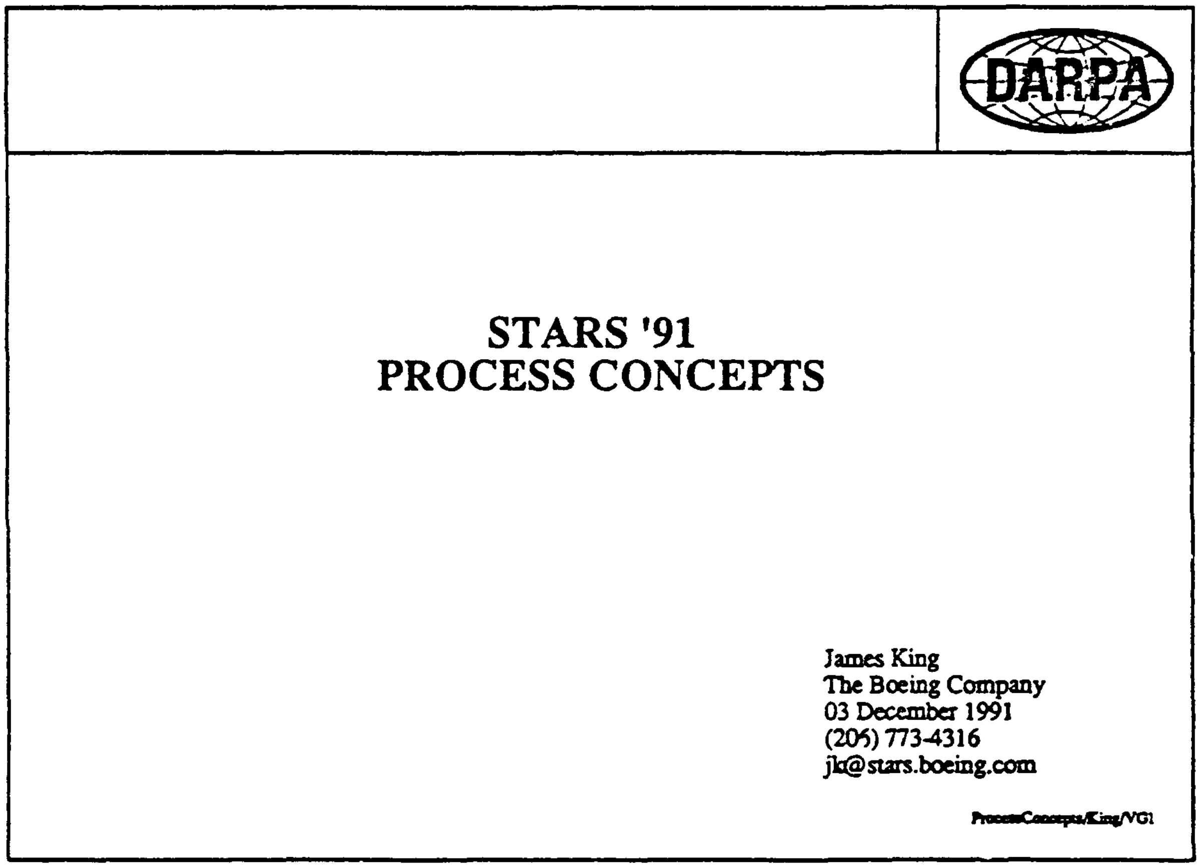
SUMMARY



Adapted from Software Productivity Consortium work.

Process Driven Development/Drake/VG20

Process driven development begins with a defined process which has been tailored to meet the specific project and product requirements. The process is performed, monitored and measured. The feedback is used to evolve the process and refine the process definition.



STARS '91
PROCESS CONCEPTS

James King
The Boeing Company
03 December 1991
(205) 773-4316
jk@stars.boeing.com

ProcessConcepts/King/VG1

STARS'91

Track: PROCESS Session: 2

Title: Process Concepts

Presenter: Dr. James E. King Organization: The Boeing Company, Defense & Space Group

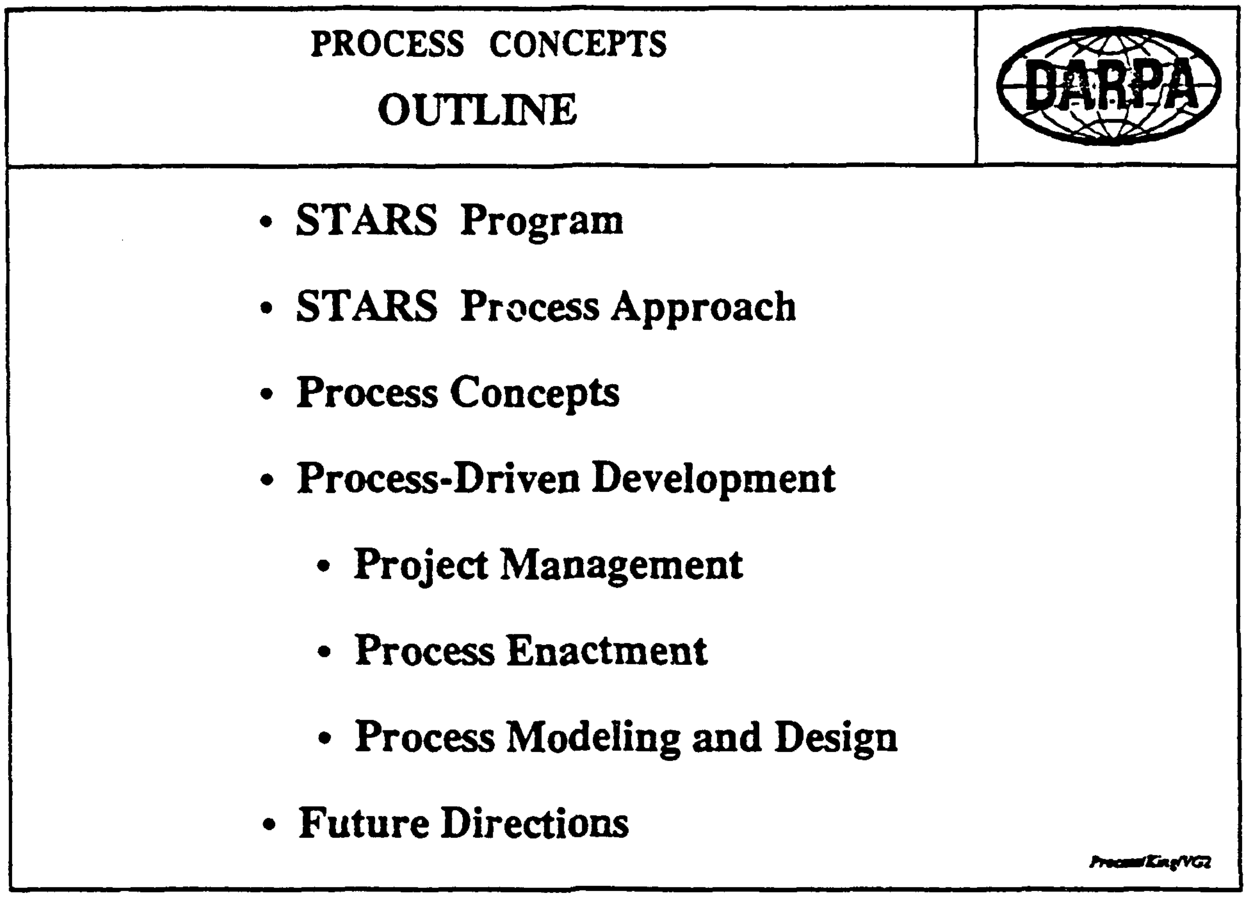
Theme:
Process-driven Environments provide the means to achieve improved quality by reducing variability in planning, by eliminating costly and error-prone sequences of tasks, and by providing data which assists in improving the defined processes being used.

Objective:

Articulate the long range concepts and identify what STARS will be able to achieve.

Abstract:

What is a process-driven software engineering environment? How will it effect the way I work? How can it help me work better? These questions and other related topics will be addressed by examining the effects on typical users of the environment, the types of activities that will be affected, and the interrelationship between users' activities that will result from the transition to megaprogramming. The concepts will be illustrated through scenarios of user interaction with the envisioned process-driven software engineering environment. The scenarios will represent user views specific to activities performed by different users at various times in the life of a system development project.



PROCESS CONCEPTS

OUTLINE

DARPA

- STARS Program
- STARS Prccess Approach
- Process Concepts
- Process-Driven Development
  - Project Management
  - Process Enactment
  - Process Modeling and Design
- Future Directions

ProcessEng/VG2

Slide 2: Title: Process Concepts

STARS is focusing its attention on an emerging paradigm, entitled Megaprogramming, which is based on incorporating:

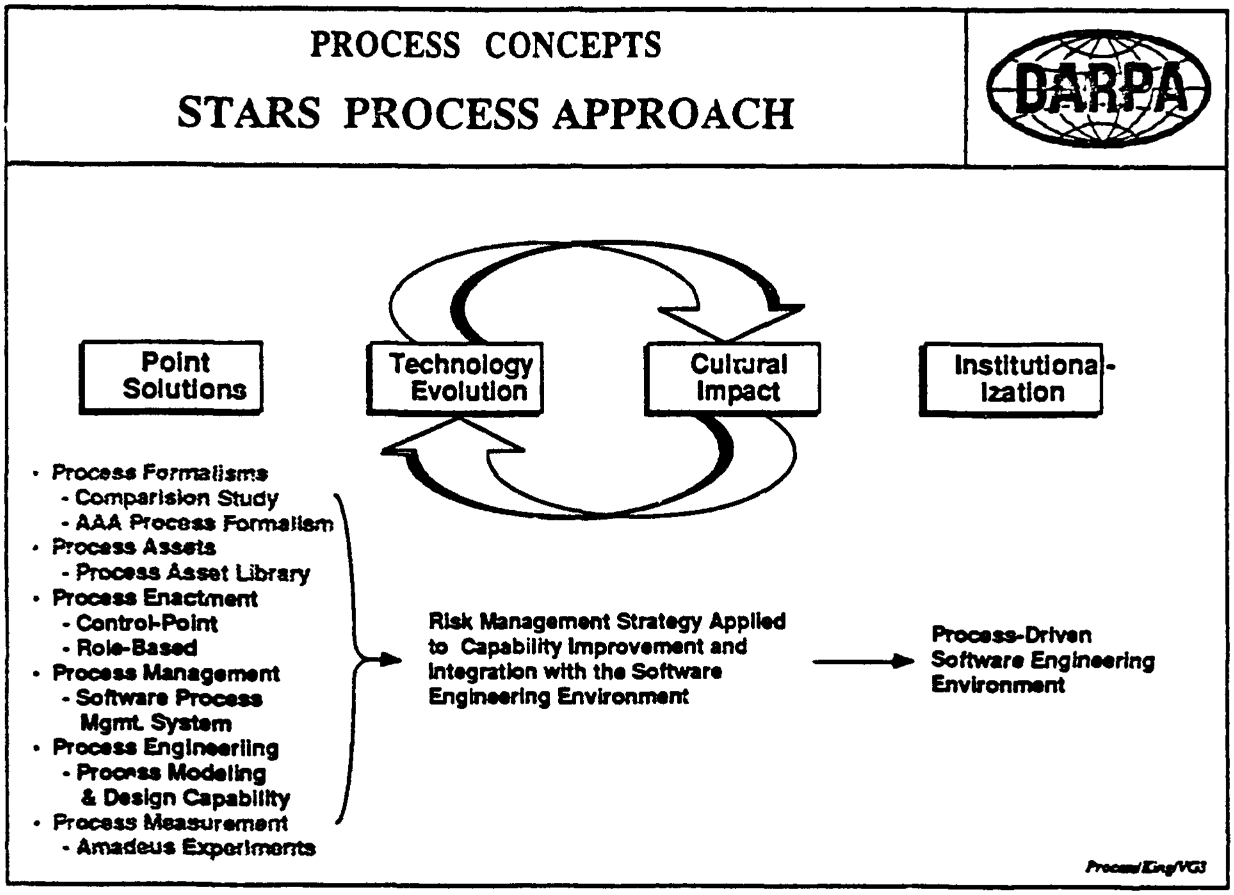
- Process-driven,
- Domain-specific reuse-based,
- Technology supported,
- Collaborative development

technologies into software development.

STARS mission is to accelerate the shift to this emerging paradigm.

Although STARS has identified these four major thrusts, they are not independent. In fact the process thrust is another instance of a specific domain. Each of these areas which are presented through the four tracks at this conference are highly interrelated. The defined process is deployed with technology support to provide a process-driven Software Engineering Environment (SEE). The SEE provides network based collaborative development and distributed computing in an open architecture.

The focus of this presentation is to elaborate the STARS view of process driven software development and how this emphasis leads to higher levels of product quality. A long-term view of process-driven development will be presented. Different views of this system will be illustrated in terms of activities associated with different users. Early point solutions will be discussed and the ground-work laid fcr the STARS process activities.



PROCESS CONCEPTS

STARS PROCESS APPROACH

Point Solutions

Technology Evolution

Cultural Impact

Institutionalization

- Process Formalisms
  - Comparision Study
  - AAA Process Formalism
- Process Assets
  - Process Asset Library
- Process Enactment
  - Control-Point
  - Role-Based
- Process Management
  - Software Process Mgmt. System
- Process Engineerling
  - Process Modeling & Design Capability
- Process Measurement
  - Amadeus Experiments

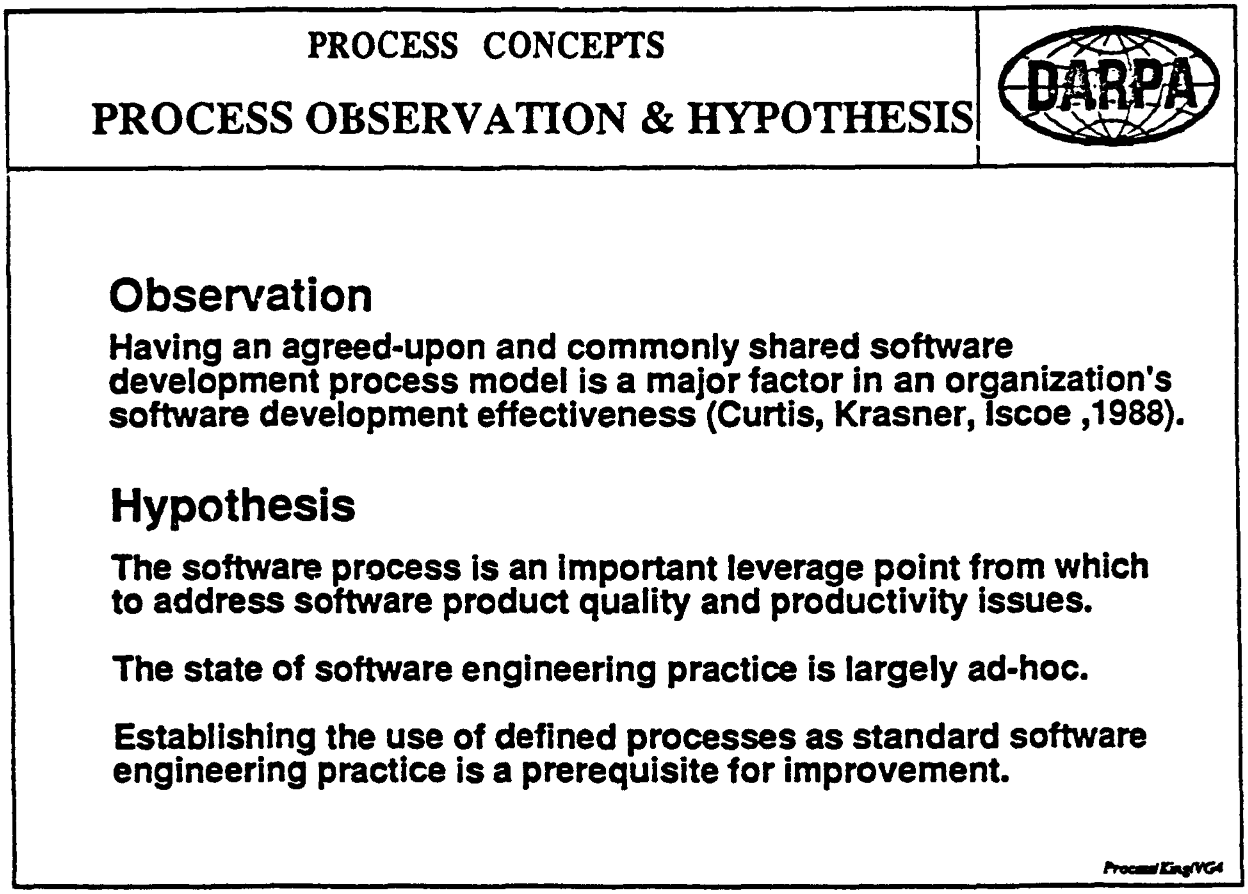
Risk Management Strategy Applied to Capability Improvement and Integration with the Software Engineering Environment

Process-Driven Software Engineering Environment

Process/Eng/VG3

Slide 3: Title: STARS Process Approach

STARS approach for developing the process-driven development capabilities is to use an iterative development, risk-reduction approach which begins with point solutions which are being or have been developed. They include (1) process formalism comparisons which is the topic of a later session in this track, (2) process formalism definition which is used to specify the process definitions used in the Control Point Process Demo, (3) process asset collection, the topic of the next session in this track, (4) enactment mechanisms such as the Control Point and Role-Based solutions which are being demonstrated at this conference and described both here and in a later session in this track, (5) process management capabilities which are discussed both here and in a later session and demonstrated with the Software Process Management System (SPMS) in the exhibit area, (6) process modeling and design capabilities discussed later in this session , and (7) process metric capability which is described in a later session in this track. These concepts and prototypes are evaluated against a risk management strategy in order to prioritize the development activities during each phase. Early solutions and capabilities are integrated as appropriate with the developing SEE in order to evaluate further capabilities and to minimize the risks in developing the process-driven SEE.



PROCESS CONCEPTS

PROCESS OBSERVATION & HYPOTHESIS

DARPA

Observation

Having an agreed-upon and commonly shared software development process model is a major factor in an organization's software development effectiveness (Curtis, Krasner, Iscoe ,1988).

Hypothesis

The software process is an important leverage point from which to address software product quality and productivity issues.

The state of software engineering practice is largely ad-hoc.

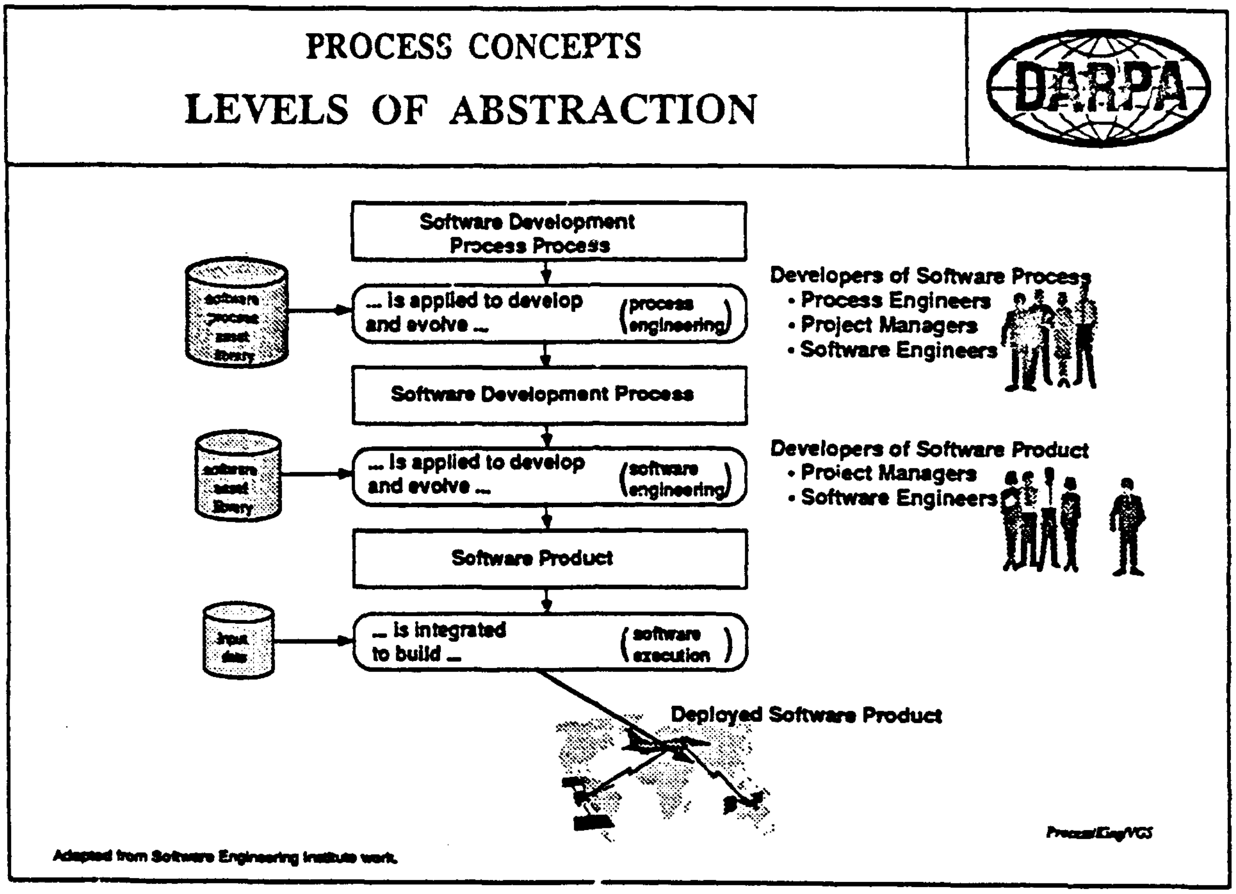
Establishing the use of defined processes as standard software engineering practice is a prerequisite for improvement.

Process/Eng/VG4

Slide 4: Title: Process Observation & Hypothesis

As background for the emphasis on process driven software development, there have been numerous studies of software projects which focused on the effectiveness of an organization to develop quality software products (see Curtis, Krasner, & Iscoe, 1988, Communications of the ACM 31 (11) 1268-87). These studies have identified a strong correlation between the product quality and the presence of an agreed-upon and commonly shared software development process. Recent articles and conference topics have identified the need for defined process to guarantee repeatability, measurability, and adaptability of process definitions in order to facilitate process improvement.

Some of the problems that can result from a lack of an explicit process model are that each software development project must manually perform the tasks necessary to produce project-specific plans, which is susceptible to costly, error-prone sequences of tasks. In addition, the plans are highly variable in content and quality, depending on the individuals involved. By not having a defined process, it is difficult to obtain meaningful measuremen's of the process that is being used so that process improvement cannot be obtained. The variability of processes used from project to project means that any historical data gathered is difficult to correlate and use to predict behavior for another project. Therefore, a defined process supports the STARS objective of getting the processes practiced in a manner that allows measurement, and consequently analysis and improvement which promotes improved product quality.



PROCESS CONCEPTS

LEVELS OF ABSTRACTION

Software Development Process Process

... is applied to develop and evolve ... (process engineering)

Software Development Process

... is applied to develop and evolve ... (software engineering)

Software Product

... is integrated to build ... (software execution)

Deployed Software Product

Developers of Software Process
- Process Engineers
- Project Managers
- Software Engineers

Developers of Software Product
- Proiect Managers
- Software Engineers

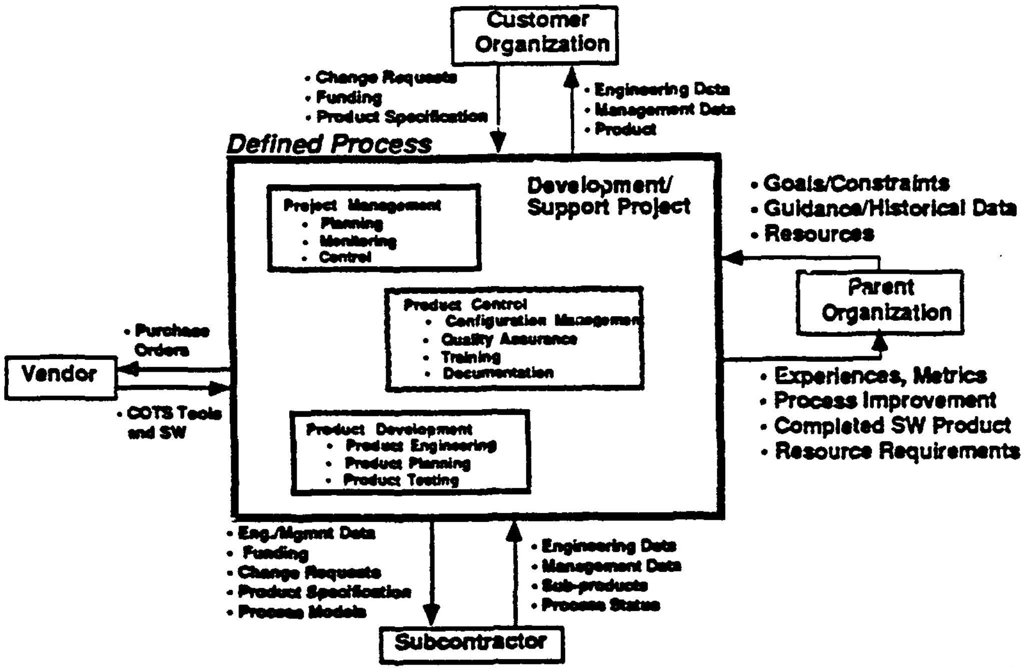
Adapted from Software Engineering Institute work.

Slide 5: Title: Levels of Abstraction

One of the most overused words in recent technical discussions is the word process. It has different meaning to different people. It often takes on different meaning based on usage such as a process for an organization compared to a process for requirements traceability. In this discussion, process is used to reference several different levels of abstraction. Most often process is used to reference the Software Development Process that is used by a project to develop and evolve the software product. However, this Software Development Process is determined by composition, adaptation, and tailoring of process blocks which are maintained in a process reuse library. This composition, adaptation, and tailoring, is itself performed by a process, the Software Development Process Process and instantiated, scheduled, and installed by a Software Process Management Process. In a process-driven development, all of these levels are present. Measurement of the processes in use suggest improvements to the software development process as well as to the higher level Software Development Process Process. As a SEE becomes more process-driven, the need for a specialist in composing processes develops. This specialist defines process blocks and works with project managers and software engineers to adapt the definitions to the needs of the project.

PROCESS CONCEPTS

ORGANIZATIONAL INTERACTION



Customer Organization

- Change Requests
- Funding
- Product Specification

- Engineering Data
- Management Data
- Product

Defined Process

Development/ Support Project

Project Management
- Planning
- Monitoring
- Control

Product Control
- Configuration Management
- Quality Assurance
- Training
- Documentation

Product Development
- Product Engineering
- Product Planning
- Product Testing

- Goals/Constraints
- Guidance/Historical Data
- Resources

Parent Organization

- Experiences, Metrics
- Process Improvement
- Completed SW Product
- Resource Requirements

Vendor

- Purchase Orders

- COTS Tools and SW

- Eng./Mgmnt Data
- Funding
- Change Requests
- Product Specification
- Process Models

- Engineering Data
- Management Data
- Sub-products
- Process Status

Subcontractor

Adapted from Software Productivity Consortium work.

ProcessEng/VG6

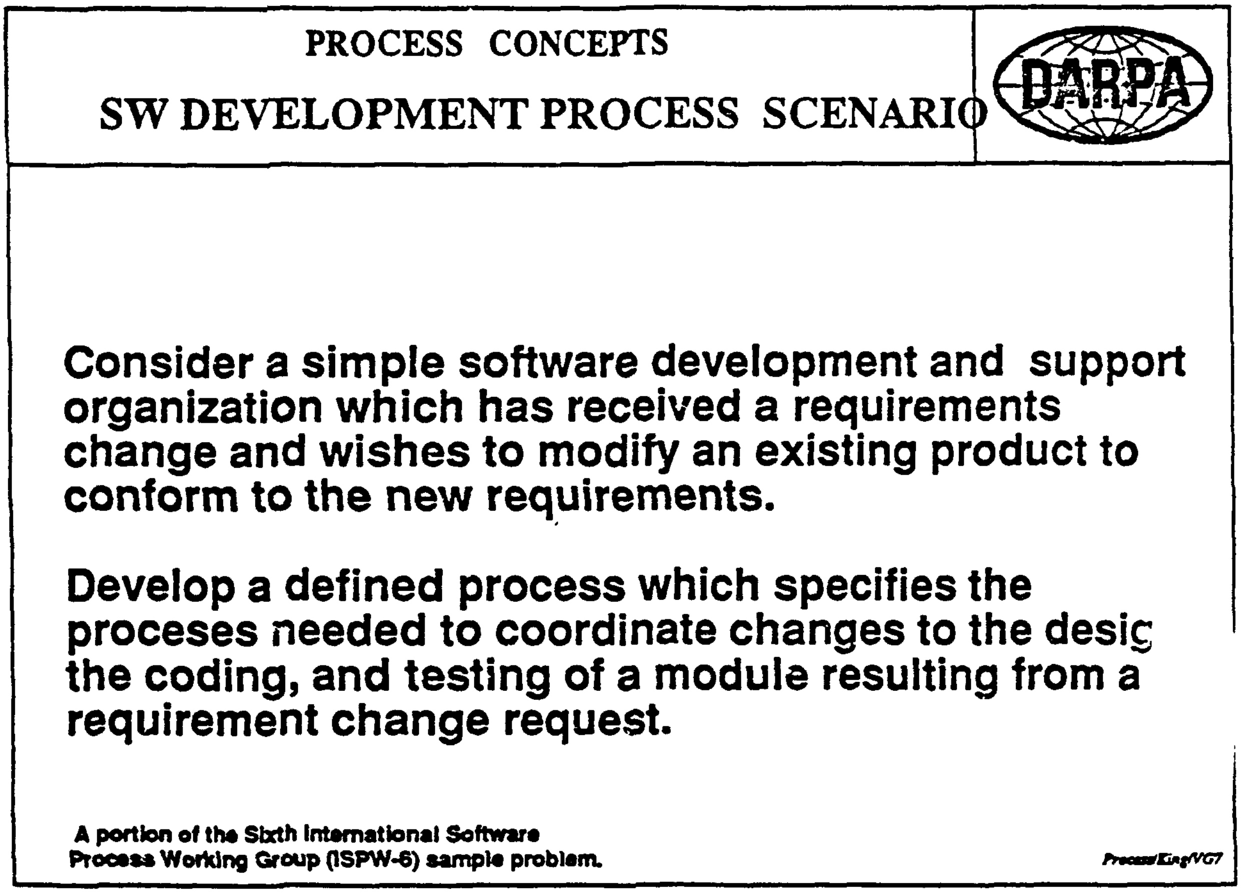
Slide 6: Title: Organizational Interaction

A defined process does not exist isolated from the interactions of the development or support project and the remainder of the organization and external stake-holders. For an organization to improve its ability to produce higher quality products, it is necessary for the defined processes to be adapted and tailored to other projects within the organization. This process use will provide historical data which will contribute to understanding improvements in processes to provide higher quality. In addition the parent organization provides goals and constraints relative to the business interests of the organization as well as resources to support the project.

Many large development projects involve numerous subcontractors that are often geographically dispersed. Processes which are designed to support network-based collaborative development provide measurement capabilities and historical information which can be used to improve the processes and provide lower risk, higher quality strategies for product development. Often, all or portions of the defined process are provided to the subcontractors.

Process activities are often performed by tools, many of which are purchased from vendors. By establishing the interface definitions for the product transformations associated with the activities of a defined process, it is possible to be able to identify new or improved tooling which can be added to or substituted for existing capabilities.

Lastly, by utilizing a defined process, the project can track the development progress more accurately and be able to identify potential risks earlier. This provides the customer with better understanding of the project development through accurate management and engineering data. A defined process also includes a process for change management so that effects of proposed changes can be evaluated accurately and quickly.



PROCESS CONCEPTS

SW DEVELOPMENT PROCESS SCENARIO

Consider a simple software development and support organization which has received a requirements change and wishes to modify an existing product to conform to the new requirements.

Develop a defined process which specifies the proceses needed to coordinate changes to the desig the coding, and testing of a module resulting from a requirement change request.

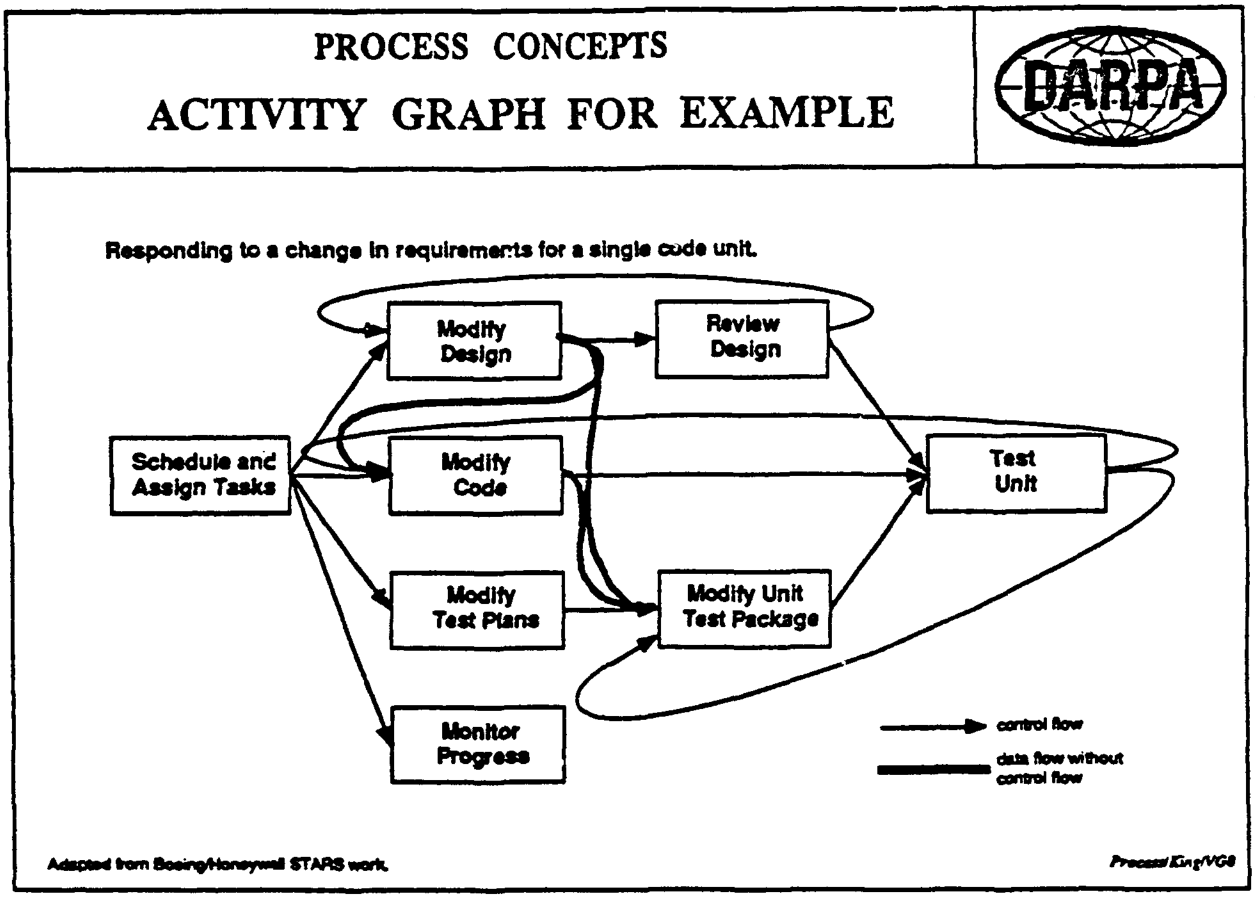
A portion of the Sixth International Software Process Working Group (ISPW-6) sample problem.

ProcessEng/VG7

Slide 7: Title: Software Development Process Scenario

In order to provide a focus for discussing a defined process, let us consider an example which has been used by the process community to examine the adequacy of a process formalism to represent some of the requirements for a viable process-driven environment. The example was developed around several process issues including:

- multiple levels of abstraction
- sequencing, constraints on sequencing, iterative and concurrent activities, looping
- decision points
- feedback
- creative activities
- object management, structure, attributes and interrelationships
- organizational responsibilities
- communication mechanisms
- process measurements
- human and tool enactment
- professional judgement or discretion
- temporal aspects including versioning and scheduling
- planned and optional sequencing between activities
- pre- and post-conditions on activities
- project management and tracking of progress



PROCESS CONCEPTS

ACTIVITY GRAPH FOR EXAMPLE

Responding to a change in requirements for a single code unit.

Adapted from Boeing/Honeywell STARS work.

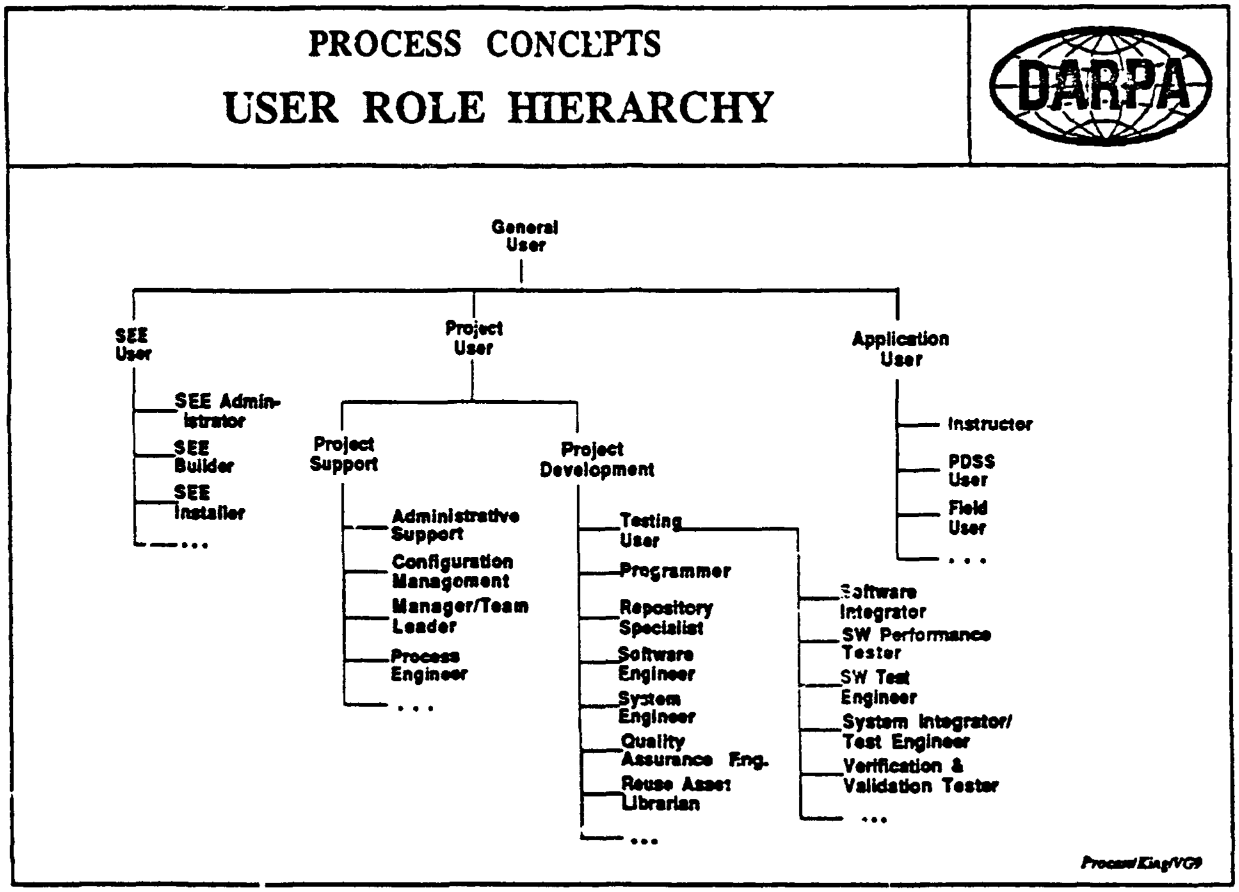
Process/Eng/VG8

Slide 8: Title: Activity Flow for Example

The example scenario identifies a typical organization involving management, software designers and programmers, and testers. In addition there is a requirements change board and a configuration management activity. For simplification, the activities primarily related to the support project are identified in this activity graph.

The support project manager is primarily involved with the scheduling and assignment of tasks to the project team. The manager also monitors the development activities. Software designers are involved with the Modify Design and Review Design activities, programmers with Modify Code, and Test Engineers with Modify Test Plans, Modify Unit Test Package, and Test Unit. The activity boxes aligned vertically can all be performed concurrently according to the example. Some proposed changes may not involve a design modification so that they could progress while other changes causing design modification are incorporated in the design. As a result of testing, there may be changes that need to be made to the test plans that are independent of the design and code. However the Unit Test Package must be modified based on the final versions of the design, code and test plans. Lastly, several steps involved potential iteration before final approval is achieved.

Each of the activities identified in this activity graph themselves represent processes which are interconnected to define the project's defined process. Each of the activities can have both pre- and post-conditions that must be met before progress can continue. For instance, suppose the Support Organization has decided that the Modify Code activity cannot begin until notice from the Modify Design activity is received indicating which parts of the software product are not affected by a design modification. This is a policy that becomes a pre-condition for starting the Modify Code activity and a post-condition for a Change Analysis step within the Modify Design activity. Other policies and conditions can be imposed based on the goals and policies of both the Support Project and the other stake-holders as indicated earlier.



PROCESS CONCEPTS

USER ROLE HIERARCHY

General User

- SEE User
  - SEE Administrator
  - SEE Builder
  - SEE Installer
  - ...
- Project User
  - Project Support
    - Administrative Support
    - Configuration Management
    - Manager/Team Leader
    - Process Engineer
    - ...
  - Project Development
    - Testing User
      - Software Integrator
      - SW Performance Tester
      - SW Test Engineer
      - System Integrator/Test Engineer
      - Verification & Validation Tester
      - ...
    - Programmer
    - Repository Specialist
    - Software Engineer
    - System Engineer
    - Quality Assurance Eng.
    - Reuse Asset Librarian
    - ...
- Application User
  - Instructor
  - PDSS User
  - Field User
  - ...

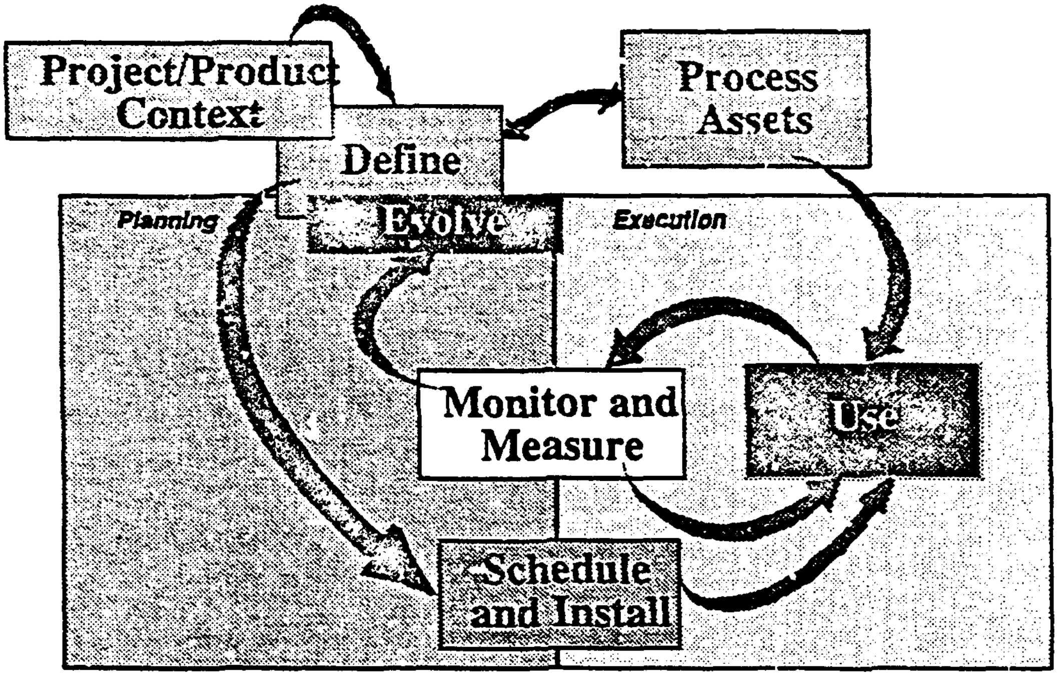
Process/Eng/VG9

Slide 9: Title: User Role Hierarchy

Clearly in a large software development project, there are numerous activities that are often performed by specialists in the related domains. Our approach in the STARS view is to identify specific user roles and determine the products and activities with which they interface. These roles are described through Entity-Relationship models to more accurately represent the role activities. The roles have been grouped into a class hierarchy in order to relate users to related organizational activities. Depending on the size of the project, one person may have several roles to fill or only a portion of the role identified in the hierarchy. The approach to establishing the processes and activities for a SEE are centered on the roles of the users of the SEE. The illustrations in the remainder of this presentation take views associated with different roles.

PROCESS CONCEPTS

PROCESS DRIVEN DEVELOPMENT



Adapted from Software Productivity Consortium work.

Process/EngVG 10

Slide 10: Title: Process Driven Development

To summarize a process-driven development, the process is defined using process examples maintained in a process asset repository and adapted to the requirements of the project and product. This process drives the creation of a plan and the scheduling of resource usage consistent with the execution environment. The instantiated process is then installed in a SEE to achieve a process-driven environment. As the process is used, measurements are periodically collected to assist in the monitoring of the development progress and to evaluate the effectiveness of the process. Analysis of the data from process usage may result in improved process definitions which are then deployed to be scheduled, installed, used, and monitored for further improvements.

During the remainder of this presentation, various user roles will help focus on portions of the process-driven development in order to understand some of the concepts involved.

Up to this point the need to have a defined process for software development that is practiced has been stressed. Many of the activities of software development are performed using computers. By instantiating the defined process activities within the environment, many of the tedious tasks such as monitoring progress can be automated. The execution of the plan under the control of the defined process using the automated techniques for monitoring forms the basis for a process-driven environment.

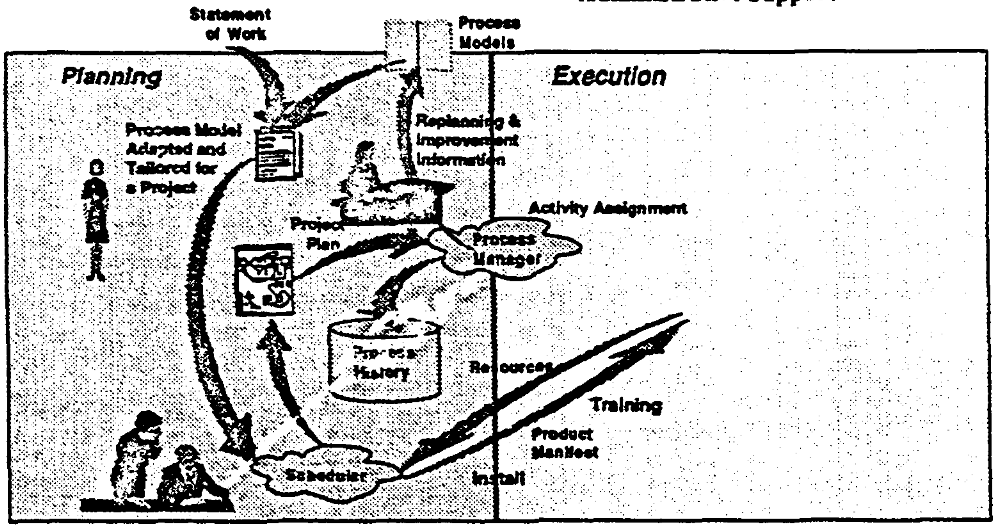
PROCESS CONCEPTS

PROCESS DRIVEN DEVELOPMENT

(Planning)

Software Management Process instantiates
Software Development Process in SEE

Important Roles:
- Project Manager
- Process Engineer
- Administrative Support



Adapted from Software Productivity Consortium work.

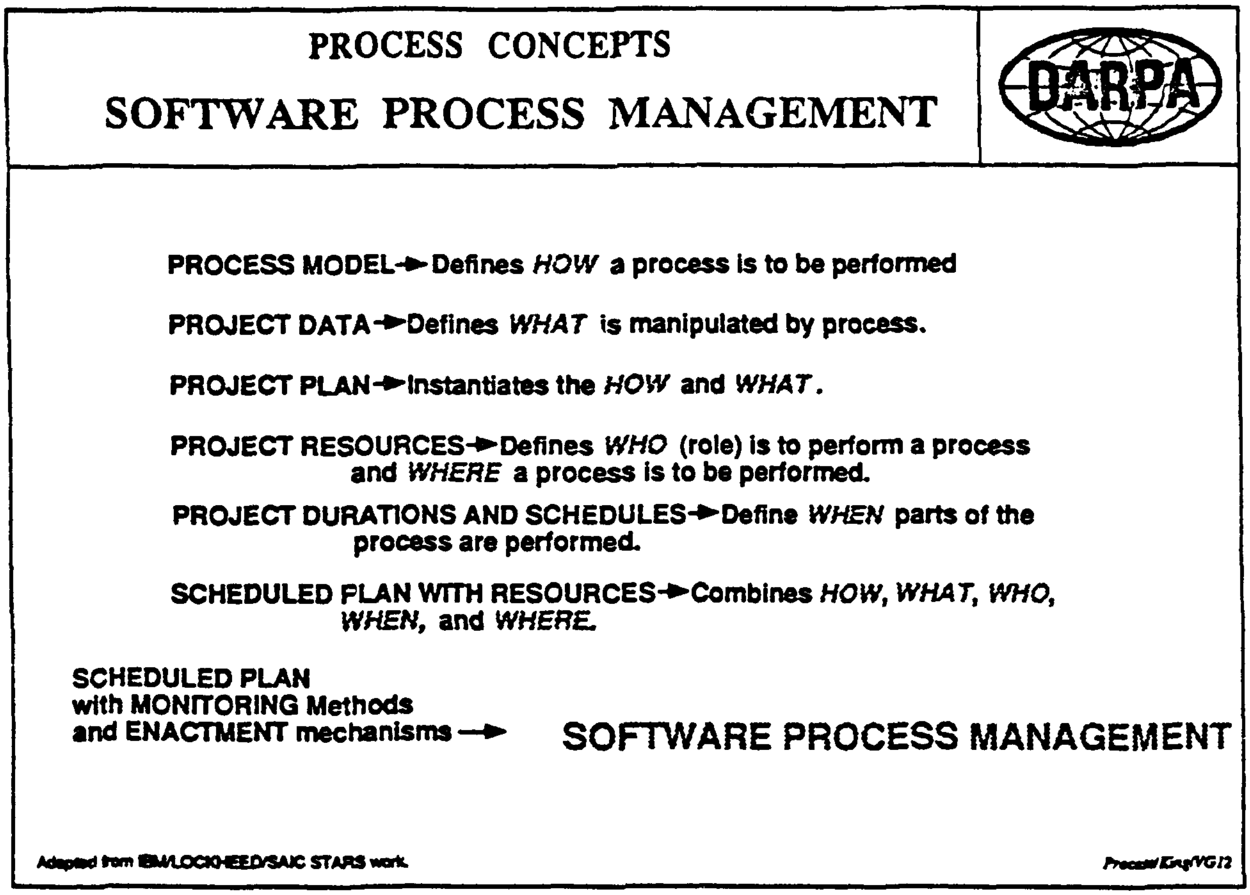
Process/EngVG11

Slide 11: Title: Process Driven Development (Planning)

The concepts of this process driven environment can be pictorially represented in more detail. Let us first look at the details involved in taking a defined process model and developing an instantiated defined process which is scheduled, installed, and used. As the process is enacted, process monitoring accumulates information about the processes and products.

This chart expands part of the definition, scheduling, and process monitoring activities. The activities identified are the primary responsibilities associated with the Project Manager's role.

The project manager is responsible for taking existing defined process models, tailoring and adapting them to the project needs, establishing a plan, allocating resources to establish a schedule, and monitoring the development for conformance with the plan. Replanning activities occur when changes or difficulties are encountered. Process improvement occurs through of analysis of process history and new ideas resulting from process use.



PROCESS CONCEPTS

SOFTWARE PROCESS MANAGEMENT

DARPA

PROCESS MODEL→Defines *HOW* a process is to be performed

PROJECT DATA→Defines *WHAT* is manipulated by process.

PROJECT PLAN→Instantiates the *HOW* and *WHAT*.

PROJECT RESOURCES→Defines *WHO* (role) is to perform a process and *WHERE* a process is to be performed.

PROJECT DURATIONS AND SCHEDULES→Define *WHEN* parts of the process are performed.

SCHEDULED PLAN WITH RESOURCES→Combines *HOW*, *WHAT*, *WHO*, *WHEN*, and *WHERE*.

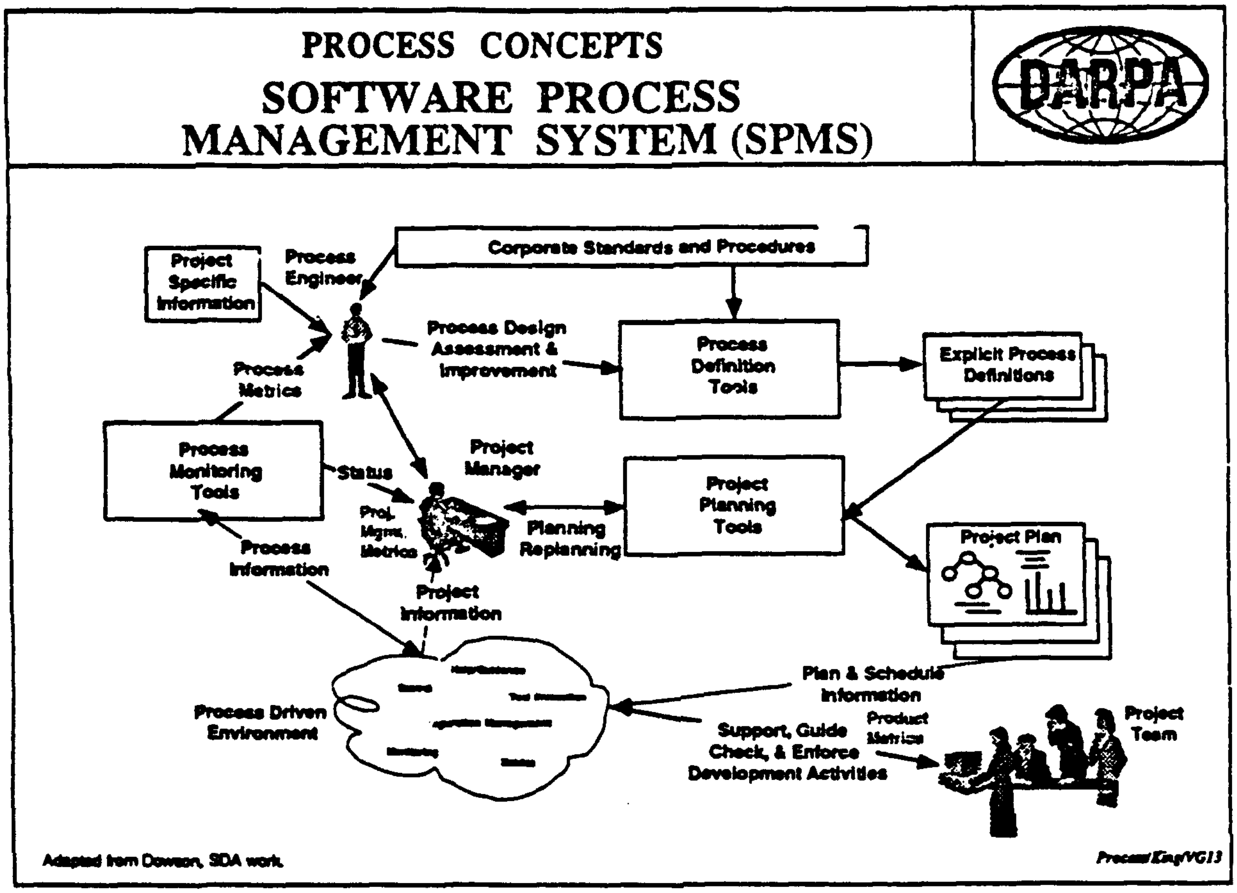
SCHEDULED PLAN with MONITORING Methods and ENACTMENT mechanisms → SOFTWARE PROCESS MANAGEMENT

Adapted from IBM/LOCKHEED/SAIC STARS work.

Process/EngVG12

Slide 12: Title: Software Process Management

Software process management can be simplistically described in terms of the following concepts. A ***process model*** in the most general sense defines ***how*** a process is to be performed. It contains prototypical sequences of tasks that must be done in order to accomplish desired goals. A ***plan*** is defined to be the instantiated and elaborated information produced by combining process model and project specific data. A process model provides the framework for producing plans that can be replicated for specific software development projects. It does not, however, provide the detail necessary for individuals to perform specific tasks to produce specific products of a desired quality for a specific cost or in a defined time frame. Process model information must be combined with project-specific information to create a detailed plan that includes the cost, schedule, and quality requirements. This project specific information includes data on ***what*** is to be built, ***who*** is available to perform the work, and ***where*** the work is to be performed. Schedules based on estimated durations of tasks and available resources provide data as to ***when*** tasks may be started and completed. Combining the project-specific information with the process model data concerning ***how***, provides the basis for conventional project management planning. If this information is combined with automated techniques for monitoring and supporting some of the tasks that individuals must perform in the execution of the plan, then it may form the kernel of a software process management system. The degree to which the software process management system automatically supports and monitors the ongoing process is, in part, determined by the process model.



PROCESS CONCEPTS

SOFTWARE PROCESS MANAGEMENT SYSTEM (SPMS)

DARPA

Corporate Standards and Procedures

Project Specific Information

Process Engineer

Process Design Assessment & Improvement

Process Definition Tools

Explicit Process Definitions

Process Metrics

Process Monitoring Tools

Status

Project Manager

Proj. Mgmt. Metrics

Project Planning Tools

Planning Replanning

Project Plan

Process Information

Project Information

Process Driven Environment

Plan & Schedule Information

Product Metrics

Support, Guide Check, & Enforce Development Activities

Project Team

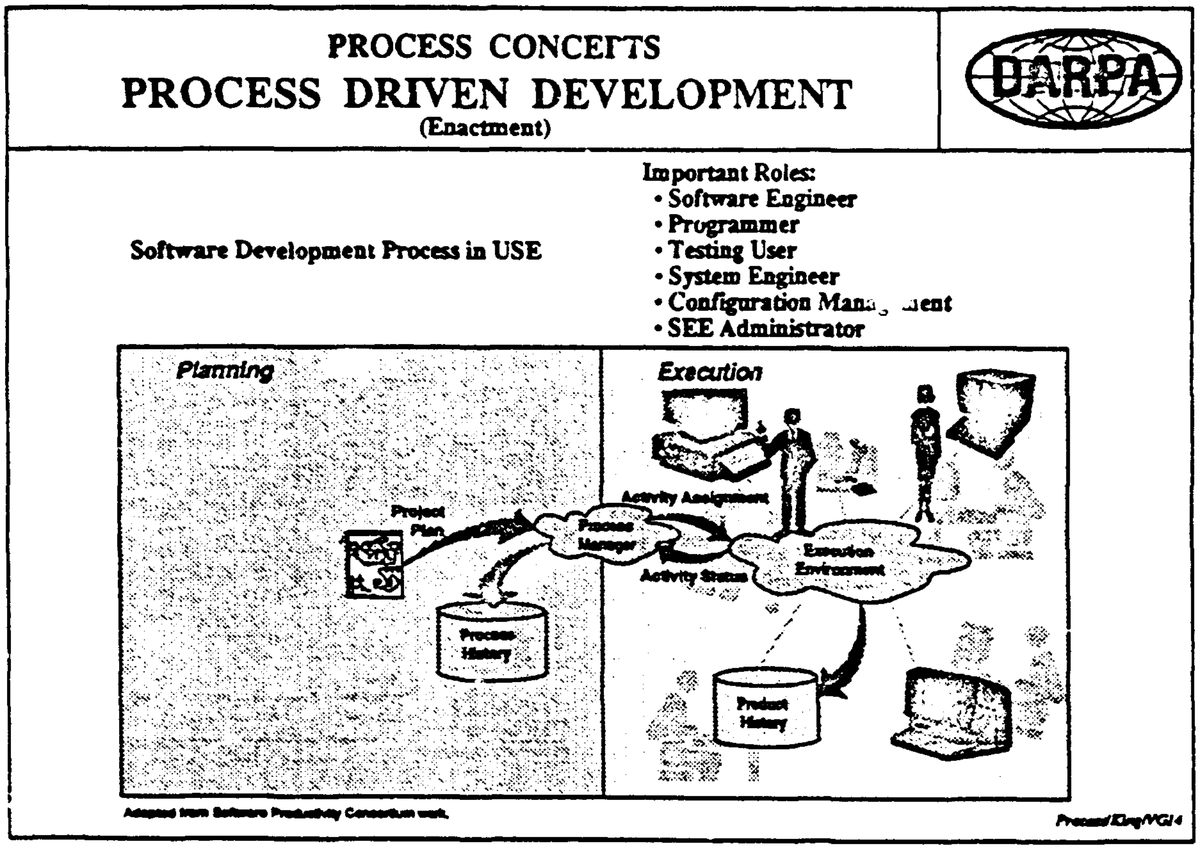
Adapted from Dowson, SDA work.

Process/EngVG13

Slide 13: Title: Software Process Management System (SPMS)

As an example of an early STARS point solution consider the Software Process Management System developed as a STARS Breakthrough Initiative and adopted as part of the IBM STARS strategy for incaporation within the STARS process solutions.

- Focuses on the activities associated with Slide 11.
- Covered in more detail in Session "Software Process Maaagement"
- Illustrated by the SPMS Demonstration in the exhibit hall.



PROCESS CONCEPTS

PROCESS DRIVEN DEVELOPMENT

(Enactment)

DARPA

Software Development Process in USE

Important Roles:

- Software Engineer
- Programmer
- Testing User
- System Engineer
- Configuration Mana‸ ‿ent
- SEE Administrator

Planning
Execution
Project Plan
Process Manager
Process History
Activity Assignment
Activity Status
Execution Environment
Product History

Adapted from Software Productivity Consortium work.

Process/Eng/VG14

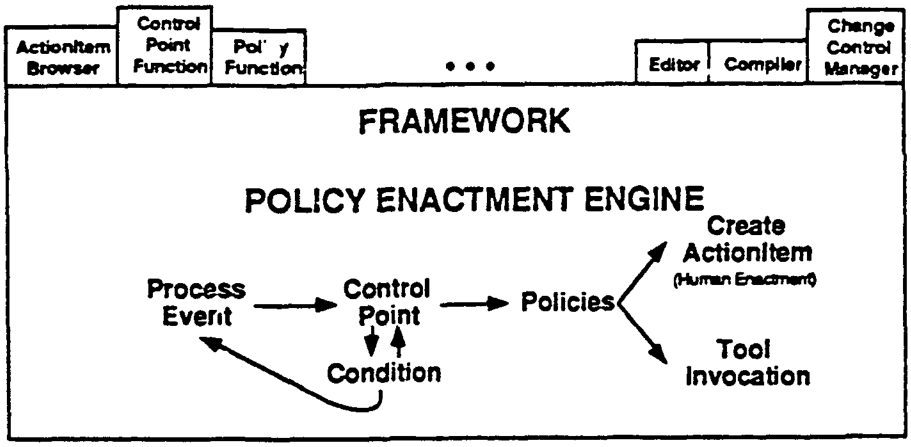
Slide 14: Title: Process Driven Development (Enactment)

Now let's focus on the programmers, software engineers and other users of a process driven development that are directly related to the enactment of the defined process. The enactment concepts of this process driven environment can be pictorially represented in more detail. This chart expands the enactment and monitoring activities. It focuses primarily on the activities associated with the engineers and developers of the project products.

As examples of STARS point solutions, two different strategies have been examined for process enactment. The first of these involves attaching actions to the pre- and post-operations of activities to notify and control the behavior of the SEE. The second involves monitoring activities in the environment and presenting to each user the appropriate current view of the development so that the user may select the next activity from choices which are appropriate to the user's role.

PROCESS CONCEPTS

Control Point Process Enactment



Adapted from Boeing/Honeywell STARS work.

ProcessEng/VG15

S'ide 15: Title: Control Point Process Enactment

The first type of enactment involves some fundamental interaction with the environment framework. Various functions are available for invocation in order to perform the activities related to the process. These include editors, compilers, browsers, and function evaluators. An underlying assumption about this type of enactment is that every operation is performed utilizing a function model which has an entry event and an exit event associated with the body of the function. A history of the events is maintained and the events can trigger actions which are called control points. Further information regarding control points is provided in the short extract from the Process Operation Concepts Document entitled Process Concepts Scenarios, available in the liturature for this track.

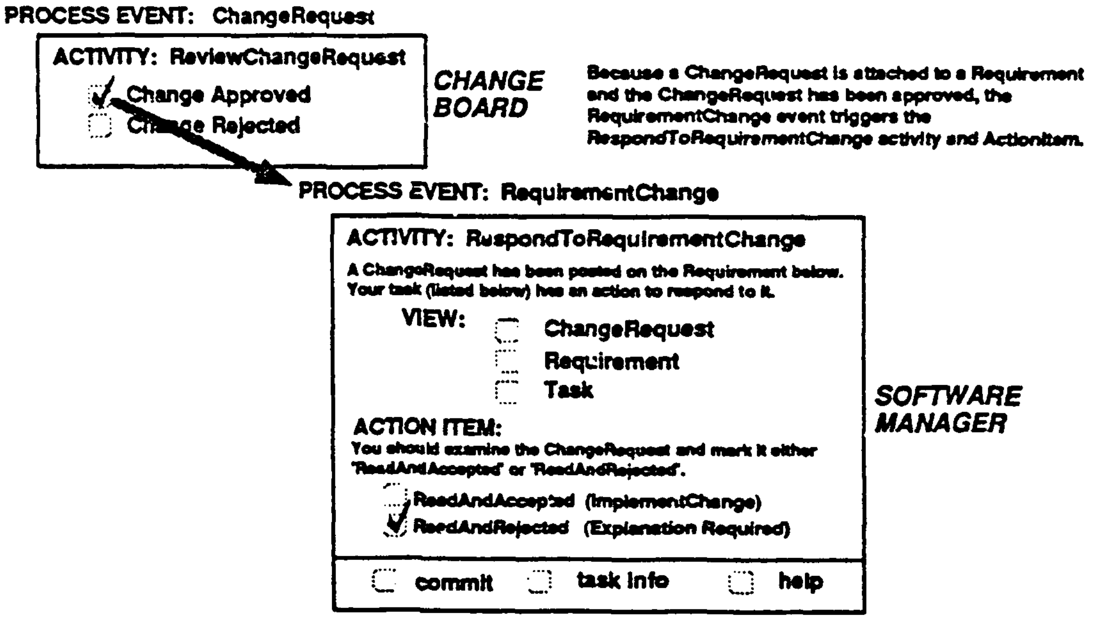
Associated with a control point is an enable/disable parameter which can be dynamically set during execution. A condition based on process and product state variables which are maintained in a persistent store is also associated with the control point. If the control point is enabled and the condition is true then the body of the control point is enacted. Control points are themselves functions so that they have the events and body to express the desired behavior.

Another factor that affects enactment is related to policies that are enforced relative to the specific behavior desired. For instance, an Ada design process may have a policy such as 'only individuals with 5 years of Ada coding experience are allowed to perform this activity'. To determine the truth of this policy, it is necessary to determine who is enacting the activity, what are the qualifications of the user and whether or not the user's qualifications meet the requirements of the policy.

Once the policies are determined to be satisfied, the enactment of the activity can occur. In this view, both humans and tools can be the enactment agents for the activity. Tool invocation occurs through the normal scheduler in the environment. If a human is the agent, the person must be notified about what is expected to be done and why. This is accomplished through an ActionItem. The ActionItem message is sent to the user with appropriate instructions for accomplishing the activity.

PROCESS CONCEPTS

PROCESS USER NOTIFICATION



PROCESS EVENT: ChangeRequest

ACTIVITY: ReviewChangeRequest

☑ Change Approved
□ Change Rejected

CHANGE BOARD

Because a ChangeRequest is attached to a Requirement and the ChangeRequest has been approved, the RequirementChange event triggers the *RespondToRequirementChange* activity and ActionItem.

PROCESS EVENT: RequirementChange

ACTIVITY: RuspondToRequirementChange

A ChangeRequest has been posted on the Requirement below. Your task (listed below) has an action to respond to it.

VIEW:
□ ChangeRequest
□ Requirement
□ Task

ACTION ITEM:
You should examine the ChangeRequest and mark it either 'ReadAndAccepted' or 'ReadAndRejected'.

□ ReedAndAccepted (ImplementChange)
☑ ReedAndRejected (Explanation Required)

□ commit □ task info □ help

SOFTWARE MANAGER

Adapted from STARS Boeing/Honeywell work.

ProcessEngVG16

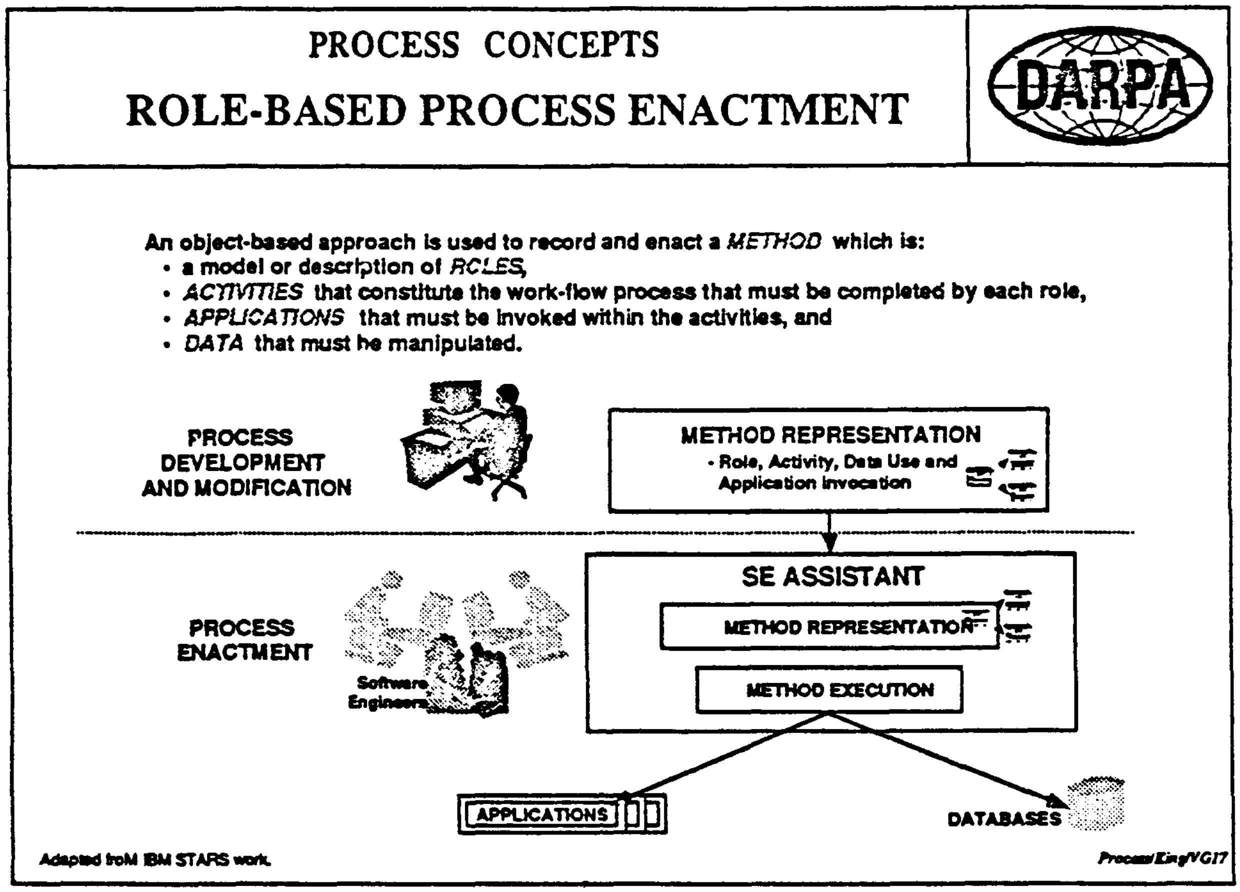
Slide 16: Title: Process User Notification

Assuming a series of ActionItems have resulted from development activities, how would the user interface appear? To facilitate navigating through the ActionItem messages, an ActionItem Browser has been developed which forms the primary user interface. When a user logs into the system, the user may select the ActionItem Browser to inspect activities that have been assigned to the user. Thus the user's view of the system is often through the ActionItem Browser.

Returning to the example described in Slides 7 and 8, consider that a request to change a requirement has just been processed by the Change Board and the Change Approved action has been selected. As part of the completion of this action, an ActionItem is created for the Software Manager to respond to the change request. As a result, the action item illustrated is sent to the manager. The ActionItem Browser allows information related to the ActionItem to be identified and collected so that the manager can review any information before making a decision about the ActionItem. For this illustration, the software manager may view the originating change request, the requirement involved in the change or the tasks involved processing the change. At any time the manager may obtain context sensitive help, review options defined for the manager's task or commit the chosen action.

This action item has identified two actions that can be taken by the manager. The manager may accept the change request and trigger the activities necessary to implement the change or can reject the change which will cause a post-condition to ask for an explanation.

This enactment mechanism has been developed into a STARS point solution and is being demonstrated at this conference by the Boeing/Honeywell STARS team.



PROCESS CONCEPTS

ROLE-BASED PROCESS ENACTMENT

An object-based approach is used to record and enact a *METHOD* which is:

- a model or description of *RCLES*,
- *ACTIVITIES* that constitute the work-flow process that must be completed by each role,
- *APPLICATIONS* that must be invoked within the activities, and
- *DATA* that must he manipulated.

PROCESS DEVELOPMENT AND MODIFICATION

METHOD REPRESENTATION
- Role, Activity, Data Use and Application invocation

PROCESS ENACTMENT

Software Engineers

SE ASSISTANT

METHOD REPRESENTATION

METHOD EXECUTION

APPLICATIONS

DATABASES

Adapted froM IBM STARS work.

Process/EngVG17

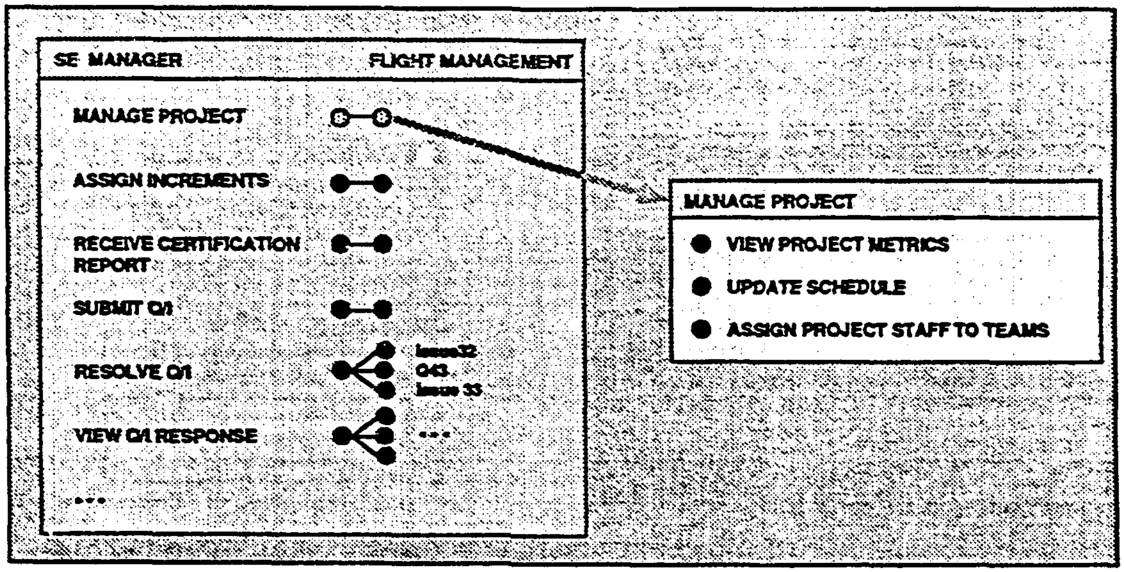
Slide 17: Title: Role-Based Process Enactment

The second mechanism of process enactment is based on guidance provided in a role-sensitive context. Many methodologies associated with software development identify the activities associated with different user roles associated with the methodology. These defined activities can be incorporated into a process definition which supports the methodology and can be organized around the associated roles. In terms of Role-Based Process Enactment, the role specific subprocess is called a ***method***. The ***method*** describes the activities, applications, and data that are associated with the role. The ***method*** is then instantiated in the environment along with process monitoring capabilities to enable user process guidance. When a user logs into the system, the user is assigned to a role. The user may change roles if he has proper authority. The steps in the enactment of the process are selected by the user until the task is completed.

This mechanism is being demonstrated by the STARS IBM team as the Cleanroom Engineering Process Assistant (CEPA). It is discussed in more detail in the Software Process Management presentation later in this track.

PROCESS CONCEPTS

SE MANAGER ROLE ENACTMENT VIEW

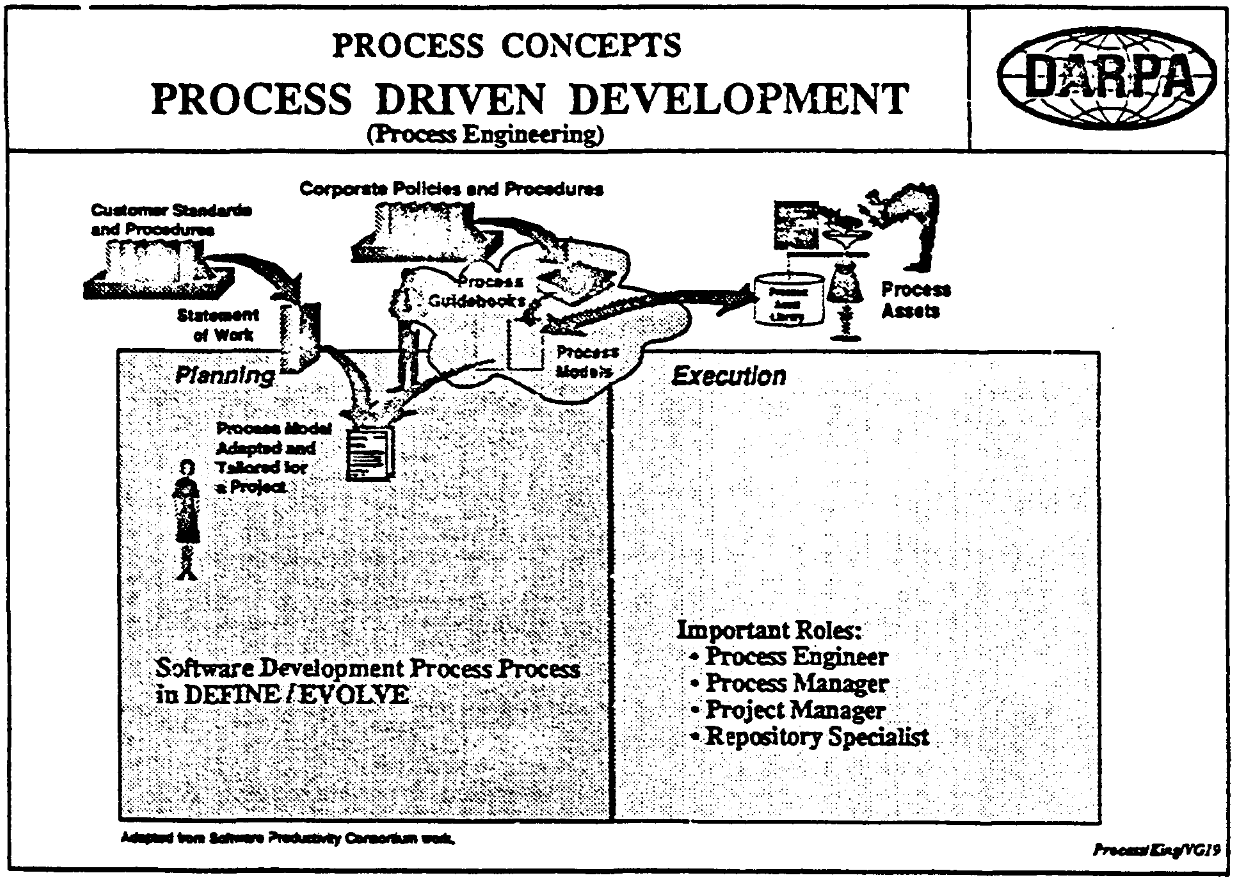


Adapted from IBM STARS work.

Process/KingVG18

Slide 18: Title: Software Engineering Manager Enactment View

Briefly, the available activities that can be performed by the user are presented through a graphical interface. The user selects an activity which results in either a subprocess activity of selecting from specific s. bactivities or results in direct invocation of applications, tailored to the specific needs of the user and current statu* of the project.



PROCESS CONCEPTS

PROCESS DRIVEN DEVELOPMENT

(Process Engineering)

DARPA
Corporate Policies and Procedures
Customer Standards and Procedures
Statement of Work
Process Guidebooks
Process Models
Process Asset Library
Process Assets
Planning
Execution
Process Model Adapted and Tailored for a Project

Software Development Process Process in DEFINE / EVOLVE

Important Roles:
- Process Engineer
- Process Manager
- Project Manager
- Repository Specialist

Adapted from Software Productivity Consortium work.

Process/Eng/VG19

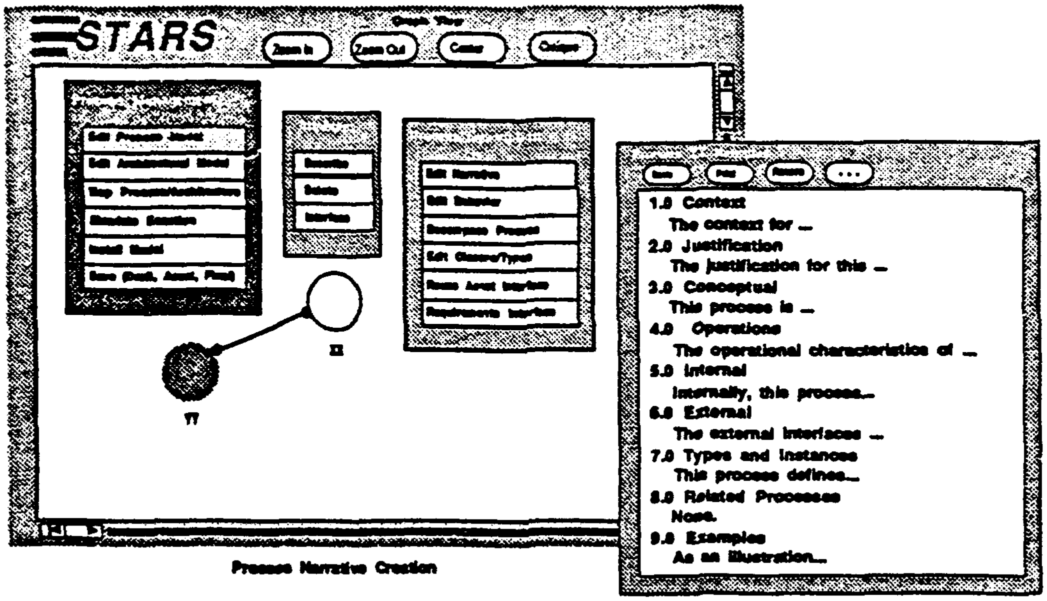
Slide 19: Title: Process Driven Development (Process Engineering)

So far we have focused on project managers and the programmers, software engineers and other users of a process driven development. An important part of a process-driven environment is the creation and modification of process definitions and models. STARS has identified this as the process engineering role. The process engineering concepts of the environment can be pictorially represented in more detail. This chart expands the definition activity and focuses primarily on the activities associated with the process engineers who create and adapt process assets, process models and tailor the process models for use by a project.

The development of process assets is the focus of the Process Asset Library presentation in this track. The development of process models has been investigated using two different approaches. The IBM/SAIC SPMS prototype previously discussed in slide 13 and part of the Software Process Management presentation incorporates composition of process models from process assets. A second approach involves developing a behavior model of the processes using approaches similar to system engineering concept development or software engineering CASE design.

PROCESS CONCEPTS

PROCESS MODELING & DESIGN CAPABILITY



1.0 Context
The context for ...
2.0 Justification
The justification for this ...
3.0 Conceptual
This process is ...
4.0 Operations
The operational characteristics of ...
5.0 Internal
Internally, this process...
6.0 External
The external interfaces ...
7.0 Types and Instances
This process defines...
8.0 Related Processes
None.
9.0 Examples
As an illustration...

Process Narrative Creation

Portions adapted from International Software Systems Inc. work.

ProcessEng/VG20

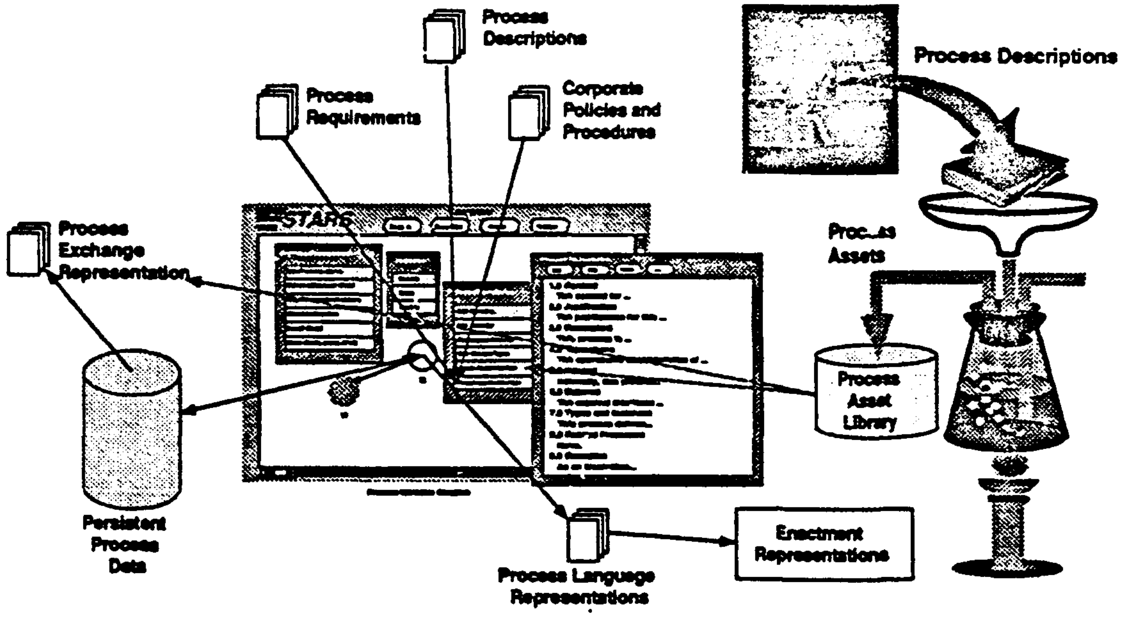
Slide 20: Title: Process Composition

The details associated with this approach are documented in a short extract from the draft Process Operational Concepts Document entitled, Process Concepts Scenarios available in the lituramre for this track.

PROCESS CONCEPTS

PROCESS ENGINEERING ACTIVITY INTEGRATION

DARPA

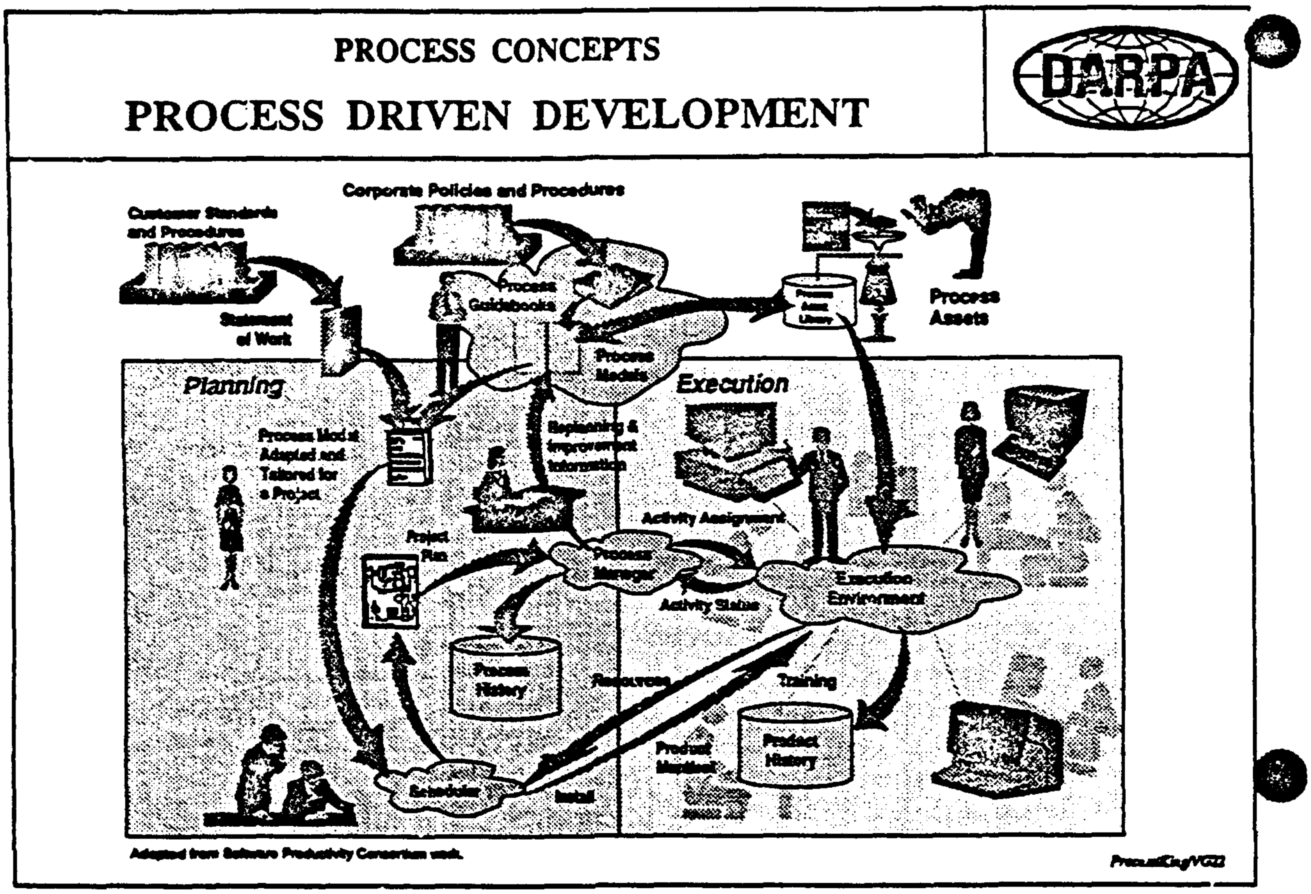


Portions adapted from International Software Systems Inc. work.

ProcessEng/VF21

Slide 21: Title: Process Engineering Activity Integration

Process engineering activities are interrelated with many other activities in the SEE. As an integration point for the activities of all the STARS participants, process engineering tries to compose a defined process which is used, measured and evolved so that higher quality products can be predictably developed. STARS is developing three instances of SEEs. Each of these SEEs must be able to enact the defined processes selected by the project for development of a software system. Individually, each environment performs similar activities. Corporately, the environments utilize assets and contribute process assets to a common Process Asset Library. The intent is to develop process support capabilities that can be instantiated on each of the instances of the SEE. This provides coordination between each of the programs and a combined value-added contribution which exceeds each participants individual contribution.



PROCESS CONCEPTS

PROCESS DRIVEN DEVELOPMENT

Adapted from Software Productivity Consortium work.

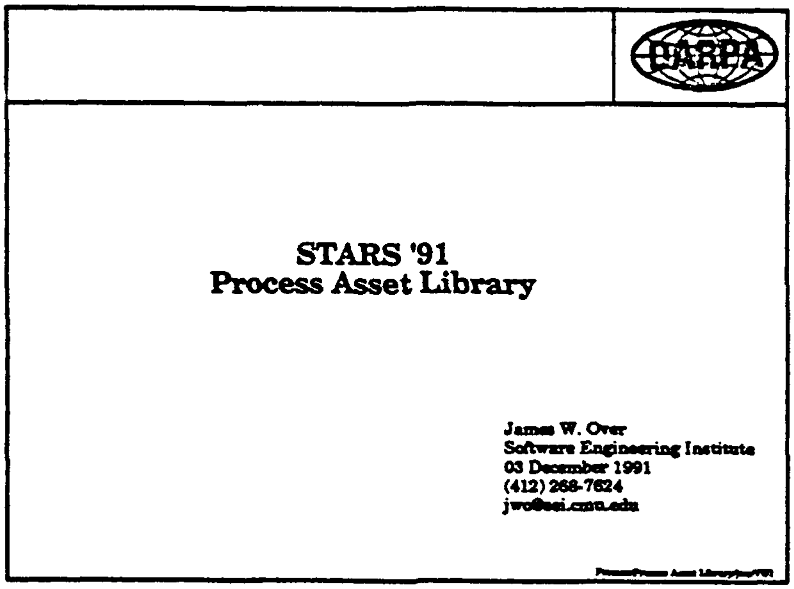
ProcessEngVG22

Slide: 22 Title: Process Driven Development

Tying all of the parts together, the high level pictorial view of a process-driven environment is obtained.

What will STARS accomplish?

- Early Point Solutions
- Risk reduction activities to understand complexity
- Planned product evolution and incorporation into the SEE
- Tailorable process assets
- Concepts tested on demonstration project and refined



STARS '91
Process Asset Library

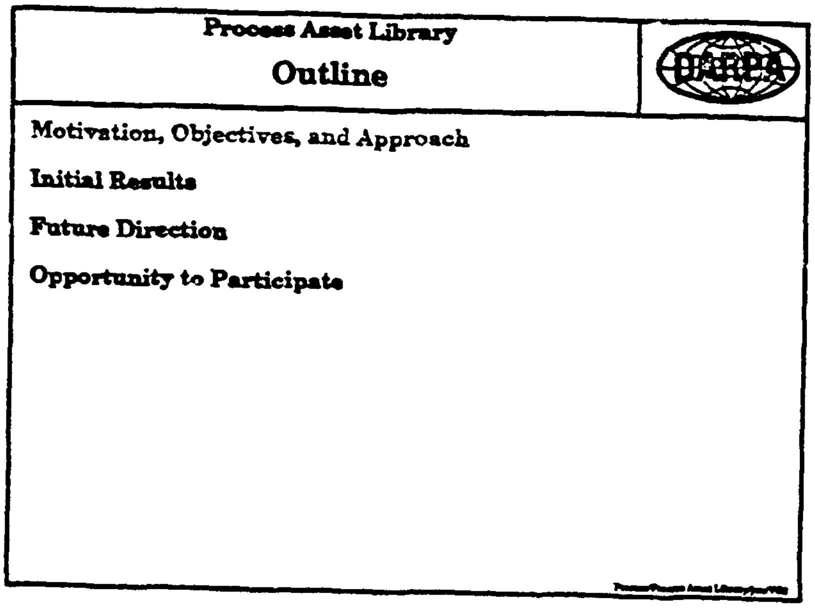
James W. Over
Software Engineering Institute
03 December 1991
(412) 268-7624
jwo@sei.cmu.edu

Abstract

Based on the results of Software Engineering Institute (SEI) assessments, the state of software engineering practice is immature. As a consequence, software cost and schedule are largely unpredictable and quality is lacking. The software process provides an important leverage point from which to address software productivity and quality. To mature and improve the software process will require the use of defined processes as standard software engineering practice.

One way to leverage this capability (defined processes) into widespread practice is to make tailorable, adaptable examples of experience-tested software processes readily available. In collaboration with the STARS prime contractors, the SEI is leading a joint effort to develop a library of reusable software engineering processes. Together, the SEI and STARS prime contractors will demonstrate the benefits of reusable process assets within the STARS process-centered environments.

This presentation provides an overview of the effort to develop the asset library, and initial results of a recent STARS/SEI workshop on process asset library concepts.



Process Asset Library

Outline

Motivation, Objectives, and Approach

Initial Results

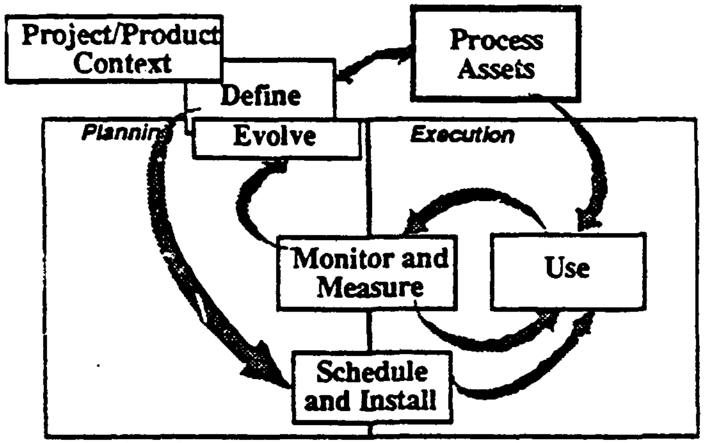
Future Direction

Opportunity to Participate

Process/Process Asset Library/[illegible]

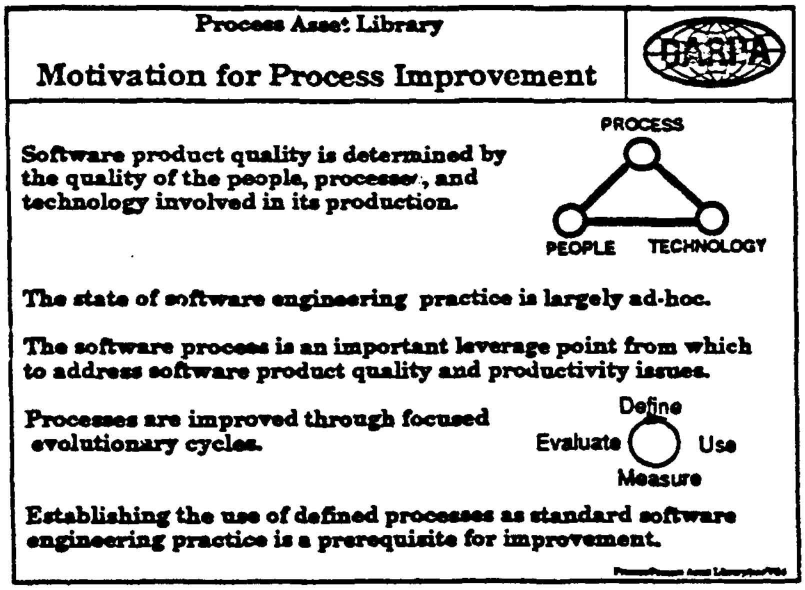
Process Asset Library

Process Driven Development



Adapted from Software Productivity Consortium work.

[illegible]



Process Asset Library

Motivation for Process Improvement

Software product quality is determined by the quality of the people, processes, and technology involved in its production.

PROCESS
PEOPLE
TECHNOLOGY

The state of software engineering practice is largely ad-hoc.

The software process is an important leverage point from which to address software product quality and productivity issues.

Processes are improved through focused evolutionary cycles.

Establishing the use of defined processes as standard software engineering practice is a prerequisite for improvement.

Many software organizations are facing the critical challenge of developing quality software in a reliable and predictable manner. People, technology, and process are the three leverage points that organizations have to meet this critical challenge.

People are currently our most important resource. Personnel/team capability is the most significant cost driver in software development . But even the best people require the infrastructure provided by a disciplined process in order to do their work.

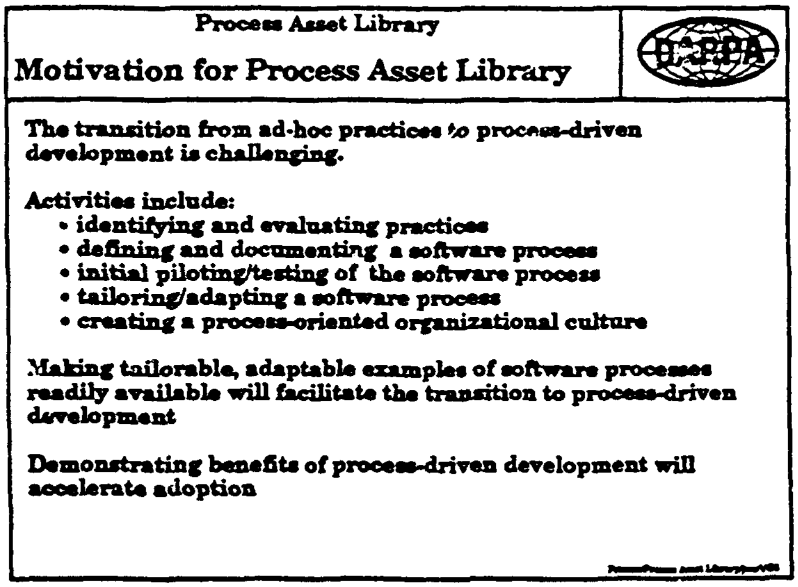
Technology has, and will continue to enable, the development of better quality software products. But the effective use of technology requires that technology be woven into the fabric of the software process.

Based on the results of SEI assessments, the state of the practice is largely ad-hoc. There is little process discipline, and low process fidelity (adherence to defined process).

Process has been proven to be a very effective quality leverage point in other indurtries, and early result from software process improvement efforts have demonstrated comparable benefits. Product defects are halved or better, return on investment is on the order of 7 to 1.

Because many organizations have the most trouble defining and executing the steps that transform user needs into a software product (i.e. the software process), focusing on installing a defined process can provide substantial benefit while maximizing the effectiveness of existing technology and people.

To improve, an organization must get defined processes into practice, then begin the measurement and evaluation cycle that leads to continuous process improvement.



Process Asset Library

Motivation for Process Asset Library

The transition from ad-hoc practices to process-driven development is challenging.

Activities include:

- **identifying and evaluating practices**
- **defining and documenting a software process**
- **initial piloting/testing of the software process**
- **tailoring/adapting a software process**
- **creating a process-oriented organizational culture**

Making tailorable, adaptable examples of software processes readily available will facilitate the transition to process-driven development

Demonstrating benefits of process-driven development will accelerate adoption

Establishing the use of defined process as a standard software engineering practice is difficult and expensive. There are many challenging activities, including:

- identifying and evaluating practices

- defining and documenting an effective software process including a process framework, process models, standards, templates, guidebooks and training

- piloting or testing the process to ensure usability and applicability

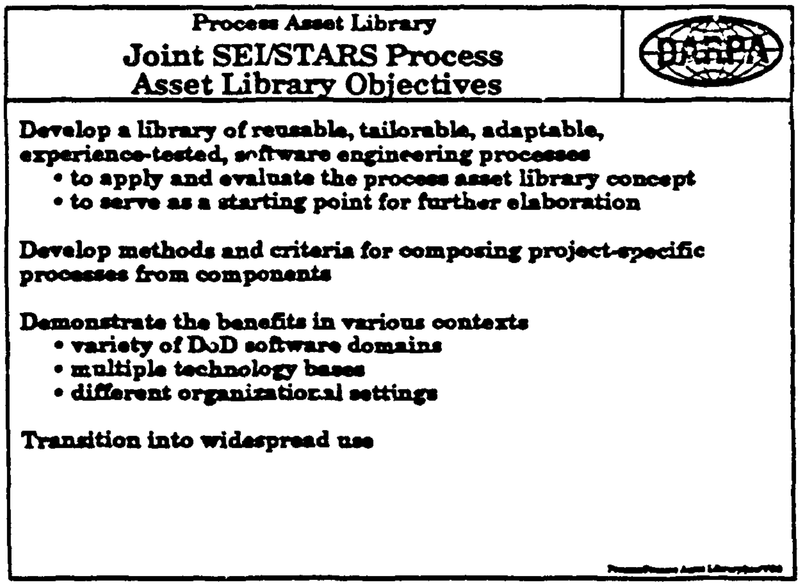
- creating an organizational culture that is based on a disciplined approach

One strategy that reduces the risk of meeting this organizational challenge is to:

- undertake at the national level a risk reducing exercise that will make available experience-tested models and examples of software process, and supporting materials that facilitate transition

- and demonstrate, through application, the benefits of this approach to convince others to take the risk

By establishing and applying a library of reusable process assets, this joint STARS/SEI effort can accelerate the adoption of defined processes as a standard software engineering practice, and thereby meet its mission objectives with respect to process-driven development



Process Asset Library

Joint SEI/STARS Process Asset Library Objectives

Develop a library of reusable, tailorable, adaptable, experience-tested, software engineering processes
- **to apply and evaluate the process asset library concept**
- **to serve as a starting point for further elaboration**

Develop methods and criteria for composing project-specific processes from components

Demonstrate the benefits in various contexts
- **variety of DoD software domains**
- **multiple technology bases**
- **different organizational settings**

Transition into widespread use

To meet this need, the SEI and STARS have undertaken this joint activity to develop a prototype of a Process Asset Library to apply and evaluate this "accelerate by example" transition strategy.

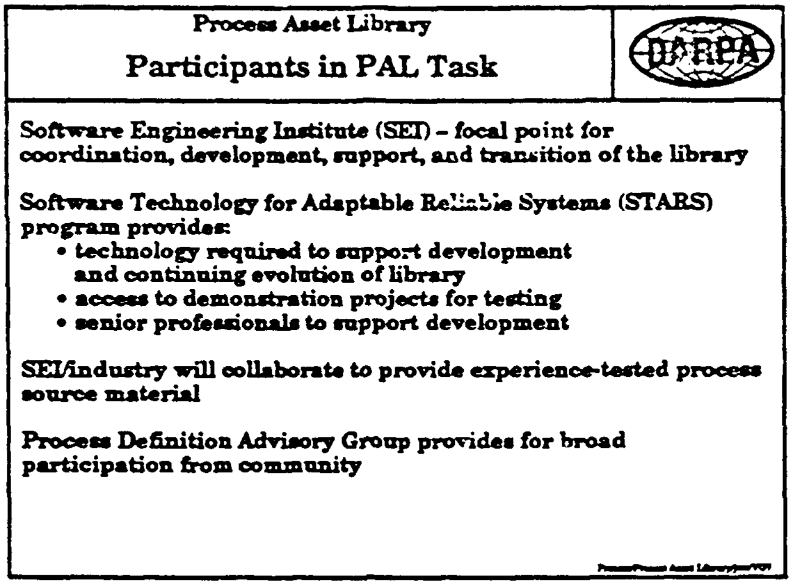
The work will serve as a starting point for further elaboration and refinement, guided by the evaluation of this initial application.

This effort will include the collection, cataloguing, analysis, partitioning, distillation, and synthesis of software processes submitted by industry, government, and academic organizations.

Other processes, methods, and criteria will be developed to support the composition, tailoring, adaptation, installation, and evolution of the library's process assets.

The SEI and STARS will participate in the use of the PAL on STARS demonstration projects in order to demonstrate the benefits of this approach.

As the library and methods mature, an effort to transition this approach into widespread use will be undertaken.



Process Asset Library

Participants in PAL Task

DARPA

Software Engineering Institute (SEI) - focal point for coordination, development, support, and transition of the library

Software Technology for Adaptable Reliable Systems (STARS) program provides:

- **technology required to support development and continuing evolution of library**
- **access to demonstration projects for testing**
- **senior professionals to support development**

SEI/industry will collaborate to provide experience-tested process source material

Process Definition Advisory Group provides for broad participation from community

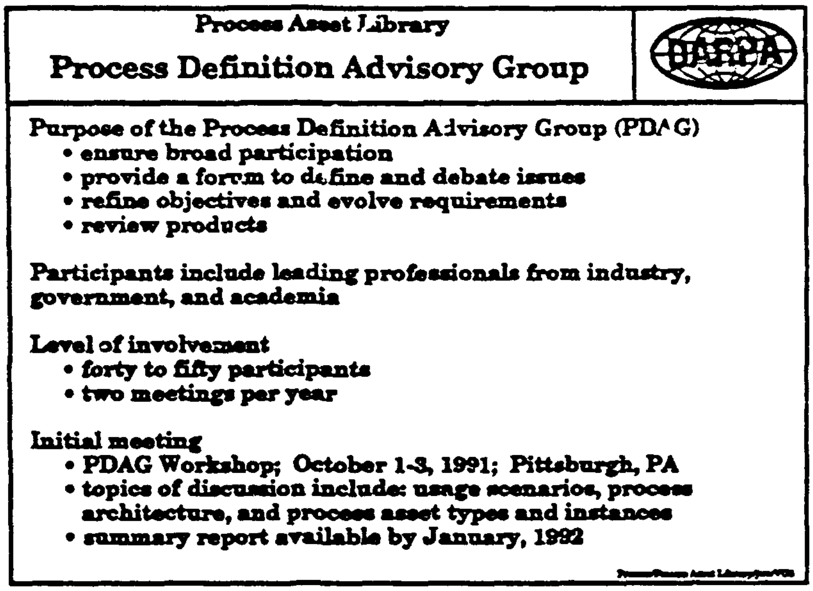
To be successful, an effort of this type requires broad participation. Many varied capabilities must be assembled and applied to the task. STARS and SEI working together have the combined organizational characteristics required.

The SEI's mission, and on-going work in the process area, including process capability maturity modeling, process assessment and evaluation, and process definition, metrics and improvement work play an important role in this task. The SEI is therefore well-positioned to serve as a focal point for the coordination, development, support, and transition activities

The STARS mission in the shift to megaprogramming, especially its focus on process and supporting technology provide a capability for maturing the required technology base and providing access to pilot projects for testing.

The SEI and industry will collaborate to make available the experience-tested processes that will be used as source material for the library

The Process Definition Advisory Group will provide the broad community participation that will be necessary to evaluate and transition the library, in order to achieve success.



Process Asset Library

Process Definition Advisory Group

Purpose of the Process Definition Advisory Group (PDAG)
- ensure broad participation
- provide a forum to define and debate issues
- refine objectives and evolve requirements
- review products

Participants include leading professionals from industry, government, and academia

Level of involvement
- forty to fifty participants
- two meetings per year

Initial meeting
- PDAG Workshop; October 1-3, 1991; Pittsburgh, PA
- topics of discussion include: usage scenarios, process architecture, and process asset types and instances
- summary report available by January, 1992

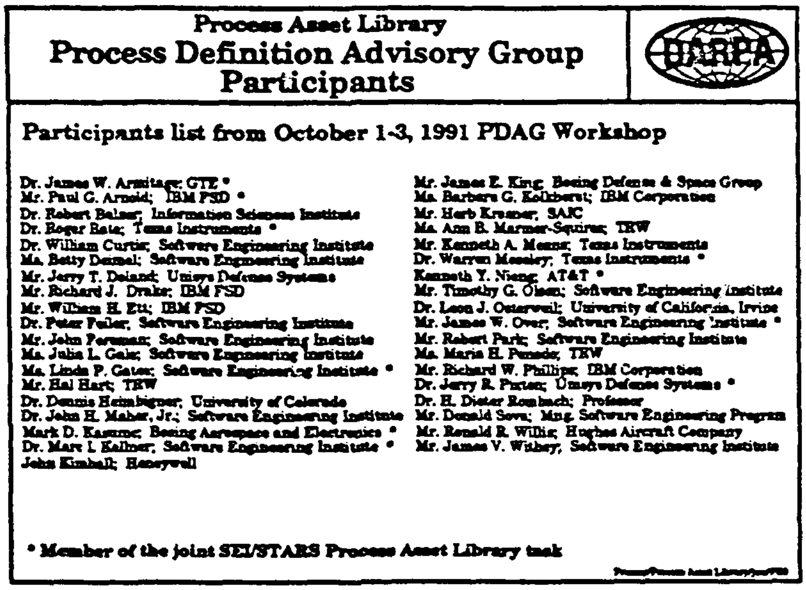
The Process Definition Advisory Group (PDAG) was convened to ensure broad participation in the Process Asset Library (PAL) effort, and to provide a forum for defining and debating issues related to the task.

The PDAG also participates in the refinement and evolution of objectives and requirements for the PAL, and in product review

Over fifty people were invited to the the first meeting, forty attended. Response to the first meeting justifies the creation of a PDAG correspondents mailing list. These members will receive copies of the PDAG meeting summary reports, and are invited to provide feedback.

The initial meeting was held on October 1 to 3, at the Holiday Inn, Pittsburgh, PA.

This meeting was conducted as a workshop, to define issues, objectives, and requirements in three areas:
- PAL usage scenarios
- Process/PAL architectural concepts
- Process asset types and instances



Process Asset Library

Process Definition Advisory Group Participants

Participants list from October 1-3, 1991 PDAG Workshop

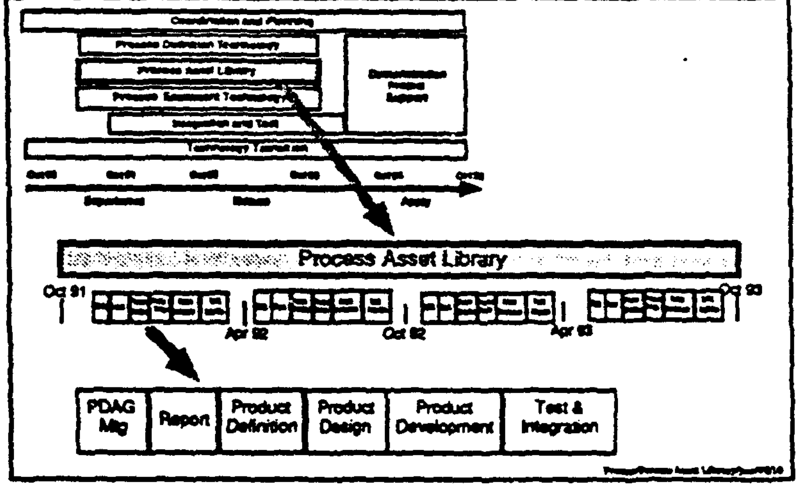
Dr. James W. Armitage; GTE *
Mr. Paul G. Arnold; IBM FSD *
Dr. Robert Balzer; Information Sciences Institute
Dr. Roger Bate; Texas Instruments *
Dr. William Curtis; Software Engineering Institute
Ms. Betty Deimel; Software Engineering Institute
Mr. Jerry T. Doland; Unisys Defense Systems
Mr. Richard J. Drake; IBM FSD
Mr. William H. Ett; IBM FSD
Dr. Peter Feiler; Software Engineering Institute
Mr. John Foreman; Software Engineering Institute
Ms. Julia L. Gale; Software Engineering Institute
Ms. Linda P. Gates; Software Engineering Institute *
Mr. Hal Hart; TRW
Dr. Dennis Heimbigner; University of Colorado
Dr. John H. Maher, Jr.; Software Engineering Institute
Mark D. Kasunic; Boeing Aerospace and Electronics *
Dr. Marc I. Kellner; Software Engineering Institute *
John Kimball; Honeywell

Mr. James E. King; Boeing Defense & Space Group
Ms. Barbara G. Kolkhorst; IBM Corporation
Mr. Herb Krasner; SAIC
Ms. Ann B. Marmor-Squires; TRW
Mr. Kenneth A. Means; Texas Instruments
Dr. Warren Moseley; Texas Instruments *
Kenneth Y. Nieng; AT&T *
Mr. Timothy G. Olson; Software Engineering Institute
Dr. Leon J. Osterweil; University of California, Irvine
Mr. James W. Over; Software Engineering Institute *
Mr. Robert Park; Software Engineering Institute
Ms. Maria H. Penedo; TRW
Mr. Richard W. Phillips; IBM Corporation
Dr. Jerry R. Pixton; Unisys Defense Systems *
Dr. H. Dieter Rombach; Professor
Mr. Donald Sova; Mng. Software Engineering Program
Mr. Ronald R. Willis; Hughes Aircraft Company
Mr. James V. Withey; Software Engineering Institute

* Member of the joint SEI/STARS Process Asset Library task

Process Asset Library

Process Asset Library Task Plan



The PAL task is one of seven elements of the STARS Process effort planned through October of 1995. These elements are:

- Coordination and planning
- Process definition technology
- Process asset library
- Process enactment technology
- Integration and test
- Demonstration project support
- Technology transition

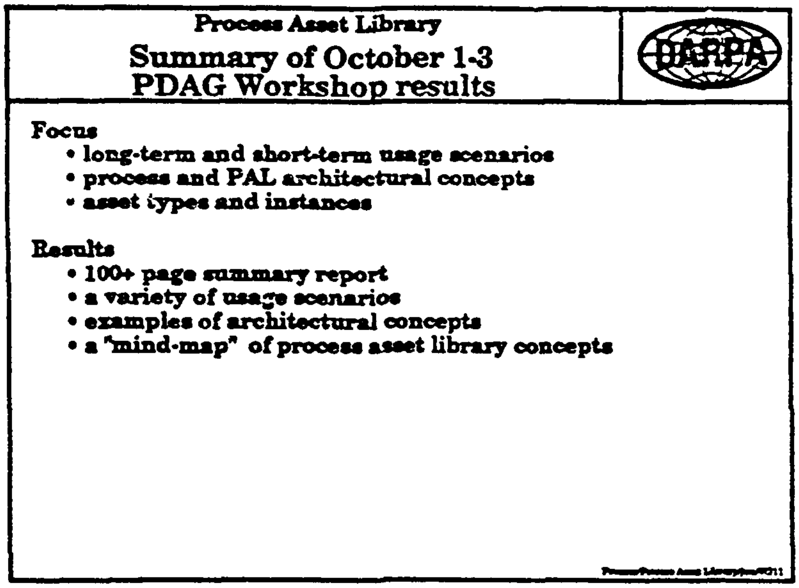
STARS technology is being evolved by experimenting with point solutions, then integrating and maturing these through early use and application. An iterative development model supports this evolution

The PAL effort will also be done iteratively, producing several iterations or increments. Each increment will consist of four product engineering activities:

- Product definition
- Product design
- Product development
- Test and integration

PDAG meetings at regular intervals will provide guidance to the development effort, and review of the work products.

Though depicted here as four equal-length phases of six months each, later increments may be longer resulting in fewer increments.



Process Asset Library

Summary of October 1-3 PDAG Workshop results

DARPA

Focus

- long-term and short-term usage scenarios
- process and PAL architectural concepts
- asset types and instances

Results

- 100+ page summary report
- a variety of usage scenarios
- examples of architectural concepts
- a "mind-map" of process asset library concepts

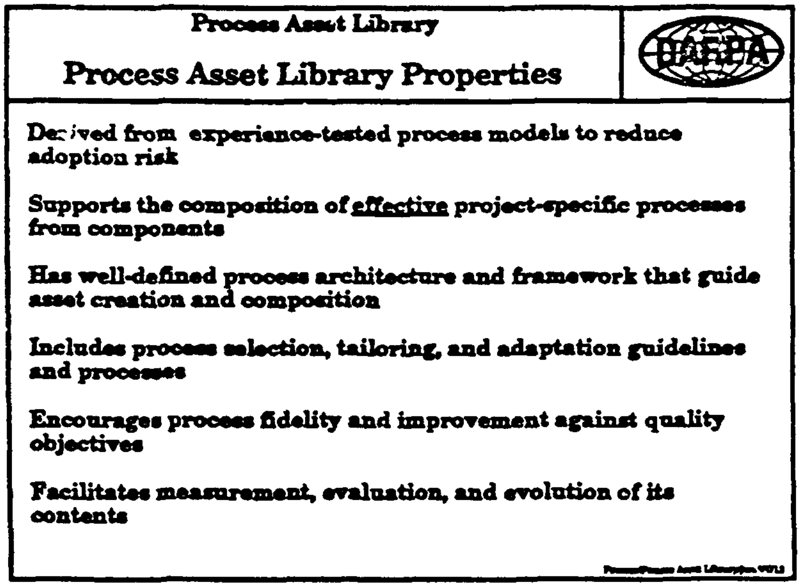
The PDAG workshop of October 1-3 produced a large quantity of information that will contribute to the development of the PAL.

The workshop focused on PAL long-term and short-term usage scenarios, architectural concepts, and components or asset types and their instances

Results were impressive and are beyond the scope of this presentation, but highlights include:

- 100+ page summary report covering output from each of six discussion groups
- a variety of usage scenarios for various roles and functions
- examples of architectural concepts from other disciplines and the application of these concepts to PAL or process architecture
- a "mind-map" containing properties, functions, and asset types for a process asset library

The summary report will contain all of the workshop products. To provide some insight into the contents, this section of the presentation will feature summaries and examples of the work products of the PDAG discussion groups .



Process Asset Library

Process Asset Library Properties

De:ived from experience-tested process models to reduce adoption risk

Supports the composition of effective project-specific processes from components

Has well-defined process architecture and framework that guide asset creation and composition

Includes process selection, tailoring, and adaptation guidelines and processes

Encourages process fidelity and improvement against quality objectives

Facilitates measurement, evaluation, and evolution of its contents

[illegible]

The following represent properties of the PAL derived from the workshop discussion which were captured on video tape.

The workshop participants agreed that the PAL should contain examples of experience-tested process in order to reduce adoption risk.

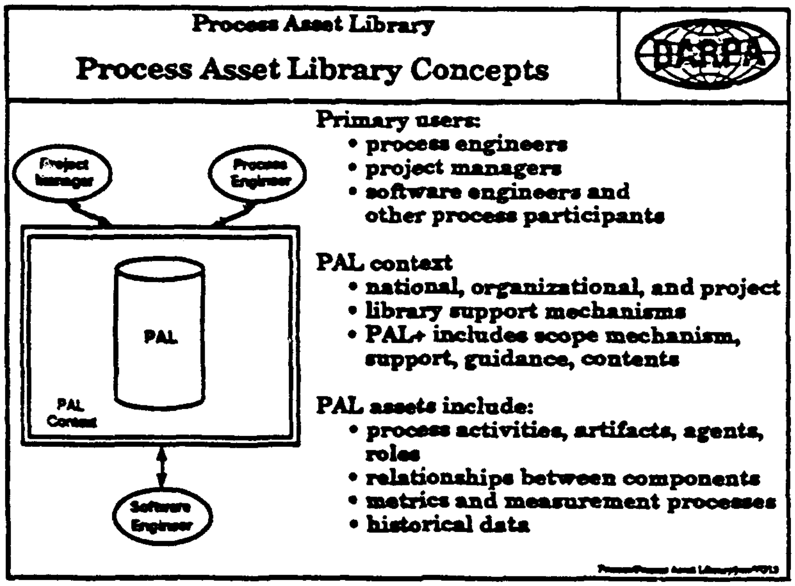
Also, the PAL needs to support the composition of effective processes form its component parts. In other words, it should make it easier to produce a usable, applicable process that promotes product quality objectives.

This property is facilitated by the process architecture and framework such that these guide the compositior. process to yield effective processes and effective assets.

The PAL needs to include its own processes for the selection, tailoring, and adaptation of assets. Guidelines or criteria that drive these processes should also be included.

The PAL must support the achievement of the key motivators such as process fidelity and process improvement

Finally, the PAL should facilitate the measurement, evaluation, and evolution or improvement of the assets it contains.


Project Manager
Process Engineer
PAL
PAL Context
Software Engineer

Process Asset Library

Process Asset Library Concepts

Primary users:
- process engineers
- project managers
- software engineers and other process participants

PAL context
- national, organizational, and project
- library support mechanisms
- PAL+ includes scope mechanism, support, guidance, contents

PAL assets include:
- process activities, artifacts, agents, roles
- relationships between components
- metrics and measurement processes
- historical data

Several interesting concepts emerged from the PDAG workshop. Some provided confirmation of existing PAL concepts, others provide fresh insight into the near and long term possibilities.

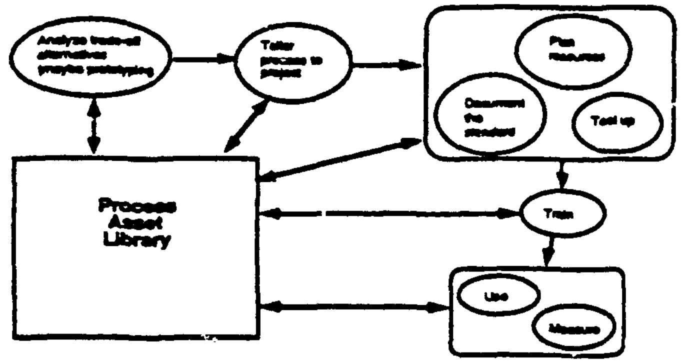
A commonly shared view among most participants was the primary users or roles that need to be supported. These include personnel from an organizations Software Engineering Process Group, referred to as process engineers. Project managers will also make use of a PAL as their role expands to include process management responsibilities. And it was felt that other users such as software engineers and other participants might need to have access to the PAL.

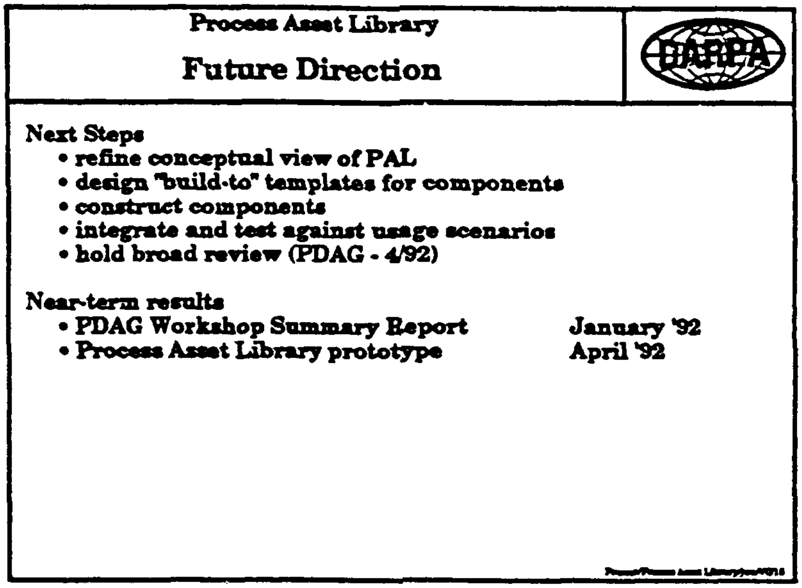
Several contexts were discussed. The notion of the PAL as a national, organizational, and project library of assets was explored. The underlying technology such as library mechanisms also received some attention. Finally to help distinguish some of these dimensions as well as others, the term PAL+ has been coined as an overarching term. Dimensions inside PAL+ include scope, mechanisms, support tools, guidance and training, and the contents of a library.

Examples of assets to be found in the library include activities, artifacts, agents, roles, relationships between components, metric and measurement processes, and historical data.

Process Asset Library

Example Usage Scenario
Process Definition and Installation!





Process Asset Library

Future Direction

DARPA

Next Steps

- refine conceptual view of PAL
- design "build-to" templates for components
- construct components
- integrate and test against usage scenarios
- hold broad review (PDAG - 4/92)

Near-term results

• PDAG Workshop Summary Report	January '92
• Process Asset Library prototype	April '92

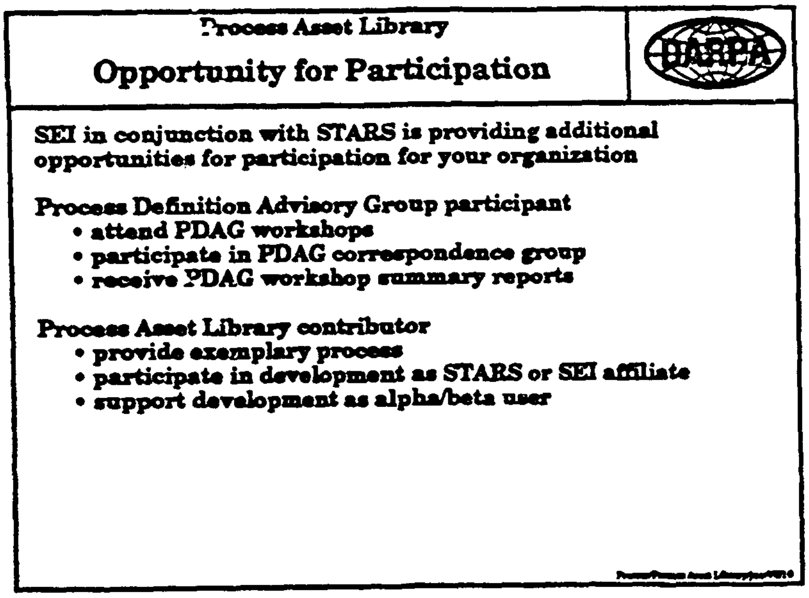
Near-term future direction includes the analysis of the October PDAG workshop to establish additional requirements, synthesize usage scenarios and architectural concepts to refine the conceptual and structural models of the PAL.

Subsequently, "build-to" templates for component types will be designed, and then populated with instances of process from the evolving archive of experience-tested processes.

The components will be integrated and tested against the usage scenarios and presented for review at the next PDAG workshop tentatively planned for April '92.

Near-term results include:

- PDAG Summary Report from the October PDAG Workshop
- Process Asset Library prototype for the April '92 PDAG Workshop



Process Asset Library

Opportunity for Participation

SEI in conjunction with STARS is providing additional opportunities for participation for your organization

Process Definition Advisory Group participant
- attend PDAG workshops
- participate in PDAG correspondence group
- receive PDAG workshop summary reports

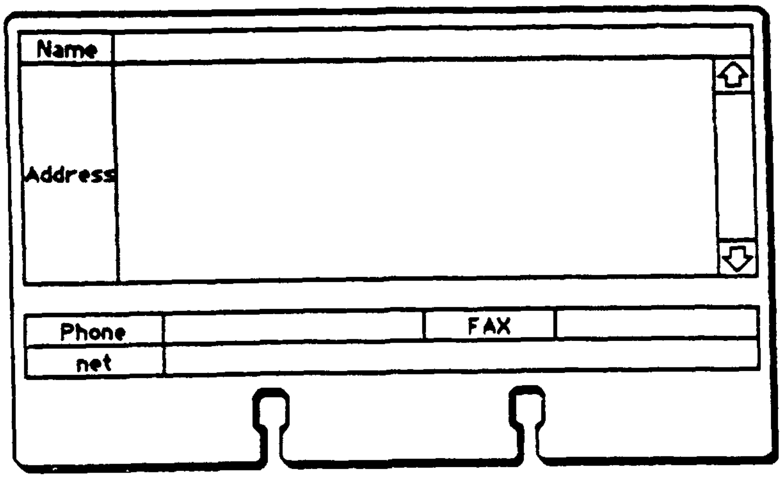
Process Asset Library contributor
- provide exemplary process
- participate in development as STARS or SEI affiliate
- support development as alpha/beta user

To further encourage broad participation in this activity, the SEI in conjunction with STARS is providing additional opportunities for participation for your organization as:
- Process Definition Advisory Group participant
- Process Asset Library contributor

Please complete the following form, indicating your desired participation and mail or FAX to:

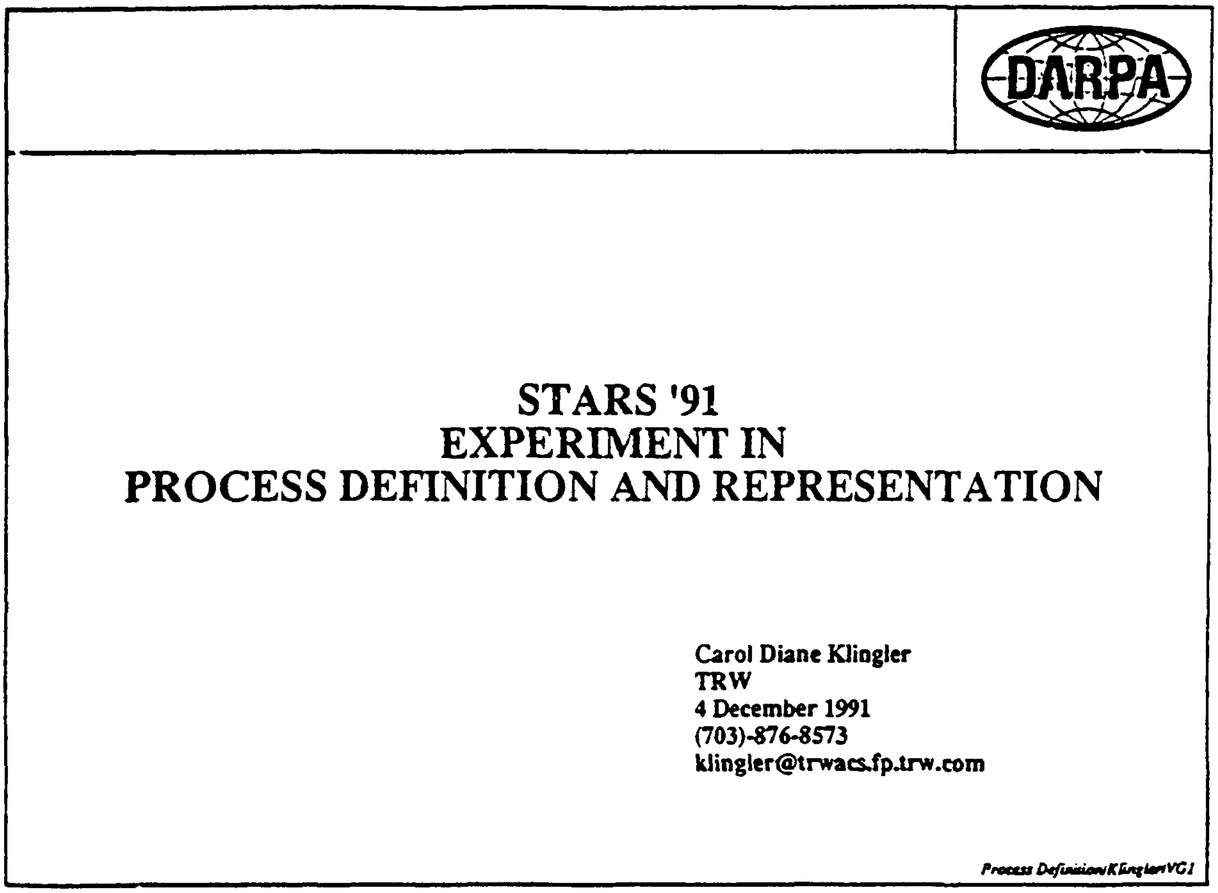
Jim Over
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
FAX (412) 268-5758

or send e-mail to: jwo@sei.cmu.edu



Name			
Address			
Phone		FAX	
net			

□ PDAG participant □ PAL contributor

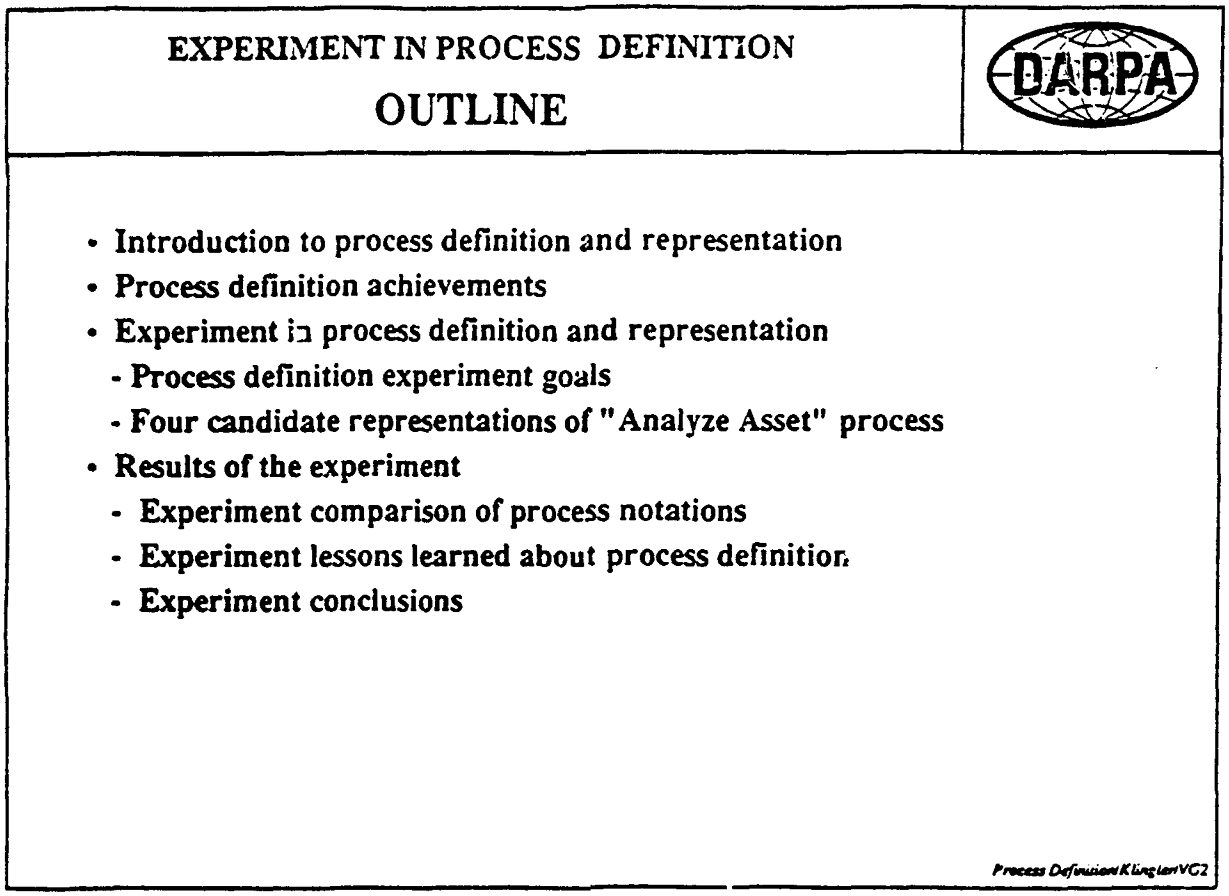


STARS '91 EXPERIMENT IN PROCESS DEFINITION AND REPRESENTATION

Carol Diane Klingler
TRW
4 December 1991
(703)-876-8573
klingler@trwacs.fp.trw.com

Process Definition/Klingler/VG1

This presentation will explain the reasons for defining a process and show some examples of a few of the many different notations that can be used for representing this process definition.



EXPERIMENT IN PROCESS DEFINITION

OUTLINE

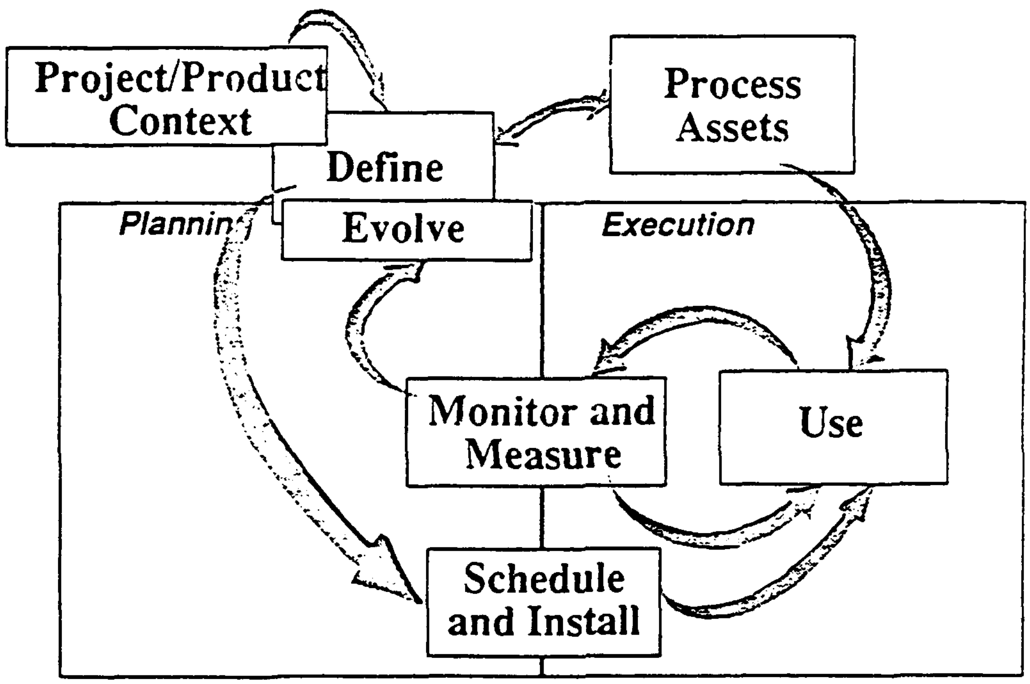
- Introduction to process definition and representation
- Process definition achievements
- Experiment in process definition and representation
  - Process definition experiment goals
  - Four candidate representations of "Analyze Asset" process
- Results of the experiment
  - Experiment comparison of process notations
  - Experiment lessons learned about process definition
  - Experiment conclusions

Process Definition/Klingler/VG2

In this presentation, we first introduce the topic of process definition and representation. We then mention the process definition achievements that have involved the STARS project. The remainder of the briefing examines one particular process experiment carried out by TRW for STARS. The goals of the experiment are shown and the example process that was examined is outlined. A small portion of the process definition is shown in four different candidate notations. At the end of the presentation, we present a comparison of the candidate notations, the lessons learned in the experiment about process definition, and the conclusions that we made from the experiment.

EXPERIMENT IN PROCESS DEFINITION

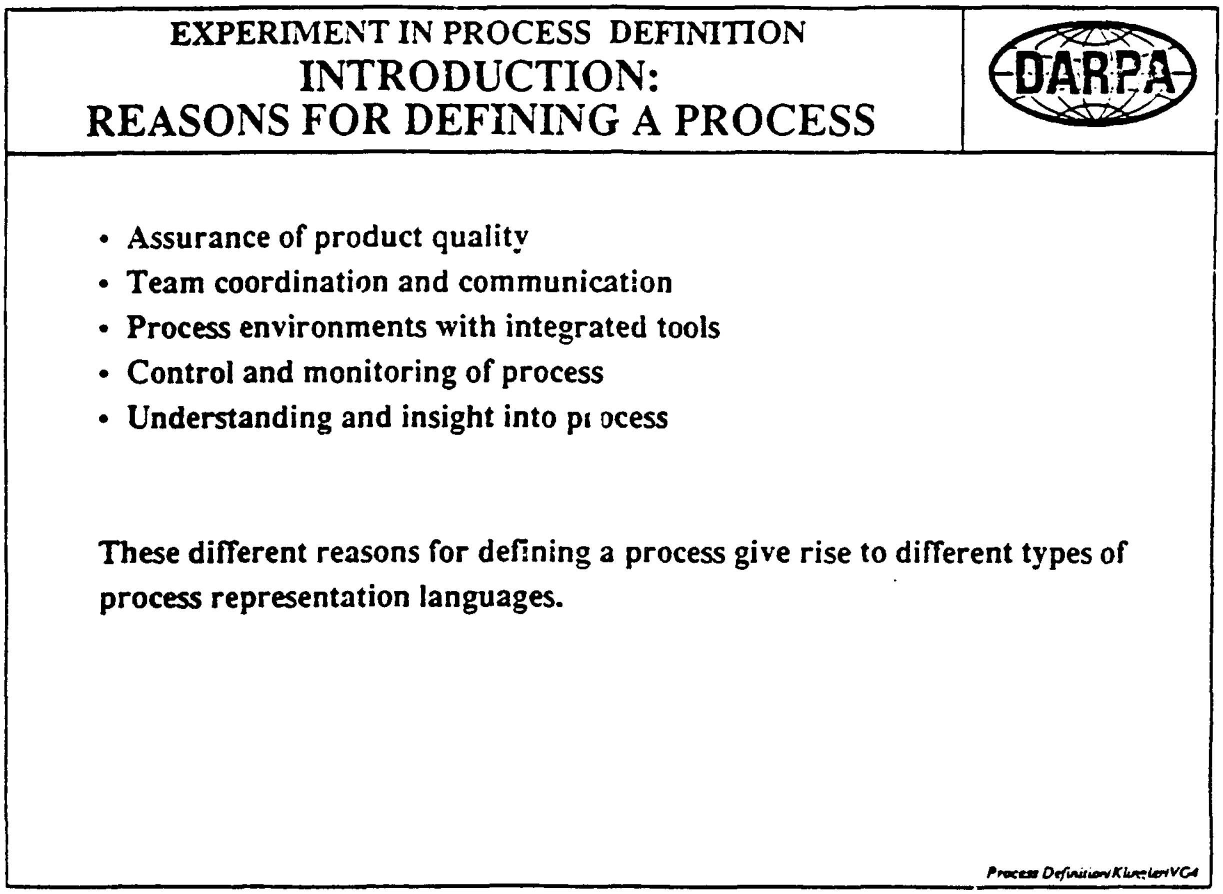
INTRODUCTION: PROCESS DRIVEN DEVELOPMENT



Adapted from Software Productivity Consortium work.

Process Definition/KLingler/VG3

To create quality software, we need to improve our software development process. Before a process can be improved, it must first be studied. To study a process, it must first be defined. This figure shows where process definition fits into the process driven development lifecycle. When a project begins, the project's software development process is defined, within the scope of the particular project or product. The process definition may include process assets found in a Process Asset Library (PAL), tailored to meet the project needs. As process definitions are developed, they may also be placed in the PAL for use by future projects. When the process definition is complete, the process is scheduled and installed. The process is then used (enacted) to execute the project. As project execution continues, the process is monitored and measured, and this data is fed back to the process evolution task in which the process definition is improved and refined.



EXPERIMENT IN PROCESS DEFINITION

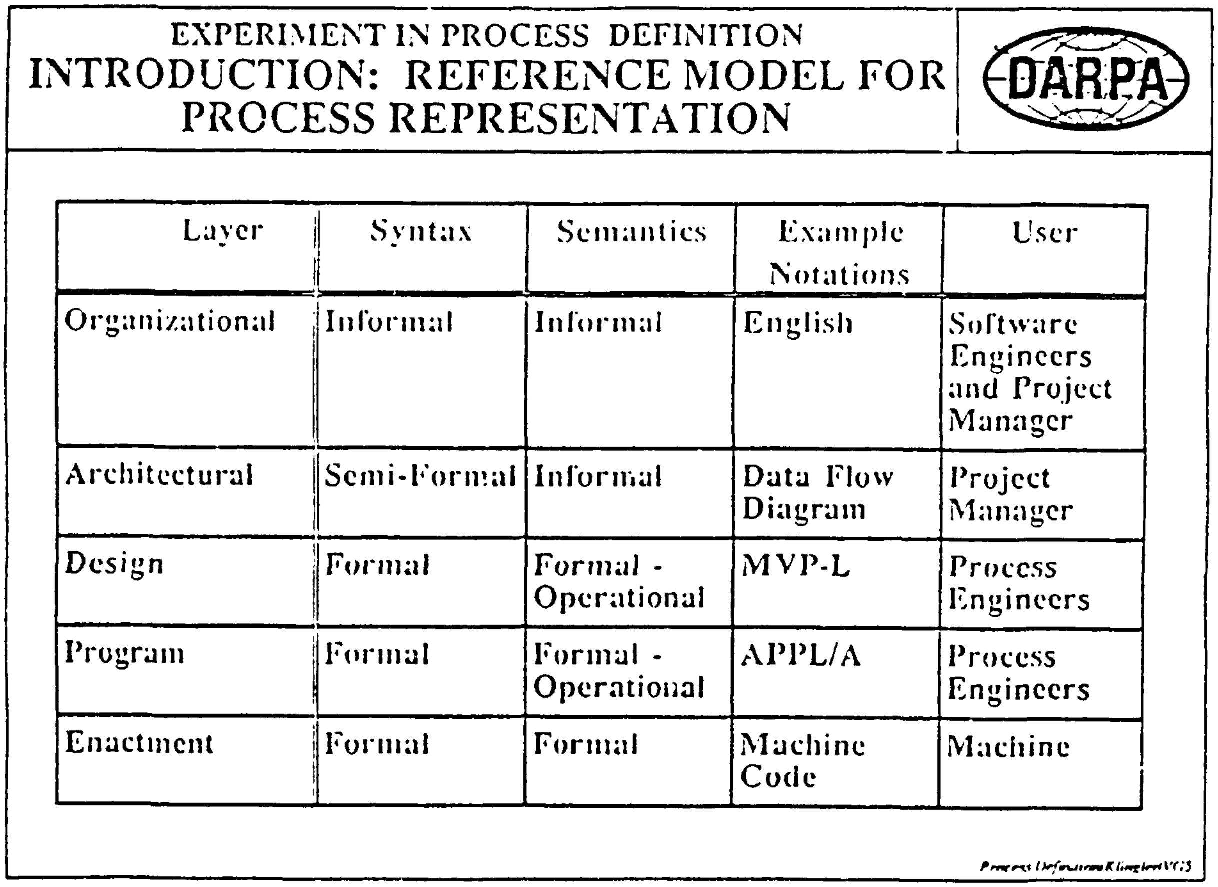
INTRODUCTION: REASONS FOR DEFINING A PROCESS

- Assurance of product quality
- Team coordination and communication
- Process environments with integrated tools
- Control and monitoring of process
- Understanding and insight into pı ocess

These different reasons for defining a process give rise to different types of process representation languages.

Process Definition/Kluwler/VG4

A process must first be defined in order to be studied and improved. This slide shows some of the reasons for defining a process. The main reason for defining a process is to study and improve the process to ensure the quality of the product that is created by the process. Another goal is to facilitate coordination and communication between team members carrying out the process, such as managers, software engineers, quality assurance, configuration managers, testers, etc. Another reason to define a process is to develop a machine-enactable process definition, which can be used to create a process environment with integrated tools to support the software engineers. A goal at the opposite extreme of the spectrum is the goal of controlling and monitoring the process to "lay down the rules" for how the process is carried out by the software engineers. The integrated support environment and control goals are two extremes that must be carefully balanced to allow software engineers the freedom to be creative while still allowing project management to control the overall process. Another reason for defining a process is to provide understanding and insight into the process, for the research aim of studying the process itself. These different reasons for defining a process give rise to many different notations that are used to represent a process definition. The different notations are not mutually exclusive. A combination of notations may be used, tailored to fit the project goals.



EXPERIMENT IN PROCESS DEFINITION

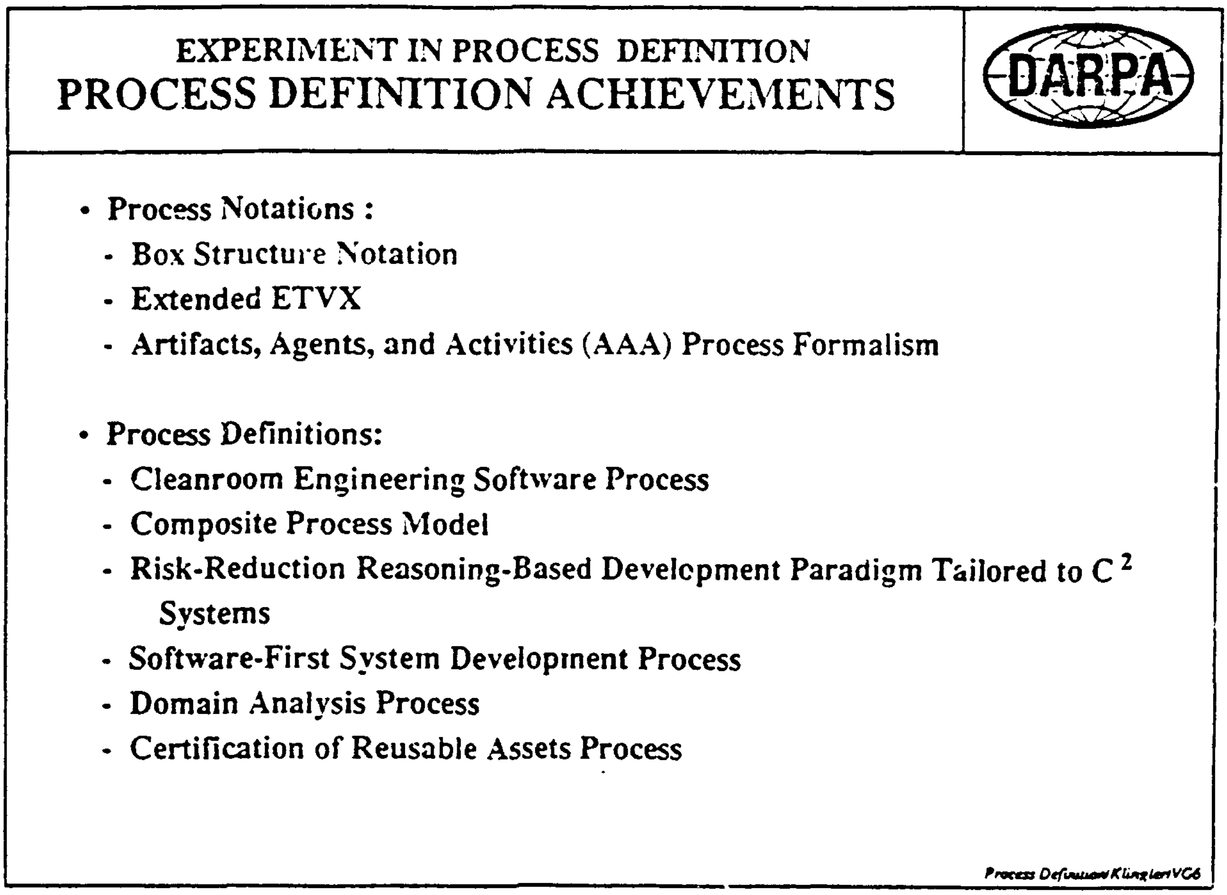
INTRODUCTION: REFERENCE MODEL FOR PROCESS REPRESENTATION

DARPA

Layer	Syntax	Semantics	Example Notations	User
Organizational	Informal	Informal	English	Software Engineers and Project Manager
Architectural	Semi-Formal	Informal	Data Flow Diagram	Project Manager
Design	Formal	Formal - Operational	MVP-L	Process Engineers
Program	Formal	Formal - Operational	APPL/A	Process Engineers
Enactment	Formal	Formal	Machine Code	Machine

ProcessDefinitionKlinglerVG3

A process is defined by represen'ing it in one or more notations. There is no one correct notation to use. Different notations may be used, at different levels of abstraction, depending on the reasons the process is being defined and the level of detail needed. This chart shows a reference model for process representation developed by STARS. This reference model serves as a means for talking about processes at various levels of detail, and as a focal point for discussing representations which are more appropriate for specific levels. For example, at the organizational layer, a process's requirements may be specifiea in English, with informal syntax and semantics. Once analyzed, these requirements may be translated into an architecture in a graphical or hybrid graphical/textual notation, such as Data Flow Diagrams. At the architectural layer, notations have a semi-formal defined syntax, which can be tailored to the needs of the particular process definition activity. Below the architectural layer, at the design layer, a textual representation language is used, such as MVP-L. This is the layer in which process engineers formally define the process in a notation with a formal syntax and formal semantics. Notations at this layer may be translated from the notation(s) used at the architectural layer. Below the design layer, at the program layer, the process is coded in a notation at a level of detail sufficient for enactment on a machine. Again the notation may be partially or fully translated from the notation(s) at the layers above. The lowest layer is the enactment layer, in which the process code is run on the machine. These process representation layers do not portray concrete, absolute boundaries for process definition activities. A specific process notation may be appropriate at more than one layer. Process representations in any layer may be enacted by humans. Lee Osterweil, in a paper entitled "Software Processes are Software Too", suggests that the development of process representations should follow a paradigm similar to development of a conventional software product, and will include activities such as requirements analysis, high-level design, low-level design, and coding. These activities correspond to the organizational, architectural, design and program layers of the reference model. Osterweil continues his analogy by suggesting that the activity of building explicit, formal software process representations be referred to as process programming, and that the resulting representations by called process programs. This is the reason why many notations suitable for use at the program layer are referred to as process programming languages. Process representations in any layer are usually defined by process engineers. Once defined, process representations in the architectural layer may be used by project managers to gain an understanding of the overall process architecture. Process representations in the organizational layer, and possibly the architectural layer, are usually used by the individuals carrying out the process to guide them in their tasks. In the design and program layers, the process representations are used primarily by the process engineers who have developed them. In the enactment layer, the process is represented by machine code, which is used by the machine to enact the process.



EXPERIMENT IN PROCESS DEFINITION

PROCESS DEFINITION ACHIEVEMENTS

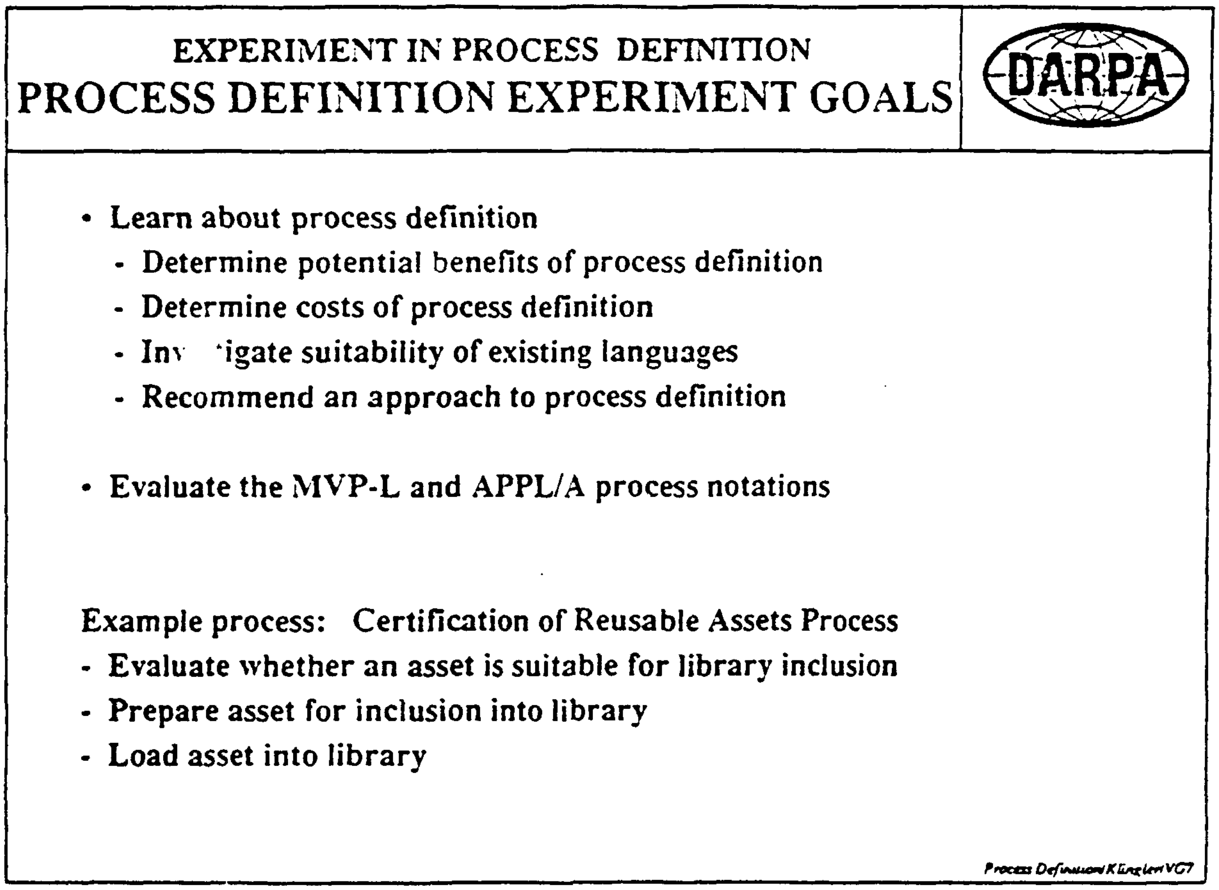
DARPA

- Process Notations :
  - Box Structure Notation
  - Extended ETVX
  - Artifacts, Agents, and Activities (AAA) Process Formalism

- Process Definitions:
  - Cleanroom Engineering Software Process
  - Composite Process Model
  - Risk-Reduction Reasoning-Based Development Paradigm Tailored to C^2 Systems
  - Software-First System Development Process
  - Domain Analysis Process
  - Certification of Reusable Assets Process

Process Definition/Klingler/VG6

STARS has been involved in many process definition activities. We have participated in efforts involving the use of a number of notations used to define processes, for example the box structure notation, used in the Cleanroom Engineering Software Process; Extended ETVX, used in the Software Process Management System (SPMS); and the Artifacts, Agents, and Activities (AAA) Process Formalism, used in the Policy Representation using Control Point Process Enactment Mechanism. Prototype demonstrations of these three systems can be seen at STARS '91, and fact sheets and other documentation is also available. STARS has also been involved in defining many processes, including the processes listed at the bottom of this chart. The remainder of this presentation discusses a STARS experiment, performed at TRW under subcontract to UNISYS, in which a Certification of Reusable Assets process definition is represented in a number of different notations.



EXPERIMENT IN PROCESS DEFINITION

PROCESS DEFINITION EXPERIMENT GOALS

DARPA

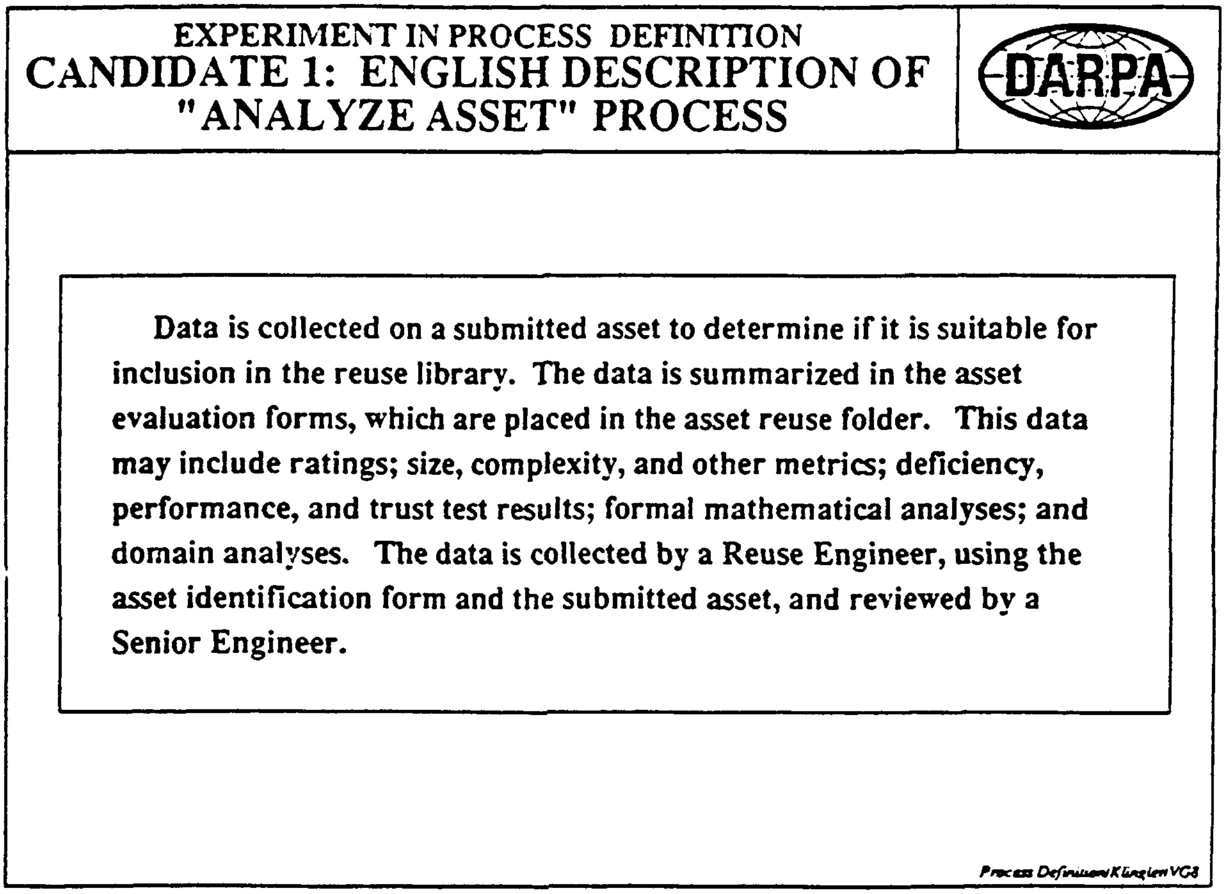
- Learn about process definition
  - Determine potential benefits of process definition
  - Determine costs of process definition
  - Investigate suitability of existing languages
  - Recommend an approach to process definition
- Evaluate the MVP-L and APPL/A process notations

Example process: Certification of Reusable Assets Process

- Evaluate whether an asset is suitable for library inclusion
- Prepare asset for inclusion into library
- Load asset into library

Process Definition/Klingler/VG7

Our experiment involved a case study in process definition and representation, by a small team of individuals. We wished to learn about the benefits and costs of defining a process. We also wished to investigate the suitability of some existing notations for use by process engineers without prior familiarity with the notations. This is an important aim because much of the work on the use of process notations is being performed by the same organizations that created the notations. We wanted to present an unbiased opinion on the appropriate uses of some notations currently available. We also planned to write down our lessons learned and recommend an approach to be used by other organizations to define processes. We examined 18 notations used to represent a software change process, in an exercise at the 6th International Software Process Workshop (ISPW). After careful considerations, the notations were narrowed down to two, MVP-L and APPL/A. These notations were used, along with English descriptions, Data Flow Diagrams, and Hierarchy Charts, to represent our process definition. The Certification of Reusable Assets process defined in the experiment consists of evaluating whether an asset is suitable for inclusion in a reuse library, using domain analyses and other evaluations, preparing accepted assets for inclusion in the library, and loading them into the library. The entire process is too large to examine in this presentation and is highlighted in our "Process Programming Languages Experimentation Report". We have also produced a paper on our use of the MVP-L process representation language. Both of these reports are available to anyone interested. In the remainder of the presentation we focus on one process element in the Certification of Reusable Assets process, namely the process of analyzing an asset to produce the evaluations used to determine if the asset is suitable for inclusion in the particular reuse library. We show the representation of this process element definition in four different candidate notations, English, Data Flow Diagram, MVP-L code, and APPL/A code. We also discuss the advantages and disadvantages of each of these notations, as determined from our experiment. These examples will illustrate a few of the many varied methods in which a process may be defined.



EXPERIMENT IN PROCESS DEFINITION

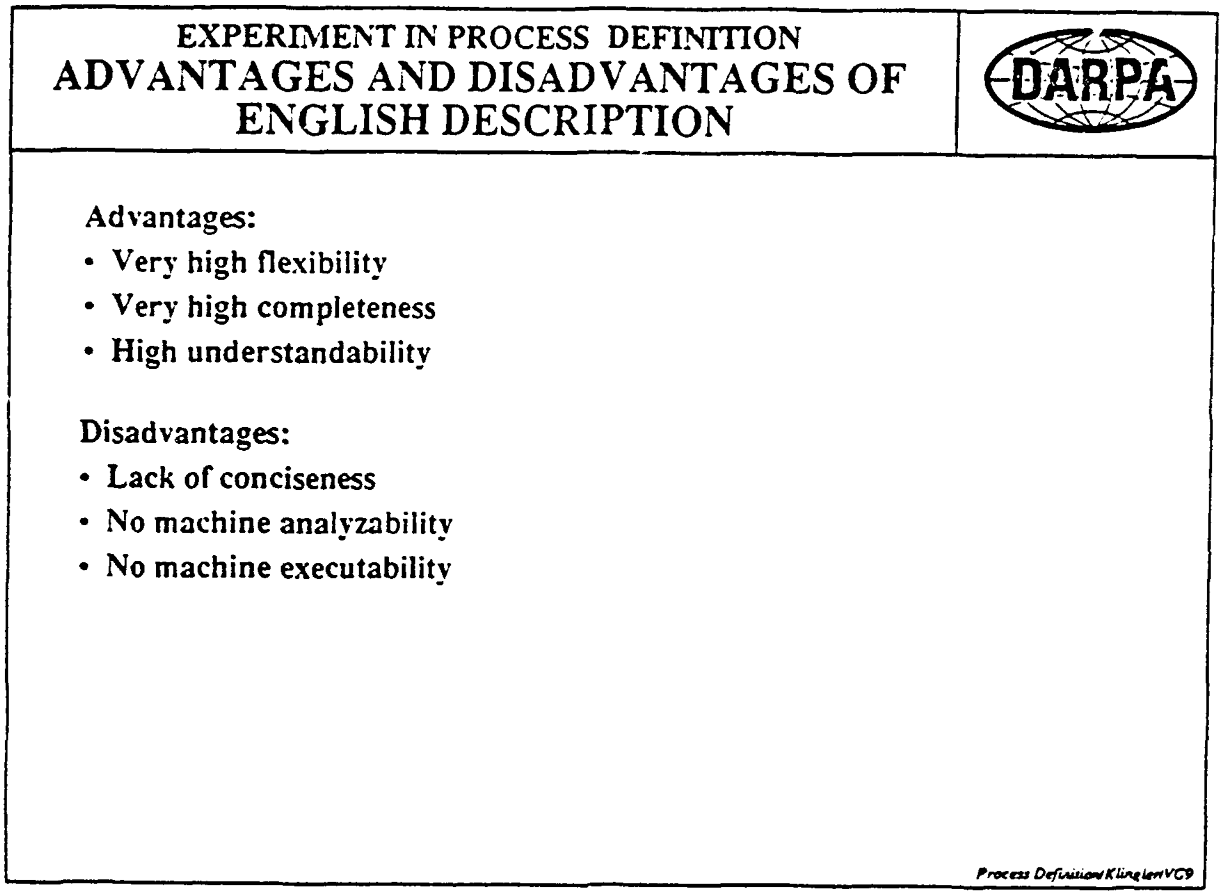
CANDIDATE 1: ENGLISH DESCRIPTION OF "ANALYZE ASSET" PROCESS

DARPA

Data is collected on a submitted asset to determine if it is suitable for inclusion in the reuse library. The data is summarized in the asset evaluation forms, which are placed in the asset reuse folder. This data may include ratings; size, complexity, and other metrics; deficiency, performance, and trust test results; formal mathematical analyses; and domain analyses. The data is collected by a Reuse Engineer, using the asset identification form and the submitted asset, and reviewed by a Senior Engineer.

Process Definition/Klingler/VG8

This slide contains an English description of the "Analyze Asset" process. Anyone involved in software development has seen English descriptions of elements of the software development process, for example, programming standards, test procedure:, and configuration management manuals. This description shows that the "Analyze Asset" process consists of a Reuse Engineer collecting data on an asset, using the asset itself and an identification form, and placing this data in the evaluation forms. These forms are then reviewed by a Senior Engineer.



EXPERIMENT IN PROCESS DEFINITION

ADVANTAGES AND DISADVANTAGES OF ENGLISH DESCRIPTION

DARPA

Advantages:

- Very high flexibility
- Very high completeness
- High understandability

Disadvantages:

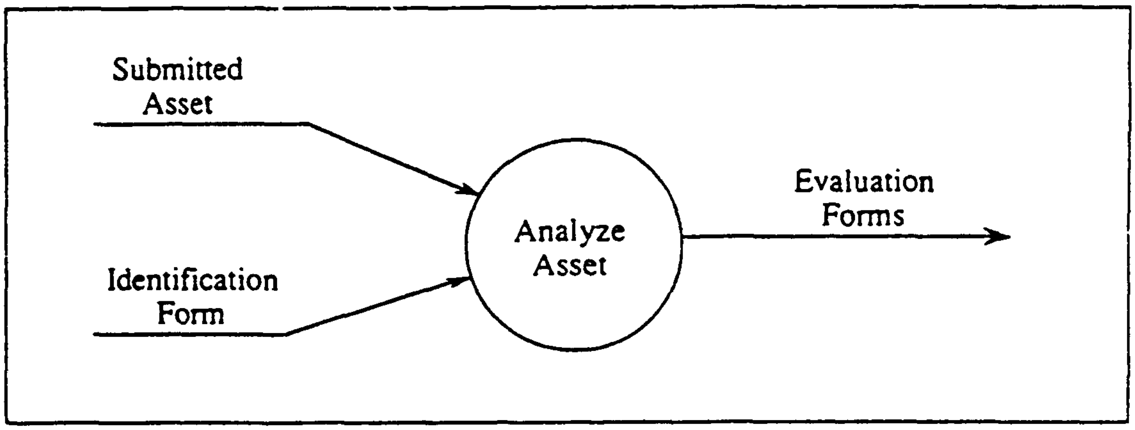
- Lack of conciseness
- No machine analyzability
- No machine executability

Process Definition/Klingler/VC9

This chart shows the advantages and disadvantages of using English descriptions to represent a process definition, as determined from our experiment. English descriptions of processes are very flexible; a process can be described in many different ways in English, depending on the process definition activity's goals. The process can be completely described in English, due to the wealth of words available. English words are also very easy for humans to understand, since no special knowledge is needed. If a lot of detail is not required, a process can be easily described in a small amount of space in an English description.

However, the English description lacks conciseness. The ambiguity of English and the lack of formal syntax and semantics leads to a process definition that can be interpreted differently by different people. Also the process definition cannot be analyzed and executed on a machine, due to the lack of formal syntax and semantics. The lack of analyzability makes it difficult to determine whether an important portion of the process is left out of the English description.

EXPERIMENT IN PROCESS DEFINITION
CANDIDATE 2: DATA FLOW DIAGRAM FOR "ANALYZE ASSET" PROCESS



Process Definition/KEngler/VG10

The Data Flow Diagram in this chart shows one possible graphical representation of the "Analyze Asset" process. This diagram illustrates that the submitted asset and identification form are input to the "Analyze Asset" process and the evaluation forms are produced.

EXPERIMENT IN PROCESS DEFINITION

ADVANTAGES AND DISADVANTAGES OF DATA FLOW DIAGRAM

Advantages:

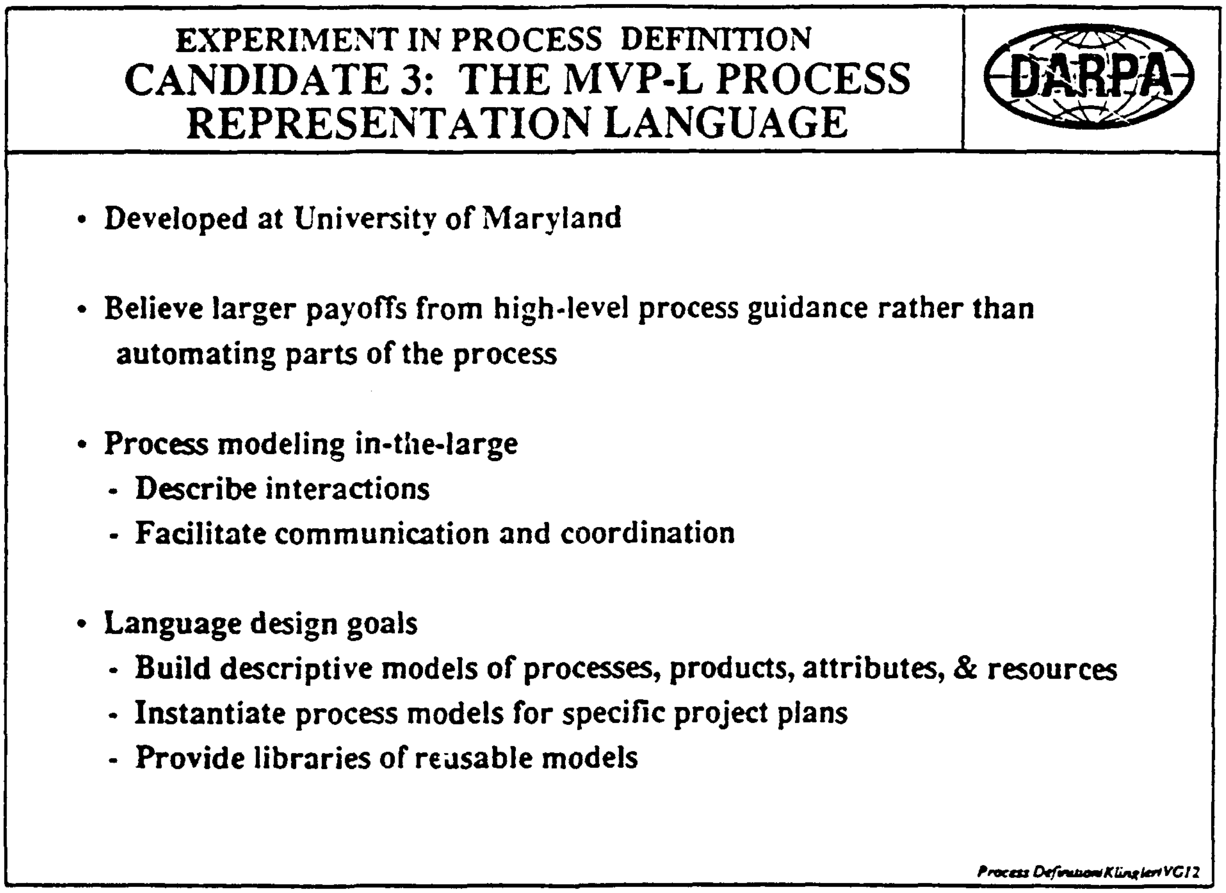
- Very high understandability
- High conciseness

Disadvantages:

- Low completeness
- Low flexibility
- Low machine analyzability
- No machine executability

Process Definition/Klingler/VG11

This chart shows the advantages and disadvantages of using Data Flow Diagrams to represent a process definition, as determined from our experiment. The Data Flow Diagram is very simple and easy to understand, and consequently a good notation to use to communicate a process definition to project managers and participants. It is a concise notation; the entire "Analyze Asset" process can be represented in one small diagram. However, there are important aspects of the process that are missing; for example the criteria used to determine when the process may start and stop and the names of the participants in the process. There is little flexibility in this notation. The notation allows some machine analysis and no machine execution of the process. Some of the more advanced graphical notations that are used to define processes have some of the characteristics that are missing from Data Flow Diagrams but we have not yet examined these graphical notations. We will be examining the SADT graphical notation in a follow-on experiment.



EXPERIMENT IN PROCESS DEFINITION

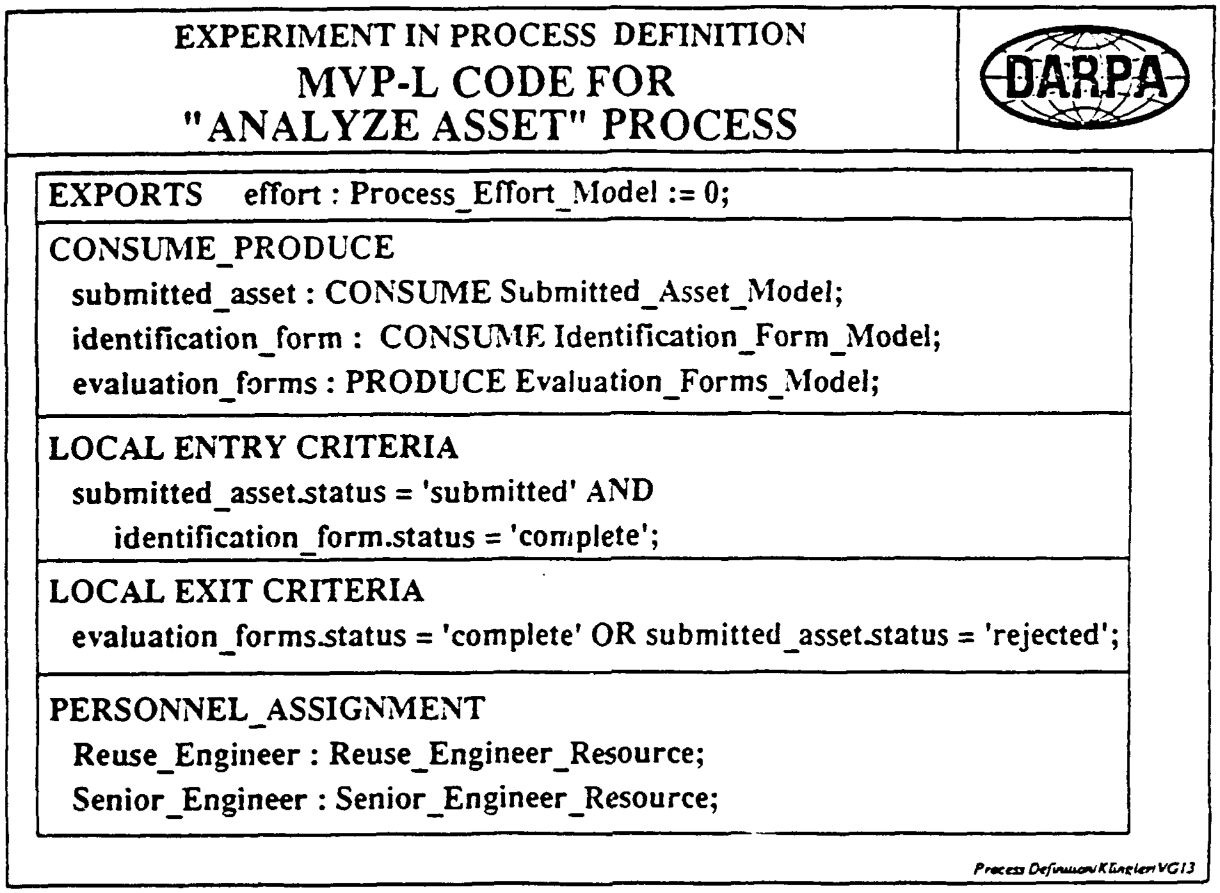
CANDIDATE 3: THE MVP-L PROCESS REPRESENTATION LANGUAGE

DARPA

- Developed at University of Maryland
- Believe larger payoffs from high-level process guidance rather than automating parts of the process
- Process modeling in-the-large
  - Describe interactions
  - Facilitate communication and coordination
- Language design goals
  - Build descriptive models of processes, products, attributes, & resources
  - Instantiate process models for specific project plans
  - Provide libraries of reusable models

Process Definition/Klingler/VG12

The Multi-View Process Modeling Project (MVP), at the University of Maryland, developed a rule-based language called MVP-L. This language is one of a number of textual process representation languages that can be used to represent a process at the design layer of representation. The MVP project believes that larger payoffs can be gained from high-level process guidance rather than by automating small process elements on a computer. Therefore, MVP-L was designed for modeling the entire process "in-the-large". The language concentrates on describing the interactions between process activities to facilitate communication and coordination between the team members who carry out the process. MVP-L allows refinement and abstraction of processes, so that the entire process can be described at the highest level and then broken down into its components. The language contains process models which describe activities carried out; product models which describe the artifacts used and created; attribute models which contain attributes of processes and products, for example, process status; and human and tool resources, which are the agents that carry out the process. Project plans assemble process models and instantiate them, adding project-specific information; for example, the total amount of time allocated to the effort. Each MVP-L process, product, attribute and resource model is a separate units to facilitate the creation of a library of reusable models, which can then be tailored for a specific project. Formal syntactic and semantic definitions of the language have been developed. At the present time, the language does not include tools for graphical viewing, analysis, or machine execution of MVP-L code, but these are important parts of the research effort that will be addressed more fully in the future.



EXPERIMENT IN PROCESS DEFINITION

MVP-L CODE FOR "ANALYZE ASSET" PROCESS

```
EXPORTS    effort : Process_Effort_Model := 0;

CONSUME_PRODUCE
  submitted_asset : CONSUME Submitted_Asset_Model;
  identification_form :  CONSUME Identification_Form_Model;
  evaluation_forms : PRODUCE Evaluation_Forms_Model;

LOCAL ENTRY CRITERIA
  submitted_asset.status = 'submitted' AND
     identification_form.status = 'complete';

LOCAL EXIT CRITERIA
  evaluation_forms.status = 'complete' OR submitted_asset.status = 'rejected';

PERSONNEL_ASSIGNMENT
  Reuse_Engineer : Reuse_Engineer_Resource;
  Senior_Engineer : Senior_Engineer_Resource;
```

Process Definition/KEngler/VG13

This slide presents a portion of the code that is needed to represent the "Analyze Asset" Process in MVP-L. The "EXPORTS" section contains the "effort", which is an attribute describing the amount of time spent so far in the "Analyze Asset" process. The "CONSUME_PRODUCE" section describes the products (documents or other artifacts) used by the process, in this case the "submitted_asset" and the "identification_form", and the products created by the process, in this case the "evaluation_forms." The "LOCAL_ENTRY_CRITERIA" describe the conditions necessary for the process to start execution. When these conditions become true, a human decides when the process actually begins. In this case, there must be an asset that was submitted and a completed identification form in order for the "Analyze Asset" process to be ready for execution. The "LOCAL_EXIT_CRITERIA" describe when the process terminates. In the "Analyze Asset" process, the process ends when the evaluation forms have been completed or the submitted asset is rejected. The submitted asset may be rejected at any time by the "Senior_Engineer" if it is felt that the effort required to evaluate the asset is not worth the time that would be spent. The "PERSONNEL_ASSIGNMENT" section indicates that "Analyze Asset" is carried out by a "Reuse_Engineer" and a "Senior_Engineer".

EXPERIMENT IN PROCESS DEFINITION

ADVANTAGES AND DISADVANTAGES OF MVP-L CODE

DARPA

Advantages:

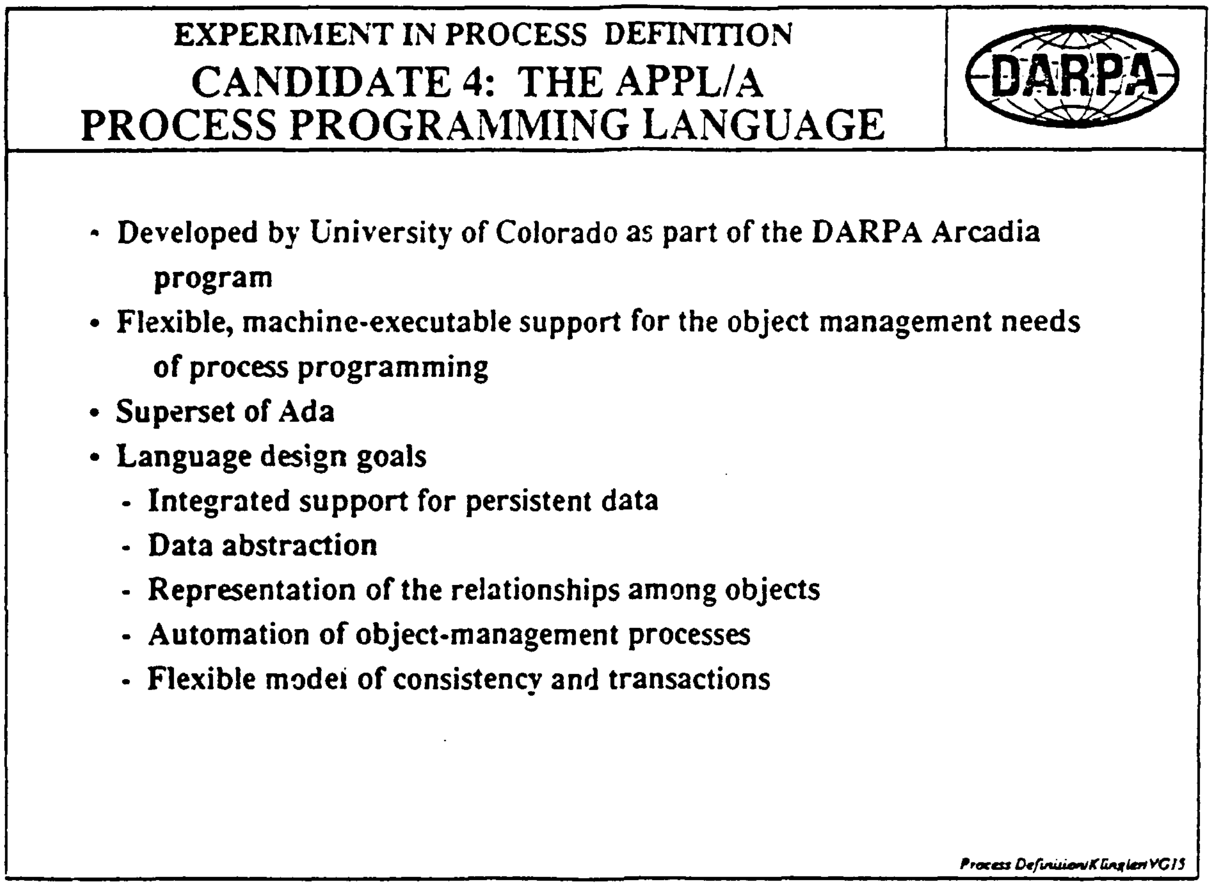
- Very high conciseness
- High completeness
- Medium understandability
- Medium flexibility

Disadvantages:

- No machine analyzability
- No machine executability

Process Definition/Klingler/VG14

This chart shows the advantages and disadvantages of using MVP-L code to represent a process definition, as determined from our experiment. MVP-L code is a very concise notation for describing in a modular manner the parts of a process, the products used and created, and the interactions between processes. These modules can be easily tailored and reused on different projects. We did not find the language to be wordy, or contain unnecessary features. The language contains all of the features needed to specify processes at the design layer, although in a very few instances we thought of alternate structures that may have been helpful. The University of Maryland is evaluating our recommendations and making any necessary changing to the MVP-L language. In our opinion, for a textual language, MVP-L is easy to understand. The language uses natural English words that make the models easy to read. It contains some flexibility, especially in the methods provided for describing how attributes change values. However, there is no static or dynamic machine analysis or machine execution currently supported by the language.



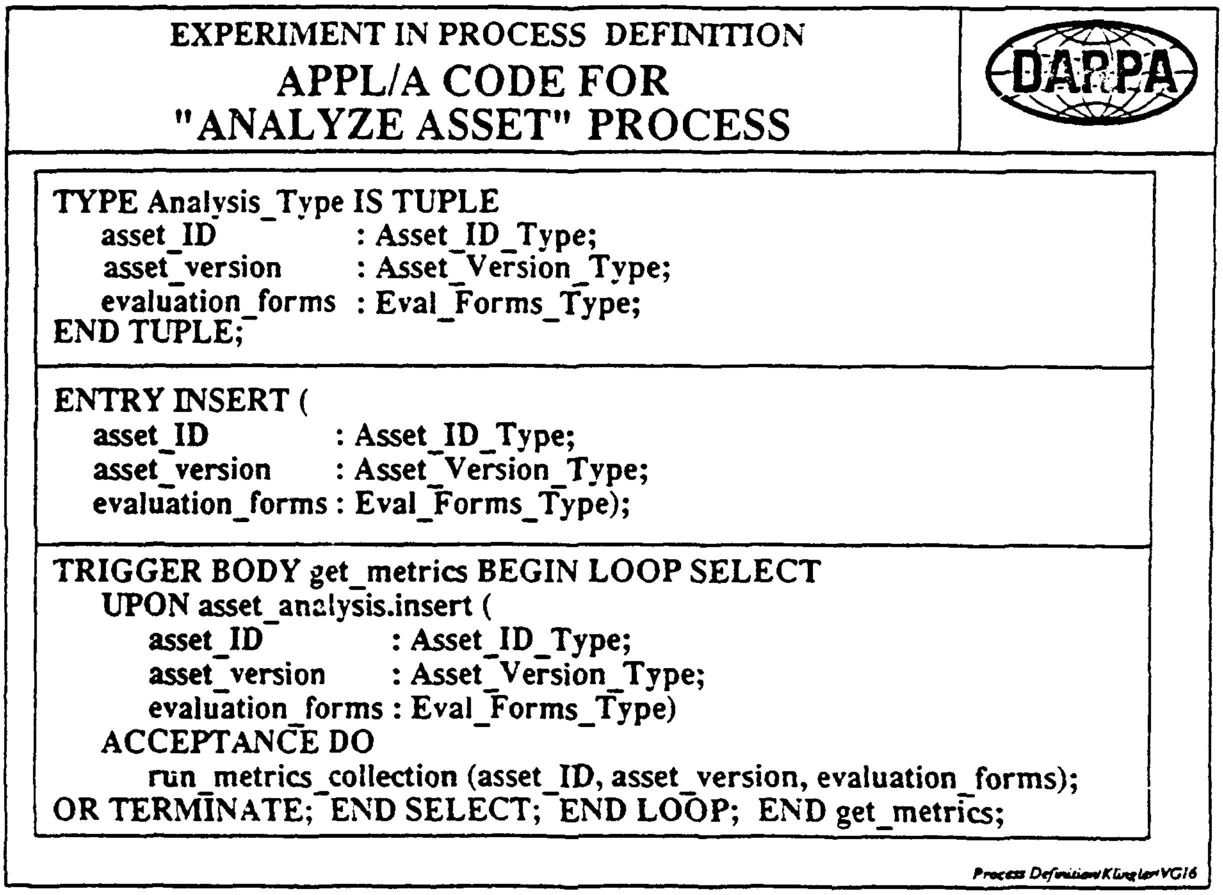
EXPERIMENT IN PROCESS DEFINITION

CANDIDATE 4: THE APPL/A PROCESS PROGRAMMING LANGUAGE

- Developed by University of Colorado as part of the DARPA Arcadia program
- Flexible, machine-executable support for the object management needs of process programming
- Superset of Ada
- Language design goals
  - Integrated support for persistent data
  - Data abstraction
  - Representation of the relationships among objects
  - Automation of object-management processes
  - Flexible model of consistency and transactions

Process Definition/KLangler/VG15

APPL/A, the Ada Process Programming Language based on Aspen, is a language useful for process representations at the program or coding layer of abstraction It was developed by the University of Colorado as part of the DARPA Arcadia project to provide flexible, machine-executable support for managing the products created and used by processes. The language contains the large variety of constructs needed to represent a process at the high level of detail necessary to allow the enactment of the process on a machine. APPL/A is a superset of Ada, which contains all of the Ada constructs plus new constructs for product management. It provides integrated support for the storage and retrieval of persistent data. Persistent data is data that needs to be retained in storage over long periods of time, even when the process has finished executing. Data abstraction is provided through Ada constructs. The products and relationships among products can be stored as data, through the use of a construct called a "relation". An example of a relationship that could be stored using a "relation" is the compilation relationship which creates object code from source code. There are also constructs available to run automated processes automatically; for example, when source code is stored, the code could be automatically compiled to create and store the object code. Other features ensure consistency and allow specialized transactions. Continuing our previous example, the consistency feature could be used to ensure that there is object code in the data store for each instance of source code. An example of the specialized transactions is the "atomic" statement, which allows a process to exclude other processes from accessing the data store when the process will be changing the data.



EXPERIMENT IN PROCESS DEFINITION

APPL/A CODE FOR "ANALYZE ASSET" PROCESS

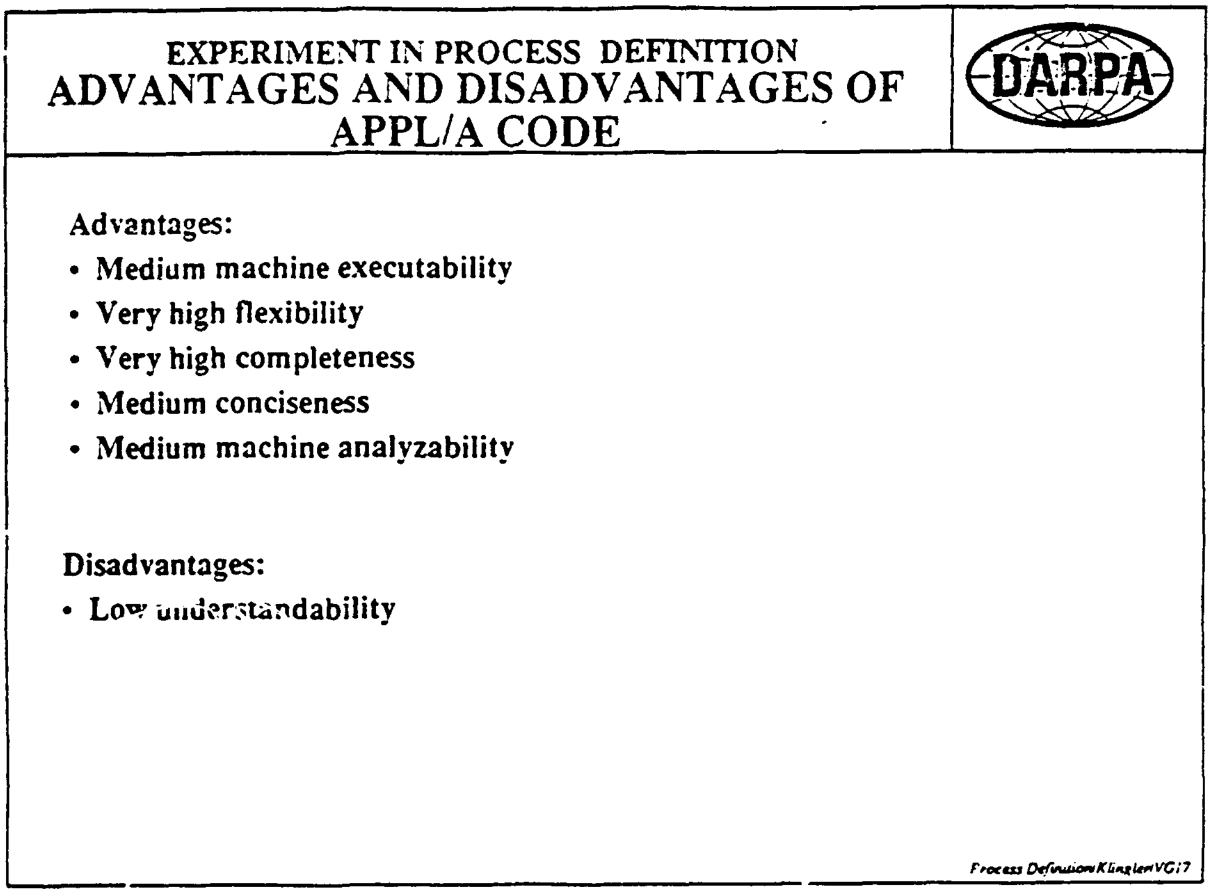
```
TYPE Analysis_Type IS TUPLE
    asset_ID          : Asset_ID_Type;
    asset_version     : Asset_Version_Type;
    evaluation_forms  : Eval_Forms_Type;
END TUPLE;
```

```
ENTRY INSERT (
   asset_ID          : Asset_ID_Type;
   asset_version     : Asset_Version_Type;
   evaluation_forms : Eval_Forms_Type);
```

```
TRIGGER BODY get_metrics BEGIN LOOP SELECT
    UPON asset_analysis.insert (
        asset_ID          : Asset_ID_Type;
        asset_version     : Asset_Version_Type;
        evaluation_forms : Eval_Forms_Type)
    ACCEPTANCE DO
        run_metrics_collection (asset_ID, asset_version, evaluation_forms);
OR TERMINATE; END SELECT; END LOOP;  END get_metrics;
```

Process Definition/Klingler/VG16

This example shows a small portion of the APPL/A code that can be used to represent the "Analyze Asset" process definition. First, there is an "Analysis_Type" "TUPLE", which shows names and types of the attributes used to store the "evaluation_forms". In the example, the "asset_ID" number, "asset_version" number, and the "evaluation_forms" themselves are stored. The "INSERT" "ENTRY" specifies the procedure call used to store the "evaluation_forms" with the other attribute values. The attributes specified in an "ENTRY" may not be the entire set of attributes specified in the "TUPLE"; for example, the object code attribute may not need to be supplied if it is compiled from the source code. The "TRIGGER" construct specifies that "run_metrics_collection" is automatically executed when the "evaluation_forms" are inserted into storage. The "run_metrics_collection" procedure collects metric data on the asset and adds it to the "evaluation_forms."



EXPERIMENT IN PROCESS DEFINITION

ADVANTAGES AND DISADVANTAGES OF APPL/A CODE

Advantages:

- Medium machine executability
- Very high flexibility
- Very high completeness
- Medium conciseness
- Medium machine analyzability

Disadvantages:

- Low understandability

Process Definition/KlinglerVG17

This chart shows the advantages and disadvantages of using APPL/A code to represent a process definition, as determined from our experiment. Most of the constructs supplied in the APPL/A language are currently executable by a machine through translation of the code to Ada. Machine executability is one characteristic of process notations that is not supported by many other notations, due to the complexity involved. There is also a high amount of flexibility in the APPL/A language, due to the very rich set of constructs available from the Ada language. Also, the product management needs of process programming were carefully analyzed when the language was created, to ensure that the language contained the complete set of constructs needed. APPL/A is not a highly concise language, due to the large number of features provided for flexibility and machine executability. Some machine analyzability is supported through translation to Ada and analysis of the Ada code. The biggest disadvantage of APPL/A code is the low level of granularity needed to specify a process at the machine enactment level, which makes the language less understandable than many process notations at higher layers.

EXPERIMENT IN PROCESS DEFINITION

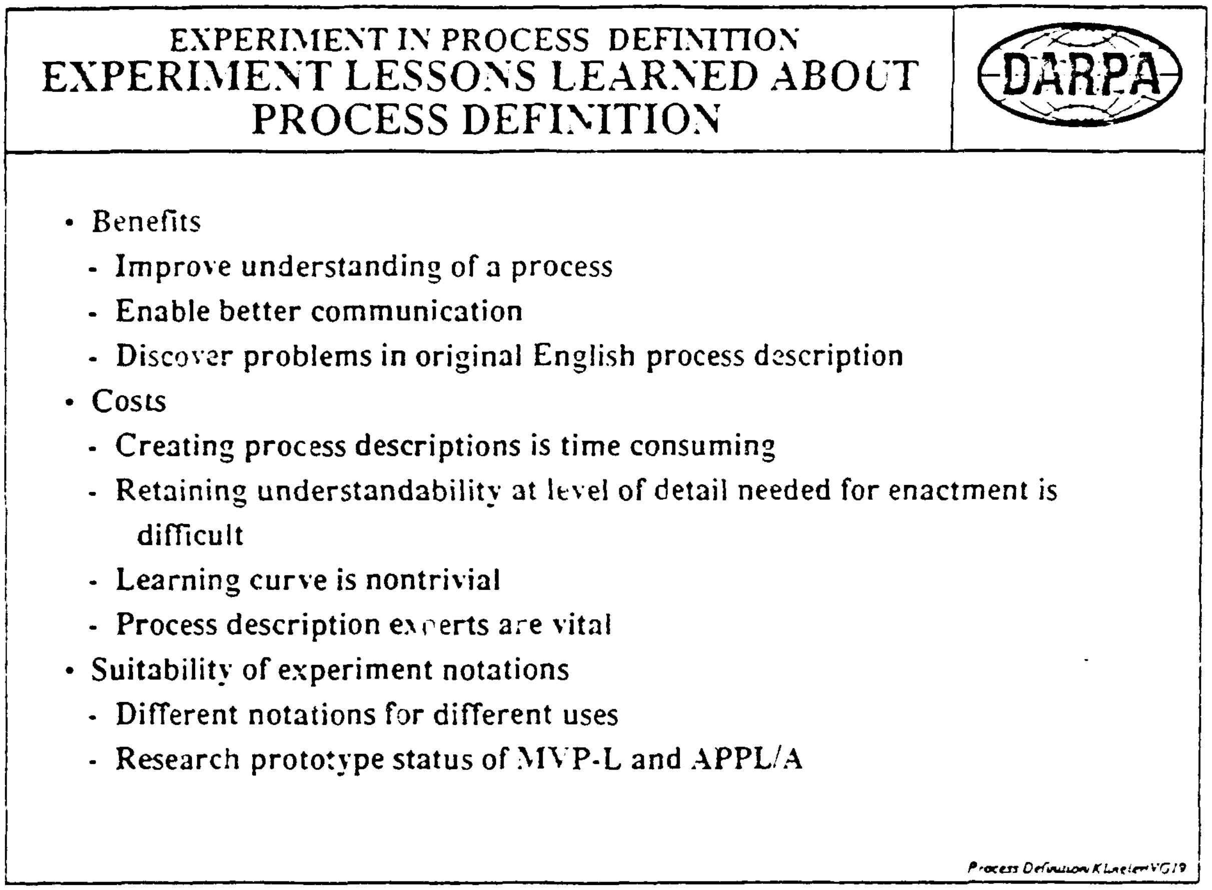
EXPERIMENT COMPARISON OF PROCESS NOTATIONS

DARPA

	Language Support			
Characteristic	English	DFD	MVP-L	APPL/A
Completeness	Very High	Low	High	Very High
Conciseness	Low	High	Very High	Medium
Understandability	High	Very High	Medium	Low
Flexibility	Very High	Low	Medium	Very High
Analyzability	None	Low	None	Medium
Executability	None	None	None	Medium

Process Definition/Klingler/VG18

This chart summarizes the comparison of process notation characteristics described in the previous slides. English is a complete, flexible notation, but contains no support for machine analyzability or executability. Data Flow Diagrams (DFDs) are understandable, but do not contain some of the constructs needed to completely describe a process. MVP-L is complete and concise, but does not support analyzability or executability. APPL/A supports some analyzability and executability, but is not as concise or understandable as the other notations.



EXPERIMENT IN PROCESS DEFINITION

EXPERIMENT LESSONS LEARNED ABOUT PROCESS DEFINITION

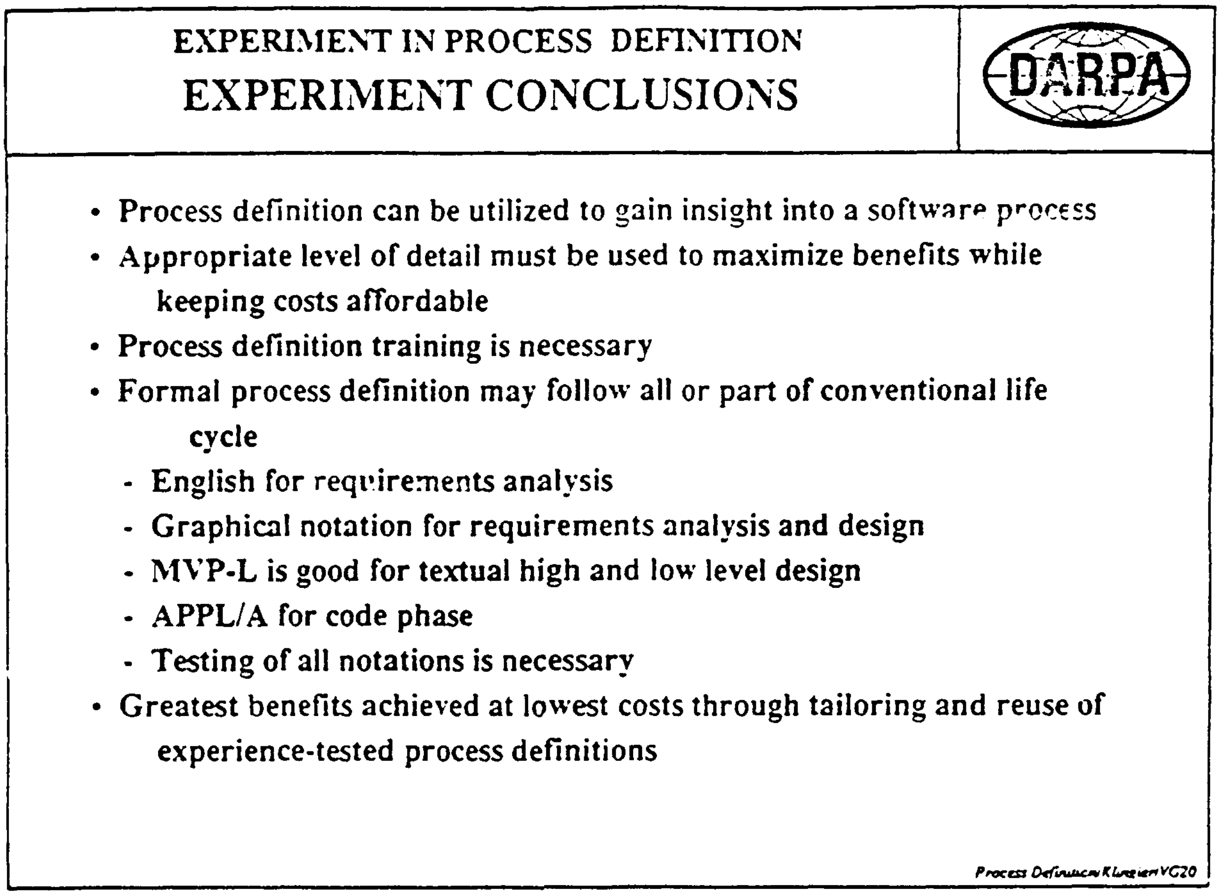
DARPA

- Benefits
  - Improve understanding of a process
  - Enable better communication
  - Discover problems in original English process description
- Costs
  - Creating process descriptions is time consuming
  - Retaining understandability at level of detail needed for enactment is difficult
  - Learning curve is nontrivial
  - Process description experts are vital
- Suitability of experiment notations
  - Different notations for different uses
  - Research prototype status of MVP-L and APPL/A

Process Definition KLetter VG19

This slide details some of the lessons learned about process definition from our experiment using the MVP-L and APPL/A process representation languages. The benefits achieved by defining our process included an improved understanding of the overall process and all of the steps that were involved. This understanding helped us to better organize the process definition. Writing down the process definition in English, graphical notations, and MVP-L made it much easier to communicate the process steps with those who were not familiar with the Certification of Reusable Assets. Another benefit of representing our process in graphical notations and MVP-L was that we discovered required information that was unintentionally left out of our original English description of the process. We are now well into the program layer in our experiment and hope to soon have machine-executable APPL/A code so that we can determine the benefits of machine executability.

There were many costs associated with creating our process definitions. We found that creating process definitions was time consuming and more difficult than we had expected. We were surprised at the large amount of text needed to represent a process formally. At the level of detail needed to represent a process definition in a notation such as APPL/A for machine enactment, it was difficult to retain understandability of the code produced. We also spent more time learning about process definition and programming than we had expected. We found that process definition experts were vital. We did not have experts available at the beginning of our task, but adding experts to the experiment team near the end. These experts were able to produce process definitions in a much shorter time than the novices. As shown on the previous chart, we found that different process notations are suitable for different uses. We also found that, due to the research prototype status of MVP-L and APPL/A there was a lack of user documentation and tool support. Other process notations, including the some used in the STARS '91 demonstrations, are not research prototypes, and would not have presented many of these difficulties.



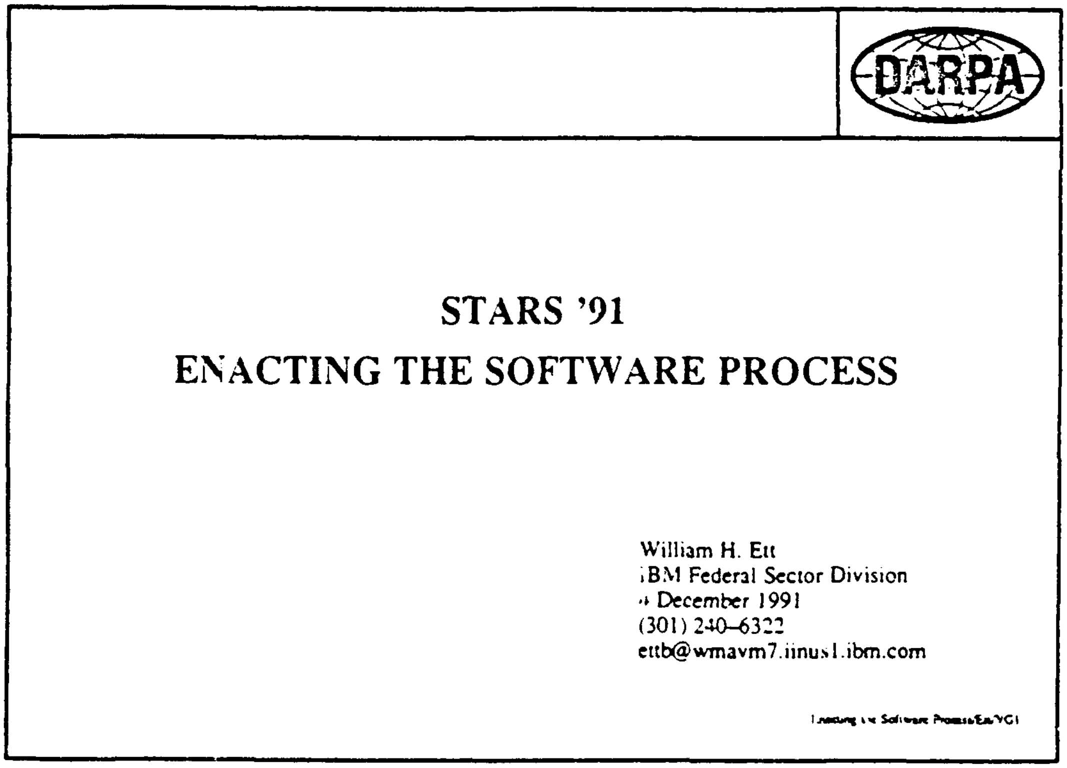
EXPERIMENT IN PROCESS DEFINITION

EXPERIMENT CONCLUSIONS

- Process definition can be utilized to gain insight into a software process
- Appropriate level of detail must be used to maximize benefits while keeping costs affordable
- Process definition training is necessary
- Formal process definition may follow all or part of conventional life cycle
  - English for requirements analysis
  - Graphical notation for requirements analysis and design
  - MVP-L is good for textual high and low level design
  - APPL/A for code phase
  - Testing of all notations is necessary
- Greatest benefits achieved at lowest costs through tailoring and reuse of experience-tested process definitions

Process Definition Klingler VG20

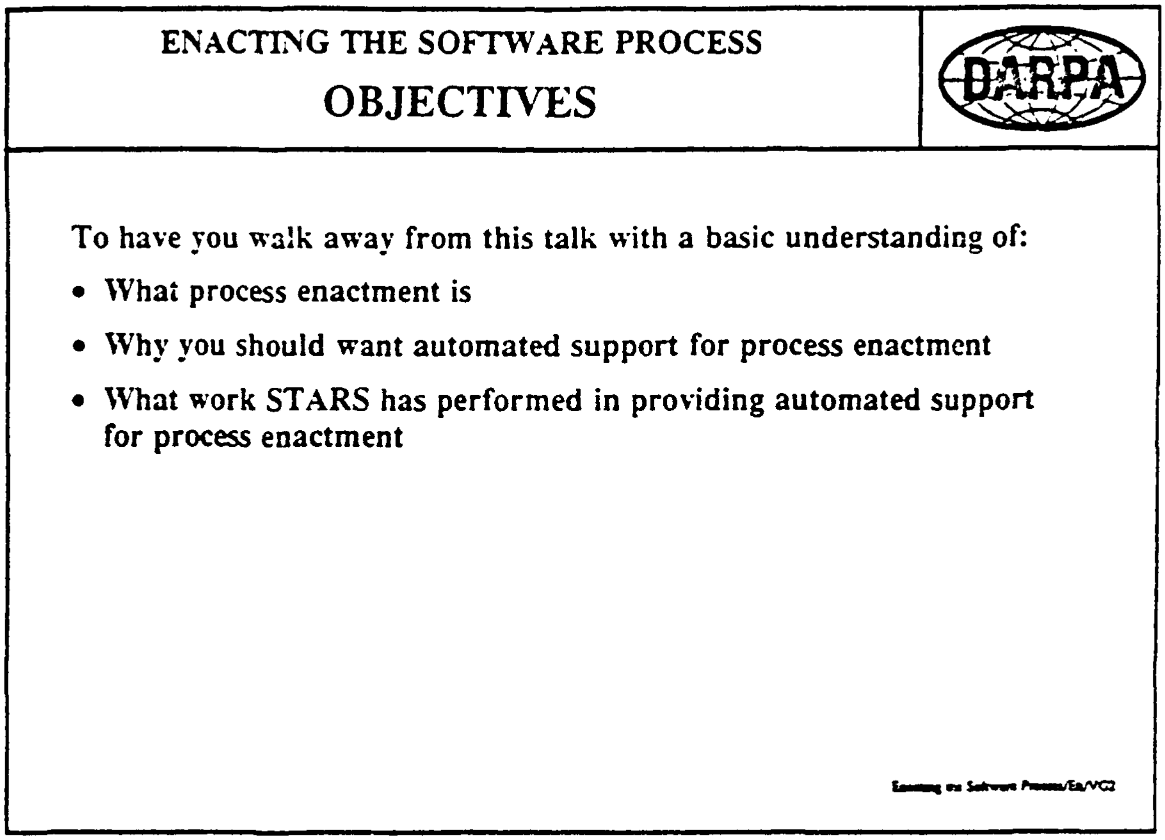
In conclusion, in our experiment we found that process definition can be of great benefit for gaining insight into a software process but it must be used with care. The appropriate level of abstraction must be used that matches the needs of the specific organization, to maximize the benefits of process definition while keeping costs within the project budget. The cost/benefit tradeoff of defining a process is similar to the tradeoff faced when deciding whether to use CASE tool technology on a project. The use of both of these methods requires more money and time spent in the early stages of the software development process for tool purchase and execution of the methodology, but a better quality software product is usually produced. Training in process definition greatly improves the productivity of engineers describing the process. In many respects, process definition does follows the same steps as the software development lifecycle, of requirements analysis, high-level design, low-level design, coding, testing, operation, and maintenance. Depending on the level of detail necessary and the layer(s) that are addressed, the coding step may be skipped. It is important to note that testing and maintenance are necessary for a process definition represented in any notation, in order for the process to evolve and improve. Various notations are appropriate at different steps in the definition of the process, depending on the project goals. However, we have found that great benefits can be achieved at a low cost through the tailoring and reuse of experience-tested process definitions. We invite the audience to read our lessons learned and guidelines for process definition, and to apply them to describe their software processes.



STARS '91
ENACTING THE SOFTWARE PROCESS

William H. Ett
IBM Federal Sector Division
4 December 1991
(301) 240–6322
ettb@wmavm7.iinus1.ibm.com

Enacting the Software Process/Ett/YG1



ENACTING THE SOFTWARE PROCESS

OBJECTIVES

To have you walk away from this talk with a basic understanding of:

- What process enactment is
- Why you should want automated support for process enactment
- What work STARS has performed in providing automated support for process enactment

Enacting the Software Process/Ea/VG2

The subject of process enactment is a difficult subject to cover in detail under a thirty minute time constraint. Therefore, this presentation will outline the basic concepts of process enactment and automated process enactment support, and will identify some of the benefits to be derived by providing automated support for enacting the software process. This knowledge will enable you to understand how software engineering environments and technology to automate process enactment can be applied to help software development teams follow a defined process and facilitate software process improvement within an organization.

Important areas that will not be covered in this presentation are:

1) Process enactment technologies being investigated by STARS. The contract deliverable reports produced by the STARS prime contractors are good sources for this information.

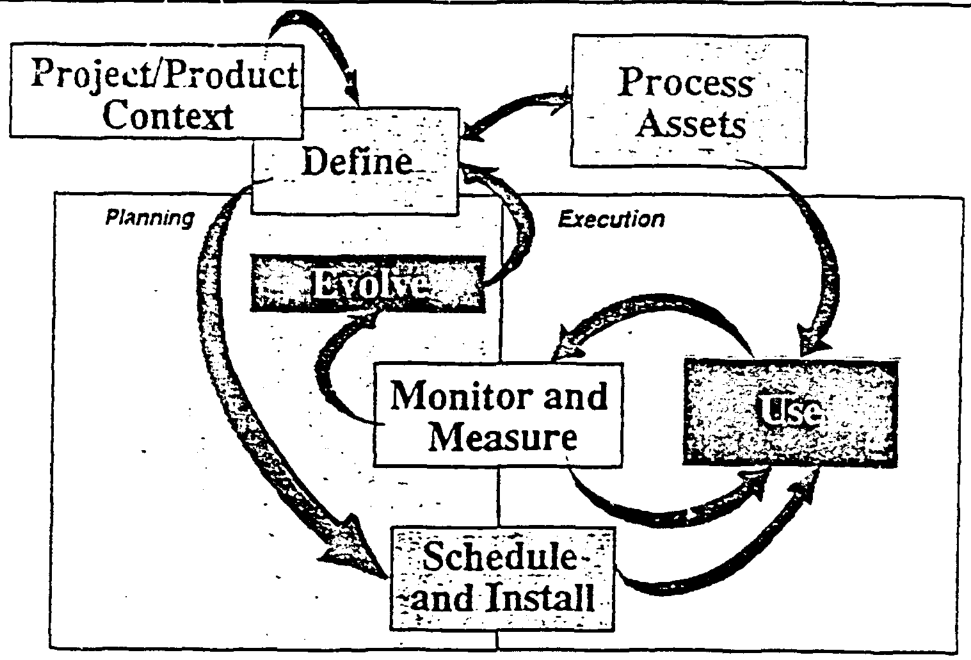
2) Planning for automated process enactment support – how to take a defined process and implement it to support process enactment. The contract deliverables produced by the IBM team's *Cleanroom Software Process Case Study* provide an example of taking a defined process, the *Cleanroom Engineering Software Development Process* and implementing it in a tool called the KI Shell. A similar example can be found in Boeing team's work on policy enactment, in which control point enactment mechanisms were used to implement the case study prepared for the *Sixth International Software Process Workshop.*

We urge you to visit the IBM and Boeing team's process enactment technology demonstrations at STARS '91 to view first hand, the STARS technology point solutions to provide automated support for enacting an organization's process.

PROCESS

PROCESS DRIVEN DEVELOPMENT

DARPA



Adapted from Software Productivity Consortium work.

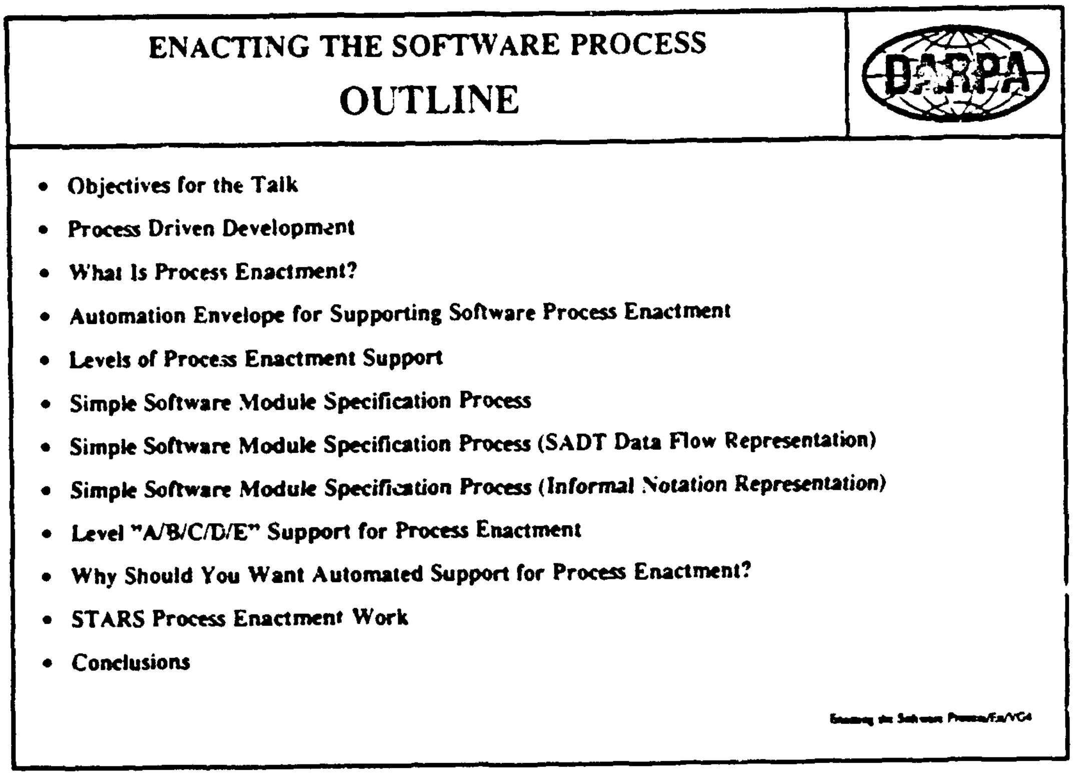
Enacting the Software Process/Ea/VG3

This chart illustrates aspects of process driven development from project and process planning, process definition, and process implementation to process enactment (*execution* or *performance*). Support for enacting the software process, whether automated or not, deals with supporting an enactment agent to:

- Follow (use) a defined software process
- Capture measurements to permit process performance analysis and improvement
- Manage process state and history data to permit management reporting on the state of process and project activities.

Before processes can be scheduled and installed, they must be planned, modeled, implemented and tested. Planning for process enactment may require the ability to simulate processes prior to their implementation, thus automated process enactment support may also serve a role in testing process definitions.

This presentation will focus on process enactment support in the "execution" area of the chart, where whether we provide automated support for enacting the process or not, our focus is on support for the *execution* or *performance* of a defined process.

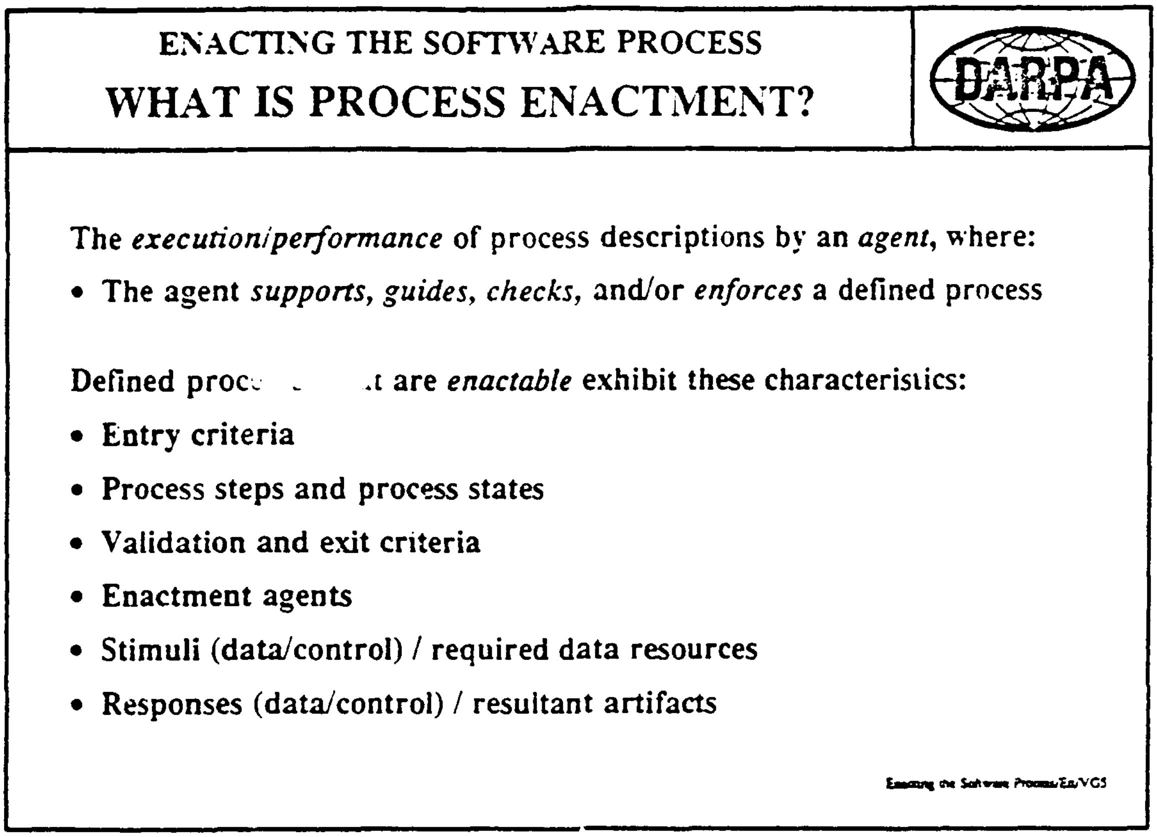


ENACTING THE SOFTWARE PROCESS

OUTLINE

- Objectives for the Talk
- Process Driven Development
- What Is Process Enactment?
- Automation Envelope for Supporting Software Process Enactment
- Levels of Process Enactment Support
- Simple Software Module Specification Process
- Simple Software Module Specification Process (SADT Data Flow Representation)
- Simple Software Module Specification Process (Informal Notation Representation)
- Level "A/B/C/D/E" Support for Process Enactment
- Why Should You Want Automated Support for Process Enactment?
- STARS Process Enactment Work
- Conclusions

Enacting the Software Process/Ea/VG4



ENACTING THE SOFTWARE PROCESS

WHAT IS PROCESS ENACTMENT?

DARPA

The *execution/performance* of process descriptions by an *agent*, where:

- The agent *supports, guides, checks,* and/or *enforces* a defined process

Defined proc.. - .t are *enactable* exhibit these characteristics:

- Entry criteria
- Process steps and process states
- Validation and exit criteria
- Enactment agents
- Stimuli (data/control) / required data resources
- Responses (data/control) / resultant artifacts

Enacting the Software Process/Ex/VGS

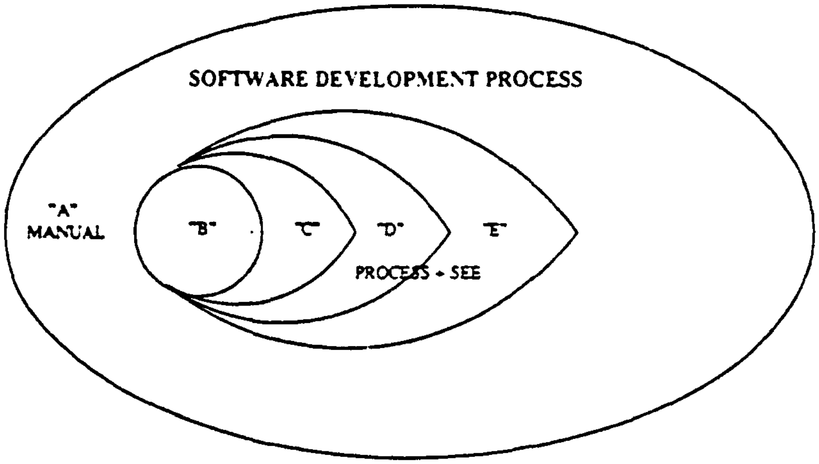
Process enactment is the *execution* or *performance* of process descriptions by an agent, where the agent enacting the process is a human or a computer system that is provided with sufficient knowledge.

Regardless of the agent of enactment, the agent's purpose is to support, guide, check, and enforce if necessary, the following characteristics of a *defined* process:

- Defined entry criteria – the conditions that must be met, before the process can be enacted
- A set of process steps, and the identification of internal process states and state transition conditions
- Validation and exit criteria – that must be met before the results of the process can be passed to other processes
- One or more agents to enact the process
- Stimuli (data/control)
- Responses (data/control) / Resultant artifacts.

ENACTING THE SOFTWARE PROCESS
AUTOMATION ENVELOPE FOR SUPPORTING SOFTWARE PROCESS ENACTMENT

DARPA



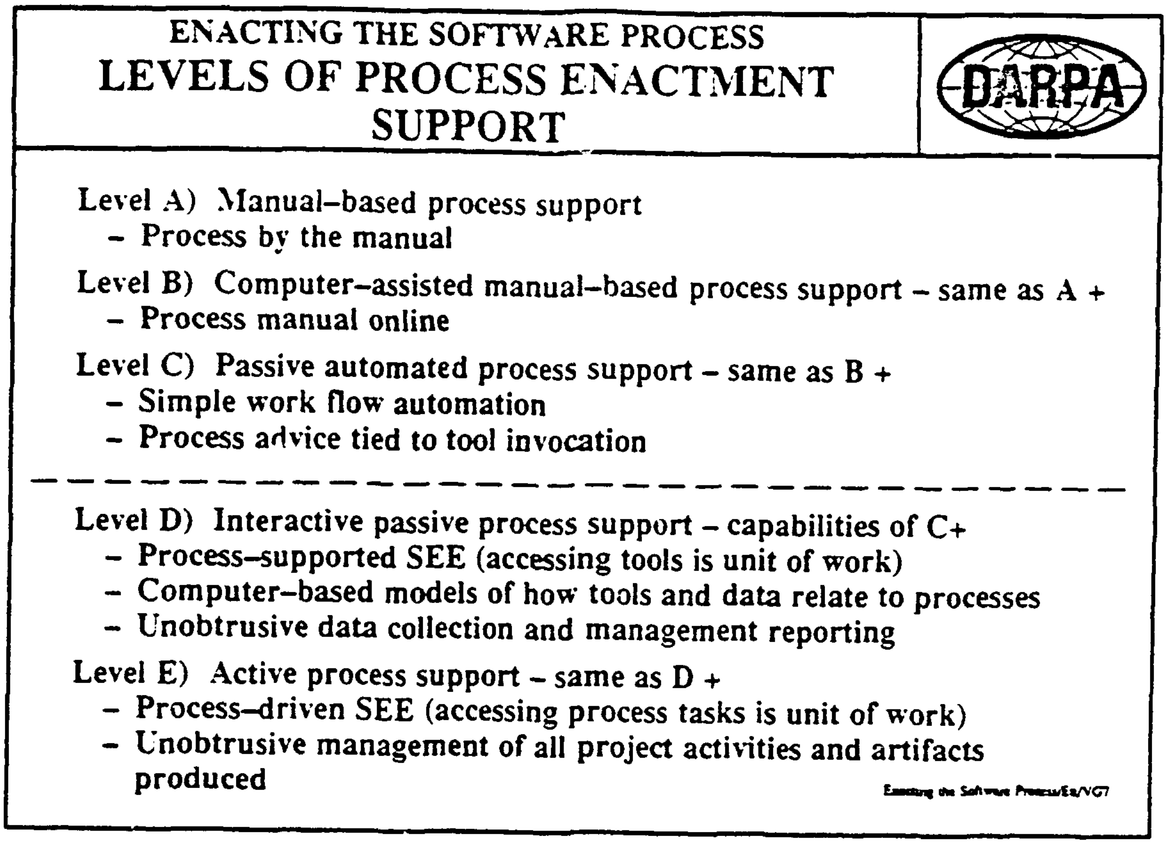
Enacting the Software Process/En/VC4

The software development process for an organization represents a system of processes put into place to meet the organization's or project's objectives of developing computer software. As in other disciplines, the ability to provide automated support depends largely on how well the processes to be automated are understood and on whether the computer systems can be provided with sufficient knowledge and capability to perform the processes.

Accounting systems existed thousands of years before the advent of the first calculator. When computers came of age, however, accounting functions were analyzed for potential automation, and subsequently many bookkeeping systems were automated. At the advent of online transaction data processing, some of these bookkeeping systems were replaced and other transaction–oriented bookkeeping applications were automated. At the advent of expert systems technology, accounting functions were analyzed to see what accounting processes this new technology could automate. Financial statement preparation was identified as a target for automation, when it was felt that an expert system could be developed to automate "routine" statement preparation tasks, so that humans could be left to produce the complicated statements that current expert systems just could not hope to prepare. An accounting system for an organization is still a complete system of methods and practices for performing bookkeeping and financial statement preparation. Automation has only redistributed how bookkeepers and accountants time is spent in supporting the accounting system.

This same analogy holds for developing systems to provide automated support for enacting an organization's software development process. We hope to redistribute how software development professional's time will be spent in developing software, permitting software engineers to concentrating on the creative aspects of software development, while still meeting our process measurement and improvement, and product quality objectives.

There are two important aspects to consider with respect to providing automated support for enacting the software process, namely (1) our ability to automate manual steps and (2) how our automation ability may change the process itself.



ENACTING THE SOFTWARE PROCESS

LEVELS OF PROCESS ENACTMENT SUPPORT

Level A) Manual–based process support
- Process by the manual

Level B) Computer–assisted manual–based process support – same as A +
- Process manual online

Level C) Passive automated process support – same as B +
- Simple work flow automation
- Process advice tied to tool invocation

Level D) Interactive passive process support – capabilities of C+
- Process–supported SEE (accessing tools is unit of work)
- Computer–based models of how tools and data relate to processes
- Unobtrusive data collection and management reporting

Level E) Active process support – same as D +
- Process–driven SEE (accessing process tasks is unit of work)
- Unobtrusive management of all project activities and artifacts produced

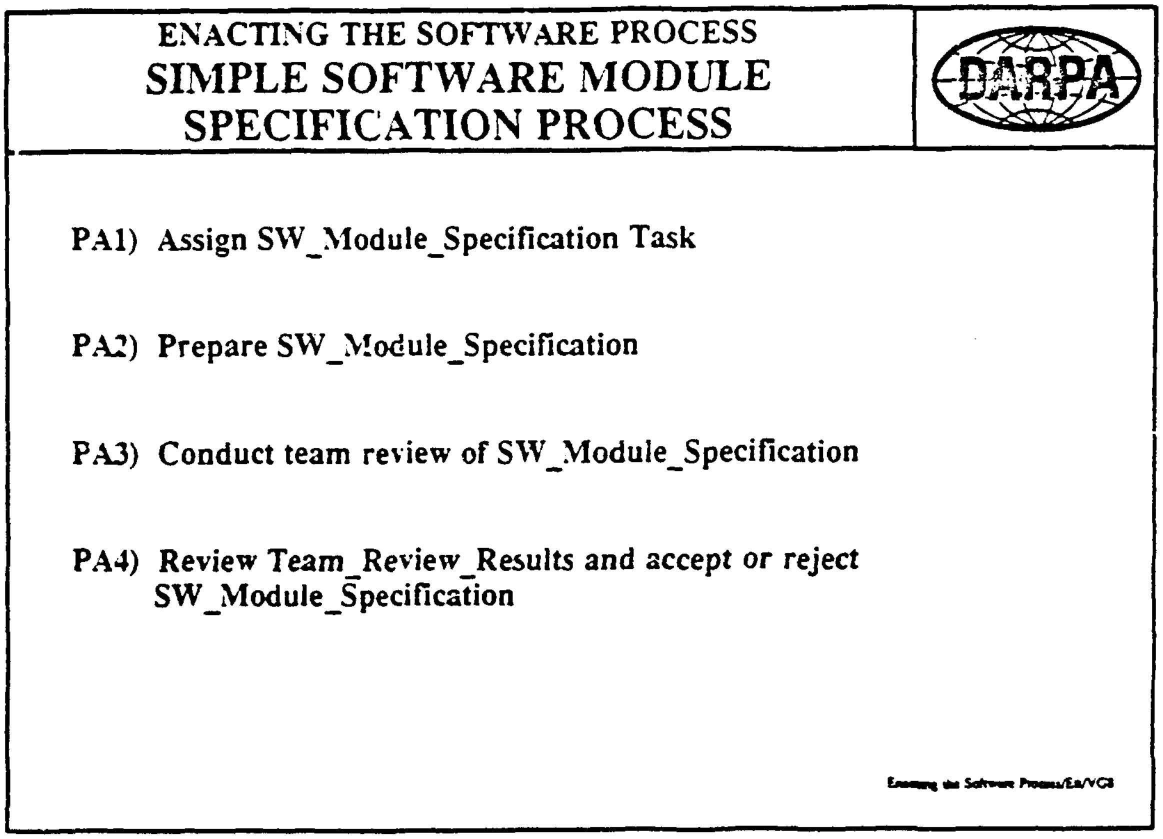
Enacting the Software Process/Ea/VG7

This chart characterizes five levels of support for enacting software processes. The purpose of this chart is not to establish a ranking system of process enactment support, but to identify levels associated with our ability to extend the "automation envelope" in providing process enactment support." Given today's state of technology in software process management, the entry criterion is a high one – an organization must have a "defined" process, as having a "defined" process is a precondition to plan for any automated process enactment support.

Levels "B" and "C" relate to computer–assisted process enactment support where the organization has a defined process for performing software development, but does so using tools that provide limited integration.

Levels "D" and "E" relate to computer–assisted process enactment support where the organization has invested in a modern software engineering environment and wishes to provide finer–grained process support for enacting the software process and for automatically performing necessary work steps.

The major difference between level "D" and "E" is how process is presented to the user and how process support is planned. At level D, the main unit of work is the invocation of tools, where finer–grained process enactment support can be provided than at level C. The intent of level "D" process enactment automation is to bring process activities and advice closer to users through their use of tools instantiated into the software engineering environment. At level "E," the process for a project is a well–planned and designed system, where the unit of work in the software engineering environment is through invoking process steps, where tools and data are made available for performing each process step.



ENACTING THE SOFTWARE PROCESS

SIMPLE SOFTWARE MODULE SPECIFICATION PROCESS

PA1) Assign SW_Module_Specification Task

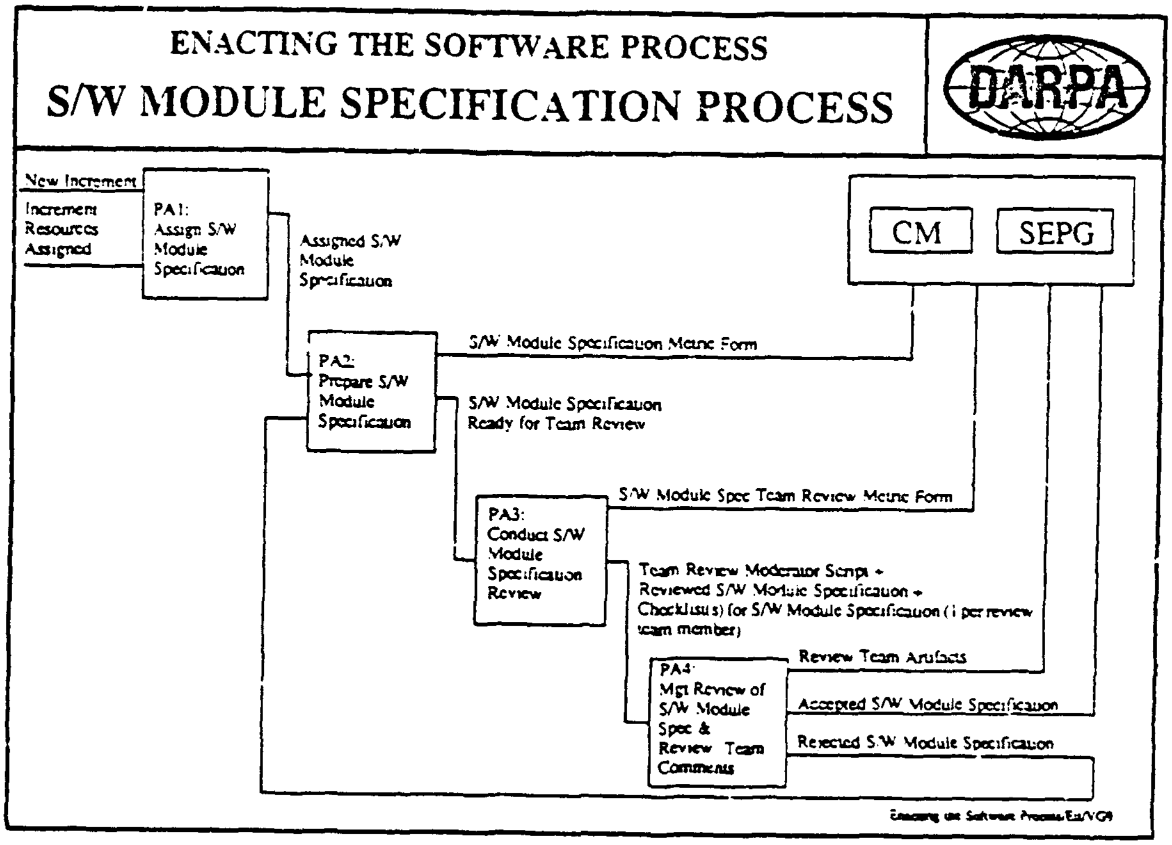
PA2) Prepare SW_Module_Specification

PA3) Conduct team review of SW_Module_Specification

PA4) Review Team_Review_Results and accept or reject SW_Module_Specification

Enacting the Software Process/Ea/VG8

This represents a simple process for preparing *Software Module Specifications*. As humans, we can view a process like this and have a good understanding of what is involved in following it. It does not, however, meet our definition of an *enactable process*.



ENACTING THE SOFTWARE PROCESS

S/W MODULE SPECIFICATION PROCESS

DARPA

New Increment

Increment Resources Assigned

PA1: Assign S/W Module Specification

Assigned S/W Module Specification

CM

SEPG

PA2: Prepare S/W Module Specification

S/W Module Specification Metric Form

S/W Module Specification Ready for Team Review

PA3: Conduct S/W Module Specification Review

S/W Module Spec Team Review Metric Form

Team Review Moderator Script + Reviewed S/W Module Specification + Checklist(s) for S/W Module Specification (1 per review team member)

PA4: Mgt Review of S/W Module Spec & Review Team Comments

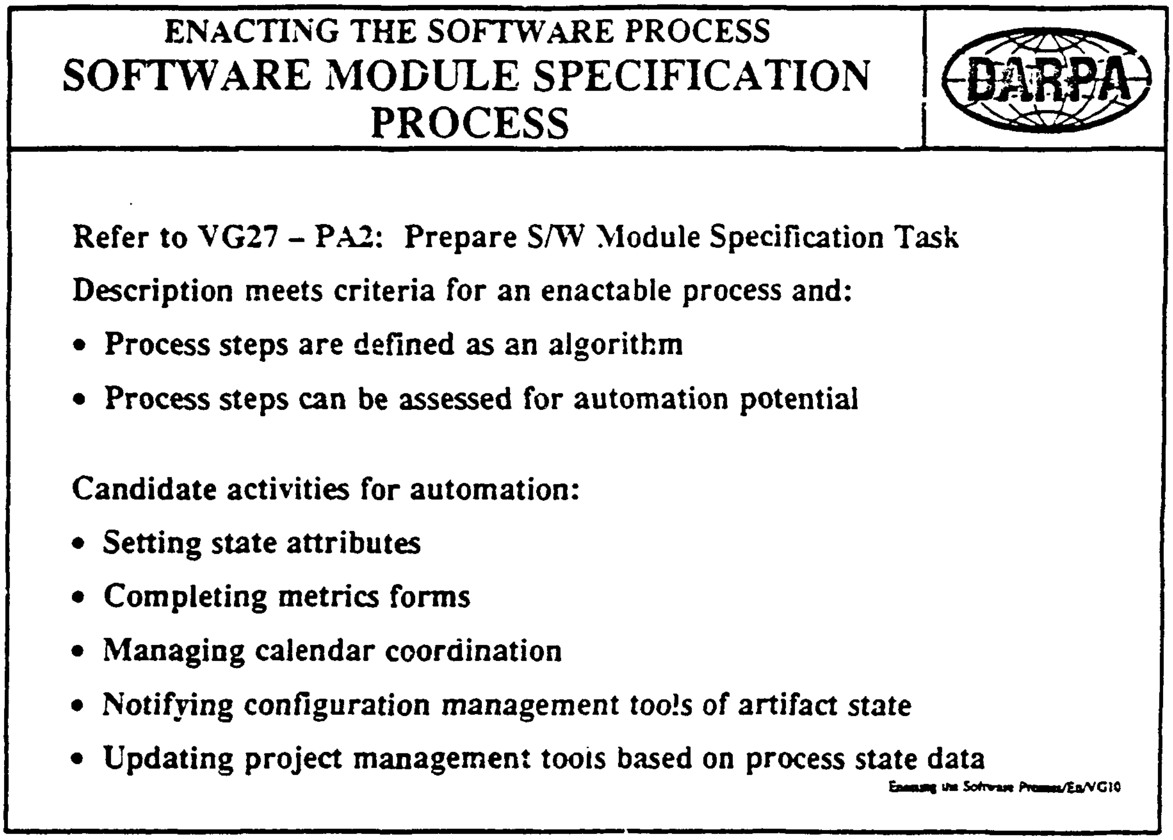
Review Team Artifacts

Accepted S/W Module Specification

Rejected S/W Module Specification

Enacting the Software Process/Ea/VG9

This representation of our simple process for preparing *Software Module Specifications* illustrates resources required for the process steps and artifacts produced. From this view, we can better understand how the process steps relate to each other by examining the data flow. This process description still does not meet our definition of an *enactable process*, however.



ENACTING THE SOFTWARE PROCESS

SOFTWARE MODULE SPECIFICATION PROCESS

Refer to VG27 – PA2: Prepare S/W Module Specification Task

Description meets criteria for an enactable process and:

- Process steps are defined as an algorithm
- Process steps can be assessed for automation potential

Candidate activities for automation:

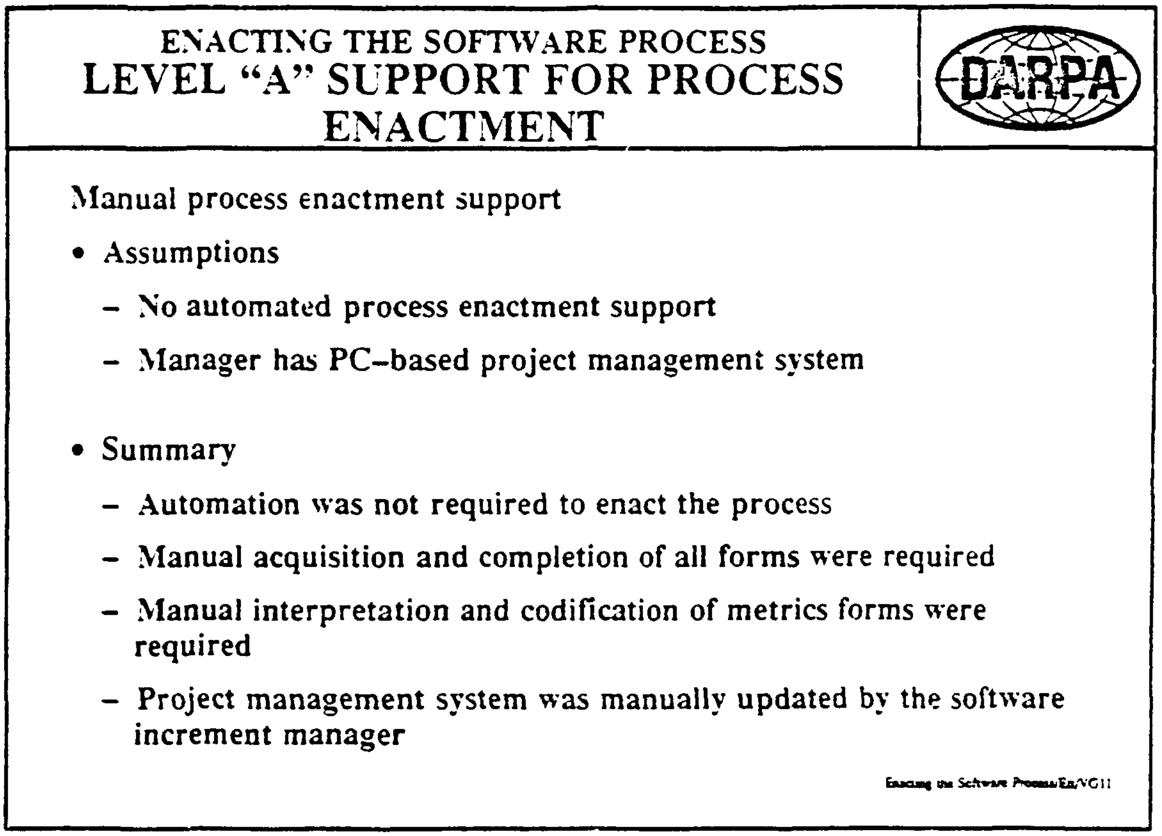
- Setting state attributes
- Completing metrics forms
- Managing calendar coordination
- Notifying configuration management tools of artifact state
- Updating project management tools based on process state data

Enacting the Software Process/Ea/VG10

A more detailed process description can be found on charts VG23a through VG30. If you refer to chart VG27 –*PA2: Prepare S/W Module Specification Task*, you will observe that this process meets our criteria for an enactable process. In addition, the process is defined as an algorithm. Further, because we have taken a more formal approach to describing the steps of our process, these steps can be more easily assessed for their automation potential.

We have identified some activities that are candidates for automation:

- Process state attributes could be automatically set at the completion of process steps
- Assistance for completing metrics forms could be provided, where process metrics could be captured as the work was performed and automatically included on the form
- Assistance for managing calendar coordination could be provided, depending on ability of the tool(s) selected to support calendar management
- Assistance for communicating the state of artifacts to a configuration management tool
- Assistance for updating project management tools, based on changes in selected process states, e.g., update the project management system database when a milestone has been reached



ENACTING THE SOFTWARE PROCESS

LEVEL “A” SUPPORT FOR PROCESS ENACTMENT

DARPA

Manual process enactment support

- Assumptions
  - No automated process enactment support
  - Manager has PC–based project management system
- Summary
  - Automation was not required to enact the process
  - Manual acquisition and completion of all forms were required
  - Manual interpretation and codification of metrics forms were required
  - Project management system was manually updated by the software increment manager

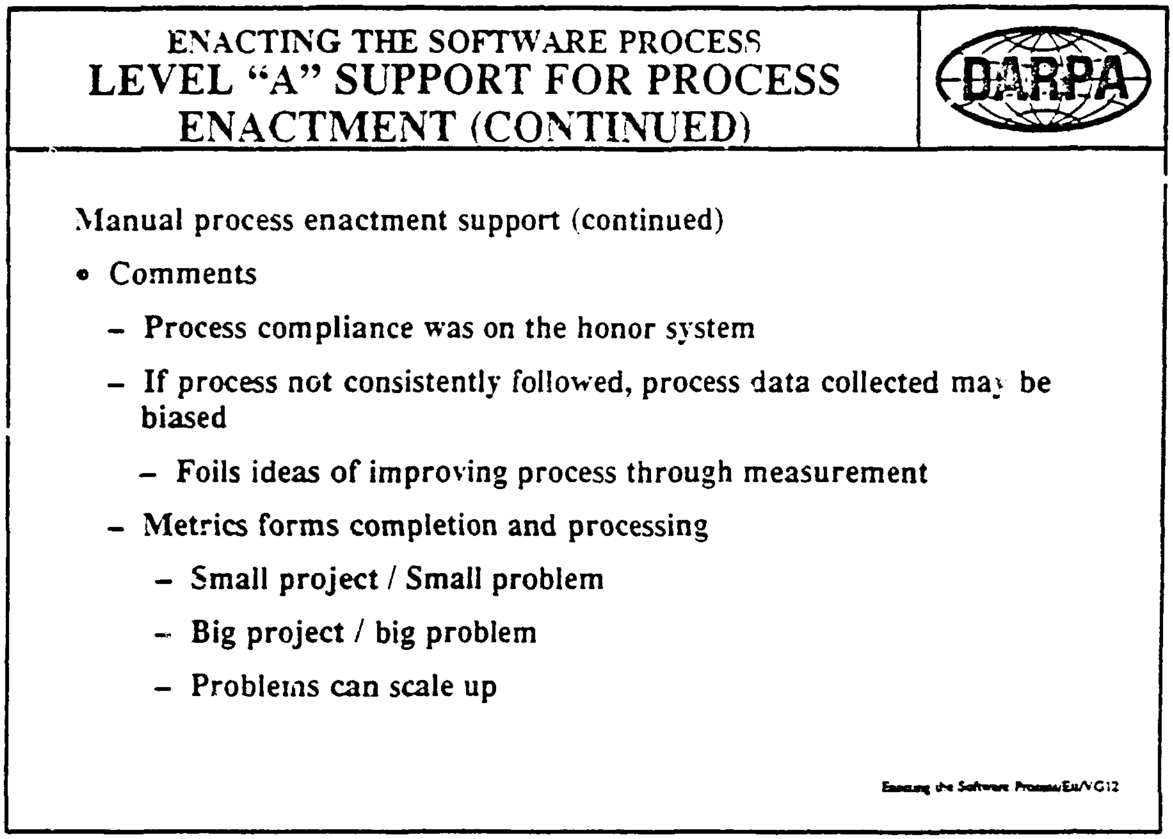
Enacting the Software Process/Ea/VG11

This chart characterizes what it would be like to manually enact our example process.

The following are assumed:

- The process is manually enacted.
- The software increment manager uses his or her favorite PC–based project management system to plan and manage the software increment.

Automation was not required to enact the process. Therefore, none of the benefits of automation were realized. All required forms needed to be manually prepared. Paper–based forms were submitted to the SEPG for manual interpretation and codification. The project management system needed to be manually updated by the software increment manager.



ENACTING THE SOFTWARE PROCESS

LEVEL "A" SUPPORT FOR PROCESS ENACTMENT (CONTINUED)

Manual process enactment support (continued)

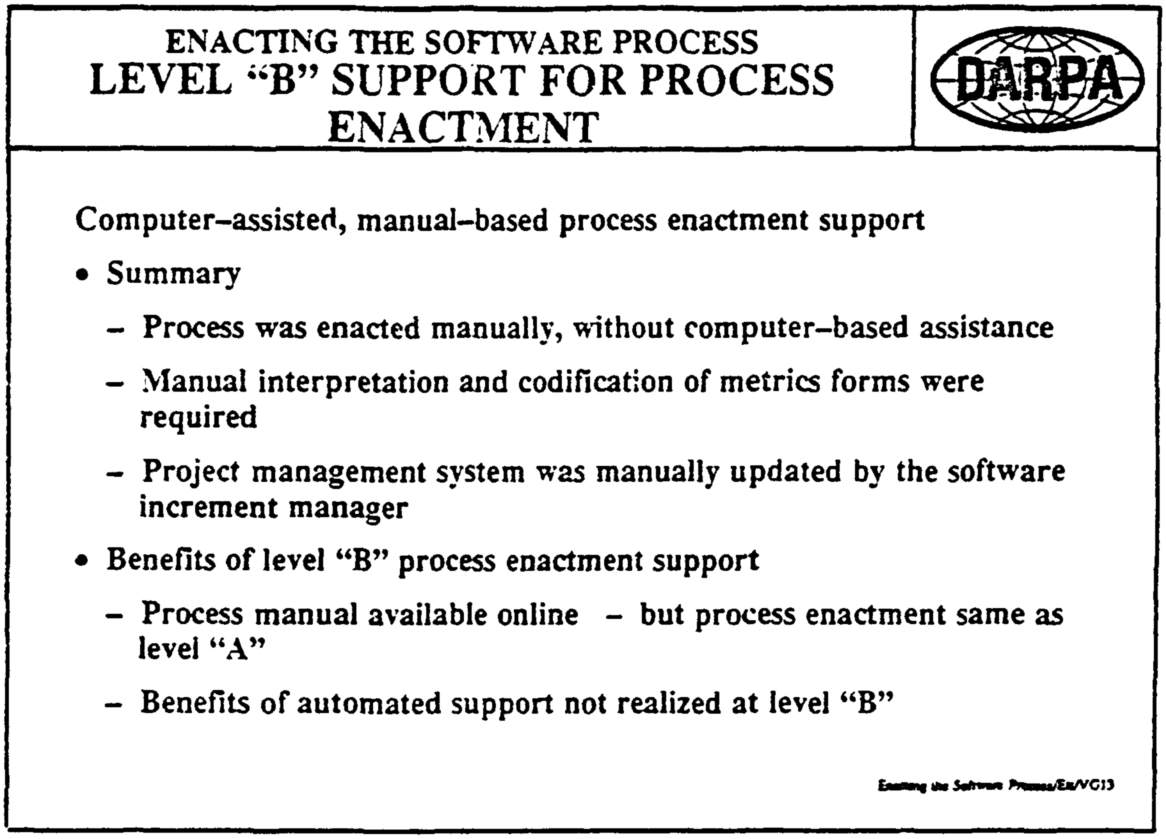
- Comments
  - Process compliance was on the honor system
  - If process not consistently followed, process data collected may be biased
    - Foils ideas of improving process through measurement
  - Metrics forms completion and processing
    - Small project / Small problem
    - Big project / big problem
    - Problems can scale up

Enacting the Software Process/Eu/VG12

Compliance to following our example process was on the "honor system." This is acceptable if software development personnel are trained in the processes they are expected to follow. However, if the processes are not consistently followed, data collected on the process (process and product metrics) will potentially be biased. This bias foils the concept of process analysis and improvement through measurement.

Regarding the metrics forms completion and processing, we have examined a number of paper–based metric collection systems. They tell us that they are probably only going to be a small problem for a small project. However, small problems can scale up to become big problems in big projects and depending on the size and organization of the project, problems may not necessarily scale linearly.

Note: We view processes as being flexibly defined but rigorously enforced. This means that processes should not be designed to "control" the software engineer's every move. It simply means that along with the creative steps of software development, there are process steps to ensure the quality of the products we produce and to collect data to determine how we can make our processes for developing software better. The ultimate goal is to minimize "busy work" for software engineers, providing them more time to spend on the creative activities of software engineering.



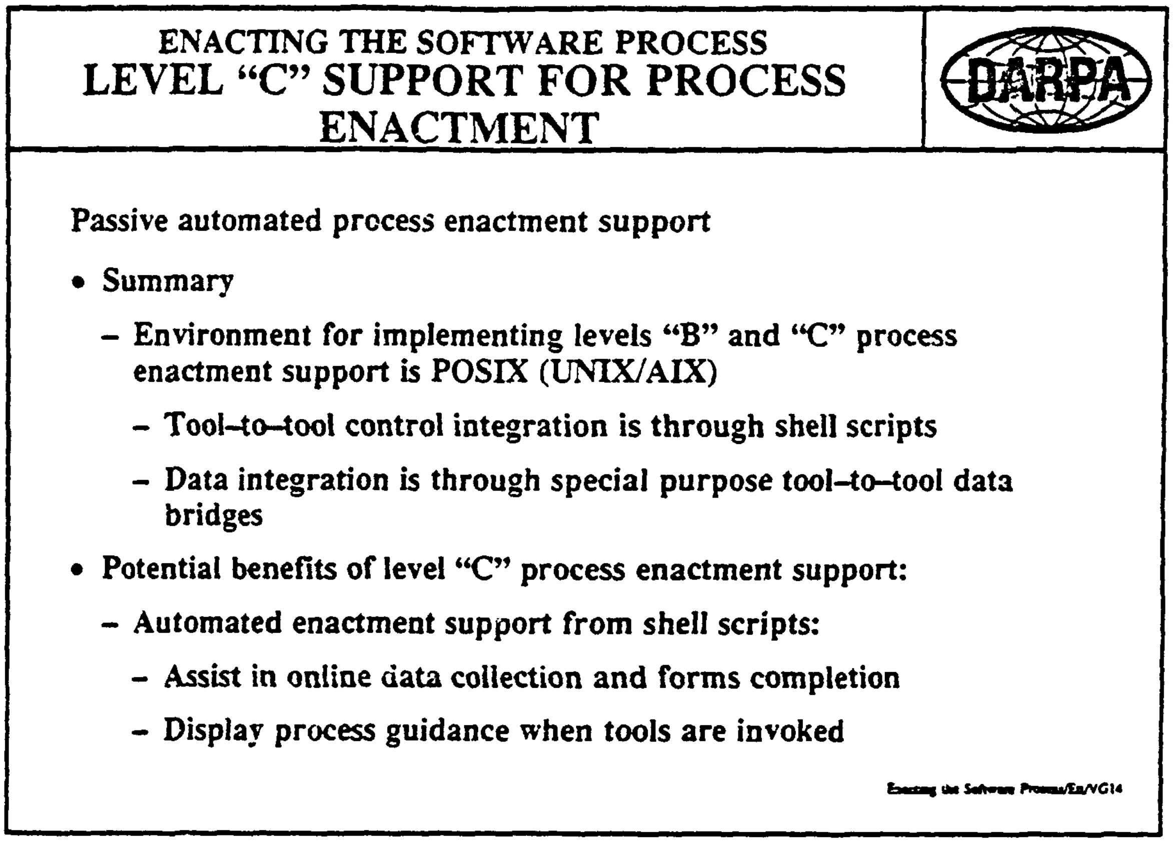
ENACTING THE SOFTWARE PROCESS
LEVEL "B" SUPPORT FOR PROCESS ENACTMENT

Computer–assisted, manual–based process enactment support

- Summary
  - Process was enacted manually, without computer–based assistance
  - Manual interpretation and codification of metrics forms were required
  - Project management system was manually updated by the software increment manager
- Benefits of level "B" process enactment support
  - Process manual available online – but process enactment same as level "A"
  - Benefits of automated support not realized at level "B"

Enacting the Software Process/En/VG13

All of the comments that applied to level "A" basically apply to level "B" as well. The big difference is that the process software developers are expected to follow is available online. Forms that the process requires could probably be extracted from the online process, and thus the forms could be completed using a text editor or word processor. However, these forms would still have to be manually processed.



ENACTING THE SOFTWARE PROCESS
LEVEL "C" SUPPORT FOR PROCESS ENACTMENT

Passive automated process enactment support

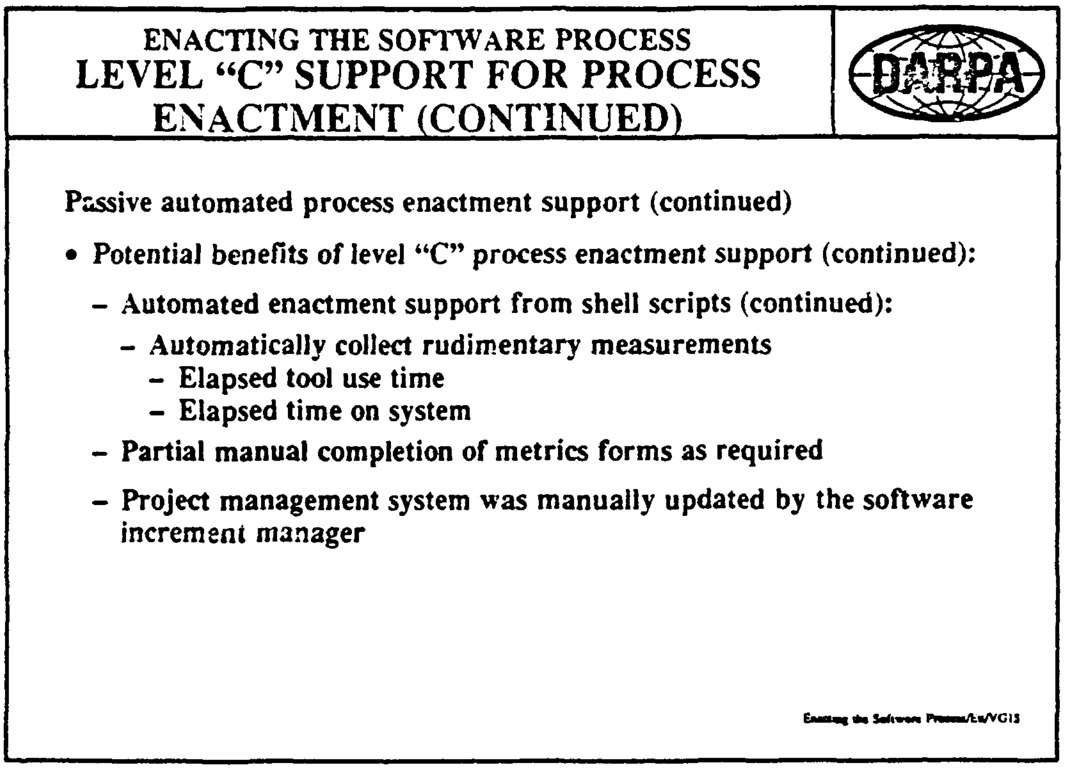
- Summary
  - Environment for implementing levels "B" and "C" process enactment support is POSIX (UNIX/AIX)
    - Tool–to–tool control integration is through shell scripts
    - Data integration is through special purpose tool–to–tool data bridges
- Potential benefits of level "C" process enactment support:
  - Automated enactment support from shell scripts:
    - Assist in online data collection and forms completion
    - Display process guidance when tools are invoked

Enacting the Software Process/Ea/VG14

Computing environments to support level "B" and level "C" are typically POSIX–compliant environments such as AIX or UNIX. At level "C," only coarse–grained process steps can easily be implemented to provide support for enacting the software process. This granularity is determined by the tools selected and the program integration capability they afford.

At level "C," potential benefits can be realized from providing automated process enactment support. Support for completing the forms required by the process can be implemented and tied into shell scripts to appear either before or after a tool is invoked. Process guidance could be provided upon tool invocation. Further, rudimentary process metrics could be collected from shell scripts, such as *elapsed tool use time* and *elapsed time on system*, associated with a particular work item. Metrics forms could be partially completed automatically, but would still rely on the software engineer to manually complete the form, based on the notes and records he/she kept.

Project management data on activities and milestones still require manual updating by the software increment manager.



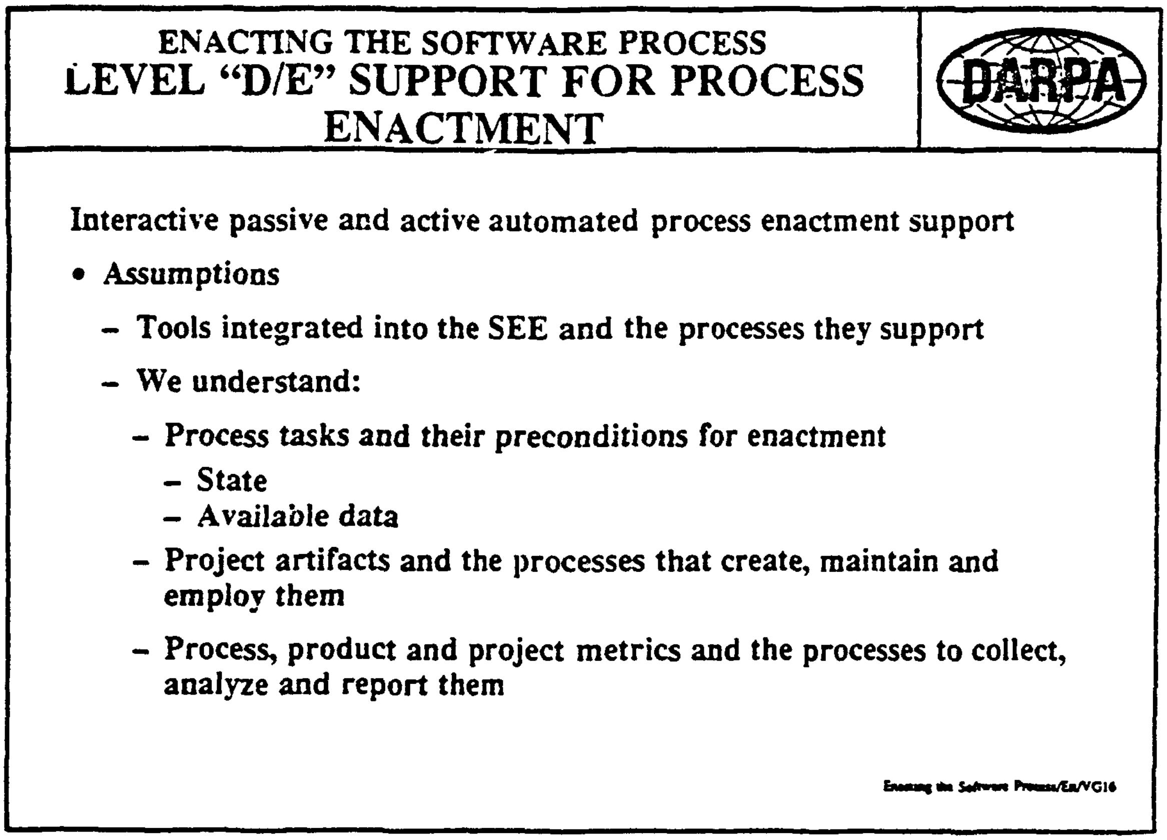
ENACTING THE SOFTWARE PROCESS

LEVEL "C" SUPPORT FOR PROCESS ENACTMENT (CONTINUED)

Passive automated process enactment support (continued)

- Potential benefits of level "C" process enactment support (continued):
  - Automated enactment support from shell scripts (continued):
    - Automatically collect rudimentary measurements
      - Elapsed tool use time
      - Elapsed time on system
  - Partial manual completion of metrics forms as required
  - Project management system was manually updated by the software increment manager

Enacting the Software Process/Lu/VG15



ENACTING THE SOFTWARE PROCESS
LEVEL “D/E” SUPPORT FOR PROCESS ENACTMENT

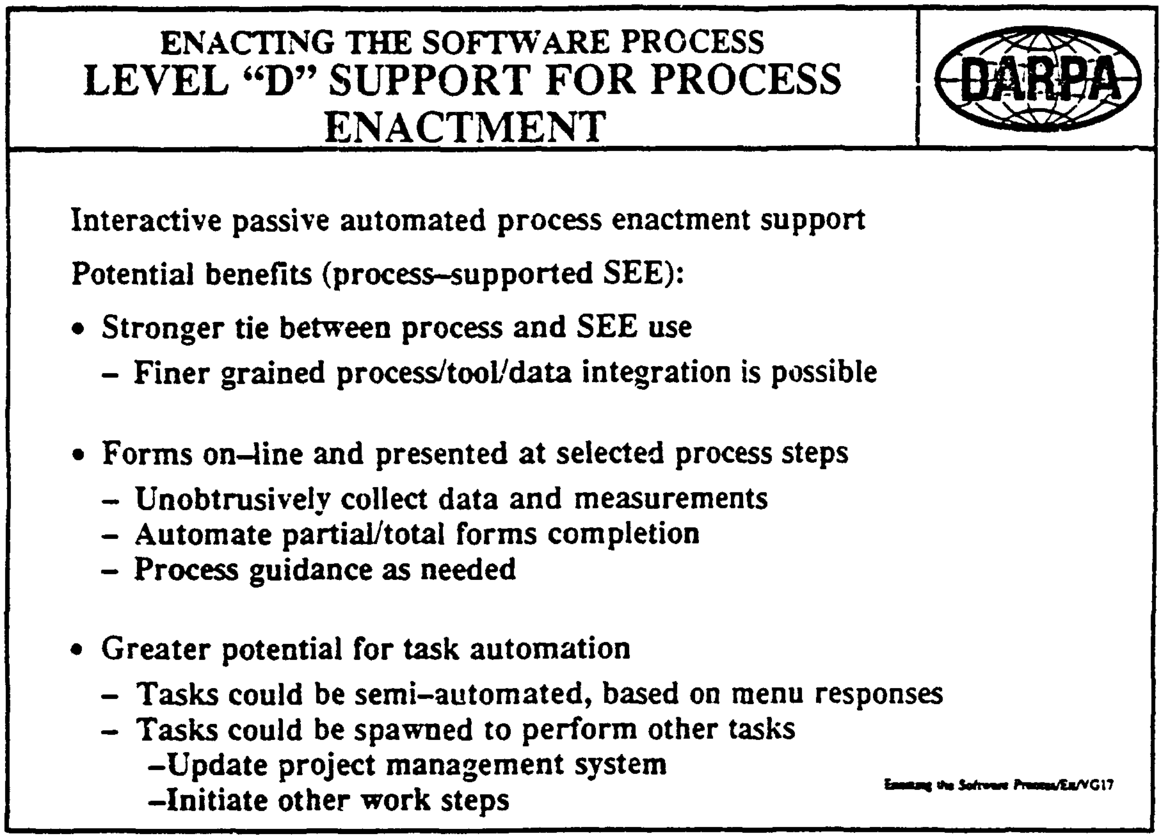
Interactive passive and active automated process enactment support

- Assumptions
  - Tools integrated into the SEE and the processes they support
  - We understand:
    - Process tasks and their preconditions for enactment
      - State
      - Available data
    - Project artifacts and the processes that create, maintain and employ them
    - Process, product and project metrics and the processes to collect, analyze and report them

Enacting the Software Process/Ea/VG16

At levels "D" and "E," we assume that process enactment support will be integrated with a modern software engineering environment where processes are defined to the SEE, as well as the tools and data required to support them. Process enactment support applications can be provided with knowledge of:

- The preconditions necessary for enacting a process step, and the rules for transitioning between process states
- Project artifacts and the processes that can create, maintain, manipulate and/or employ them
- Process, product, and project metrics and the processes that need to collect, analyze, and report them.



ENACTING THE SOFTWARE PROCESS

LEVEL "D" SUPPORT FOR PROCESS ENACTMENT

Interactive passive automated process enactment support

Potential benefits (process–supported SEE):

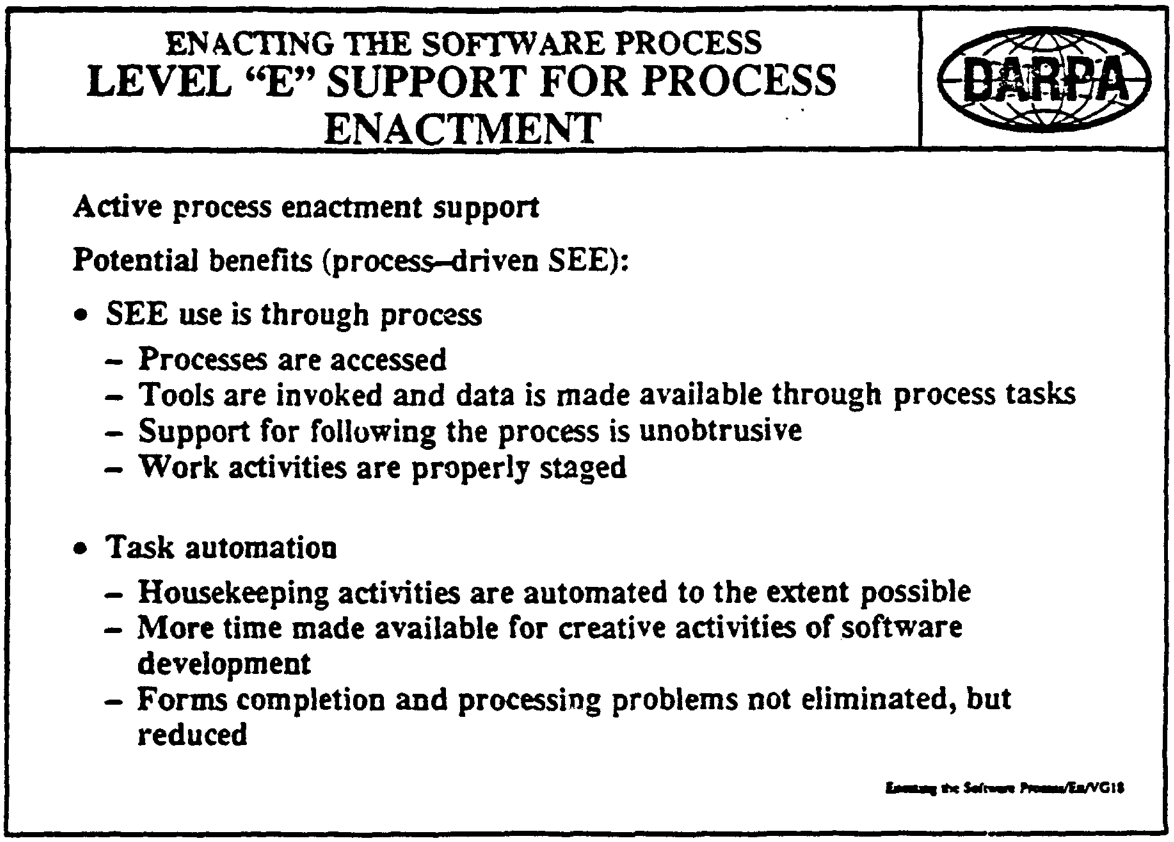
- Stronger tie between process and SEE use
  - Finer grained process/tool/data integration is possible
- Forms on–line and presented at selected process steps
  - Unobtrusively collect data and measurements
  - Automate partial/total forms completion
  - Process guidance as needed
- Greater potential for task automation
  - Tasks could be semi–automated, based on menu responses
  - Tasks could be spawned to perform other tasks
    - –Update project management system
    - –Initiate other work steps

Enacting the Software Process/Ex/VG17

Among the potential benefits, at levels "D" and "E," we have the ability to provide a stronger tie between the process enactment support tool(s) and the SEE. At level "D," we are preparing the *process–supported SEE.*

With the *process–supported SEE* we want to provide finer grained process/tool/data integration to provide:

- Process support where process support is needed, not just where it is convenient to implement;
- Better automated support for metrics collection and required forms completion. For example, at level "C" we could only capture data on elapsed time using a tool or elapsed time working on a system for a given work item. At level "D" we can collect metrics and data associated with a process step, based on the results of the process step. Consequently, some of the process metrics that we required the software engineer to manually maintain can now be totally captured automatically, relieving the requirement to complete process metrics forms entirely;
- Greater potential for task automation where tasks could be semi–automated, based on human responses to menus, etc., and where tasks can be automatically spawned to perform other tasks, such as the automatic updating of a project management system when a milestone has been reached.



ENACTING THE SOFTWARE PROCESS

LEVEL "E" SUPPORT FOR PROCESS ENACTMENT

Active process enactment support

Potential benefits (process–driven SEE):

- SEE use is through process
  - Processes are accessed
  - Tools are invoked and data is made available through process tasks
  - Support for following the process is unobtrusive
  - Work activities are properly staged

- Task automation
  - Housekeeping activities are automated to the extent possible
  - More time made available for creative activities of software development
  - Forms completion and processing problems not eliminated, but reduced

Enacting the Software Process/En/VG18

At level "E" our focus is to make SEE users process users, where the SEE is used through the process. At this level, we are preparing the *process–driven SEE.*

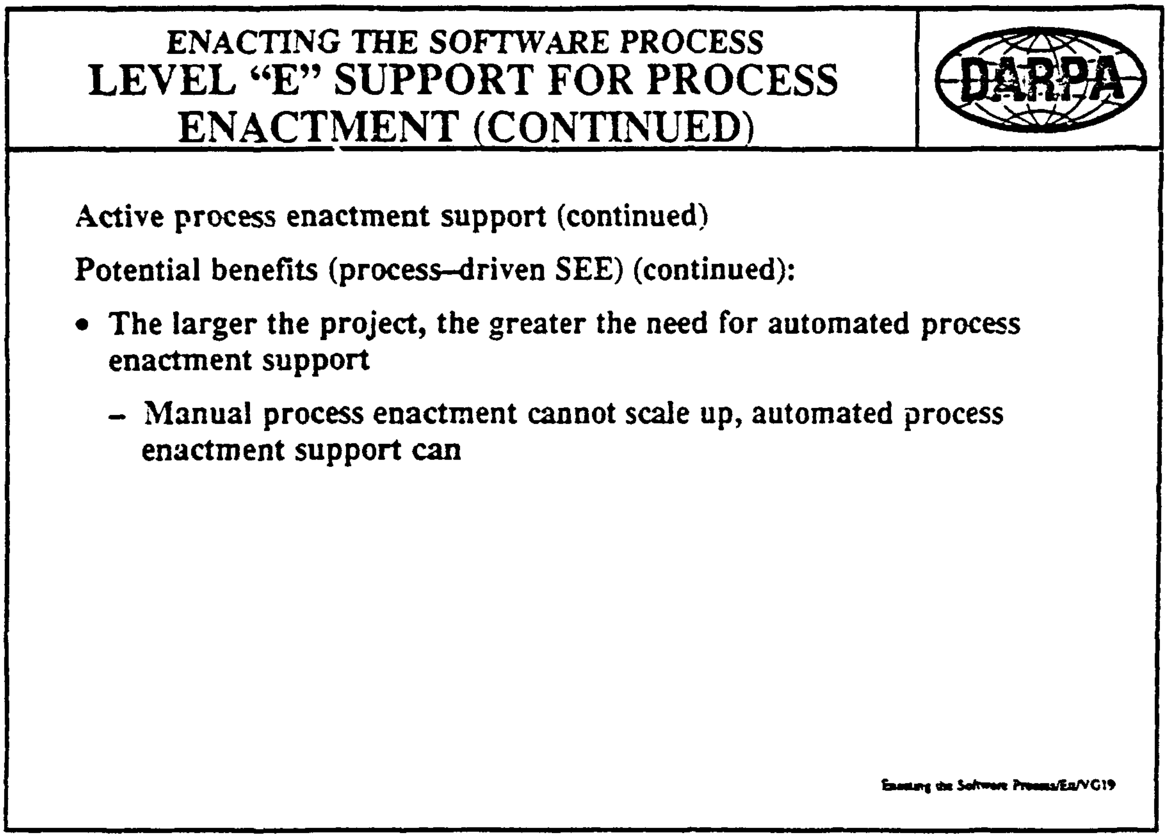
With the *process–driven SEE* :

- Process tasks are accessed, not tools. Tools are invoked and data is made available through the invocation of process tasks.

- Process support is unobtrusive. Users still do the same work they did before and probably use the same tools. The real difference is their method of invocation and the automatic housekeeping being performed.

- Process tasks are properly staged to present tasks to users when the conditions for performing the tasks have been met.

For task automation, housekeeping activities, such as metric and data collection are automated to the extent possible. Further, tasks not requiring human invocation where the necessary preconditions have been met, can be automatically enacted. Although forms completion and forms processing problems will probably not be totally eliminated, the effort required to complete and process them will be reduced. With this automation we hope to achieve our goals of making more time available to the software engineer to concentrate on the creative activities of software development, while unobtrusively collecting metrics from a consistently applied process to facilitate process analysis and improvement.



ENACTING THE SOFTWARE PROCESS

LEVEL "E" SUPPORT FOR PROCESS ENACTMENT (CONTINUED)

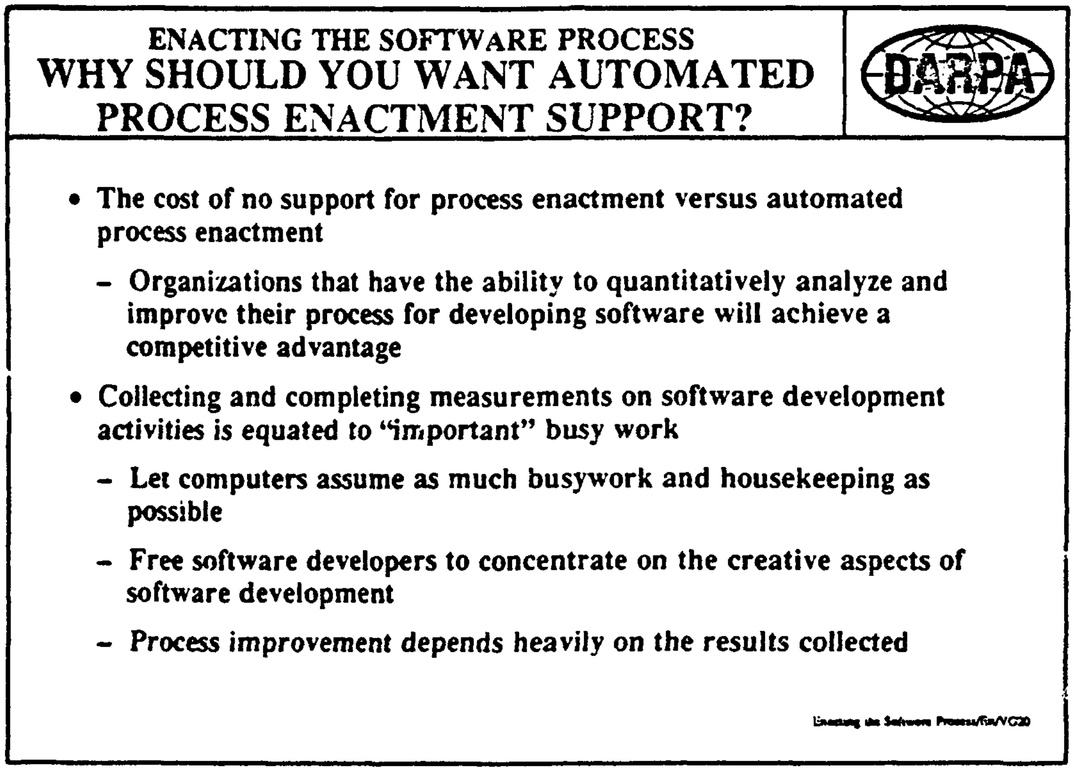
Active process enactment support (continued)

Potential benefits (process–driven SEE) (continued):

- The larger the project, the greater the need for automated process enactment support
  - Manual process enactment cannot scale up, automated process enactment support can

Enacting the Software Process/En/VG19

The larger the project, the greater the need for automated process enactment support. Problems realized from manual process enactment can scale up. Automated process enactment support is required to address those problems.

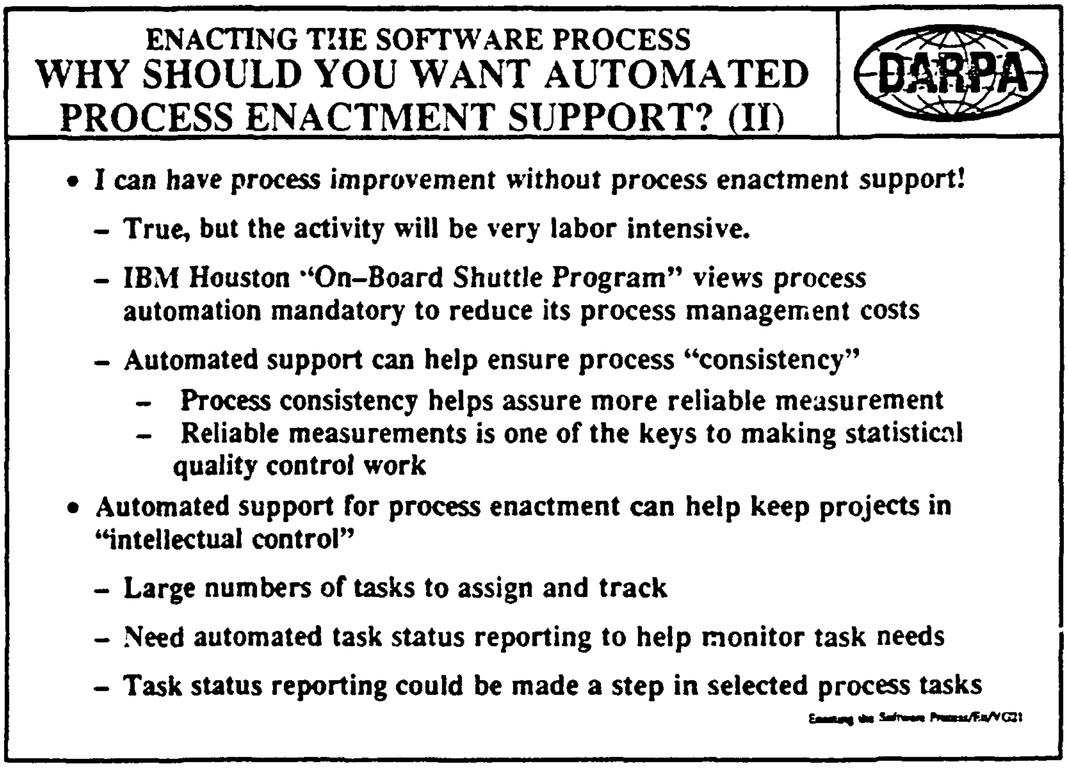


ENACTING THE SOFTWARE PROCESS

WHY SHOULD YOU WANT AUTOMATED PROCESS ENACTMENT SUPPORT?

- The cost of no support for process enactment versus automated process enactment
  - Organizations that have the ability to quantitatively analyze and improve their process for developing software will achieve a competitive advantage
- Collecting and completing measurements on software development activities is equated to "important" busy work
  - Let computers assume as much busywork and housekeeping as possible
  - Free software developers to concentrate on the creative aspects of software development
  - Process improvement depends heavily on the results collected

Enacting the Software Process/Fin/VG20

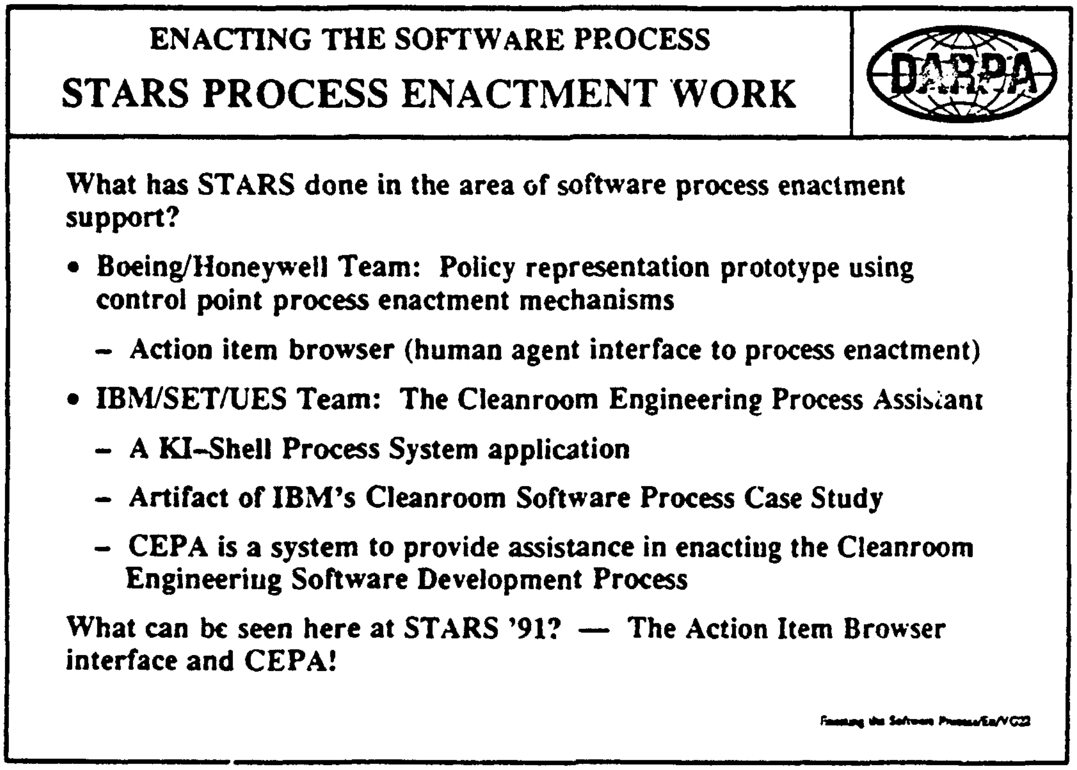


ENACTING THE SOFTWARE PROCESS

WHY SHOULD YOU WANT AUTOMATED PROCESS ENACTMENT SUPPORT? (II)

- I can have process improvement without process enactment support!
  - True, but the activity will be very labor intensive.
  - IBM Houston "On–Board Shuttle Program" views process automation mandatory to reduce its process management costs
  - Automated support can help ensure process "consistency"
    - Process consistency helps assure more reliable measurement
    - Reliable measurements is one of the keys to making statistical quality control work
- Automated support for process enactment can help keep projects in "intellectual control"
  - Large numbers of tasks to assign and track
  - Need automated task status reporting to help monitor task needs
  - Task status reporting could be made a step in selected process tasks

Enacting the Software Process/En/VG21



ENACTING THE SOFTWARE PROCESS

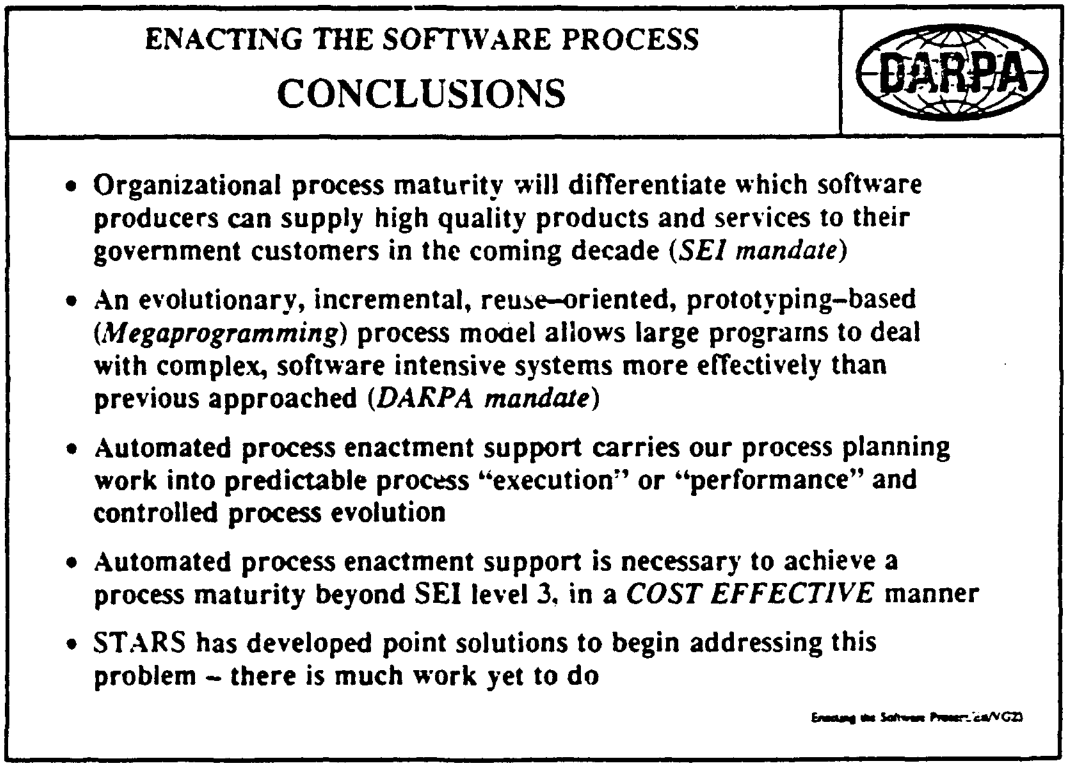
STARS PROCESS ENACTMENT WORK

What has STARS done in the area of software process enactment support?

- Boeing/Honeywell Team: Policy representation prototype using control point process enactment mechanisms
  - Action item browser (human agent interface to process enactment)
- IBM/SET/UES Team: The Cleanroom Engineering Process Assistant
  - A KI–Shell Process System application
  - Artifact of IBM's Cleanroom Software Process Case Study
  - CEPA is a system to provide assistance in enacting the Cleanroom Engineering Software Development Process

What can be seen here at STARS '91? — The Action Item Browser interface and CEPA!

Enacting the Software Process/Ea/VG22



ENACTING THE SOFTWARE PROCESS

CONCLUSIONS

- Organizational process maturity will differentiate which software producers can supply high quality products and services to their government customers in the coming decade (*SEI mandate*)
- An evolutionary, incremental, reuse–oriented, prototyping–based (*Megaprogramming*) process moael allows large programs to deal with complex, software intensive systems more effectively than previous approached (*DARPA mandate*)
- Automated process enactment support carries our process planning work into predictable process "execution" or "performance" and controlled process evolution
- Automated process enactment support is necessary to achieve a process maturity beyond SEI level 3, in a *COST EFFECTIVE* manner
- STARS has developed point solutions to begin addressing this problem – there is much work yet to do

Enacting the Software Process/ea/VG23

ENACTING THE SOFTWARE PROCESS

S/W MODULE SPEC PROCESS

PA1) ASSIGN SW_MODULE_SPECIFICATION TASK

PA2) PREPARE SW_MODULE_SPECIFICATION

PA3) CONDUCT TEAM REVIEW OF SW_MODULE_SPECIFICATION

PA4) REVIEW TEAM_REVIEW_RESULTS AND ACCEPT OR REJECT SW_MODULE_SPECIFICATION

Notes on reading the following process:

1) Arguments:

a) INCR = increment
b) SWENG = software engineer id number
c) WKI = work item, such as a SW_MODULE_SPECIFICATION
d) RTNO = count of individuals on the review team

ENACTING THE SOFTWARE PROCESS

ASSIGN SW_MODULE_SPECIFICATION TASK

ENTRY CRITERIA FOR PA1:

```
IF SOFTWARE_INCREMENT_STATE(INCR) = "RELEASED" AND
   TEAM_RESOURCES_STATE(INCR) = "ASSIGNED"
THEN DO PA1;
```

TASKS FOR PA1:

PAT11: PLAN SOFTWARE INCREMENT

```
AGENT: SOFTWARE_INCREMENT_MANAGER;
SUBTASKS:
1) DO PLAN_SOFTWARE_INCREMENT UNTIL COMPLETED;
2) INCREMENT_PLANNED_STATE(INCR) := "YES";
VC11:    IF    INCREMENT_PLANNED_STATE(INCR) = "YES"
         THEN DO PAT12;
```

PAT12: ASSIGN SOFTWARE MODULE_SPECIFICATION TASK TO AN AVAILABLE SOFTWARE ENGINEER

```
AGENT: SOFTWARE_INCREMENT_MANAGER;
SUBTASKS:
1) DO IDENTIFY_AVAILABLE_SOFTWARE_ENGINEER UNTIL COMPLETED;
2) DO ASSIGN_SOFTWARE_ENGINEER_TO_TASK UNTIL COMPLETED;
3) SW_ENG_INCREMENT_RESOURCE_STATE(INCR,SWENG,WKI) := "ASSIGNED_WKI"
VC12:    IF SW_ENG_INCREMENT_RESOURCE_STATE(INCR,SWENG,WKI) =
            "ASSIGNED_WKI"
         THEN DO PAT13;
/* SOFTWARE ENGINEER CAN ACCEPT WORKLOAD AND HAS BEEN ASSIGNED A
SOFTWARE_MODULE_SPECIFICATION TO WORK ON */
```

ENACTING THE SOFTWARE PROCESS

ASSIGN SW_MODULE_SPECIFICATION TASK (CONT)

PAT13: SCHEDULE WORK ASSIGNMENT REVIEW

```
AGENT:  SOFTWARE_INCREMENT_MANAGER;
SUBTASKS:
1) DO SCHEDULE_WORK_ASSIGNMENT_REVIEW UNTIL COMPLETED;
2) REVIEW_WORK_ASSIGNMENT_STATE(INCR,WKI) := "SCHEDULED"
VC13:   IF REVIEW_WORK_ASSIGNMENT_STATE(INCR,WKI) = "SCHEDULED"
        THEN DO PAT14;
```

PAT14: REVIEW WORK ASSIGNMENT WITH SW-ENGINEER

```
AGENT:  SOFTWARE_INCREMENT_MANAGER, SOFTWARE_ENGINEER;
SUBTASKS:
1) DO REVIEW_WORK_ASSIGNMENT_WITH_SW_ENGINEER UNTIL COMPLETED;
2) REVIEW_WORK_ASSIGNMENT_STATE := "COMPLETED";
VC14:   IF REVIEW_WORK_ASSIGNMENT_STATE(INCR,WKI) = "COMPLETED"
        THEN DO PAT15;
```

PAT15: RECORD WORK ASSIGNMENT IN PROJECT MANAGEMENT TOOL

```
AGENT:  SOFTWARE_INCREMENT_MANAGER;
SUBTASKS:
1) DO RECORD_WORK_ASSIGNMENT UNTIL COMPLETED;
2) RECORD_WORK_ASSIGNMENT(INCR,WKI) := "TRUE";
VC15:   IF RECORD_WORK_ASSIGNMENT(INCR,WKI) = "TRUE"
        THEN SOFTWARE_MODULE_SPEC_STATE(INCR,WKI) := "WIP"
        AND PASS_PA1(INCR,WKI) := "TRUE";
        /* VERIFY SOFTWARE MANAGER HAS RECORDED THE WORK */
```

EXIT CRITERIA: PA1

```
IF  PASS_PA1(INCR,WKI) = "TRUE"
THEN DO PA2;
```

ENACTING THE SOFTWARE PROCESS

PA2: PREPARE S/W MODULE SPECIFICATION TASK

ENTRY CRITERIA FOR PA2:

```
IF      PASS_PA1(INCR,WKI) = "TRUE"
THEN PASS_PA1 := "DONE",
        DO PAT21;
```

TASKS FOR PA2:

PAT21: DEVELOP SOFTWARE MODULE SPECIFICATION (USING BLACK BOX TECHNIQUES

```
AGENT:  SOFTWARE_ENGINEER;
SUBTASKS:
1) DO BLACK BOX STEPS;
2) DO SOFTWARE_MODULE_SPEC_SELF_VALIDATION UNTIL COMPLETED;
3) SOFTWARE_MODULE_SPEC_STATE(INCR,WKI) := "SELF_VALIDATED";
4) DO PREPARE_SW_MODULE_SPEC_METRIC_FORM(INCR,WKI) UNTIL COMPLETED;
5) SW_MODULE_SPEC_METRIC_FORM_STATE(INCR,WKI) := "COMPLETED";

VC21:
IF      SOFTWARE_MODULE_SPEC_STATE(INCR,WKI) = "SELF_VALIDATED"
AND     SW_MODULE_SPEC_METRIC_FORM_STATE(INCR,WKI) = "COMPLETED"
THEN DO PAT22;
/* SOFTWARE ENGINEER VALIDATES THAT ALL STIMULI AND RESPONSES IDENTIFIED HAVE
BEEN ACCOUNTED FOR */
```

PAT22: REQUEST TEAM REVIEW OF SOFTWARE_MODULE_SPECIFICATION

```
AGENT:  SOFTWARE_ENGINEER;
SUBTASKS:
1) DO REQUEST_TEAM_REVIEW_OF_SW_MODULE_SPEC UNTIL COMPLETED;
2) SW_MODULE_TEAM_REVIEW_STATE(INCR,WKI) := "REQUESTED";

VC22:   IF SW_MODULE_TEAM_REVIEW_STATE(INCR,WKI) = "REQUESTED"
        THEN DO PAT23;
```

ENACTING THE SOFTWARE PROCESS

PA2: PREPARE S/W MODULE SPECIFICATION TASK (CONTINUED)

PAT23: SCHEDULE TEAM REVIEW

```
AGENT: SOFTWARE_INCREMENT_MANAGER;
SUBTASKS:

1) DO TEAM_CALENDAR_COORDINATION;
2) DO SCHEDULE_SW_MODULE_SPECIFICATION UNTIL COMPLETED;
3) SOFTWARE_MODULE_TEAM_REVIEW_STATE(INCR,WKI) := "SCHEDULED";

VC23:

IF SOFTWARE_MODULE_TEAM_REVIEW_STATE(INCR,WKI) = "SCHEDULED"
THEN DO PAT24;
/* SOFTWARE ENGINEERING MANAGER SCHEDULES TEAM REVIEW */
```

PAT24: FREEZE SOFTWARE_MODULE_SPECIFICATION

```
AGENT: SOFTWARE_INCREMENT_MANAGER;
SUBTASKS:

1) DO PREPARE_CONFIGURATION_MANAGEMENT_ENTRY(INCR,WKI) UNTIL COMPLETED;
2) SOFTWARE_MODULE_SPEC_STATE(INCR,WKI) := "IN_REVIEW";

VC24:

IF     SOFTWARE_MODULE_SPEC_STATE(INCR,WKI) = "IN_REVIEW"
THEN PASS_P2(INCR,WKI) := "TRUE";
```

EXIT CRITERIA: PA2

```
IF     PASS_PA2(INCR,WKI) = "TRUE"
THEN DO PA3;
```

ENACTING THE SOFTWARE PROCESS

PA3: CONDUCT TEAM REVIEW OF THE SOFTWARE_MODULE_SPECIFICATION

ENTRY CRITERIA FOR PA3:

```
IF     SOFTWARE_MODULE_TEAM_REVIEW_STATE(INCR,WKI)  =  "SCHEDULED"
AND    START_START_REVIEW_DATE(INCR,WKI) = CURRENT_DATE
THEN   PASS_PA2 := "DONE",
       DO PAT31;
```

TASKS FOR PA3:

PAT31: PRESENT SOFTWARE MODULE SPECIFICATION AND COMPLIANCE TO VALIDATION CRITERIA

```
AGENT:   SOFTWARE_ENGINEER, TEAM_MEMBERS(RTNO), TEAM_MODERATOR;
SUBTASKS:

1)  DO TEAM_REVIEW_PRESENTATION UNTIL COMPLETED;
2)  TEAM_REVIEW_PRESENTATION_STATUS(INCR,WKI) := "FINISHED";
3)  DO COMPLETE_TEAM_REVIEW_METRIC_FORM UNTIL COMPLETED;
4)  TEAM_REVIEW_PRESENTATION_STATUS(INCR,WKI) := "FINISHED";
/* REVIEW 1) TEAM MODERATOR RECORDS, 2) ALL SPECIFICATION PROBLEMS AND
TYPES ENCOUNTERED DURING THE REVIEW AND 3) TIMES TAKEN FOR THE REVIEW
*/
5)  DO HOLD_ACCEPT_OR_REJECT_DISCUSSIONS UNTIL COMPLETED;
6)  TEAM_REVIEW_SW_MODULE_SPEC_ACCEPTANCE(INCR,WKI):= "PASS" | "FAIL",
7)  SW_MODULE_TEAM_REVIEW_STATE(INCR,WKI) := "COMPLETED";
8)  DO COMPLETE_TEAM_REVIEW_MODERATOR_SCRIPT UNTIL COMPLETED;
9)  TEAM_REVIEW_MODERATOR_SCRIPT_STATUS := "COMPLETED";
```

VC31:

```
IF     TEAM_REVIEW_PRESENTATION_STATUS(INCR,WKI) = "FINISHED"
AND    TEAM_REVIEW_METRIC_FORM_STATUS(INCR,WKI) = "COMPLETED"
AND    TEAM_REVIEW_SW_MODULE_SPEC_ACCEPTANCE(INCR,WKI) =
           "PASS" | "FAIL",
AND    TEAM_REVIEW_MODERATOR_SCRIPT_STATUS = "COMPLETED"
THEN DO PAT32;
```

ENACTING THE SOFTWARE PROCESS

PA3: CONDUCT TEAM REVIEW OF THE SOFTWARE_MODULE_SPECIFICATION (CONTINUED)

PAT32: COMPLETE CHECKLIST FOR SOFTWARE_MODULE_ SPECIFICATION REVIEW

```
AGENT:  TEAM_MEMBERS (INCLUDING TEAM_MODERATOR AND PRESENTOR);
SUBTASKS:

1) DO COMPLETE_CHECKLIST_FOR_SW_MODULE_SPEC(INCR,WKI,RTNO)
      UNTIL COMPLETED;
2) TEAM_REVIEW_SW_MODULE_SPEC_CHECKLIST(INCR,WKI,RTNO) :=
      "COMPLETED"
2) DO SEND_FORM UNTIL COMPLETED;
3) REVIEW_FORM_SENT(INCR,WKI,RTNO) := "TRUE";

VC32:

IF      TEAM_REVIEW_SW_MODULE_SPEC_CHECKLIST(INCR,WKI) = "COMPLETED"
AND     REVIEW_FORM_SENT(INCR,WKI) = "TRUE"
THEN    PASS_PA3(INCR,WKI) := "TRUE";
/* TEAM REVIEW CHECKLIST FOR SOFTWARE MODULE SPECIFICATIONS HAVE
BEEN COMPLETED AND SENT TO THE MEETING REQUESTOR */
```

EXIT CRITERIA: PA3

```
IF      PASS_PA3(INCR,WKI) = "TRUE"
THEN    DO PA4;
```

ENACTING THE SOFTWARE PROCESS

PA4: REVIEW TEAM_REVIEW_RESULTS AND ACCEPT OR REJECT SW_MODULE_SPECIFICATION

ENTRY CRITERIA FOR PA4:

```
IF      PASS_PA3(INCR,WKI) = "TRUE"
THEN PASS_PA(INCR,WKI) := "DONE"
        DO PA41;
```

TASKS FOR PA4:

PAT41: REVIEW SOFTWARE MODULE SPECIFICATION TEAM REVIEW RESULTS AND ACCEPT OR REJECT

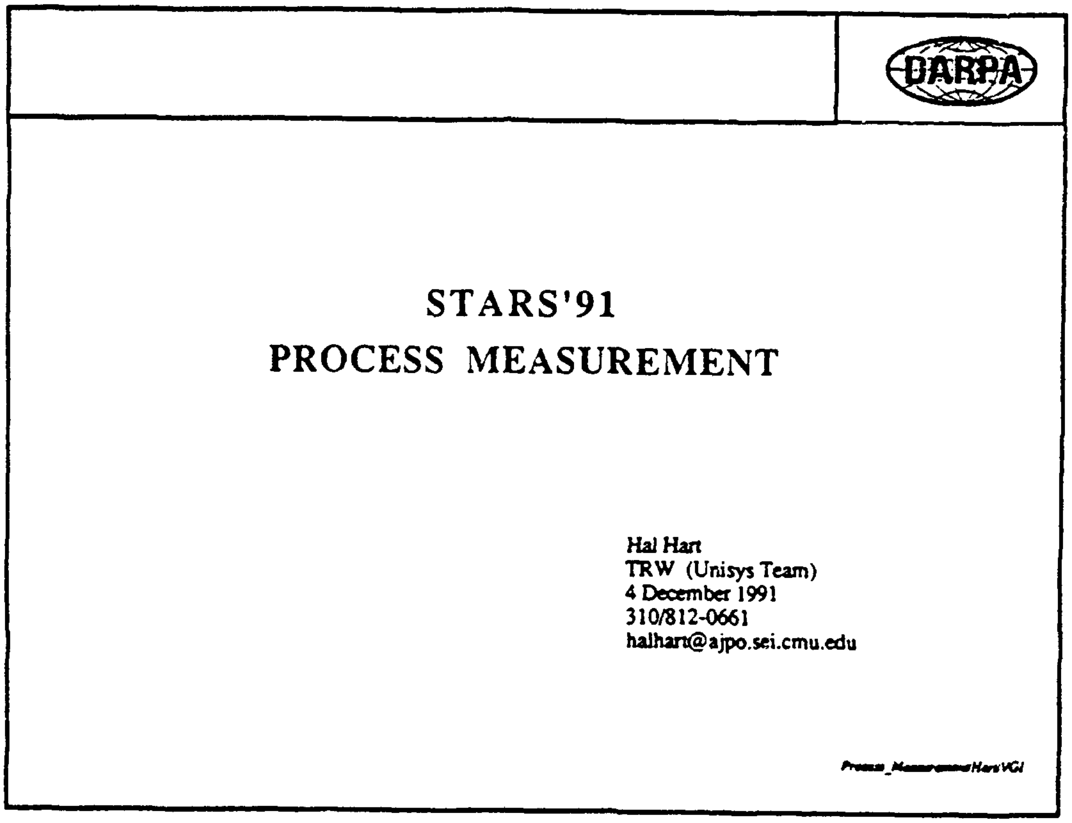
```
AGENT:   SOFTWARE_INCREMENT_MANAGER;
SUBTASKS:

1)  DO REVIEW (    TEAM_REVIEW_MODERATOR_SCRIPT(INCR,WKI),
                   SW_MODULE_SPECIFICATION(INCR,WKI),
                   (CHECKLIST_FOR_SW_MODULE_SPEC(INCR,WKI,RTNO)
                      FOR RTNO = 1 THRU REVIEW_TEAM_COUNT)        )
    UNTIL COMPLETED;
2)  IF      TEAM_REVIEW_SW_MODULE_SPECIFICATION = "PASS"
    AND     IS FOUND ACCEPTABLE BY THE SOFTWARE INCREMENT MANAGER
    THEN    SW_MODULE_SPECIFICATION_STATE(INCR,WKI) := "COMPLETED"
    ELSE    SW_MODULE_SPECIFICATION_STATE(INCR,WKI) := "WIP";

VC41: IF      SW_MODULE_SPECIFICATION_STATE = "COMPLETED"
      THEN PASS_PA4(INCR,WKI) := "TRUE"
      ELSE PASS_PA4(INCR,WKI) := "FALSE";
```

EXIT CRITERIA: PA4

```
IF      PASS_PA4(INCR,WKI) = "TRUE"
THEN EXIT
ELSE  DO PA2;
```

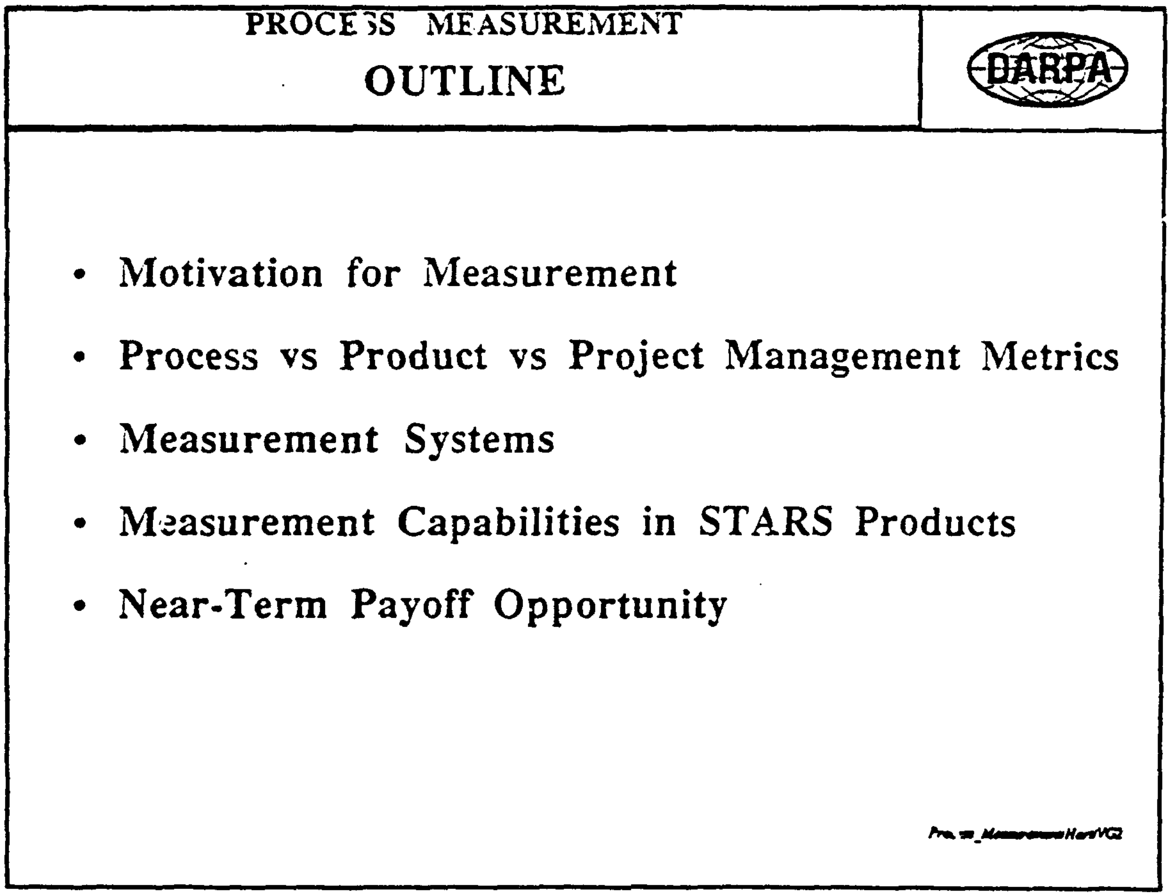


DARPA

STARS'91
PROCESS MEASUREMENT

Hal Hart
TRW (Unisys Team)
4 December 1991
310/812-0661
halhart@ajpo.sei.cmu.edu

Process_MeasurementHart.VGI



PROCESS MEASUREMENT

OUTLINE

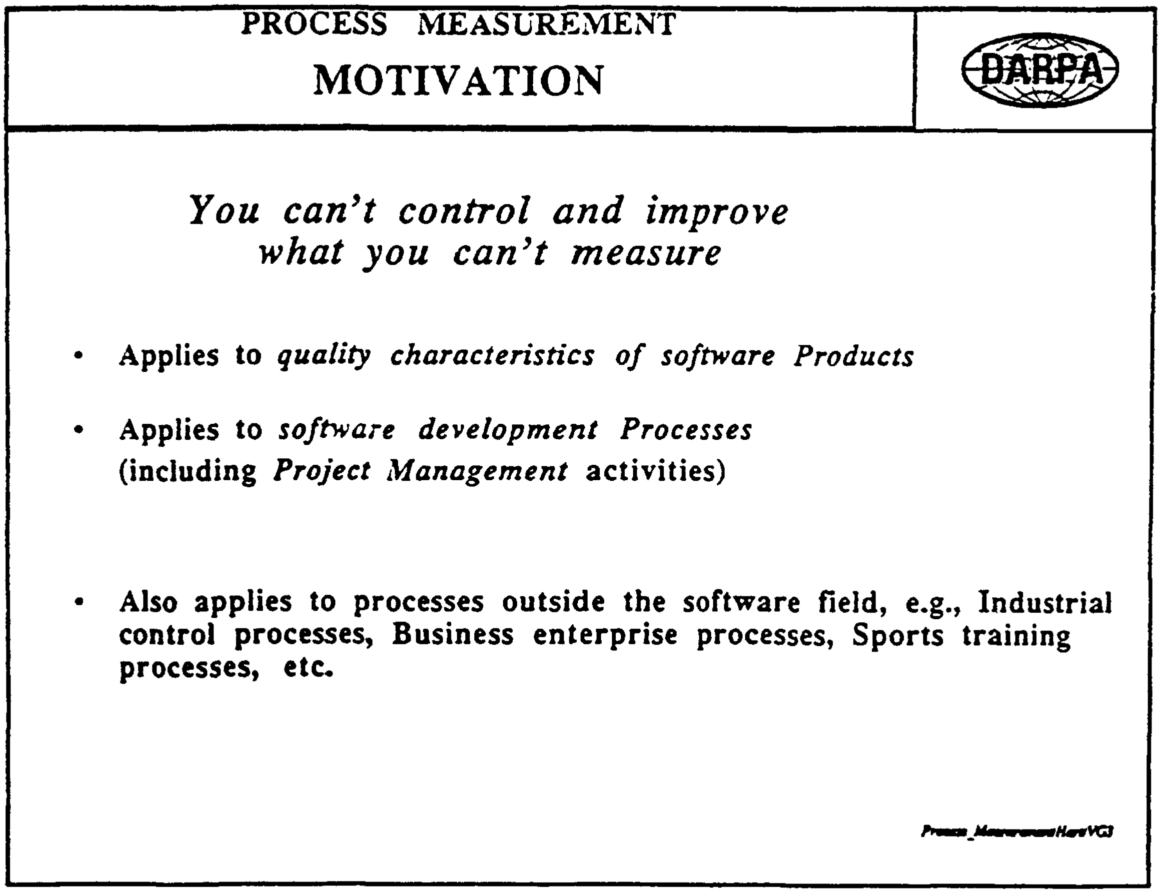
DARPA

- Motivation for Measurement
- Process vs Product vs Project Management Metrics
- Measurement Systems
- Measurement Capabilities in STARS Products
- Near-Term Payoff Opportunity

Process_Measurement/Hart/VG2

OUTLINE

I'm going to present to you the "WHY," the "WHICH," the "WHAT," the "WHERE," and the "WHEN" of the Measurement facet of process-driven development.



PROCESS MEASUREMENT

MOTIVATION

DARPA

You can't control and improve what you can't measure

- Applies to *quality characteristics of software Products*
- Applies to *software development Processes* (including *Project Management* activities)
- Also applies to processes outside the software field, e.g., Industrial control processes, Business enterprise processes, Sports training processes, etc.

Process_MeasurementHartVG3

MOTIVATION

Measurement —

How can we tell that we have a good process? — We have to measure it!

How can we tell what to change if our process isn't good enough? — We have to be able to measure it!

How can we tell if the change was a step forwards or a step backwards? — We have to be able to measure it.

How can we tell if the change had enough benefit to offset the cost of instituting the change? — We have to be able to measure it!

You can't control and improve what you can't measure!

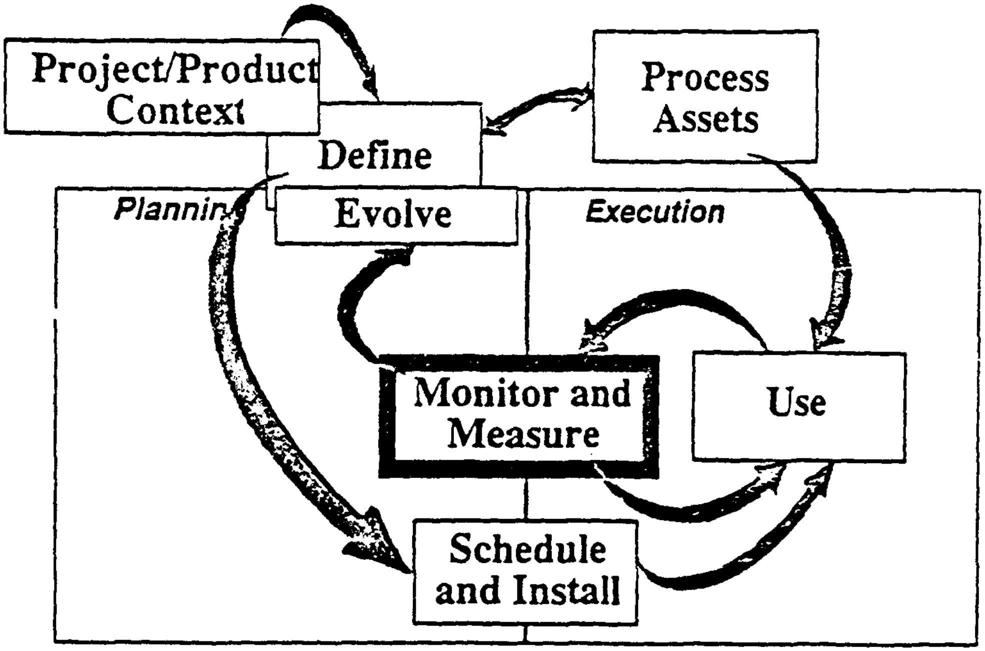
It has been fairly well accepted that measurements and metrics are useful, if not crucial, to judging the QUALITY of software products developed by teams.

The same applies to software development processes, and those aspects of development processes that interact with Project Management activities, which are subset of a project's total activities.

Of course, most of what we're talking about here in the measurement area, and most of all our efforts in the whole Process area, apply outside the software development domain as well — industrial control processes, business enterprises, etc.

PROCESS MEASUREMENT

PROCESS DRIVEN DEVELOPMENT



Adapted from Software Productivity Consortium work.

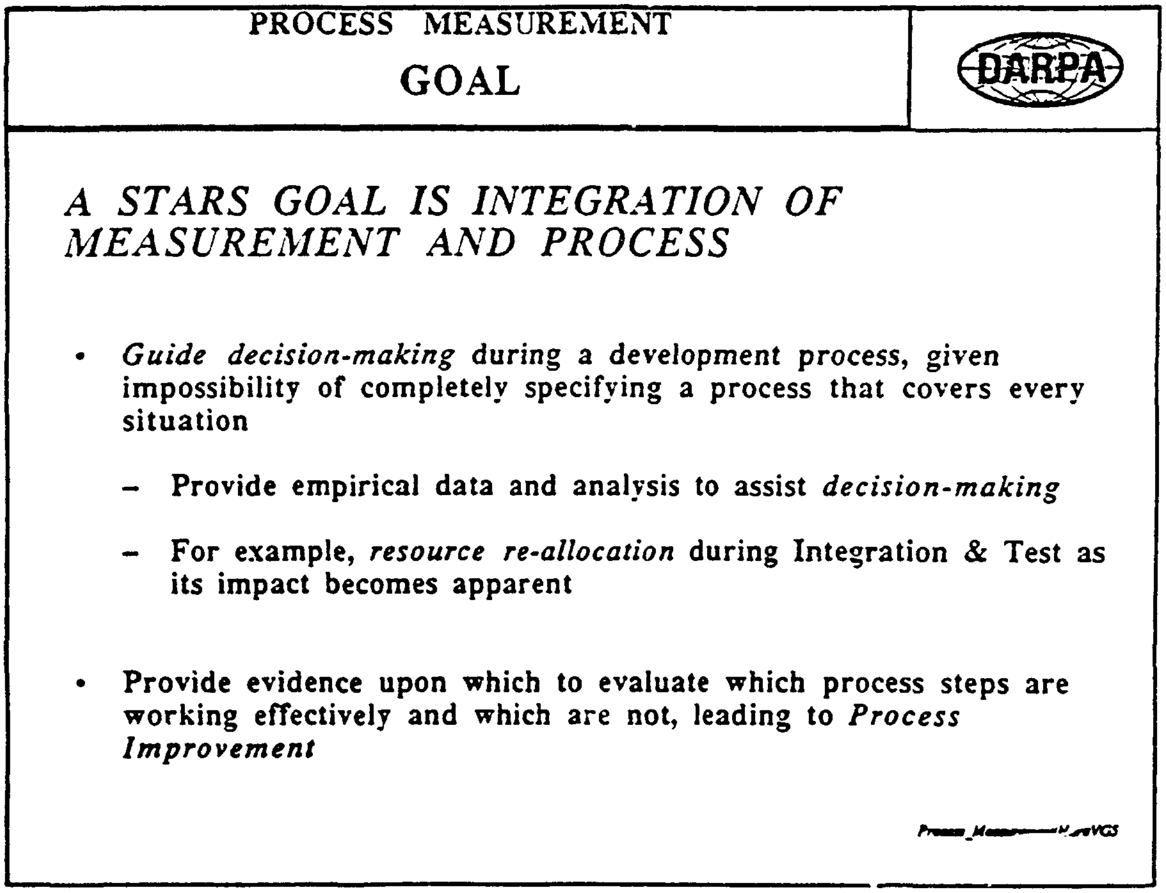
Process_Measurement/Hart/VGA

RELATIONSHIP TO OTHER STARS PROCESS ACTIVITIES

Measurement plays a very central role in the vision STARS has for Process.

"Monitor and Measure" is right in the middle, in the middle of a large loop, of our Process-Driven development operational concept. It effects just what I said: Being able to evolve and improve the Process Definitions and improve the Processes in Action in the future.

There's another loop shown here though, where measurement can actually, *within* (during) a project, be used to improve the activities of a project. This is DECISION MAKING, Measurement-assisted decision making, empirical data measurements assisting human decisions during a project.



PROCESS MEASUREMENT

GOAL

DARPA

A STARS GOAL IS INTEGRATION OF MEASUREMENT AND PROCESS

- *Guide decision-making* during a development process, given impossibility of completely specifying a process that covers every situation
  - Provide empirical data and analysis to assist *decision-making*
  - For example, *resource re-allocation* during Integration & Test as its impact becomes apparent
- Provide evidence upon which to evaluate which process steps are working effectively and which are not, leading to *Process Improvement*

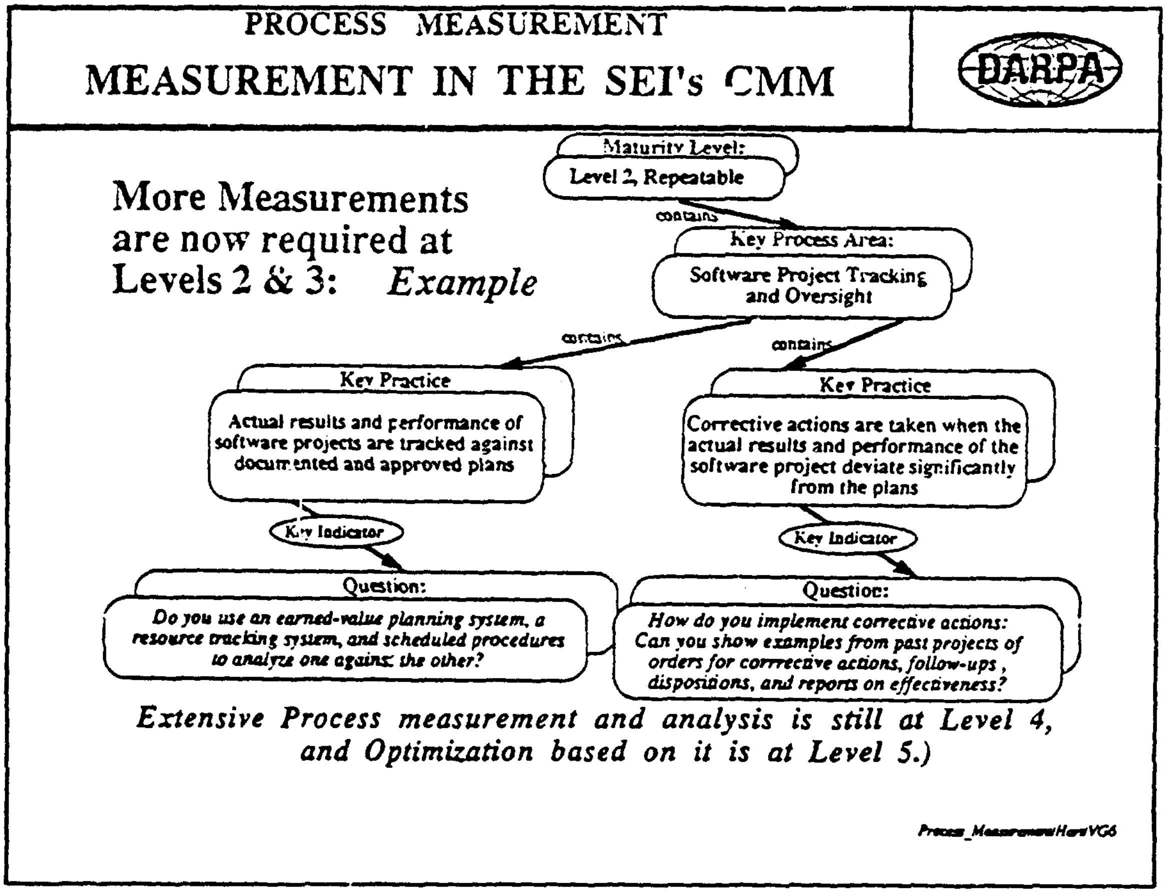
Process_Measurement/HartVGS

A STARS GOAL IS INTEGRATION OF MEASUREMENT AND PROCESS

So, I repeat, one of STARS's two major goals in the Process Measurement area is to assist or guide Decision Making, collecting and presenting for use empirical evidence data that makes project decision making on firmer ground than it was without quantitative data.

For example, being able to re-allocate resources during Integration and Test (I&T) as certain modules are found to not integrate, and perhaps causing ripple effects if there are a select few modules found to be troublesome in their interfaces with many other modules,. This has an impact on the process and particularly on Project Management activities in their allocation of manpower

But then, it's overall Process Improvement, institutionally, over the long run, spanning all of an organization's projects over the years, that is the other dimension where Measurement is critical, and is the other major STARS goal in the Process Measurement area. This was the big loop I showed first on the previous chart.



PROCESS MEASUREMENT

MEASUREMENT IN THE SEI's CMM

DARPA

More Measurements are now required at Levels 2 & 3: *Example*

Maturity Level:
Level 2, Repeatable

contains

Key Process Area:
Software Project Tracking and Oversight

contains

Key Practice
Actual results and performance of software projects are tracked against documented and approved plans

Key Indicator

Question:
Do you use an earned-value planning system, a resource tracking system, and scheduled procedures to analyze one against the other?

contains

Key Practice
Corrective actions are taken when the actual results and performance of the software project deviate significantly from the plans

Key Indicator

Question:
How do you implement corrective actions: Can you show examples from past projects of orders for corrrective actions, follow-ups , dispositions, and reports on effectiveness?

Extensive Process measurement and analysis is still at Level 4, and Optimization based on it is at Level 5.)

Process_MeasurementHart/VG6

MEASUREMENT IS MORE IMPORTANT IN THE NEW SEI CMM: An Example

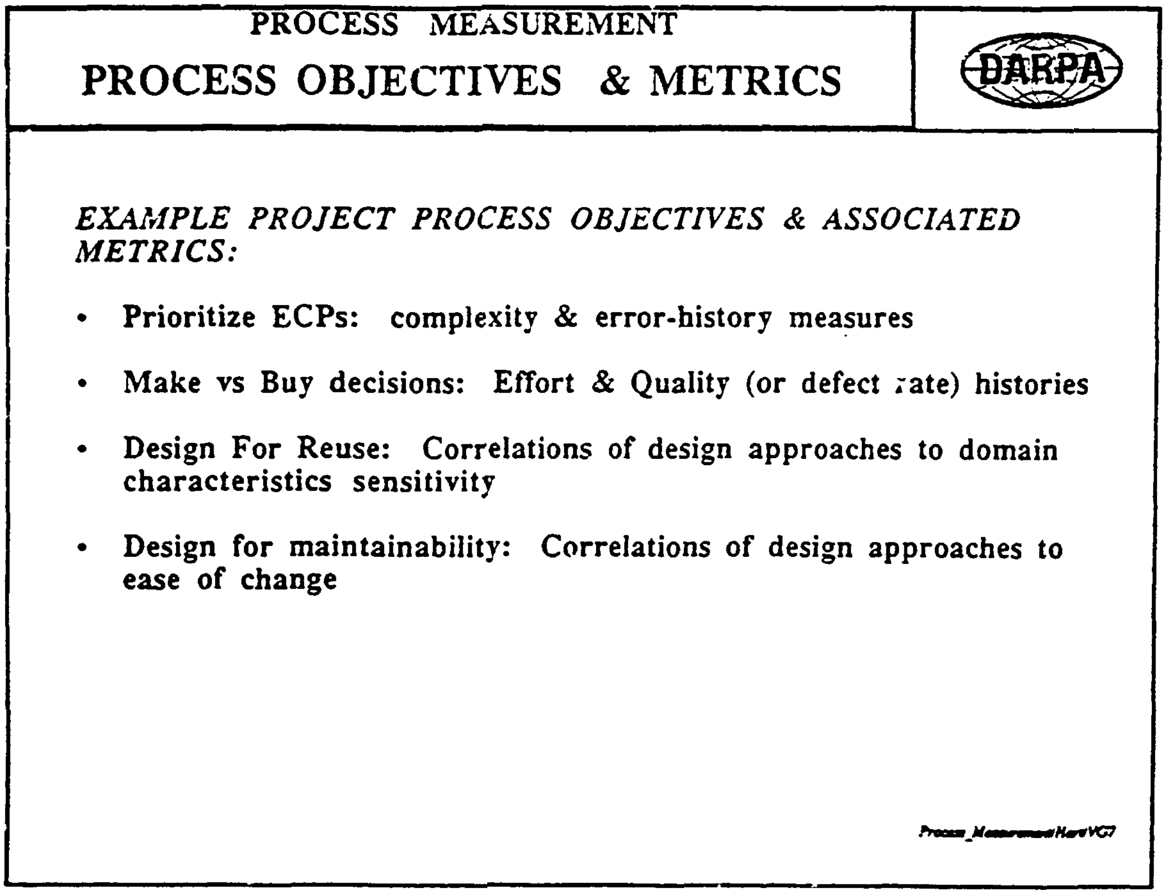
Here's another reason why I think everybody in the room ought to be concerned about Measurement: Measurement plays a prominent role in the SEI's process Capability Maturity Model (CMM) . In fact, in the 1991 version of the CMM, Measurement is more important at Level 2 than it was before. Using the style of the Key Practices assessment model there, here's an example from Level 2:

"Software Project Tracking and Oversight" is a key area at Level 2 now. Two key practices establishing that Level-2 capability are (1) tracking actual project results against approved plans, and (2) taking corrective actions when actual project results deviate significantly from plans.

Key indicators of these key practices are the following two questions which might be asked of an organization undergoing a capability assessment or a contractor evaluation: (1) "Do you use an earned-value planning and reporting system, a resource tracking system, and regularly scheduled procedures to compare one to the other?" and (2) "How do you implement corrective actions? Can you produce documented examples from past projects of orders for corrective actions, an follow-up reports, reports of dispositions, and reports assessing effectiveness of such corrective actions?"

Overall, there are more measurements are now called out in the Key Practices at Levels 2 and 3 than there were before.

Extensive incorporation of process measurement and analysis for identifying improvement opportunities is still the essence of Level 4, and incorporating Process Improvements based on that is still the essence of Level 5.



PROCESS MEASUREMENT

PROCESS OBJECTIVES & METRICS

DARPA

EXAMPLE PROJECT PROCESS OBJECTIVES & ASSOCIATED METRICS:

- Prioritize ECPs: complexity & error-history measures
- Make vs Buy decisions: Effort & Quality (or defect ;ate) histories
- Design For Reuse: Correlations of design approaches to domain characteristics sensitivity
- Design for maintainability: Correlations of design approaches to ease of change

Process_MeasurementHartVG7

PROCESS OBJECTIVES and METRICS

I will introduce and distinguish 3 different kinds of metrics on the next chart: Product, Process, and Project Management metrics.

Some quick examples of Process objectives and the associated metrics:

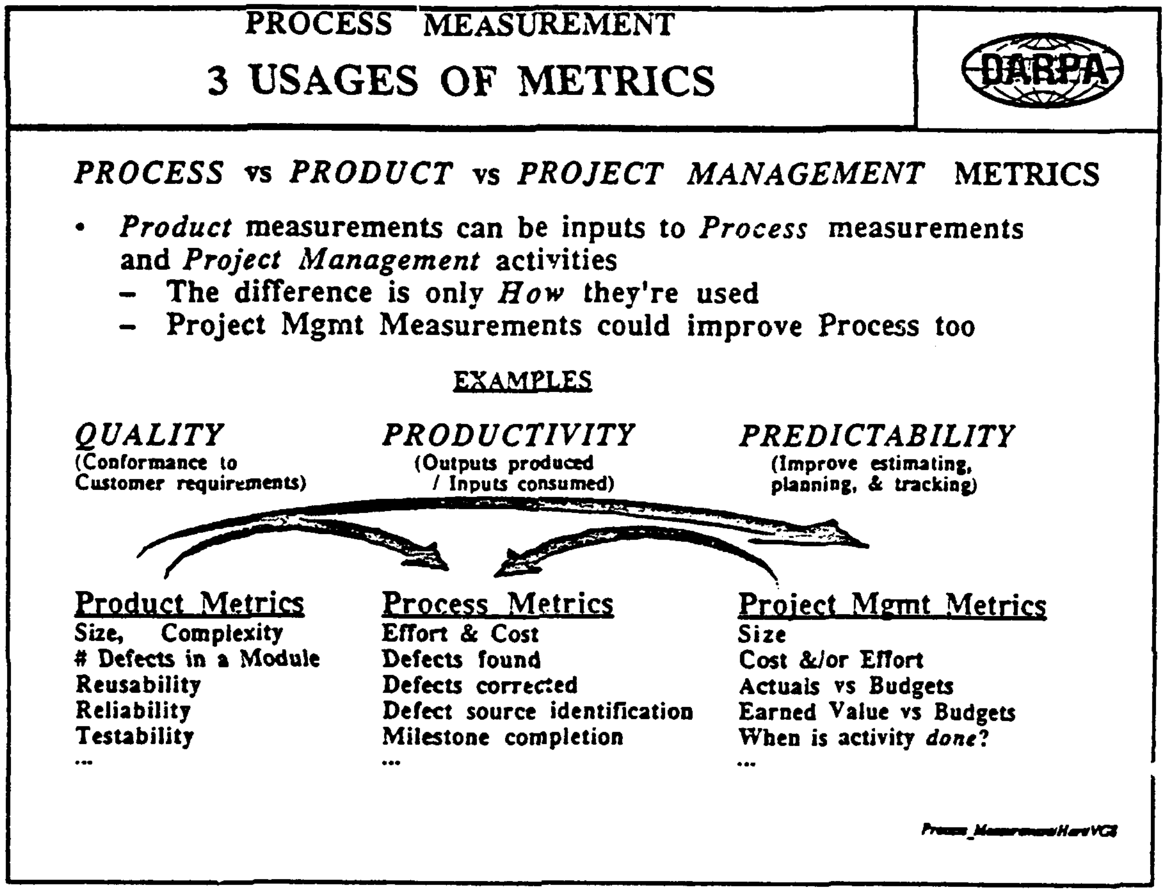
Prioritizing Engineering Change Proposals — how do you tell which ones are the most important to work first, or which ones require the most staffing?: Complexity and error-history measures, error-proneness and past histories of trouble with particular modules are very useful measures.

Make vs Buy decisions: Does the Effort offset the gain in Quality relative to buy ... Effort and Quality (or defect rate) histories give helpful indications of how to make this decision.

Designing For Reuse: Reuse is an increasingly important project objective; do you have some data that indicates for your application domain, one design approach will increase the reusability index relative to others? Correlations of design approaches to domain characteristics sensitivity are results of measurements.

Same for Maintainability, and the answer may or may not always be the same as the answer for Reuse. Correlations of design approaches to ease of change are key to examine here, and a history of measurements of change activity against various design approaches in the subject domain would help.

These questions can be answered if you have an empirical, quantitative history database.



PROCESS MEASUREMENT

3 USAGES OF METRICS

DARPA

PROCESS vs *PRODUCT* vs *PROJECT MANAGEMENT* METRICS

- *Product* measurements can be inputs to *Process* measurements and *Project Management* activities
  - The difference is only *How* they're used
  - Project Mgmt Measurements could improve Process too

EXAMPLES

QUALITY
(Conformance to Customer requirements)

PRODUCTIVITY
(Outputs produced / Inputs consumed)

PREDICTABILITY
(Improve estimating, planning, & tracking)

Product Metrics
Size, Complexity
# Defects in a Module
Reusability
Reliability
Testability
...

Process Metrics
Effort & Cost
Defects found
Defects corrected
Defect source identification
Milestone completion
...

Project Mgmt Metrics
Size
Cost &/or Effort
Actuals vs Budgets
Earned Value vs Budgets
When is activity *done*?
...

Process_Measurement/HartVG8

PROCESS vs PRODUCT vs PROJECT MANAGEMENT METRICS

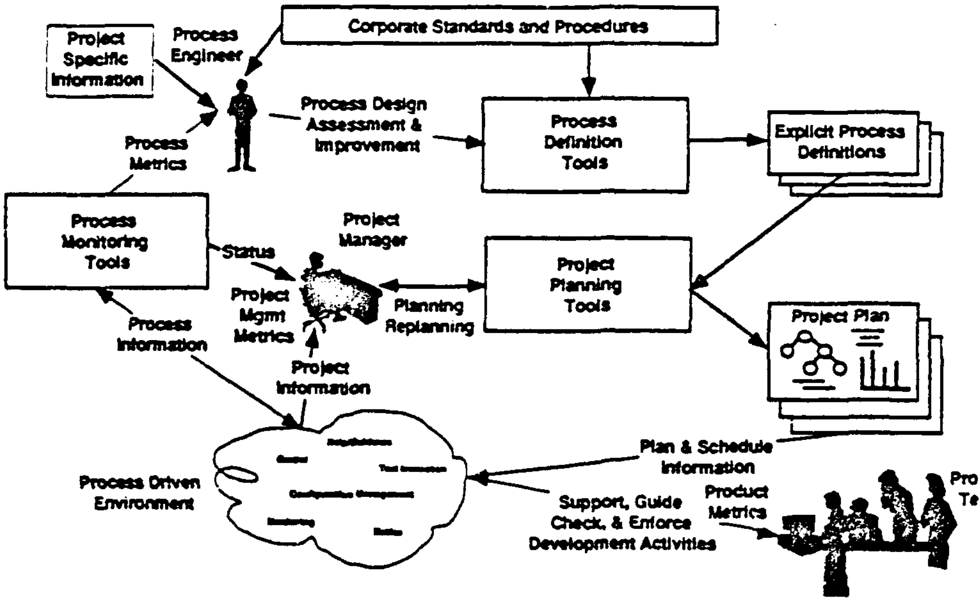
Now, looking at all 3 kinds of metrics together (Product, Project Management, and Process metrics), *Product* measurements at the lowest level are what is usually collected in practice today, sometimes manually, sometimes by tools. How you establish relationships or analyze particular product metrics may make them also inputs to Process Measurements, or they may make them into Project Management measurements. As I asserted before, Project Management activities are just a subset of total project process, so project measurement may be rightly regarded as a subset of process measurements.

The chart lists some examples of each of the three kinds of metrics No surprises here for Product metrics. Some of them may be hard to measure other than based on comparisons with empirical feedback from previously deployed modules. Most *Product* Metrics relate to QUALITY objectives. Software module Complexity has long been regarded as a key indicator or immediate and future problems; many of you are familiar with some of the varying Complexity metrics sets, such as (McCabe's) Cyclomatic Complexity numbers, (Halstead's) Software Sciences metrics, function points, path analysis, fan-in/fan-out counting, etc. Now, with *Process* metrics, PRODUCTIVITY is the encompassing goal. Clearly, Productivity includes Quality as a dimension, for example, Quality measurements directly affect the numerator in a quotient of *Product Value* (divided by effort and other resources consumed) which calculates Productivity. In other words, QUALITY is an essential factor in Process measurement! Project management metrics and activities, generally deal with PREDICTABILITY — estimating, planning, and tracking of progress against that plan, which may be same as the Process Plan.

You see that some of these metrics,for example, size, appear in more than one of these columns. What this really illustrates is that most of the Product metrics are among the basic inputs combined to establish whether estimates are good or bad and to improve estimating, to develop plans and to assist replanning. For example, under Project Management, Tracking is a relationship between various of these Product metrics and schedule or calendar.

PROCESS MEASUREMENT

METRICS IN PROCESS MGMT PARADIGM

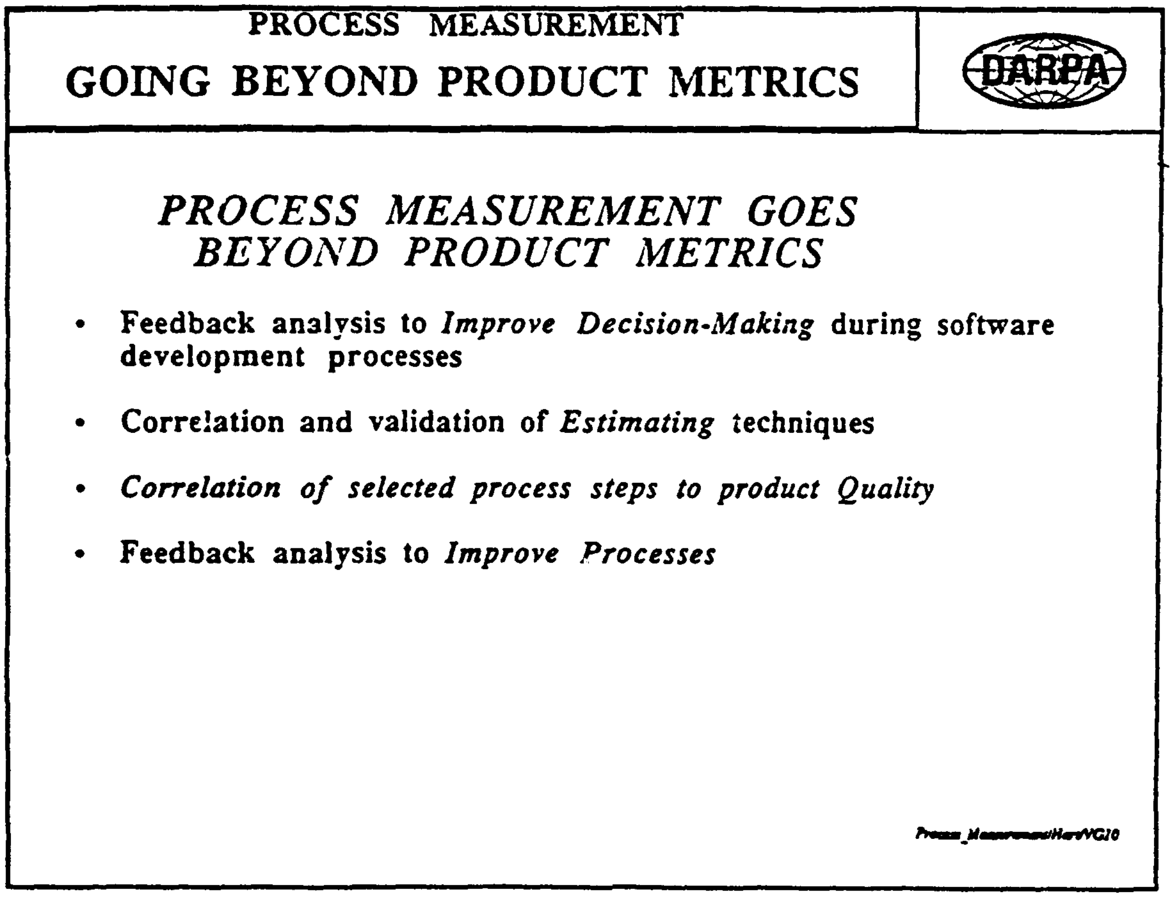


Process_Measurement/Hart/VG9

METRICS IN THE PROCESS MANAGEMENT PARADIGM

This diagram you saw in Jim King's presentation and it puts the roles of these 3 kinds of process metrics, their similarities and differences, in the total context of the Process Management paradigm of definition through instantiation on particular projects through the carrying out of the process for a particular project in a software engineering environment — the measurement role and the feedback to be able to improve the definitions, either of the projects or institutionally. Added on here are, so you can see the relationships between them are process metrics, project metrics, and project management metrics, which may be collected by monitoring tools separately from the Software Engineering tools, or they may arise from part of the information collected by Software Engineering tools. Some of those are analyzed in ways that make their usage project management information, and some of them are product metrics used by the project team to gauge their convergence on customer satisfaction (quality) goals.

The metrics are used for different purposes, but there's a lot of similarity between them. And, there's a lot of similarly between where they're detected and stored in the SEE. As I said before, product metrics in most cases are also inputs for calculating Project Management and Process metrics.



PROCESS MEASUREMENT

GOING BEYOND PRODUCT METRICS

DARPA

PROCESS MEASUREMENT GOES BEYOND PRODUCT METRICS

- Feedback analysis to *Improve Decision-Making* during software development processes
- Correlation and validation of *Estimating* techniques
- *Correlation of selected process steps to product Quality*
- Feedback analysis to *Improve Processes*

Process_Measurement/HartVG10

GOING BEYOND PRODUCT METRICS

So, let's recap: How does Process Measurement go beyond the in-practice notion of Product metrics?

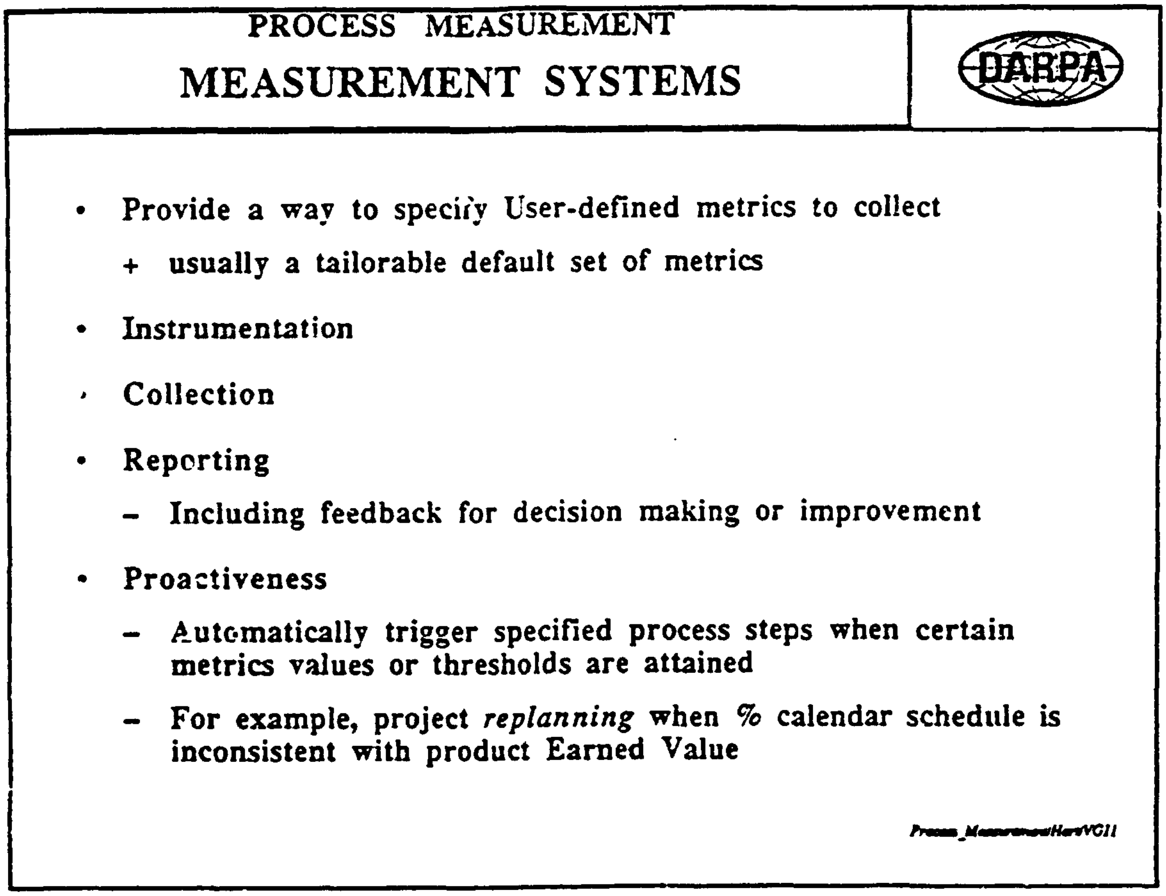
It's the PROACTIVE usage of measurement that distinguishes Process metrics activities from Product metrics. Collecting the metrics might be clearly a product measurement, but *project actions that are influenced or changed due to analysis of metrics are Process actions.*

For example, feedback analysis to Improve Decision-Making during software development processes.

Or, refinement of Estimating techniques by comparing past estimating procedure outputs to subsequent actuals, thereby either validating the estimating procedure or suggesting improvements based on the empirical data.

Or, determination of which process steps correlate most directly to product Quality, in the sense that increasing effort in particular process steps leads most directly to Quality gains, thereby improving an organization's process(es) for future projects.

This is essence of why we deal with Process Measurement. If you don't have the vision of improving and institutionalizing processes, you may collect many product metrics and not accomplish any improvement. You might not even be getting basic assistance in achieving your current products customer-satisfaction objectives.



PROCESS MEASUREMENT

MEASUREMENT SYSTEMS

DARPA

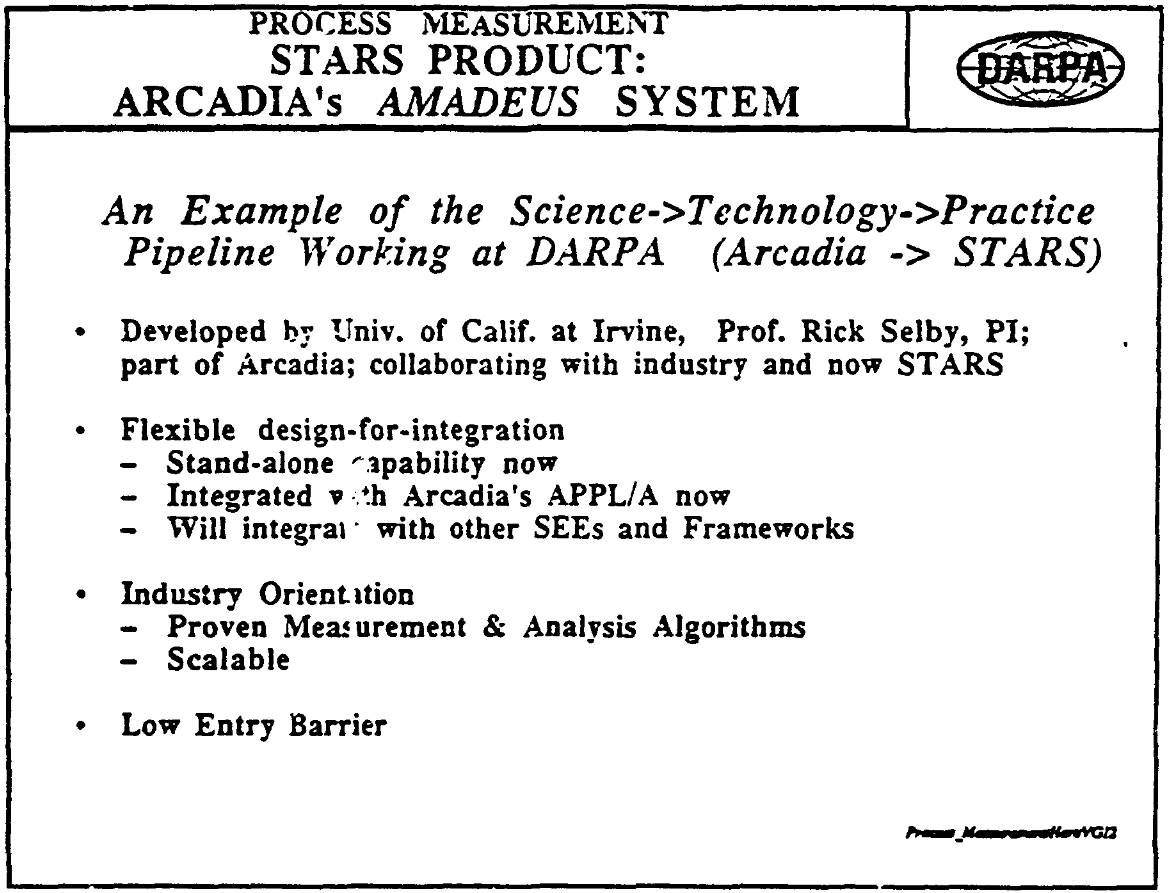
- Provide a way to speciŕy User-defined metrics to collect
  + usually a tailorable default set of metrics
- Instrumentation
- Collection
- Reporting
  - Including feedback for decision making or improvement
- Proactiveness
  - Automatically trigger specified process steps when certain metrics values or thresholds are attained
  - For example, project *replanning* when % calendar schedule is inconsistent with product Earned Value

Process_MeasurementHartVG11

MEASUREMENT SYSTEMS are how one collects these metrics. Many of the metrics are specific to a project or an organization, and hence are user defined, in the sense of the project member or another organization representative such as SEPG members. Usually there's a core set of default metrics that everyone's interested in, so they're all provided by a Measurement System as automatic user selections, for example, (Source) Lines of Code (SLOC or LOC) predicted and produced [no one's favorite metric, but too widely used to ignore] ...And so, measuring, storing, and counting SLOCS associated with modules and subsystems is one of those metrics you expect to be collected in anybody's minimalistic Measurement System. These systems provide for instrumentation, that is the identification of circumstances or points in a process at which different metrics are established and available. They provide collection of measurements based on what's been instrumented into activities. They provide reporting, which can be done in a variety of ways, including numerical calculations and analyses for feedback or presentation, perhaps interactively, to project managers or perhaps other team members in making the decisions that inevitably come up because we don't have perfect insight into defining every minute detail of processes ahead. So, there's always a degree of uncertainty about software processes (as with almost all human endeavors) going into the project.

Proactiveness — that's one of our goals. To be able to really support the well disciplined recognition of when particular metrics cross thresholds or attain specific relationships that should trigger specific events, process steps, for example, when I&T has expended 50% of its budget and only 20% of the modules have come together (or maybe even 50/50 is a red flag, since you might expect the easier ones to be done first) somebody ought to take corrective actions, perhaps replanning, perhaps changing test cases, perhaps changing integration method. Or, when a certain % of the total project calendar has passed and the measured system product's Earned Value that is collected in the EV system is far less than that % (assuming the plan calls for, and has EV measurements supporting, EV growing according to resource or schedule expenditure).

Notice I didn't say anything on this chart about Automation. All of this can be done, and much of it is being done in practice today, manually in many organizations. It's an obvious opportunity for automation, although not absolutely mandatory. STARS's objective is to provide low-cost measurement capabilities, based on automated aids, that clearly outweigh their costs in providing thorough, timely, reliable, flexible measurement information for usage by all of project technical personnel, project managers, process engineers, and organizations such as SEPG's. Not just Process Metrics, but also Product and Project Management metrics, remembering that the collected measurements input for these 3 purposes are often the same. But, Process purposes, which in a sense subsumes the purposes for the other kinds of measurements also, is our main motivation in STARS developments.



PROCESS MEASUREMENT

STARS PRODUCT: ARCADIA's *AMADEUS* SYSTEM

An Example of the Science->Technology->Practice Pipeline Working at DARPA (Arcadia -> STARS)

- Developed by Univ. of Calif. at Irvine, Prof. Rick Selby, PI; part of Arcadia; collaborating with industry and now STARS
- Flexible design-for-integration
  - Stand-alone capability now
  - Integrated with Arcadia's APPL/A now
  - Will integrate with other SEEs and Frameworks
- Industry Orientation
  - Proven Measurement & Analysis Algorithms
  - Scalable
- Low Entry Barrier

Process_MeasurementHart/VG12

A STARS PRODUCT: ARCADIA's AMADEUS SYSTEM

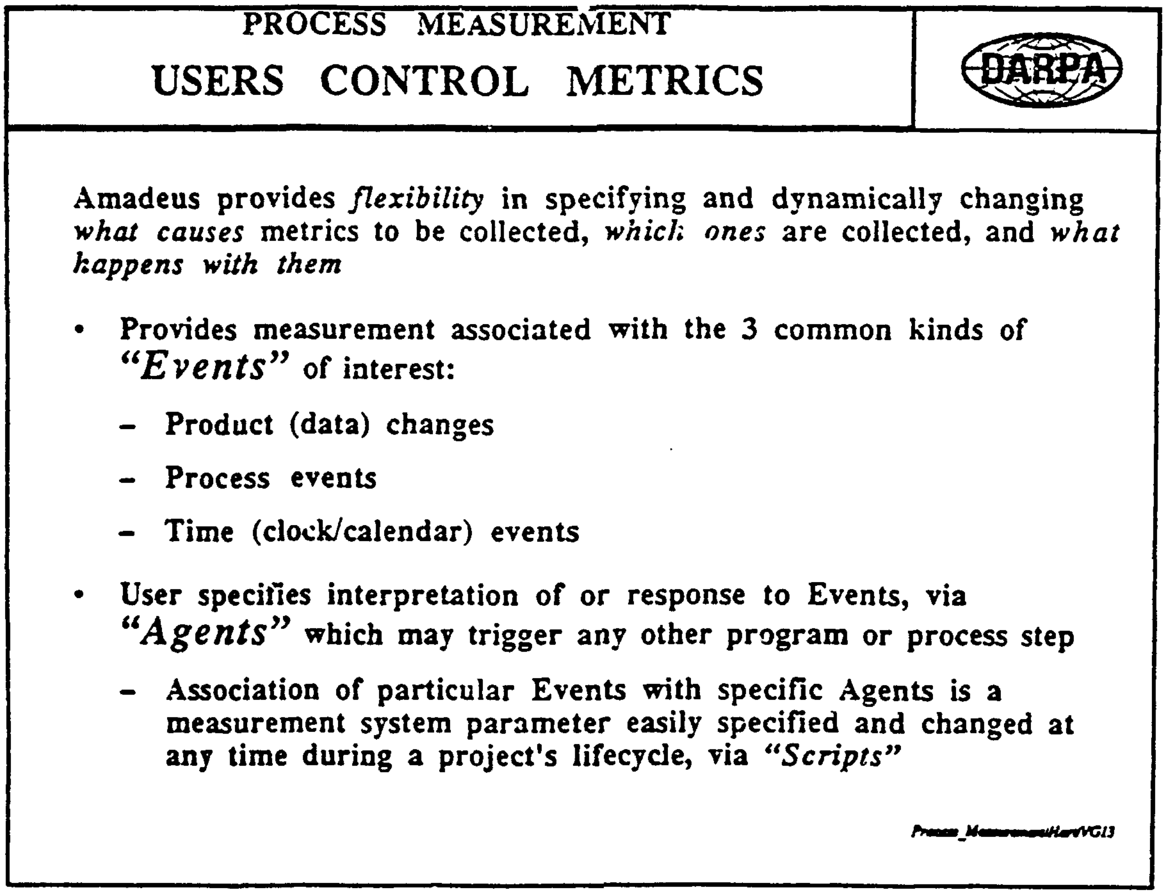
One measurement system that STARS is cooperating with now is Amadeus ...

This is a good example of the Science->Technology->Practice Pipeline, which is DARPA's long-term mission, working between two DARPA programs: Arcadia, which is very much and R&D program, and STARS.

Amadeus is developed by Univ. of Calif. at Irvine, with Prof. Rick Selby as the Principal Investigator.

Prof. Selby has been working on Amadeus as part of the Arcadia consortium for about 4 years, and also in collaboration with local industries to provide history databases and collected metrics, to validate his ideas and approaches about how a measurement system should be constructed and used by interacting with people responsible in companies for defining and carrying out measurement and improvement activities, and to acquire additional algorithms for computing various measures. So, Amadeus has already advanced beyond the level of university prototype. And, collaboration with STARS is now also underway.

Following charts will overview some of the reasons why STARS has selected Amadeus as a candidate for insertion into STARS SEE's, work started now and anticipated to be completed in the 1993 timeframe. These important features of Amadeus include flexible design-for-integration, proven incorporation of measurements and decision aids useful in industry, and a low-cost of starting to use it in current tool platforms (such as Unix).



PROCESS MEASUREMENT

USERS CONTROL METRICS

Amadeus provides *flexibility* in specifying and dynamically changing ***what causes*** metrics to be collected, ***which ones*** are collected, and ***what happens with them***

- Provides measurement associated with the 3 common kinds of *"Events"* of interest:
  - Product (data) changes
  - Process events
  - Time (clock/calendar) events
- User specifies interpretation of or response to Events, via *"Agents"* which may trigger any other program or process step
  - Association of particular Events with specific Agents is a measurement system parameter easily specified and changed at any time during a project's lifecycle, via *"Scripts"*

Process_MeasurementHandVG13

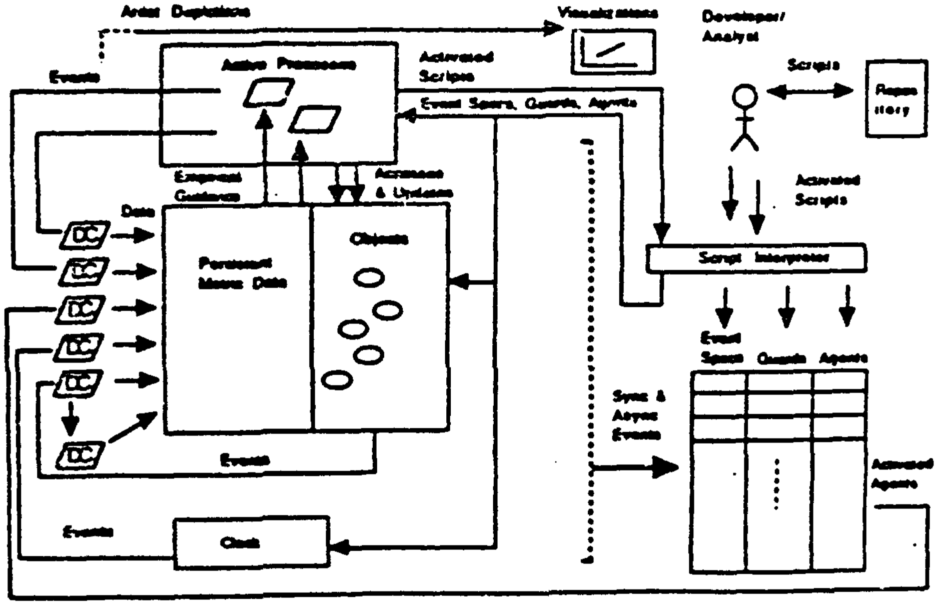
THE USERS CONTROL WHICH METRICS

The Amadeus system,has the ability for users to specify events of 3 different kinds of Events:... (1) Changes in project products or data (documents, software, test database); (2) Process events (for example, completion of milestones such as PDR or I&T, or usage of a design tool); and (3) clock-based events (passage of specified intervals of calendar time).

In Amadeus system, there is the capability to specify *Events* of interest at the time projects are started, or when tools are installed, or even by resetting notification mechanisms on databases, frameworks, mailers, etc.. Such Events trigger the collection of particular metrics into a persistent database. This specification is detached from the specification of *Agents*, which invoke analysis procedures, or interact with project members in a decision-making dialog. Because of that decoupling of Agents from Events, Amadeus offers a great deal of flexibility to modify a project's measurement activities, and thereby improve the process, during a project.

Amadeus Project

Conceptual Operation of Amadeus



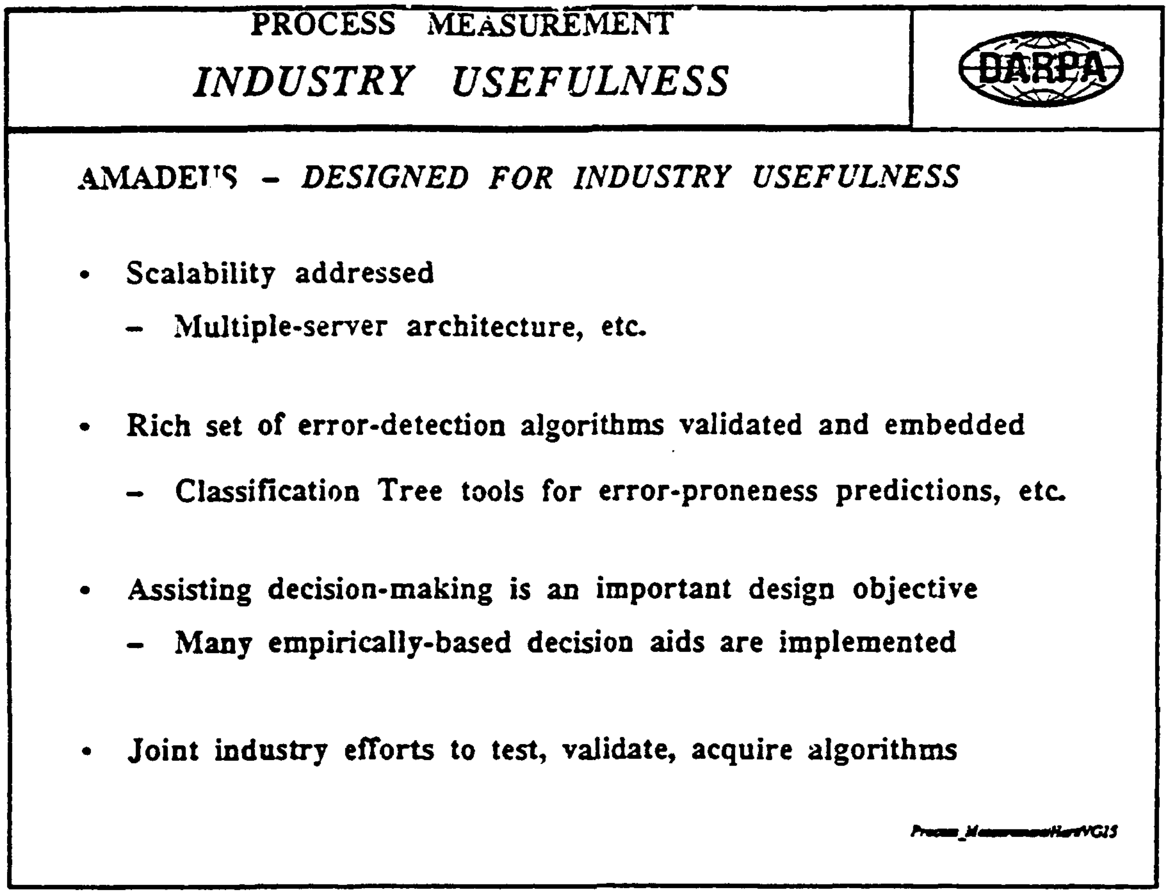
Process_MeasurementHart/VG14

University of California, Irvine

CONCEPTUAL OPERATION OF AMADEUS

This a quick conceptual operation illustration for Amadeus.

The scripts represent Events that are of interest and the associated Agents. Each script specifies one Event-Agent pair. These are what are set up at project initiation and that can be changed during the carrying out of the project. An Agent may trigger Data Collection activities, or they may trigger analysis programs. The Events may come from Process Programs; they may come from clocks; or they may come from changes to the data store. The Amadeus system is tied to an environment's persistent storage, and provides an interpretive approach for installing and running scripts, thereby providing the ability to dynamically change what's measured during a project, due to the interpretive, not hard-wired, carrying out of specified measurement activities. Amadeus provides the ability to install Analysis Tools that may be become available after project start-up, perhaps in response to Events that are determined after project start-up to be worthy of signalling for data collection or other measurement or analysis actions. There's an event stream that flows through this interpreter, and if there's no script currently installed and activated that cares about a particular event, it just "goes on by." But if you later decide that, for example, designs weren't passing reviews even after rework, project management may want to install a new script to compute and collect Complexity metrics over designs after every update to the design database. So, a user can either author a script, or take a script from a translation of a changed Process Program that describes this refinement to the process, and install it into the active script table, and that particular process improvement would henceforth be effected on the project automatically.



PROCESS MEASUREMENT

INDUSTRY USEFULNESS

AMADEUS - *DESIGNED FOR INDUSTRY USEFULNESS*

- Scalability addressed
  - Multiple-server architecture, etc.
- Rich set of error-detection algorithms validated and embedded
  - Classification Tree tools for error-proneness predictions, etc.
- Assisting decision-making is an important design objective
  - Many empirically-based decision aids are implemented
- Joint industry efforts to test, validate, acquire algorithms

Process_MeasurementHart/VG15

INDUSTRY USEFULNESS

Prof. Selby has done several things to distinguish Amadeus from blue-sky prototypes.

One of them is scalability. He has observed and learned that, without special engineering of the software prototypes, too often prototypes that work well for one-person situations fail to work for hundred- or thousand-person situations without serious, costly, sometimes unachievable redesign.

His approach to scalability is to have concurrent interpreters, multiple instances of that measurement script interpreter I mentioned on the previous chart, so that, depending on the underlying equipment architecture, that functionality can be distributed to avoid bottlenecks with monitoring of a highly concurrent set of activities corresponding to a large project team. Additionally, they architected their persistent storage as general interfaces that are known to be typically implementable on existing commercial database systems, so that the performance is at least predictable and gradable with respect to very large volumes of data.

I mentioned before the decoupling given by Amadeus's design between the events that signal moments when the state of project development is worth examining, and the Agents that collect specific data at those moments or perform actions specified in response to arriving at such moments. Recall that an Agent can call a particular analysis algorithm, and also recall that the Agents can be remapped to Events (meaning analysis changes) by changing scripts during a project, as well as between projects across an organization. Amadeus provides a fairly extensive set of useful analysis algorithms, from both the research community and industry that they have collected (some of which Prof. Selby and his colleagues developed). An example follows on the next chart — Classification Tree tools for predicting error-proneness modules.

Assisting decision making is an important objective in Amadeus. The Amadeus system is available with implementations of basic decision aids (like those I've mentioned earlier) based on the included metrics and algorithms. Of course, the script mechanism allows project users to develop their own decision aids and install them into their environment flexibly using Amadeus.

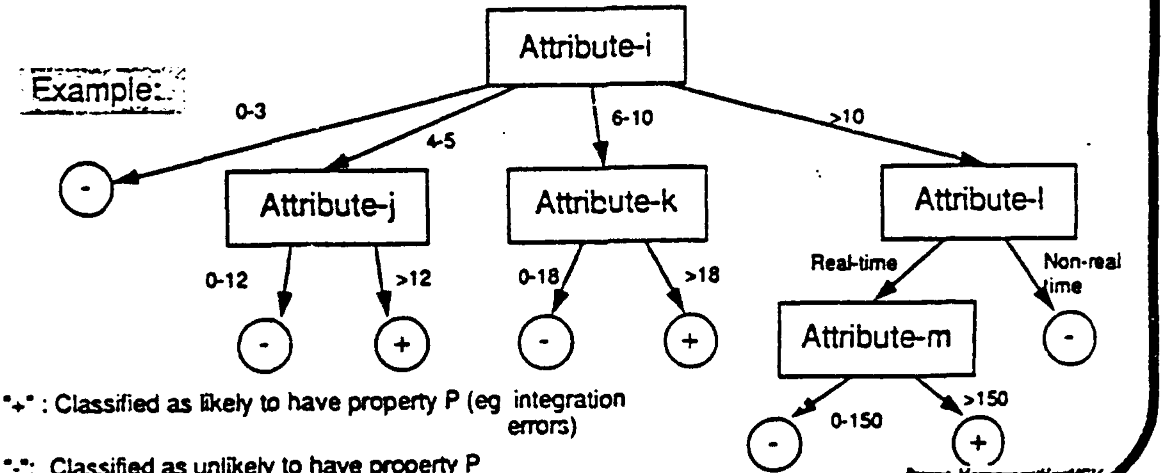
Amadeus Project

Empirically Based Techniques

- Scalable to large projects
- Calibratable to new environments
- Measurements are integratable
- Leverage previous experience
- Process, product & team attributes

Focus on High-Payoff Areas: the 80:20 rule

Example:



"+" : Classified as likely to have property P (eg integration errors)

"-": Classified as unlikely to have property P

University of California, Irvine

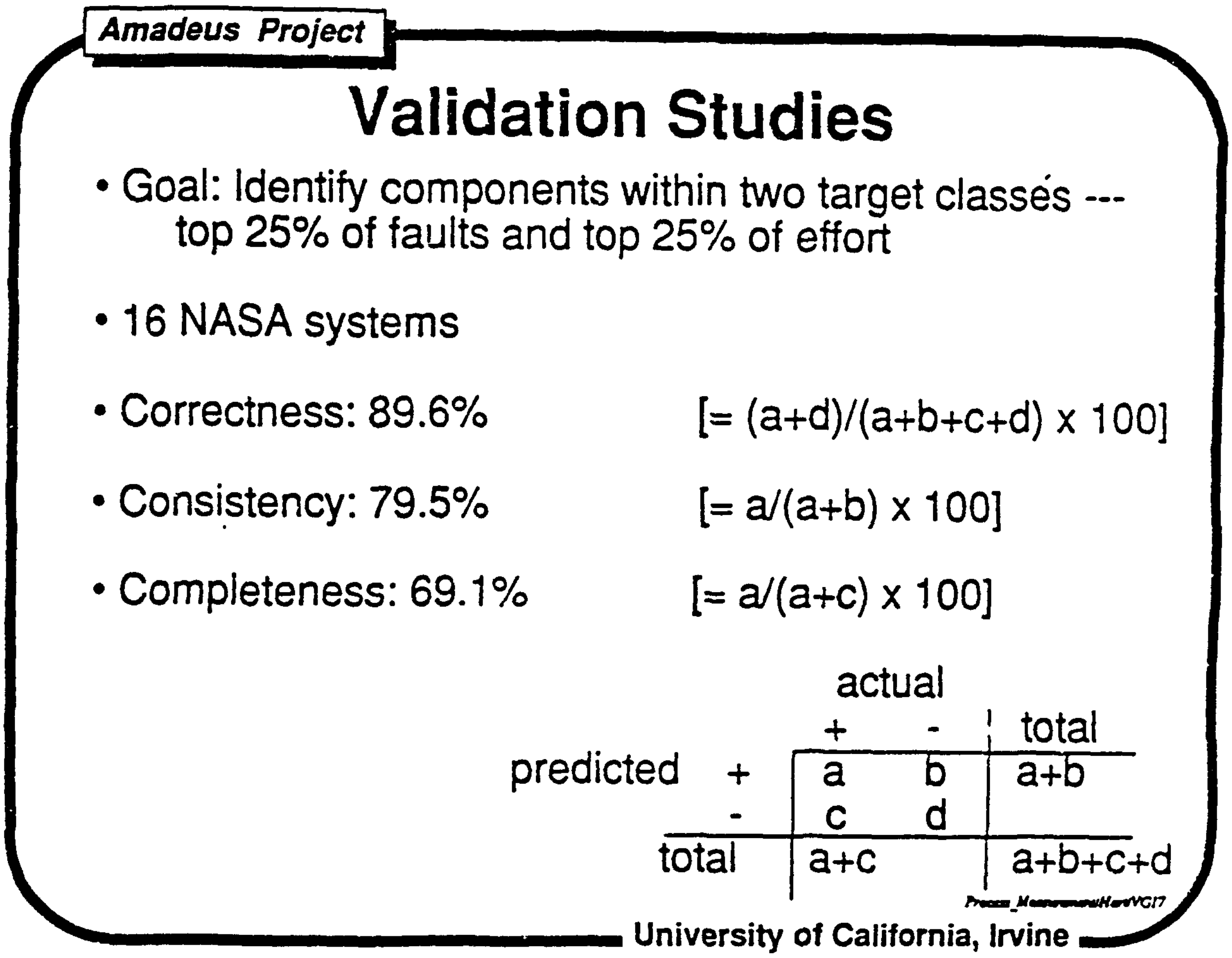
EMPIRICALLY BASED TECHNIQUES: A CLASSIFICATION TREE EXAMPLE

Attribute J is a product metric gathered or computed on a software module under development. It doesn't necessarily have to be quantified (numerical); it could be "Yes" versus "No," it could be "developed by computer programmer" vs "by system architect" vs "by process engineer" vs "by I&T team"; it could be a reflection of a reused modules heritage. Based on ranges of values of that attribute J, other product metrics of interest may then be indicated as worth examining; at the 2nd level, the algorithm might just compute one metric broken into many ranges, or maybe some different metrics with fewer ranges each, as illustrated in this example which depicts a classification tree to identify modules likely to exhibit some troublesome property. This repeated classification might be repeated hierarchically at several levels (2 in the case illustrated at level, 3 at right). eventually boiling down to a definitive prediction as to whether or not a module is likely or unlikely to be among the 20% most error prone, or to exhibit some other property which is the subject of a different classification tree.

A surprising finding in the field is that there is no single or small number of root-node product metrics that are always most useful to be examined first in calculating error proneness. This varies so much between organization practices, application domains, equipment architectures, test methods, etc., that it is almost impossible to develop generally applicable classification tree algorithms that are not highly parameterized by such application-domain and other characteristics of the setting of the software development project. This means that it is almost impossible to implement a simpler approach than indicated here as a calculation over multiple collected measurements — notice that these are at least 3 different metrics depending upon the range generated by the first metric (the root node).

Prof. Selby's group is part of is a broad community that is building up empirical databases and learning about what the relevant metrics are to place in classification trees under what circumstances, and thereby refining the partitioning reflected by the branching in the tree. They're learning by building up a body of knowledge to improve these algorithms: They're learning about application domains and the correlations between particular properties of those domains and particular properties (measured by product metrics) of software that indicate,for example, error proneness, or reusability, or ease of change, etc. Given a set of characteristics, there's a growing likelihood that they have already identified the root-node product metric to compute and probably the next level metrics set too.

So, there's a lot of pragmatic analysis, underlying theory and supporting empirical data, and evolutionary learning and expanding capabilities already accompanying Amadeus.



Amadeus Project

Validation Studies

- Goal: Identify components within two target classes --- top 25% of faults and top 25% of effort
- 16 NASA systems
- Correctness: 89.6% [= (a+d)/(a+b+c+d) x 100]
- Consistency: 79.5% [= a/(a+b) x 100]
- Completeness: 69.1% [= a/(a+c) x 100]

		actual		
		+	-	total
predicted	+	a	b	a+b
	-	c	d	
	total	a+c		a+b+c+d

Process_Measurement/Hurt/VG17

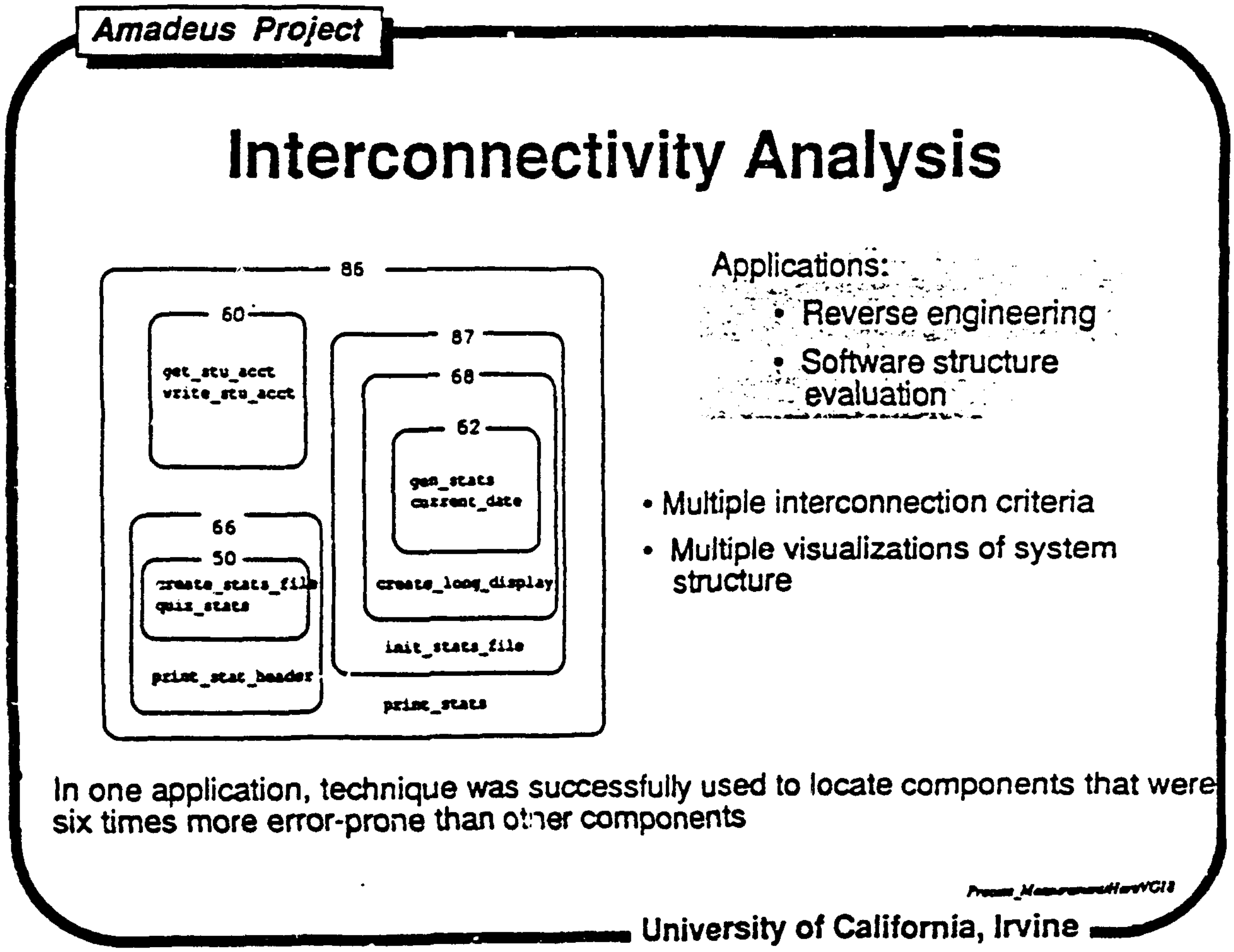
University of California, Irvine

VALIDATION STUDIES (OF THE CLASSIFICATION TREE EXAMPLE)

Two classification trees like that illustrated on the previous chart have been recently validated against 16 NASA systems: one predicting top-quartile error proneness and the other predicting high effort upper quartiles for software modules entering an I&T activity.

Correctness is measured as the sum of modules predicted correctly to be error-prone plus those correctly predicted to not be error-prone, divided by the total number of all modules (which includes those predicted wrong either way as later determined by integration experience). 90% correctness is regarded as outstanding, and would certainly be welcome predictions for project management prioi to a.ı I&T activity.

Another approach to validating the error or effort prediction technique is Consistency, measured as percent of those predicted in the top quartile that actually turned out to be. And, Completeness is percent of actual top-quartile modules that were identified by the predictor. Validations of almost 80% and 70% prove how useful these algorithms would be distributing labor during I&T.



86
60
get_stu_acct
write_stu_acct
87
68
62
gen_stats
current_date
66
50
create_stats_file
quiz_stats
create_loog_display
init_stats_file
print_stat_header
print_stats

Amadeus Project

Interconnectivity Analysis

Applications:
- Reverse engineering
- Software structure evaluation

- Multiple interconnection criteria
- Multiple visualizations of system structure

In one application, technique was successfully used to locate components that were six times more error-prone than other components

Process_MeasurementHart/VG18

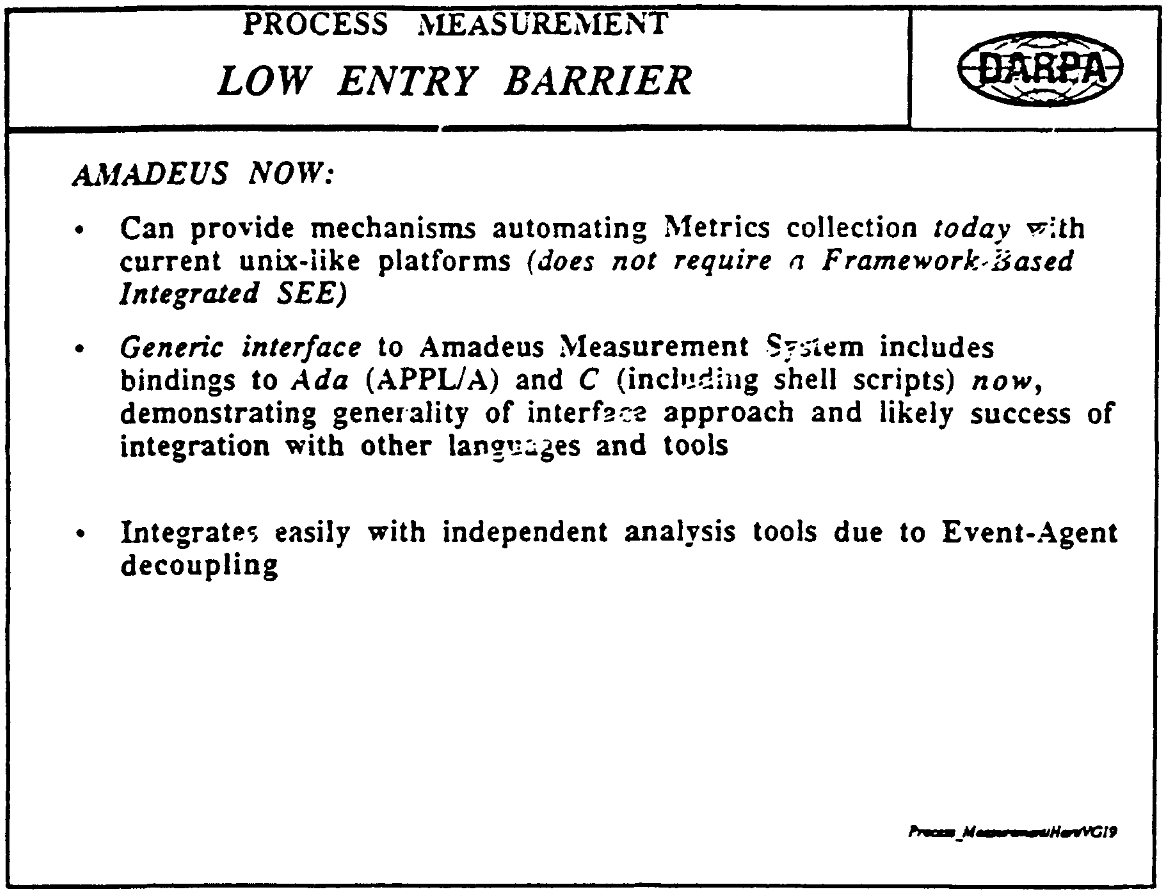
University of California, Irvine

INTERCONNECTIVY ANALYSIS: ANOTHER AMADEUS-SUPPORTED PROCESS AID

A quick glimpse at one more kind of analysis — Interconnectivity analysis.

This tells you, given a "troublesome" module, what other yet-untested modules are most likely to share the same troublesomeness. That's the essence of Interconnectivity analysis.

Some Interconnectivity analysis algorithms are available with Amadeus, and Prof. Selby's group is part of active research developing more such algorithms.



PROCESS MEASUREMENT

LOW ENTRY BARRIER

AMADEUS NOW:

- Can provide mechanisms automating Metrics collection *today* with current unix-like platforms *(does not require a Framework-Based Integrated SEE)*
- *Generic interface* to Amadeus Measurement System includes bindings to *Ada* (APPL/A) and *C* (including shell scripts) *now*, demonstrating generality of interface approach and likely success of integration with other languages and tools
- Integrates easily with independent analysis tools due to Event-Agent decoupling

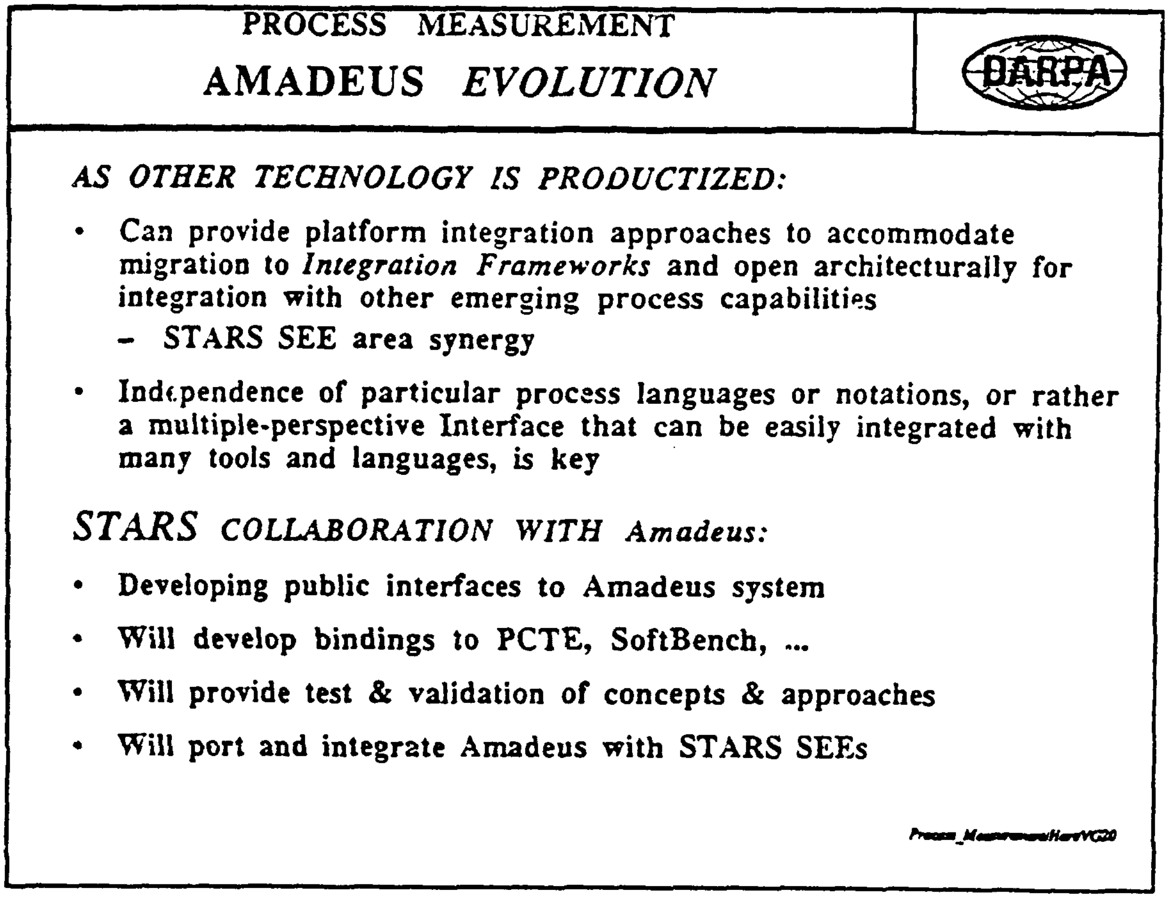
Process_Measurement/Hart/VG19

LOW ENTRY BARRIER

Finally, "low entry barrier" was an important objective in the design of the Amadeus system, meaning that organizations can insert the Amadeus measurement systems into their development settings regardless of the sophistication or integration existing in their available tool platforms or environments. Stated otherwise, that an organization could very gradually, at small start-up cost and training, insert some or all of Amadeus's automated capabilities.

Amadeus runs stand-alone on Unix-like platforms *today*. It can be instrumented and controlled by C-shell-like scripts *today*. The interface to the underlying Amadeus system already has bindings to Ada (therefore APPL/A) and C *today*. This already demonstrates the success of the generic interface approach in Amadeus's design, and indicates the likelihood that integration with other languages, including process description languages like MVP, will also be straightforwardly accomplished; such further language integration, as well as platform portability, are being started as collaborative efforts with STARS.

And, Amadeus supports the independent insertion of newly developed or newly available analysis tools, via the Event-Agent script paradigm I described earlier. This promotes both initial integration of analysis tools already in practice and familiar to a project team, and the expansion of analysis tools available to the team.



PROCESS MEASUREMENT

AMADEUS *EVOLUTION*

AS OTHER TECHNOLOGY IS PRODUCTIZED:

- Can provide platform integration approaches to accommodate migration to *Integration Frameworks* and open architecturally for integration with other emerging process capabilities
  - STARS SEE area synergy
- Indtpendence of particular process languages or notations, or rather a multiple-perspective Interface that can be easily integrated with many tools and languages, is key

STARS COLLABORATION WITH Amadeus:

- Developing public interfaces to Amadeus system
- Will develop bindings to PCTE, SoftBench, ...
- Will provide test & validation of concepts & approaches
- Will port and integrate Amadeus with STARS SEEs

Process_Measurement/Hart/VG20

AMADEUS EVOLUTION

As other process and SEE technology is productized, Amadeus's architecture, based on the generic (language-independent) interface approaches which characterize all of Arcadia's environment architectural approaches, facilitates introduction of and evolution to the highly integrated, synergistic framewcrk-based SEEs of the future. This is already an instance of significant synergy between STARS's SEE and Process developments areas.

And additionally, as I mentioned before, there is the promise of interfacing Amadeus to other process programming languages and specification approaches than APPL/A — meaning that process engineers might use statements in these languages to indicate where and what metrics and measurements and analyses and possibly what red-flagged actions are of interest, and then translators for those languages produce outputs that directly or indirectly lead to Amadeus scripts.

I have already described a significant Amadeus implementation that runs today and is being productized, refined, extended, and validated by current activities in collaboration with many organizations, particularly industry. STARS organizations are part of that effort.

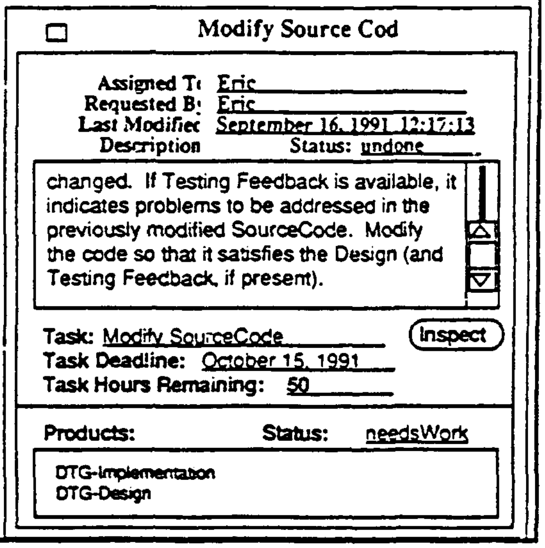
Furthermore, STARS is directly supporting several activities toward productization of Amadeus that we have determined to be vital to STARS's objectives. These include (1) developing documented public interfaces to Amadeus system; (2) developing bindings to PCTE and probably SoftBench and other framework-like products; (3) testing and validating Amadeus concepts and approaches via trial usage on friendly projects at the cites of STARS contractors, including incorporation of in-practice industry metrics and analysis algorithms back into Amadeus as possible; and (4) porting and integrating Amadeus with STARS SEEs. The first two of these activities are underway now, the third has started via collaboration between UC Irvine and TRW's CCPDS-R project, and the fourth will be facilitated by the first two (or three).

PROCESS MEASUREMENT

STARS PRODUCT: PREIS APPROACH

DARPA

ActionItem Message to Programmer



□ Modify Source Cod

Assigned Tı Eric
Requested B: Eric
Last Modifiec September 16, 1991 12:17:13
Description Status: undone

changed. If Testing Feedback is available, it indicates problems to be addressed in the previously modified SourceCode. Modify the code so that it satisfies the Design (and Testing Feedback, if present).

Task: Modify SourceCode
Task Deadline: October 15, 1991
Task Hours Remaining: 50

Products: Status: needsWork

DTG-Implementation
DTG-Design

Adapted from Boeing/Honeywell STARS work.

Process_MeasurementHart/VG21

STARS PRODUCT: PREIS APPROACH

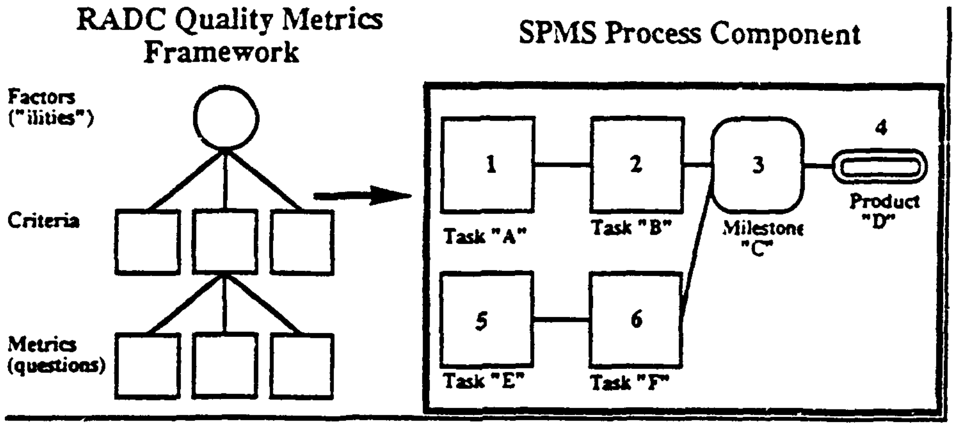
There are at least three prototype STARS products that you have heard about or can see in the demonstrations that incorporate varying aspects of measurement capabiliues. All of these are products being delivered in the 2-year STARS timeframe, and you already see the recognition of the importance c: integrating measurement with these diverse Process-automation products.

The first is the prototype EIS system. You see that it obviously has a set of collected measurements, here reflected in the Action Item Browser's *Process Metric Notification* feature. As enactment occurs in the Control Point enactment system, notices in the form of ActionItems are sent to the appropriate users. Included in these messages is information concerning the product and process data collected during process execution from the metric parameters defined for the process. This represents a fairly fixed set of metrics associated with the particular process definition embodied in EIS now.

PROCESS MEASUREMENT

STARS PRODUCT: SPMS APPROACH

SPMS Integrates Quality Metrics and Process Models



Adapted from IBM/SAIC STARS Work

Process_MeasurementHartVG22

PSTARS PRODUCT: SPMS APPROACH

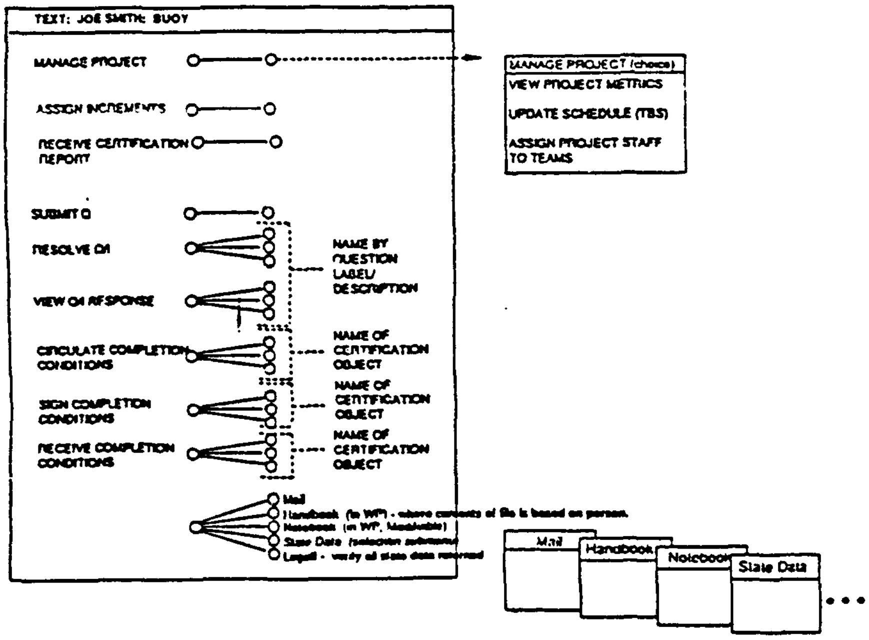
SPMS Integrates Quality Metrics and Process Models.

The RADC Quality Metrics Framework represents 13 factors ("ilities"), 29 software-oriented criteria, and several hundred metrics/questions/formulas for calculating software product quality attributes. SPMS has stored the entire RADC framework and provides the tailoring tools so that a workable subset can be defined for a given project. Specific metrics from the tailored quality model can then be associated with the exit criteria of tasks in the defined process model by the creation of "data collection forms" (DCFs) that will trigger when a particular task with such an exit criteria is "executed". The data input at that point will then be used along with the pre-stored formulas to compute a value that is compared to a "threshold" established in the process model task which causes the execution of the task to pass or fail. Failure would then be handled by cloning a rework network, etc. Such capabilities are being provided extensively in SPMS's process simulator.

The RADC Framework was chosen because it is so fully defined, truly capturing years of Metrics research and avoiding time-consuming, costly development on STARS, and guaranteeing project usefulness in the 2-year STARS window. But additionally, SPMS intends to add other metrics capabilities (for example, those from Selby and his colleagues as done by Amadeus).

PROCESS MEASUREMENT

STARS PRODUCT: CEPA APPROACH


TEXT: JOE SMITH: BUOY
MANAGE PROJECT
ASSIGN INCREMENTS
RECEIVE CERTIFICATION REPORT
SUBMIT Q
RESOLVE OI
VIEW OI RESPONSE
CIRCULATE COMPLETION CONDITIONS
SIGN COMPLETION CONDITIONS
RECEIVE COMPLETION CONDITIONS
NAME BY QUESTION LABEL/ DESCRIPTION
NAME OF CERTIFICATION OBJECT
NAME OF CERTIFICATION OBJECT
NAME OF CERTIFICATION OBJECT
Mail
Handbook (in WP) - where contents of file is based on person.
Notebook (in WP, Modifiable)
State Data
Logoff - verify all state data returned
MANAGE PROJECT (choice)
VIEW PROJECT METRICS
UPDATE SCHEDULE (TBS)
ASSIGN PROJECT STAFF TO TEAMS
Mail
Handbook
Notebook
State Data

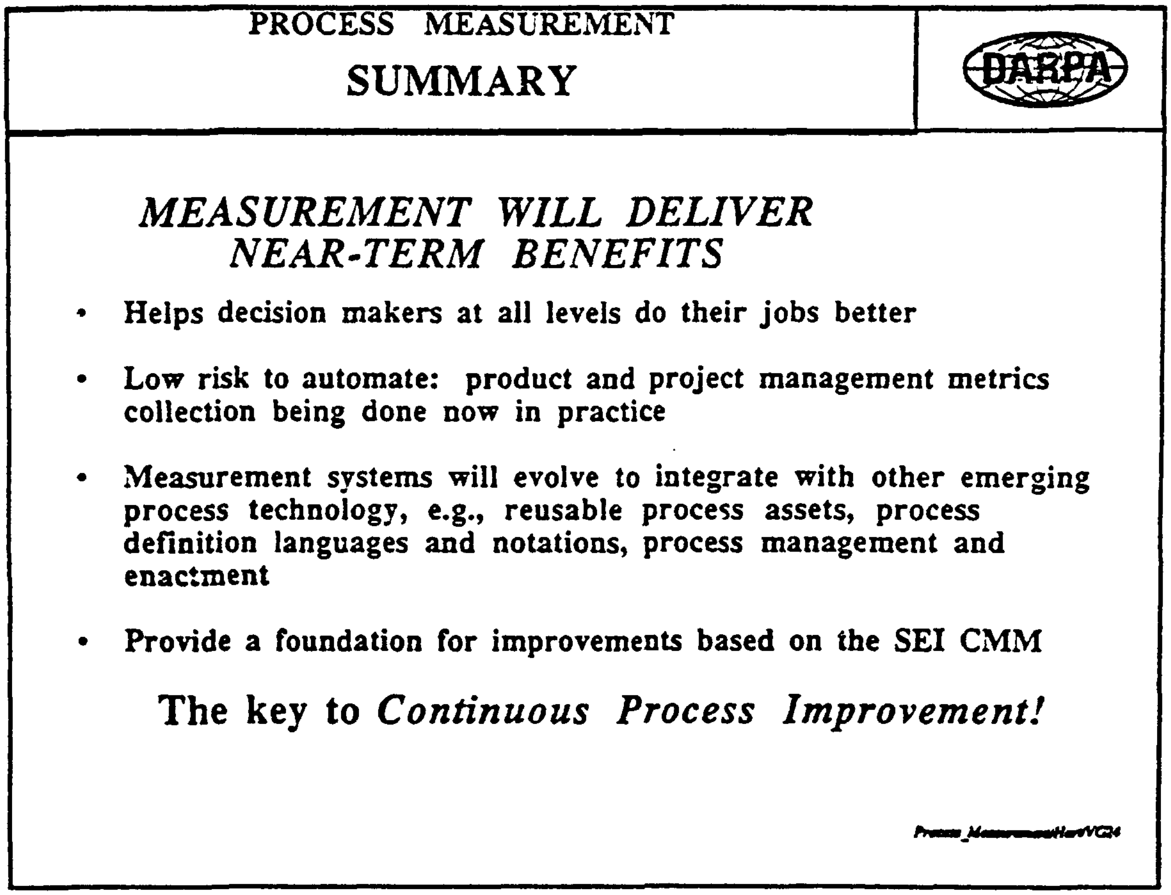
Process_MeasurementHartVG23

STARS PRODUCT: CEPA APPROACH

The Cleanroom Engineering Process Assistant, implementing a particular defined process, also provides evidence of the integrated role of measurements and metrics in Process Management.

So, you see, metrics is pervasive in all good processes and supported in the automation of good processes. Metrics and measurement cut across them all. You should not be surprised that all of these other products include facets of metrics and measurement.

Hence, common underlying support, supportive of many metrics and analyses but not necessarily tied to any one set of them, looks like a cost-effective solution. That's why Amadeus is constructed to be used the way I described; and Amadeus could be be installed and integrated with Process Automation systems such as seen here in PREIS, SPMS, and CEPA, without necessarily changing any of the user screens shown for those systems, but offering flexible extensions of those and other user screens presenting measurements triggered by Amadeus Events and processed by Agents.



PROCESS MEASUREMENT

SUMMARY

DARPA

MEASUREMENT WILL DELIVER NEAR-TERM BENEFITS

- Helps decision makers at all levels do their jobs better
- Low risk to automate: product and project management metrics collection being done now in practice
- Measurement systems will evolve to integrate with other emerging process technology, e.g., reusable process assets, process definition languages and notations, process management and enactment
- Provide a foundation for improvements based on the SEI CMM

The key to *Continuous Process Improvement!*

Process_Measurement/Hart/VG24

SUMMARY

So, in summary, Measurement technology will deliver near-term benefits, for example, helping makers at all levels do their jobs better. This is Process, and it's Process improvement within the context of an ongoing project.

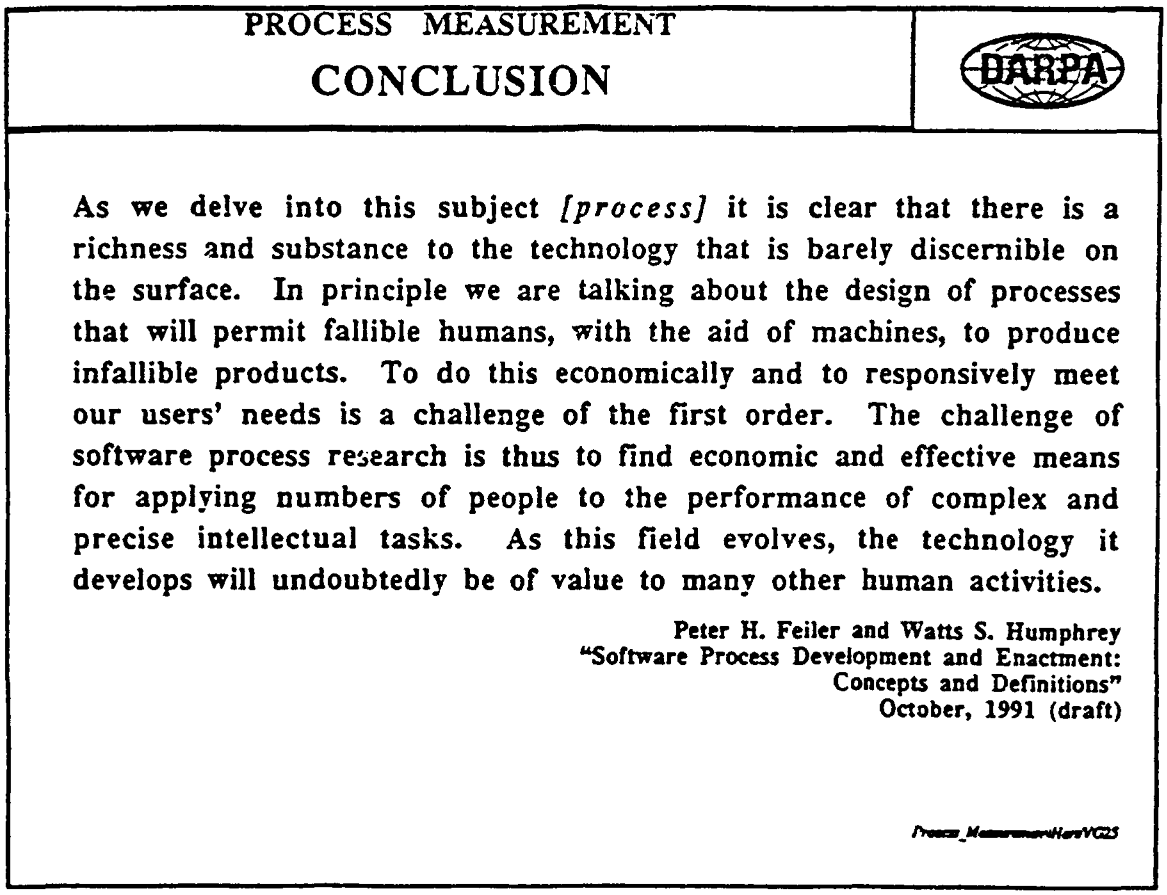
Extensive measurement represents a low risk to automate — significant automation of product and project management metrics collection is in place now in practice in many organizations. And, it does not require the existence of a SEE integration framework to be able to use it now.

Measurement systems will evolve to integrate with other emerging process technology and process capabilities you've heard about in this track, for example, reusable process assets, process definition languages and notations, process management systems, and enactment aids.

Going to automated measurement, which you can really use to assess, guide, and control project activities, will provide a foundation for improvements based on the SEI's Capability Maturity Model.

And, as has been stated repeatedly: You don't know what to improve, or if attempted improvements are paying off, if you don't have real measurement.

Measurement is The key to Continuous Process Improvement!



PROCESS MEASUREMENT

CONCLUSION

As we delve into this subject *[process]* it is clear that there is a richness and substance to the technology that is barely discernible on the surface. In principle we are talking about the design of processes that will permit fallible humans, with the aid of machines, to produce infallible products. To do this economically and to responsively meet our users' needs is a challenge of the first order. The challenge of software process research is thus to find economic and effective means for applying numbers of people to the performance of complex and precise intellectual tasks. As this field evolves, the technology it develops will undoubtedly be of value to many other human activities.

Peter H. Feiler and Watts S. Humphrey
"Software Process Development and Enactment:
Concepts and Definitions"
October, 1991 (draft)

Process_MeasurementHart/VG25

IN CONCLUSION OF THIS TRACK

I would like to read you an interesting quote from Peter Feiler and Watts Humphrey in an SEI report on Process concepts and definitions that will soon be published.

As we delve into this subject *[process]* it is clear that there is a richness and substance to the technology that is barely discernible on the surface. In principle we are talking about the design of processes *[that's our Process Definition activities]* that will permit fallible humans, with the aid of machines, *[there's our Process Management or Process Automation activities]* to produce infallible products. *[At least, that's the vision]* To do this economically and to responsively meet our users' needs *[which Measurement technology helps us gauge and determine if we're succeeding, that's the essence of the Quality-oriented Product measurements]* is a challenge of the first order. The challenge of software process research is thus to find economic and effective means *[How to you know if they're "economic" and "effective" if you can't Measure them? These are the Productivity and Predictability objectives of Process and Project Management metrics] [Also, just as software Reuse might be our highest-leverage software approach, so this is the Process Asset Library's justification, so we learn and leverage from the best process practices and achieve economies and effectiveness]* for applying numbers of people to the performance of complex and precise intellectual tasks *[by carefully defining disciplined processes using Process Definition languages and Notations]*. As this field evolves, the technology it develops will undoubtedly be of value to many other human activities *[outside the software domain]*.

STARS '91

TRACK 2 INTRODUCTION

DARPA

Tuesday December 3, 1991

2:00–2:45	Domain-Specific Reuse—Vision, Strategies and Achievements	*Teri Payton, Unisys Defense Systems, Inc.*
2:45–3:15	Break	
3:15–4:00	Reuse Concepts	*Maggie Davis, Boeing*
4:00–4:30	Break	
4:30–5:15	Integrating Reuse into a Life-Cycle Process	*Bonnie Danner, TRW*
	Domain Analysis Process Model	*Dr. Ruben Prieto-Diaz, Reuse, Inc.*
8:00–9:30	Community Involvement Working Group: Domain-Specific Reuse	

STARS '91

TRACK 2 INTRODUCTION

DARPA

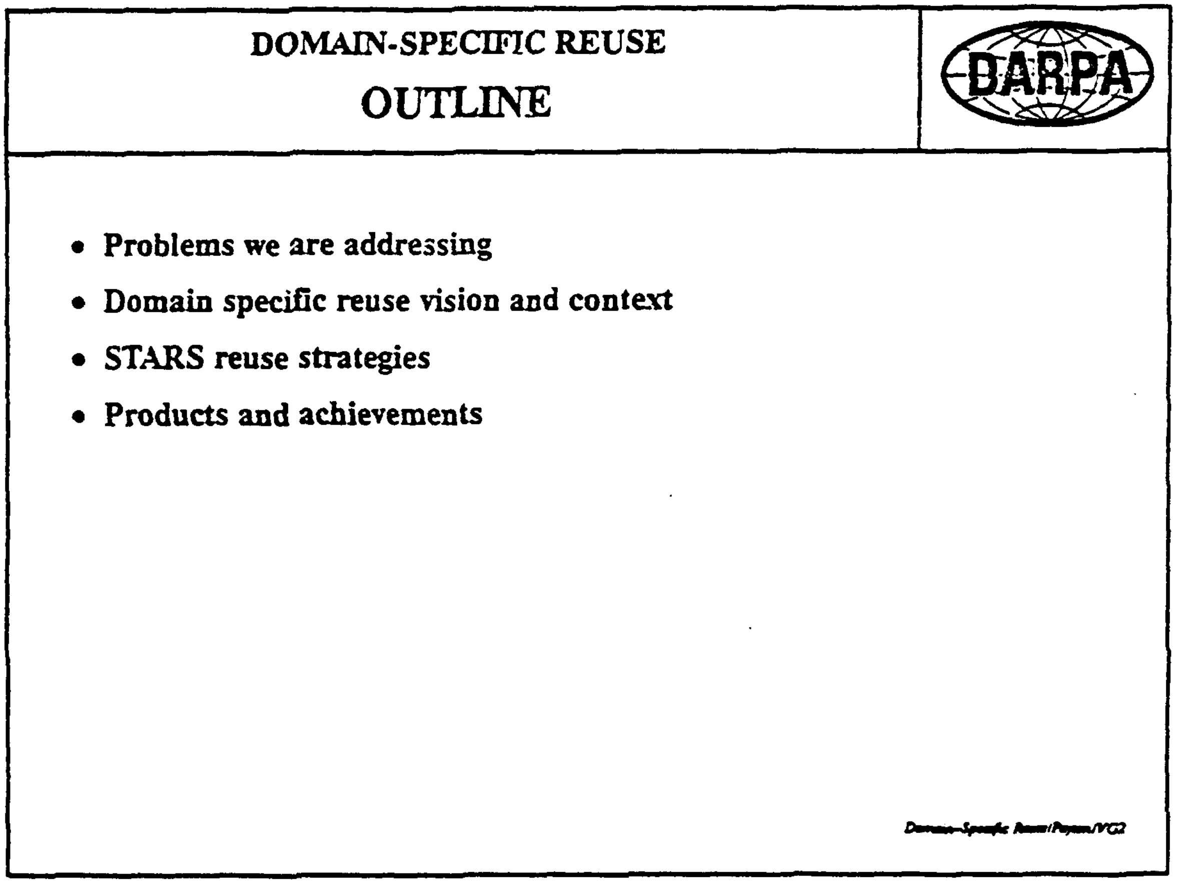
Wednesday December 4, 1991

8:30–9:15	STARS Asset Library Open Architecture Framework (ALOAF)	*Dick Creps, Unisys Defense Systems, Inc.*
9:15–9:45	Break	
9:45–10:30	STARS Library Mechanisms: Comparisons and Experiences	*Marlene Hazle, MITRE*
10:30–11:00	Break	
11:00–11:45	ASSET CARDS	*Jim Moore, IBM* *Rose Armstrong, EWA*
11:45–1:45	Lunch	
1:45–2:30	Domain-Specific Reuse Feedback Session	*Dave Ceely, IBM*

DARPA

DOMAIN-SPECIFIC REUSE: VISION, STRATEGIES AND ACHIEVEMENTS

Teri F. Payton
STARS Reuse Architect
Unisys Defense Systems, Inc.
3 December 1991
(703) 620-7770/(703) 351-5308
payton@stars.reston.unisys.com



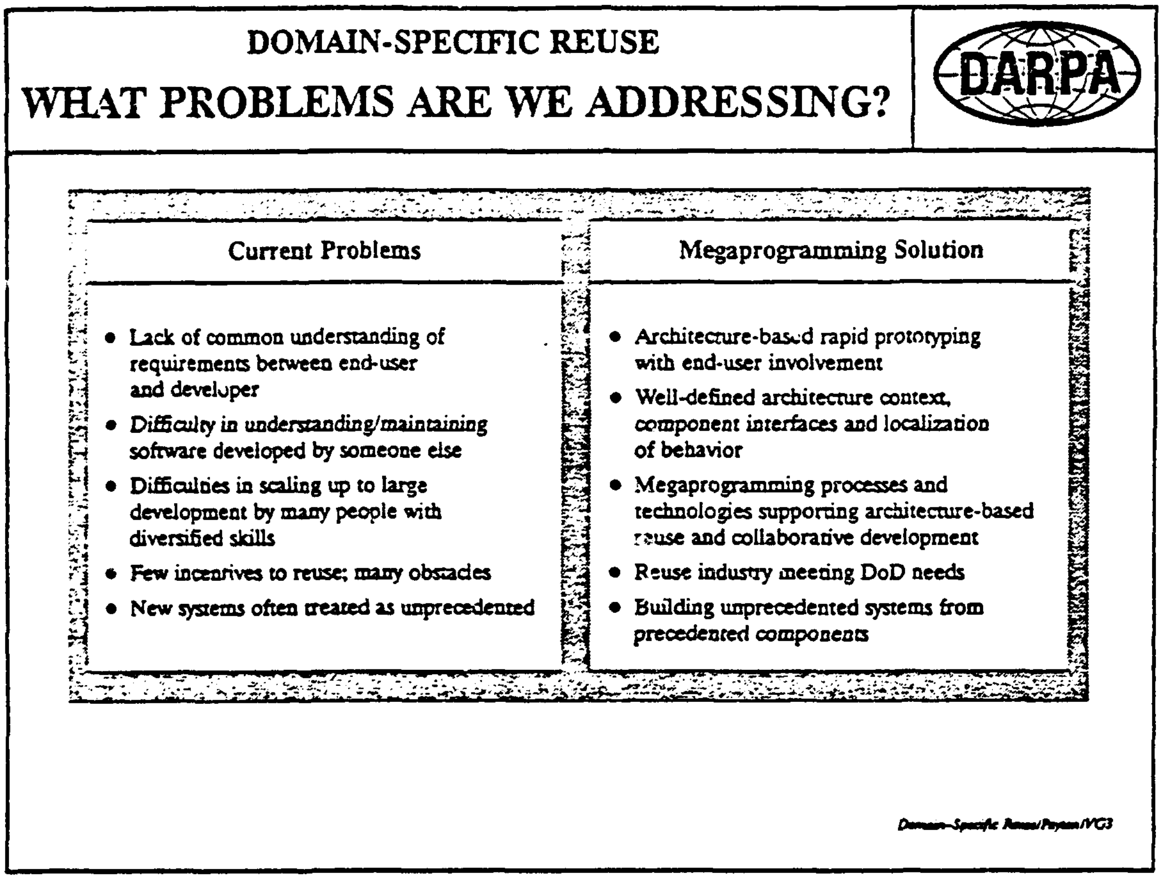
DOMAIN-SPECIFIC REUSE

OUTLINE

- Problems we are addressing
- Domain specific reuse vision and context
- STARS reuse strategies
- Products and achievements

Domain-Specific Reuse/Payton/VG2

This presentation will begin to articulate the problems that architecture-based domain-specific reuse addresses. It will elaborate the megaprogramming vision with respect to domain-specific reuse and provide a top-level view of STARS reuse strategies that assist in transitioning to megaprogramming. Highlights of STARS achievements to date will be presented. Currently available interim reuse products will be identified. STARS is interested in working with Technology Transfer Affiliates for review, trial usage and feedback on these interim products. Subsequent presentations in this track as well as the STARS demonstrations in the demo area will provide more detail on the interim products.



DOMAIN-SPECIFIC REUSE

WHAT PROBLEMS ARE WE ADDRESSING?

Current Problems	Megaprogramming Solution
• Lack of common understanding of requirements between end-user and developer	• Architecture-based rapid prototyping with end-user involvement
• Difficulty in understanding/maintaining software developed by someone else	• Well-defined architecture context, component interfaces and localization of behavior
• Difficulties in scaling up to large development by many people with diversified skills	• Megaprogramming processes and technologies supporting architecture-based reuse and collaborative development
• Few incentives to reuse; many obstacles	• Reuse industry meeting DoD needs
• New systems often treated as unprecedented	• Building unprecedented systems from precedented components

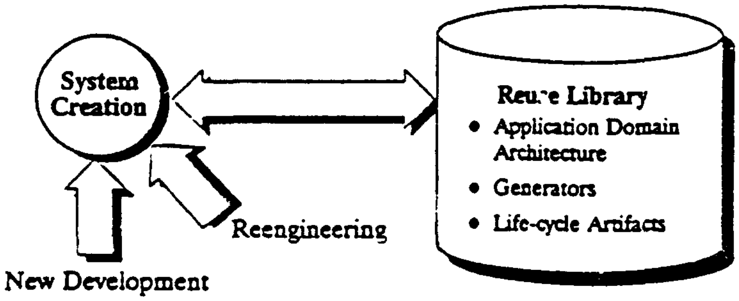
Domain-Specific Reuse/Payton/VG3

The problems that are addressed by domain specific reuse are not simply the commonly recognized issues with respect to the need to decrease cost or increase reliability by reusing existing well tested software. It goes much further than that in terms of grappling with the underlying problems of building DoD software that truly meets the end-user needs. Numerous past studies have identified lack of a common understanding of requirements as a significant problem. Domain specific reuse enables a change in the way we do business by facilitating architecture-based component-supported prototyping in which the end-user can be involved prior to definitization of the requirements. This will allow improved cost, schedule and functionality tradeoffs.

DOMAIN-SPECIFIC REUSE

VISION

- Guided by Reuse Process
- Based on Application Domain Architecture
- Systems Composed From Reusable Assets
- Assets Include Any/All Life-Cycle Artifacts
- Supports Continuous Improvement in Reuse Process/Products



Domain-Specific Reuse/Payton/VG4

In the future, we envision reuse-based software engineering processes guiding the development/evolution of software intensive systems. Specific system softw:r: architectures would be based on the generic application domain architecture and associated gen:ric requirements set. The system would be created or evolved using reusable assets. These assets can include application generators, reusable requirements and tests and any relevant life-cycle artifacts.

The processes would include reuse-based prototyping to assist in requirements definitization. Both rapid prototypes and eventual systems would be composed from reusable assets based on application domain architectures. We envision application generators to become increasingly important as one of the means of capturing and reusing application domain knowledge.

A system may also includ· reengineered components but that reengineering effort needs to be done in the context of the domain architecture. The reengineered components could then be provided back to the reuse library for usage on other programs.

DOMAIN-SPECIFIC REUSE

REAL-WORLD SUCCESS EXAMPLES

DARPA

- There are several successful instances within DoD and external to DoD of process-driven domain-specific reuse based development
  - Australian C^3
  - Foxboro process control
  - NobleTech (BOFORS)
  - Navy FCDSSA RNTDS
  - CCPDSR
- STARS goal:
  - Provide processes and automated capabilities to enable more DoD mission areas to transition to this approach
  - Work within the DoD community to catalyze removal of political, cultural, business and technical barriers

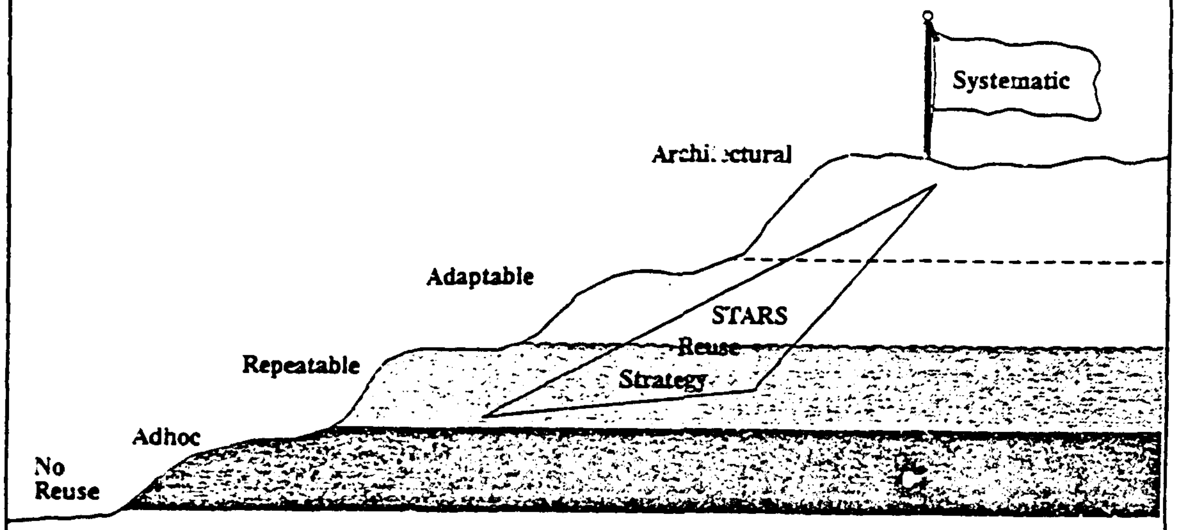
There are successful instances of reuse today which have resulted in significant cost savings and quality improvement for the organizations involved. The list provides a few examples ranging from commercial process control applications through European, Australian and American defense experiences. One item of significance is that the Navy Fleet Combat Directorate Systems Support Activity which is a post deployment support organization has been successful at reuse in its maintenance activity on several ship platforms as part of the Restructured Navy Tactical Data Systems. They have achieved over 80% reuse and significantly reduced the size of the workforce and the overall DoD costs in evolving these systems.

STARS goal is to enable more of these success stories in additional domains of interest to DoD by providing processes and automated capabilities to support reuse-based software engineering and by assisting DoD in understanding and addressing the non-technical as well as the technical barriers to reuse.

DOMAIN-SPECIFIC REUSE

STAGES OF REUSE

DARPA



Adapted from SPC Mount Reuse

Domain-Specific Reuse/Payton/VG4

The Software Productivity Consortium has defined a model for staged introduction of reuse. It is depicted in the diagram and often referred to as Mount Reuse. Within the community our goal is to transition from adhoc reuse to systematic reuse over time. Most organizations are dealing in adhoc reuse today. Individuals scavenge and find something to reuse. Adhoc reuse is based primarily on individual initiative and knowledge rather than corporate knowledge that is retained across people and projects.

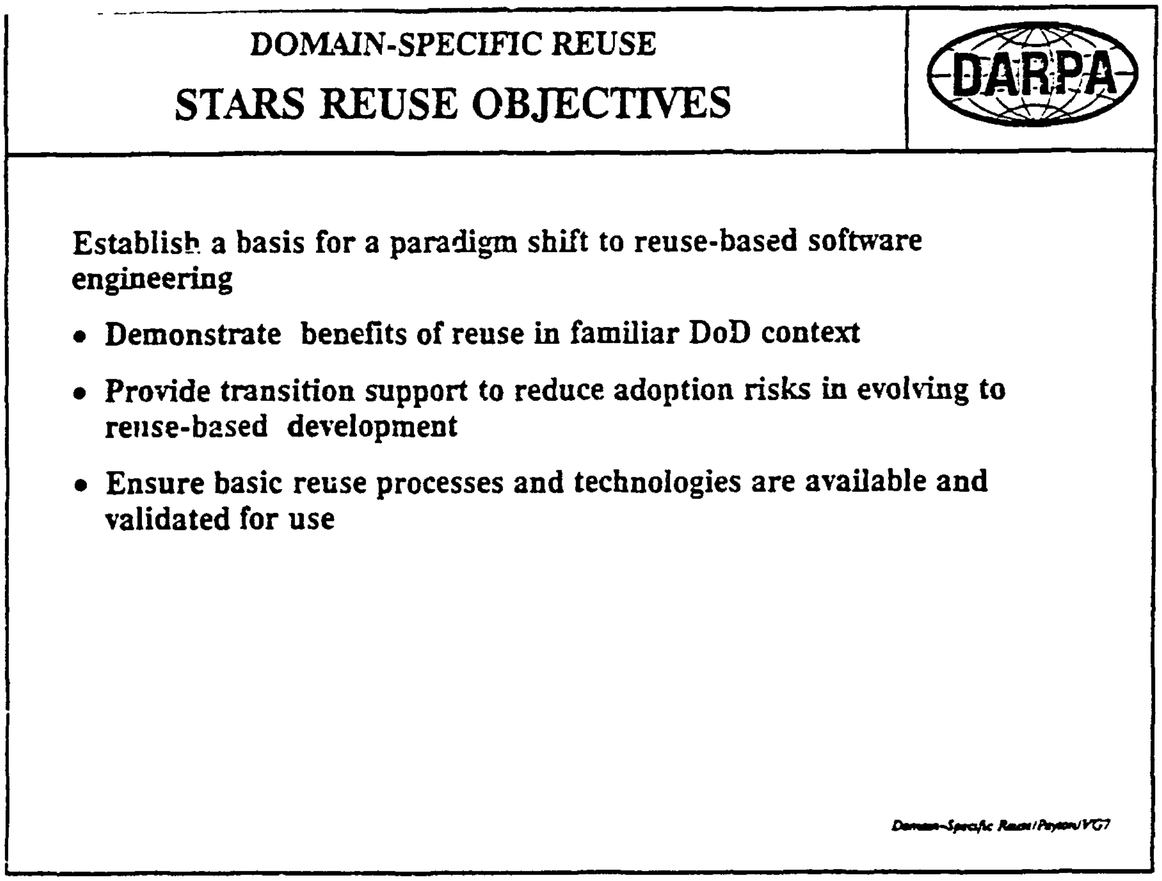
Organizations are beginning to construct libraries that contain multiple interrelated assets (designs, code, tests, documentation). This is the repeatable level. There begins to be a handoff between the producer of an asset and the consumer of the asset. The assets are gathered in structured libraries.

In the portable or adaptable level, software components are specifically designed to be more portable or usable in more general contexts.

The goal is architecture based reuse where a domain analysis has been performed within the application domain and a degree of consensus has been established on the generic software architecture and interfaces.

Finally, in the systematic reuse stage, we can talk about true engineering of the domain and adapting and generating systems based on the architectures and reusable assets.

Domain-specific reuse will come about through the work of many organizations and programs. STARS role is outlined by the triangle. In establishing our strategy, we could have elected to place all our emphasis at the repeatable level to really try to institutionalize repeatable reuse. Instead we have been following a strategy that cuts across levels. It provides the enabling technology for repeatable reuse (library mechanisms) while also addressing processes and techniques for supporting architecture-based reuse.



DOMAIN-SPECIFIC REUSE

STARS REUSE OBJECTIVES

Establish a basis for a paradigm shift to reuse-based software engineering

- **Demonstrate benefits of reuse in familiar DoD context**
- **Provide transition support to reduce adoption risks in evolving to reuse-based development**
- **Ensure basic reuse processes and technologies are available and validated for use**

Domain-Specific Reuse/Payton/VG7

STARS seeks to establish a basis to enable a paradigm shift to reuse-based software engineering The basis will be expanded over time by other programs as the community moves forward with megaprogramming. STARS is working to accelerate the movement forward. To do this STARS is focusing on 3 main reuse objectives.

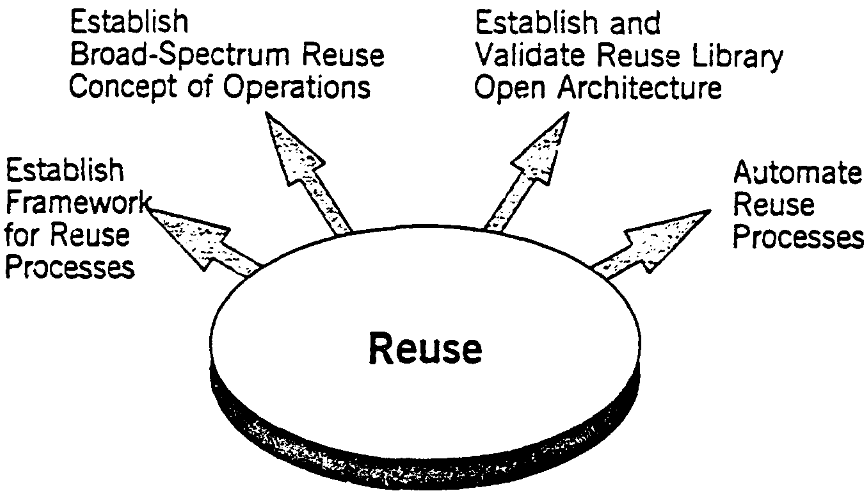
STARS will demonstrate the benefits of domain specific reuse. This will help the community understand reuse, the investment costs and the benefits to be gained. It will thus help motivate others to invest in architecture-based reuse.

STARS understands that there are both technical and non-technical barriers to reuse. STARS will provide transition support such as guidelines and migration paths that should make it easier for others to introduce reuse into their organization's way of doing business.

STARS will work within the community to ensure there are well defined reuse processes and basic technology (tools) to support reuse-based software engineering. STARS is developing some of the processes and tools and working with others to leverage their developments.

DOMAIN-SPECIFIC REUSE

STARS ACTIVITIES



Establish Broad-Spectrum Reuse Concept of Operations

Establish and Validate Reuse Library Open Architecture

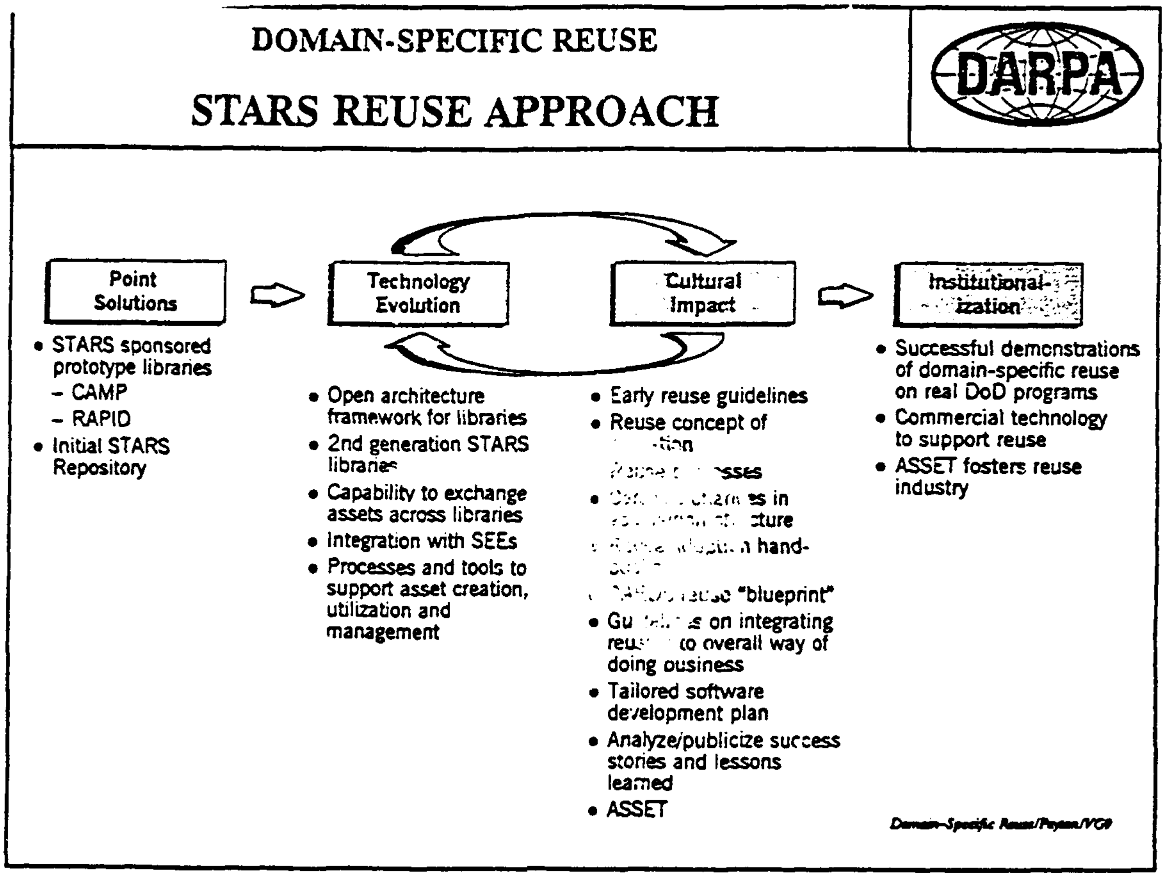
Establish Framework for Reuse Processes

Automate Reuse Processes

Reuse

Domain-Specific Reuse/Payton/VG8

Key STARS reuse activities are depicted here. STARS is establishing a framework for understanding reuse and reuse processes and for understanding where standards or common interfaces would facilitate reuse. STARS has been working on several specific reuse processes, e.g., a domain analysis process and asset certification process. STARS has also b‿en investigating how to tailor life-cycle processes to the needs of specific application domains and how to include r‿use into a life-cycle process. Examples of the reuse process work will be discussed later on today. STARS is interested in automating reuse processes. To date, most of the automation work has focused on developing reuse library mechanisms. A presentation of these will be given tomorrow and demonstrations are available on the demo floor. As reuse processes evolve, STARS will investigate other aspects of automation.



DOMAIN-SPECIFIC REUSE

STARS REUSE APPROACH

Point Solutions

- STARS sponsored prototype libraries
  - CAMP
  - RAPID
- Initial STARS Repository

Technology Evolution

- Open architecture framework for libraries
- 2nd generation STARS libraries
- Capability to exchange assets across libraries
- Integration with SEEs
- Processes and tools to support asset creation, utilization and management

Cultural Impact

- Early reuse guidelines
- Reuse concept of operation
  Reuse processes
- [illegible] changes in acquisition structure
- Reuse adoption hand-book
- STARS reuse "blueprint"
- Guidelines on integrating reuse into overall way of doing business
- Tailored software development plan
- Analyze/publicize success stories and lessons learned
- ASSET

Institutional-ization

- Successful demonstrations of domain-specific reuse on real DoD programs
- Commercial technology to support reuse
- ASSET fosters reuse industry

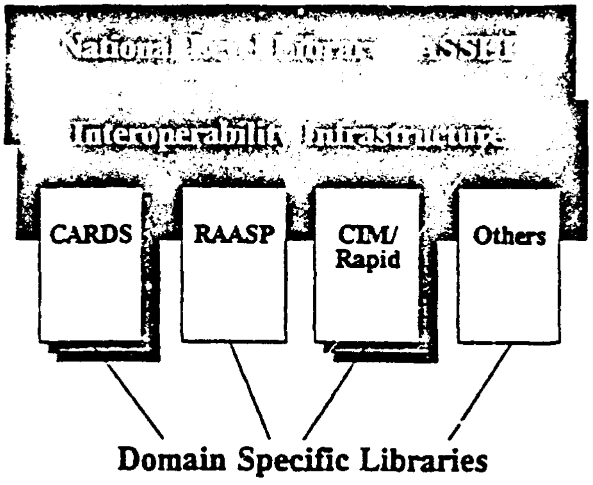
Domain-Specific Reuse/Payton/VG9

The STARS approach to accelerating the shift to megaprogramming involves evolving from adhoc point solutions to a new way of doing business. The slide depicts the overall STARS strategy in each of the areas.

In the reuse area we are currently on our first iteration between technology evolution and cultural impact. We have created a reuse process framework, developed 2nd generation library mechanisms and begun to get feedback from their usage. We have defined the basis for an asset library open architecture and begun to prototype those interfaces within our library mechanisms. We have sample reuse processes and have recommended changes to the acquisition regulations in order to foster reuse. The Asset Source for Software Engineering Technology (ASSET) has been established as a focal point for reuse to help stimulate a national reuse industry.

DOMAIN-SPECIFIC REUSE

REUSE LIBRARY RELATIONSHIPS/VISION


National Level Library ASSET
Interoperability Infrastructure
CARDS
RAASP
CIM/ Rapid
Others
Domain Specific Libraries

- National Level
  - "Yellow Pages"/Inter-library service
  - Process assets
  - Multi-domain components/algorithms such as Ada Bindings
- Inte-operability Infrastructure
  - Network evolution, interconnectivity
  - Open architecture definition
- Domain Specific Libraries
  - Assets particular to application areas or companies (may contain common assets)

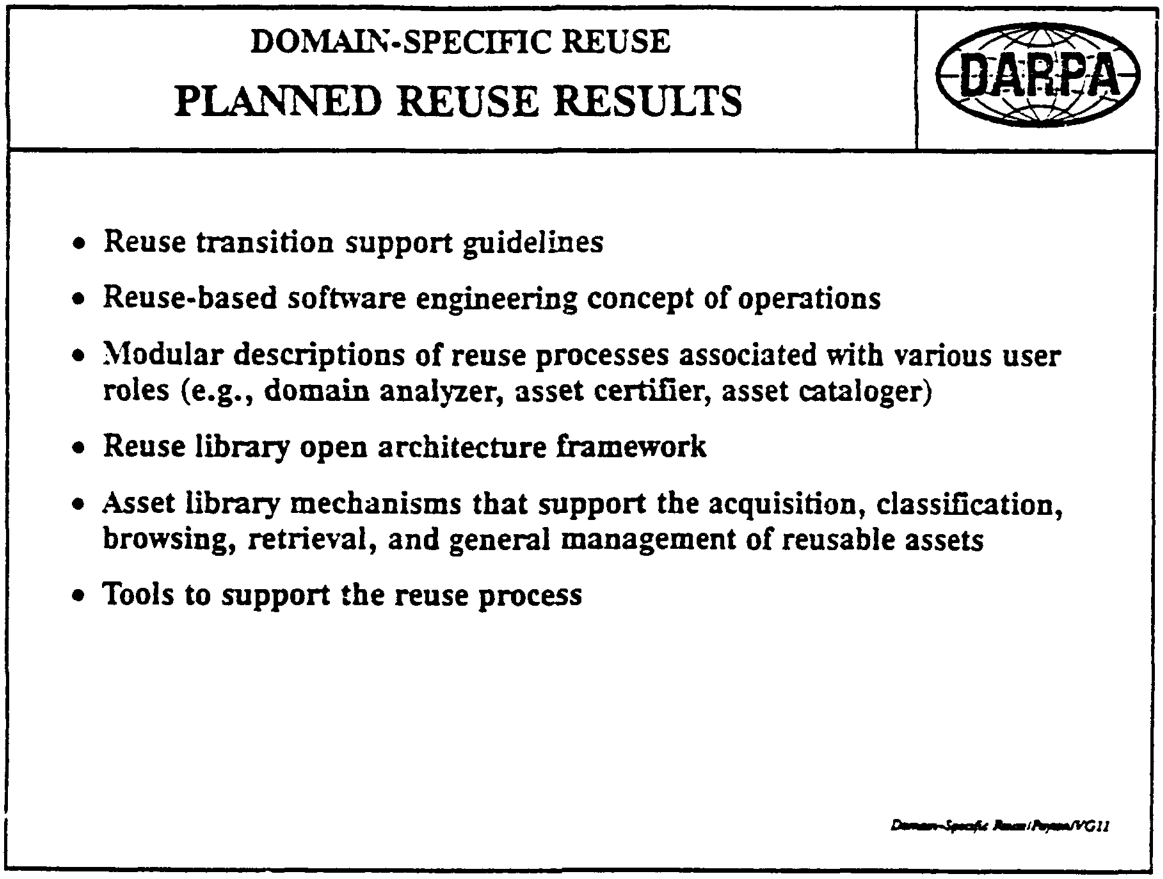
Domain-Specific Reuse/Payton/VG10

STARS envisions that the future will involve a distributed network of interoperating reuse libraries.

There will be multiple libraries using different underlying technology and access schemes dependent on user needs and preferences. Projects, organizations, application domains etc. may each have their own library.

There are some issues that need to be addressed at a national level. ASSET was established to address the national level issues and will provide a "yellow pages" across multiple geographically distributed libraries.

There are other interoperability issues on which consensus within the community is necessary. STARS is seeking to understand where common interfaces are needed and would assist the evolution of a reuse industry. To support interoperability among reuse libraries STARS has begun work on an Asset Library Open Architecture Framework which will be discussed tomorrow. A demonstration of early capabilities for asset interchange is available in the demo area. STARS has helped to establish the Reuse library Interoperability Group (RIG)—an independent pre-standards group of over 25 organizations—that is working towards consensus on reuse library interoperability issues.



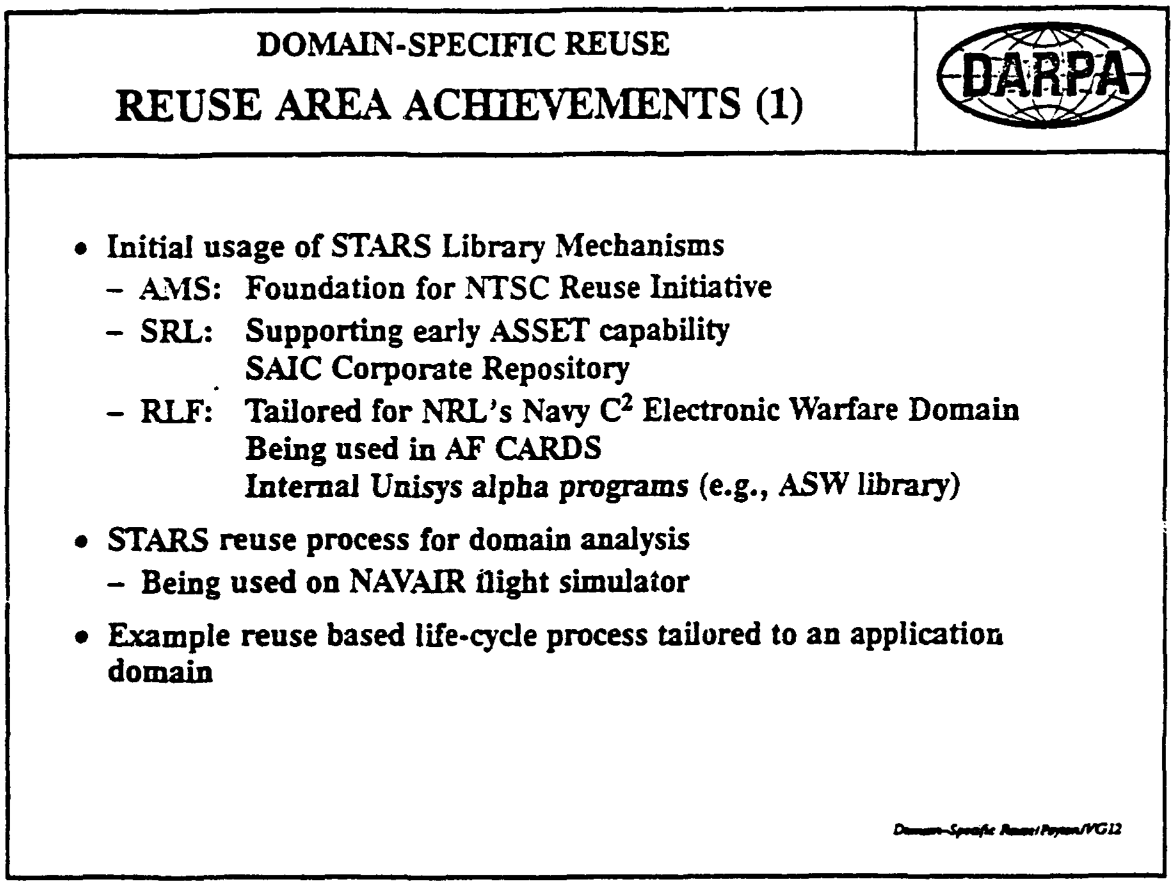
DOMAIN-SPECIFIC REUSE

PLANNED REUSE RESULTS

- Reuse transition support guidelines
- Reuse-based software engineering concept of operations
- Modular descriptions of reuse processes associated with various user roles (e.g., domain analyzer, asset certifier, asset cataloger)
- Reuse library open architecture framework
- Asset library mechanisms that support the acquisition, classification, browsing, retrieval, and general management of reusable assets
- Tools to support the reuse process

Domain-Specific Reuse/Payton/VG11

The next two slides identify some of the key achievements in the reuse area. We have emphasized usage and feedback of interim work rather than identifying particular interim products.



DOMAIN-SPECIFIC REUSE

REUSE AREA ACHIEVEMENTS (1)

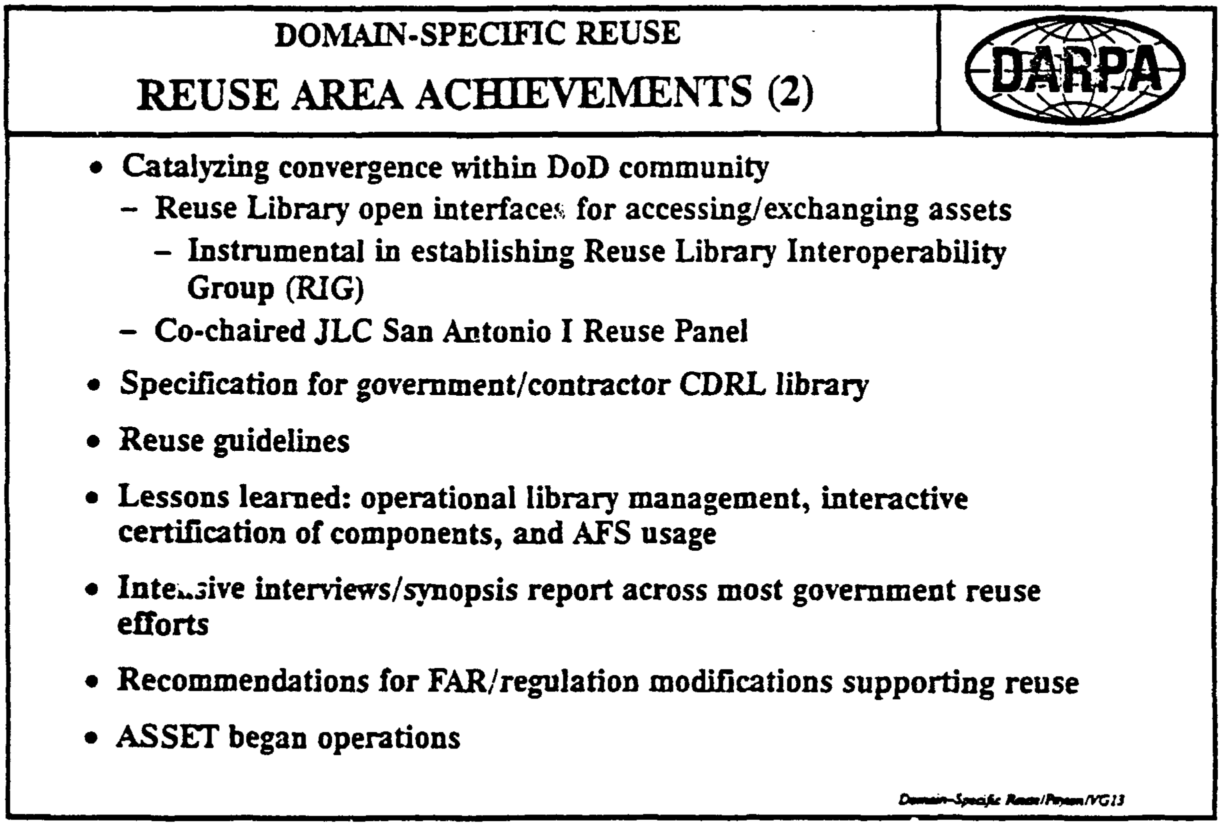
- Initial usage of STARS Library Mechanisms
  - AMS: Foundation for NTSC Reuse Initiative
  - SRL: Supporting early ASSET capability
    SAIC Corporate Repository
  - RLF: Tailored for NRL's Navy C^2 Electronic Warfare Domain
    Being used in AF CARDS
    Internal Unisys alpha programs (e.g., ASW library)
- STARS reuse process for domain analysis
  - Being used on NAVAIR flight simulator
- Example reuse based life-cycle process tailored to an application domain

Domain-Specific Reuse/Payton/VG12

STARS is interested in working within the software engineering community to catalyse a transition to reuse based software engineering. STARS was instrumental in establishing the RIG and continues to actively participate in RIG. It is our intent that the STARS libraries be upgraded from compliance with our own asset library open architecture to compliance with RIG pre-standards as they become available.

STARS staff participated in the JLC San Antonio I reuse panel to identify barriers and actionable recommendations for DoD to make reuse a reality. STARS continues to work with the Army CECOM in supporting the JLC's efforts in this area.

The other achievements identified on this slide represent early STARS work in providing transition support and addressing the cultural issues involved in moving to reuse-based development.



DOMAIN-SPECIFIC REUSE

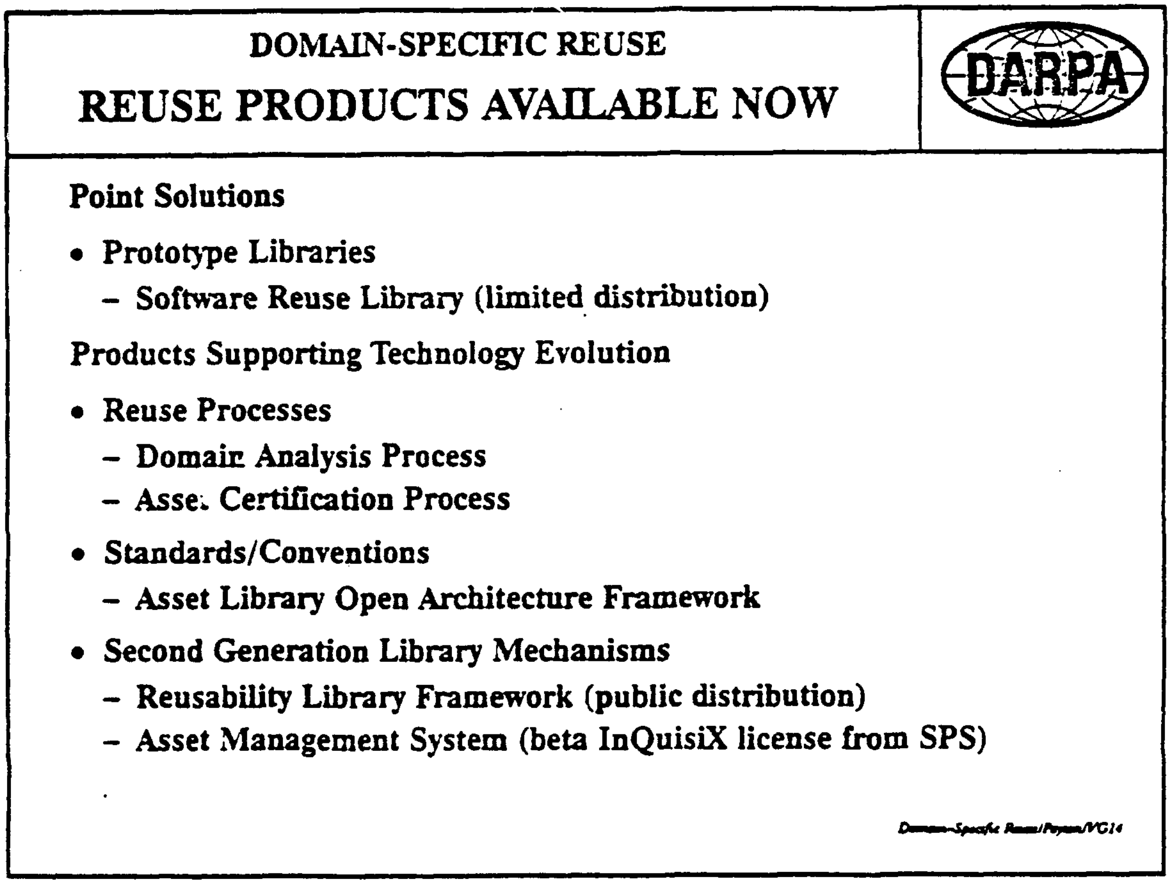
REUSE AREA ACHIEVEMENTS (2)

DARPA

- Catalyzing convergence within DoD community
  - Reuse Library open interfaces for accessing/exchanging assets
    - Instrumental in establishing Reuse Library Interoperability Group (RIG)
  - Co-chaired JLC San Antonio I Reuse Panel
- Specification for government/contractor CDRL library
- Reuse guidelines
- Lessons learned: operational library management, interactive certification of components, and AFS usage
- Inte..3ive interviews/synopsis report across most government reuse efforts
- Recommendations for FAR/regulation modifications supporting reuse
- ASSET began operations

Domain-Specific Reuse/Payton/VG13

A high level view of final STARS reuse products is provided by this slide. Interim versions of the products will receive trial use within alpha-test projects by the Primes, in the demonstration projects, in the AF CARDS program and by Technology Transition Affiliates. The final products will reflect the feedback from this usage. The STARS reuse products address both an evolution of the technology base and transition support guidelines that help address the cultural issues involved in moving towards a new 'vay of doing business.



DOMAIN-SPECIFIC REUSE

REUSE PRODUCTS AVAILABLE NOW

Point Solutions

- Prototype Libraries
  - Software Reuse Library (limited distribution)

Products Supporting Technology Evolution

- Reuse Processes
  - Domain Analysis Process
  - Asse: Certification Process
- Standards/Conventions
  - Asset Library Open Architecture Framework
- Second Generation Library Mechanisms
  - Reusability Library Framework (public distribution)
  - Asset Management System (beta InQuisiX license from SPS)

Domain-Specific Reuse/Payton/VG14

The interim products listed on these two slides are organized according to the stage of the STARS approach the product supports. Thus they are characterized as: point solutions, products supporting technology evolution and products supporting cultural change.

Further information on the processes, tools and reports listed on these slides can be found in the STARS Catalog or the remainder of the presentations at STARS '91. Many of the documents identified on the next two slides will be handed out to you today. All of the library mechanisms are being demonstrated in the STARS booth. We invite you to join with us in accelerating the transition to megaprogramming by working as Technology Transition Affilitates and providing us feedback on these early products.

DOMAIN-SPECIFIC REUSE

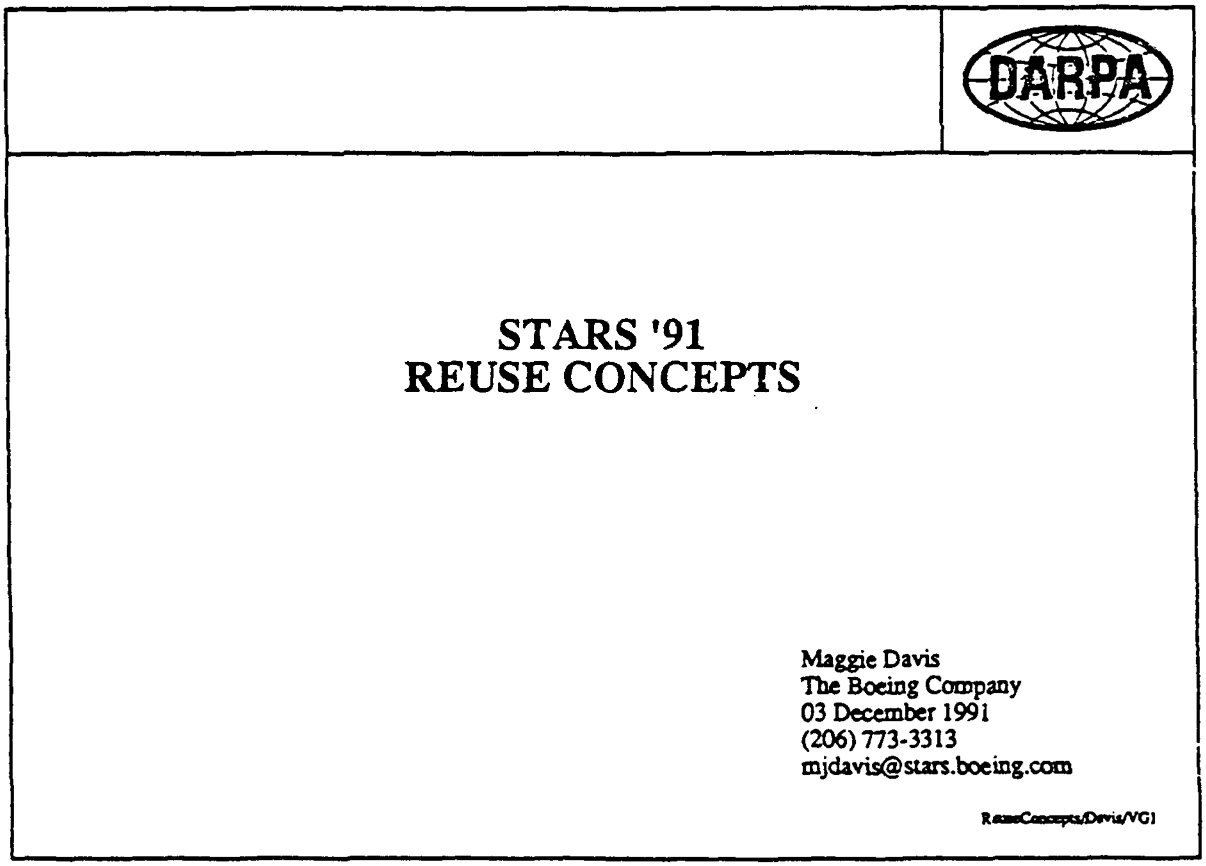
REUSE PRODUCTS AVAILABLE NOW

DARPA

Products Supporting Cultural Change

- Transition guidelines
  - Reuse Concept of Operations
  - Composite Process Model integrating reuse
  - Sample process tailoring to application domain risks
  - Reusable Software Acquisition Environment report

Domain-Specific Reuse/Payton/VG15



STARS '91
REUSE CONCEPTS

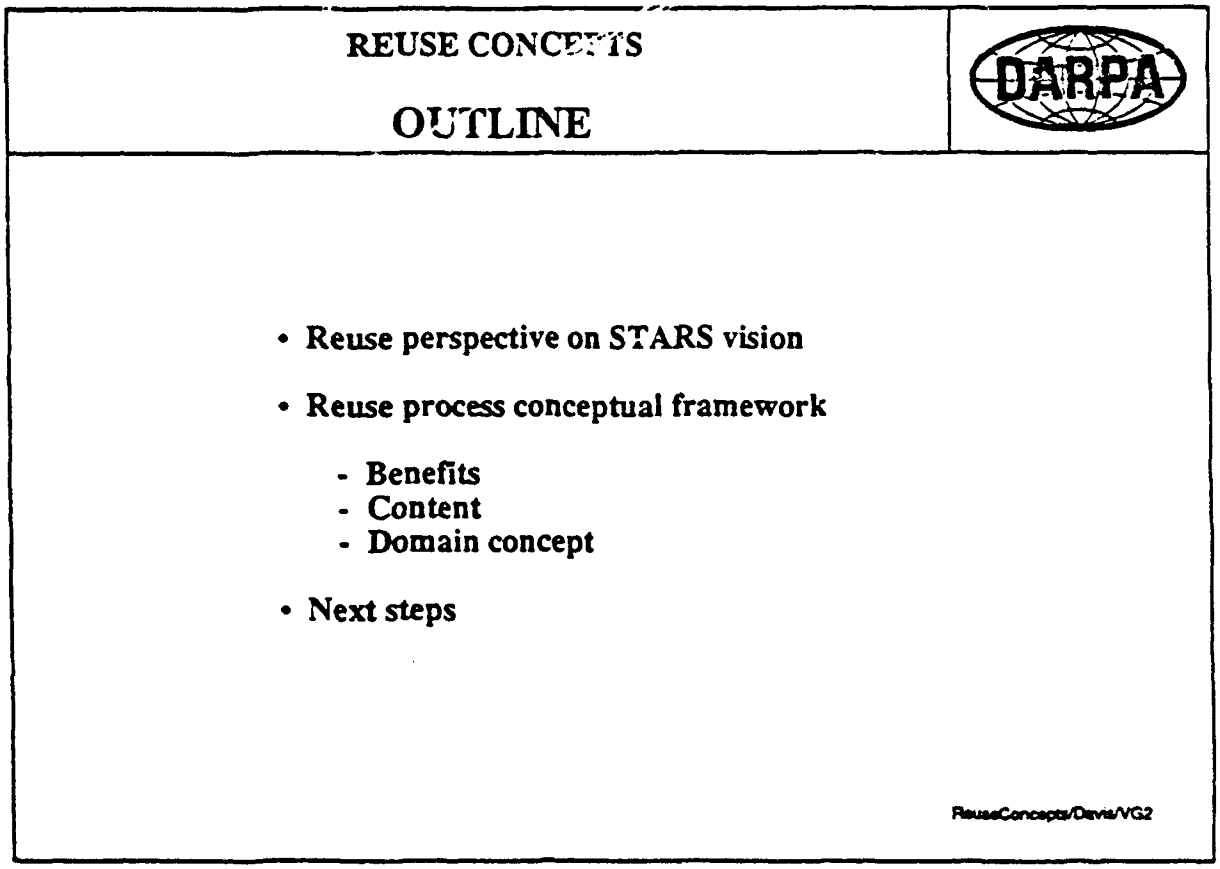
Maggie Davis
The Boeing Company
03 December 1991
(206) 773-3313
mjdavis@stars.boeing.com

ReuseConcepts/Davis/VG1

In support of the ST/\RS mission, representatives from each of the three prime contractors (Boeing, IBM, and Unisys) along with representatives from Mitre and the SEI were chartered as a reuse concepts joint activity team. The team has been developing a consensus description of the reuse aspects of software engineering following the envisioned STARS paradigm.

This description is captured in an evolving document called the STARS Reuse Concept of Operations. The focus of version 0.5, a recently-released first draft, is a framework for considering and defining reuse supporting composable processes.

The framework supports composing these processes into broader contexts such as reuse-based organizational strategies and product life cycles.



REUSE CONCEPTS

OUTLINE

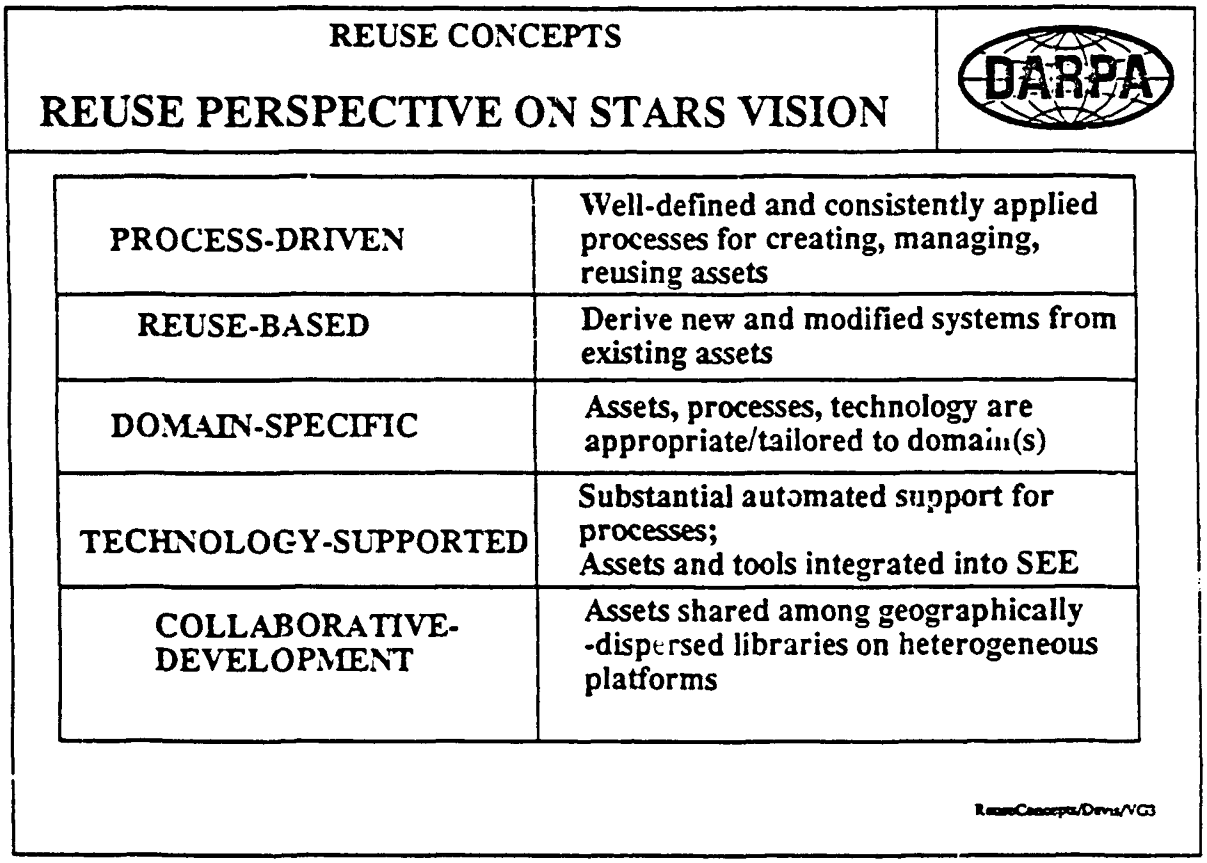
- Reuse perspective on STARS vision
- Reuse process conceptual framework
  - Benefits
  - Content
  - Domain concept
- Next steps

ReuseConcepts/Davis/VG2

This talk gives an overview of the reuse perspective on the STARS vision with regard to specific terms of the vision statement and with regard to the reuse concepts document.

This talk will describe assumptions and benefits of the reuse process conceptual framework as well as describe what the framework contains.

The final topic will address what we believe are the logical next steps in evolving and refining reuse concepts for STARS.



REUSE CONCEPTS

REUSE PERSPECTIVE ON STARS VISION

PROCESS-DRIVEN	Well-defined and consistently applied processes for creating, managing, reusing assets
REUSE-BASED	Derive new and modified systems from existing assets
DOMAIN-SPECIFIC	Assets, processes, technology are appropriate/tailored to domain(s)
TECHNOLOGY-SUPPORTED	Substantial automated support for processes; Assets and tools integrated into SEE
COLLABORATIVE-DEVELOPMENT	Assets shared among geographically -dispersed libraries on heterogeneous platforms

ReuseConcepts/Davis/VG3

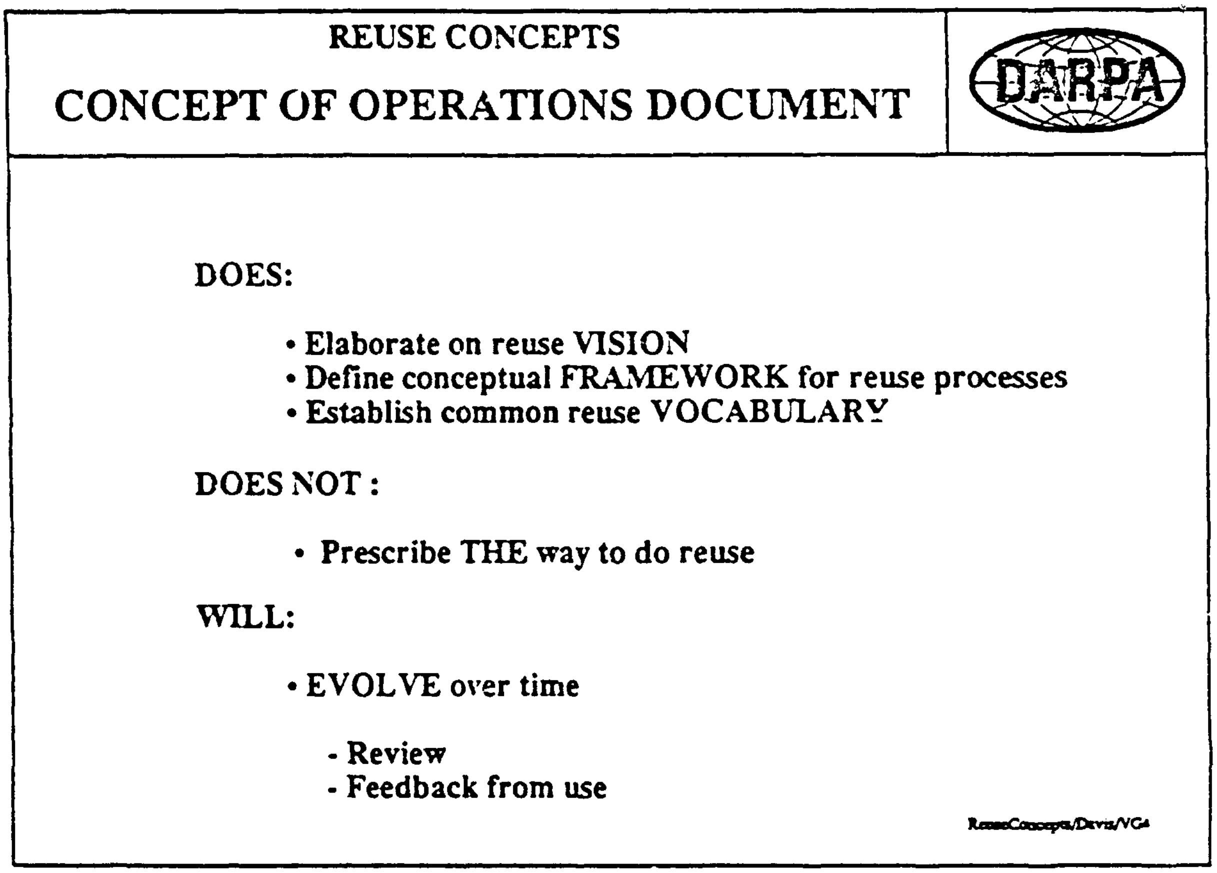
Being PROCESS-DRIVEN means that software engineering is done in accordance with well defined processes that are consistently applied. Support for guidance, monitoring, and definition of processes is provided by the software engineering environment.

Being REUSE-BASED means that the standard approach to software-intensive system development and evolution is to derive new and modified systems principally from existing assets rather than to create them anew. Note, this approach requires that relevant assets be available, as well as processes defining how to use the assets to produce systems. The reusable assets assumed to be available include not only the software components most commonly associated with reuse but also additional kinds of information such as requirements, specifications, architectures, designs, test procedures, domain knowledge models, data dictionaries, algorithms, process definitions, and rationale.

Being DOMAIN-SPECIFIC means that the reusable assets, the development processes, and the supporting technology are appropriate to, perhaps tailored for, the domain in which the software is being developed. We believe that the same reuse concepts and the same generic processes and technology apply to domains of various types and levels.

Being TECHNOLOGY SUPPORTED means that there is substantial automated support for the reuse processes. Further, the reusable assets and the support tools are integrated in the software engineering environment being used.

Doing COLLABORATIVE DEVELOPMENT means that reusable assets can be shared among libraries that are geographically distributed and hosted on heterogeneous platforms. The vision is that a user can use a single interface to interact with all libraries, unaware of whether or not an asset comes from a local or remote library and of the particulars of the user interface or of the data model associated with the originating library.



REUSE CONCEPTS

CONCEPT OF OPERATIONS DOCUMENT

DARPA

DOES:

- Elaborate on reuse VISION
- Define conceptual FRAMEWORK for reuse processes
- Establish common reuse VOCABULARY

DOES NOT :

- Prescribe THE way to do reuse

WILL:

- EVOLVE over time
  - Review
  - Feedback from use

ReuseConcepts/Davis/VG4

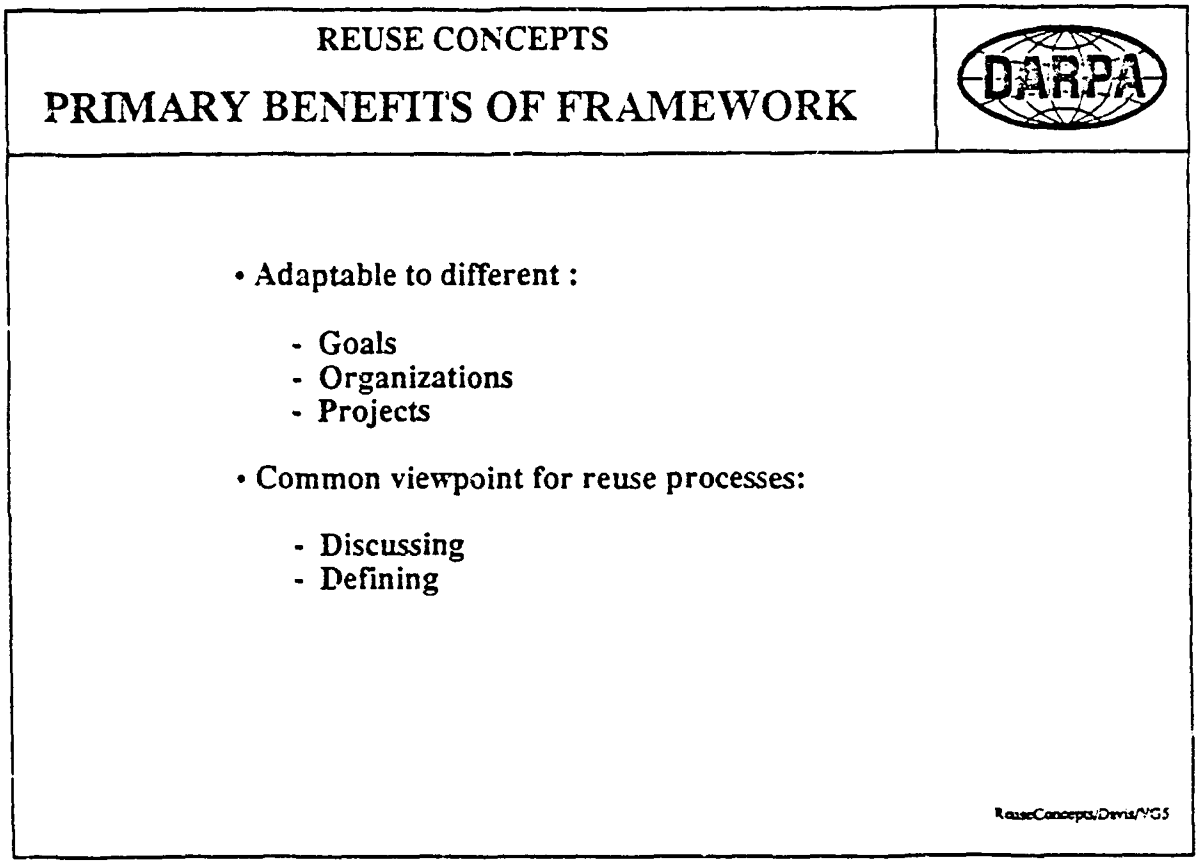
The STARS reuse concept of operations document is the first step in providing guidance on how to evolve reuse-based approaches and in making sure that appropriate reuse support capabilities are known. Thus, the document articulates STARS concepts and expectations for reuse with respect to system and software development by:

– elaborating on the STARS reuse vision;
– defining a framework for definition of reuse processes;
– establishing a common STARS terminology for reuse;
– addressing the impact and opportunities for use of distributed, heterogeneous asset libraries as a reuse-enabling technology (this topic will be covered in the following session of the Reuse Track); and,
– providing a context for understanding STARS reuse plans and products.

The STARS reuse concepts joint activity team believes that there is no one "right" software development process that is applicable to all organizations, applications, piojects, or methodologies. Thus, the reuse concept o: operations document does NOT:

– provide a concept of operations for a total software development process;
– provide a concept of operations for a specific organization; or
– prescribe "the" way to do reuse.

We expect to release version 1, volume I in January 1992. This new version will reflect technical review by individuals inside and outside of STARS. Furthermore, we expect to continuously evolve this volume as other organizations provide feedback from reading it and from trying to use it as guidance. Volume II, which will contain elaborations on the processes within the reuse process framework will be incrementally released as process descriptions become available. It is our hope that these two volumes will be used by those technologists who create, monitor, administer, and modify systems and software development and maintenance processes.


REUSE CONCEPTS

PRIMARY BENEFITS OF FRAMEWORK

- Adaptable to different :
  - Goals
  - Organizations
  - Projects
- Common viewpoint for reuse processes:
  - Discussing
  - Defining

ReuseConcepts/Davis/VG5

To reinforce our belief that there is not one way to do reuse, the reuse concepts joint activity team explicitly developed the reuse process framework to be generic, and thus, adaptable with respect to its application by specific organizations, within specific methodologies or approaches, or as supported by a specific software engineering environment.

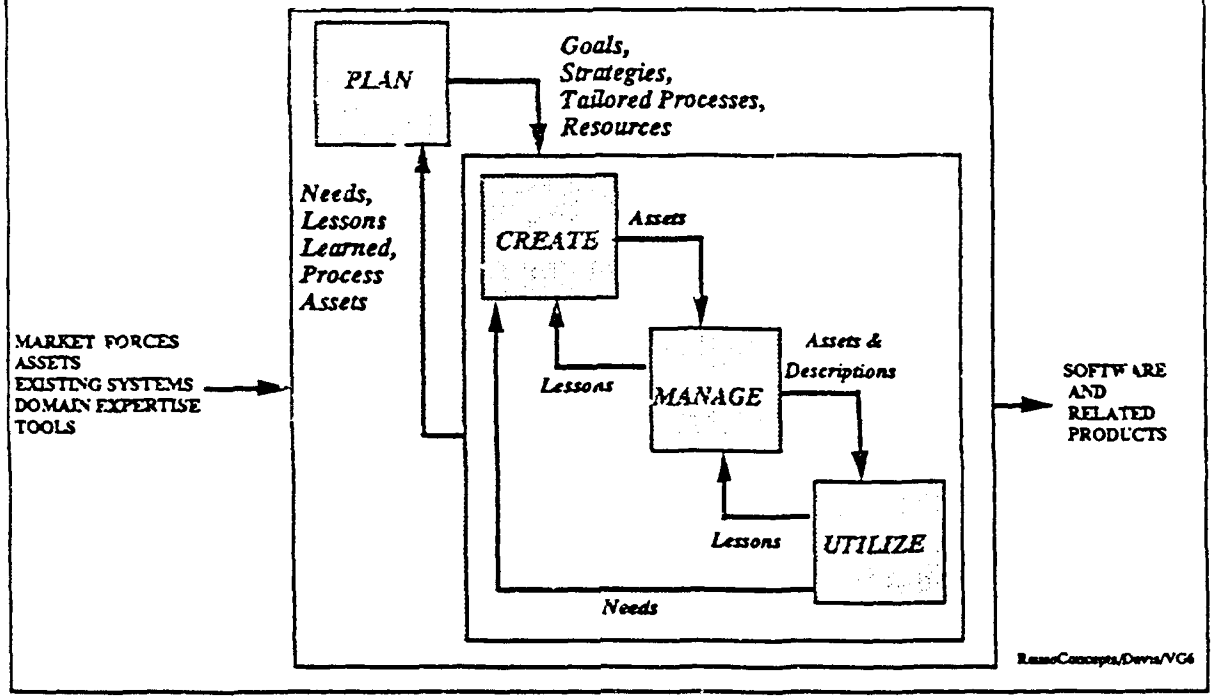
It is our intent that the reuse process framework will aid in understanding the technical issues involved in integrating reuse throughout a system or software life cycle process. This assistance is a consequence of the framework providing a common viewpoint for discussing and defining reuse processes.

We expect that this framework will be of interest to:

- Software Program Managers in understanding how reuse may affect the development process and be incorporated into project planning;
- Acquisition Planners who plan acquisition strategies and prepare request for proposal (RFP) packages;
- Acquisition Policy Makers who are seeking to better understand how to foster reuse; and,
- Process Engineers developing composable reuse processes and merging them into larger process contexts such as life cycle models.

REUSE CONCEPTS

REUSE PROCESS FRAMEWORK



MARKET FORCES
ASSETS
EXISTING SYSTEMS
DOMAIN EXPERTISE
TOOLS

PLAN

Goals,
Strategies,
Tailored Processes,
Resources

Needs,
Lessons
Learned,
Process
Assets

CREATE

Assets

MANAGE

Lessons

Assets &
Descriptions

UTILIZE

Lessons

Needs

SOFTWARE
AND
RELATED
PRODUCTS

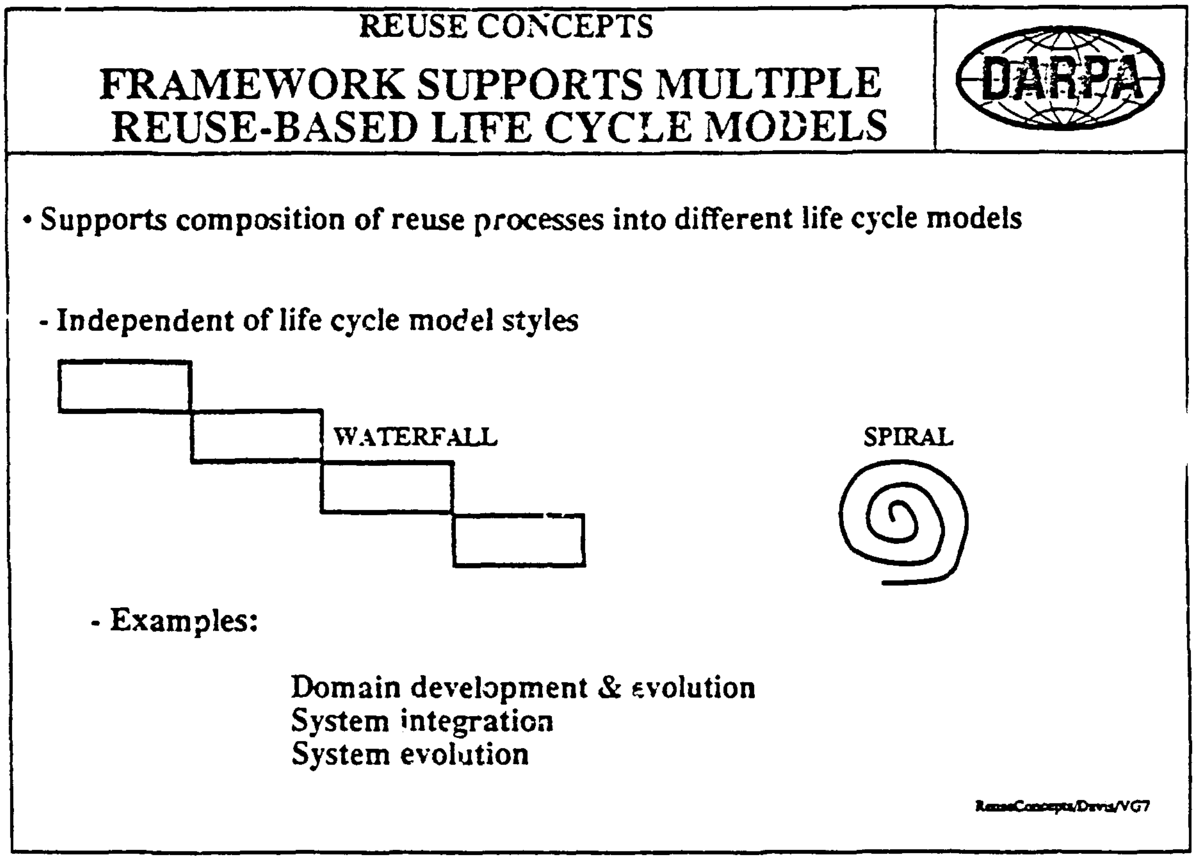
ReuseConcepts/Davis/VG4

The STARS Reuse Process Framework identifies functions and processes supporting reuse in the context of software-intensive system development and maintenance. The framework has been organized into four families of processes, whose names emphasize the primary purpose of each.The arrows in the figure represent the extensive information flow, influence, and feedback among the four process families. In general, the arrows represent the flow of decisions, constraints, experience lessons, and assets.

The families of the reuse process framework can be decomposed further to identify processes and functions focusing on different aspects of each family's purpose. In the viewgraphs following, I will describe the decomposition we used in the document. However, the reuse concepts joint activity team recognizes that individual organizations may use different decompositions of these families to suit their goals and business strategies.

Planning processes set goals and strategies, select and effect the tailoring of processes consistent with the goals and strategies, and identify and allocate existing resources. The asset creation process family produces software and software related assets. The asset management process family evaluates, describes, and organizes the assets provided by the asset creation process family. The asset utilization process family accesses the organized assets to construct software-intensive systems.

Lessons learned regarding the usage, applicability, quality, and reusability of assets are feedback from the asset utilization processes to the asset management processes. Lessons learned regarding missing assets or possible asset generalizations are feedback from the asset utilization processes into the asset creation processes. Lessons learned regarding asset quality and description are feedback from the asset management processes to the asset creation processes. Needs for new assets; lessons learned regarding process usage, applicability, and quality; and new process assets are feedback from the asset creation, asset management, and asset utilization processes into the asset planning processes.



REUSE CONCEPTS

FRAMEWORK SUPPORTS MULTIPLE REUSE-BASED LIFE CYCLE MODELS

- Supports composition of reuse processes into different life cycle models

- Independent of life cycle model styles

WATERFALL
SPIRAL

- Examples:

Domain development & evolution
System integration
System evolution

ReuseConcepts/Davis/VG7

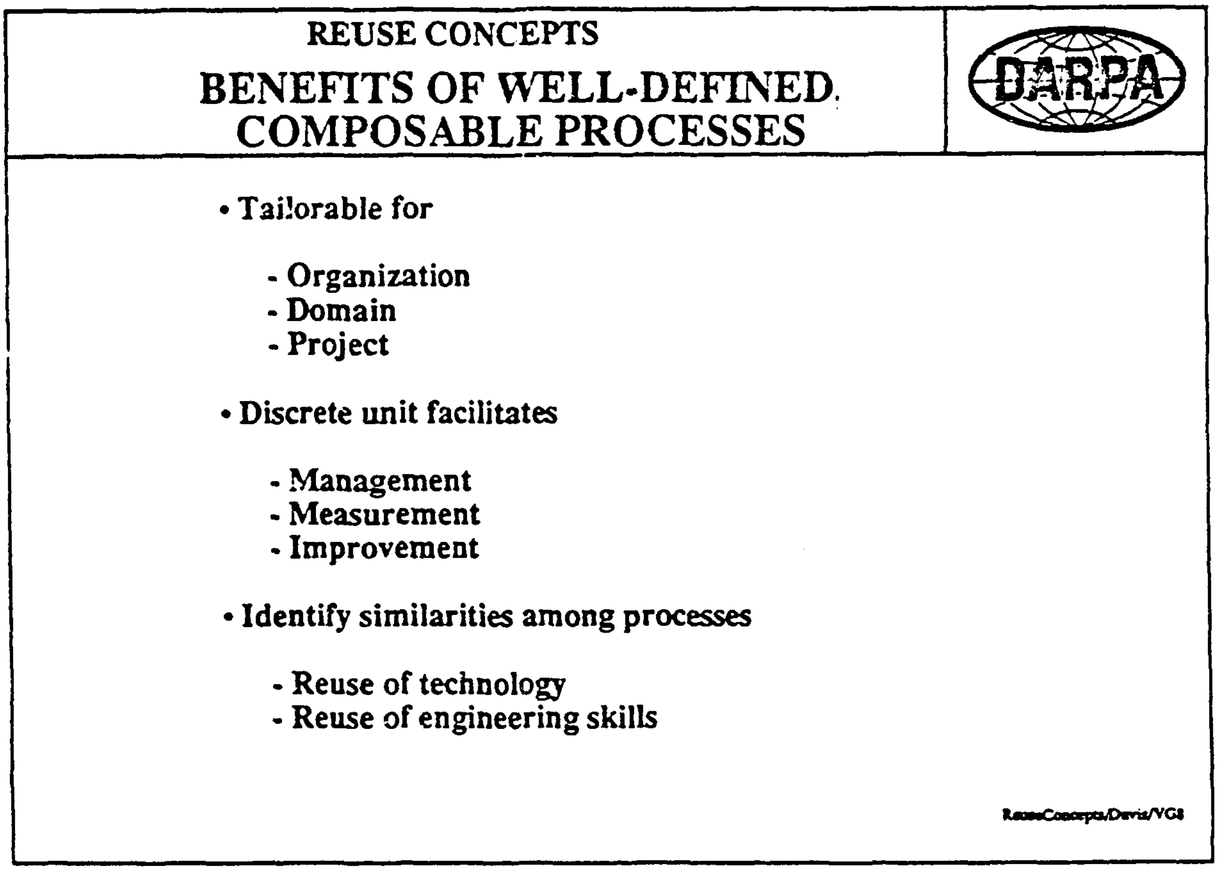
Historically, organizations have based their software development plans on methodology, technique, or tool selections made to implement an idealized project life cycle rather than on composable process selections. Indeed, software development has mostly been considered as one gigantic waterfall life cycle divided into major phases encompassing system conception to demise. In contrast, STARS is promoting the concept that there are multiple, valid modern software life cycle models appropriate for different organizational goals, strategies, and strengths. That is, STARS is generalizing the concept of life cycle model from a strategy for software SYSTEM development to strategies for software PRODUCT development, where product includes components, interface and protocol standards, architectures, domain models, application generators, and systems.

We expect that the reuse process framework will be used to guide composition and instantiation of reuse-based software life cycle models by selecting compatible processes from among its process families. The processes selected should be compatible among themselves, with organizational goals, strategies, and strengths, with project requirements and constraints, and with characteristics of the domain.

Please note that the reuse process framework is also independent of any particular life cycle model style. By style, we mean the model's structure with respect to elapsed time, such as waterfall or spiral. The framework has no pre-defined entry point but it does indicate what information flows among the process families.

Some example reuse-based life cycle models are:

- Domain Development and Evolution, whose goal is production and evolution of reusable assets in a single domain;
- System Integration, whose goal is constructing new, complex software-intensive systems that are integrations of reusable assets from multiple(sub)domains; and,
- System Evolution, whose goal is maintaining the viability of a system as its underlying domain and solution technology mature and evolve.



REUSE CONCEPTS

BENEFITS OF WELL-DEFINED, COMPOSABLE PROCESSES

- Tailorable for
  - Organization
  - Domain
  - Project
- Discrete unit facilitates
  - Management
  - Measurement
  - Improvement
- Identify similarities among processes
  - Reuse of technology
  - Reuse of engineering skills

ReuseConcepts/Davis/VG8

There is one, very basic assumption underlying the reuse process framework. The assumption is that processes can be defined in DISCRETE, WELL-DEFINED units that can be composed into broader contexts. This is the reuse technical area's leverage point with STARS process technical area.

We believe the benefits of this assumption to be:

- Easier implementation and tailoring of life cycle models in support of individual domains, organizations, and engineers.
- Simplified management, measurement, monitoring, and improvement, of life cycle model implementations and improvement in life cycle models.
- Identification of the similarities in appropriate methods, techniques, and tools supporting various life cycle models and processes.
- Identification of similarities among required engineering skills.

These benefits accrue because discrete, composable processes are easier to define, may have formal representations, have definite begin and end points, have definite start and stop criteria, span a shorter time duration than life cycle phases, and can be customized to available tools and environment support.

REUSE CONCEPTS

REUSE PLANNING PROCESS FAMILY

DARPA

ORGANIZATION & PROJECT CHARACTERISTICS

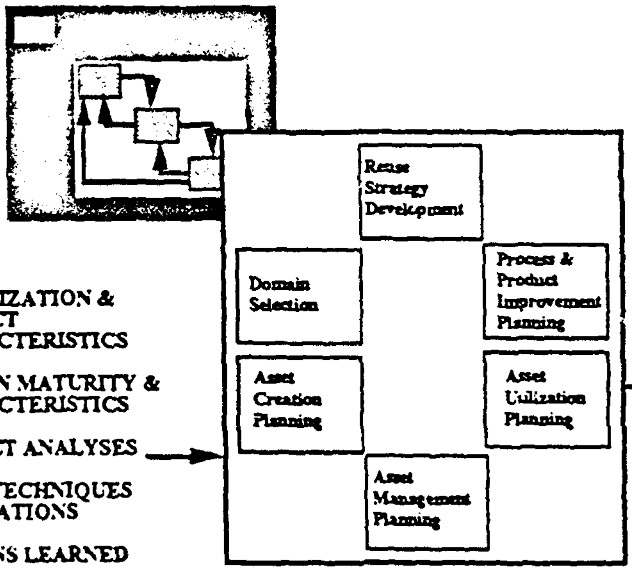
DOMAIN MATURITY & CHARACTERISTICS

MARKET ANALYSES

TOOL/TECHNIQUES EVALUATIONS

LESSONS LEARNED

PROCESS ASSETS


Reuse Strategy Development
Domain Selection
Process & Product Improvement Planning
Asset Creation Planning
Asset Utilization Planning
Asset Management Planning

ORGANIZATION STRATEGIES

PROJECT PLANS

PROCESSES

LIFE CYCLE MODELS

GOALS

ReuseConcepts/Davis/VG9

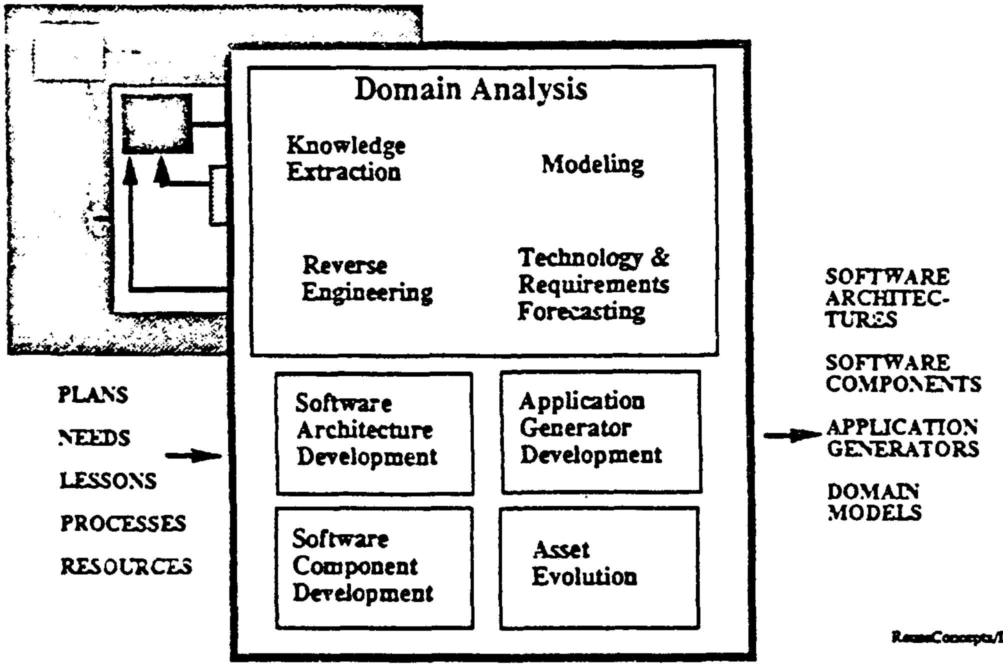
An important function of the planning activity in the reuse process framework is to define a reuse strategy and plan for its implementation within the organization that is undertaking a reuse program. A second function is to implement the strategy in plans and processes for a specific project. A related function is to measure and evolve the process for executing the plans. Note that many of the planning activities and products are appropriate at both the organizational and specific project levels.

Reuse Strategy Development: A reuse strategy is used to guide the asset creation, management, and utilization processes. The activities required to define the strategy will depend on the nature of the organization, e.g., whether it is a company seeking to market reusable components, or develop systems based on them, a DoD Program Executive Officer establishing a reuse program for a given dom[illegible] Program Manager developing a specific system, or a maintenance organization. The strategy will be ini[illegible] y the organization's goals and top level reuse policy. The reuse strategy may define processes that identify, evaluate and select domains for reuse; define a set of methods for asset creation that are compatible with the methods for asset utilization; create plans for asset creation, management, and utilization; and define goals to measure the effectiveness of reuse. A software reuse strategy may include, but is not limited to, a domain selection method, an asset creation plan, an asset management plan, an asset utilization plan, and process and product improvement plans.

Process & Product Improvement planning: The reuse process measurement and evolution function receives input in the form of data captured about the asset creation, management, and utilization processes and products. It also receives lessons learned, asset requirements, process requirements, and any other form of relevant feedback from individuals involved in those processes. Feedback from the users of the software products is also input to this function.

REUSE CONCEPTS

ASSET CREATION PROCESS FAMILY



ReuseConcepts/Devts/VG10

The goal of DOMAIN ANALYSIS is to develop a domain model, reusable requirements, and domain variability description applicable to solution systems within the domain. Note that domain is being used here in its broadest sense, i.e, as an area of activity or knowledge. At a high level, domain analysis is a combination of reverse engineering, knowledge extraction, technology and requirements forecasting, and modeling.

The purpose of SOFTWARE ARCHITECTURE DEVELOPMENT is to produce an architecture that can be used to implement numerous systems for the domain as defined by the domain analysis.

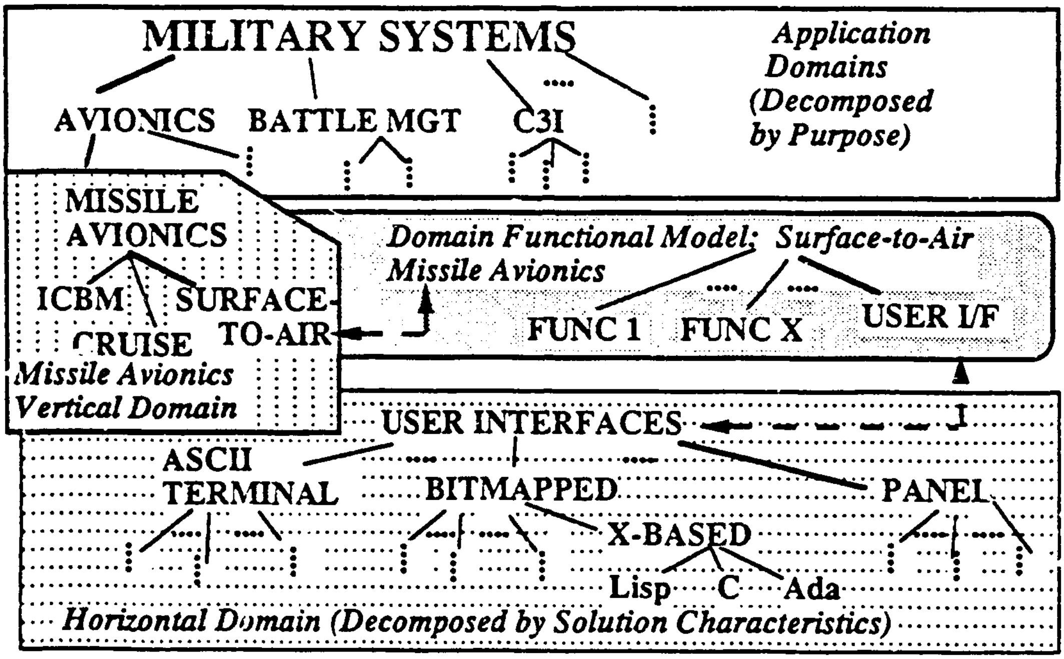
The goal of SOFTWARE COMPONENT DEVELOPMENT is to develop reusable software components that implement the previously developed domain-specific architecture. Before this activity is undertaken, reuse planning has already evaluated whether component development is more appropriate than or complementary to application generator development or use. Reuse planning activities will also have evaluated whether translation of code from legacy systems may also be appropriate.

The goal of APPLICATION GENERATOR DEVELOPMENT is to provide a capability that allows a reuser or application developer to create software (sub)systems using the concepts and terms belonging to the domain. The point is to support the end user in stating "what" is desired rather than detailing "how" the desired effect is to be achieved. This "what" orientation can also be termed requirements-based.

The goal of ASSET EVOLUTION is to respond to the feedback of asset evaluations from the asset management and asset utilization processes. There should be explicit processes that receive and analyze this feedback with the objective to enhance the appropriate domain model, software architecture and components, and application generators. The feedback may also be used to improve or better tailor the processes of modeling, component and architecture creation, and application generator development to the needs of particular domains or organizations.

REUSE CONCEPTS

DOMAIN : AREA OF EXPERTISE



MILITARY SYSTEMS

AVIONICS BATTLE MGT C3I

Application Domains (Decomposed by Purpose)

MISSILE AVIONICS

ICBM SURFACE-TO-AIR CRUISE

Missile Avionics Vertical Domain

Domain Functional Model: Surface-to-Air Missile Avionics

FUNC 1 FUNC X USER I/F

USER INTERFACES

ASCII TERMINAL BITMAPPED PANEL

X-BASED

Lisp C Ada

Horizontal Domain (Decomposed by Solution Characteristics)

ReuseConcepts/Davis/VG11

Domains have been characterized as application, horizontal, or vertical, technology, computer science, execution, execution models, etc.. The figure graphically depicts relationships among some characterizations of domains.

In the figure, application domains represent the knowledge and concepts that pertain to a particular computer application area such as battle management, avionics, C3I, and nuclear physics.

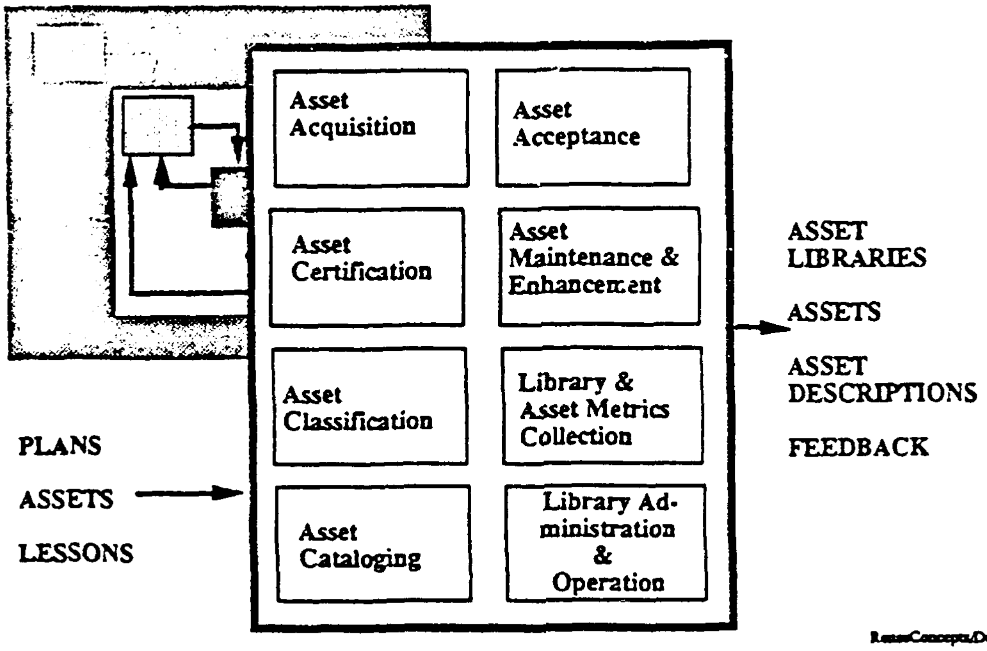
Mid left on the figure is a depiction of a vertical domain. A vertical domain is a representation of the the essential functionality of a restricted set of systems that pertain to a particular member of an application (sub)domain. This figure also attempts to show that a domain model should be related to one branch of a vertical decomposition of an application domain.

At the bottom of the figure, horizontal domains are depicted as the knowledge and concepts that pertain to a particular functionality of a set of software components that can be utilized across more than one application domain. Example horizontal domains include user interfaces, database systems, and statistics. Most horizontal domains can be decomposed into a tree or family of more specialized (sub)domains where the decomposition is guided by characteristics of the solution software. Distinguishing characteristics may be software decomposition style (functional, object-oriented, data-oriented, control-oriented, declarative, etc.), conceptual underpinning (relational, hierarchical data models), or particular requirements for hardware or performance characteristics. These requirement characterizations may be used to relate particular sets of software components to a specific domain model for instantiation in a desired system.

The reuse concepts joint activity team agrees this is very complicated. We have attempted to develop a simpler view, and will continue to do so We feel that reaching a consensus on what we mean by domain contributes to understanding where and how domain modeling and analysis fits into reuse-based development.

REUSE CONCEPTS

ASSET MANAGEMENT PROCESS FAMILY



Asset Acquisition

Asset Acceptance

Asset Certification

Asset Maintenance & Enhancement

Asset Classification

Library & Asset Metrics Collection

Asset Cataloging

Library Ad-ministration & Operation

PLANS
ASSETS
LESSONS

ASSET LIBRARIES
ASSETS
ASSET DESCRIPTIONS
FEEDBACK

ReuseConcepts/Devu/VG12

The goal of asset ACQUISITION is to obtain assets from external asset libraries and other sources in support of asset creation and asset utilization activities.

The goal of asset ACCEPTANCE is to ensure that an asset satisfies all legal and policy constraints and that sufficient information is available to catalog the asset.

The goal of asset CLASSIFICATION is to develop a scheme for categorizing assets on the basis of their domain-relevant characteristics. The classification scheme provides library users with an organizational framework for locating and understanding domain assets.

Asset CATALOGING is broken down into three steps: asset categorization, asset description, and asset installation. Asset CATEGORIZATION is the process of determining where an asset belongs within the classification scheme. Asset DESCRIPTION is the process of creating, capturing, or adapting all the information that is needed to describe the asset in the context of the library's data model, once the asset has been categorized. Asset INSTALLATION is the process of installing the categorized and described asset in the library system.

The ultimate goal of asset CERTIFICATION is to guarantee that software assets implement their requirements and that their execution will be error free in their intended environment.

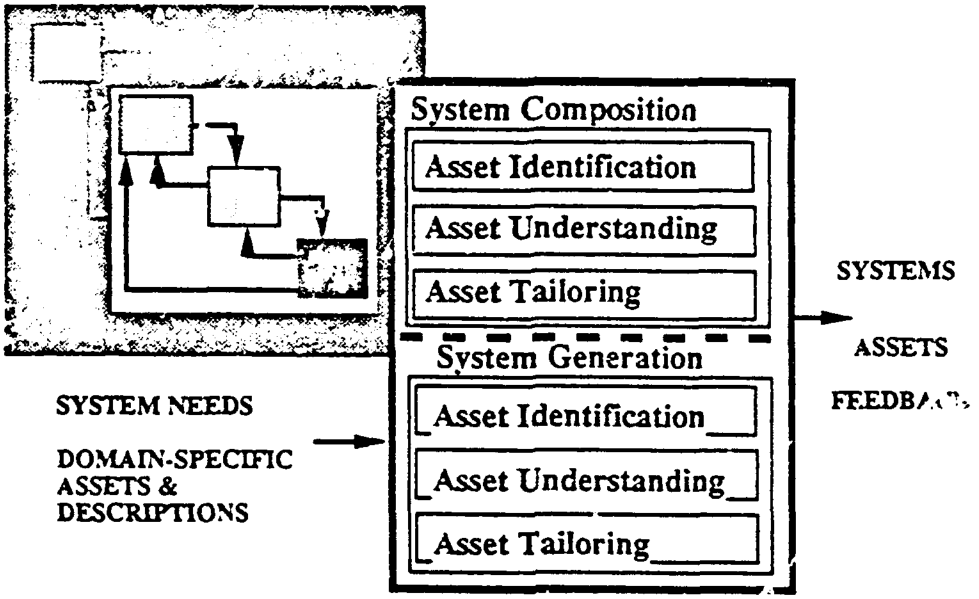
The goal of LIBR.,RY AND ASSET METRICS COLLECTION is to improve the effectiveness of the library in supporting reuse processes within client organizations.

The goal of LIBRARY ADMINISTRATION and operation is to assure the availability of the asset library for asset creation and asset utilization activities.

The goal of the asset MAINTENANCE and enhancement process is to iteratively improve the assets in the library relative to user and domain needs.

REUSE CONCEPTS

ASSET UTILIZATION PROCESS FAMILY



System Composition

Asset Identification

Asset Understanding

Asset Tailoring

System Generation

Asset Identification

Asset Understanding

Asset Tailoring

SYSTEM NEEDS

DOMAIN-SPECIFIC ASSETS & DESCRIPTIONS

SYSTEMS

ASSETS

FEEDBACK

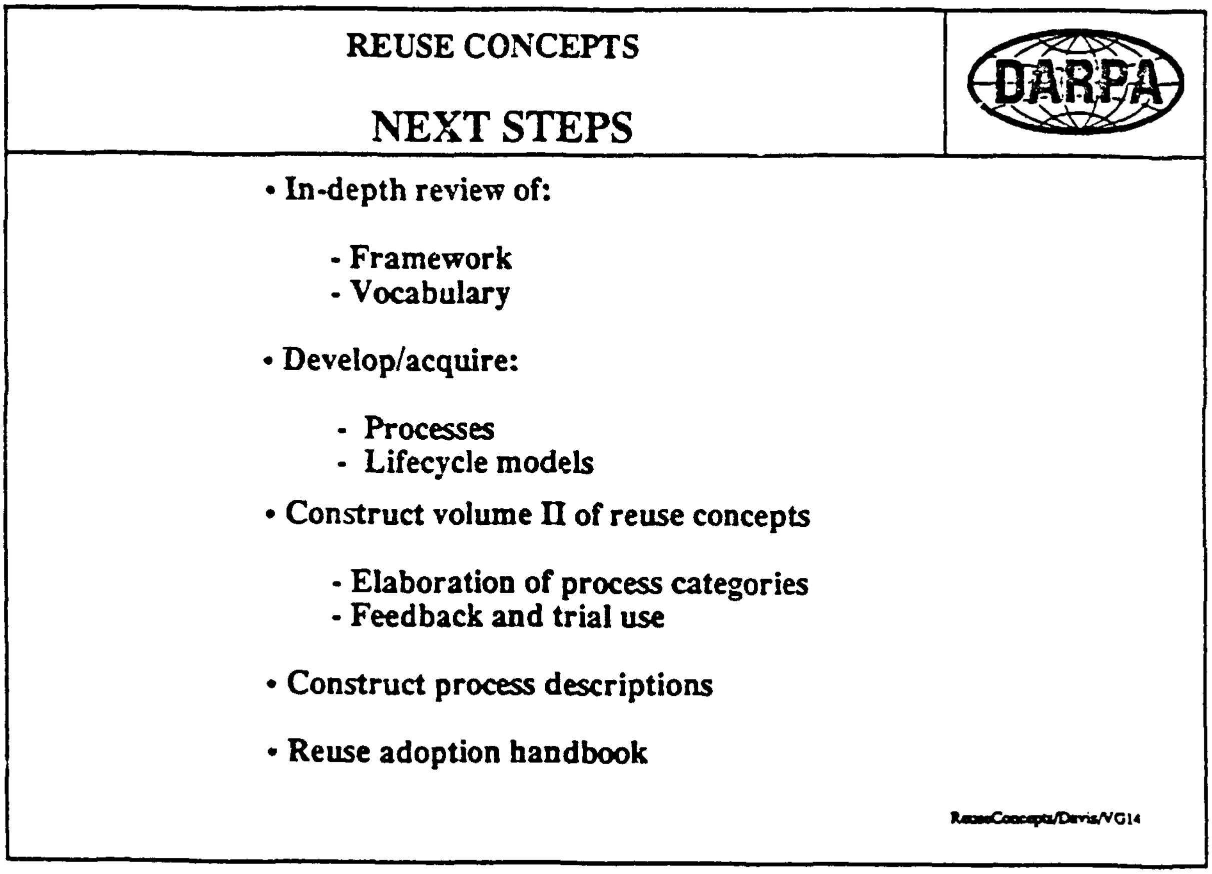
ReuseConcepts/Davis/VG13

There are two primary methods of asset utilization, corresponding to system composition and system generation. These two asset utilization methods are complementary and can both be employed within the same domain or for a single system development. The other processes shown here (asset identification, asset understanding/evaluation/selection, and asset tailoring/integration) each have the same goals but are are approached differently within each utilization method.

Asset-based system COMPOSITION is a process in which the software engineer constructs new products (e.g., requirements, design, code, tests, documentation) from previously developed or newly generated parts. This is typically done by identifying, understanding, evaluating, and selecting appropriate generalized domain assets and tailoring and integrating them to meet specific system needs.

System GENERATION is a process for producing systems or subsystems that ideally incorporates all the variation in a domain into a set of parameters expressed in terms of a specification language or template. A generation tool accepts specifications from engineers that define values for the domain parameters and resolves the variation accordingly to generate components of the target system.

Asset utilization may reveal the need to amend the domain model, to construct new assets, and other or to change or delete other assets. Similarly, each reuse-based development effort should yield lessons that can be applied to asset management within the domain. Engineers' experiences with browsing and querying the library may result in recommendations for refining or correcting aspects of the library taxonomy or asset descriptions; experiences with the tools used to facilitate asset understanding, tailoring, integration, and generation may yield recommendations for additional tools or improvements to the existing tools; problems with assets that were thought to be well-qualified may reveal inadequacies in the asset qualification process; lack of adequate access to the remote libraries may result in recommendations for improved library connectivity or interoperability.



REUSE CONCEPTS

NEXT STEPS

- In-depth review of:
  - Framework
  - Vocabulary
- Develop/acquire:
  - Processes
  - Lifecycle models
- Construct volume II of reuse concepts
  - Elaboration of process categories
  - Feedback and trial use
- Construct process descriptions
- Reuse adoption handbook

ReuseConcepts/Davis/VG14

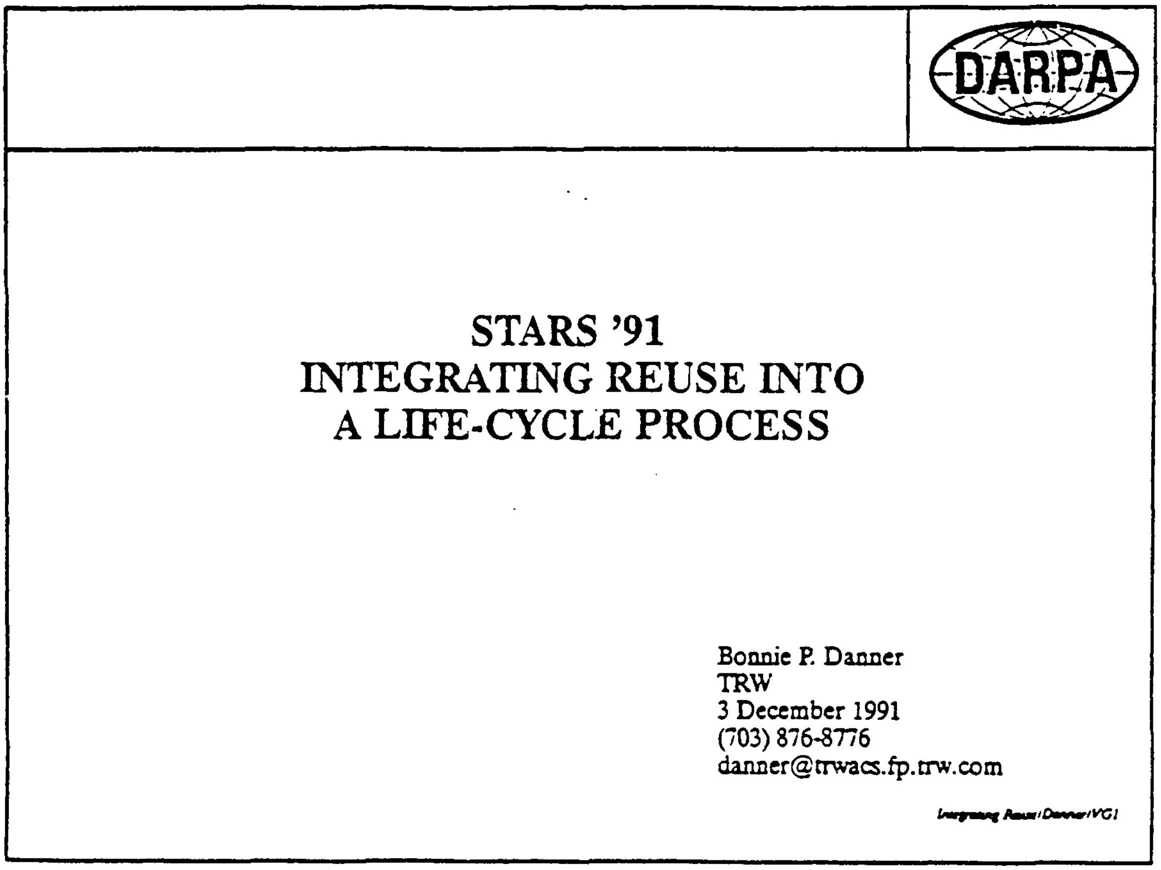
We have reached the final viewgraph in this presentation. We want to tell you where we think we go from here.

We will be improving and refining both the framework and the set of fundamental terms. We invite and would welcome review by you. Version 0.5 of the Reuse Concepts document is available here toaay to facilitate your participation.

We will be developing or acquiring composable process definitions and life cycle models. Contributions from you and your organizations would be most welcome. They may also be made to the process asset library being discussed in the process track.

Our immediate next step is to construct volume II of the reuse concepts document. This volume will elaborate on processes we have identified in the decomposition we used in volume I. It will also elaborate on the considerable flow of information shared among different process families. This is again an opportunity for you to provide us with feedback and recommendations.

Results from DSD Laboratories and the CARDS program will be used in developing a reuse adoption handbook that addresses the non-technical barriers to reuse.



DARPA

STARS '91
INTEGRATING REUSE INTO
A LIFE-CYCLE PROCESS

Bonnie P. Danner
TRW
3 December 1991
(703) 876-8776
danner@trwacs.fp.trw.com

Integrating Reuse/Danner/VG1

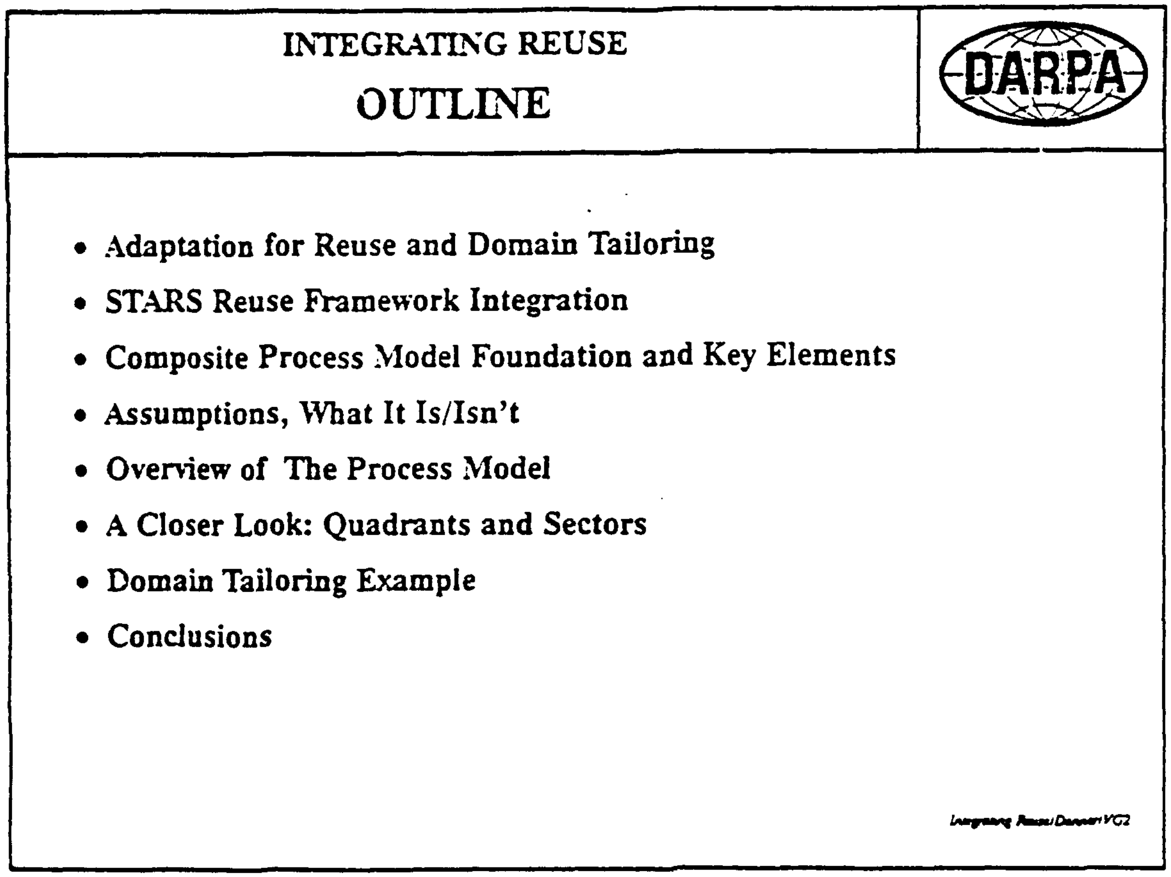
Under STARS tasking, we integrated software development reuse activities into a risk-driven, spiral-based process model. We also initiated the adaptation of the reuse-based process to a specific application domain.

This work is documented in two separate reports:

1) *STARS Subtask US40.2 Composite Paradigm Report for Software Technology for Adaptable Reliable Systems*

2) *US40 - Risk-Reduction Reasoning-Based Development Paradigm Tailored to Navy* C^2 *Systems*

Copies of these reports are here on the table and will be available today after the briefings.



INTEGRATING REUSE

OUTLINE

DARPA

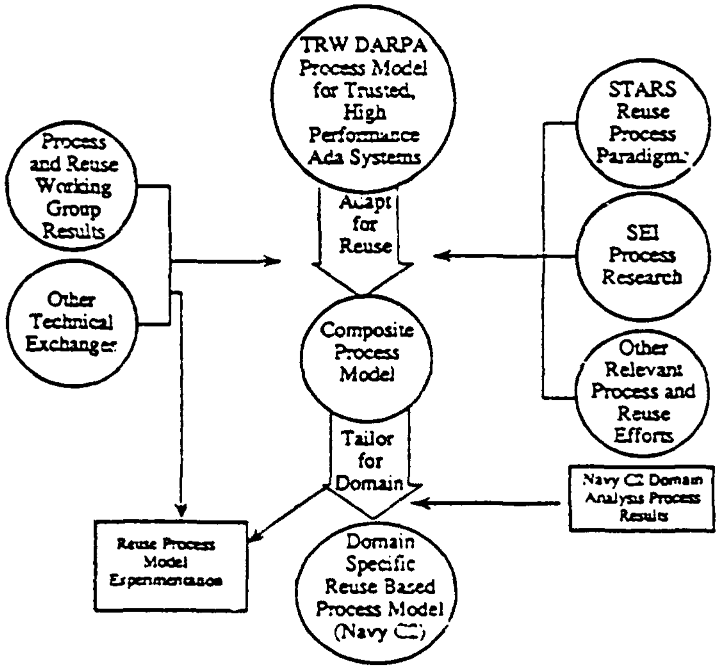
- Adaptation for Reuse and Domain Tailoring
- STARS Reuse Framework Integration
- Composite Process Model Foundation and Key Elements
- Assumptions, What It Is/Isn't
- Overview of The Process Model
- A Closer Look: Quadrants and Sectors
- Domain Tailoring Example
- Conclusions

Integrating Reuse/Denver VG2

This briefing provides an overview of reuse-based process model enhancements and an initial domain tailoring of the life-cycle process descriptions. We will examine the process model foundation and the composition of the integrated process. We will then review the domain tailoring example and discuss conclusions and recommendations.

INTEGRATING REUSE
ADAPTATION FOR REUSE AND DOMAIN TAILORING

DARPA

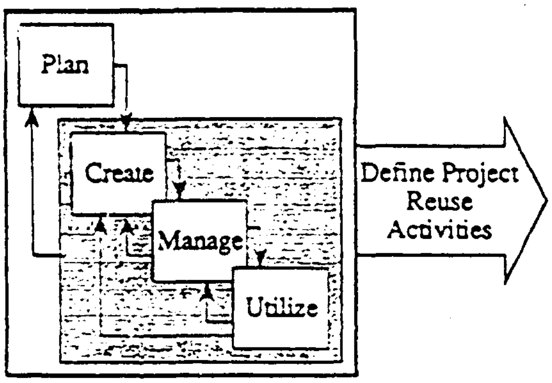

TRW DARPA Process Model for Trusted, High Performance Ada Systems
Process and Reuse Working Group Results
Other Technical Exchanges
STARS Reuse Process Paradigm
SEI Process Research
Other Relevant Process and Reuse Efforts
Adapt for Reuse
Composite Process Model
Tailor for Domain
Navy C2 Domain Analysis Process Results
Reuse Process Model Experimentation
Domain Specific Reuse Based Process Model (Navy C2)

Integrating Reuse/Denver/VG3

For the STARS Program Unisys tasking, we adapted previous TRW process modeling work under the DARPA SISTO project, Advanced Computing Systems (ACS). The Composite Process Model is a risk-driven, reuse-based process model for high assurance software development. The model provides an initial framework for specific process activities and embeds reuse into the spiral based paradigm. Task goals for the Composite Process Model were to integrate STARS reuse processes into a full life-cycle paradigm, to adapt previous DARPA work, to provide a foundation for more detailed reuse process descriptions, and to provide top level guidance for reuse-based process descriptions in a risk-driven paradigm. The model represents an interim step toward broad STARS goals for domain-specific reuse as an element of megaprogramming support. This task illustrates the integration of reuse into a life-cycle process. Current reuse and process efforts and the STARS reuse process framework provide input into major spiral stages of activity. As an interim step, the Composite Process Model is applicable to future domain-specific process modeling. For this tasking, the model has been combined with preliminary Navy C^2 domain analysis work to define a domain specific process model. The focus of the resulting model is domain specific; however, the overall paradigm has more general application as a process model representation. In addition, reuse process model representation experiments were initiated under this tasking.

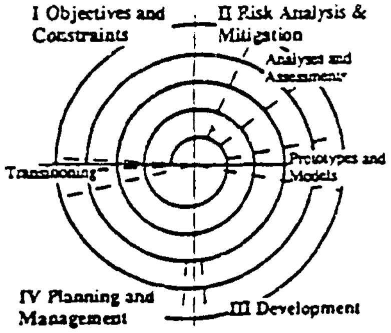
INTEGRATING REUSE
STARS REUSE FRAMEWORK INTEGRATION

DARPA



STARS Reuse Process Framework

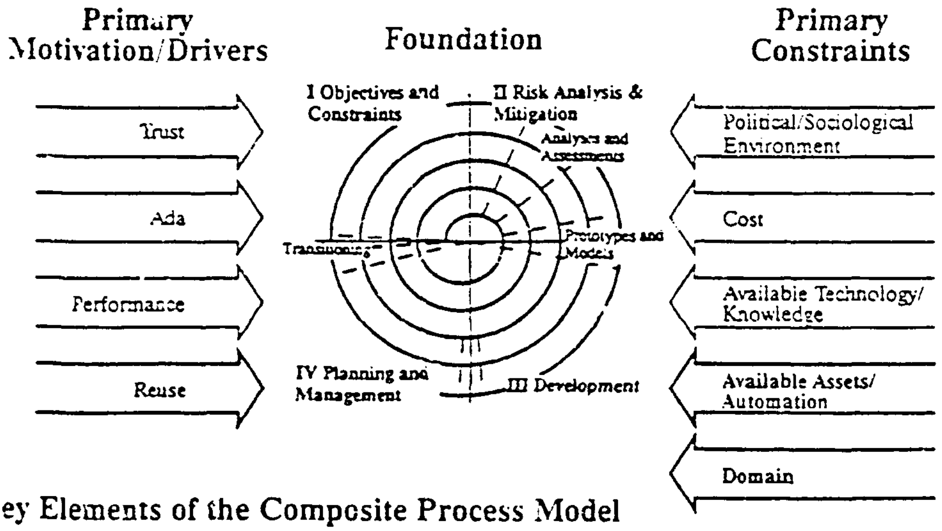
Integrate and Adapt


I Objectives and Constraints
II Risk Analysis & Mitigation
Analyses and Assessments
Prototypes and Models
Transitioning
IV Planning and Management
III Development

Process Model for Trusted High-Performance Ada Systems

The fundamental reuse process activities that make up the STARS reuse process framework were interpreted for project implementation and integrated into the risk-driven, life-cycle spirals of the TRW DARPA ACS Process model. Project reuse activities were identified for each of five major spirals for high assurance system development. These spirals will be discussed later in this briefing. The spiral basis for the Composite Process Model is illustrated here with circular clockwise rotations around four defined quadrants of activities. Spiral activities are initated in Quadrant I. Starting at Quadrant I (9:00), the objectives and constraints of the spiral stage are determined. In Quadrant II, spiral risk analysis and risk mitigation are accomplished. Principle activities are analyses and assessments as well as prototypes and simulations depending on the particular spiral stage and objectives. In Quadrant III the spiral products are developed. In Quadrant IV, the project spiral planning and management activities are conducted. Transitioning criteria support evaluations before a decision is made to advance to the next major spiral. There is no intended implication of elapsed time within a spiral. Some spirals may be of long duration while others may represent a very rapid set of activities and products. In addition, there may be subspirals within a spiral to address specific risks, and spirals may overlap within a project life-cycle. A true conceptual view of the spiral process will depend on an actual project realization.

INTEGRATING REUSE
COMPOSITE PROCESS MODEL FOUNDATION AND KEY ELEMENTS

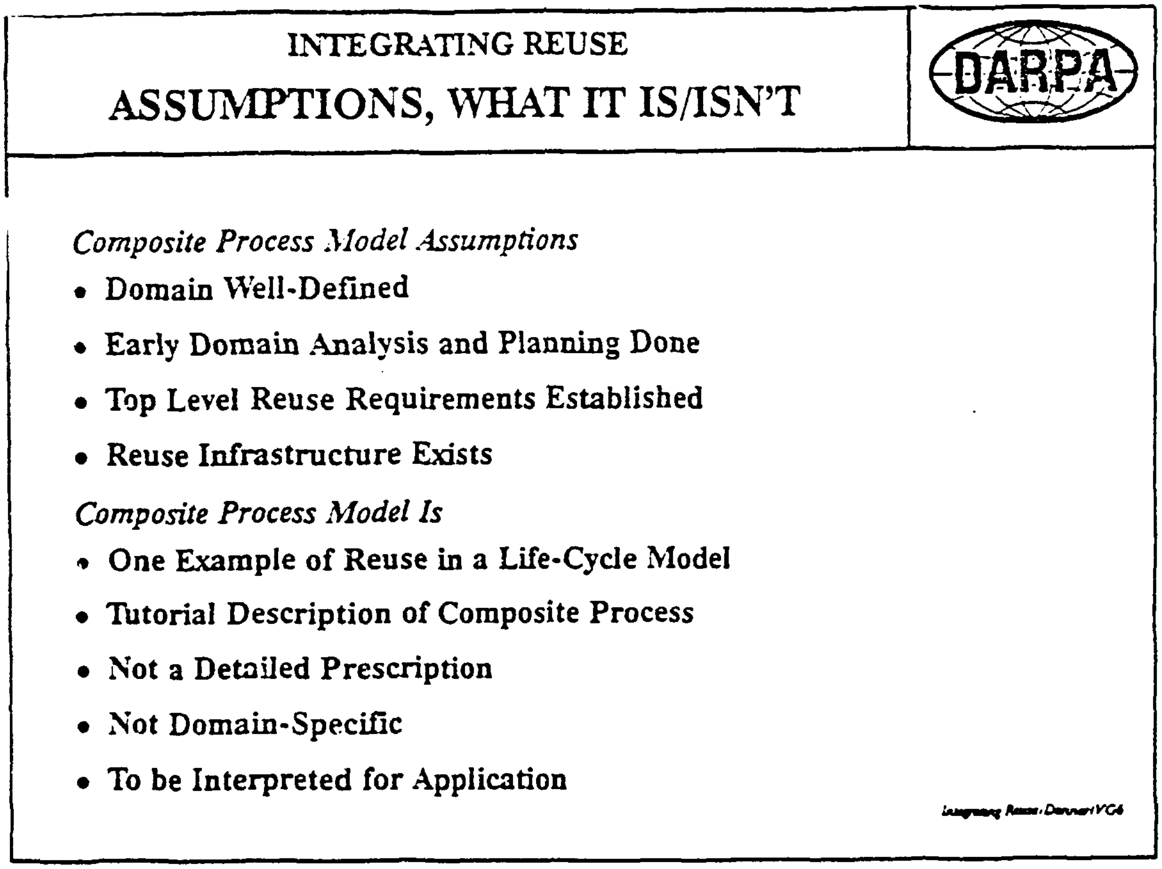

Primary Motivation/Drivers
Trust
Ada
Performance
Reuse
Foundation
I Objectives and Constraints
II Risk Analysis & Mitigation
Analysis and Assessments
Prototypes and Models
Transitioning
IV Planning and Management
III Development
Primary Constraints
Political/Sociological Environment
Cost
Available Technology/ Knowledge
Available Assets/ Automation
Domain

Key Elements of the Composite Process Model

- Risk Management
- Ada
- Engineering for Trust, Performance and Reuse
- Control and Assurance

Integrating Reuse/Denver/VGS

Using the TRW DARPA ACS process model as a technical foundation,we developed the reuse based activities with two primary strategies. First, the Composite Process Model stresses the early identification of risks (a characteristic of its spiral foundation) and organizes subsequent development activities to mitigate them. Second, the Composite Process Model calls for the integration of reuse, trust and performance engineering with modern software practices. The figure shows the motivations, drivers and key elements of the resulting model. Reuse-based drivers and constraints are highlighted. The domain of a specific application will constrain the adaptation of the Composite Process Model. The spiral insert illustrates the conceptual base for four generic quadrant classes of each cycle and the segments of activities within each quadrant. We will look more closely at conceptualizations of spiral activities later in this briefing. The model key elements may include, for example, 1) Risk Management: formal risk methodologies, modeling, planning for reuse, prototyping and demonstrations, analysis of reuse candidates and incremental development; 2) Engineering for Trust, Performance and Reuse: architecture assessment (model, prototype), critical mechanism prototyping and integration of critical, reusable assets; 3) Ada: homogeneous representation, consistent metrics and language support for reuse; 4) Control and Assurance: reasoning-based analysis/assurance, reuse of assurance results, CM and control, control/management of reuse library.



INTEGRATING REUSE

ASSUMPTIONS, WHAT IT IS/ISN'T

Composite Process Model Assumptions

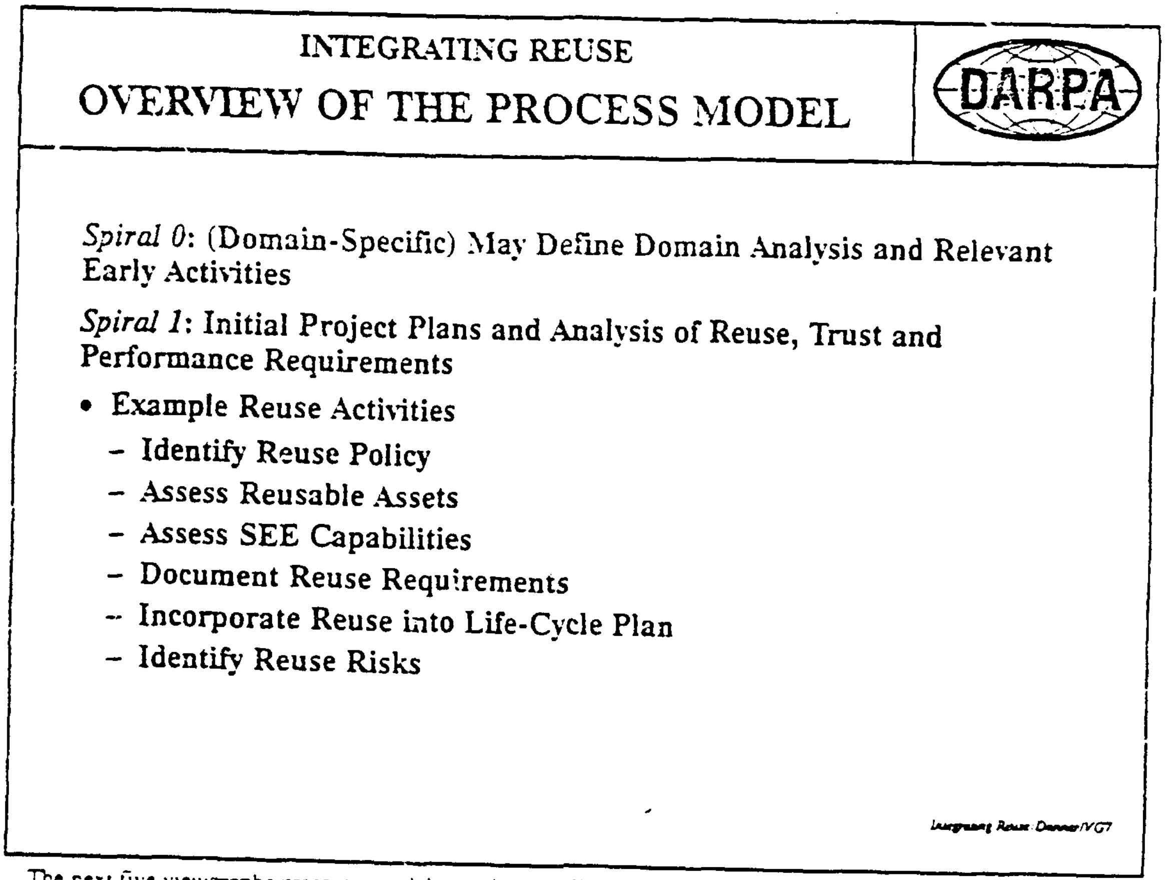
- **Domain Well-Defined**
- **Early Domain Analysis and Planning Done**
- **Top Level Reuse Requirements Established**
- **Reuse Infrastructure Exists**

Composite Process Model Is

- **One Example of Reuse in a Life-Cycle Model**
- **Tutorial Description of Composite Process**
- **Not a Detailed Prescription**
- **Not Domain-Specific**
- **To be Interpreted for Application**

Integrating Reuse, Denver/VC4

The Composite Process Model describes domain-independent, top level project activities that must necessarily fol!ow project conceptualization, planning and an already established initial approach. There are fundamental assumptions about the state of reuse prior to the first spiral of activities. These assumptions are: the domain of interest/application is well-defined; early domain analysis and planning are already accomplished, top level reuse requirements (goals for project use, management and creation of reusable assets) are established and a reuse infrastructure (library of assets, engineering environment, methodology and tools) exists. The Composite Process Model is an interim step toward a domain-specific process model. As previously mentioned, we initially modeled the process for domain analysis and precontract activities in the Navy Command and Control (C^2) domain with a Spiral 0 followed by the five spirals identified in the composite process, tailored to the application domain. The Composite Process Model is one example of reuse in a complete project life-cycle model. Because of its vast scope and its top level objectives, it is not a detailed prescription, rather it is a tutorial description of the composite process that incorporates reuse and traditional project activities into a risk-driven process model. As mentioned, the Composite Process Model is not domain-specific, it must be tailored to a specific domain. Importantly, the model must be interpreted for application to any real world project; and prescriptive guidance with specific activities, development environment characteristics and management controls must be defined.



INTEGRATING REUSE

OVERVIEW OF THE PROCESS MODEL

DARPA

Spiral 0: (Domain-Specific) May Define Domain Analysis and Relevant Early Activities

Spiral 1: Initial Project Plans and Analysis of Reuse, Trust and Performance Requirements

- Example Reuse Activities
  - Identify Reuse Policy
  - Assess Reusable Assets
  - Assess SEE Capabilities
  - Document Reuse Requirements
  - Incorporate Reuse into Life-Cycle Plan
  - Identify Reuse Risks

Integrating Reuse Denver/VG7

The next five viewgraphs present a model overview that lists the initial domain-based process Spiral 0 and the five major spirals that make up the composite process model. Principle reuse-based activities are listed for the Composite Spirals 1-5.

The Composite Process Model consists of spirals representing the five major groups of activities in the development of high assurance systems. These spirals are discussed in the Composite Process Model report, and a list of primary activities is detailed for each one. In addition, a conceptual view of each spiral is illustrated with major activities defined in each quadrant. Reuse activities are integrated into each quadrant and spiral. Implicitly and explicitly, reuse is a daily driver of the project. The prescribed, ordered sets of activities within each quadrant stem from iterative attempts over several cycles to reduce the crucial technical and program risks in reuse-driven developments and in systems requiring high trust and performance. Assuming a domain-specific basis, the Composite Process Model incorporates the considerations for reuse activities within the quadrant segments for :

Spiral 1 Initial Project Plans and Analysis of Reuse, Trust and Performance Requirements
Spiral 2 Reuse and Trust Enforcement Strategy and Basic Architecture
Spiral 3 Critical Elements and Architecture Refinement
Spiral 4 System Development and Assurance
Spiral 5 Maintenance.

An initial spiral for domain analysis may be conceptualized as Spiral 0. The actual representation of spirals and the number of spirals may vary depending on a specific project's need. For example, Spirals 1 and 2 could be conceptually combined into a single spiral on a small enough project that has relatively low risk while on the other hand, Spiral 4 may require partitioning into multiple spirals on a more complex project. Example reuse activities which represent a subset of the full list of activities for Spirals 1 through 5 are presented here to illustrate some of the defined activities. Each one of these activities can be thought of as a subprocess to be modeled for a particular project within a defined application domain.

INTEGRATING REUSE

OVERVIEW OF THE PROCESS MODEL

DARPA

Spiral 2: Reuse and Trust Enforcement Strategy and Basic Architecture

- Example Reuse Activities
  - Develop/Refine Reuse Strategy
  - Assess Reuse Technology as Required
  - Assess PM Application and Initial SEE Support
  - Initiate Prototypes for Reuse
  - Define Basic Architecture that Applies Reuse
  - Tailor SEE for Reuse Needs
  - Revise Reuse Plans

INTEGRATING REUSE

OVERVIEW OF THE PROCESS MODEL

DARPA

Spiral 3: Critical Elements and Architecture Refinement

- Example Reuse Activities
  - Incorporate Reuse Constraints into Critical Elements Analysis
  - Develop Critical Reuse Elements
  - Assess Prototype Reuse Qualifications
  - Reassess Reuse Risks
  - Prototype Reuse Approach
  - Establish Reuse-Relevant Architecture
  - Provide Asset Assurance
  - Revise Reuse Plans

INTEGRATING REUSE

OVERVIEW OF THE PROCESS MODEL

Spiral 4: System Development and Assurance

- Example Reuse Activities
  - Use Acceptable Assets in Development
  - Assess Reuse Requirements Compliance
  - Test, Evaluate and Certify Reused Assets
  - Document Assets for Future Reuse
  - Apply CM for Reuse and Trust
  - Develop Guidelines for Maintenance and Reuse
  - Review Lessons Learned

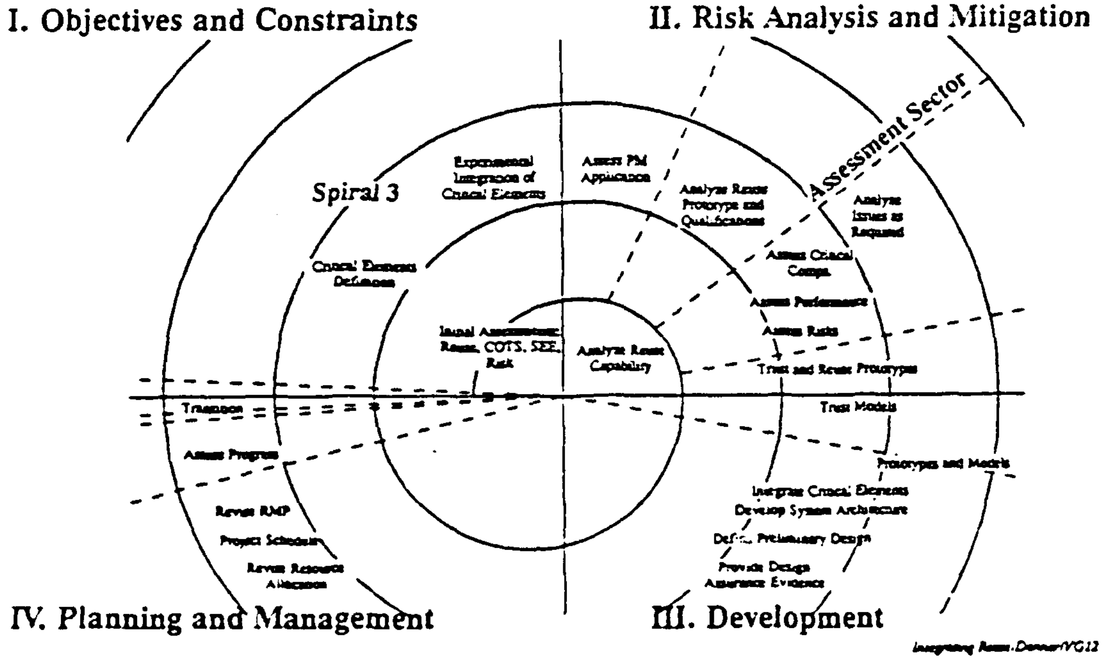
INTEGRATING REUSE

OVERVIEW OF THE PROCESS MODEL

Spiral 5: Maintenance

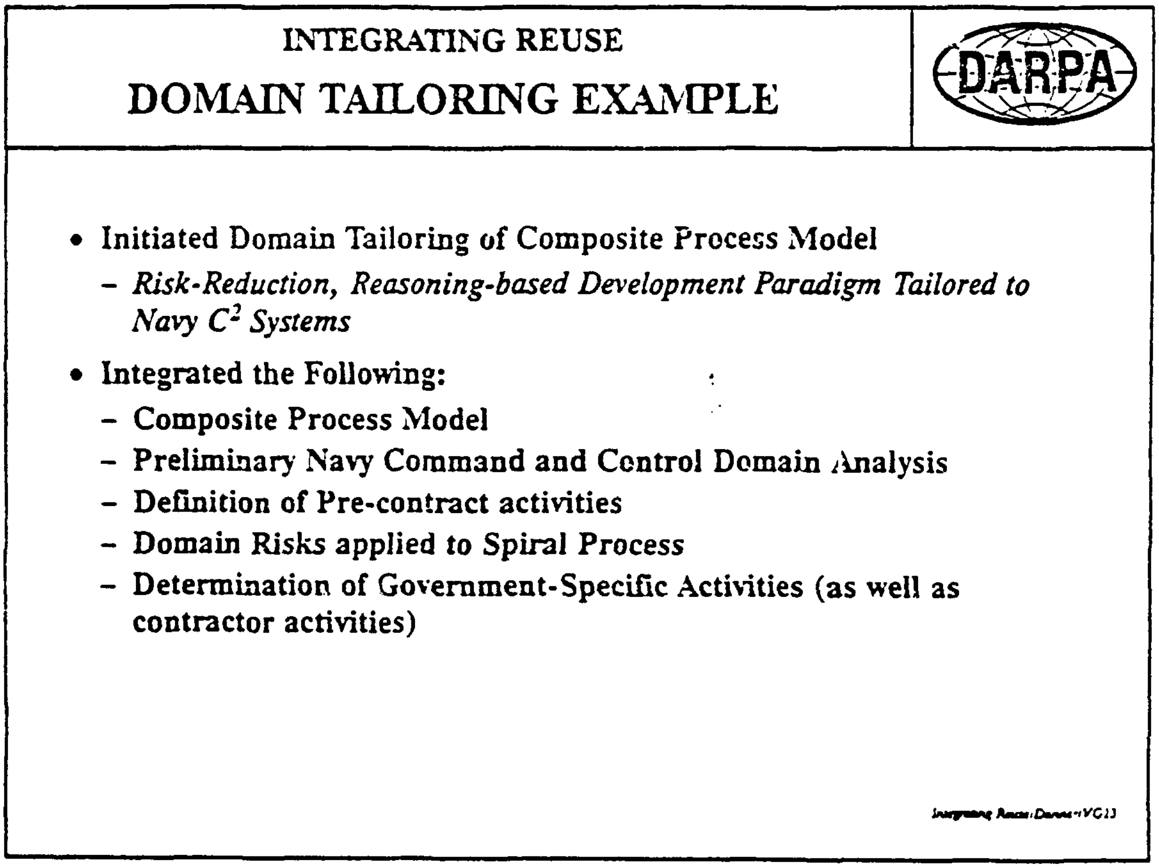
- Example Reuse Activities
  - Implement Reuse Change Tracking
  - Maintain Baselined Assets
  - Update Reuse Constraints
  - Assess Impact of Proposed Changes to Assets
  - Assess Technology to Support Reuse
  - Identify New Reuse Risks and Ways to Mitigate
  - Develop Design/Design Revisions to Assets (as needed)
  - Retest, Reevaluate, Recertify as needed
  - Revise Reuse Risks as Needed
  - Review Lessons Learned
  - Plan for Future Reuse

INTEGRATING REUSE
A CLOSER LOOK: QUADRANTS AND SECTORS


I. Objectives and Constraints
II. Risk Analysis and Mitigation
Assessment Sector
Spiral 3
Experimental Integration of Critical Elements
Assess PM Application
Analyze Reuse Prototype and Qualifications
Analyze Issues as Required
Critical Elements Definition
Assess Critical Comps.
Assess Performance
Assess Risks
Initial Assessments: Reuse, COTS, SEE, Risk
Analyze Reuse Capability
Trust and Reuse Prototypes
Transition
Trust Models
Assess Progress
Prototypes and Models
Integrate Critical Elements
Develop System Architecture
Revise RMP
Project Schedule
Define Preliminary Design
Revise Resource Allocation
Provide Design Assurance Evidence
IV. Planning and Management
III. Development

Integrating Reuse.DonnerIVG12

This figure illustrates a conceptual view of Spiral 3, Critical Elements and Architecture Refinement of the Composite Process Model. It also shows an Assessment Sector that overlaps a number of spirals. In the sector starting in Quadrant I, we can see the Spiral 1 (and 2) activities that lead to project objectives and risk mitigation include initial assessments for reuse, proposed COTS products, the software engineering environment and an assessment of project reuse (and other) risks. Spirals 1 and 2 include analyses of reuse capabilities. If we follow the Spiral 3 activities starting in Quadrant I (Objectives), we trace the definition of spiral objectives and experimentation with critical project elements. These elements are essential for project success and may represent such things as high-risk reuse asset utilization, trust mechanisms, high performance elements, or user interface functions. Spiral 3 Quadrant II (Risk Analysis) activities illustrate risk mitigation that includes an assessment of the application of the process model to date, analyses of reuse prototypes and the qualifications of reusable assets, assessments of critical components, performance assessments (as relevant), revised risk assessments, trust/reuse/critical function prototypes as applicable and trust modeling as needed. The Prototype and Modeling Sector may also be conceptualized as being part of Quadrant III (Development) since project products may result from those activities. Development products may include design documentation, system architecture descriptions and assurance evaluation evidence. Quadrant IV represents project planning, plan revisions and evaluation of current progress before the transition to Spiral 4 can occur.



INTEGRATING REUSE

DOMAIN TAILORING EXAMPLE

DARPA

- Initiated Domain Tailoring of Composite Process Model
  - *Risk-Reduction, Reasoning-based Development Paradigm Tailored to Navy C^2 Systems*
- Integrated the Following:
  - Composite Process Model
  - Preliminary Navy Command and Control Domain Analysis
  - Definition of Pre-contract activities
  - Domain Risks applied to Spiral Process
  - Determination of Government-Specific Activities (as well as contractor activities)

Integrating Reuse:Domain VG13

Under STARS tasking, we conducted an initial domain analysis and identified a preliminary set of domain risks and characteristics for Navy C^2 System development. The Composite Process model was adapted to the Navy C^2 domain, and the resulting Navy C^2 Process Model (NCCPM) is documented in the STARS report, *Risk-Reduction, Reasoning-Based Development Paradigm Tailored to Navy C^2 Systems*. The preliminary domain analysis work is summarized in an Appendix A to this report. The NCCPM work adapts the Composite Process Model to the preliminary oomain analysis, defines pre-contract activities for the Navy and other organizations, applies domain risks to each spiral of activity and determines and integrates specific Government and contractor development activities for each spiral.

INTEGRATING REUSE

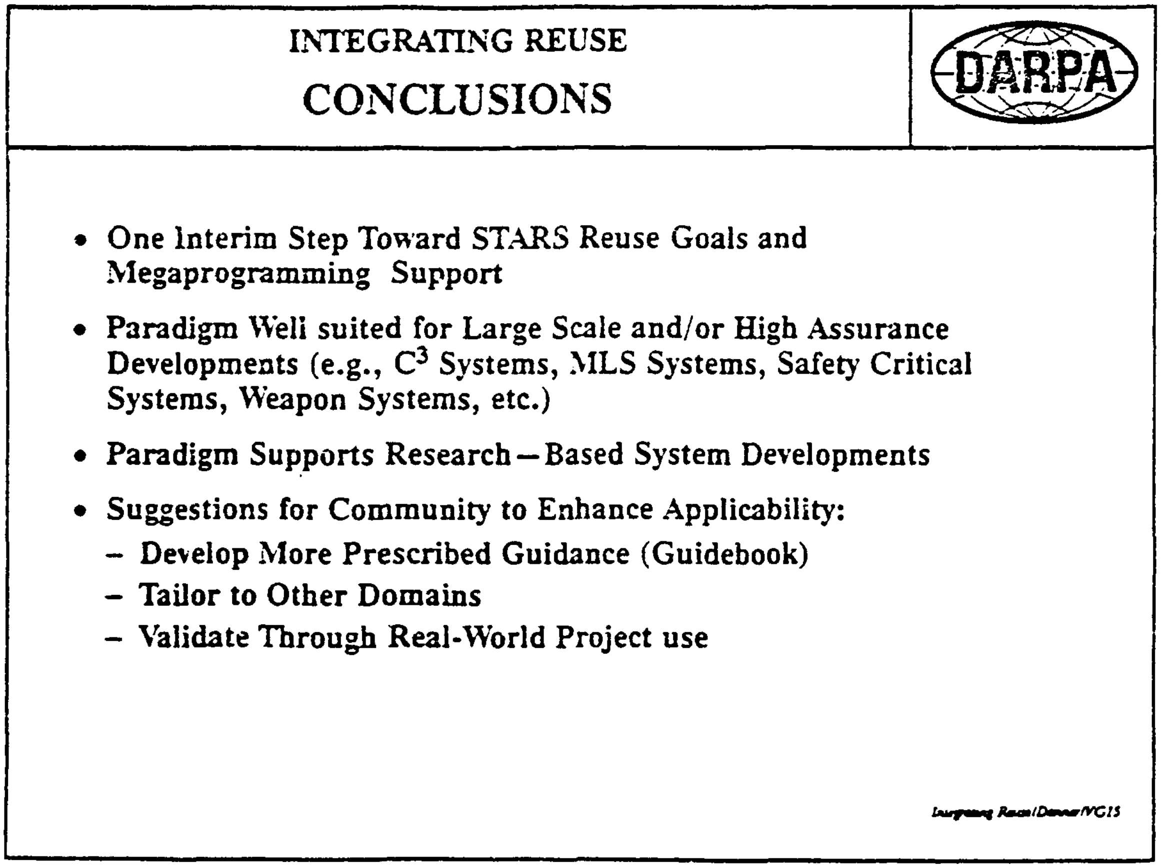
DOMAIN TAILORING EXAMPLE

DARPA

- Developed Risk Summary Tables
  - Technical and Programmatic Risks
  - Risk Mitigation Activities Mapped to Relevant Spirals
- Defined Preliminary Tables Mapping Standards to Spirals
- Created Spiral O: Concept Through Contract Award
  - Domain Analysis Activities
  - Pre-Contract Activities (Sponsoring and Performing Organizations)
  - Description of Domain-Specific Reuse Activities
  - Five Subspirals

Integrating Reuse/Domain/VG14

The NCCPM report specifies domain-specific reuse risks and activities that mitigate those risks throughout the process life-cycle. These activities are summarized in risk tab that map risks to spirals (0-5). The Composite Process Model tailoring to a specific domain also includes initial tables that map such standards as DoD 2167A and DoD 5200.25-STD to project spirals. The NCCPM includes a Spiral 0 for domain analysis and precontract process modeling for both sponsoring and performing organization activities. Spiral 0: Concept Through Contract Award consists of five subspirals that list the many early activities required to define the system concept, initial specifications and RFP, preparations to respond to the RFP, and the writing and evaluation of proposals.



INTEGRATING REUSE

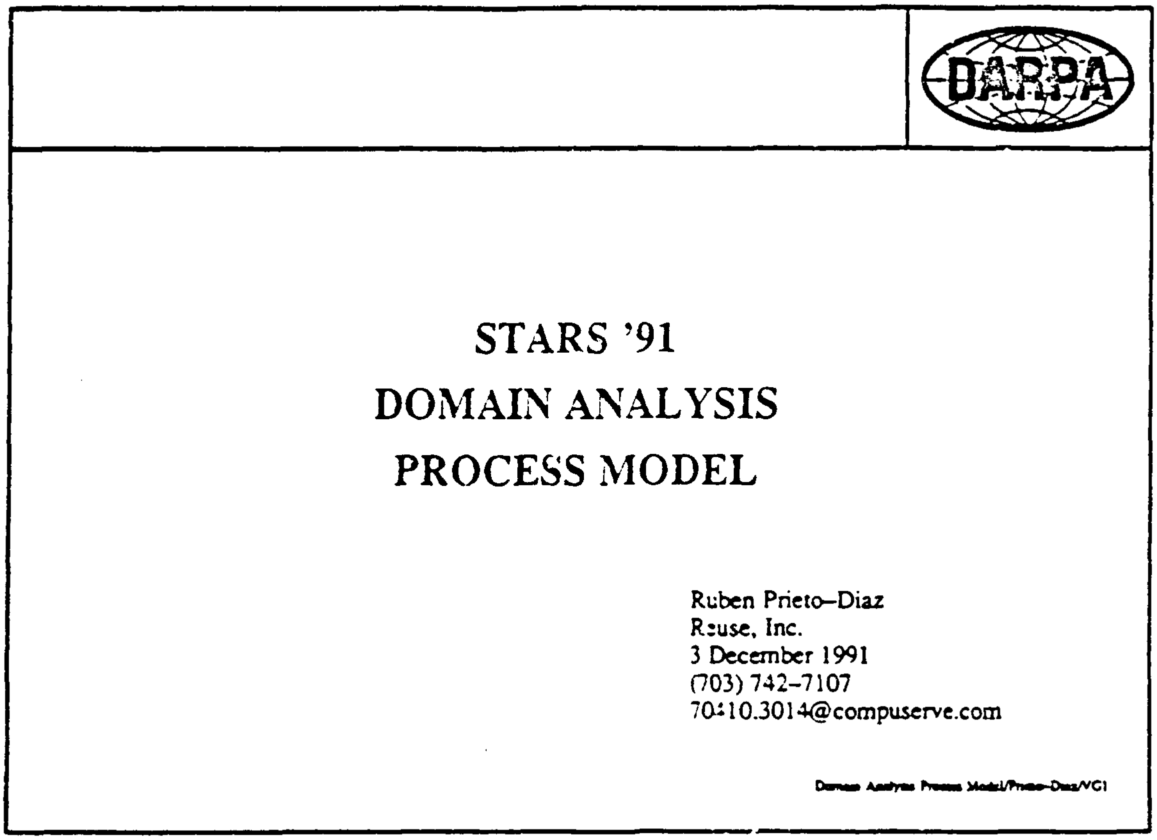
CONCLUSIONS

DARPA

- One Interim Step Toward STARS Reuse Goals and Megaprogramming Support
- Paradigm Well suited for Large Scale and/or High Assurance Developments (e.g., C^3 Systems, MLS Systems, Safety Critical Systems, Weapon Systems, etc.)
- Paradigm Supports Research—Based System Developments
- Suggestions for Community to Enhance Applicability:
  - Develop More Prescribed Guidance (Guidebook)
  - Tailor to Other Domains
  - Validate Through Real-World Project use

Integrating Reuse/Denver/VG15

The Composite Process Model is one example of integrating reuse into a process model; one interim step toward the STARS reuse goals. Interpreted for a specific application domain, the Composite Process Model is one approach for STARS domain-specific, reuse-based megaprogramming support. The model addresses the full life-cycle development process for a risk-driven system development. It provides a paradigm for large scale systems and for the development of high assurance (trusted) systems. Some of the types of developments that are especially suited to the paradigm are safety critical systems such as flight control, medicine dispensing applications, etc, and highly trusted systems such as multilevel secure systems. In addition, because of its built-in flexibility and risk mitigation emphasis, the Composite Process Model is clearly appropriate for research-based system developments. (e.g., In phase II of the DARPA ACS research project, TRW and its subcontractors are applying the foundation process model to the development of a trusted X Window System prototype aimed at the B3 level of trust.) There are future tasks that would enhance the applicability of the Composite Process Model. Our suggestions to the software engineering community are: 1) Develop a guidebook for the Composite Process Model that would provide more prescriptive steps for interpretation and use of the model for a real-world application; 2) Tailor the Composite Process Model to other example domains and provide feedback for improving the current process as well as more explicit process model descriptions that are applicable for their specific domains; 3) Validate the Composite Process Model through actual project use; provide feedback for process improvement so that the model can evolve to a viable, visible supporting element of the STARS reuse vision.



STARS '91 DOMAIN ANALYSIS PROCESS MODEL

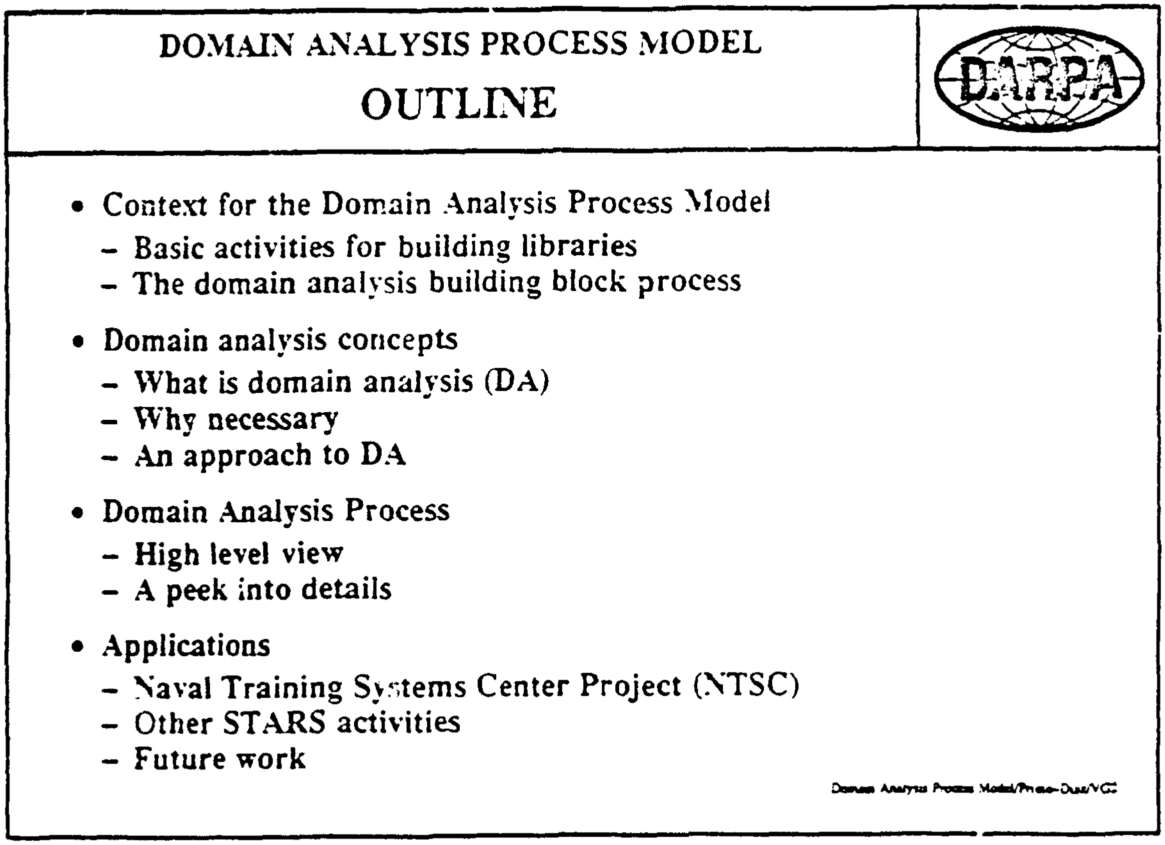
Ruben Prieto–Diaz
Reuse, Inc.
3 December 1991
(703) 742–7107
70410.3014@compuserve.com

Domain Analysis Process Model/Prieto–Diaz/VG1

This presentation is a brief introduction and overview of the STARS Domain Analysis Process Model.

It is one of the building blocks of a model for reuse library processes being developed for STARS.

The model for Reuse Library Processes is available in report format from STARS. It is entitled STARS Reuse Library Process Model.



DOMAIN ANALYSIS PROCESS MODEL

OUTLINE

- Context for the Domain Analysis Process Model
  - Basic activities for building libraries
  - The domain analysis building block process
- Domain analysis concepts
  - What is domain analysis (DA)
  - Why necessary
  - An approach to DA
- Domain Analysis Process
  - High level view
  - A peek into details
- Applications
  - Naval Training Systems Center Project (NTSC)
  - Other STARS activities
  - Future work

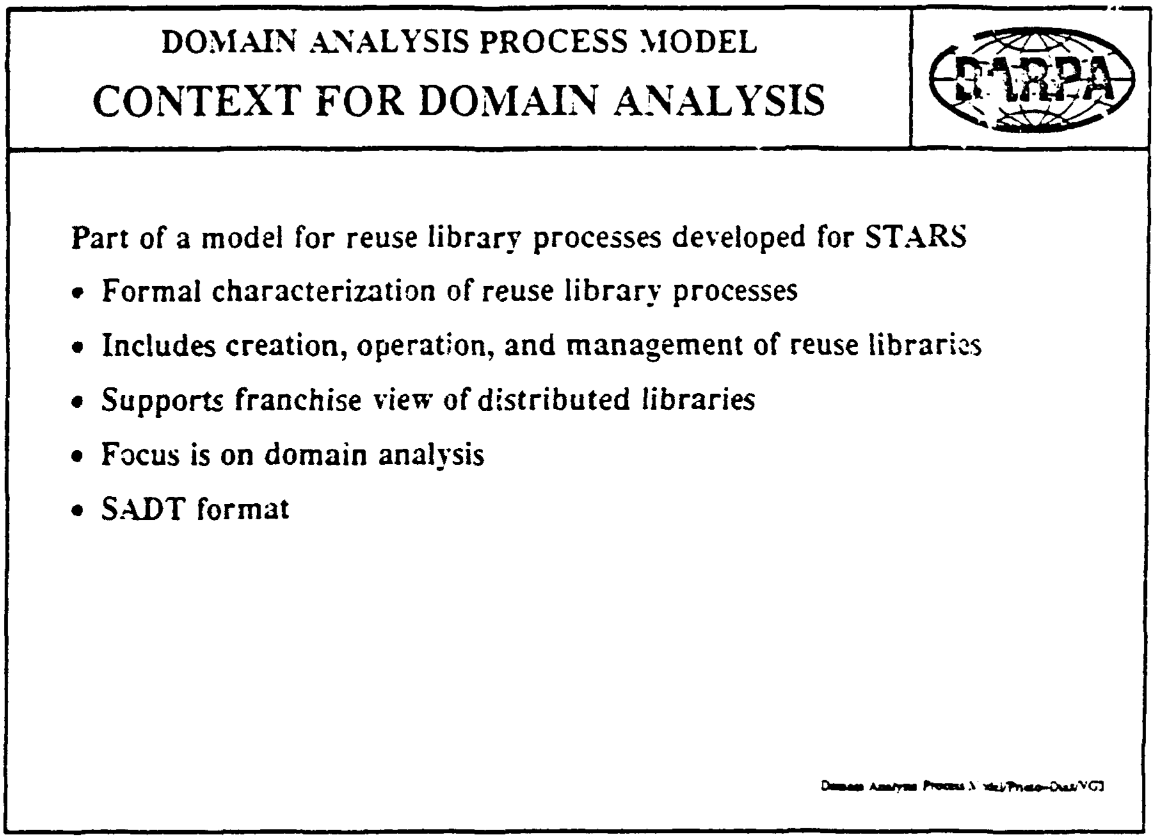
Domain Analysis Process Model/Prieto-Diaz/VGC

The Domain Analysis Process Model was developed as one of the subprocesses for developing and managing reuse libraries.

This presentation describes briefly the activities of the library process model to illustrate the role of domain analysis within the context of creating reuse libraries.

Some basic concepts, justification, and views of domain analysis will be introduced along with a description of the process. The process is summarized by listing the high level activities involved. Selected diagrams are also shown.

The presentation concludes with description of some of the applications of the model inside and outside STARS.



DOMAIN ANALYSIS PROCESS MODEL

CONTEXT FOR DOMAIN ANALYSIS

DARPA

Part of a model for reuse library processes developed for STARS

- Formal characterization of reuse library processes
- Includes creation, operation, and management of reuse libraries
- Supports franchise view of distributed libraries
- Focus is on domain analysis
- SADT format

Domain Analysis Process Model/Phase-Out/VG1

The objective of the library process model is to formally characterize the various processes that take place in the context of Reuse Libraries.

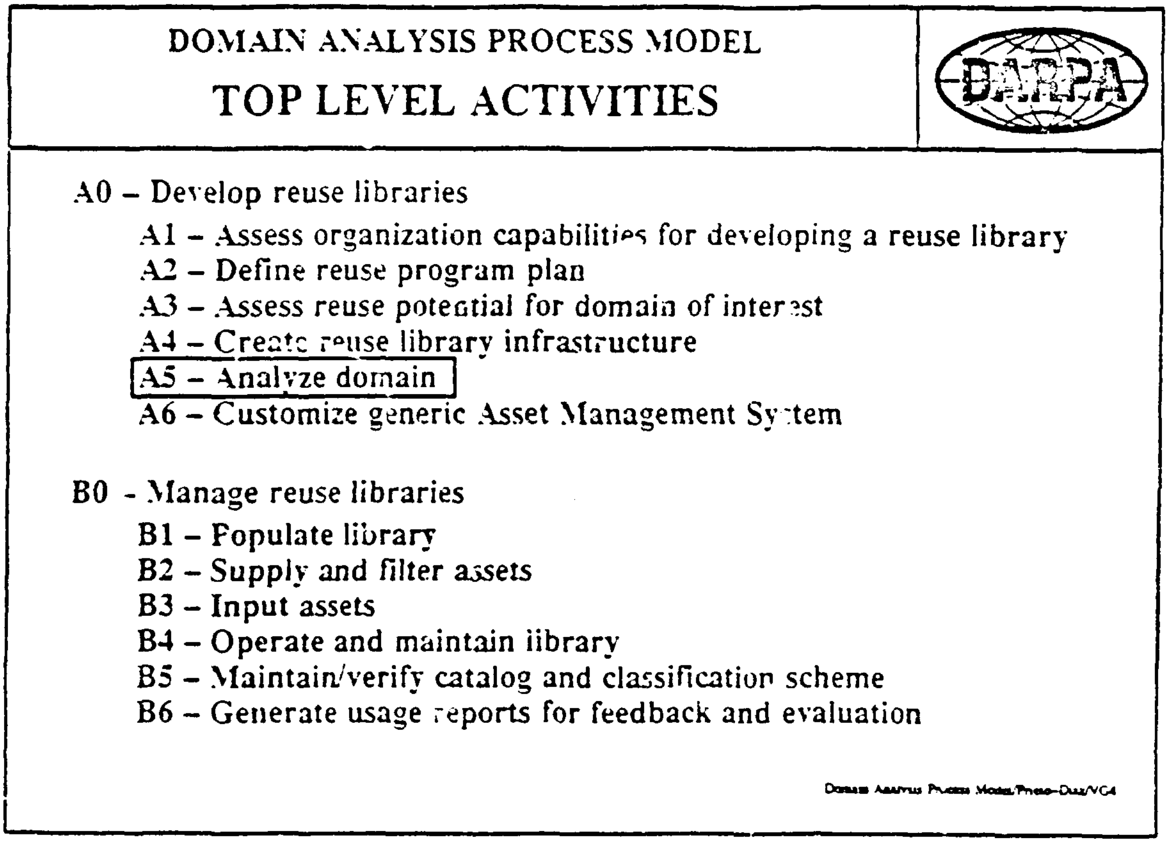
These processes include not only the operation and management of reuse libraries but the preparation and analysis work required for establishing such libraries in participating organizations.

This model supports a *franchise* view of library development. The current version of the library process model concentrates on how to create standard reuse libraries to facilitate the implementation of a nationwide reuse program.

Standard guidelines make possible the development of independent libraries with common requirements and at the same time provide certain degree of flexibility for organizations to specialize in specific application domains. This flexibility allows for a quick and decentralized development of specialized libraries resulting in a rapid expansion of the reuse library base.

The Domain Analysis component plays a key role in creating domain specific libraries that support a reuse-based software development. The model is decomposed to very low detail.

The presentation format is SADT, a trademark of SofTech. SADT is a technique for thinking in a structured way about large and complex problems and includes a graphical notation that is clear and precise in communicating system functionality.



DOMAIN ANALYSIS PROCESS MODEL

TOP LEVEL ACTIVITIES

A0 – Develop reuse libraries
- A1 – Assess organization capabilities for developing a reuse library
- A2 – Define reuse program plan
- A3 – Assess reuse potential for domain of interest
- A4 – Create reuse library infrastructure
- A5 – Analyze domain
- A6 – Customize generic Asset Management System

B0 - Manage reuse libraries
- B1 – Populate library
- B2 – Supply and filter assets
- B3 – Input assets
- B4 – Operate and maintain library
- B5 – Maintain/verify catalog and classification scheme
- B6 – Generate usage reports for feedback and evaluation

Domain Analysis Process Model/Phase-Dua/VG4

Library activities are divided in two main groups: library development and library management.

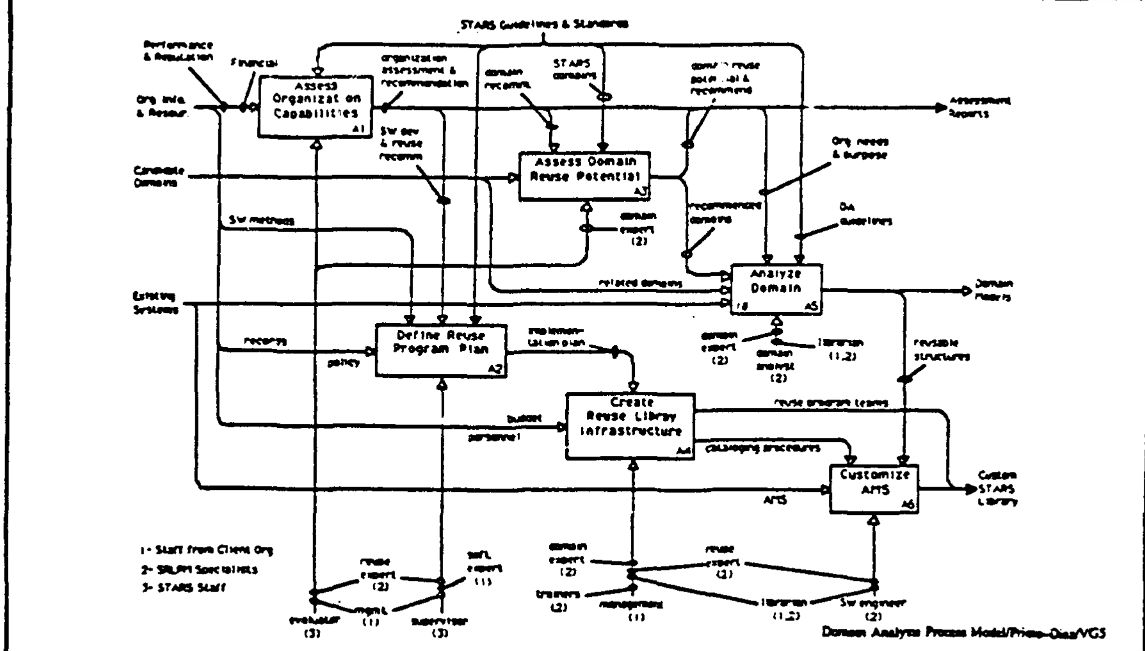
Library development includes assessing an organization's capability to develop and maintain a reuse library, defining a reuse plan tailored to the organization's needs, assessing the reuse potential for the application domain where the library is going to be used, the analysis of the domain, and tailoring a generic library system to support system construction based on domain specific architectures.

Library management includes the initial population of the library, supply of quality assets, cataloging and classifying new assets, library operation and maintenance, maintenance and verification of catalog and classification scheme, and report generation and evaluation.

To date only the domain analysis process has been developed thoroughly and to a very low level of detail. Although the remaining activities are only described in general, the library model provides a complete framework for the overall process.

DOMAIN ANALYSIS PROCESS MODEL

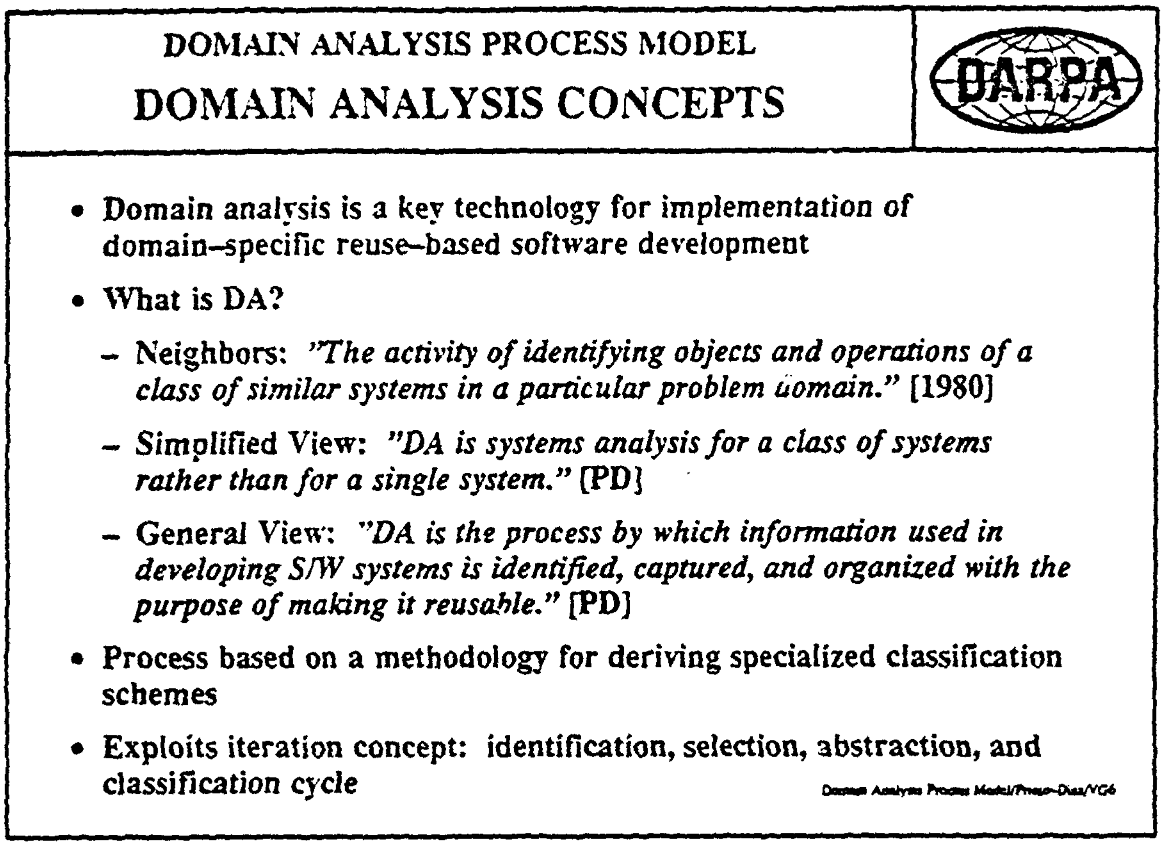
DEVELOP REUSE LIBRARIES



Domain Analysis Process Model/Prieto-Diaz/VGS

This is the top level diagram for developing reuse libraries. Inputs are decomposed into their more specific components. The Organization Information input, for example, consists of financial information, performance history, budget, etc. Mechanisms are also decomposed but their classification is indicated by a number (bottom left in the diagram). Mechanisms of class (1) are from the client organization, class (2) are reuse library specialists, and class (3) are commissioned staff from the STARS program. Intermediate outputs are named in the diagram and their association with respective arrows is indicated, in most cases, with a line and a small circle. The rightmost outputs are aggregations of intermediate outputs.

There are several intermediate outputs that are used as controls or inputs to other activities. The implementation plan created in A2, for example, is used to guide (i.e., control) the creation of the reuse infrastructure A4.



DOMAIN ANALYSIS PROCESS MODEL

DOMAIN ANALYSIS CONCEPTS

- Domain analysis is a key technology for implementation of domain–specific reuse–based software development
- What is DA?
  - Neighbors: *"The activity of identifying objects and operations of a class of similar systems in a particular problem uomain."* [1980]
  - Simplified View: *"DA is systems analysis for a class of systems rather than for a single system."* [PD]
  - General View: *"DA is the process by which information used in developing S/W systems is identified, captured, and organized with the purpose of making it reusahle."* [PD]
- Process based on a methodology for deriving specialized classification schemes
- Exploits iteration concept: identification, selection, abstraction, and classification cycle

Domain Analysis Process Model/Prieto-Diaz/VG6

Domain analysis holds the key to a systematic, formal and effective practice of software reuse. To be effective, reusable assets must integrate easily within a predefined, and preferably, standard architecture. The task in domain analysis is to develop such an architecture and provide the information needed to specify standard components.

Most proposed approaches and methods for domain analysis assume that domain knowledge exists and is readily available and usable. Experience indicates however, that acquiring and structuring knowledge is a bottleneck of domain analysis.

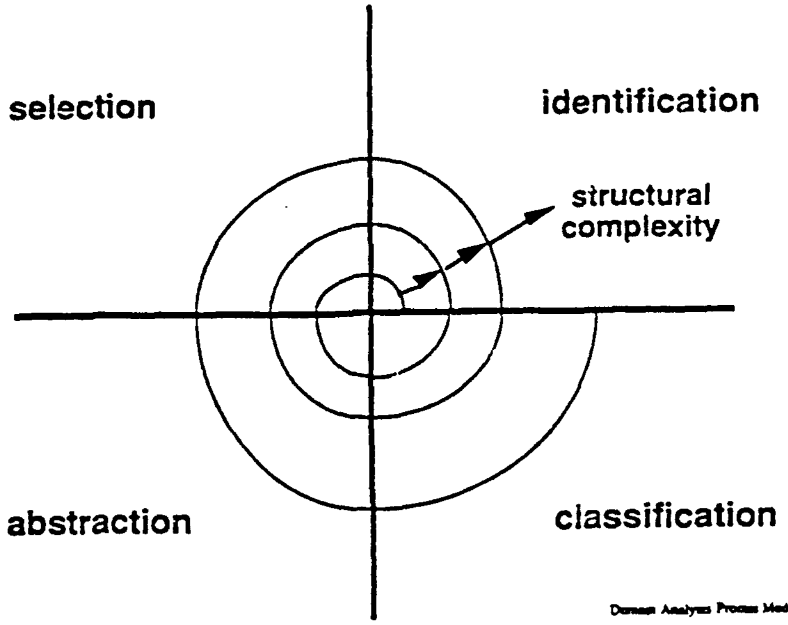
The current model is based on an earlier model proposed in 1987 based on a method for deriving faceted classification schemes for special collections.

The key concept is an iterative loop of identification, selectiun, abstraction, and classification of domain information.

This loop can be represented as a spiral.

DOMAIN ANALYSIS PROCESS MODEL

SPIRAL VIEW OF DA



Domain Analysis Process Model/Phase-Desc/VG7

The spiral starts with identification of specific objects, operations, and relationships at a high level of abstraction.

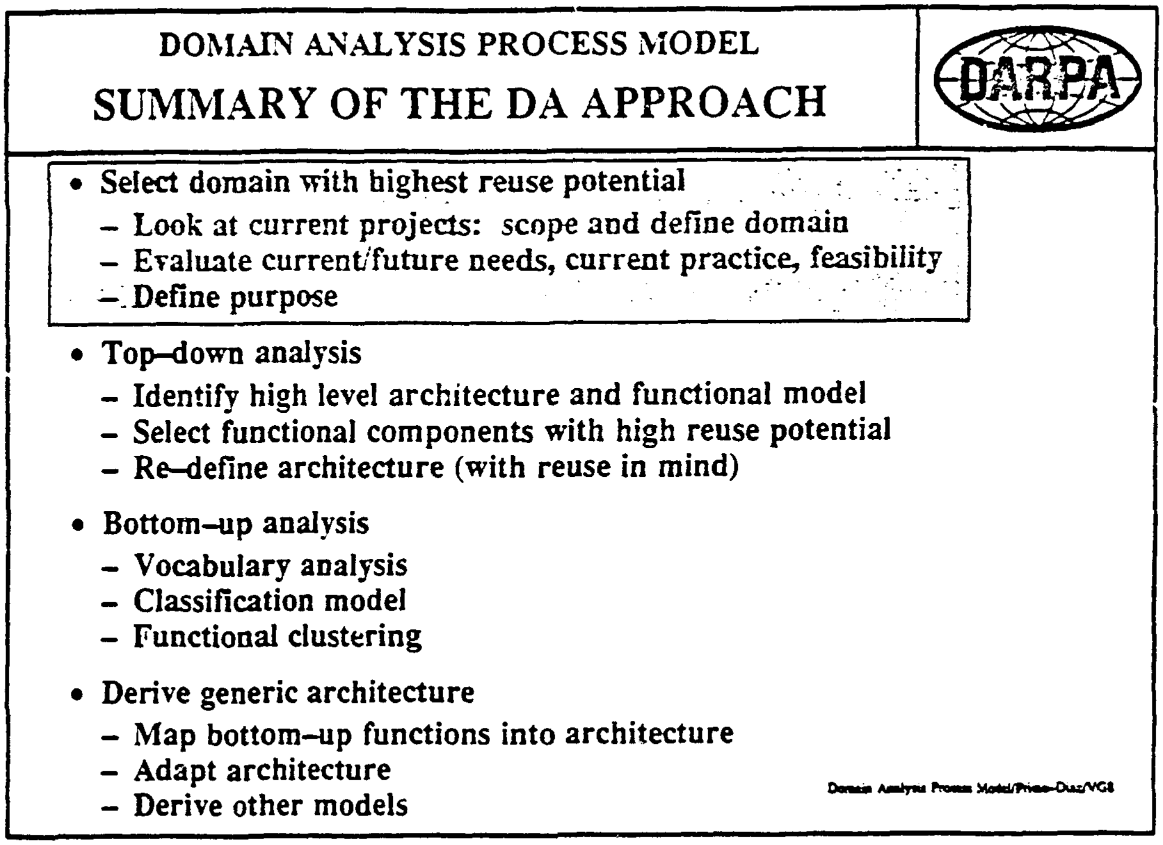
Relevant objects and operations are selected and then abstracted to capture their essential attributes/characteristics or features.

The objects and operations are then classified by their common attributes.

As the spiral progresses, structures are integrated into larger elements of the architecture.

This basic process is similar to a process practiced in library science to derive classification schemed called *literary warrant*.

The new approach complements this bottom-up approach with a top-down preliminary identification of generic architectures.



DOMAIN ANALYSIS PROCESS MODEL

SUMMARY OF THE DA APPROACH

- Select domain with highest reuse potential
  - Look at current projects: scope and define domain
  - Evaluate current/future needs, current practice, feasibility
  - Define purpose
- Top–down analysis
  - Identify high level architecture and functional model
  - Select functional components with high reuse potential
  - Re–define architecture (with reuse in mind)
- Bottom–up analysis
  - Vocabulary analysis
  - Classification model
  - Functional clustering
- Derive generic architecture
  - Map bottom–up functions into architecture
  - Adapt architecture
  - Derive other models

Domain Analysis Process Model/Prieto-Diaz/VGI

This is a summary of the proposed domain analysis process.

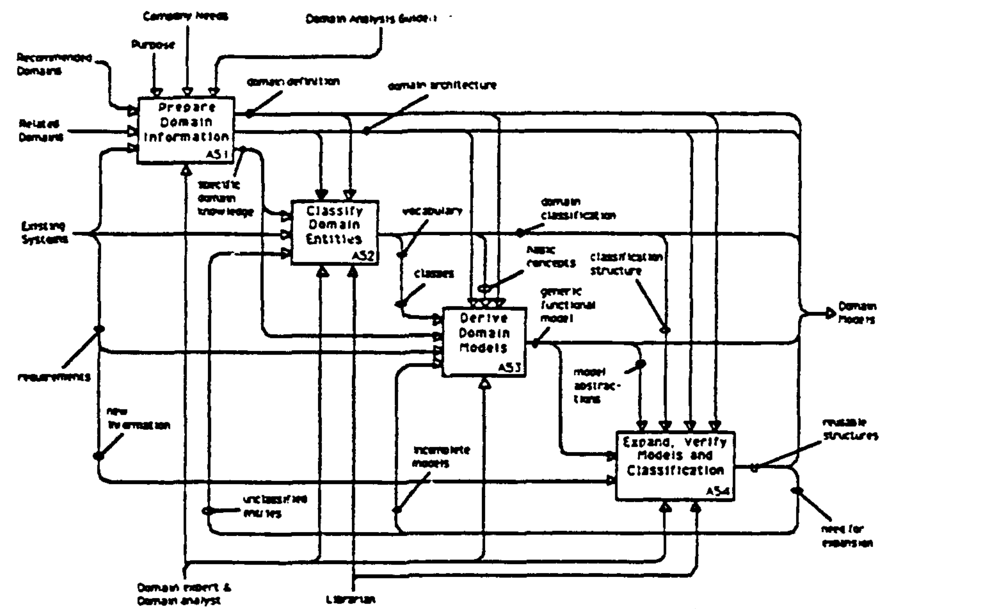
The activities in the dotted box can be considered as preparation for domain analysis. The objective is to assess the reuse potential of a selected domain and to define the purpose of the analysis. The purpose of a domain analysis may range from providing a basic understanding of the domain to developing a common architecture including specifications for reusable building blocks.

The process follows a *sandwich* approach. It is based on a combination bottom–up and top–down approach similar to the one used in developing software systems. During the bottom–up stage, low level requirements, source code, and documentation from existing systems are analyzed to produce a preliminary vocabulary, a taxonomy, a classification structure, and standard descriptors. During the top–down stage, high level designs and requirements of current and new systems are analyzed for commonality. The outcome of this analysis includes a canonical structure common to all systems in the domain, identification of stable and variable characteristics, a generic functional model, and information on the interrelationships among the structure elements.

The outcomes of both approaches are integrated into reusable structures. This integration process consists of associating the products of the bottom–up analysis with the structures derived by the top–down analysis. The result is a natural match between high level generic models and low level components. Assembly of new systems from basic components, thus, becomes a library search and retrieval operation using the domain models as skeleton guides.

DOMAIN ANALYSIS PROCESS MODEL

DOMAIN ANALYSIS PROCESS



AS: Analyze Domain

Domain Analysis Process Model/Prieto-Diaz/VG9

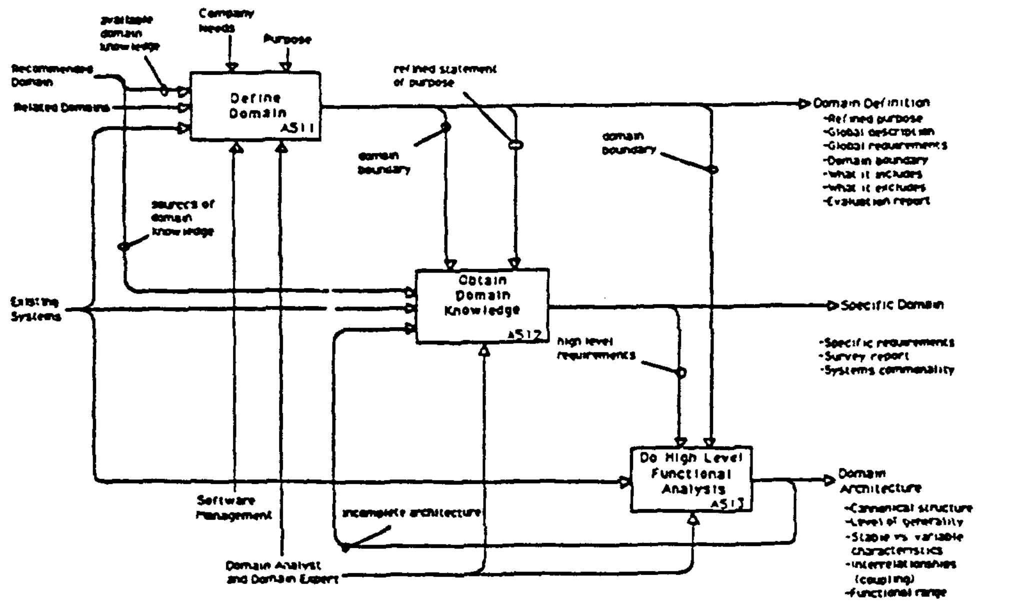
This diagram represents the high level decomposition of the domain analysis process. The four activities are highly interrelated. The first activity, "Prepare Domain Information" includes a preliminary top-down analysis to propose a basic "descriptive" architecture. The next two activities, "Classify Domain Entities" and "Derive Domain Models" are the core of the bottom-up analysis. In the last activity, "Expand and Verify Models and Classification", the preliminary high level architecture is revised and modified to include the domain models. After some iterations, a "prescriptive" architecture made of reusable structures is generated.

The diagram shows the inputs for each activity, the intermediate outputs, the feed-back loops, their respective controls and mechanisms, and the final outputs.

It is decomposed into several subactivities, each described in detail and with examples.

DOMAIN ANALYSIS PROCESS MODEL

PREPARE DOMAIN INFORMATION



Domain Analysis Process Model/Prieto-Diaz/VG10

This diagram is an example of a third level decomposition.

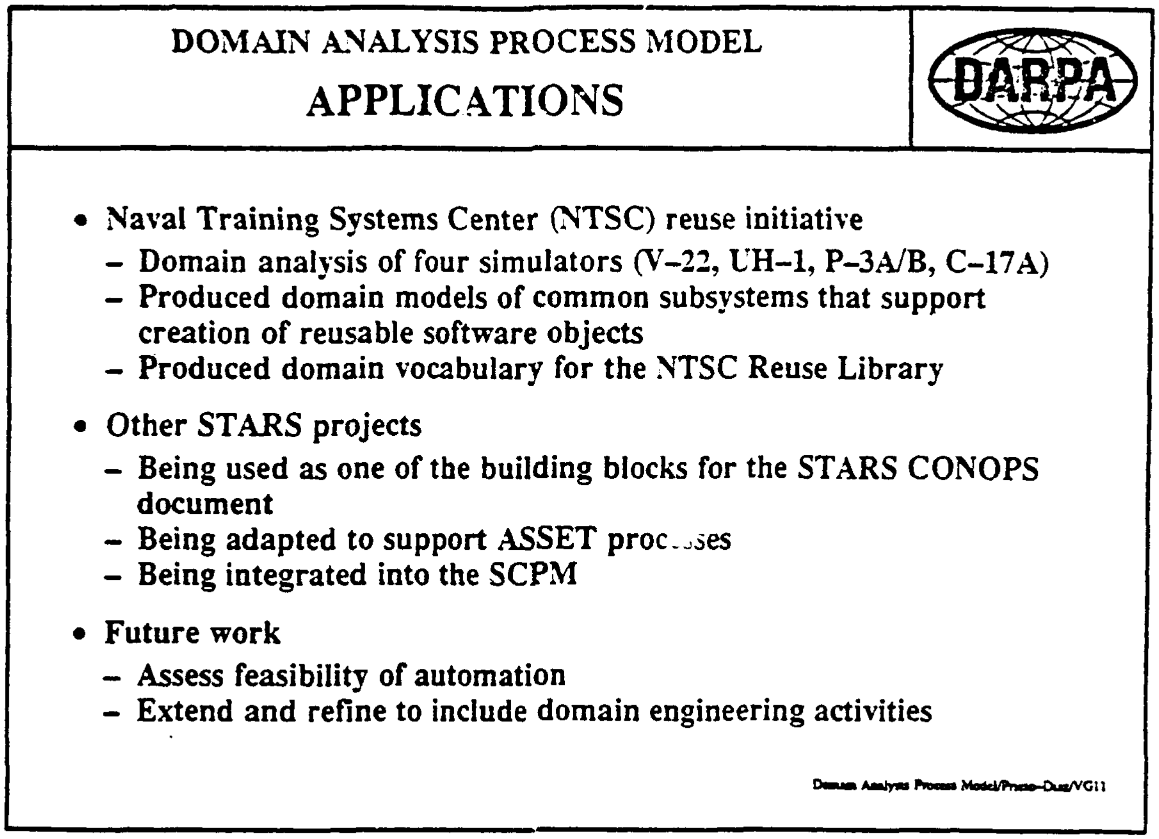
It shows the activities involved in preparing domain information and the outputs expected.

The outputs are mainly in the form of reports.

Lower level decompositions produce specific, well defined outputs.

Decomposition is carried to level five.

Many of the lowest level activities could be automated and most of them with current technology.



DOMAIN ANALYSIS PROCESS MODEL

APPLICATIONS

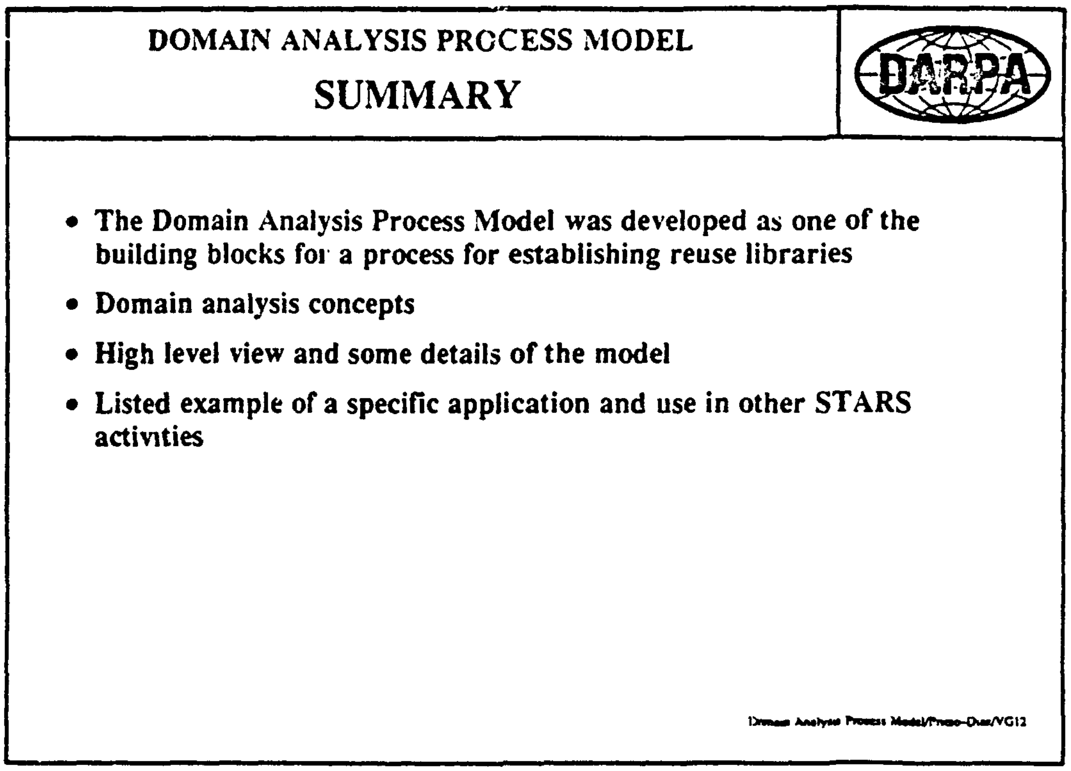
- Naval Training Systems Center (NTSC) reuse initiative
  - Domain analysis of four simulators (V-22, UH-1, P-3A/B, C-17A)
  - Produced domain models of common subsystems that support creation of reusable software objects
  - Produced domain vocabulary for the NTSC Reuse Library
- Other STARS projects
  - Being used as one of the building blocks for the STARS CONOPS document
  - Being adapted to support ASSET processes
  - Being integrated into the SCPM
- Future work
  - Assess feasibility of automation
  - Extend and refine to include domain engineering activities

Domain Analysis Process Model/Prieto-Diaz/VG11

The STARS Domain Analysis Process Model has been used in the flight simulation domain. The Naval Training Center (NTSC) Reuse Initiative project used it to do a domain analysis of four flight simulators, the V-22 Operational Flight Trainer, the UH-1 Flight Simulator, the P-3A/B Tactical Navigation Modernization Operational Flight Trainer, and the C-17A Weapons System Trainer. The objective of this domain analysis was to produce a set of domain models which will support the creation and reuse of software objects for the NTSC Reuse Library. This in turn supports reuse-based software development.

The model is also being used as the basis for the "Create Assets" process in the STARS CONOPS document. it is being modified to support ASSET domain analysis activities within their reuse library processes, and it is being integrated into the STARS Composite Process Model.

Future work include analysis and assessment of primitive activities for possible automation and integration into a software development environment, and extension to include some domain engineering activities dealing with asset creation.

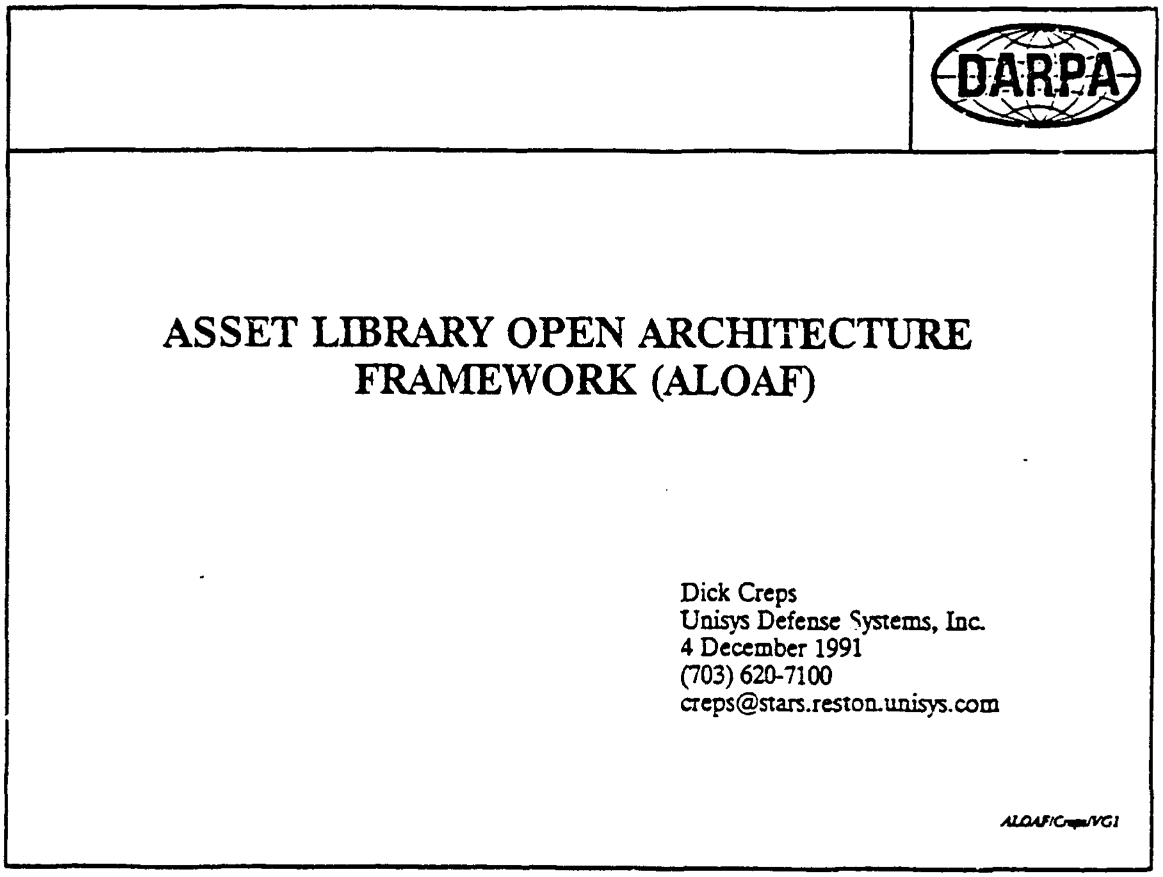


DOMAIN ANALYSIS PRCCESS MODEL

SUMMARY

- The Domain Analysis Process Model was developed as one of the building blocks foı a process for establishing reuse libraries
- Domain analysis concepts
- High level view and some details of the model
- Listed example of a specific application and use in other STARS activıties

Domain Analysis Process Model/Proto-Ovr/VG12



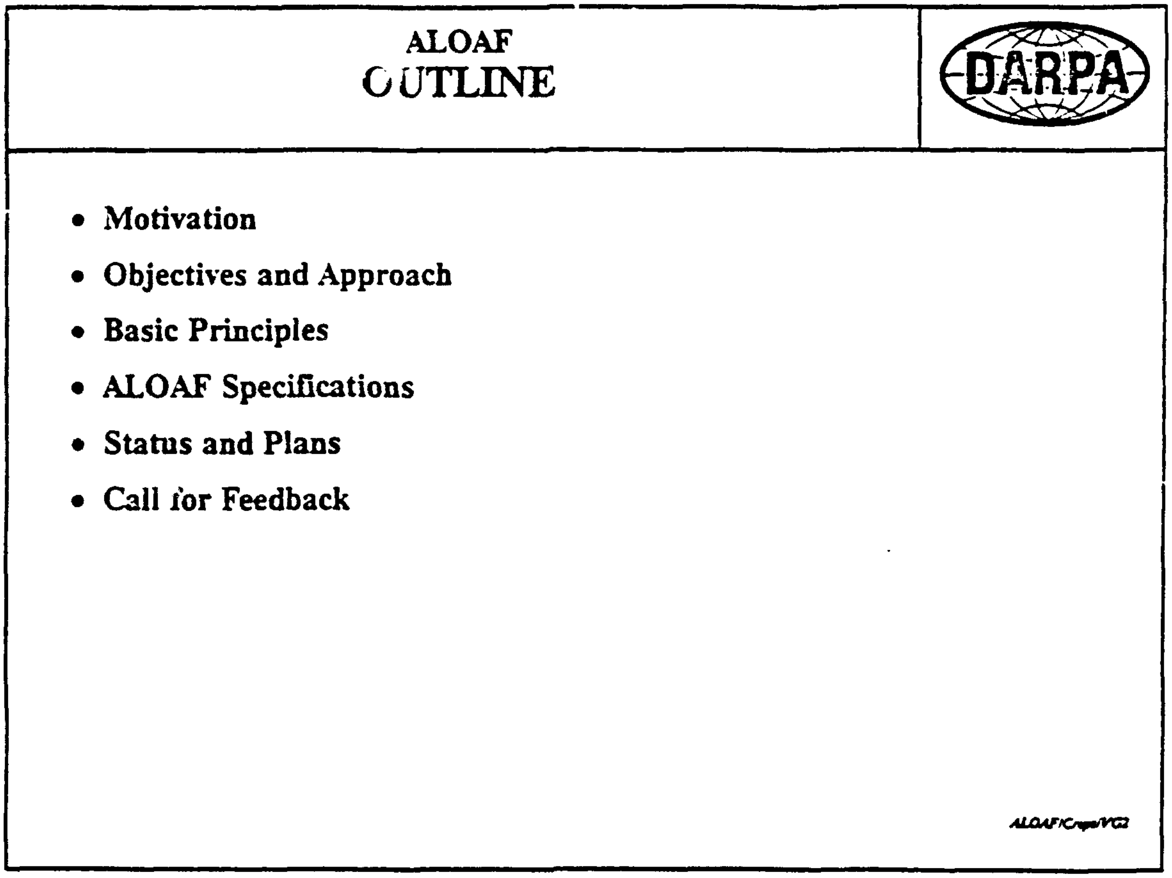
DARPA

ASSET LIBRARY OPEN ARCHITECTURE FRAMEWORK (ALOAF)

Dick Creps
Unisys Defense Systems, Inc.
4 December 1991
(703) 620-7100
creps@stars.reston.unisys.com

ALOAF/Creps/VG1

This presentation is about the Asset Library Open Architecture Framework, known as the ALOAF, that is being developed under STARS. The ALOAF is a set of interfaces that is being defined to facilitate interoperability between heterogeneous asset libraries and portability of tools across different asset library mechanisms.



ALOAF

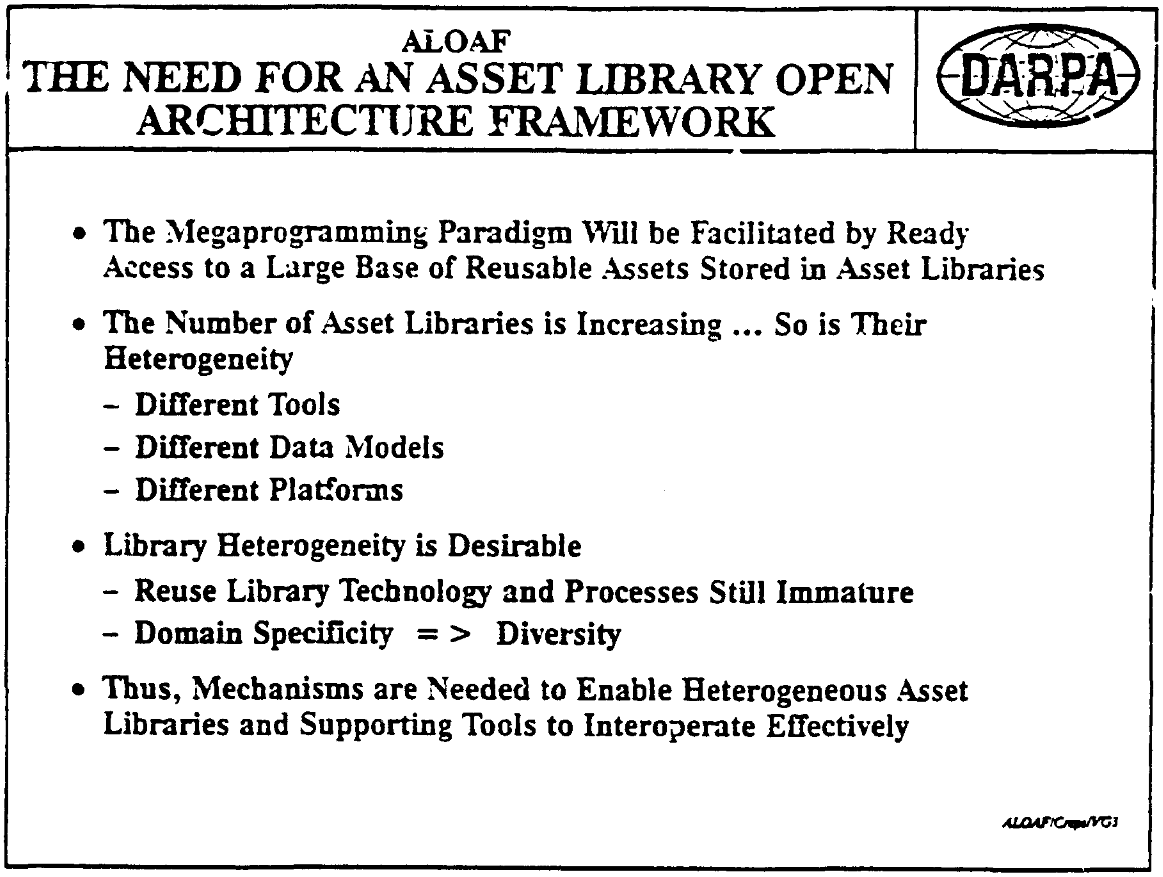
OUTLINE

DARPA

- Motivation
- Objectives and Approach
- Basic Principles
- ALOAF Specifications
- Status and Plans
- Call for Feedback

ALOAF/Crypo/VG2

This talk will first discuss why STARS believes an Asset Library Open Architecture Framework is needed. I: will then focus on the specific objectives of the ALOAF effort and describe the approach that is being taken. Some technical background information and some of the basic principles underlying the ALOAF will then be discussed, followed by a description of the ALOAF specifications defined to date, in significant detail. Next, the current status of the ALOAF effort and the plans for 1992 will be presented. In closing there will be a call for participation and feedback from STARS affiliates and the STARS community as a whole.



ALOAF
THE NEED FOR AN ASSET LIBRARY OPEN ARCHITECTURE FRAMEWORK

DARPA

- The Megaprogramming Paradigm Will be Facilitated by Ready Access to a Large Base of Reusable Assets Stored in Asset Libraries
- The Number of Asset Libraries is Increasing ... So is Their Heterogeneity
  - Different Tools
  - Different Data Models
  - Different Platforms
- Library Heterogeneity is Desirable
  - Reuse Library Technology and Processes Still Immature
  - Domain Specificity => Diversity
- Thus, Mechanisms are Needed to Enable Heterogeneous Asset Libraries and Supporting Tools to Interoperate Effectively

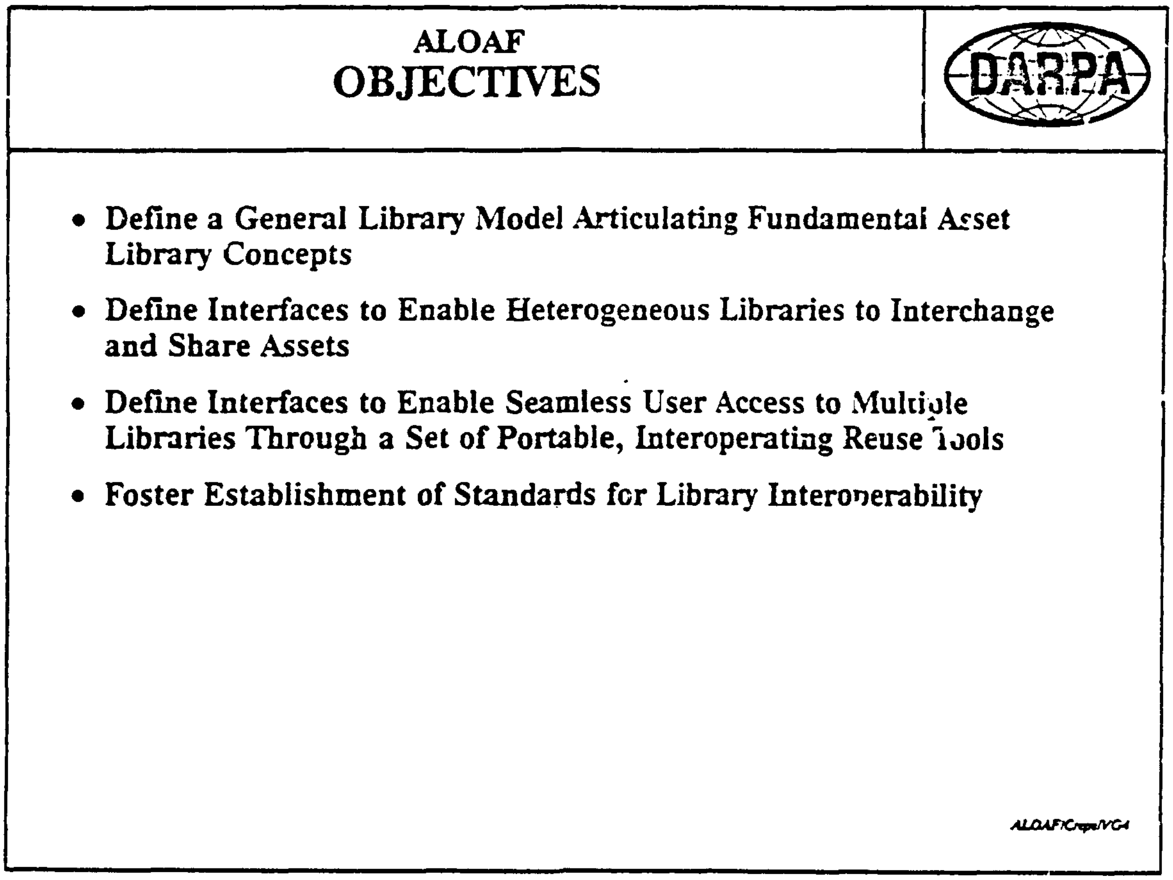
ALOAF/Cmp/VG1

The megaprogramming paradigm that STARS is addressing focuses strongly on the development of systems through the reuse of existing assets. This paradigm will be facilitated if software engineers have ready access to a large base of reusable assets stored in asset libraries.

Fortunately, the number of asset libraries capable of supporting megaprogramming is increasing. However, those libraries are proving to be quite heterogeneous for a variety of reasons, such as their use of different asset library mechanisms and tools, their dependency on different underlying hardware and operating systems, and their use of different data models (and even different styles of data models) to describe the assets they contain.

This degree of library heterogeneity is not necessarily bad, and is actually desirable at the present time. One of the chief reasons for this is that reuse library technology and associated reuse-related processes are still highly immature, and it is important at this time to conduct additional experimentation to assess the technology and acquire lessons learned before standardizing on some small number of specific approaches. Furthermore, one of the key trends in reuse libraries today is to emphasize a focused, domain-specific approach. This approach will necessarily promote heterogeneity because the specific library data models for different domains will naturally be quite diverse. In addition, it is important at this stage in the development of reuse technology to promote competition among technology development efforts to motivate needed advancements.

Considering that library heterogeneity is here to stay for the foreseeable future, it is clear that mechanisms are needed to enable the growing population of heterogeneous asset libraries and their supporting tools to interoperate effectively to satisfy megaprogrammer needs.



ALOAF

OBJECTIVES

- Define a General Library Model Articulating Fundamental Asset Library Concepts
- Define Interfaces to Enable Heterogeneous Libraries to Interchange and Share Assets
- Define Interfaces to Enable Seamless User Access to Multiple Libraries Through a Set of Portable, Interoperating Reuse Tools
- Foster Establishment of Standards for Library Interoperability

ALOAF/Cncpts/VG4

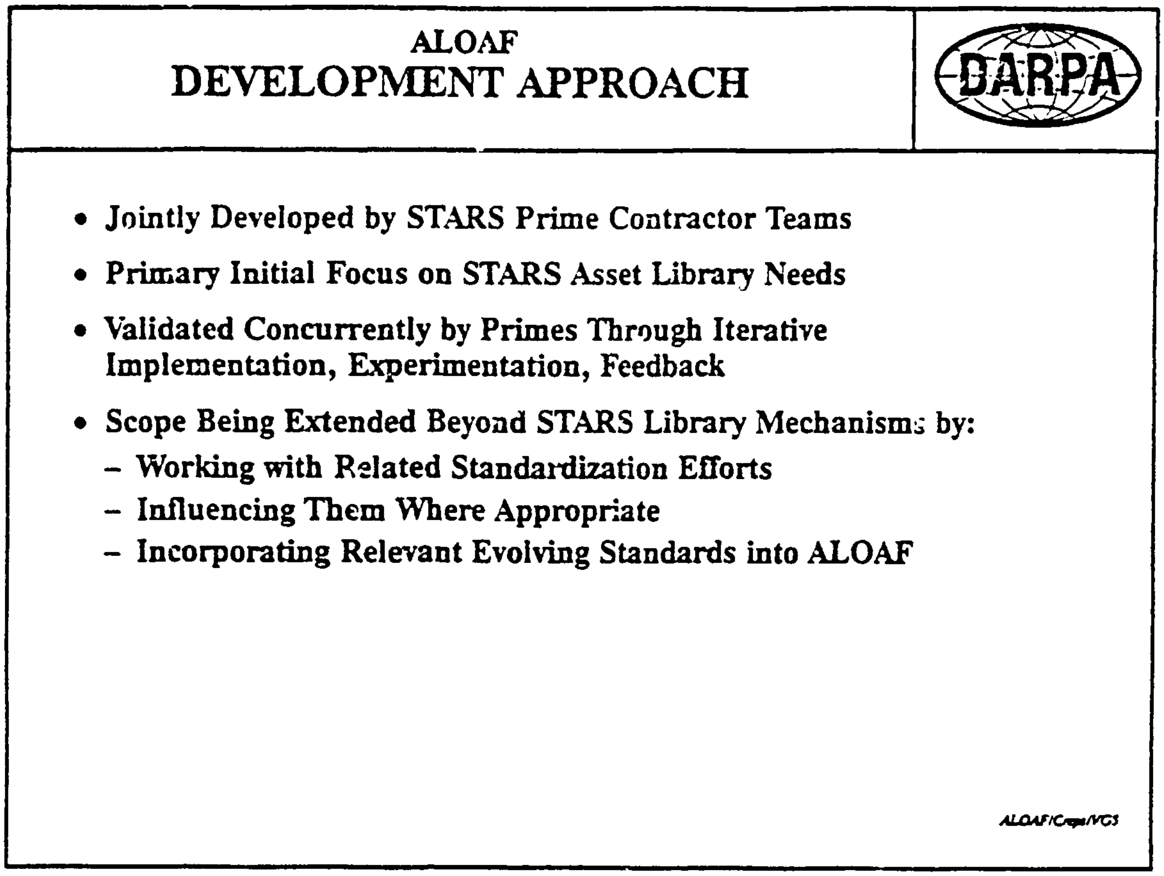
Four specific ALOAF objectives have been defined to address the identified needs.

First of all, we felt it necessary to define a general, conceptual model of an asset library, to help us articulate fundamental library concepts and properly define and scope ALOAF capabilities.

Secondly, we wish to define a set of interfaces to enable libraries to interchange asset descriptions and asset contents, and possibly to share single copies of asset data where appropriate.

In addition, we are striving to define a set of programmatic interfaces to enable seamless user access to a variety of asset libraries through a set of interoperating reuse tools that are portable across different asset library mechanisms.

Finally, since one of the key objectives of the STARS program as a whole is to promote development and adoption of software engineering standards, perhaps the most important ALOAF objective is to foster establishment of widely accepted standards for library interoperability.



ALOAF
DEVELOPMENT APPROACH

DARPA

- Jointly Developed by STARS Prime Contractor Teams
- Primary Initial Focus on STARS Asset Library Needs
- Validated Concurrently by Primes Through Iterative Implementation, Experimentation, Feedback
- Scope Being Extended Beyond STARS Library Mechanisms by:
  - Working with Related Standardization Efforts
  - Influencing Them Where Appropriate
  - Incorporating Relevant Evolving Standards into ALOAF

ALOAF/Cmps/VG5

This slide describes the overall approach that STARS is taking to develop the ALOAF.

The ALOAF is being jointly developed by representatives from each of the STARS prime contractor teams and MITRE Corporation. This joint approach will ensure that a broad spectrum of views is reflected in the document and that the needs of all the primes are adequately addressed. As this implies, the initial focus of the ALOAF effort has been on STARS' perceived asset library needs.

A key element of the approach is that the STARS contractors, who are also tasked to develop and integrate asset library mechanisms, are an ideal testbed for the ALOAF results. The ALOAF will thus be validated by the primes, concurrent with ALOAF development and evolution, through an iterative cycle of implementing the ALOAF mechanisms, experimenting with and assessing the mechanisms, and providing feedback to the ALOAF team to enhance and evolve the mechanisms.

However, in accord with our objective to foster the establishment of standards for library interoperability, we will also strive to extend the scope of the ALOAF beyond just the STARS library mechanisms. Our approach here is to work closely with related standardization efforts such as the Reuse Library Interoperability Group (RIG), influence those efforts where appropriate, and incorporate the relevant emerging standards back into the ALOAF.

ALOAF
DEVELOPMENT STRATEGY

- Short-Term:
  - Focus on Asset Interchange Between Libraries, in Terms of a Simple Common Data Model for Describing Assets
  - Validate Initial Asset Interchange Through Experiments With STARS Asset Library Mechanisms
- Long-Term:
  - Focus on Seamless Library/Tool Interoperability and Generalized Asset Interchange
  - Define Extensive Set of Asset Library Services
    - Provided by ALOAF-Compliant Servers
    - Accessed Directly by Client Tools Distributed Across a Network
  - Establish Interfaces for Generalized Description and Interchange of Library Data

ALOAF/C[illegible]

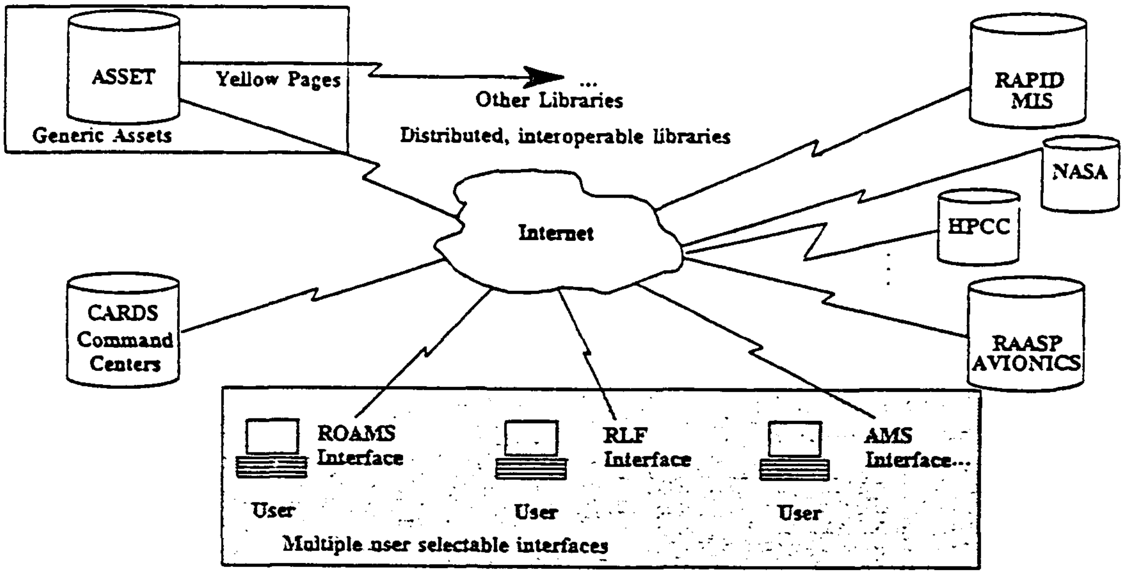
STARS is focusing on both a short-term and a long-term strategy for ALOAF development and validation.

The short-term strategy, designed to take advantage of existing technology and provide a basic library interoperability capability that can be employed now, focuses on the interchange of assets between libraries. This interchange is facilitated by a Common Data Model for describing the general characteristics of assets in a common form. An initial experiment is currently under way to assess and validate this approach by using the STARS asset library mechanisms, and additional such experiments will be undertaken as the ALOAF evolves.

The long-term strategy focuses on the notion of "seamless" interoperability of libraries and their supporting tools, as well as on generalizing the asset interchange capabilities. In a seamless environment, the library user will have network access to a variety of libraries that all appear to be relatively homogeneous, with the probable exception of their data models. This will be effected by defining a set of standard asset library services accessible from ALOAF-compliant servers by client tools distributed across a network, via some form of client-server protocol. Libraries will continue to interoperate via asset interchange, as well, but the interchange capabilities will be generalized to enable libraries to interchange a much richer set of information than can be represented by the Common Data Model.

ALOAF
DISTRIBUTED LIBRARY CONCEPT

DARPA

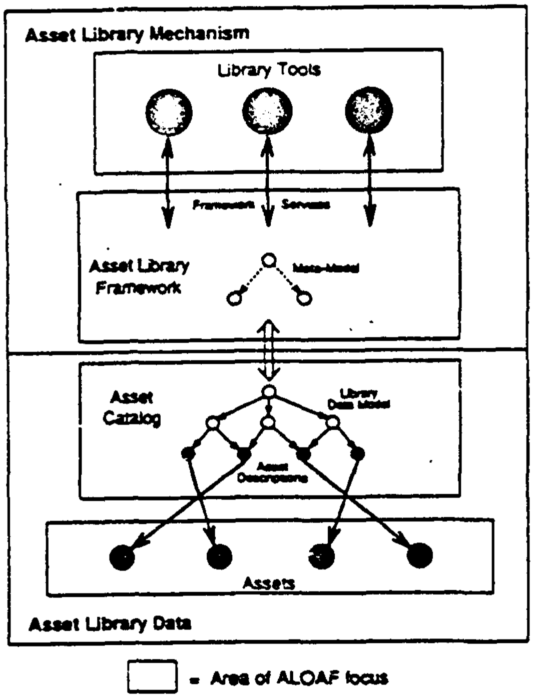


ALOAF/Cmpt/VG7

This figure may help crystallize the ideas discussed in the preceding slide. We see here a somewhat futuristic view in which there are numerous asset libraries all interconnected via the Internet. Many of them, such as CARDS, RAPID, RAASP,and so on, are libraries focusing on specific application domains, while other libraries, such as ASSET, provide access to components with more general applicability. Some of the libraries, such as ASSET as depicted here, may provide "Yellow Pages" services to refer (or possibly directly connect) users to libraries containing specific assets or classes of assets.

Within this distributed library context, there are, in the shaded area, a collection of client tools providing distinct user interfaces, such as those provided by ROAMS, RLF, AMS, and so on. This is meant to indicate that a user at any particular site could choose whichever user interface is most suitable and be able to access each of the libraries with ALOAF-compliant servers through that user interface in a natural manner. However, even in a futuristic scenario, not all libraries will be expected to have ALOAF-compliant servers, but these libraries will still be able to interoperate with other libraries via asset interchange.

ALOAF
ASSET LIBRARY MODEL

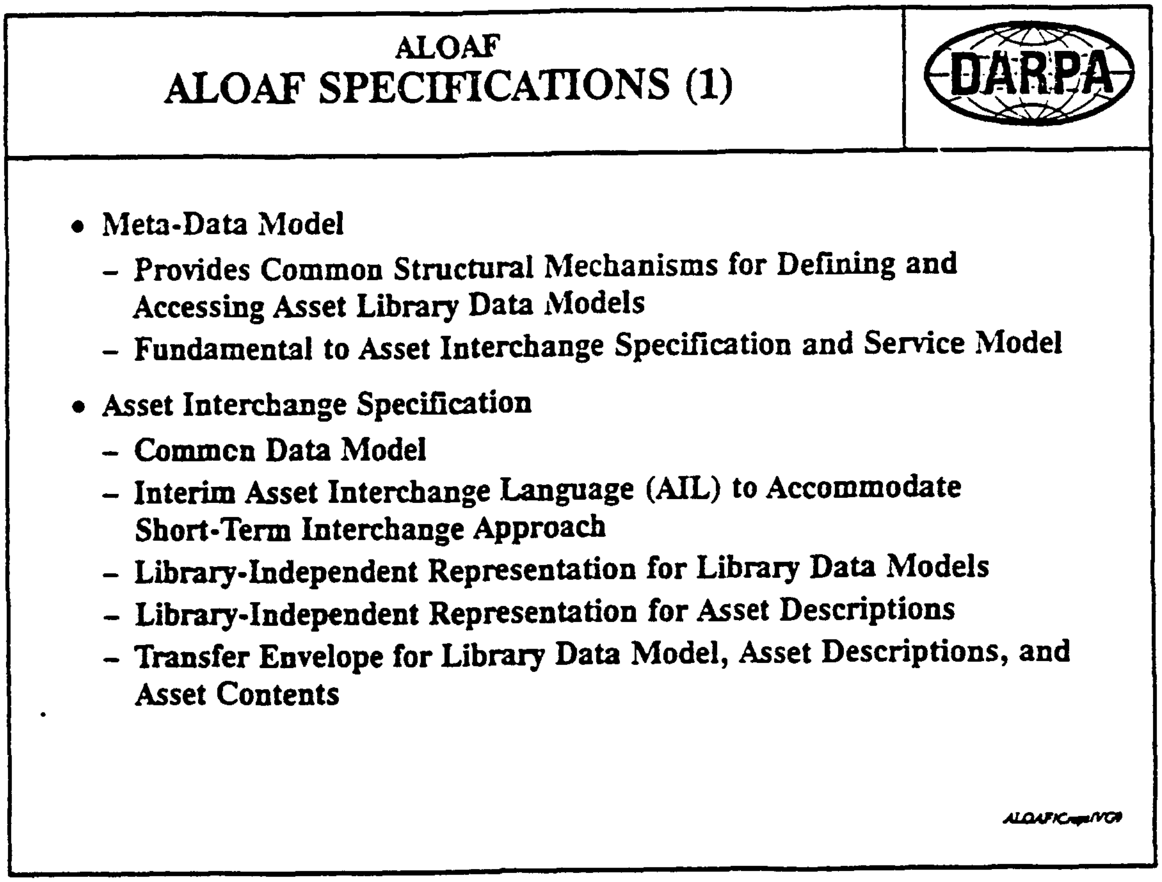


ALOAF/Cmpt/VG1

This figure is a high-level conceptual view of the general asset library model we have defined. In this view, an Asset Library consists of two major components: the Asset Library Mechanism that provides the general functionality of the library, and the Asset Library Data that includes both descriptions of the library contents as well as the contents themselves.

The Asset Library Mechanism contains an Asset Library Framework that provides Framework Services to a set of Library Tools through which a variety of library users (e.g., asset reusers, library administrators, asset certifiers) interact with the library. The Asset Library Mechanism encapsulates an implicit meta-data model which is realized through the Framework Services to enable the definition cf a Library Data Model. The Library Data Model defines the structure of the Asset Descriptions that describe individual assets. The Library Data Model and the Asset Descriptions together are referred to as the Asset Catalog. The Asset Descriptions point to the files or objects in the underlying SEE that constitute the actual Assets.

The shaded area in the middle of the figure indicates the area of ALOAF focus. The ALOAf is focusing on defining Framework Services to enable the definition, manipulation, and interchange of Library Data Models and Asset Descriptions. The Assets (i.e., the contents) are considered to be outside the scope of the ALOAF because they are best managed and accessed using underlying SEE services.



ALOAF

ALOAF SPECIFICATIONS (1)

- Meta-Data Model
  - Provides Common Structural Mechanisms for Defining and Accessing Asset Library Data Models
  - Fundamental to Asset Interchange Specification and Service Model
- Asset Interchange Specification
  - Commcn Data Model
  - Interim Asset Interchange Language (AIL) to Accommodate Short-Term Interchange Approach
  - Library-Independent Representation for Library Data Models
  - Library-Independent Representation for Asset Descriptions
  - Transfer Envelope for Library Data Model, Asset Descriptions, and Asset Contents

ALOAF/C[illegible]/VG

The next three viewgraphs provide a more detailed overview of the ALOAF specifications we are defining.

At the core of the specifications is a meta-data model which defines a set of common structural mechanisms for defining and accessing individual asset library data models. Specifically, the meta-model enables the definition of data models in the form of class hierarchies, wherein each class can be assigned attributes and participate in relationships, and the attributes and relationships are inherited down the hierarchy. The meta-data model is fundamental to both the Asset Interchange Specification and the Service Model (another portion of the ALOAF specifications, described in a subsequent slide), because both of these aspects of the ALOAF rely heavily on a common method for describing and manipulating library data models.

The Asset Interchange Specification will provide a means for interchanging assets between libraries by describing the assets and associated data models using a set of common formats. The short-term approach to asset interchange required that two things be defined:

- o A Common Data Model (CDM) allowing assets to be described in terms of a set of general asset characteristics that are typically used to describe assets in modern asset libraries.
- o An Asset Interchange Language (AIL) for describing assets textually in terms of the CDM. AIL specification are imported and exportea by libraries to effect asset interchange. The AIL is an interim language that will be superceded by the long-term asset interchange solution.

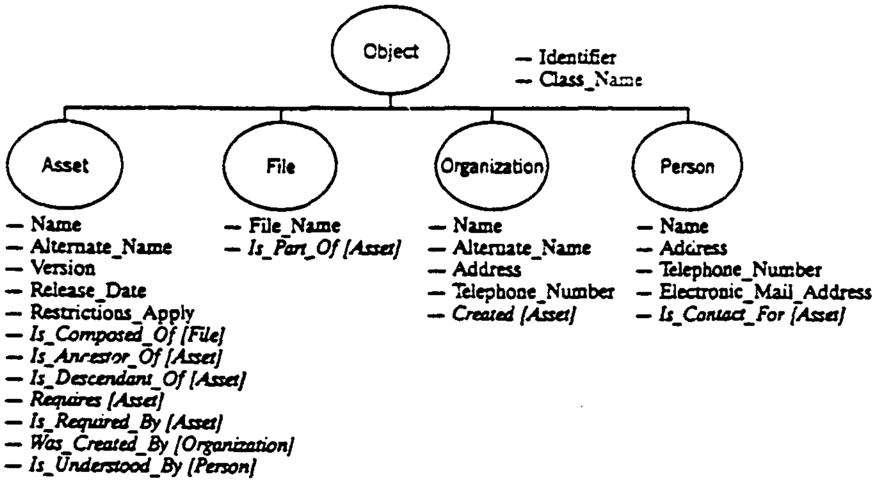
The long-term approach to asset interchange will require the following:

- o A library-independent textual notation for representing library data models in terms of a common meta-data model
- o A library-independent textual notation for representing asset descriptions in terms of a particular library data model
- o A transfer envelope to encapsulate library data model specifications, asset description specifications, and asset contents

The ALOAF is assessing a number of existing standards, such as CDIF, IRDS, and SGML, for their applicability to these latter three needs.

ALOAF

ALOAF SPECIFICATIONS (2)
COMMON DATA MODEL



- Identifier
- Class_Name

- Name
- Alternate_Name
- Version
- Release_Date
- Restrictions_Apply
- *Is_Composed_Of [File]*
- *Is_Ancestor_Of [Asset]*
- *Is_Descendant_Of [Asset]*
- *Requires [Asset]*
- *Is_Required_By [Asset]*
- *Was_Created_By [Organization]*
- *Is_Understood_By [Person]*

- File_Name
- *Is_Part_Of [Asset]*

- Name
- Alternate_Name
- Address
- Telephone_Number
- *Created [Asset]*

- Name
- Address
- Telephone_Number
- Electronic_Mail_Address
- *Is_Contact_For [Asset]*

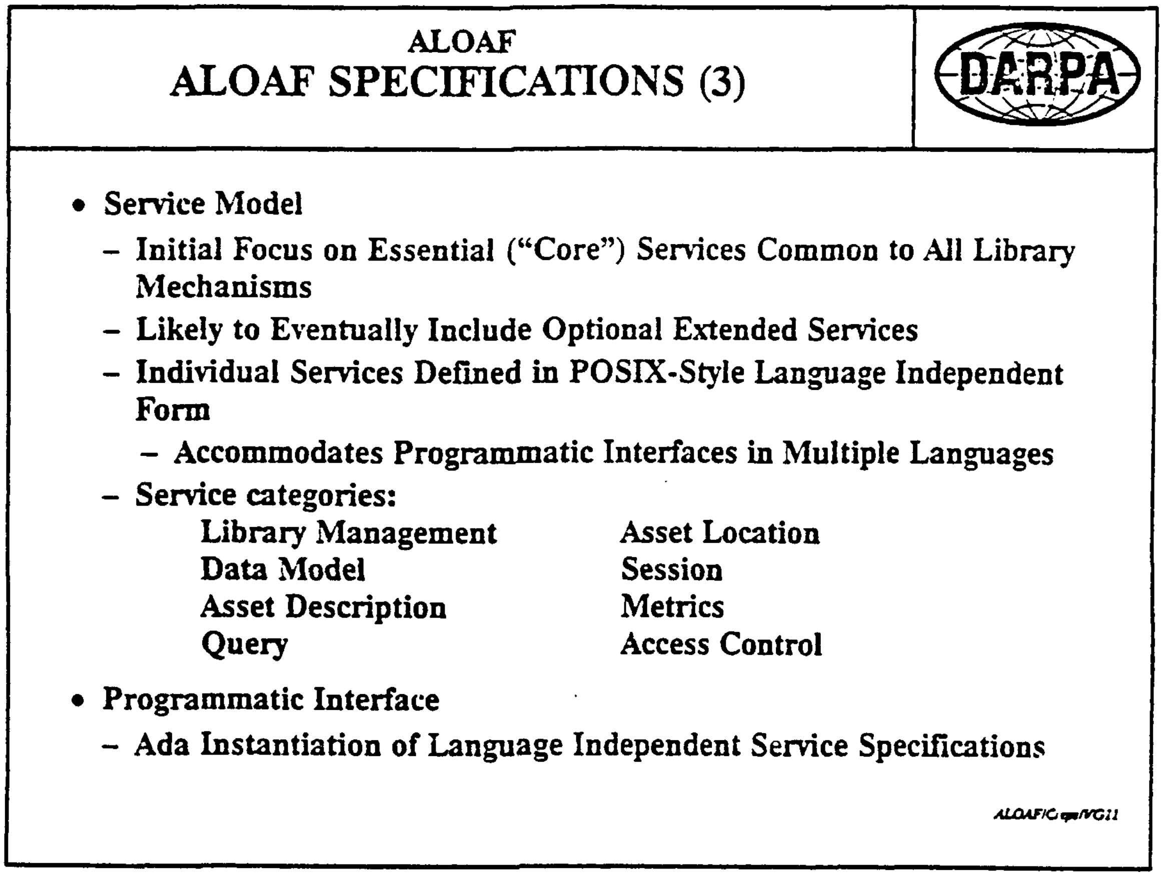
ALOAF/Crsp/VG10

This is a pictorial representation of the ALOAF Common Data Model (CDM). It consists of a rather simple class hierarchy, with a class called Object at the root of the hierarchy. There are a couple of attributes used for essentially bookkeeping purposes defined in the Object class, and these attributes are inherited by the other classes in the model.

The heart of the Common Data Model is the Asset class. It has a variety of attributes to describe various general properties of assets. The attributes at the top of the list, in the standard font, have integer or string values. The attributes in italics represent relationships emanating from the class, with the target class of each relationship noted in brackets.

The Organization, and Person classes are intended to supplement the Asset class by providing information about the organizations and people who created or are points of contact for assets. The File class exists to represent information about the files that constitute the "contents" of an asset.

Using the Common Data Model, assets are represented as objects that are instances of CDM classes. A typical asset might be represented by a single Asset object, one (or possibly more) Organization objects, one (or possibly more) Person objects, and one or more File objects, all related to one another appropriately through the CDM relationship attributes.



ALOAF

ALOAF SPECIFICATIONS (3)

- Service Model
  - Initial Focus on Essential ("Core") Services Common to All Library Mechanisms
  - Likely to Eventually Include Optional Extended Services
  - Individual Services Defined in POSIX-Style Language Independent Form
    - Accommodates Programmatic Interfaces in Multiple Languages
  - Service categories:
    - Library Management
    - Data Model
    - Asset Description
    - Query
    - Asset Location
    - Session
    - Metrics
    - Access Control
- Programmatic Interface
  - Ada Instantiation of Language Independent Service Specifications

ALOAF/Cqs/VG11

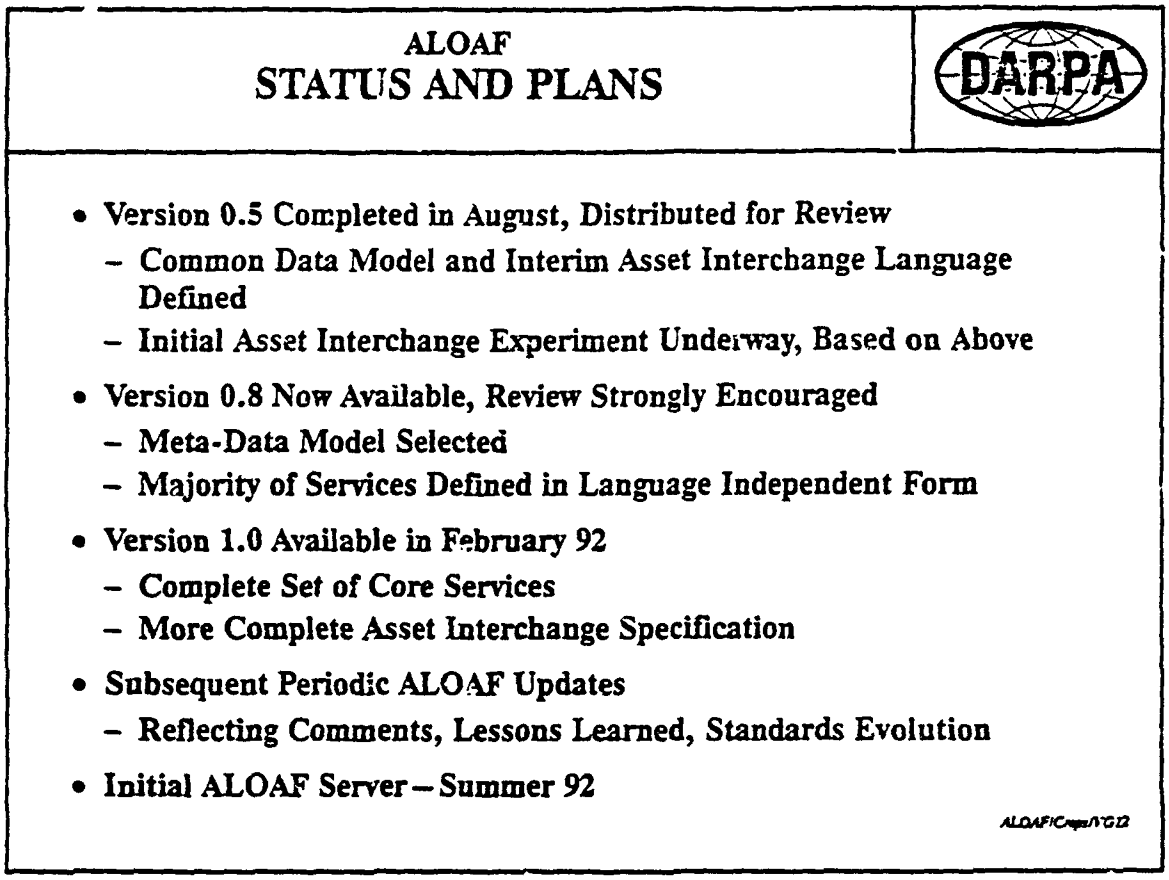
The ALOAF Service Model is a collection of asset library framework services that are provided programmatically or through a client/server communications protocol. These services are intended to be common among ALOAF conformant library mechanisms and thus to enable library tool portability between library mechanisms. The focus of the ALOAF Service Model activities to date has been on a set of "core" services representing capabilities that are considered essential to asset library operation. In the future, it is envisioned that the Service Model will include a variety of optional extended services.

The services are being defined in a programming-language-independent form that is very similar to that in which the POSIX services are defined. This specification approach does not bias the services toward any specific programming language and thus should readily accommodate ALOAF programmatic interfaces in multiple languages. However, in addition to the language-independent interfaces, the ALOAF will eventually include an Ada instantiation of those interfaces for implementation within STARS.

The services have been decomposed into eight specific categories:

- o Library Management services, to enable the overall management of asset libraries and their high level contents
- o Data Model services, to enable the creation and manipulation of library data models in accordance with the ALOAF meta-model
- o Asset Description services, to enable the creation and manipulation of asset descriptions in accordance with a particular library data model
- o Query services, to enable asset descriptions to be queried in terms of a particular library data model
- o Asset Location services, to provide information needed to access asset contents via underlying SEE services
- o Session services, to control the initiation and termination of sessions during which ALOAF services are accessed
- o Metrics services, to enable the collection, manipulation, and extraction of library usage metrics
- o Access Control services, to control user access to particular library elements

Another potential ALOAF specification that is not included in this slide is a detailed client-server communications protocol that would be used by ALOAF client applications across a network to obtain services from ALOAF servers. A future version of the ALOAF is likely to include such a specification.



ALOAF

STATUS AND PLANS

- Version 0.5 Completed in August, Distributed for Review
  - Common Data Model and Interim Asset Interchange Language Defined
  - Initial Asset Interchange Experiment Underway, Based on Above
- Version 0.8 Now Available, Review Strongly Encouraged
  - Meta-Data Model Selected
  - Majority of Services Defined in Language Independent Form
- Version 1.0 Available in February 92
  - Complete Set of Core Services
  - More Complete Asset Interchange Specification
- Subsequent Periodic ALOAF Updates
  - Reflecting Comments, Lessons Learned, Standards Evolution
- Initial ALOAF Server – Summer 92

ALOAF/C[illegible]

This slide provides information about the current status of the ALOAF and our plans for its future development, particularly during 1992.

Version 0.5 of the document was completed in August and was distributed to a limited set of individuals for review. Version 0.5 included the Common Data Model (CDM) and the interim Asset Interchange Language (AIL) to facilitate initial asset interchange experimentation. An initial asset interchange capability has been developed based on the CDM and AIL; that capability was demonstrated at TRI-Ada and an improved version is being demonstrated here at STARS '91.

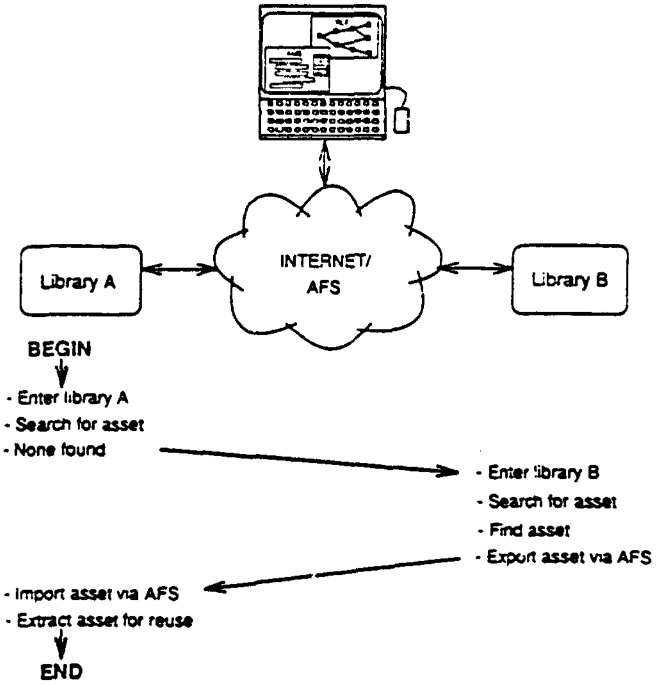
Version 0.8 of the document is now available for STARS '91 attendees to take home with them. We strongly encourage those who do take a copy to review it and provide us with comments. In Version 0.8, the initial ALOAF meta-data model is defined, and a majority of the services in the Service Model have been defined in programming-language-independent form.

ALOAF Version 1.0 will be available in February 1992. It is expected to specify all the core services. It should also contain a more complete Asset Interchange Specification, including either the library data model representation, the asset description representation, or both. Version 1.0 will be considered the baseline against which initial ALOAF service implementation will be performed.

Beyond Version 1.0, the document will be updated periodically to reflect both internal and external comments, lessons learned through ALOAF implementation and experimentation, and evolution of related standards. Among the first things to be done after Version 1.0 will be the specification of Ada service interfaces.

SAIC, under subcontract to IBM, will be producing the initial ALOAF implementation in the Summer 1992 timeframe. This implementation will be an ALOAF server, providing service access via a client-server communications protocol.

ALOAF
ASSET INTERCHANGE EXPERIMENT



BEGIN

- Enter library A
- Search for asset
- None found

- Enter library B
- Search for asset
- Find asset
- Export asset via AFS

- Import asset via AFS
- Extract asset for reuse

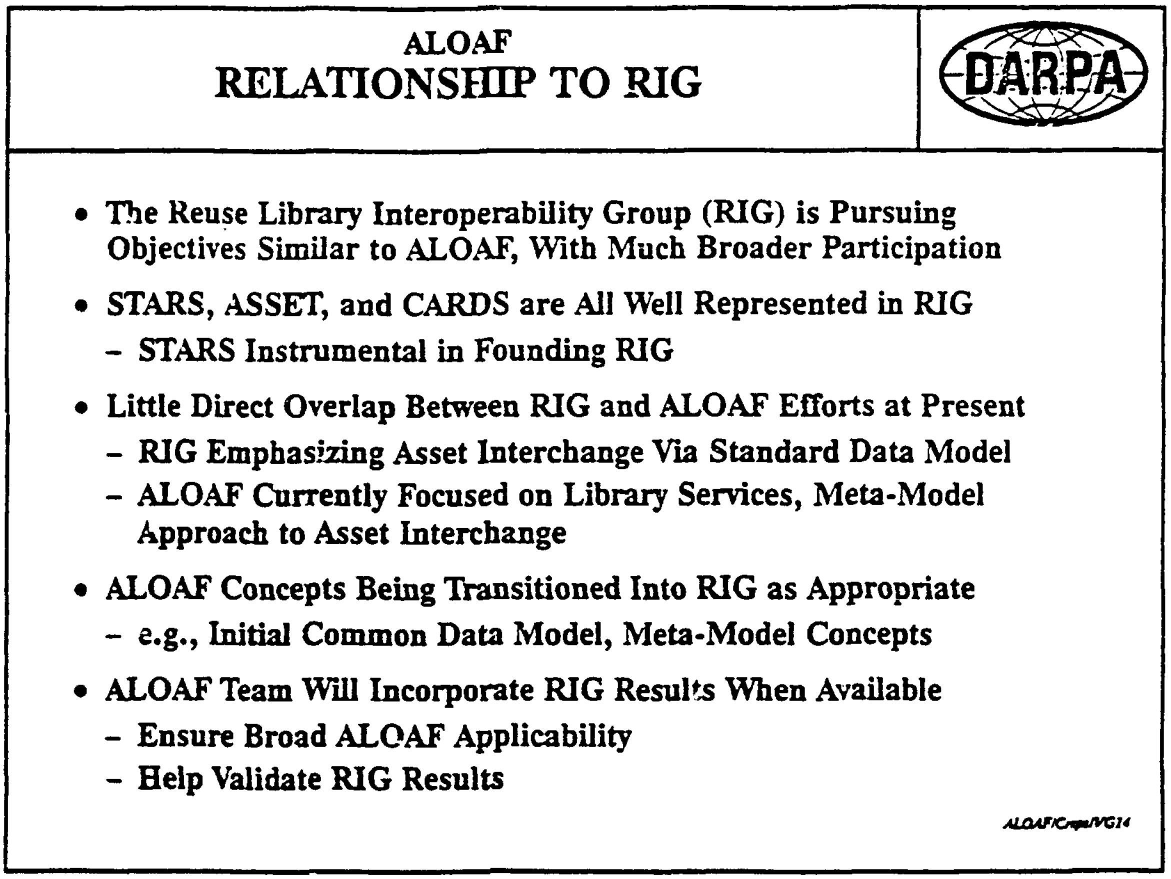
END

ALOAF/Cmpt/VG13

As noted in the previous slide, we have been conducting an initial asset interchange experiment based on the Common Data Model (CDM) and Asset Interchange Language (AIL). The Unisys and IBM asset library mechanisms have been modified to export and import asset descriptions using the AIL. This figure illustrates the basic interchange scenario that is being used.

At the top we see a workstation having access to both a Unisys RLF library and an IBM AMS library via the Internet. (Note that the workstation and library sites also have access to AFS, the wide-area network file system product from Transarc Corporation. AFS enables all sites to directly access a common set of files across the Internet.) In this scenario, a user (for example, a library administrator), searches for a particular asset or class of assets in Library A, but finds none. Thinking it desirable to have such a class of assets installed in Library A (perhaps because Library A is at the user's local site), the user accesses Library B and conducts an analogous search in that library. Upon finding a suitable asset, the user exports the asset by generating an AIL specification describing the asset and placing it in an AFS file (the files constituting the contents of the asset are also accessible via AFS). The user then switches back to Library A and imports the asset using the AIL spec, thus making the asset directly available to other users of Library A.

We have found this style of asset interchange to be useful, but not completely satisfying, primarily because the Common Data Model describes only general characteristics of assets, rather than their detailed domain-specific characteristics. However, extending the Common Data Model to accommodate all domains would render it enormous, unwieldy, and very difficult to understand. What is needed is a capability to describe individual library data models in accordance with a particular meta-data model, so that exported assets can be described in terms of their native data models. This will provide the human performing asset import with a more complete and focused set of information about the asset to facilitate some of the more difficult aspects of asset import, such as asset classification. Such an approach is highly consistent with the long-term ALOAF asset interchange strategy.



ALOAF

RELATIONSHIP TO RIG

- The Reuse Library Interoperability Group (RIG) is Pursuing Objectives Similar to ALOAF, With Much Broader Participation
- STARS, ASSET, and CARDS are All Well Represented in RIG
  - STARS Instrumental in Founding RIG
- Little Direct Overlap Between RIG and ALOAF Efforts at Present
  - RIG Emphasizing Asset Interchange Via Standard Data Model
  - ALOAF Currently Focused on Library Services, Meta-Model Approach to Asset Interchange
- ALOAF Concepts Being Transitioned Into RIG as Appropriate
  - e.g., Initial Common Data Model, Meta-Model Concepts
- ALOAF Team Will Incorporate RIG Results When Available
  - Ensure Broad ALOAF Applicability
  - Help Validate RIG Results

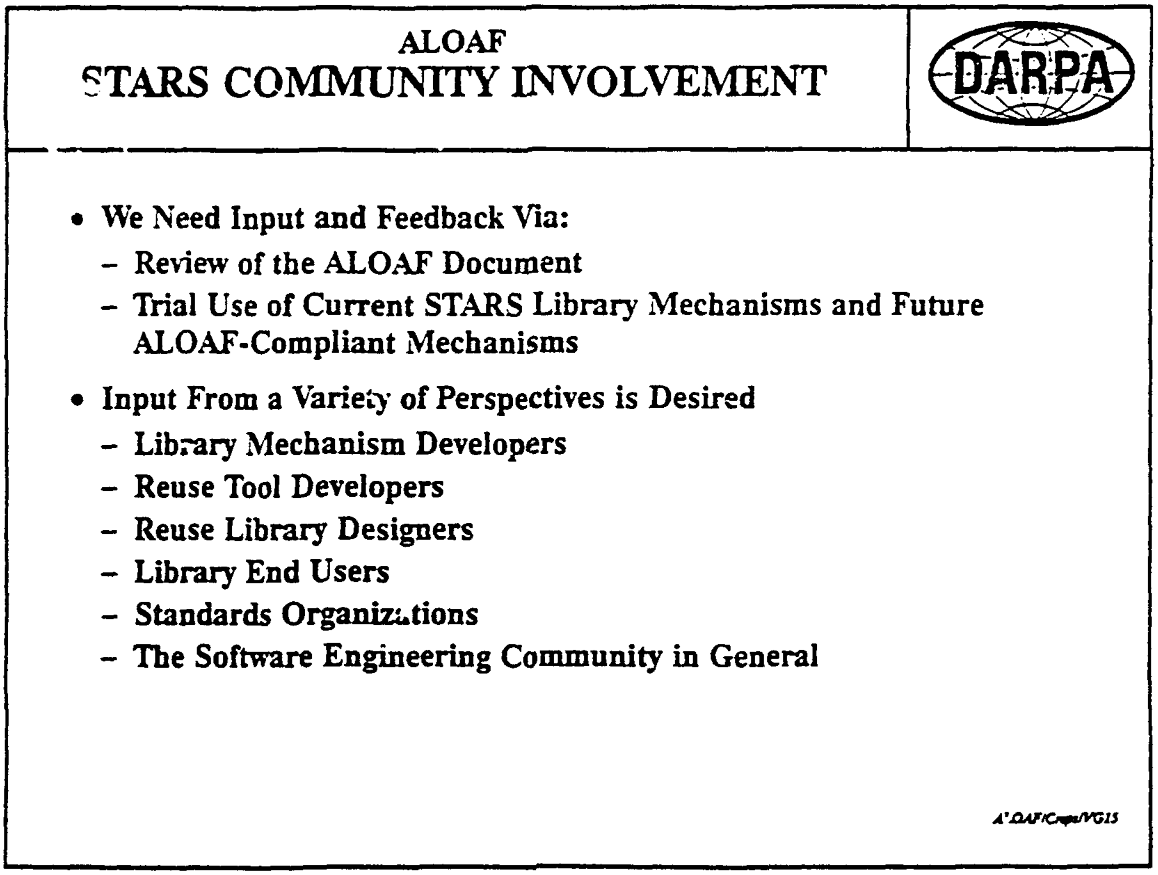
ALOAF/Crps/VG14

A key aspect of the ALOAF approach to foster standards for library interoperability is the STARS relationship with the Reuse Library Interoperability Group (RIG). The RIG is a pre-standards organization that is pursuing objectives generally similar to those of the ALOAF. However, the RIG features a much broader base of participation, with over 20 organizations currently involved. You are encouraged to pick up a copy of the RIG press release here at STARS '91 to learn more about the organization.

STARS, ASSET, and CARDS are among the organizations that are actively participating in RIG activities. Several individuals with direct STARS affiliations are working actively in RIG technical subcommittees. Also, it is worth noting that STARS (including ASSET) was instrumental in founding the RIG earlier this year.

At the present time, there is little direct overlap in the work that the RIG and the ALOAF are each actively pursuing. The RIG is currently emphasizing asset interchange via a standard data model, although they are considering meta-model issues relating to asset interchange, as well. The ALOAF has already visited the issue of asset interchange via the Common Data Model and has conducted some live experiments, but the ALOAF efforts appear less ambitious than the RIG's in this area. On the other hand, the ALOAF is now actively developing an extensive set of library services and is moving forward relatively aggressively on meta-model based asset interchange.

The two efforts thus appear to be complementary at this time, with the ALOAF pushing the frontiers of library interoperability and offering its results for RIG consideration, while the RIG is playing the more conservative role of identifying areas that are ready for standardization and fleshing out candidate standards in those areas. ALOAF team members will continue to work directly with the RIG to transition ALOAF results as appropriate, and will also be poised to incorporate RIG results back into the ALOAF when they become available. In this latter role, STARS will ensure that ALOAF applicability extends beyond just the STARS library mechanisms, and will also act as a testbed to help validate RIG results.



ALOAF

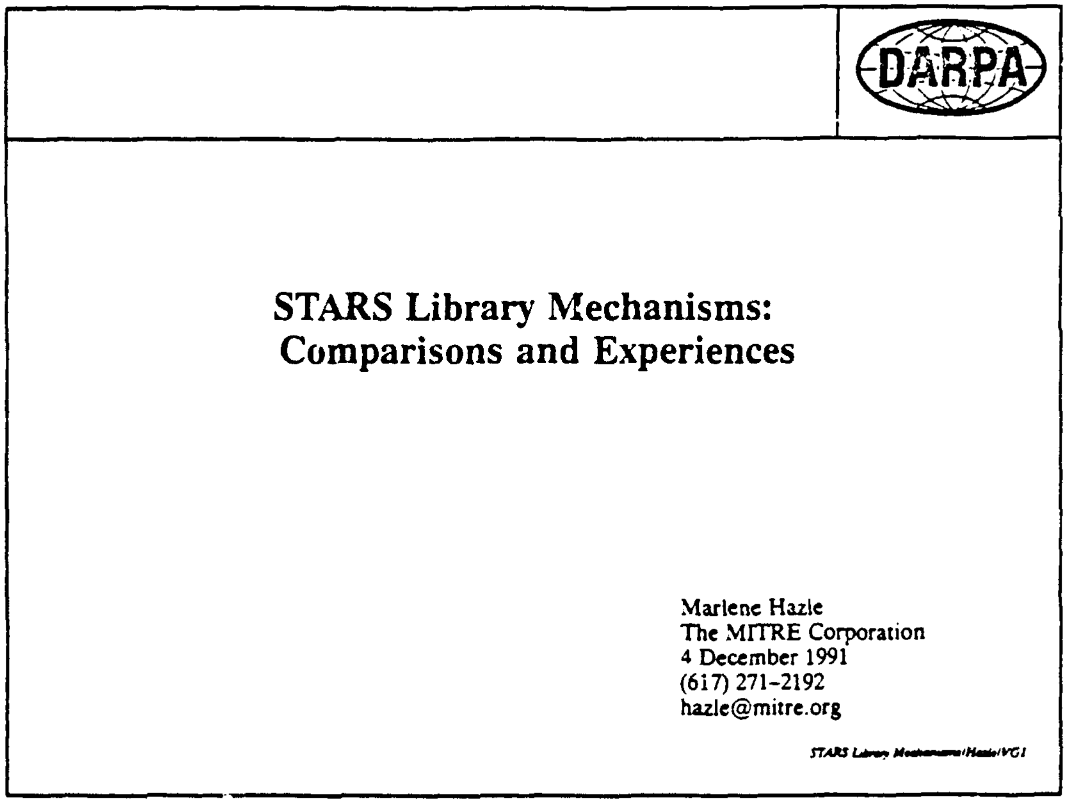
STARS COMMUNITY INVOLVEMENT

- We Need Input and Feedback Via:
  - Review of the ALOAF Document
  - Trial Use of Current STARS Library Mechanisms and Future ALOAF-Compliant Mechanisms
- Input From a Variety of Perspectives is Desired
  - Library Mechanism Developers
  - Reuse Tool Developers
  - Reuse Library Designers
  - Library End Users
  - Standards Organizations
  - The Software Engineering Community in General

ALOAF/Cmpt/VG15

In closing, I would like to extend a call for you and your organizations to contribute to the ALOAF effort by becoming directly involved as TT affiliates or simply by taking the time to review the ALOAF document and provide us with comments. We need input and feedback in a variety of forms, ranging from review of the document to trial use of the current STARS library mechanisms with asset interchange capabilities and the future fully ALOAF-compliant library mechanisms.

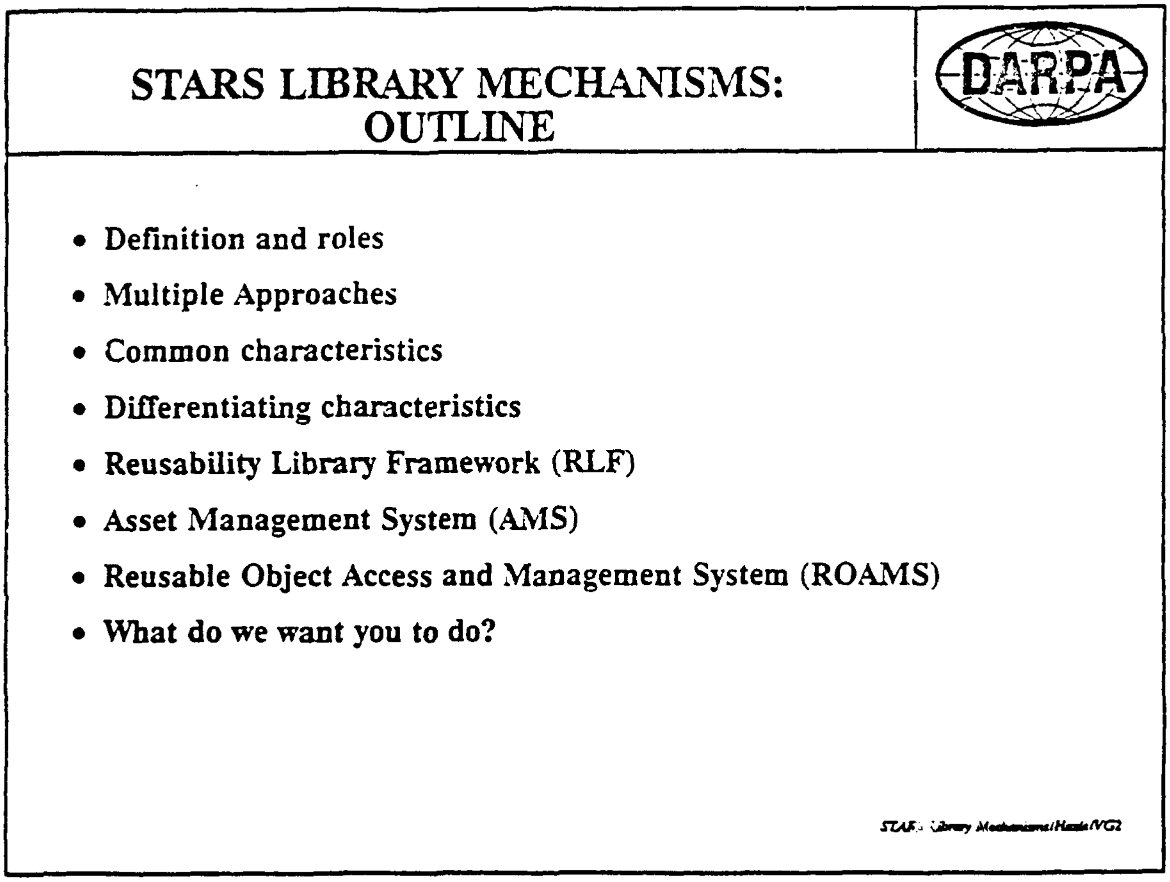
Also, we would like feedback on the ALOAF from a variety of perspectives, including library mechanism developers, developers of ALOAF client tools. reuse library designers and data modelers, library end users impacted by asset interchange and seamless interoperability, related standards organizations, and the software engineering community in general, which is grappling with similar but more complex issues in the area of SEE frameworks.



STARS Library Mechanisms: Comparisons and Experiences

Marlene Hazle
The MITRE Corporation
4 December 1991
(617) 271-2192
hazle@mitre.org

STARS Library Mechanisms/Hazle/VG1



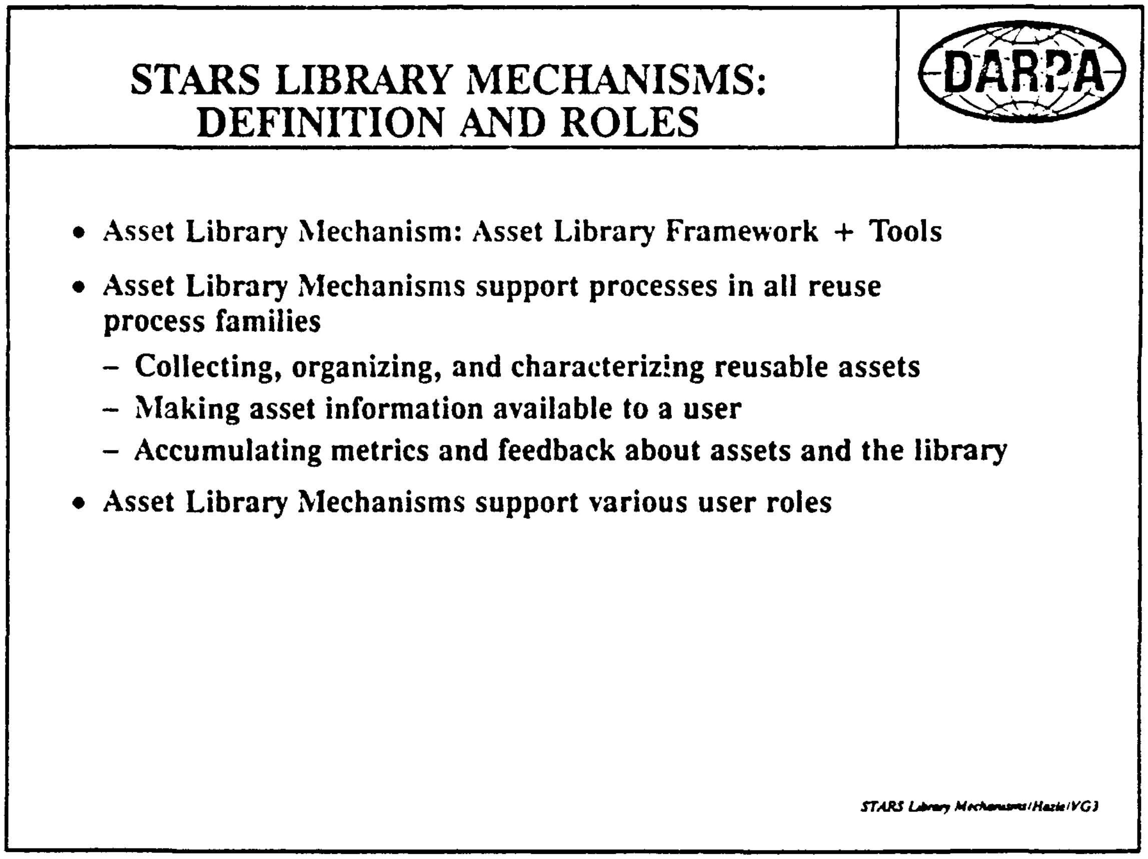
STARS LIBRARY MECHANISMS: OUTLINE

- Definition and roles
- Multiple Approaches
- Common characteristics
- Differentiating characteristics
- Reusability Library Framework (RLF)
- Asset Management System (AMS)
- Reusable Object Access and Management System (ROAMS)
- What do we want you to do?

STARS Library Mechanisms/Hasle/VG2

By this time in STARS'91, and indeed in the STARS program, I hope that you know something about the STARS library mechanisms. If you have not yet seen the current versions demonstrated here, I encourage you to do so.

My purpose is not to describe the mechanisms in detail, but rather to talk about the role of library mechanisms in our concept of reuse-based development, why we have multiple mechanisms, and the common characteristics among the STARS library mechanisms. Then for each of the mechanisms I will discuss the characteristics that differentiate among them. I will also present summary information about the use of the mechanisms thus far and the material available to support their further use and evaluation by STARS affiliates.



STARS LIBRARY MECHANISMS: DEFINITION AND ROLES

DARPA

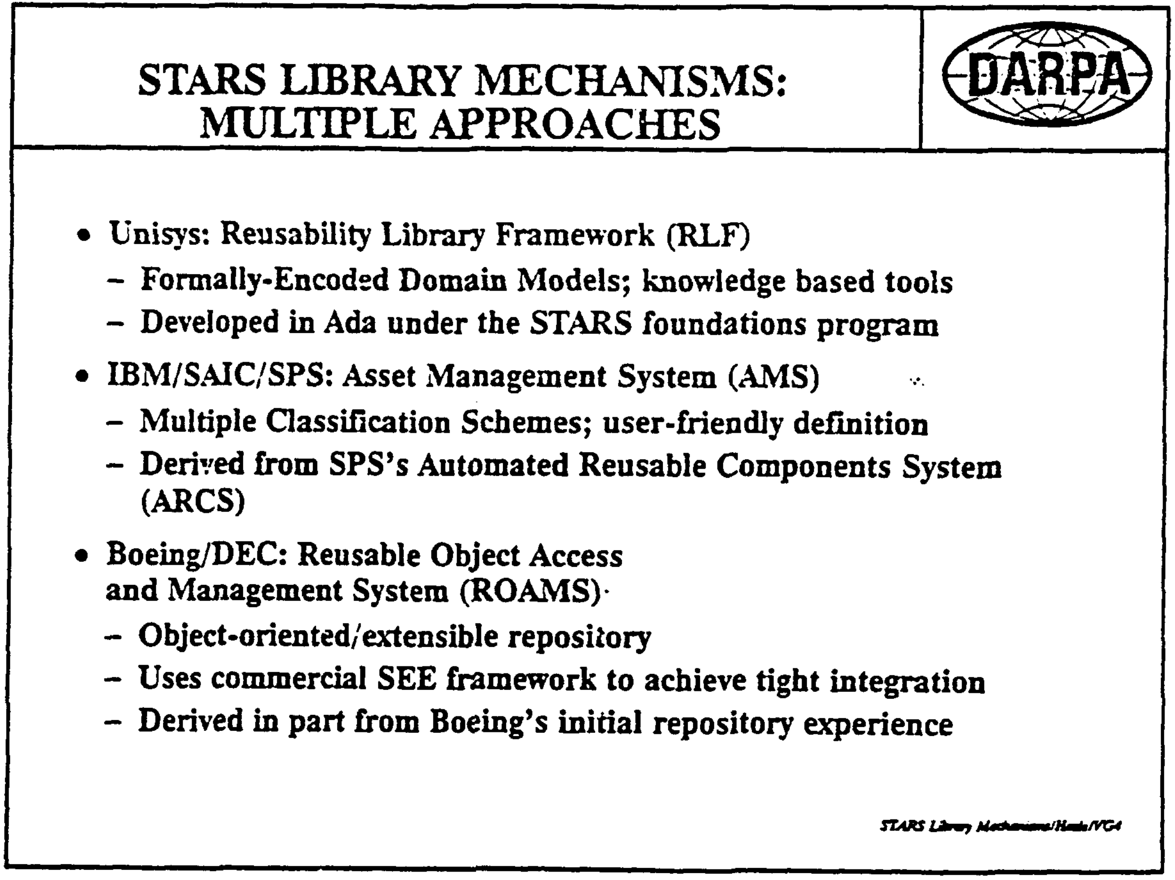
- Asset Library Mechanism: Asset Library Framework + Tools
- Asset Library Mechanisms support processes in all reuse process families
  - Collecting, organizing, and characterizing reusable assets
  - Making asset information available to a user
  - Accumulating metrics and feedback about assets and the library
- Asset Library Mechanisms support various user roles

STARS Library Mechanisms/Hazle/VG3

As you have seen in the earlier ALOAF presentation, we define an asset library system or library mechanism to consist of a set of framework services plus tools. The framework services provide for basic operations on library data models and asset information. The tools aggregate and sequence services to provide higher level capabilities.

Asset library mechanisms support processes in all of the reuse process families identified in the Reuse CONOPS. They support the domain analysts and asset developers in capturing models and assets. They support the library managers in organizing and characterizing assets. They enable the software engineers who need the assets to search for and understand them. They also provide a means for accumulating information about the assets and the library operation that is fed back into the planning and management function.

Because it supports these multiple processes and roles of a reuse operation, the library mechanism can can be seen, together with the library data model, as the means to bind the processes together by facilitating the capture and dissemination of information about the domain, the assets, and the reuse operation.



STARS LIBRARY MECHANISMS: MULTIPLE APPROACHES

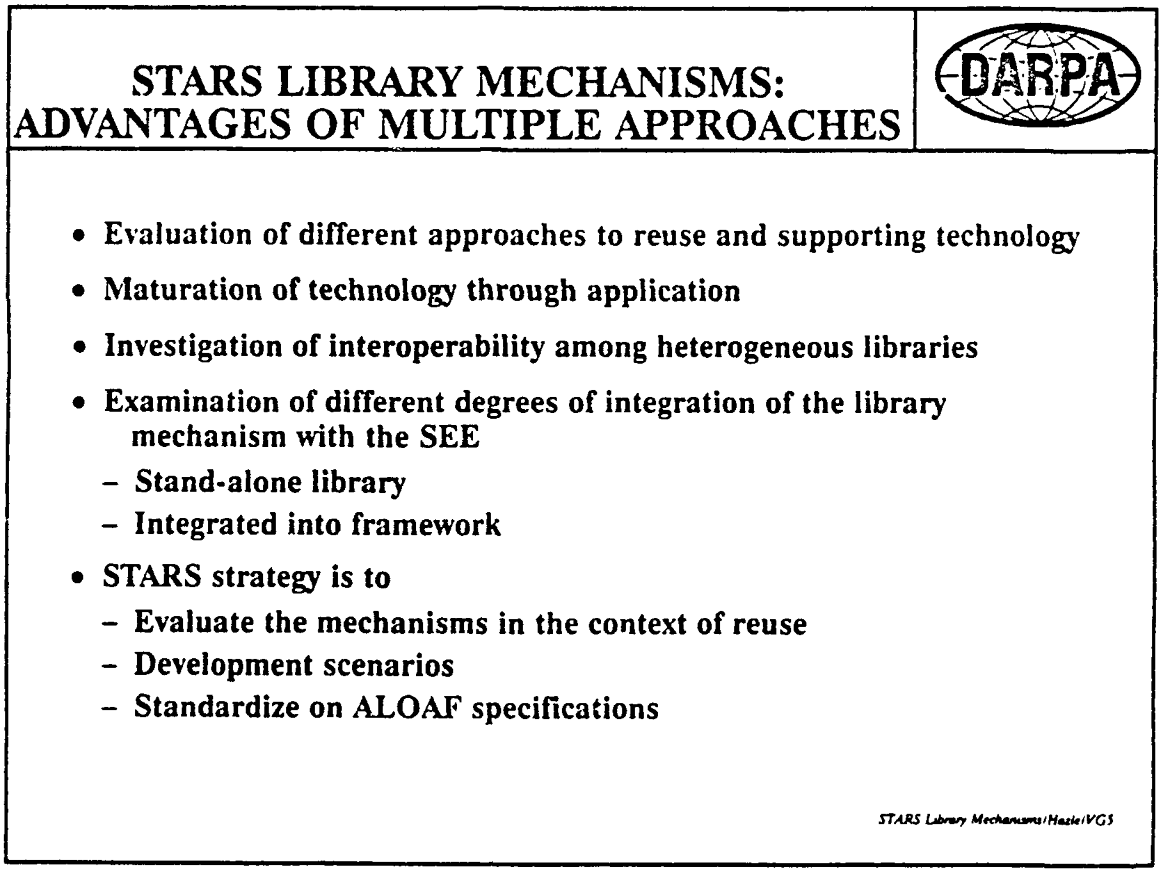
- Unisys: Reusability Library Framework (RLF)
  - Formally-Encoded Domain Models; knowledge based tools
  - Developed in Ada under the STARS foundations program
- IBM/SAIC/SPS: Asset Management System (AMS)
  - Multiple Classification Schemes; user-friendly definition
  - Derived from SPS's Automated Reusable Components System (ARCS)
- Boeing/DEC: Reusable Object Access and Management System (ROAMS)
  - Object-oriented/extensible repository
  - Uses commercial SEE framework to achieve tight integration
  - Derived in part from Boeing's initial repository experience

STARS Library Mechanisms/Hasle/VG4

The three STARS library mechanisms arc RLF, AMS, and ROAMS. The Unisys Reusability Library Framework (RLF) is motivated by the notion that formally-encoded domain models and associated tools are fundamental to domain specific reuse. It supports knowledge based techniques. RLF was initiated and has been evolved under the STARS program over the past four years.

The IBM team's library mechanism is the Asset Management System (AMS). AMS is guided by the notion that multiple classification techniques and search modes are needed to support various domains, and that classification scheme definition should be supported through a user-friendly interface. AMS is derived from the SPS ARCS, which was developed under an Army CECOM SBIR program.

The Boeing Reusable Object Access and Management System (ROAMS) provides an object oriented library capability. It achieves tight integration of the library with the SEE by being implemented as extensions of the DEC COHESION and Common Data Dictionary products. It is based in part on Boeing's earlier STARS repository prototype.



STARS LIBRARY MECHANISMS: ADVANTAGES OF MULTIPLE APPROACHES

DARPA

- Evaluation of different approaches to reuse and supporting technology
- Maturation of technology through application
- Investigation of interoperability among heterogeneous libraries
- Examination of different degrees of integration of the library mechanism with the SEE
  - Stand-alone library
  - Integrated into framework
- STARS strategy is to
  - Evaluate the mechanisms in the context of reuse
  - Development scenarios
  - Standardize on ALOAF specifications

STARS Library Mechanisms/Hazle/VGS

Because little reuse-based development has been done thus far, there is a lack of experience from which to draw requirements for reuse processes and their supporting tools. However, given the widespread conviction that effective reuse is key to improved software engineering, several reuse library systems have been developed, both inside and outside of STARS. These mechanisms vary from each other in many ways, reflecting different hypotheses about how to best support reuse-based software engineering.

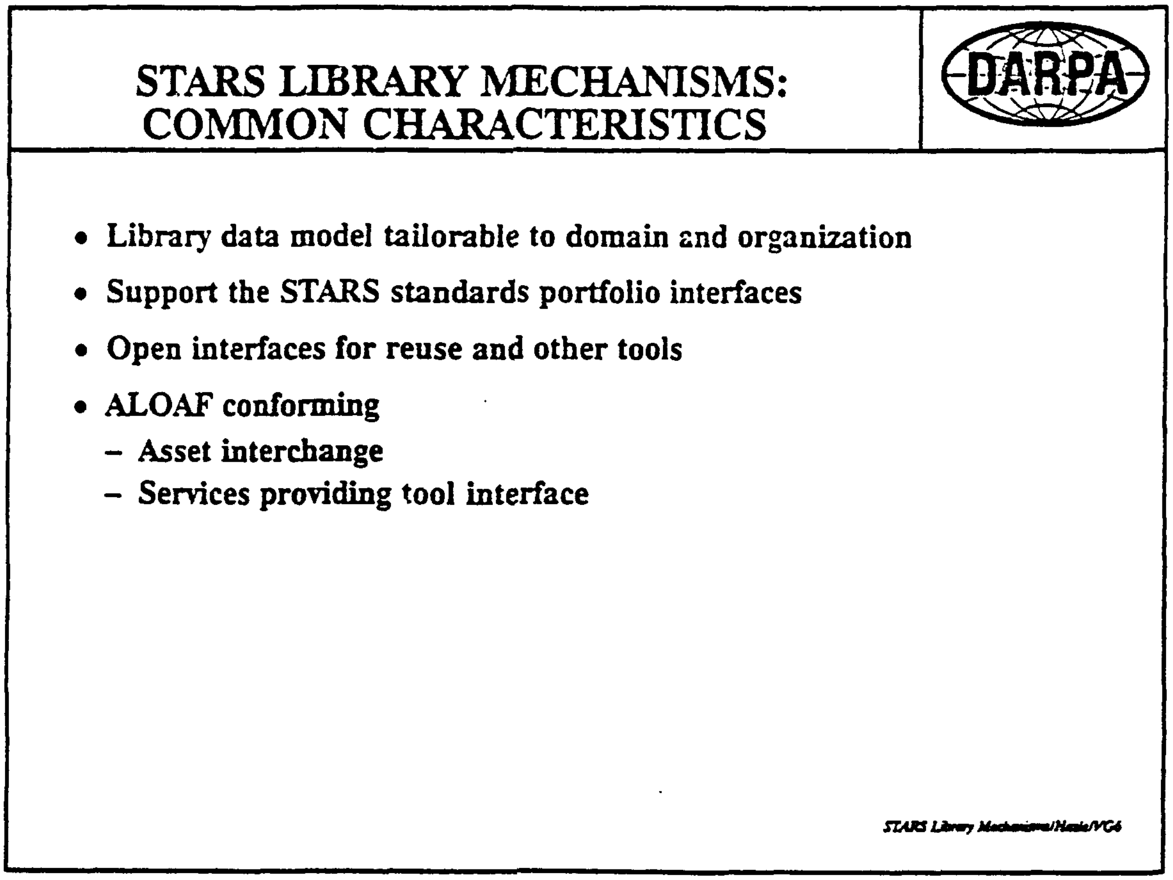
Although the STARS mechanisms have been used to capture domain information and assets, and to support the STARS primes in accessing and exchanging assets, they have not yet been practically used to support reuse-based development or maintenance of software systems. This is by and large true of other library mechanisms as well, with the notable exception of the CAMP library. There is also little experience with the use of libraries to support domain specific reuse or to manage many different types of assets.

We believe that it is advantageous to apply different mechanisms in realistic scenarios, so that we can understand how the characteristics of library mechanisms affect reuse. We want to see how different classification techniques affect asset retrieval. We want to understand what impact architecture orientation and the nature of the domain have on library mechanism requirements.

Our pursuit of multiple library mechanisms today enables us not only to evaluate different approaches in internal and affiliate applications, but in so doing to, to mature different technologies. The maturation will allow the mechanisms to be applied, and thus more thoroughly assessed for their effectiveness, in the STARS demonstration projects and beyond.

The existence of multiple library mechanisms within the STARS program enables us to prototype and experiment with the ALOAF interoperability provisions, concurrently with their development. We are also in a position to look at the effect of different degrees of library integration into a SEE.

The STARS strategy is to continue to pursue multiple approaches to providing library support, to evaluate the mechanisms in the context of reuse-based development and maintenance operations, and to implement the ALOAF standards in the library mechanisms. We will thus achieve interoperability among the STARS library systems and promote the sharing of reuse tools while we are gaining experience about the impact of library mechanism characteristics on the reuse processes.



STARS LIBRARY MECHANISMS: COMMON CHARACTERISTICS

DARPA

- Library data model tailorable to domain and organization
- Support the STARS standards portfolio interfaces
- Open interfaces for reuse and other tools
- ALOAF conforming
  - Asset interchange
  - Services providing tool interface

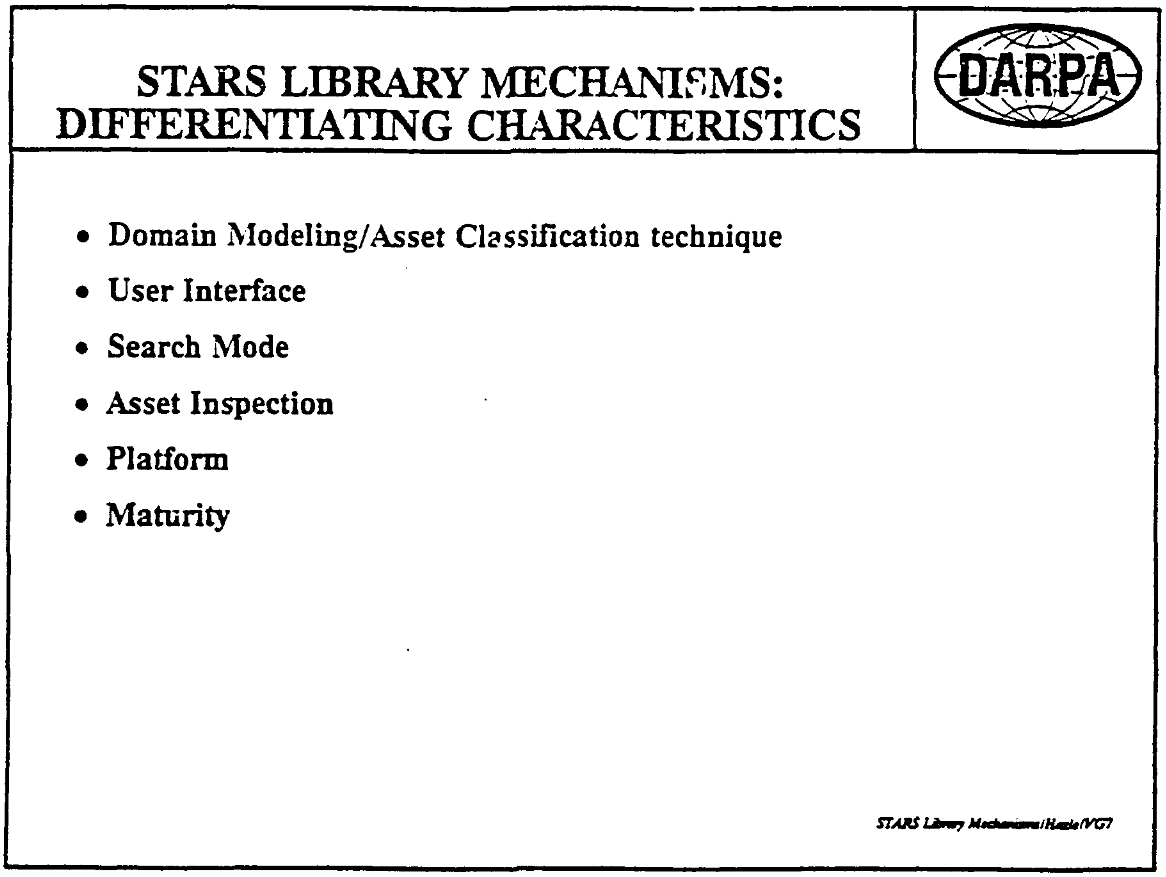
STARS Library Mechanisms/Hasle/VG6

Important common characteristics of the STARS library mechanisms are shown here. All STARS library mechanisms support user definition and modification of the library data model. This means that the data model can readily be tailored to a specific domain and/or to a specific organization to accommodate the kinds of assets and kinds of information needed.

Each of the library mechanisms will support/integrate the open interfaces supported by the STARS (and the prime) SEE, as appropriate. The STARS Standards Portfolio identifies those interfaces, which include, for example, X-windows.

Each library mechanism today has open interfaces to the library services so that reuse tools, and other tools, can utilize those services.

Each library mechanism will be ALOAF conforming and thus able to interoperate with the other STARS libraries—exchanging assets and enabling direct access to assets by the other primes' tools.



STARS LIBRARY MECHANISMS: DIFFERENTIATING CHARACTERISTICS

DARPA

- Domain Modeling/Asset Classification technique
- User Interface
- Search Mode
- Asset Inspection
- Platform
- Maturity

STARS Library Mechanisms/Hazle/VG7

Other characteristics vary across the STARS library systems and help to differentiate among them. The first four are characteristics whose impact we wish to understand better. The last two may influence the choice an affiliate would make in selecting a library mechanism for evaluation.

STARS LIBRARY MECHANISMS: RLF

- Domain Modeling/Asset classification technique
  - Structured inheritance network
    - Class/object hierarchy, arbitrary relationships/attributes, multiple inheritance
    - Accommodates variety of classification methods e.g., taxonomic, faceted, keyword
  - Rules to capture domain heuristics, provide user guidance
- User Interface
  - X-windows; graphical presentation of network hierarchy
- Search Mode
  - Browse through displayed network
  - Textual Query
  - Rule-based guided search

STARS Library Mechanisms/Hazle/VG8

The RLF provides the capability to define and store a class/object hierarchy with user determined relationships and attributes and with multiple inheritance. It can be used to implement a variety of classifications methods for assets. It also provides a knowledge-based capability in that rules may be associated with the nodes of the network and used to provide user guidance for traversing the network.

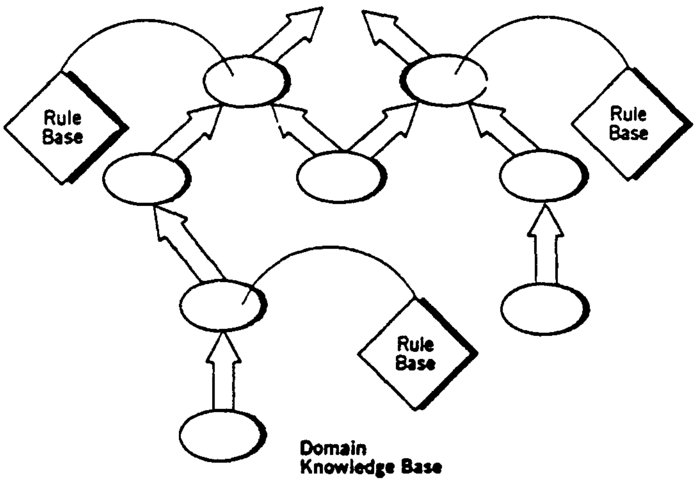
The user interface is organized primarily around a graphical presentation of the network hierarchy. At each network node RLF provides a context-sensitive set of commands to support browsing, inspection, and retrieval of library assets.

The search modes that are supported by RLF include the capability to browse through the network by pointing and clicking. A textual query capability and the assist mode of rule-based guidance also facilitate search. The asset inspection capability of the RLF enables the user to inspect textual information such as abstracts, source code, and documentation. It also has mechanisms to invoke external tools to provide other ways of understanding an asset such as looking at design diagrams or doing an on-line test of the asset. Other special functionality provided by the RLF is the ability to import and export assets in accordance with the current asset exchange specification that is a part of ALOAF.

STARS LIBRARY MECHANISMS: RLF

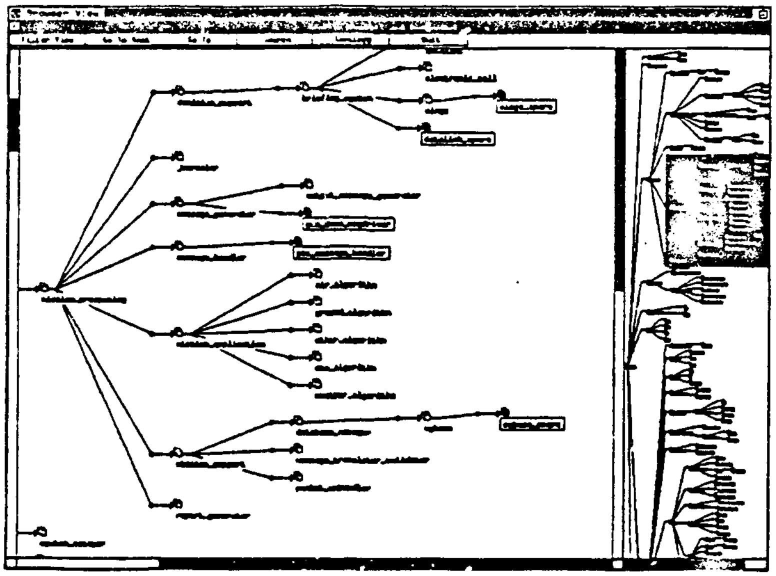
- Asset Inspection
  - Mechanisms to inspect textual asset information e.g., abstract, source, documentation
  - Mechanisms to invoke external tools to analyze other asset information e.g., to display design graphics, inspect formatted documents
- Other Functionality
  - Import/export of assets

STARS LIBRARY MECHANISMS: RLF APPROACH TO DOMAIN MODELING

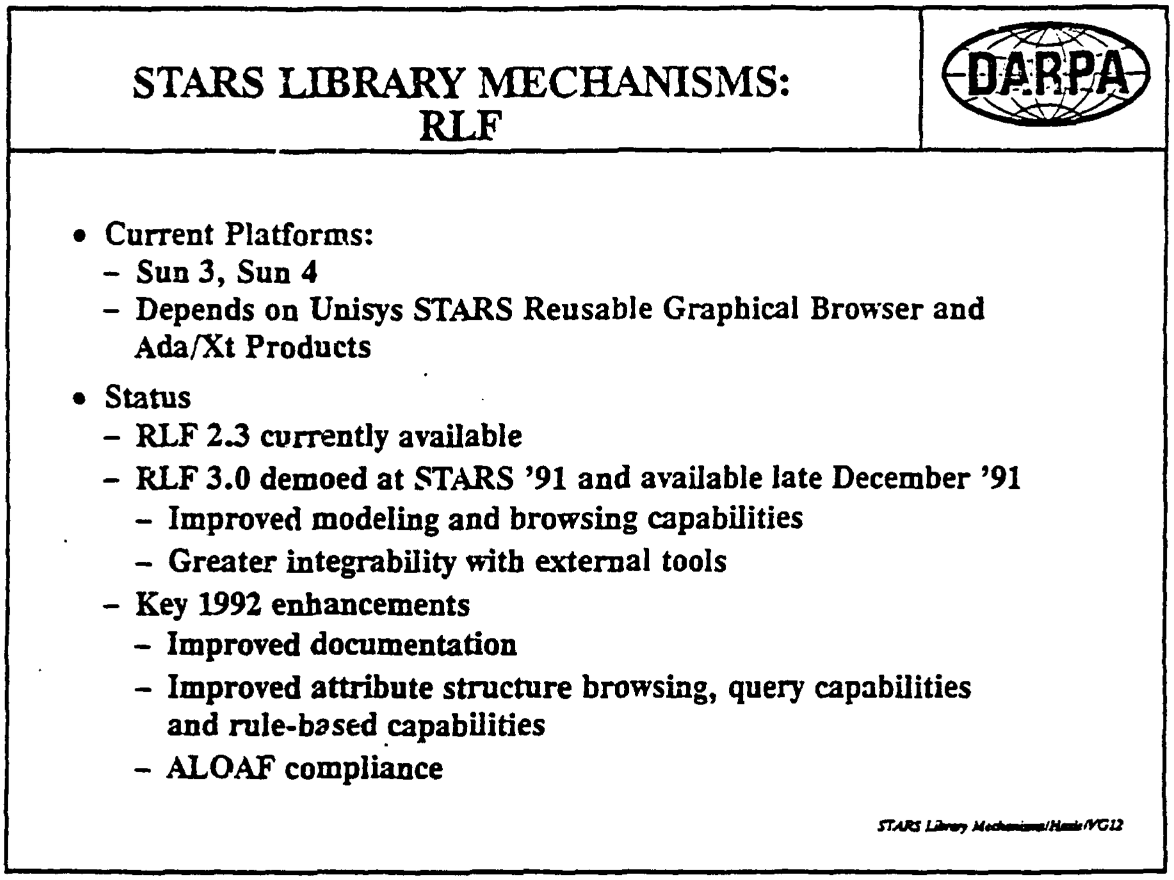


STARS Library Mechanisms/Hanks/VG10

STARS LIBRARY MECHANISMS: RLF USER INTERFACE



STARS Library Mechanisms/Hasle/VG11



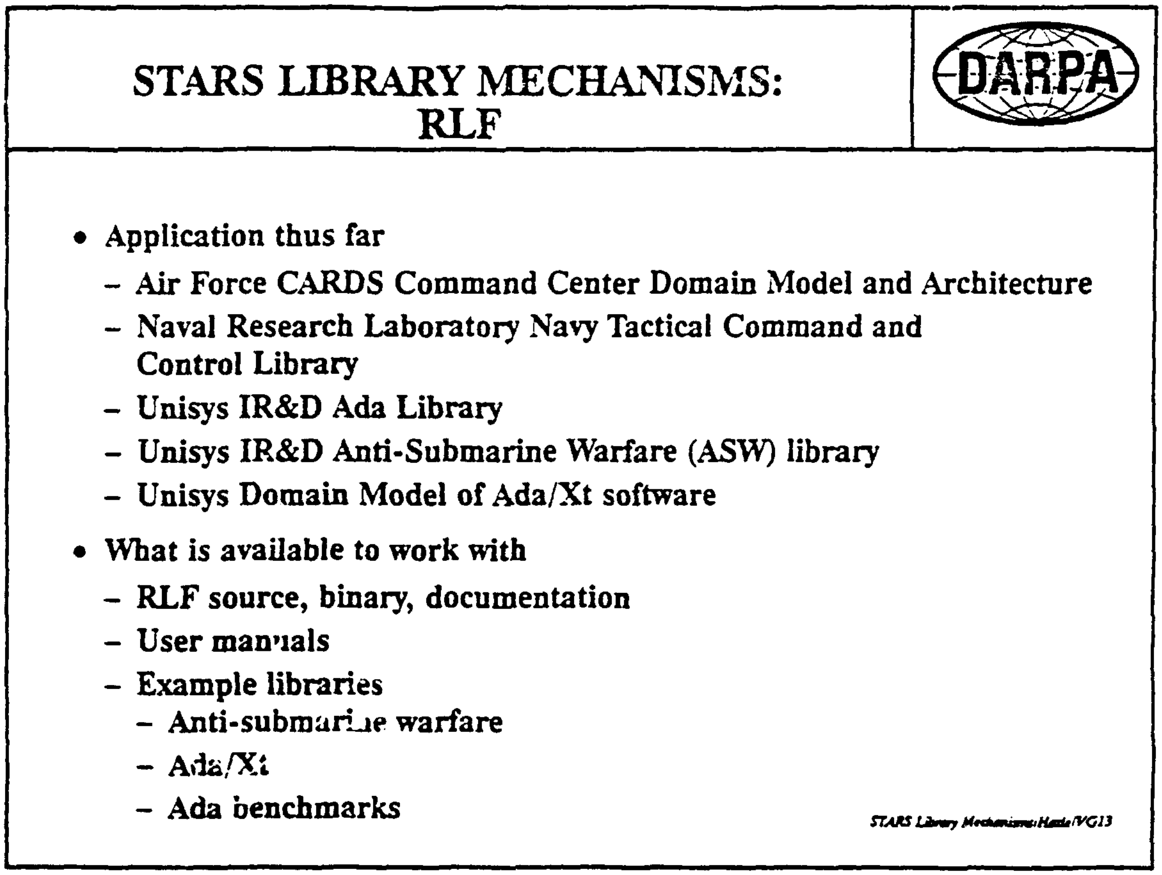
STARS LIBRARY MECHANISMS: RLF

- Current Platforms:
  - Sun 3, Sun 4
  - Depends on Unisys STARS Reusable Graphical Browser and Ada/Xt Products
- Status
  - RLF 2.3 currently available
  - RLF 3.0 demoed at STARS '91 and available late December '91
    - Improved modeling and browsing capabilities
    - Greater integrability with external tools
  - Key 1992 enhancements
    - Improved documentation
    - Improved attribute structure browsing, query capabilities and rule-based capabilities
    - ALOAF compliance

STARS Library Mechanisms/Hauk/VG12

The platforms on which RLF currently runs are Sun 3 and Sun 4 workstations with SunOS 4.1 (and later versions) and the Verdix 6.0.3 Ada compiler. The RLF depends on two other Unisys STARS products, the Reusable Graphical Browser and the Ada XT implementation.

Version 2.3 of RLF is currently available. The version that you see here at STARS 91 will be available by the end of this year. Over the next year several existing capabilities of the RLF will be enhanced and the RLF will be made compliant with the ALOAF asset interchange and service specifications.



STARS LIBRARY MECHANISMS: RLF

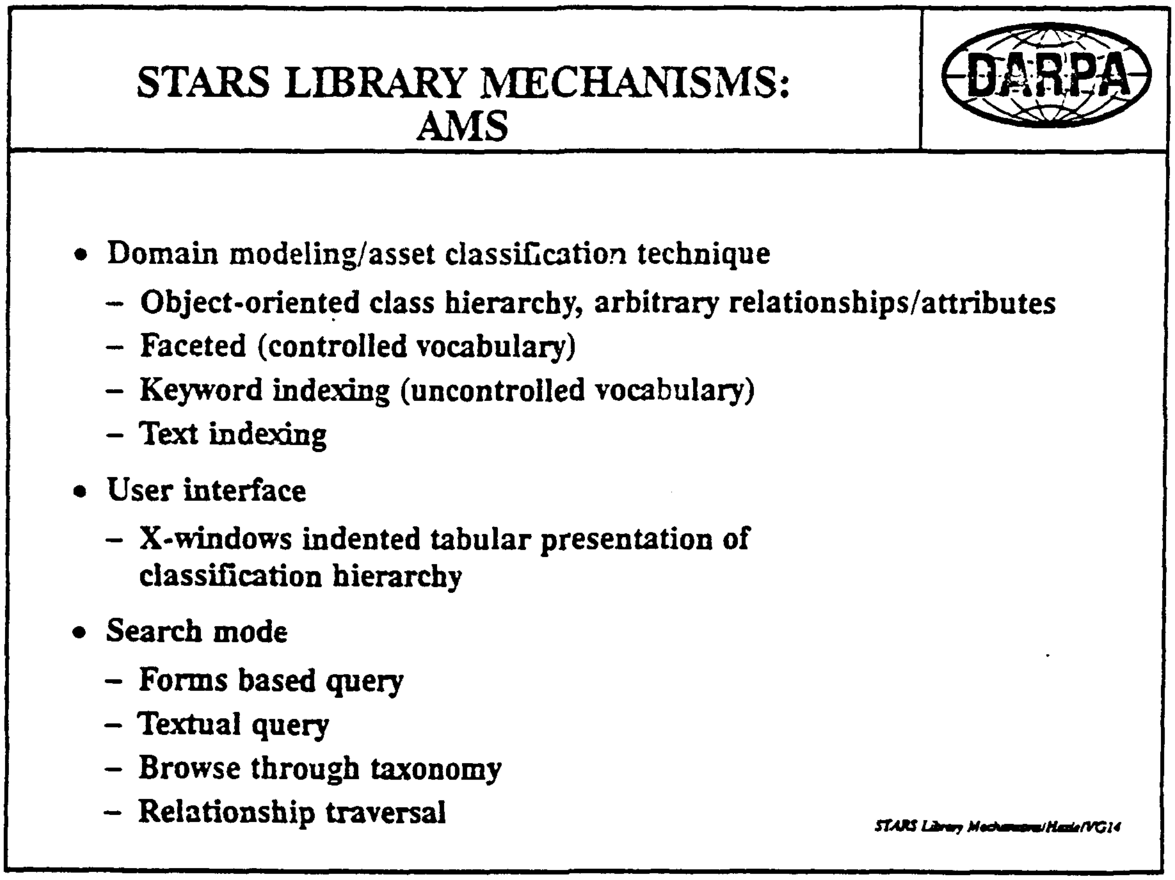
DARPA

- Application thus far
  - Air Force CARDS Command Center Domain Model and Architecture
  - Naval Research Laboratory Navy Tactical Command and Control Library
  - Unisys IR&D Ada Library
  - Unisys IR&D Anti-Submarine Warfare (ASW) library
  - Unisys Domain Model of Ada/Xt software
- What is available to work with
  - RLF source, binary, documentation
  - User man'ıals
  - Example libraries
    - Anti-submariɹe warfare
    - Ada/Xt
    - Ada benchmarks

STARS Library Mechanisms:Hasle/VG13

The RLF has been used by people other than its developers in several applications as shown in this slide. It is the library mechanism supporting the Air Force CARDS library, an operational library that you will be hearing more about in the next presentation.

The material available to potential users, in addition to the RLF software and associated documeɹtation, consists of user manuals and examples of the library data models that have been constructed using RLF. There are user manuals for the graphical browser, for the librarian, for data model construction and for rule base construction.



STARS LIBRARY MECHANISMS: AMS

DARPA

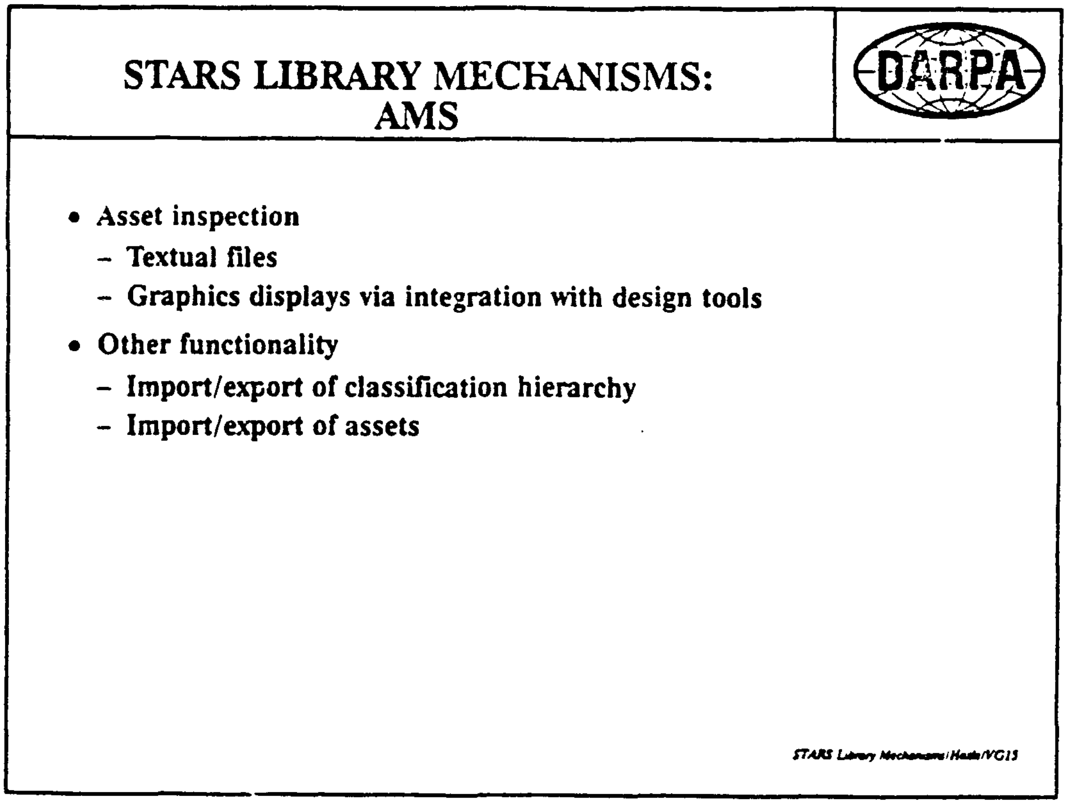
- Domain modeling/asset classification technique
  - Object-oriented class hierarchy, arbitrary relationships/attributes
  - Faceted (controlled vocabulary)
  - Keyword indexing (uncontrolled vocabulary)
  - Text indexing
- User interface
  - X-windows indented tabular presentation of classification hierarchy
- Search mode
  - Forms based query
  - Textual query
  - Browse through taxonomy
  - Relationship traversal

STARS Library Mechanisms/Hazle/VG14

The domain modeling and asset classification techniques offered by the AMS are based on an object oriented class hierarchy that allows for arbitrary relationships and attributes associated with the objects. The AMS provides explicit support for the faceted classification technique that was discussed in yesterday's briefing on the Domain Analysis Process Model. It also provides for indexing of keywords and of textual material associated with an asset.

The user interface presents classification information in tabular form. Search modes currently supported by AMS are form based queries and textual queries. AMS also supports browsing through the object collection based on user defined taxonomies and traversal of the object base using attribute relationships within the data model. The asset inspection capabilities allow for display of textual information and for the display of graphic information by integrated design tools.

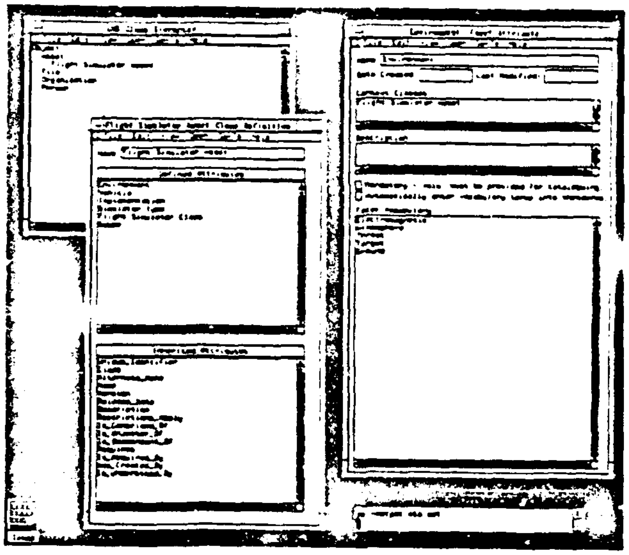
The AMS currently supports the import and export of the classification hierarchy, i.e., the data model, as well as the asset information.



STARS LIBRARY MECHANISMS: AMS

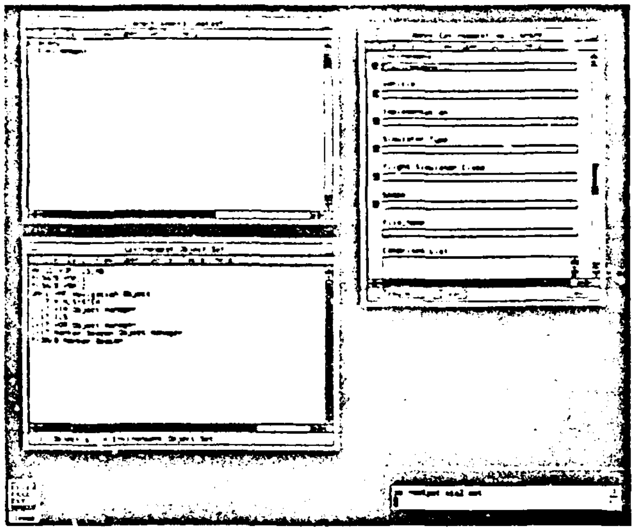
- Asset inspection
  - Textual files
  - Graphics displays via integration with design tools
- Other functionality
  - Import/export of classification hierarchy
  - Import/export of assets

STARS LIBRARY MECHANISMS: AMS CLASSIFICATION TECHNIQUE

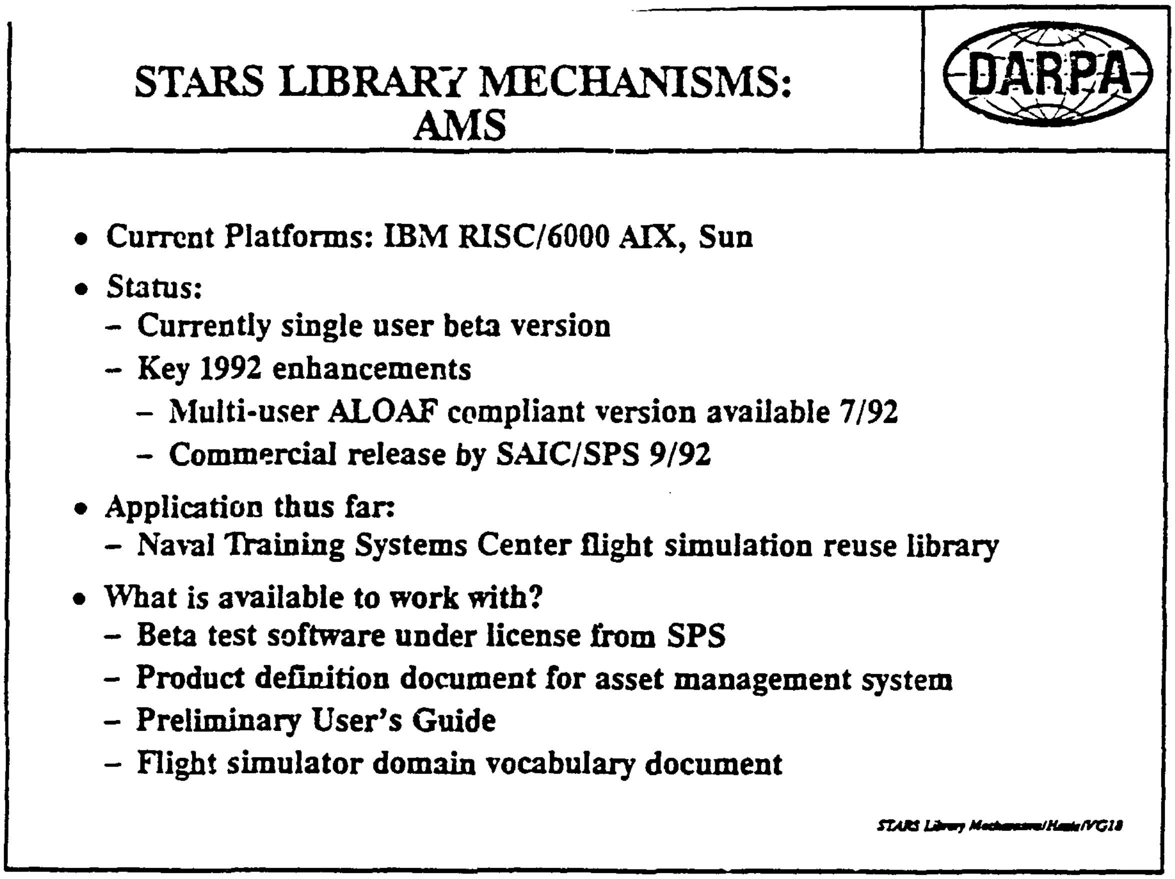


STARS Library Mechanisms Hayle VG16

STARS LIBRARY MECHANISMS: AMS USER INTERFACE



STARS Library Mechanisms·Hazle·VG17



STARS LIBRARY MECHANISMS: AMS

DARPA

- Current Platforms: IBM RISC/6000 AIX, Sun
- Status:
  - Currently single user beta version
  - Key 1992 enhancements
    - Multi-user ALOAF compliant version available 7/92
    - Commercial release by SAIC/SPS 9/92
- Application thus far:
  - Naval Training Systems Center flight simulation reuse library
- What is available to work with?
  - Beta test software under license from SPS
  - Product definition document for asset management system
  - Preliminary User's Guide
  - Flight simulator domain vocabulary document

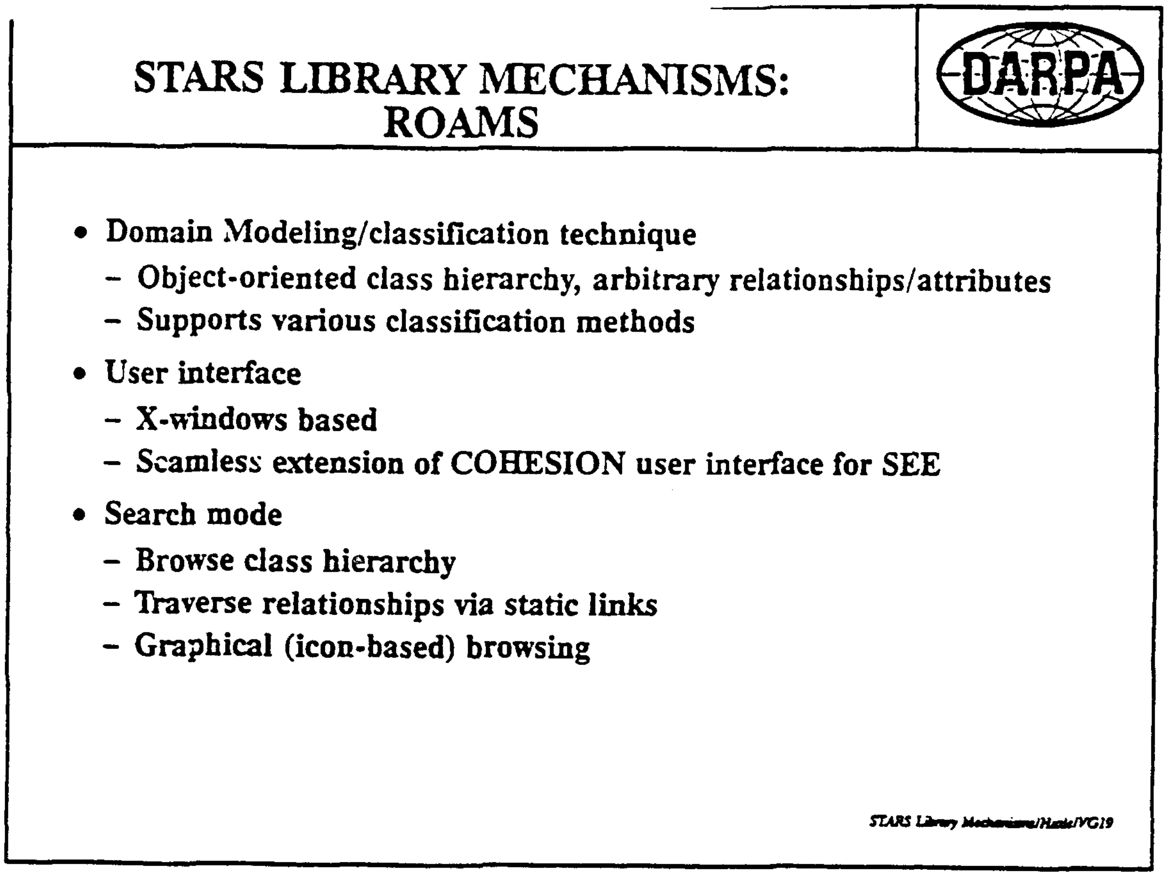
STARS Library Mechanisms/Hank/VG18

The AMS currently runs on the IBM RISC/6000 AIX workstation and on Sun 3 workstations using the Verdix compiler.

The version of AMS that you are seeing demonstrated is a single user Beta version. During 1992 a multi-user ALOAF compliant version will be developed, with a commercial release planned for next fall.

The AMS has been used on a Naval Training Systems Center reuse library for the flight simulation domain.

Material that is available to potential users of the AMS is the Beta test software under license from SPS, a product definition document that describes AMS user capabilities, and a preliminary users guide. An example of a domain specific library built using AMS is also available in both document and classification scheme import format.



STARS LIBRARY MECHANISMS: ROAMS

- Domain Modeling/classification technique
  - Object-oriented class hierarchy, arbitrary relationships/attributes
  - Supports various classification methods
- User interface
  - X-windows based
  - Scamless extension of COHESION user interface for SEE
- Search mode
  - Browse class hierarchy
  - Traverse relationships via static links
  - Graphical (icon-based) browsing

STARS Library Mechanisms/Hask/VG19

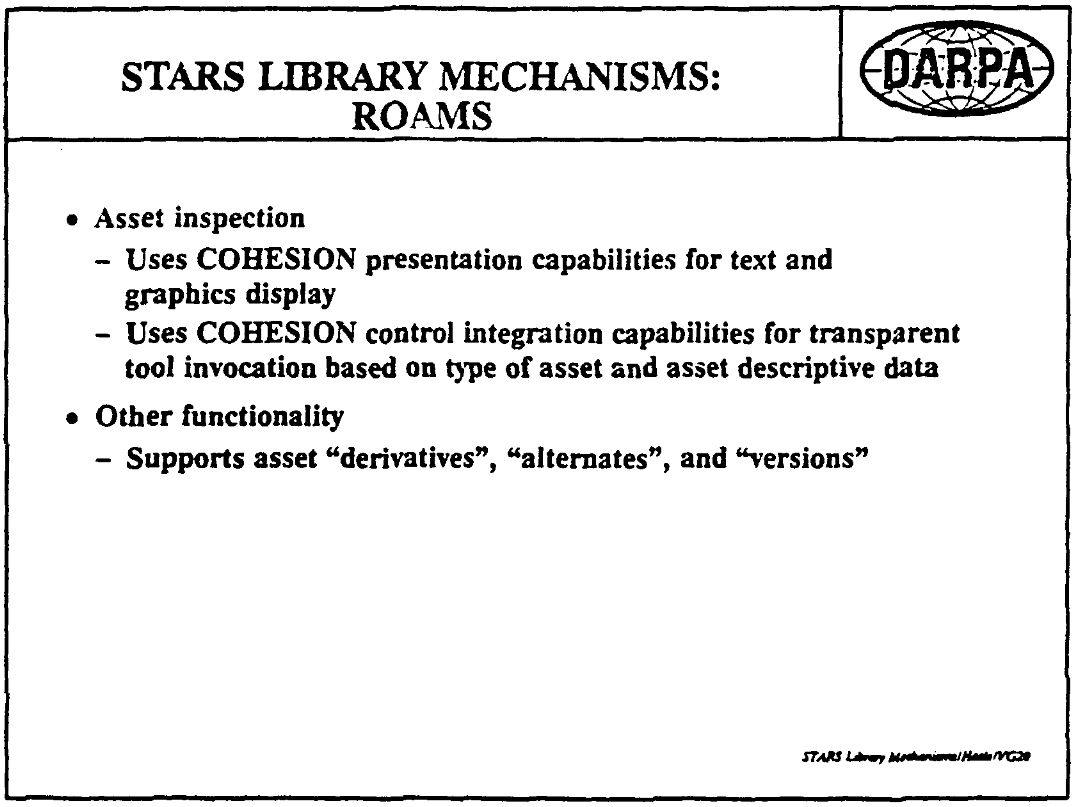
The domain modeling classification technique supported by ROAMS is also an object-oriented class hierarchy that allows for arbitrary relationships and attributes. It also supports various classification methods determined by the needs of the domain. ROAMS does come with a "starter" library data model for which some explicit special support is provided.

The user interface of ROAMS is, like the others, X-windows based. It is an extension of the COHESION user interface of the Boeing SEE.

The search modes supported by ROAMS allow a user to browse the class hierarchy, to traverse relationships among assets through links in the data model, and to browse through the library by clicking on icons associated with asset types.

Asset inspection within ROAMS is provided by the COHESION framework capabilities for presentation of textual and graphical material and also by the COHESION framework capability for invocation of tools that can provide other views or means of understanding assets.

Other functionality currently supported by ROAMS is that which facilitates keeping track of and manipulating derivatives, alternates and versions of assets within the library.

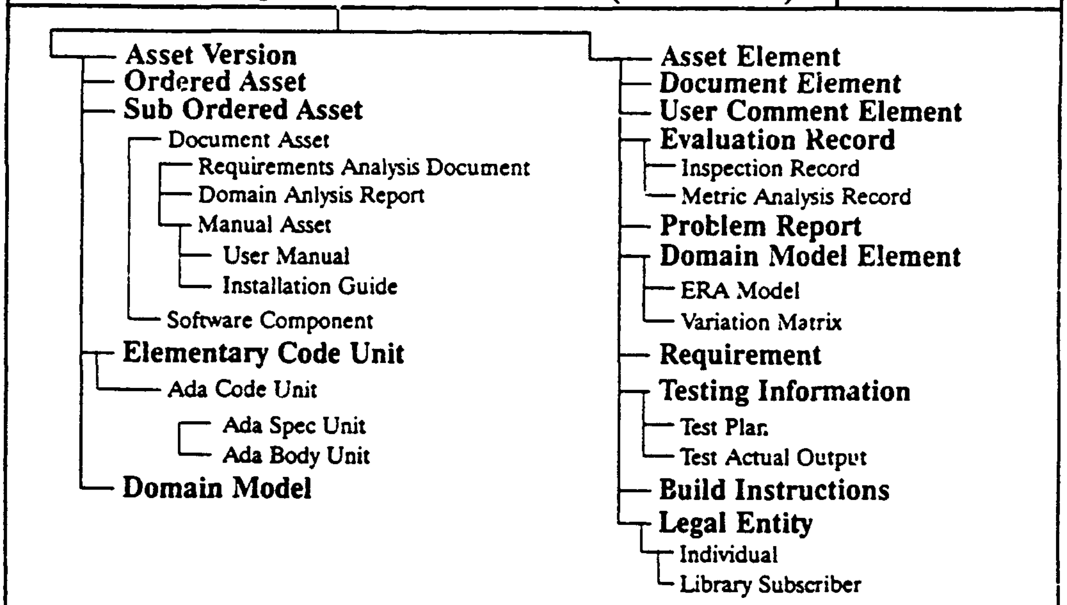


STARS LIBRARY MECHANISMS: ROAMS

- Asset inspection
  - Uses COHESION presentation capabilities for text and graphics display
  - Uses COHESION control integration capabilities for transparent tool invocation based on type of asset and asset descriptive data
- Other functionality
  - Supports asset “derivatives”, “alternates”, and “versions”

STARS Library Mechanisms/Heats/VG20

STARS LIBRARY MECHANISMS: ROAMS OBJECT HIERARCHY (PARTIAL)



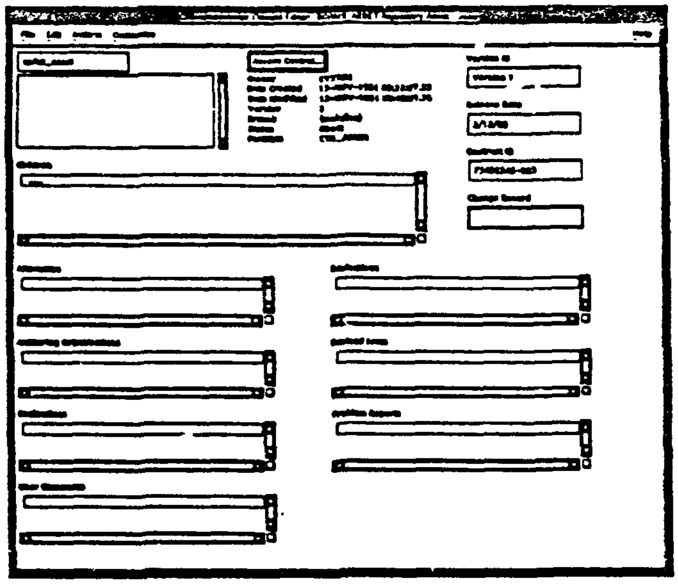
- Asset Version
- Ordered Asset
- Sub Ordered Asset
  - Document Asset
    - Requirements Analysis Document
    - Domain Anlysis Report
    - Manual Asset
      - User Manual
      - Installation Guide
  - Software Component
- Elementary Code Unit
  - Ada Code Unit
    - Ada Spec Unit
    - Ada Body Unit
- Domain Model

- Asset Element
- Document Element
- User Comment Element
- Evaluation Record
  - Inspection Record
  - Metric Analysis Record
- Problem Report
- Domain Model Element
  - ERA Model
  - Variation Matrix
- Requirement
- Testing Information
  - Test Plan
  - Test Actual Output
- Build Instructions
- Legal Entity
  - Individual
  - Library Subscriber

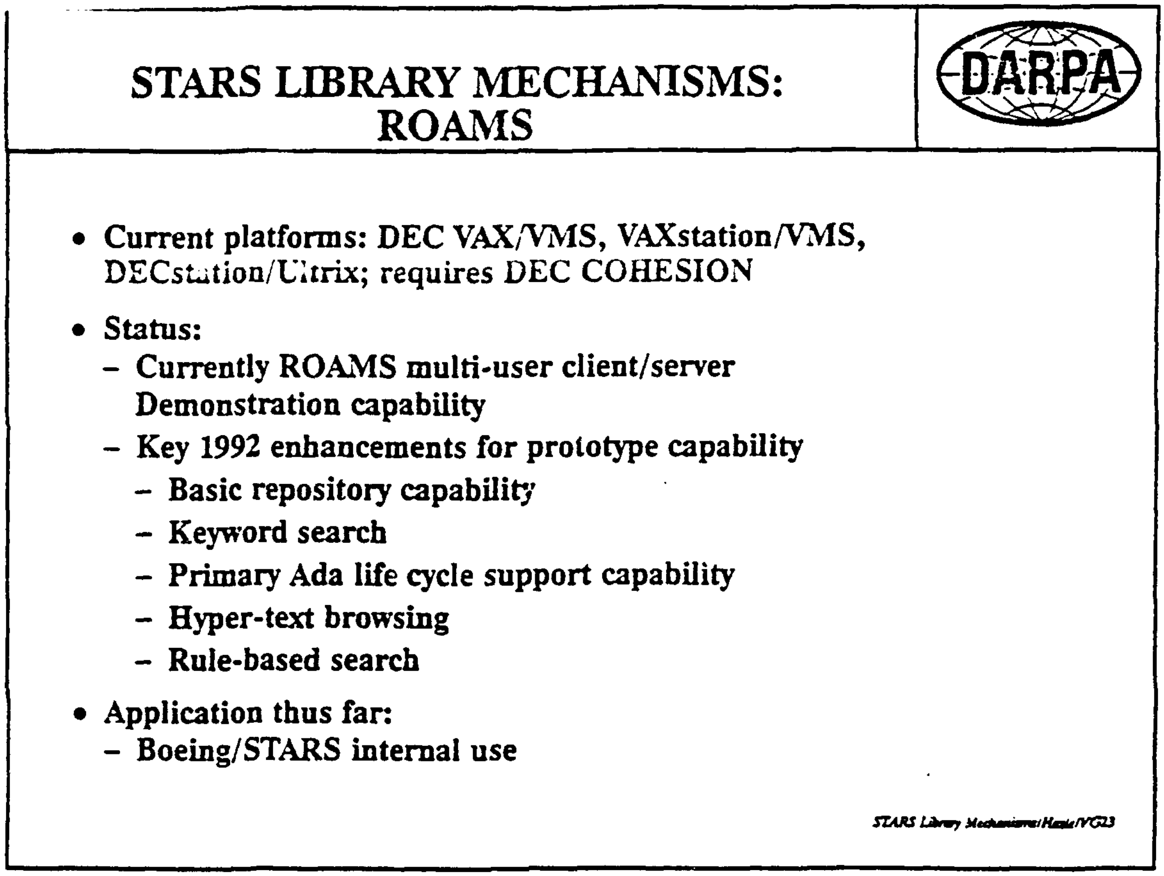
STARS Library Mechanisms/Haste/VG21

STARS LIBRARY MECHANISMS: ROAMS USER INTERFACE

DARPA



STARS Library Mechanisms/Haste/VG22



STARS LIBRARY MECHANISMS: ROAMS

DARPA

- Current platforms: DEC VAX/VMS, VAXstation/VMS, DECstation/Ultrix; requires DEC COHESION
- Status:
  - Currently ROAMS multi-user client/server Demonstration capability
  - Key 1992 enhancements for prototype capability
    - Basic repository capability
    - Keyword search
    - Primary Ada life cycle support capability
    - Hyper-text browsing
    - Rule-based search
- Application thus far:
  - Boeing/STARS internal use

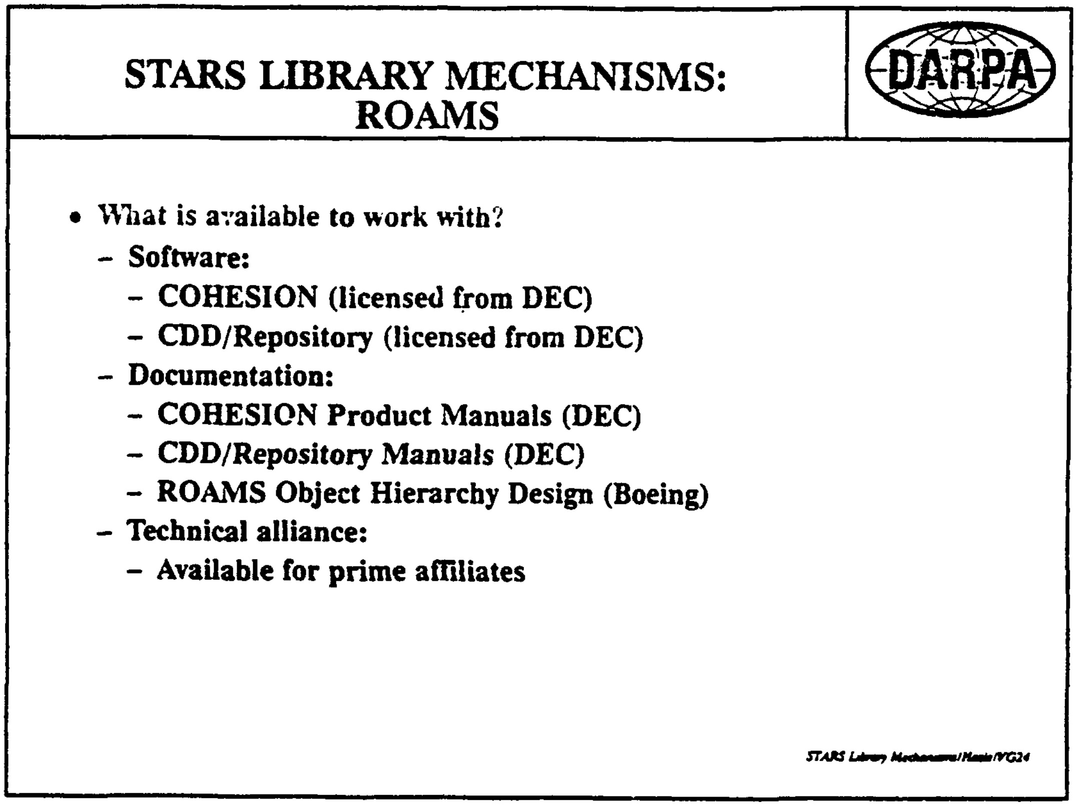
STARS Library Mechanisms/Hasle/VG23

ROAMS currently runs on the indicated Digital Equipment platforms and requires the COHESION framework.

The ROAMS that you are seeing here is a demonstration capability. In the next year ROAMS will be evolved into a working prototype with the additional capabilities indicated on the chart.

Thus far ROAMS has been used only internally by the Boeing STARS effort.

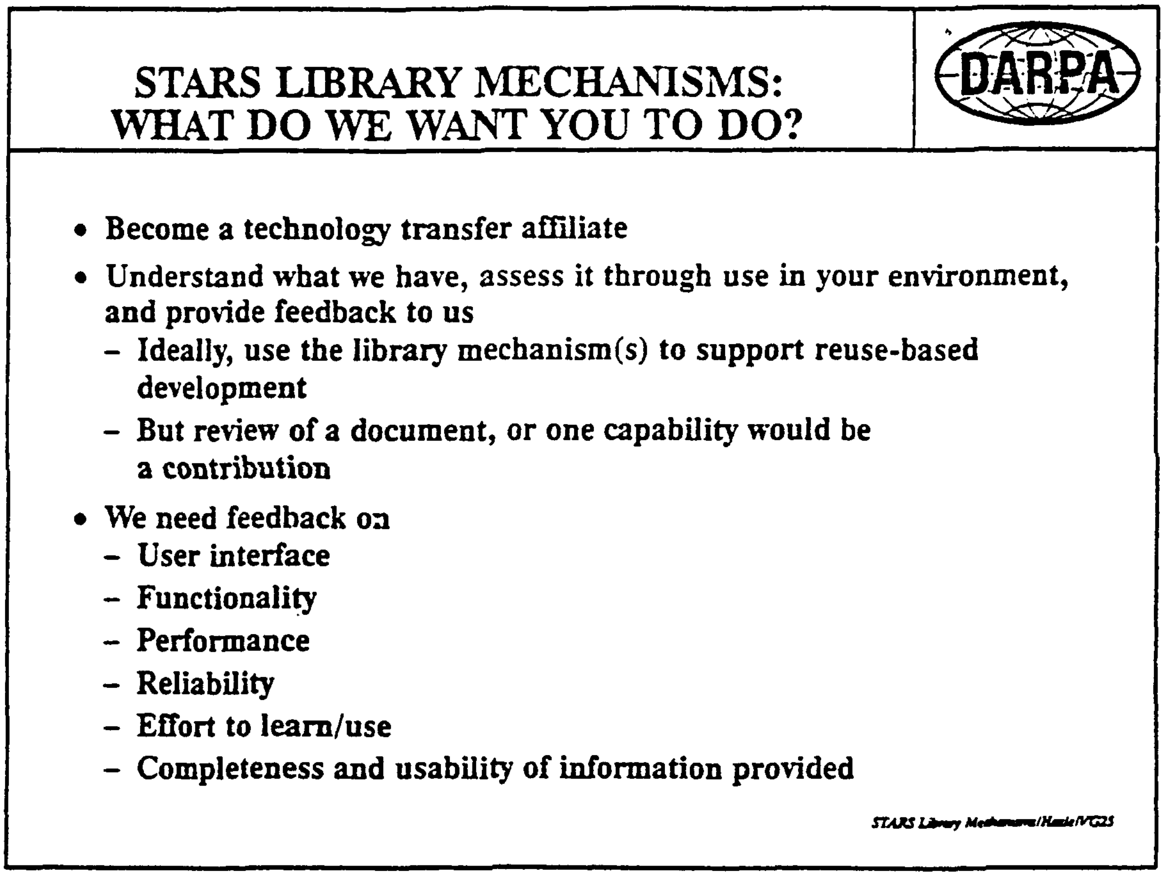
Material available to understand and work with ROAMS at the present is licensable software from DEC and associated manuals. That is augmented by the ROAMS object hierarchy design.



STARS LIBRARY MECHANISMS: ROAMS

- What is available to work with?
  - Software:
    - COHESION (licensed from DEC)
    - CDD/Repository (licensed from DEC)
  - Documentation:
    - COHESION Product Manuals (DEC)
    - CDD/Repository Manuals (DEC)
    - ROAMS Object Hierarchy Design (Boeing)
  - Technical alliance:
    - Available for prime affiliates

STARS Library Mechanisms/[illegible]/VG24



STARS LIBRARY MECHANISMS: WHAT DO WE WANT YOU TO DO?

DARPA

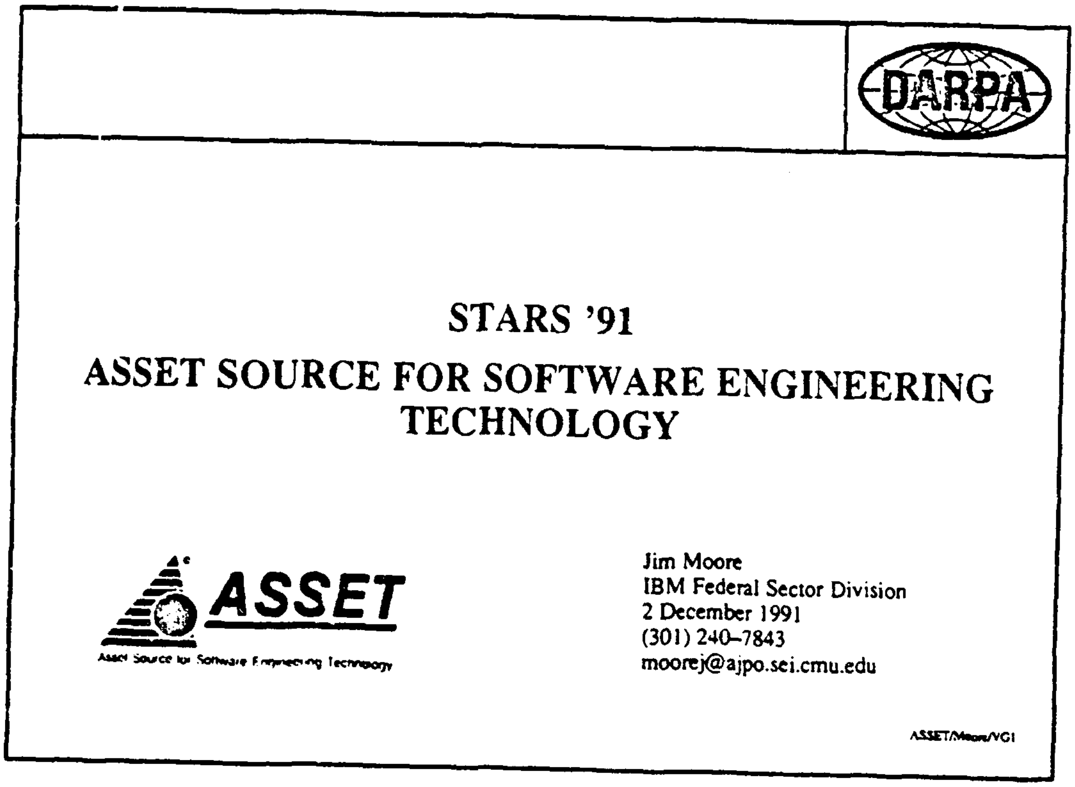
- Become a technology transfer affiliate
- Understand what we have, assess it through use in your environment, and provide feedback to us
  - Ideally, use the library mechanism(s) to support reuse-based development
  - But review of a document, or one capability would be a contribution
- We need feedback on
  - User interface
  - Functionality
  - Performance
  - Reliability
  - Effort to learn/use
  - Completeness and usability of information provided

STARS Library Mechanisms/Haslet/VG25

We would like you to become a technology transfer affiliate and join us in the effort to accelerate the shift to domain specific, reuse-based software development. The prime teams have applied the evolving library mechanisms internally and will continued to do. However we need application and evaluation from a spectrum of potential users to guide the evaluation of the STARS processes and mechanisms towards effective, production quality products.

We would like to have the library mechanisms applied and evaluated in the context of a reuse-based operation, encompassing many life cycle activities. But we welcome lesser contributions such as the review of documentation or the assessment of a single capability—for example the library data model construction capability, or asset browsing capability.

We seek your feedback about these listed characteristics of the current STARS library mechanisms. We also need your input about additional capabilities or features of the library systems that will facilitate reuse-based software development.



STARS '91

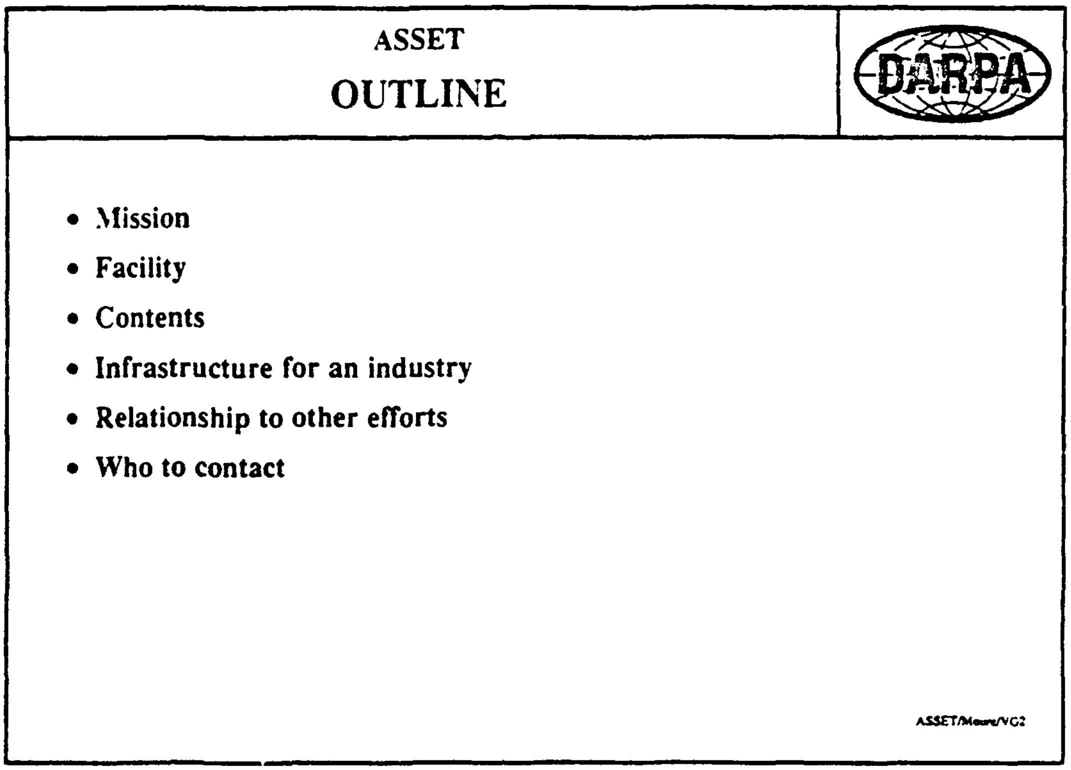
ASSET SOURCE FOR SOFTWARE ENGINEERING TECHNOLOGY

ASSET

Asset Source for Software Engineering Technology

Jim Moore
IBM Federal Sector Division
2 December 1991
(301) 240-7843
moorej@ajpo.sei.cmu.edu

ASSET/Moore/VG1



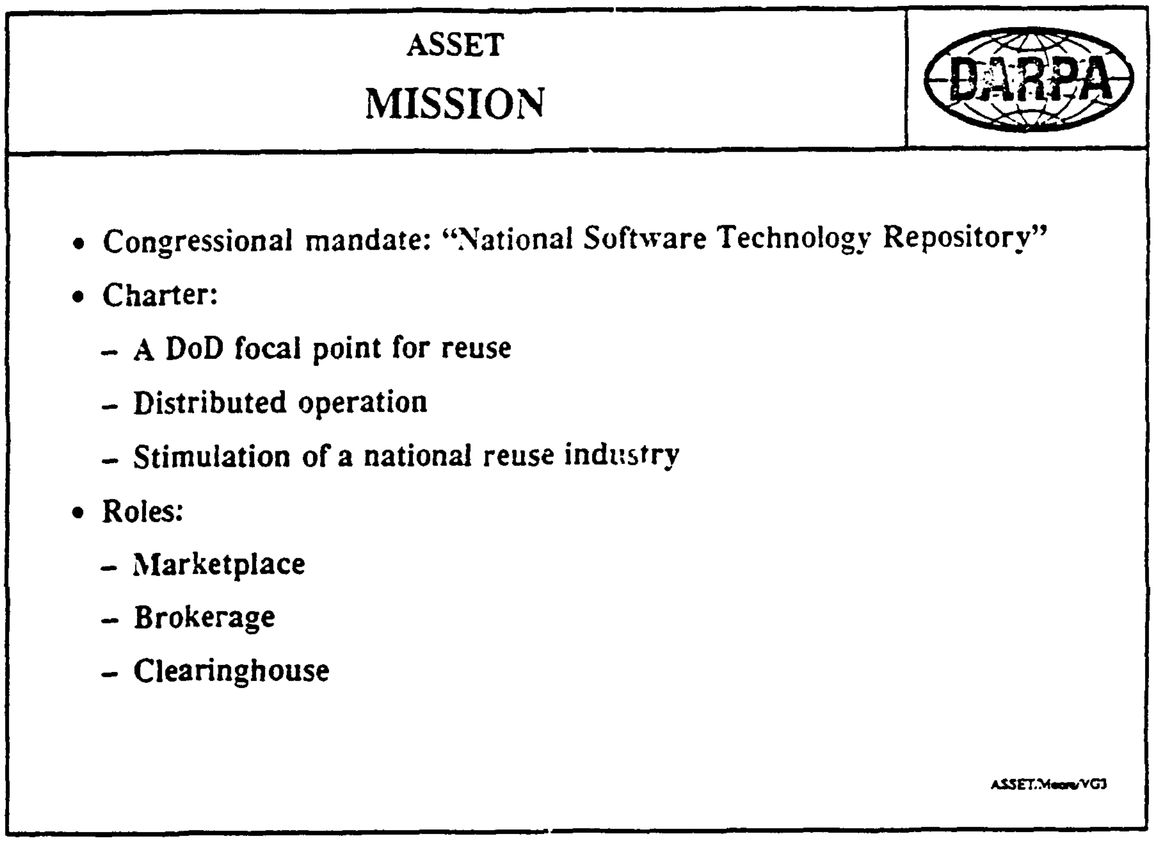
ASSET

OUTLINE

DARPA

- Mission
- Facility
- Contents
- Infrastructure for an industry
- Relationship to other efforts
- Who to contact

ASSET/Moore/VG2



ASSET

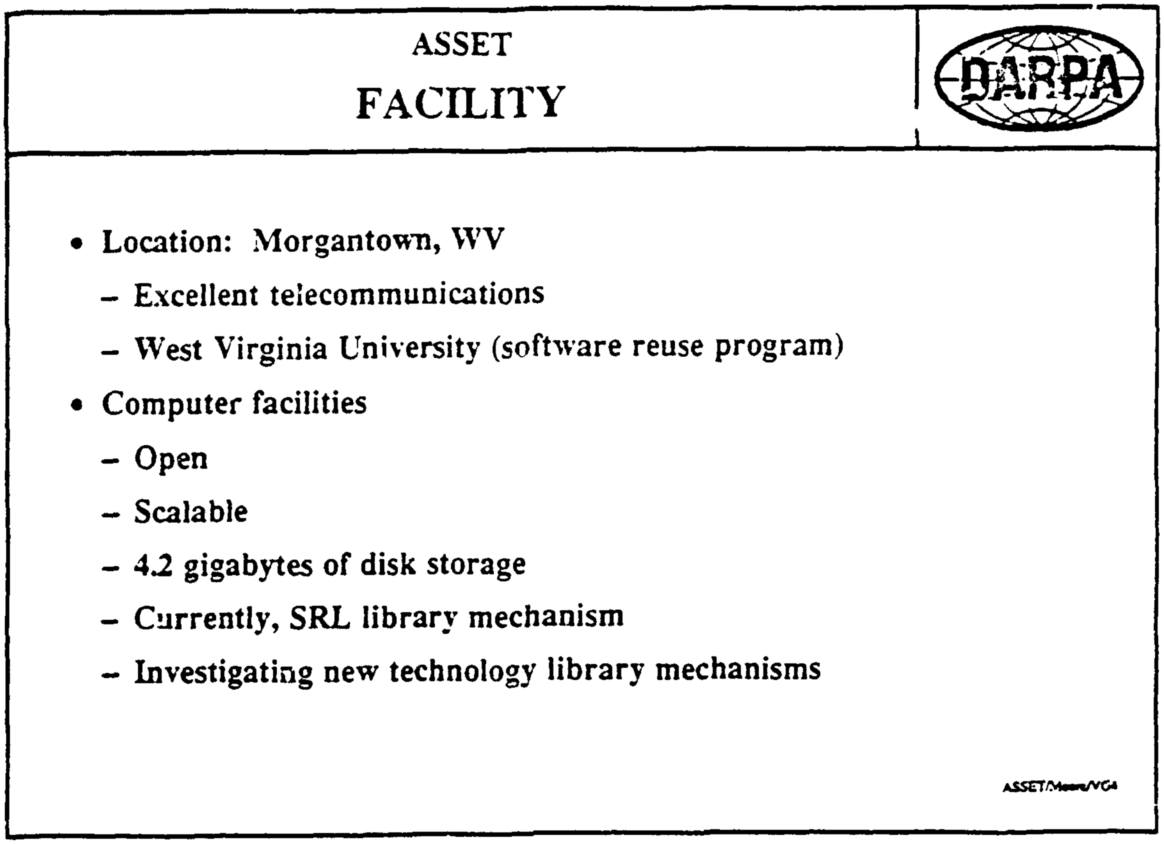
MISSION

- Congressional mandate: "National Software Technology Repository"
- Charter:
  - A DoD focal point for reuse
  - Distributed operation
  - Stimulation of a national reuse industry
- Roles:
  - Marketplace
  - Brokerage
  - Clearinghouse

ASSET.Moore/VG3

Congress mandated that the STARS program create a "National Software Technology Repository". DARFA responded by creating the ASSET project and awarding it in January 1991 to IBM and their major subcontractor SAIC. ASSET is intended to provide a focal point for software reuse within the Department of Defense. We plan to achieve this through a distributed network of reuse libraries. In the long term, we believe that this network and the other activities of STARS and ASSET will act as a catalyst to help stimulate the development of a national industry in reusable software components and ensure that the needs of the DoD are served by that industry.

We perceive three roles for ASSET, which we can express as metaphors. The first metaphor, "marketplace," means that ASSET will help create the electronic marketplace where commerce in software components can be conducted by both public and private consumers and producers. The second metaphor, "brokerage," means that ASSET will actively seek to match up producers and consumers in this marketplace. The third metaphor, "clearinghouse," means that ASSET will take a leadership role in stimulating the development of national standards which will enable electronic commerce in reusable components.



ASSET

FACILITY

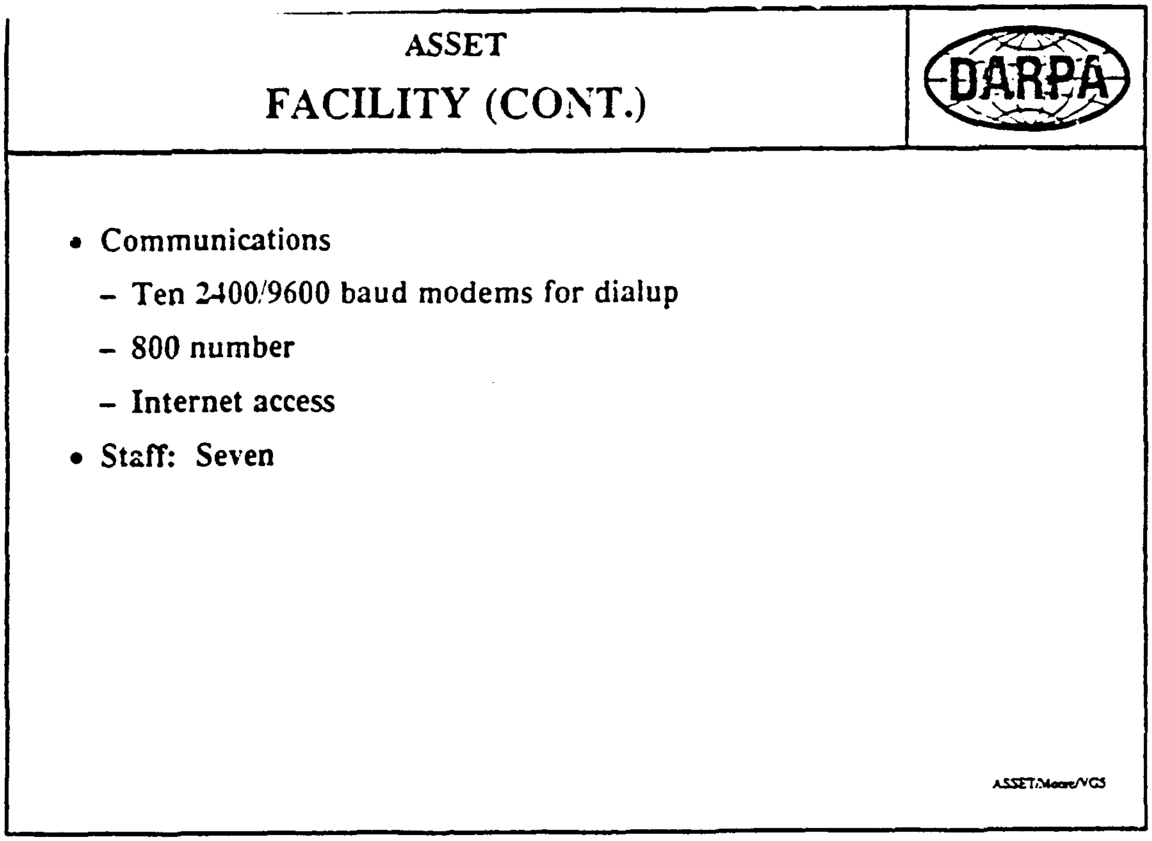
- Location: Morgantown, WV
  - Excellent telecommunications
  - West Virginia University (software reuse program)
- Computer facilities
  - Open
  - Scalable
  - 4.2 gigabytes of disk storage
  - Currently, SRL library mechanism
  - Investigating new technology library mechanisms

ASSET/[illegible]/VG4

The first instance of the ASSET repository is located in Morgantown, WV. Aside from other advantages like a generally low cost of living, West Virginia offers ASSET a specific advantage in its excellent state–of–the–art telecommunications infrastructure. Fiber–optic cabling and digital phone switching are common throughout the state. Morgantown, in particular, is attractive because of the ongoing software reuse program at West Virginia University and because of its closeness to the Software Engineering Institute in Pittsburgh.

Our initial computer configuration is built upon an open, scaleable architecture: "Open" because it is built upon industry standards like POSIX and X; "scalable" because it is configured around a token ring permitting the easy addition of processing or storage resources. Currently, we have installed 4.2 gigabytes of direct access storage.

Our initial library access mechanism is the STARS Reuse Library (SRL) tool prototype developed earlier in the program. We are currently investigating newer technology mechanisms and plan to select one shortly.



ASSET

FACILITY (CONT.)

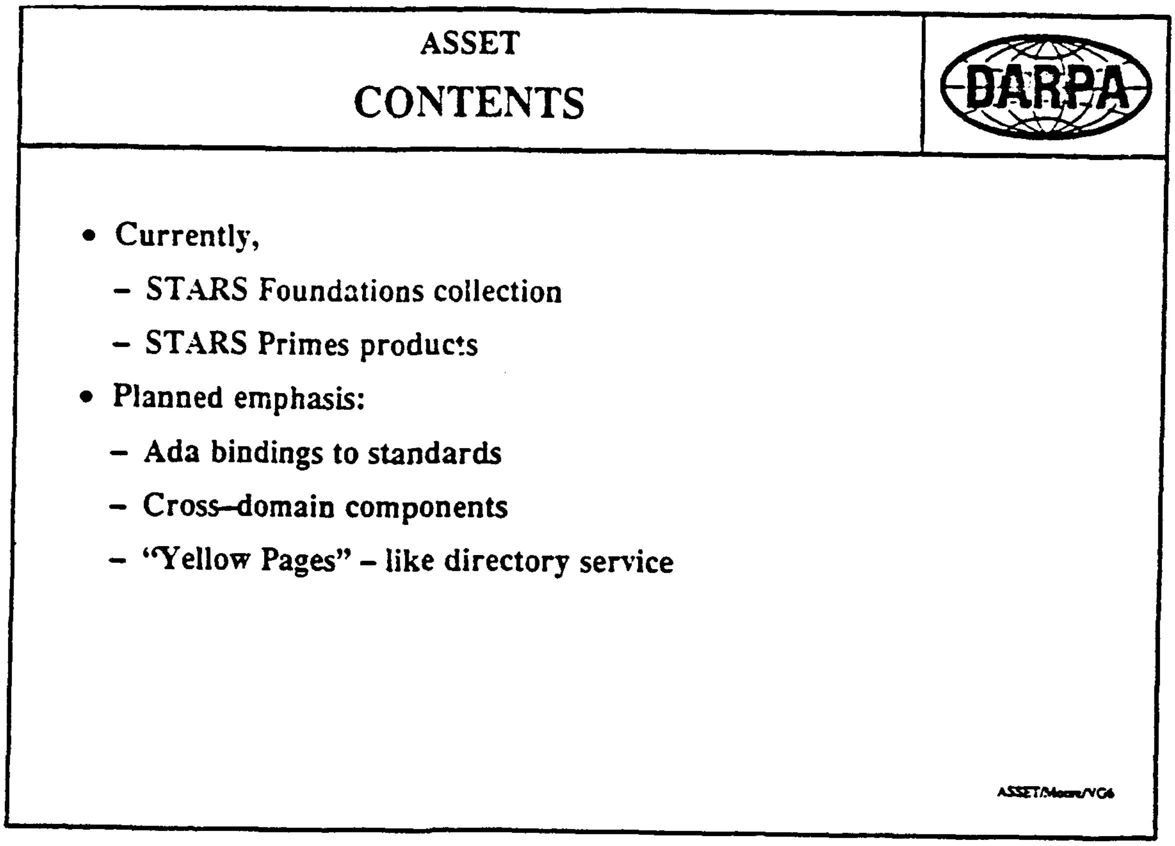
DARPA

- Communications
  - Ten 2400/9600 baud modems for dialup
  - 800 number
  - Internet access
- Staff: Seven

ASSET:Moore/VCS

We provide communications access to users via both dial-up and Internet. For dial-up, we have installed ten 2400/9600 bps modems which can be reached via an 800 number. Internet access is provided through a router connected to a regional network. In addition, we are conducting interoperability experiments via direct connection to other reuse libraries.

The current staffing at ASSET totals seven people on site plus additional support provided by other IBM and SAIC people.



ASSET

CONTENTS

- Currently,
  - STARS Foundations collection
  - STARS Primes products
- Planned emphasis:
  - Ada bindings to standards
  - Cross–domain components
  - "Yellow Pages" – like directory service

ASSET/Moore/VG6

ASSET has been operational for less than three months, so it is important to differentiate current capabilities from planned capabilities.

Currently, ASSET provides access to the STARS Foundations collection and to selected products of the STARS Primes contracts. ASSET will provide additional services to STARS program personnel and to Technology Transfer Affiliates.

In adding to our collection, we plan to emphasize Ada bindings to standards and cross–domain components. In our experience, Ada bindings are among the most requested reusable software components; we plan to provide explanatory information, public–domain bindings, and references to commercial products which provide standard Ada bindings.

We plan to provide both government–owned and proprietary cross–domain components, but do not plan to specialize in any particular application domain ourselves. We will leave domain specialization to other reuse libraries and will provide easy reference to those libraries by implementing a "yellow pages" directory which will assist users in finding appropriate libraries and vendors.

ASSET

INFRASTRUCTURE FOR AN INDUSTRY

- Distributed network of libraries
- Interoperation among libraries
- Diverse characteristics
  - Domain specialization
  - User interfaces
  - Fee structures
- Mixed public/private market:
  - Products
  - Search/retrieval methods
  - Value–added services

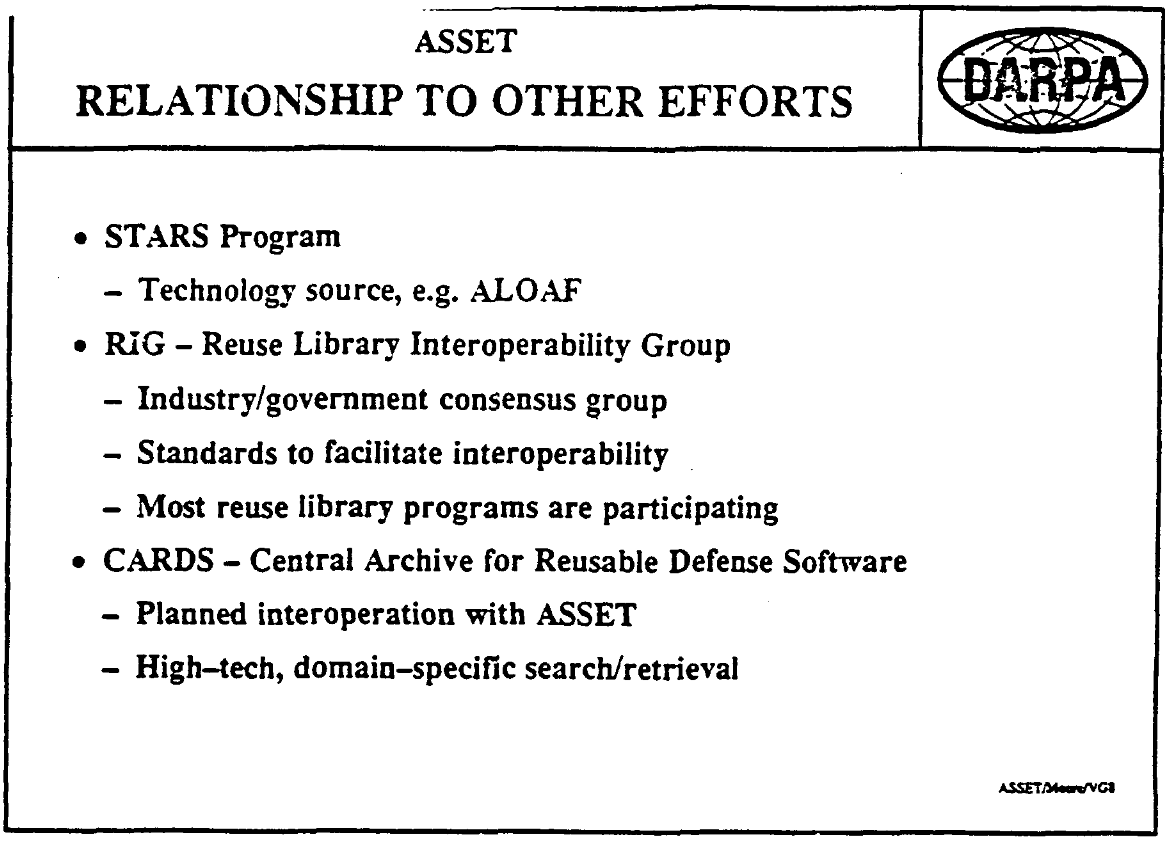
ASSET/Moore/VG7

I mentioned that ASSET's long–term goal is to catalyze the development of a national industry in software components. We think that industry is developing anyway and it's important to ensure that it will deal with the unique requirements of the Department of Defense. It is inevitable, for cultural reasons, that there will be many reuse libraries. So, the challenge is to ensure that these libraries will be able to interoperate. ASSET is working with other projects of the STARS program to obtain necessary technology, like ALOAF, and with the Reuse Library Interoperability Group (RIG) to formulate proposed standards for interoperation.

This variety of libraries will offer a diverse set of characteristics to users and we think that's a good thing. Different libraries will specialize in different application domains, will provide user interfaces suitable for different users, and will have fee structures suitable to different kinds of businesses.

We anticipate that the market will be a mixed public/private market. Some products for example will be public–domain; others will be government–owned; and others will be offered for sale by private entrepreneurs. We can anticipate that the search and retrieval mechanisms offered by individual libraries will be supplemented by value–added mechanisms provided for a fee by private entrepreneurs. We can also anticipate that private companies will offer other value–added services like consulting and systems integration.

All o these things will happen anyway. It is ASSET's job to plan, facilitate, and catalyze so that the needs of the Department of Defense are satisfied by this industry.



ASSET

RELATIONSHIP TO OTHER EFFORTS

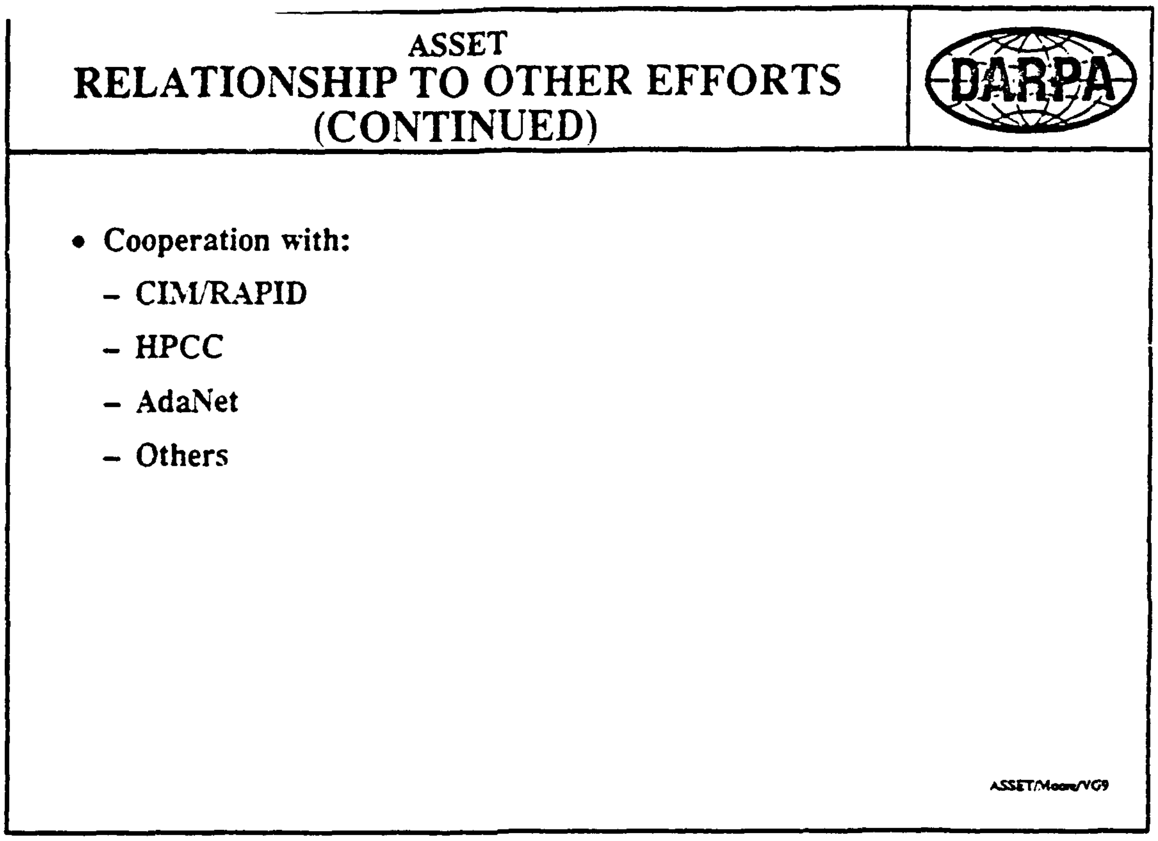
- STARS Program
  - Technology source, e.g. ALOAF
- RIG – Reuse Library Interoperability Group
  - Industry/government consensus group
  - Standards to facilitate interoperability
  - Most reuse library programs are participating
- CARDS – Central Archive for Reusable Defense Software
  - Planned interoperation with ASSET
  - High–tech, domain–specific search/retrieval

ASSET/Moore/VG8

Of course, ASSET is one task of the DARPA STARS program. Other projects within STARS are important sources of technology, e.g. ALOAF.

ASSET is a member of the Reuse Library Interoperability Group (RIG). The RIG is a voluntary government/industry consensus group which currently has twenty–three members including some major corporations, like IBM, some government agencies, like NIST, and most of the major reuse library programs. The RIG's mission is to investigate the problems of interoperability among reuse libraries and to propose standards which address those problems. These proposals will be forwarded to standards–making bodies, like ANSI, for their action.

ASSET is cooperating with the CARDS (Central Archive for Reusable Defense Software) program in performing interoperability experiments. CARDS is an example of a library will be apply high–technology search and retrieval techniques to a specific application domain.



ASSET

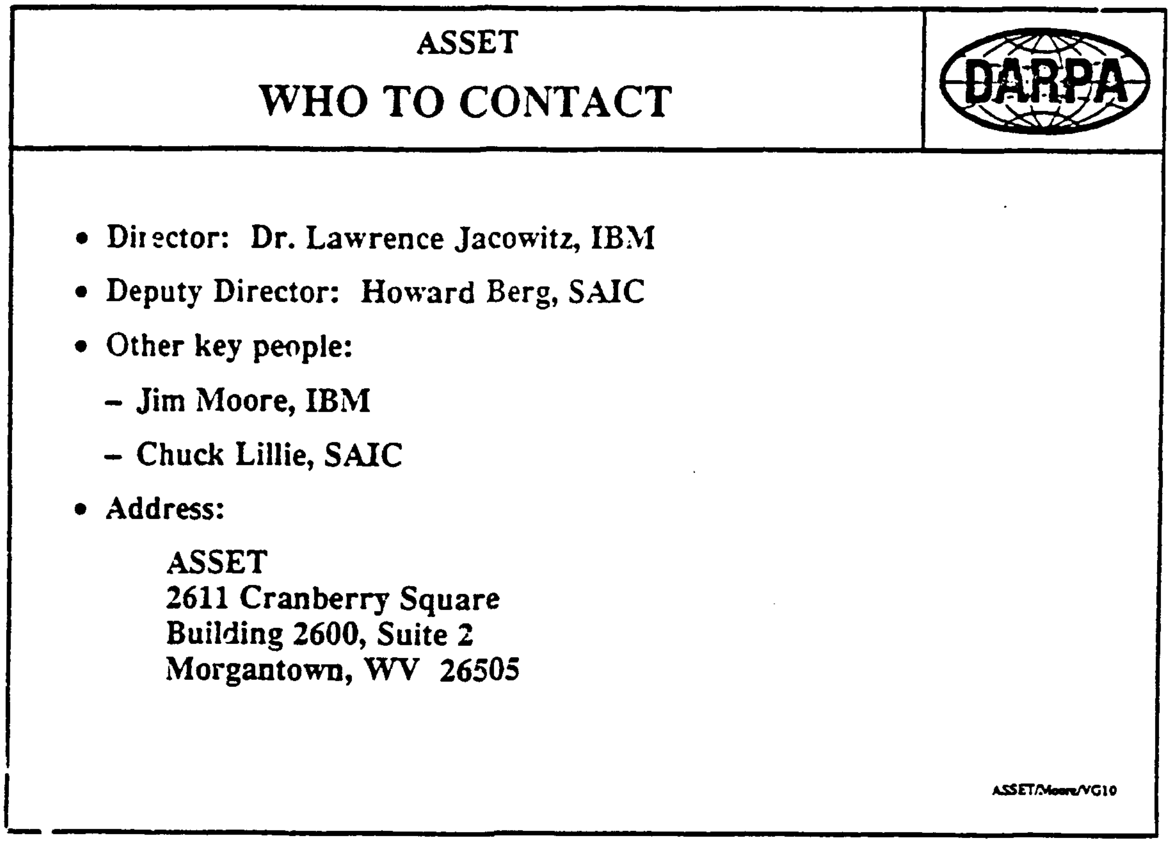
RELATIONSHIP TO OTHER EFFORTS (CONTINUED)

DARPA

- Cooperation with:
  - CIM/RAPID
  - HPCC
  - AdaNet
  - Others

ASSET/Moore/VG9

In addition, ASSET is cooperating with other efforts like DISA's CIM/RAPID library, the inter-agency High Performance Computing and Communications initiative, NASA's AdaNet library, and other programs.



ASSET

WHO TO CONTACT

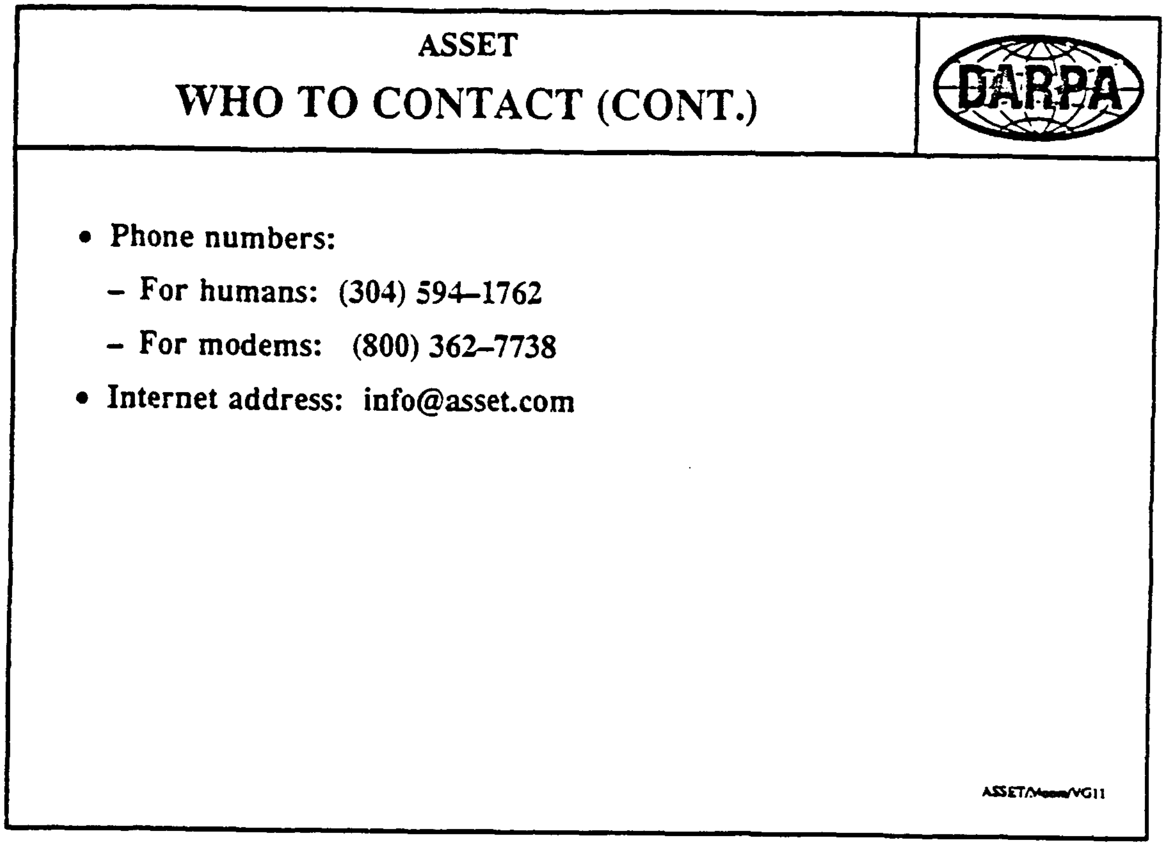
- Director: Dr. Lawrence Jacowitz, IBM
- Deputy Director: Howard Berg, SAIC
- Other key people:
  - Jim Moore, IBM
  - Chuck Lillie, SAIC
- Address:

ASSET
2611 Cranberry Square
Building 2600, Suite 2
Morgantown, WV 26505

ASSET/Moore/VG10

IBM is the prime contractor responsible for the ASSET project and SAIC is the major subcontractor responsible for the operation of the library. The key people in Morgantown are Dr. Lawrence Jacowitz of IBM, the Director of the ASSET library, and Howard Berg of SAIC, the Deputy Director. Key people in the Washington, DC area are Jim Moore of IBM, (301) 240-7343, and Dr. Charles Lillie of SAIC, (703) 749-8732.

The address of the ASSET library is given on the slide.



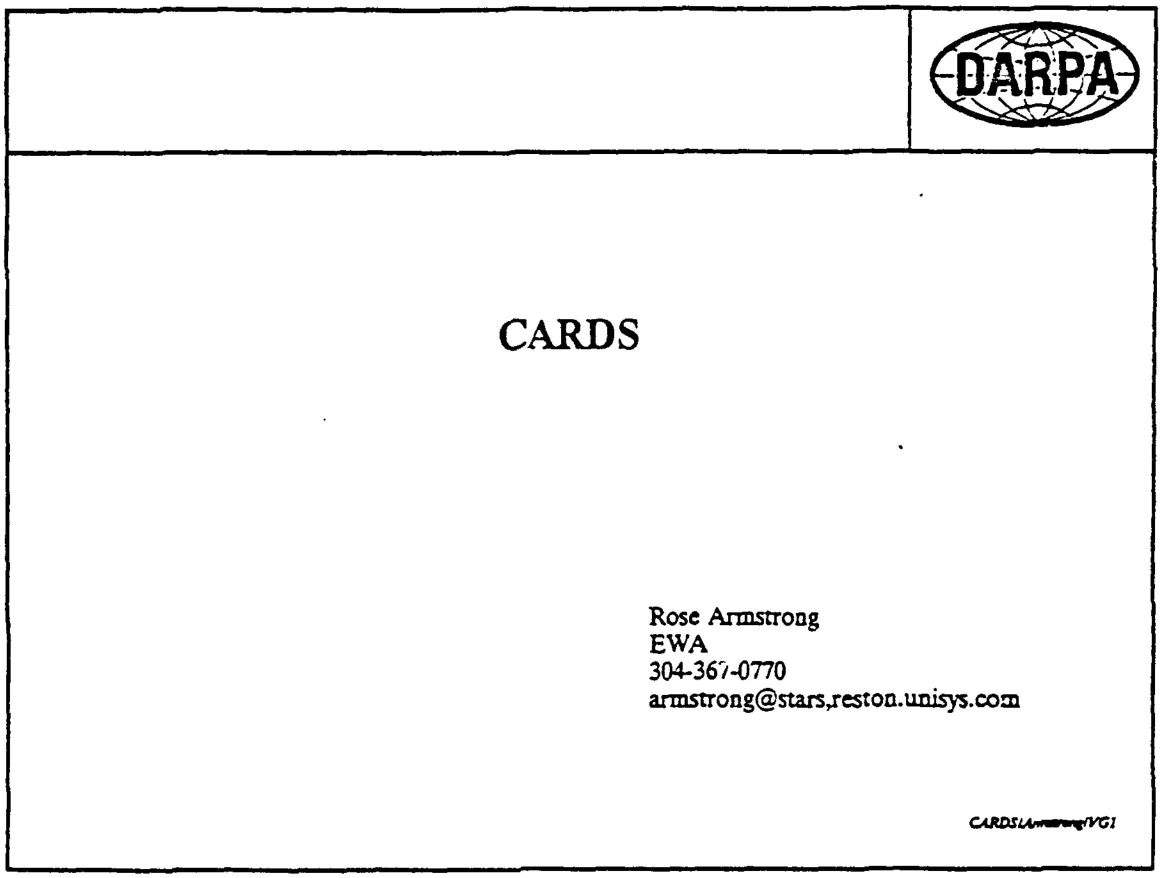
ASSET

WHO TO CONTACT (CONT.)

- Phone numbers:
  - For humans: (304) 594–1762
  - For modems: (800) 362–7738
- Internet address: info@asset.com

ASSET/Moore/VG11

One can contact the people at ASSET by calling the number on the slide. Once an account is established, one can access the services of ASSET by phoning the 800 number given on the slide. ASSET personnel can also be contacted via Internet at the given address.



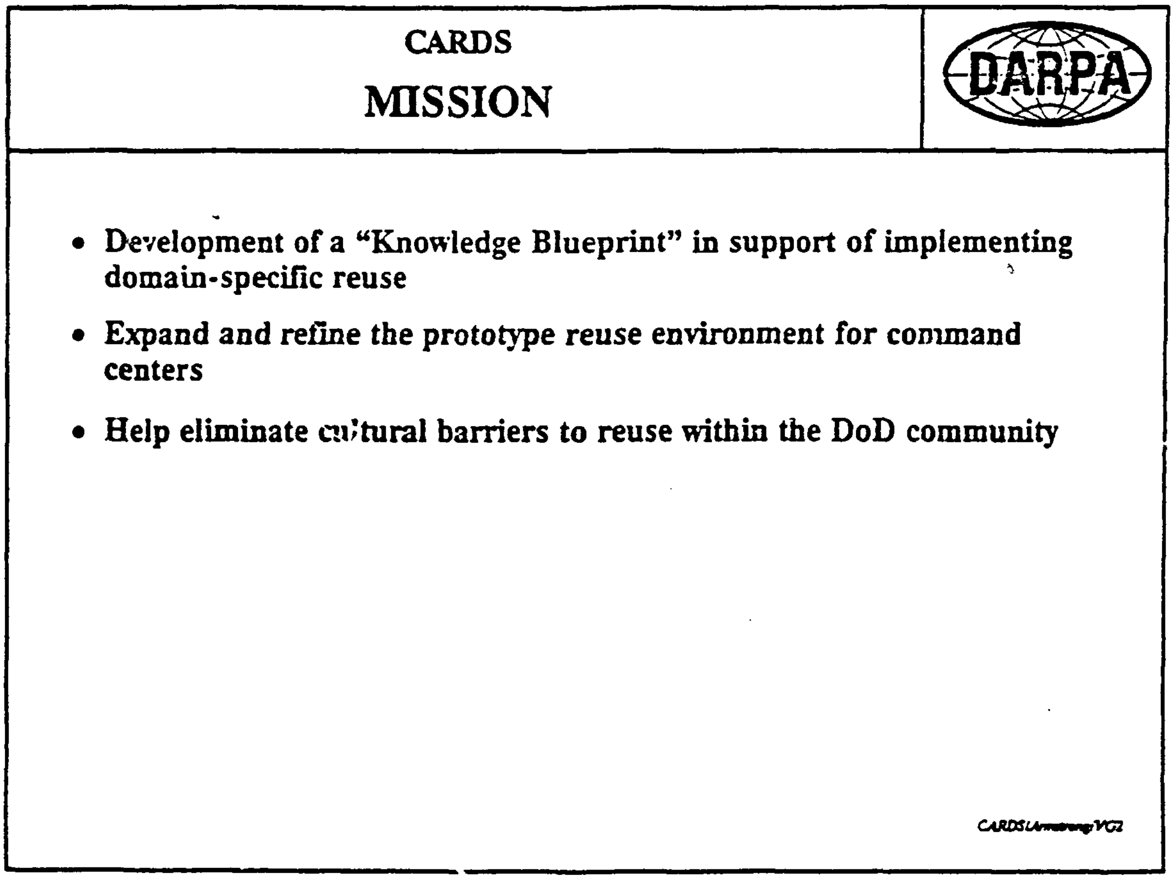
DARPA

CARDS

Rose Armstrong
EWA
304-367-0770
armstrong@stars,reston.unisys.com

CARDS/Armstrong/VG1

CARDS is an Air Force sponsored program contracted under the auspices of STARS. There have been successes with domain specific reuse and architecture-based components, but it is difficult to make this approach the standard way of doing business. The CARDS project will address the issues that must be handled in order to make domain specific reuse happen.



CARDS

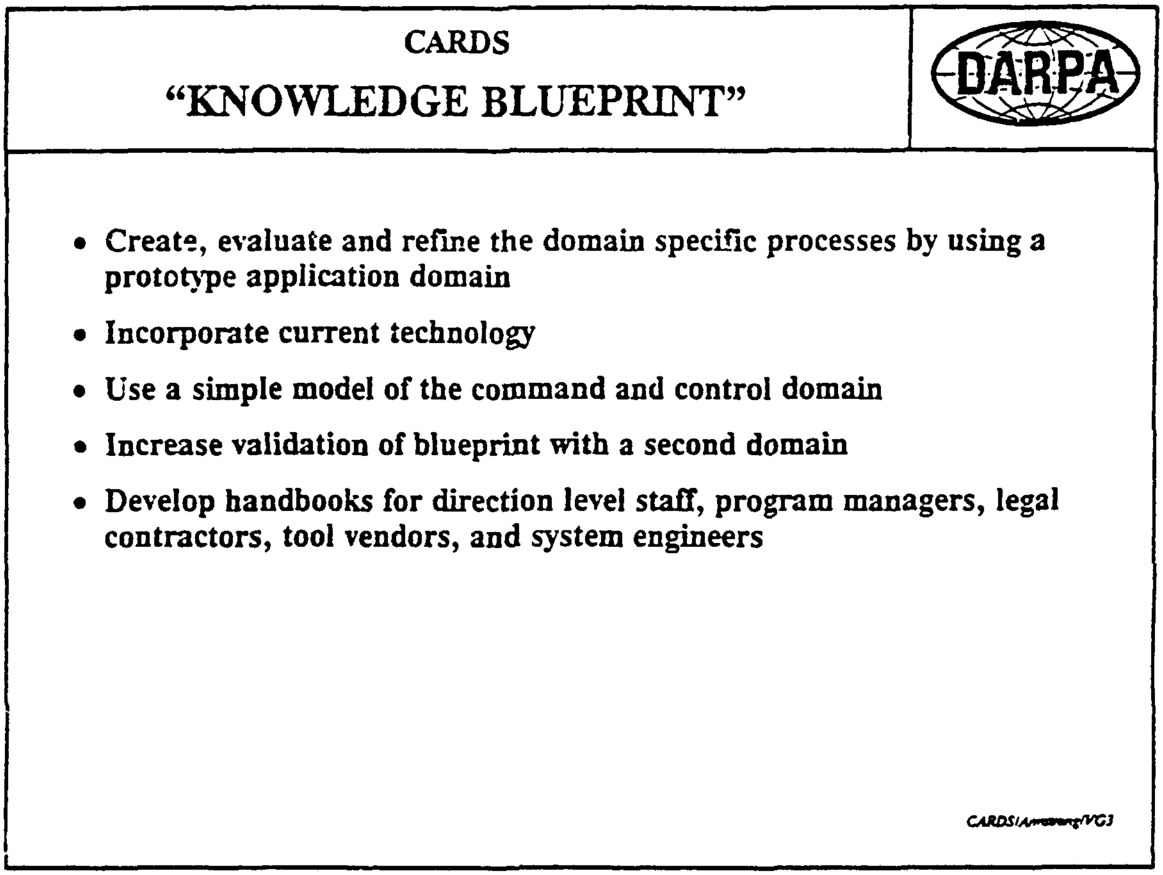
MISSION

DARPA

- Development of a "Knowledge Blueprint" in support of implementing domain-specific reuse
- Expand and refine the prototype reuse environment for command centers
- Help eliminate cu'tural barriers to reuse within the DoD community

CARDS/Armstrong/VG2

CARDS is defining a knowledge blueprint for domain specific reuse. CARDS has developed a simple model of the command and control domain in order to validate the domain reuse processes. Some of the processes that will be included are the domain analysis process, the incorporation of COTS software for use in domain architecture, incorporation of reuse into the software development, and the development of tools to support reuse. The CARDS project will d·velop methods that will help organizations incorporate reuse and provide reuse incentives. Reuse must be integrated completely into the software development lifecycle. It is our intent to look for ways to eliminate many of the barriers to reuse in DoD: to facilitate reuse being treated as an inseparable aspect of the overall software engineering process and to provide recommendations and guidance so DoD can create incentives for reuse.



CARDS

"KNOWLEDGE BLUEPRINT"

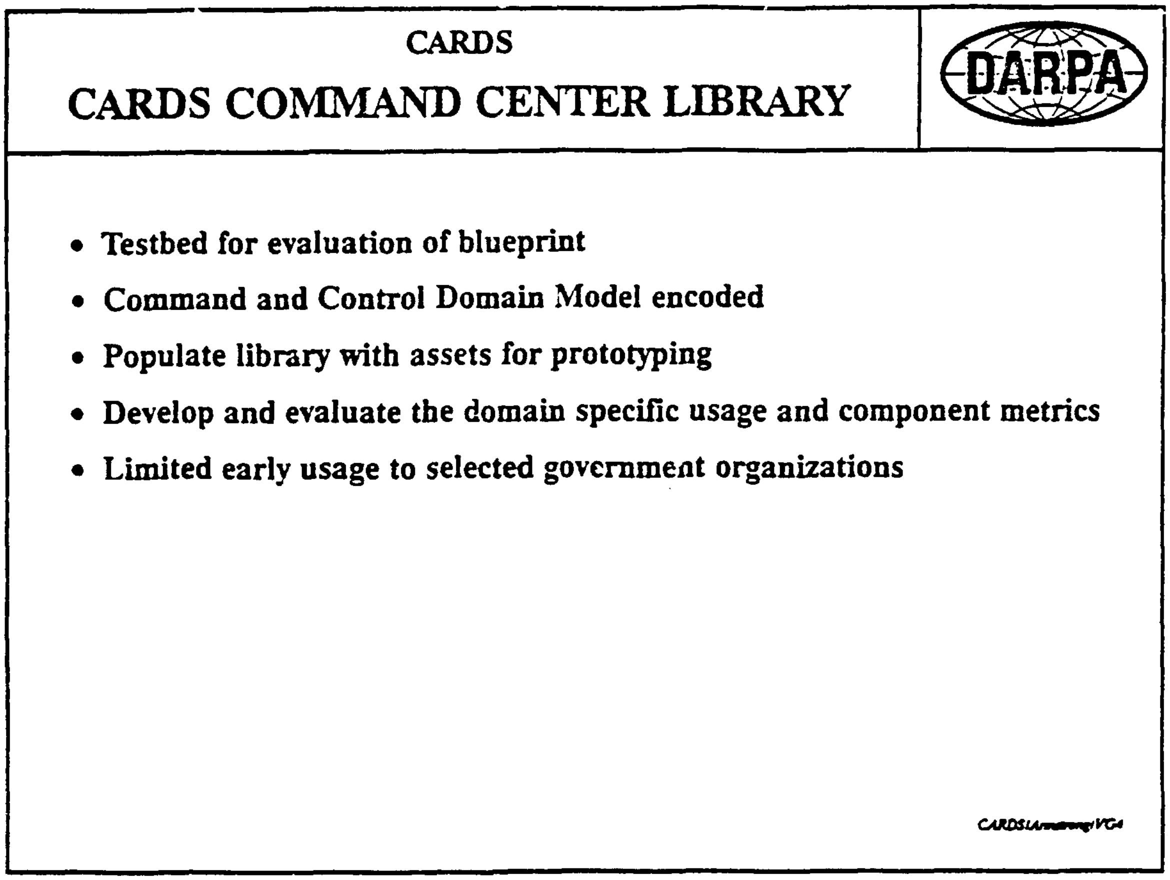
DARPA

- **Create, evaluate and refine the domain specific processes by using a prototype application domain**
- **Incorporate current technology**
- **Use a simple model of the command and control domain**
- **Increase validation of blueprint with a second domain**
- **Develop handbooks for direction level staff, program managers, legal contractors, tool vendors, and system engineers**

CARDS/[illegible]/VG3

The Knowledge Blueprint is a flexible plan that will define the domain specific reuse process. Many people within the software development reuse community have developed reuse processes for various stages of the software development life-cycle. The CARDS project will evaluate and refine already developed processes and create new processes where necessary. CARDS will work with STARS and DoD reuse efforts. The blueprint will be based on the STARS Reuse Concept of Operations and will provide instruction for tailoring other processes for domain specific reuse. Automation possibilities will be investigated for domain specific asset generation and qualification, asset usage evaluation, and system composition.

The Blueprint will be addressed in a series of handbooks that will explain and support domain specific reuse for different audiences. These handbooks will form a large part of the blueprint for reuse being developed by the DoD. A Direction Level Handbook will address domain specific reuse for individuals having responsibility for an application domain. An Acquisition Handbook will be targeted to DoD program managers, legal and contracting personnel. Issues such as data rights, cost benefits of reuse, license agreements, incentives and model contract wording will be addressed. Acquisition and management strategies of existing government and industry reuse programs and prototypes will be reviewed. An Engineer's Handbook will provide clear guidance with specific actions necessary to fully exploit the benefits of reuse-based rapid prototyping using a domain specific library and domain knowledge. A Component Developer's and Tool Vendor's Handbook will define actions necessary to develop reusable assets and support tools.



CARDS

CARDS COMMAND CENTER LIBRARY

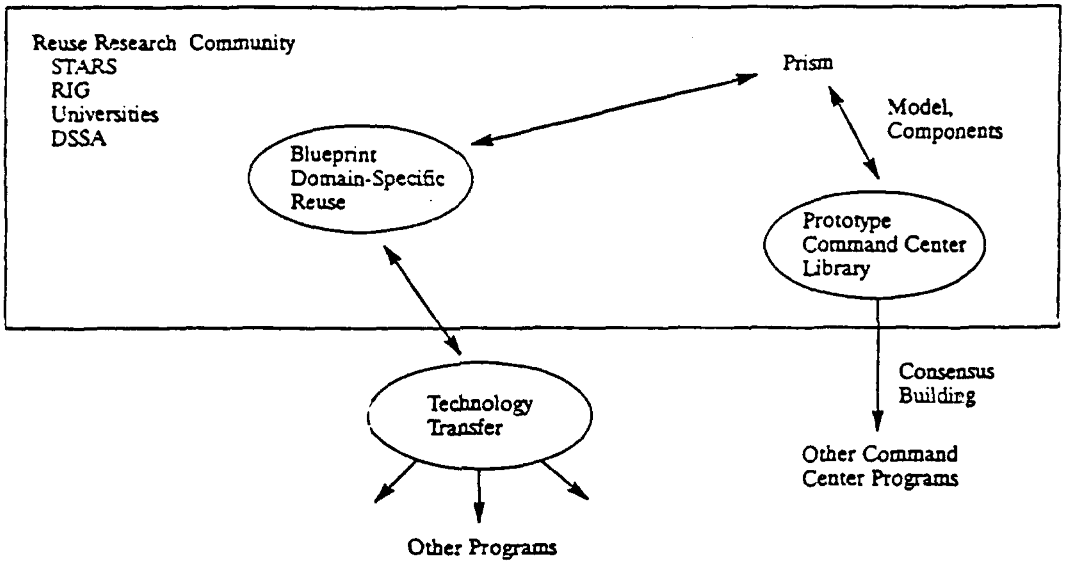
- Testbed for evaluation of blueprint
- Command and Control Domain Model encoded
- Populate library with assets for prototyping
- Develop and evaluate the domain specific usage and component metrics
- Limited early usage to selected government organizations

CARDSLA[illegible]VG4

The CARDS command center library has been encoded with a simple command and control domain model done by ESD/AVS domain experts. The library has been populated with components provided by ESD/AVS. CARDS has begun identification of domain specific usage and component metrics. Library functions are based on processes already developed by other reuse libraries. CARDS command center library primary goals are to validate the CARDS blueprint (discussed in Track 4), and thus facilitates the introduction of Domain-Specific Reuse into other organizations. At the present time only four user sites will be designated with two possible sites added within the next year.

CARDS

DARPA

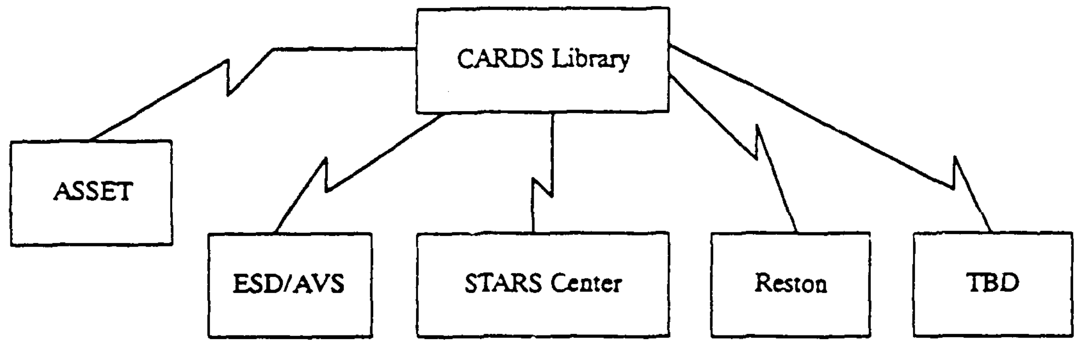


CARDS\A———VGS

This slide graphically illustrates the cooperative work going on between CARDS, STARS, RIG, Universities, DSSA, Prism and other projects within the reuse development community. The blueprint will be developed with cooperation of other organizations. The RIG and ALOAF will support the necessary processes needed for interoperability between libraries. In order to change the cultural barriers to reuse, education must take place within academia. CARDS will support the development of reuse based software engineering curriculums. Prism will be developing a formal command and control domain model, defining requirements and developing components necessary for this domain. CARDS hopes to interact with the Prism staff in order to validate the domain model, requirements, components and the necessaiy processes.

CARDS

CARDS NETWORK



CARDS Library

ASSET

ESD/AVS

STARS Center

Reston

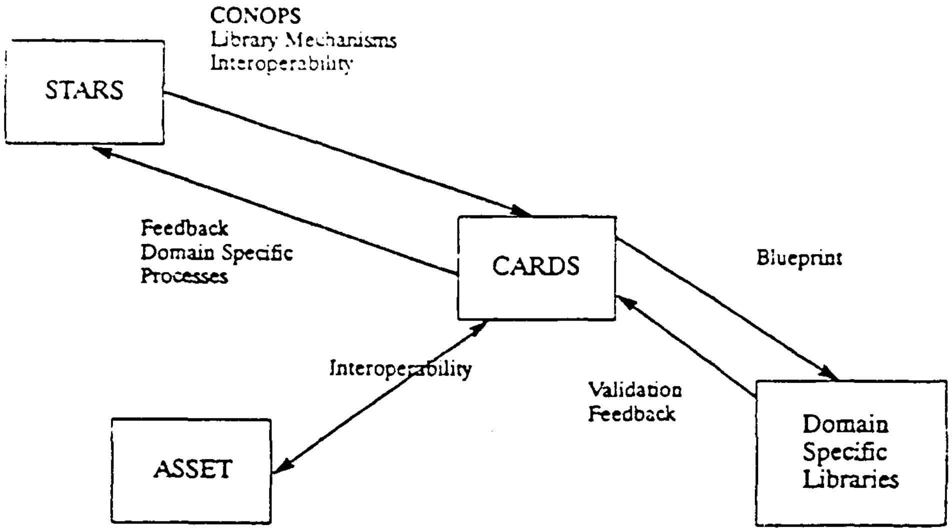
TBD

CARDS/[illegible]/VG6

CARDS will be connected to the following sites via Internet. Interoperability with ASSET will continue to mature over the next year. A direct line to ESD/AVS will be activated by the new year. The fourth user site still needs to be determined.

CARDS

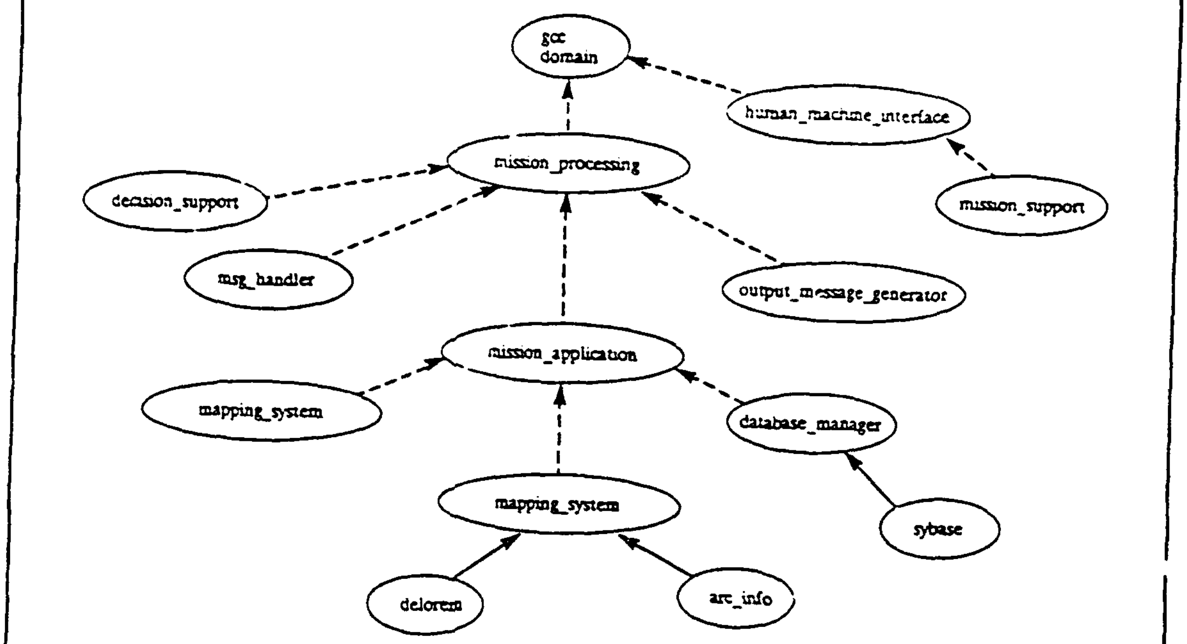
REUSE PIPELINE



CARDS/[illegible]/VG7

CARDS as part of a reuse pipeline is illustrated by this slide. The STARS CONOPS is the basis of the reuse blueprint. CARDS is implemented using the RLF, one of the library mechanisms developed under STARS. CARDS will support and validate the standards for interoperability that is produced by the RIG, a STARS initiative group and ALOAF by interoperating with the ASSET project. CARDS will provide back to the STARS programs feedback on the processes and tools that are used by the CARDS project. The blueprint will be validated with a second domain.

CARDS
FORMALLY ENCODED MODEL—GENERIC COMMAND AND CONTROL SAMPLE



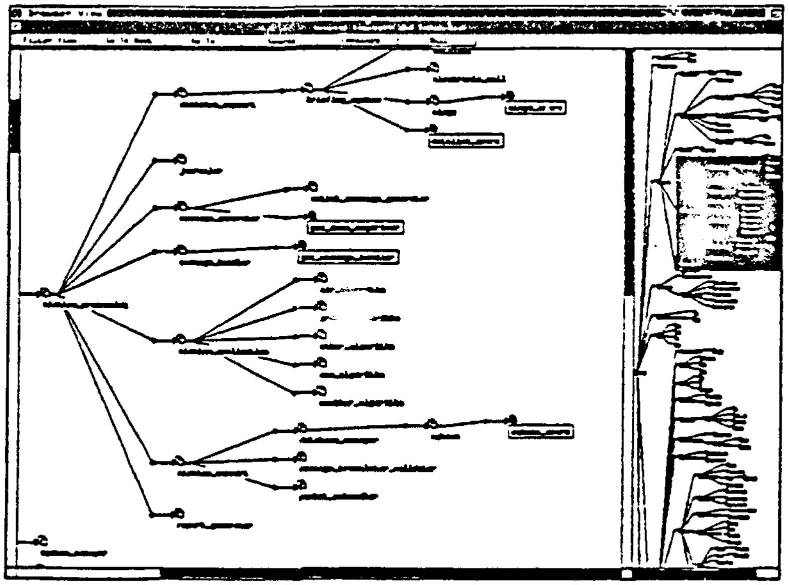
CARDS\Armstrong/VG8

The encoded model for CARDS is the command and control domain. This generic domain model was developed by the Air Force at ESD/AVS. The model as shown illustrates the high level of abstractions within a semantic net that CARDS supports. The components in the instances shown on this slide are: sybase, delorem and arc_info. For example, Sybase is the instance that meets the requirements for the database manager. The database manager is necessary to the implementation of the mission application portion of command and control.

CARDS

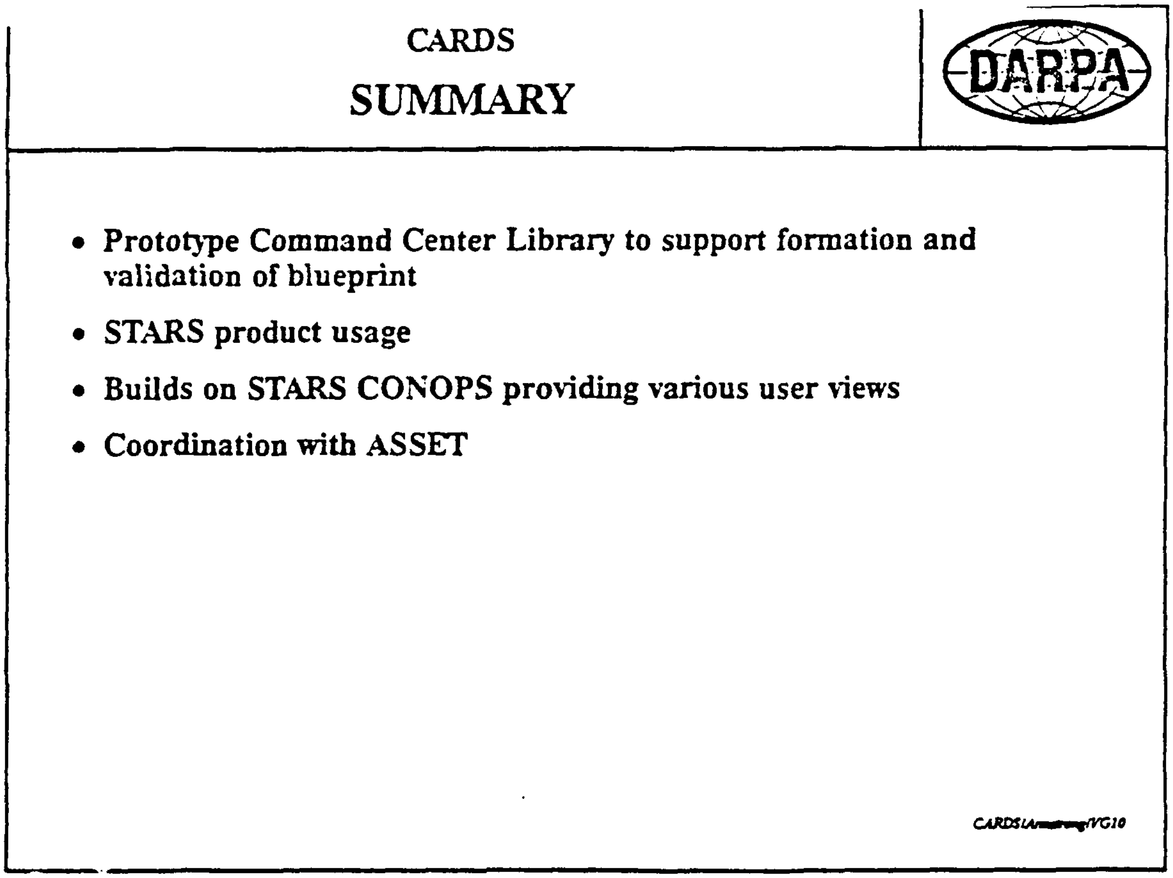
FORMALLY ENCODED MODEL—GENERIC COMMAND AND CONTROL SAMPLE

DARPA



CARDS LA/[illegible]/VG9

Illustrated is a snapshot of the encoded command and control domain model within the RLF. One of the modules of the RLF is a Graphical Browser, the user interface. The Graphical Browser builds a tree structure that illustrates some of the relationships between the components in the domain architecture. Pop up menus are used to navigate the structure and define the relationships and attributes of the domain model components. The topography will indicate to the user their placement within the domain structure at any given time.



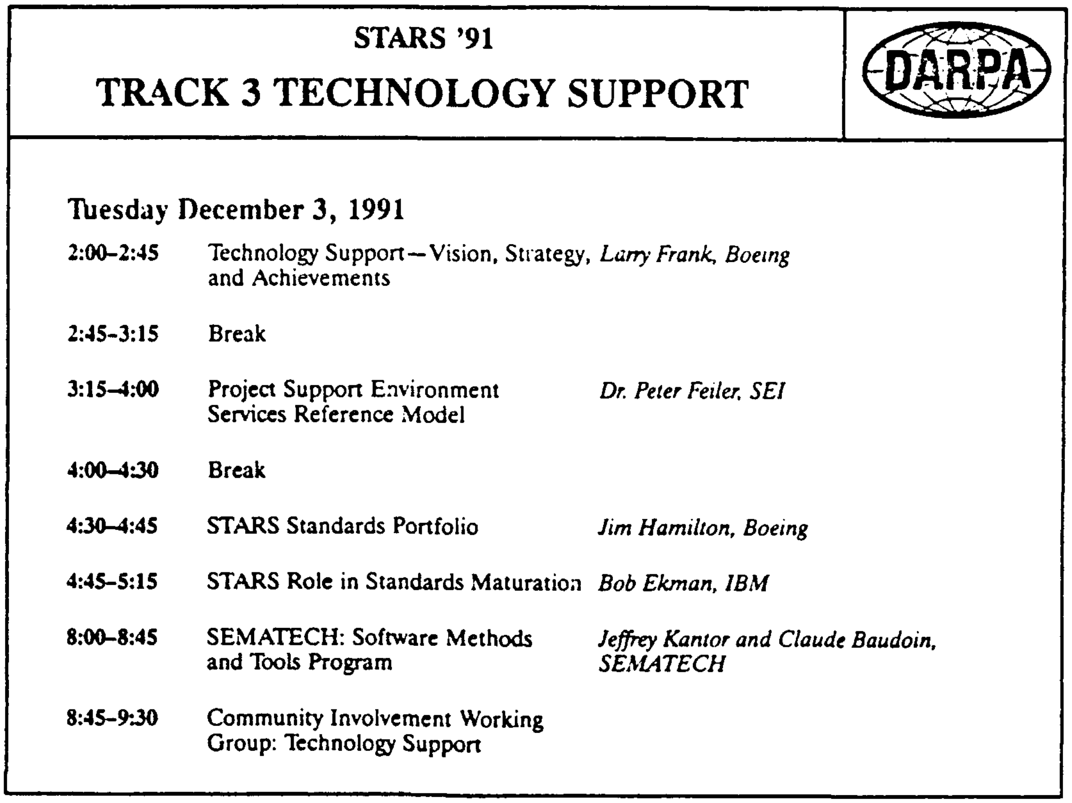
CARDS

SUMMARY

- Prototype Command Center Library to support formation and validation of blueprint
- STARS product usage
- Builds on STARS CONOPS providing various user views
- Coordination with ASSET

CARDS/Armstrong/VG10

In summary, I would like to emphasize the goals of CARDS. CARDS primary goal is to develop and validate the "knowledge blueprint" by prototyping the command and control domain and a second domain. CARDS will build on, validate and feedback to the process strengthen over the next year. We already have meetings on a regular basis with the ASSET staff. The purpose of these meeting are to collaborate efforts wherever possible. For instance, metrics, library policies and procedures, backup exchange and interoperability are some of the current areas of coordination.



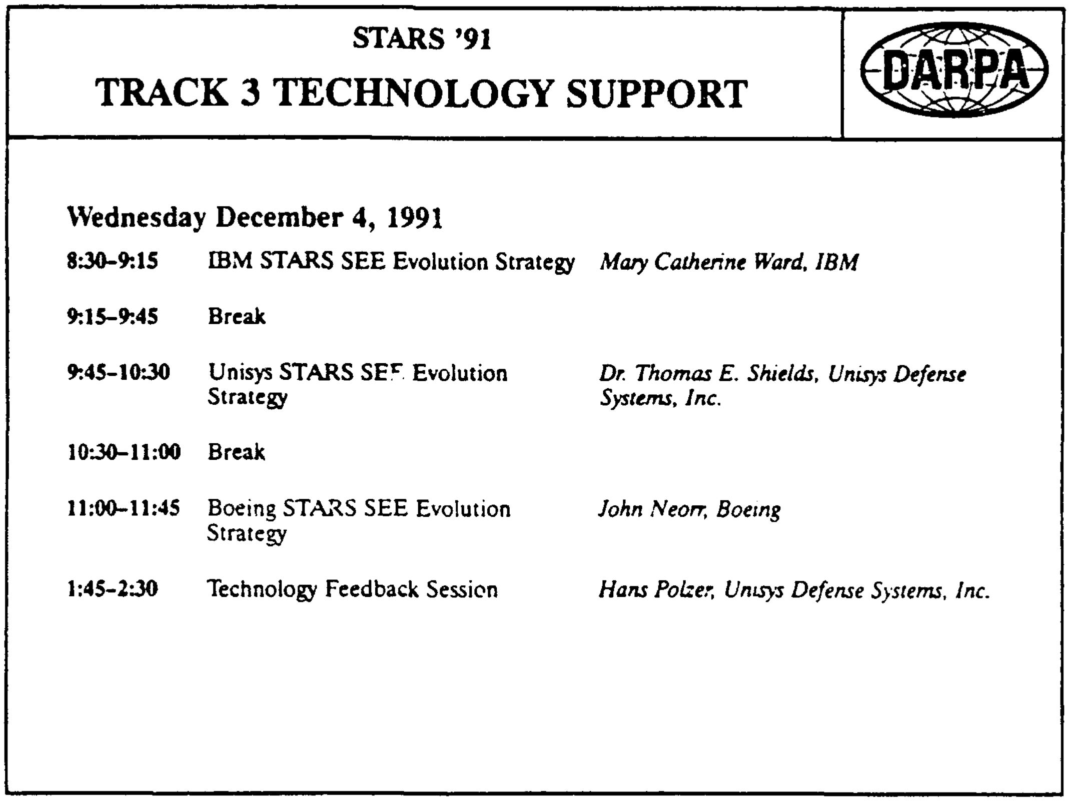
STARS '91

TRACK 3 TECHNOLOGY SUPPORT

DARPA

Tuesday December 3, 1991

2:00–2:45	Technology Support—Vision, Strategy, and Achievements	*Larry Frank, Boeing*
2:45–3:15	Break	
3:15–4:00	Project Support Environment Services Reference Model	*Dr. Peter Feiler, SEI*
4:00–4:30	Break	
4:30–4:45	STARS Standards Portfolio	*Jim Hamilton, Boeing*
4:45–5:15	STARS Role in Standards Maturation	*Bob Ekman, IBM*
8:00–8:45	SEMATECH: Software Methods and Tools Program	*Jeffrey Kantor and Claude Baudoin, SEMATECH*
8:45–9:30	Community Involvement Working Group: Technology Support	



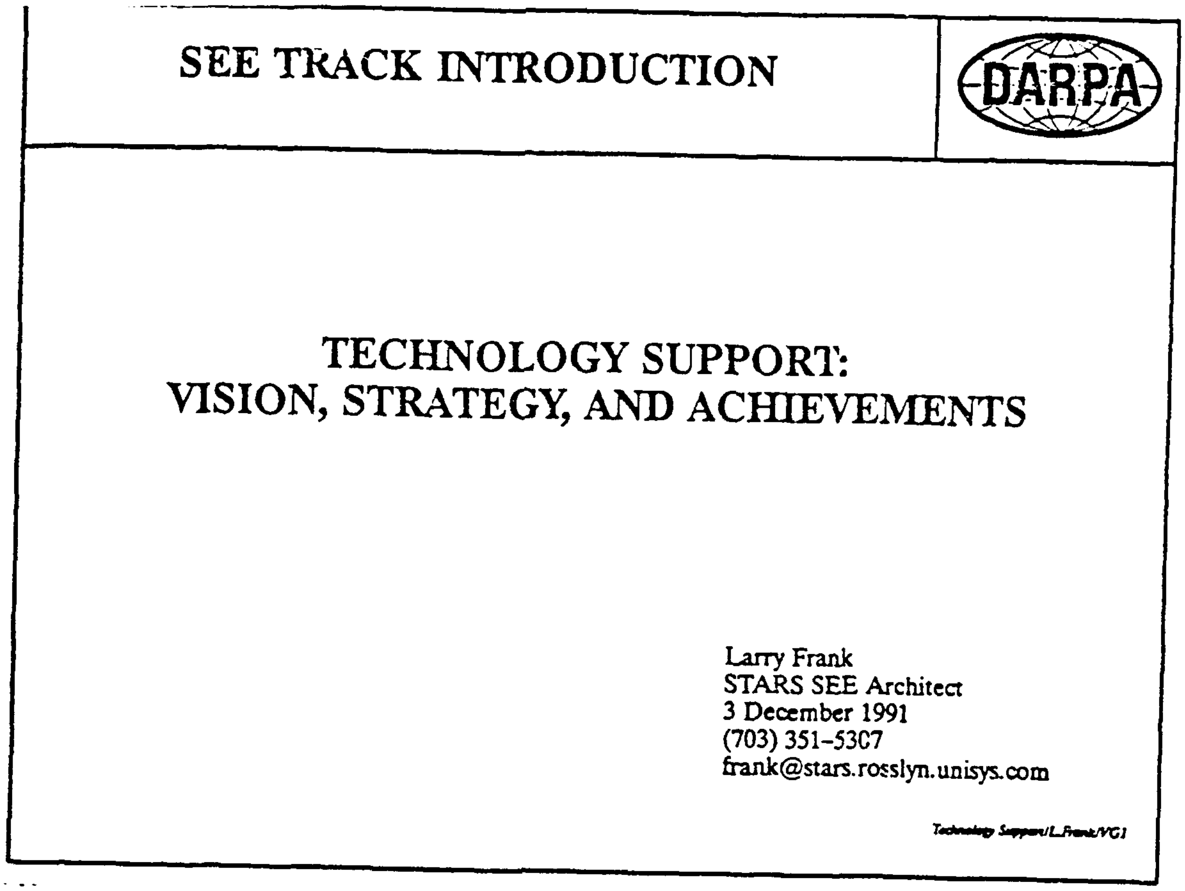
STARS '91

TRACK 3 TECHNOLOGY SUPPORT

DARPA

Wednesday December 4, 1991

8:30–9:15	IBM STARS SEE Evolution Strategy	*Mary Catherine Ward, IBM*
9:15–9:45	Break	
9:45–10:30	Unisys STARS SEE Evolution Strategy	*Dr. Thomas E. Shields, Unisys Defense Systems, Inc.*
10:30–11:00	Break	
11:00–11:45	Boeing STARS SEE Evolution Strategy	*John Neorr, Boeing*
1:45–2:30	Technology Feedback Session	*Hans Polzer, Unisys Defense Systems, Inc.*



SEE TRACK INTRODUCTION

DARPA

TECHNOLOGY SUPPORT: VISION, STRATEGY, AND ACHIEVEMENTS

Larry Frank
STARS SEE Architect
3 December 1991
(703) 351-5307
frank@stars.rosslyn.unisys.com

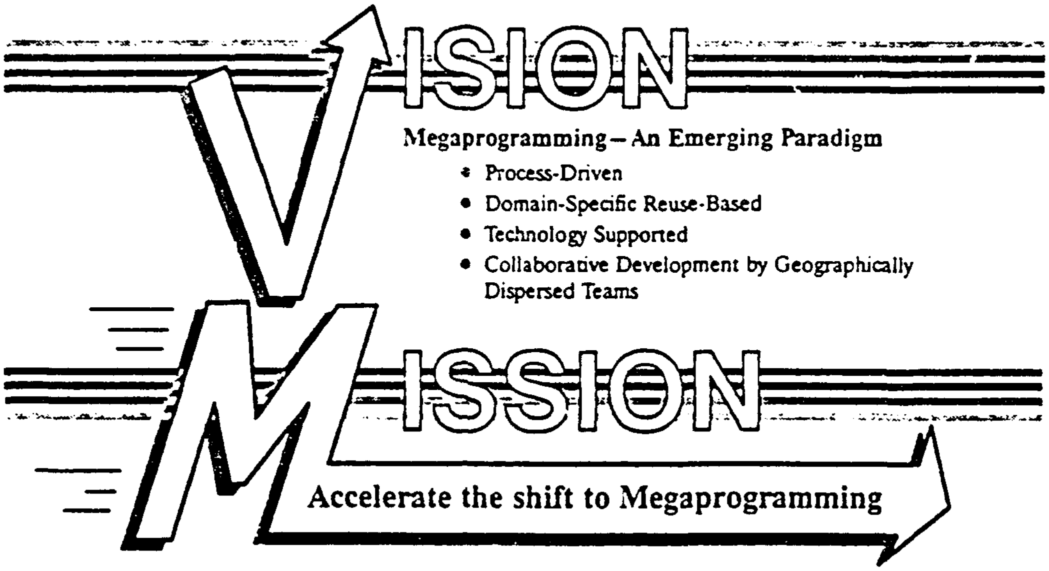
Technology Support/L_Frank/VG1

VG1 Cover

TECHNOLOGY SUPPORT

STARS VISION MISSION

DARPA



Megaprogramming – An Emerging Paradigm

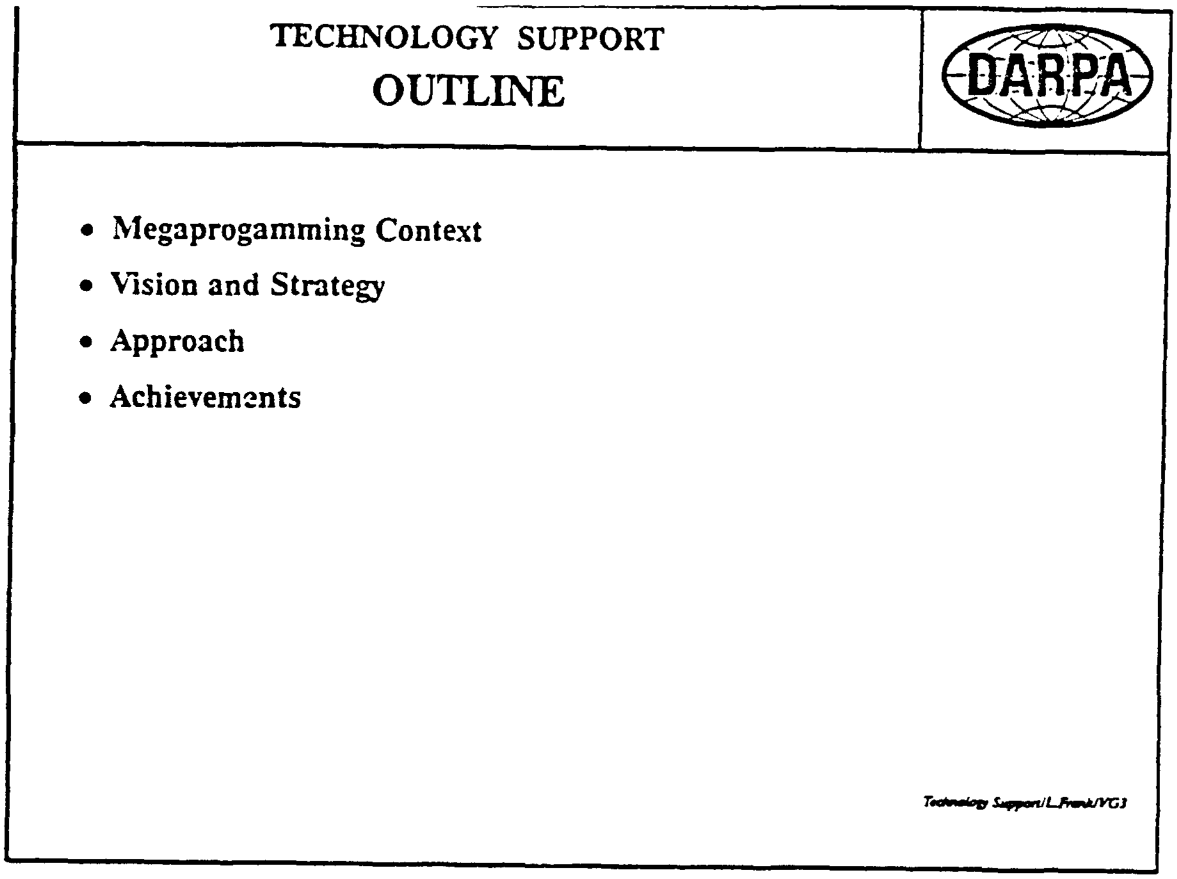
- Process-Driven
- Domain-Specific Reuse-Based
- Technology Supported
- Collaborative Development by Geographically Dispersed Teams

Accelerate the shift to Megaprogramming

Technology Support/L.Frank/VG2

VG2 Title: STARS Vision/Mission

In his opening plenary presentation, John Foreman addressed the STARS vision, mission, and strategy in accelerating the shift to a megaprogramming model of software development. This presentation will describe, at a global level, how technology will be incorporated within a Software Engineering Environment (SEE) and evolved into a well-integrated, adaptable, tailorable environment supporting a process-driven, reuse-based engineering approach to megaprogramming.



TECHNOLOGY SUPPORT

OUTLINE

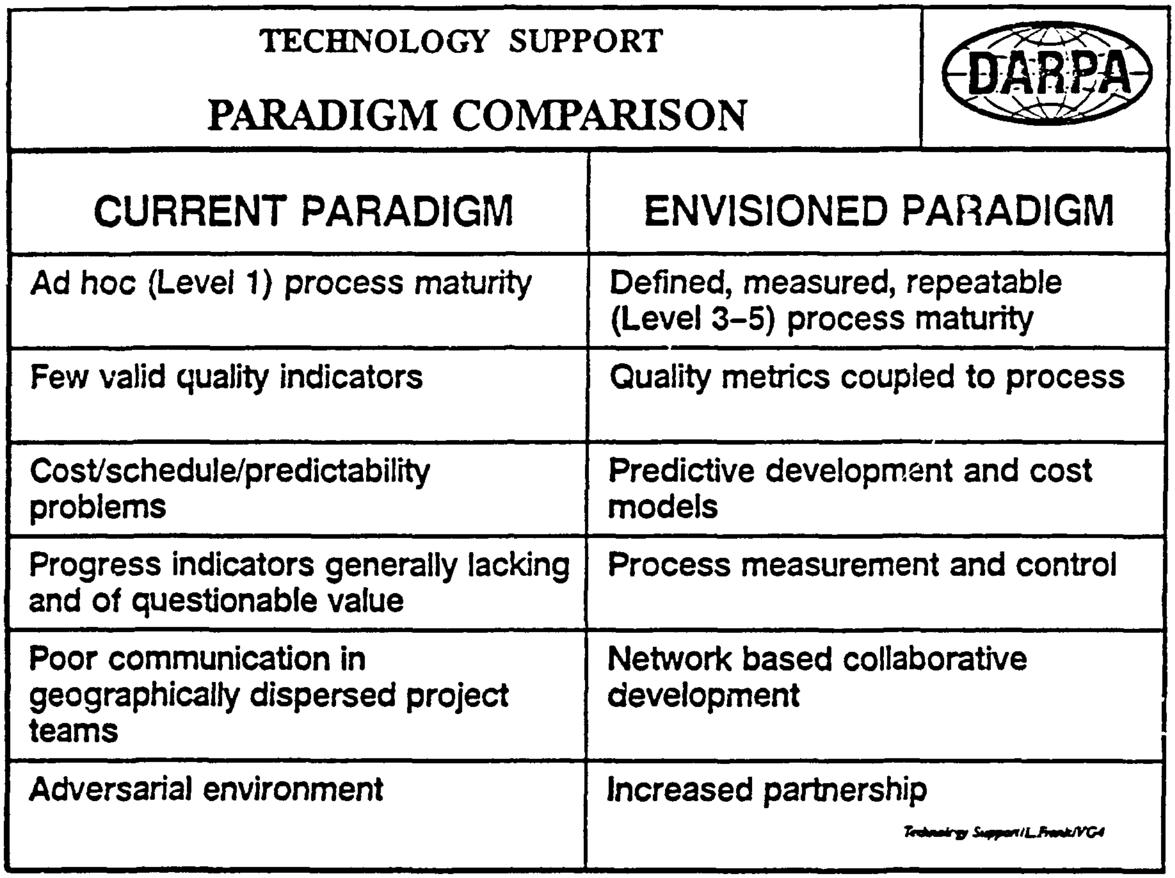
- Megaprogamming Context
- Vision and Strategy
- Approach
- Achievements

Technology Support/L_Frank/VG3

VG3 Title: Outline

In order to understand the nature of the requisite technology support for this shift to a megaprogramming paradigm, we will context megaprogramming by comparing the envisioned paradigm with current practices. A model of how SEEs are evolving will be presented along with a vision of what is entailed in fostering this evolution and the strategy for achieving it.

The high-level activities reflecting the STARS approach will be discussed along with the process of evolving the SEE. Various aspects of the SEE evolution will be explored and notable achievements discussed.



TECHNOLOGY SUPPORT

PARADIGM COMPARISON

CURRENT PARADIGM	ENVISIONED PARADIGM
Ad hoc (Level 1) process maturity	Defined, measured, repeatable (Level 3–5) process maturity
Few valid quality indicators	Quality metrics coupled to process
Cost/schedule/predictability problems	Predictive development and cost models
Progress indicators generally lacking and of questionable value	Process measurement and control
Poor communication in geographically dispersed project teams	Network based collaborative development
Adversarial environment	Increased partnership

Technology Support/L.Frank/VG4

VG4–VG5 Title: Paradigm Comparison

Message: The goal of the megaprogramming software engineering approach is to develop unprecedented systems from precedented components using defined, repeatable, and measureable processes and to be able to predict the costs of the development.

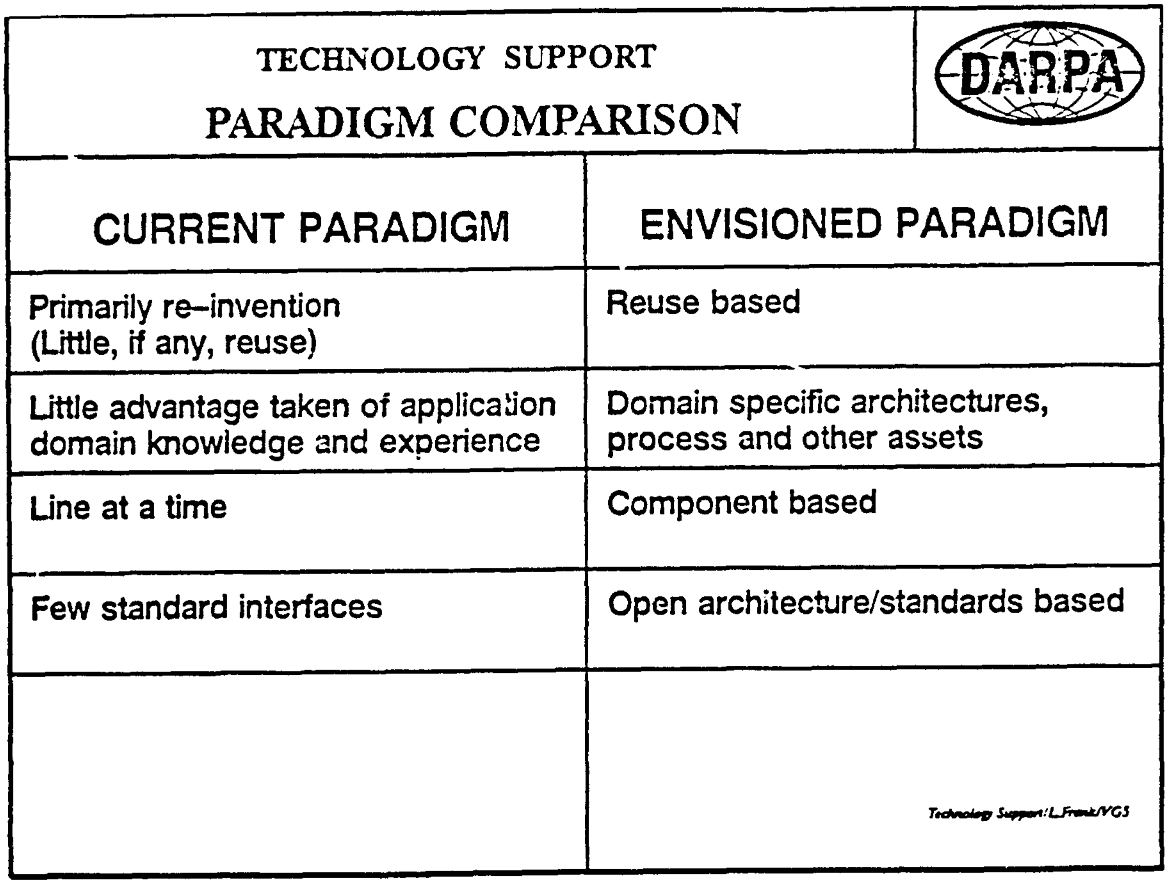
We offer some insight into what the characteristics of the megaprogramming paradigm is by contrasting and comparing it with those of the current software engineering approach. Any engineering or development approach is predicated on a development methodology and the underlying processes which support it.

The notion of being process–driven is based on the ability to manage development based on well–defined, repeatable, and measureable processes in contrast to trying to manage development activities based on an ad–hoc approach. In order to provide continuous quality improvement, it is necessary to quantify, capture, and analyze quality indicators both in terms of the product and the process(es) employed to produce that product.

Greater organizational process maturity (as indicated by the SEI process maturity levels), embodying the above principles, leads to the ability to predict development costs earlier in the system life–cycle, and hence, provides the lead time to take corrective action.

In large scale, complex development projects, development teams will, likely, be split across organizational boundaries and across geographically dispersed sites. This further exacerbates the already difficult task of communication among these teams. Existing processes will have to evolve to enable the effective management and control of collaborative efforts. Tooling will also evolve to meet demands of network–based collaborative development.

Currently, reuse of assets occur, primarily, by transferring domain knowledge from one project to another in the form of the knowledge base represented by the project engineers and developers. The envisioned paradigm calls forth the (re)use of domain architectures, process and other reusable assets. In particular, we would like to be able to synthesize larger and larger components from precedented components. (continued)



TECHNOLOGY SUPPORT

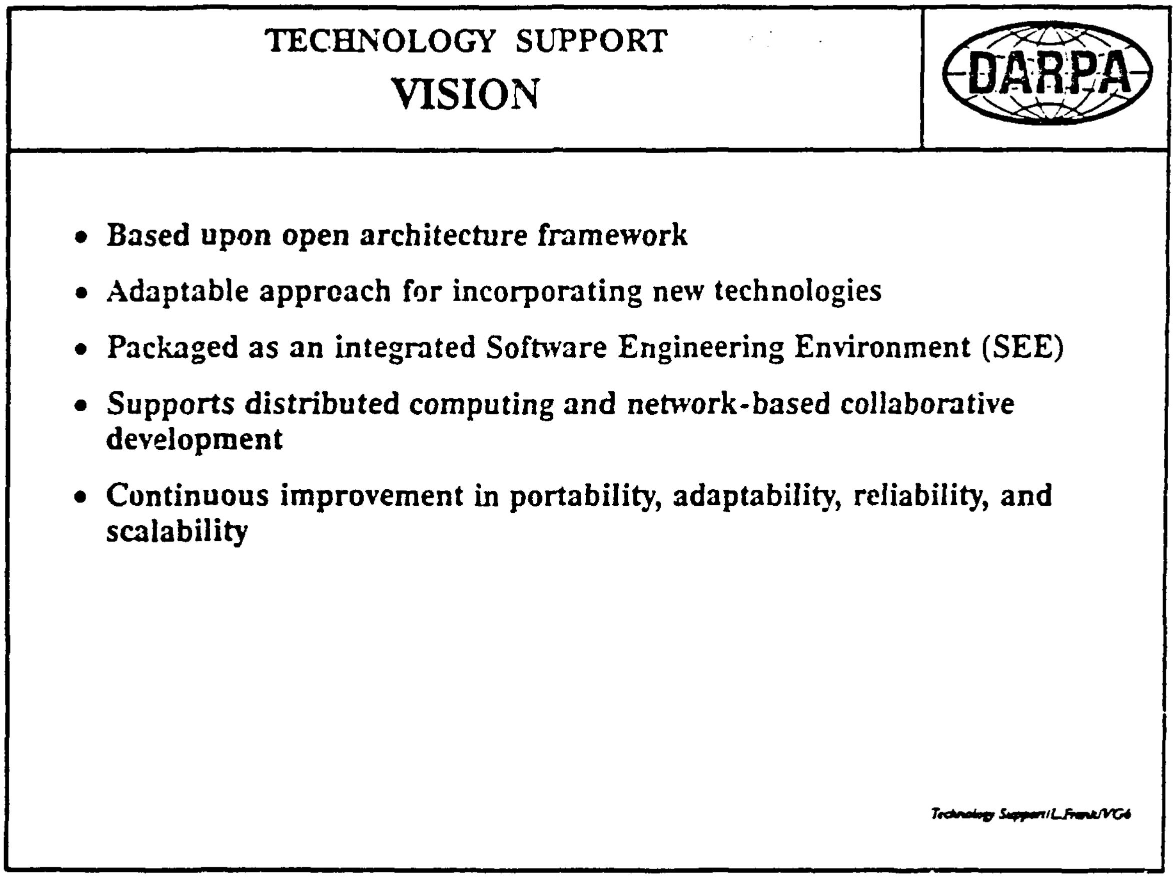
PARADIGM COMPARISON

CURRENT PARADIGM	ENVISIONED PARADIGM
Primarily re–invention (Little, if any, reuse)	Reuse based
Little advantage taken of application domain knowledge and experience	Domain specific architectures, process and other assets
Line at a time	Component based
Few standard interfaces	Open architecture/standards based
	Technology Support:L.Frank/VG5

VG4–VG5 Title: Paradigm Comparison (continued)

Current development is too often based upon treating each new development effort as unprecedented. Algorithms, utility "modules", and, at times, specifications are reused, but on an ad-hoc basis. Under the envisioned paradigm, composition rules and "module" interface formalisms will be defined, evolved, and matured to enable the synthesis of unprecedented systems from precedented components.

Finally, in order to achieve the full benefits of megaprogramming, increased partnership among users, developers, and the government will become necessary. Increased partnership between the user community and the developers is becoming a reality. Many current development strategies are predicated upon it, when practiced. But many barriers between the government and system developers currently exist. In particular, current acquisition and procurement policies do not provide the requisite incentives to the contractors (developers) to warrant the full employment and sharing of reuse processes and assets across projects.



TECHNOLOGY SUPPORT

VISION

- Based upon open architecture framework
- Adaptable approach for incorporating new technologies
- Packaged as an integrated Software Engineering Environment (SEE)
- Supports distributed computing and network-based collaborative development
- Continuous improvement in portability, adaptability, reliability, and scalability

Technology Support/L_Frank/VG6

VG6 Title: Vision

Message: The SEE is the delivery vehicle for provisioning services to the systems builder and integrator, and ultimately to the end user. The underiying framework serves as the integration platform upon which tools are deplcyed and integrated on a needs-driven basis.

Past attempts at providing monolithic environments that are responsive to development needs, irrespective of application domain or projects within domains, have not met with great success. The STARS approach is based upon an open architecture framework. This framework consists of extensible core services which utilize a set 1st line of text: Thames 10pt flush leftof open standards as documented in the STARS Standards Portfolio SSP).

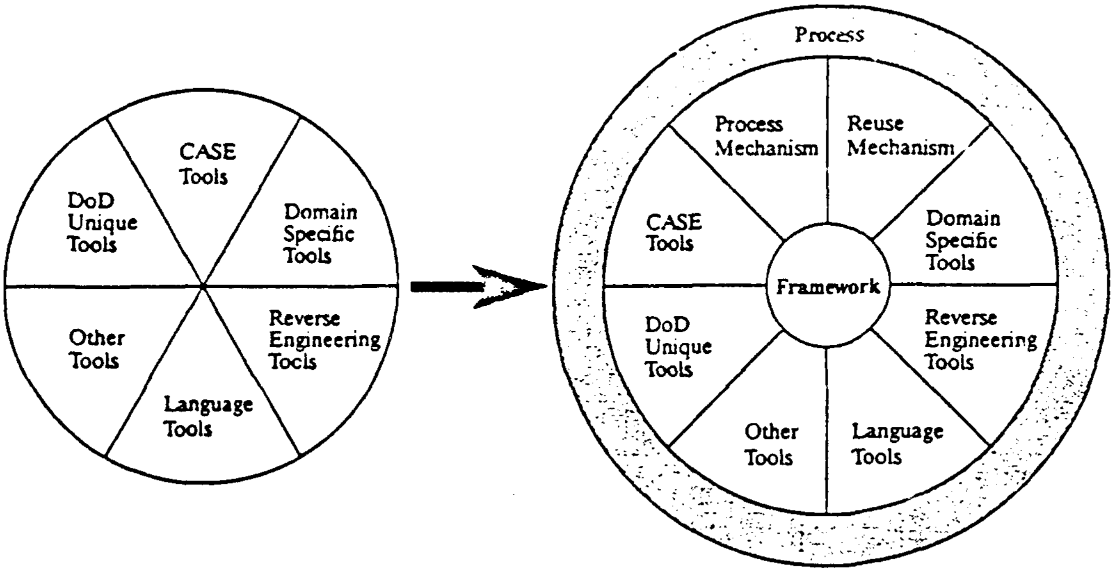
As tools are integrated into this framework-based, open architecture environment to meet the demands and needs of the development project, further standards will be identified, as needed, within the profile to accommodate tool and data interoperability. This open SEE architecture will support a "plug and play" environment, easily adaptable to the application domain and tailorable to the specific project needs.

It will also enable continuous improvement in tool portability across platforms and environments, adaptability of tools to changing development processes and needs, reliability of installed components and the environment itself, and the scalability of system components within the environment reflecting the scalabiltity of process formalisms supporting distributed computing and network collaborative development.

TECHNOLOGY SUPPORT

SEE EVOLUTION

- Transition to Framework – Based Environments Incorporating Process and Reuse Capabilities



Technology Support/L.Frank/VG7

VG7 Title: SEE Evolution

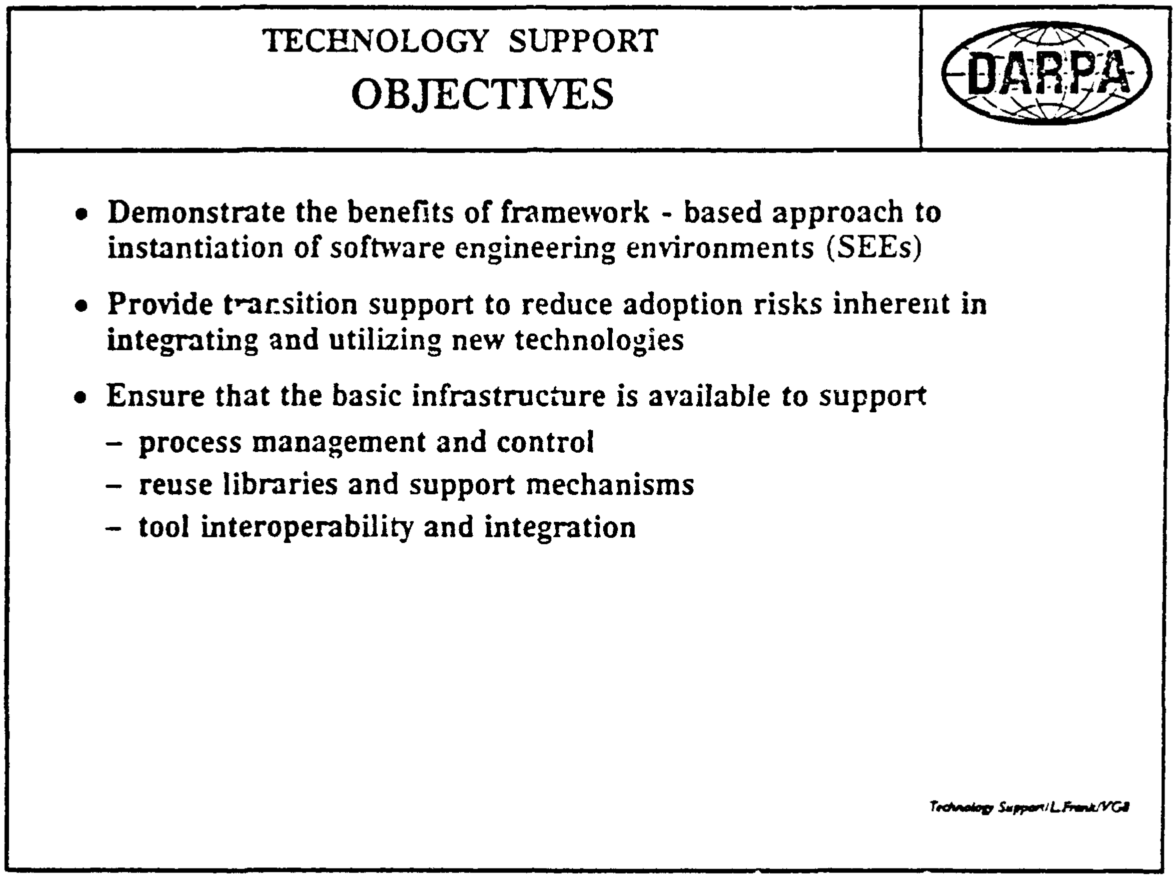
Message: Framework-based environments provide greater flexibility in the ability to adapt and tailor environments to project needs. They also provide the potential for vendors to reduce their tool development costs. Also, STARS provides value-added capabilities by way of process and reuse mechanisms.

Current SEEs exact a heavy cost both to the software developer and the tool builder. To integrate N tools within an environment potentially requires dealing with O(N**2) interfaces with respect to information sharing between and among those N tools. By integrating these same tools with the framework, and using framework provided services as well as the repository services, the integration problem can be reduced to dealing with O(N) interfaces. Thus, the problem can be reduce by an order of magnitude (as a function of the number of tools within the environment).

Also, vendors currently provide (within each tool) many of the selfsame services as are provided in the framework. With the advent of the framework-based environment, tool builders could take advantage of these framework services to obviate the necessity of developing and maintaining these services as part of the tool infrastructure.

Ultimately, and ideally (from a environment integrator's perspective), we would like to see tool architectures that provide visibility of functional interfaces within the tool (or tool suite) so that other layered services or tools could invoke that functionality while conforming to the standards that infuse an open systems architecture.

While STARS provides value-added capabilities in the areas of reuse and process mechanisms, the envisioned environment, instantiated for a given development project, is guided by the organization's business processes as well as the processes that inform the development methodology. This is the basis for what is meant by process-driven development.



TECHNOLOGY SUPPORT

OBJECTIVES

- Demonstrate the benefits of framework - based approach to instantiation of software engineering environments (SEEs)
- Provide transition support to reduce adoption risks inherent in integrating and utilizing new technologies
- Ensure that the basic infrastructure is available to support
  - process management and control
  - reuse libraries and support mechanisms
  - tool interoperability and integration

Technology Support/L.Frank/VG8

VG8 Title: Objectives

Message: In order to realize the potential productivity increases that megaprogramming portends, it must be technically feasible, reduce overall life-cycle development costs, and deliver quality systems in a timely manner (better, cheaper, faster).

Megaprogramming will dramatically impact the way systems will be built. Tools, by themselves, can not provide the needed productivity gains. The system development methodology and associated processes and disciplines must change to reflect the new development paradigm. Tools, in a real sense, reflect the automation of these (sub)processes.

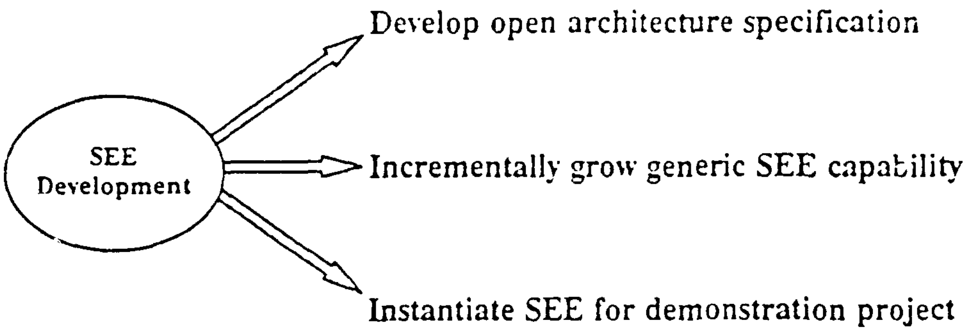
As with any fundamental change in the way business is done, the shift to a megaprogramming paradigm will encounter cultural impediments within the adopting organization. To overcome these barriers and to reduce the inherent risks to its adoption, STARS will demonstrate the efficacy and efficiency of the framework-based approach on real DoD programs.

STARS will provide transition support to the selected demonstration projects to aid project staff in fully exploiting its capabilities while minimizing the impacts of adoption within the project organization itself.

TECHNOLOGY SUPPORT

ACTIVITIES

DARPA



SEE Development

Develop open architecture specification

Incrementally grow generic SEE capability

Instantiate SEE for demonstration project

Technology Support/L.Frank/VG9

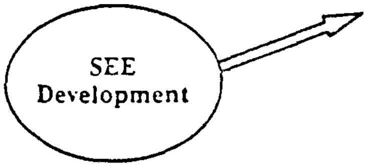
VG9 Title: Activities

Message: Monolithic SEEs fail to meet the evolving needs of the development environment. They are difficult to adapt to multiple application domains and to tailor to specific project needs within those domains. Moreover, they are expensive to maintain.

To fulfill the stated objectives, the STARS strategy is: to identify and build the SEE infrastructure based on an open architecture framework; to augment its basic capabilities by integrating tools within the framework-based environment; and to instantiate and deploy these SEEs on the selected demonstration projects.

TECHNOLOGY SUPPORT

ACTIVITIES DETAIL



Develop open architecture specification

- Identify candidate industry standards
- Identify core service requirements
- Support open architecture working group
- Involve user and vendor communities
- Evolve specification
- Conduct risk reduction prototyping activities
- Develop top level information model

Technology Support/L.Frank/VG10

VG10 Title: Detailed Activities

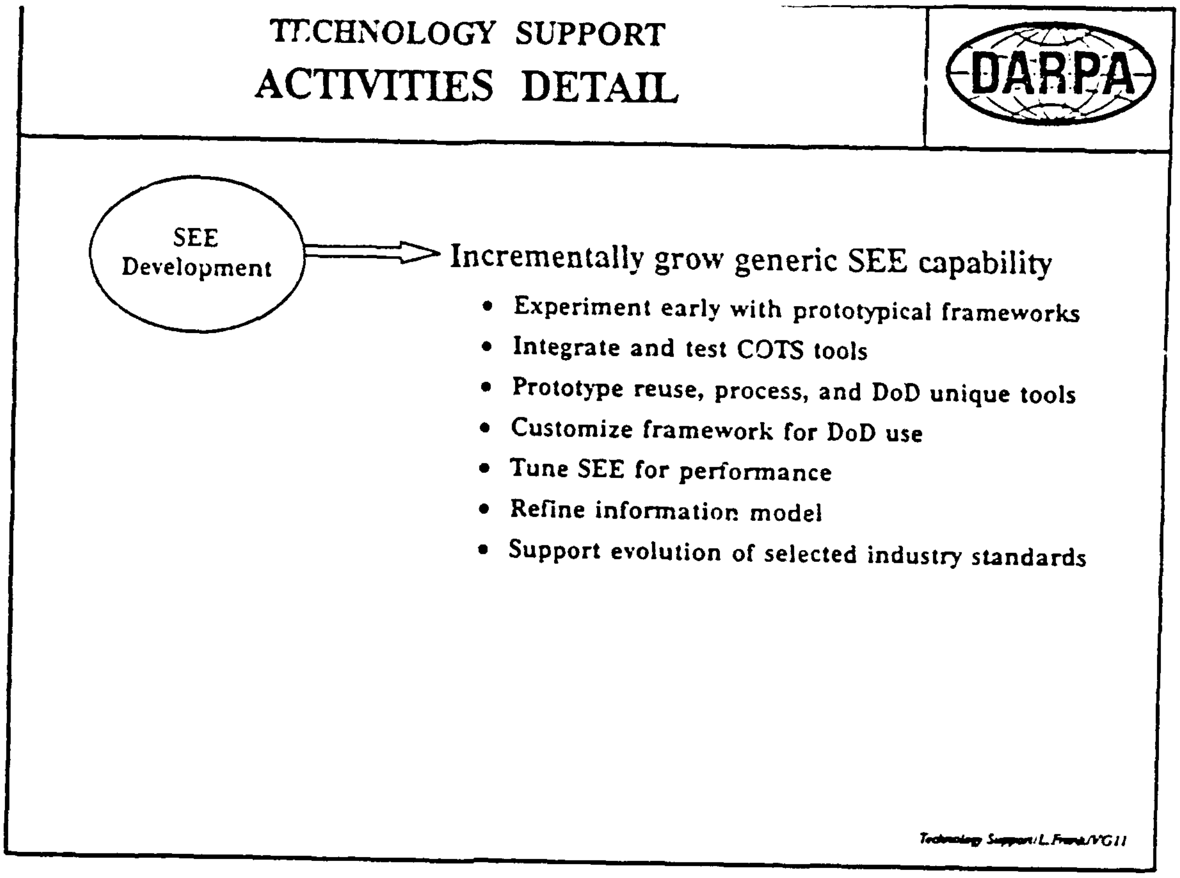
Message: The critical elements supporting an open architecture specification are: identification of the core services, relevant standards, and supporting information model.

STARS has defined the requi:ements and criteria that characterize the extensible set of framework services with input from external working groups addressing similar problems. These were used to identify candidate open standards which were subsequently profiled in the SSP (see Jim Hamilton's and Bob Ekman's presentations).

Since STARS seeks to fully exploit commercial products in the SEE build–outs, the vendor community was briefed on this standards portfolio at the CASE Vendors Workshop in July 1991. The vendor community was given the opportunity to react to the standards portfolio. The reaction was, generally, favorable.

To improve the usability of the framework–based SEE, STARS is currently engaged in prototyping tool portability and tool-to- framework integration and interoperability. The results of the experiments and protyping activities will be documented in reports and lessons learned documents.

To further improve tool and framework interoperability, we are working closely with the primes' commercial counterparts and other interested parties to derive top level information models supporting both framework services and augmented SEE capabilities as they are integrated within the environment.



TECHNOLOGY SUPPORT

ACTIVITIES DETAIL

DARPA

SEE Development

Incrementally grow generic SEE capability

- Experiment early with prototypical frameworks
- Integrate and test COTS tools
- Prototype reuse, process, and DoD unique tools
- Customize framework for DoD use
- Tune SEE for performance
- Refine information model
- Support evolution of selected industry standards

Technology Support/L.Frank/VG11

VG11 Title: Detailed Activities

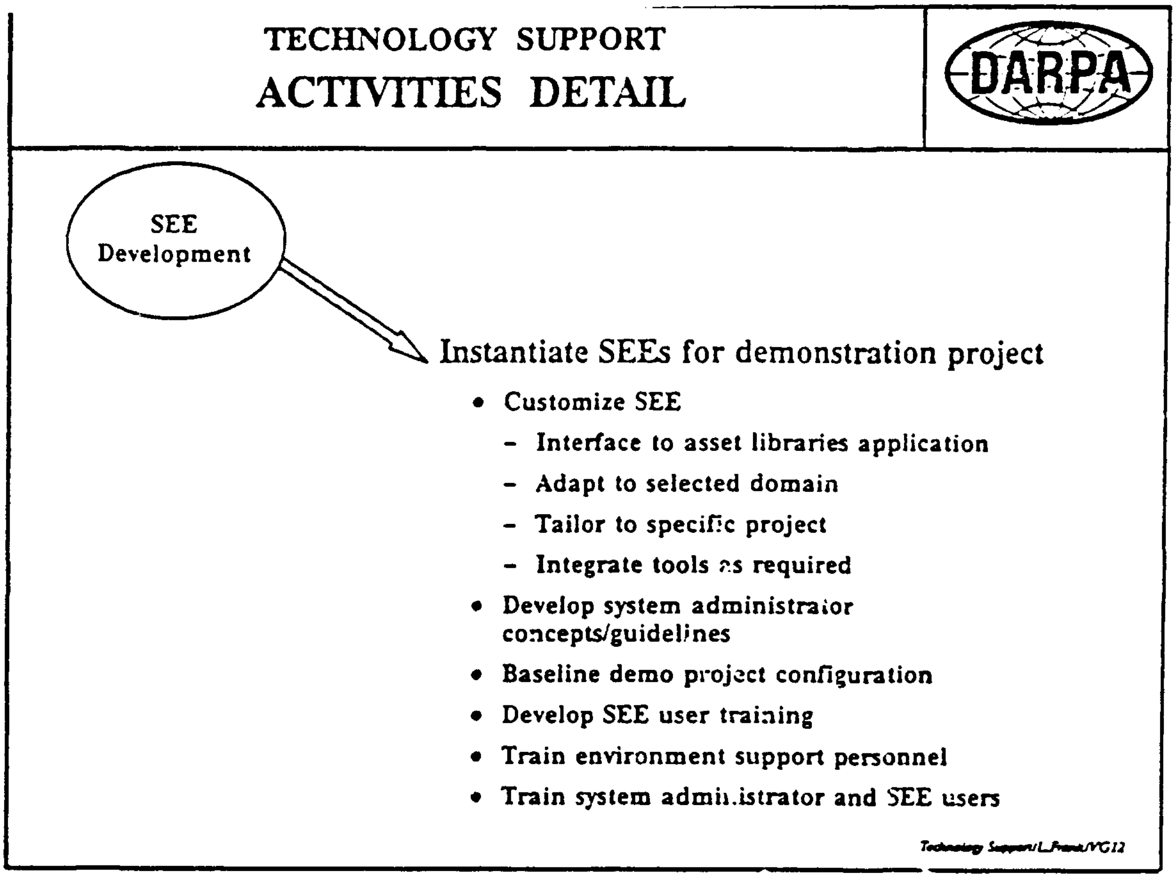
Message: Prototyping/experimentation with the integration of various COTS tools is being carried out to ensure the usability of these environments on the demonstration projects.

In order that the SEEs, that the primes will instantiate for the demonstration projects, be usable on those projects, prototyping efforts are being conducted in the areas of: tool-to-tool interoperability, tool-to-framewrok integration, and framework/SEE administration.

COTS tools are being integrated within the SEEs to improve their eventual usability on the demonstrations. Whatever tools are instantiated for use on these projects will be integrated (to some level) within these environments. The lessons learned with these prototyping efforts will be documented and employed to reduce the integration efforts on behalf of the demonstration projects.

Also, reuse and process technologies developed on the STARS program (and elsewhere) will be integrated, where feasible and where supported by the business case, to improve the usability and effectivity of the SEEs deployed on the projects.

As additional functional capabilities and tools are integrated within the SEE, the supporting information model will be refined and extended. Moreover, as new tools are integrated, the SSP must be augmented to address additional standards necessary to maintain the openness of the architecture and to support tool interoperability as well as the interoperability of the information model, and further, to promote data sharing across the tools that will populate the environment.



TECHNOLOGY SUPPORT

ACTIVITIES DETAIL

DARPA

SEE Development

Instantiate SEEs for demonstration project

- Customize SEE
  - Interface to asset libraries application
  - Adapt to selected domain
  - Tailor to specific project
  - Integrate tools as required
- Develop system administrator concepts/guidelines
- Baseline demo project configuration
- Develop SEE user training
- Train environment support personnel
- Train system administrator and SEE users

Technology Support/L_Frame/VG12

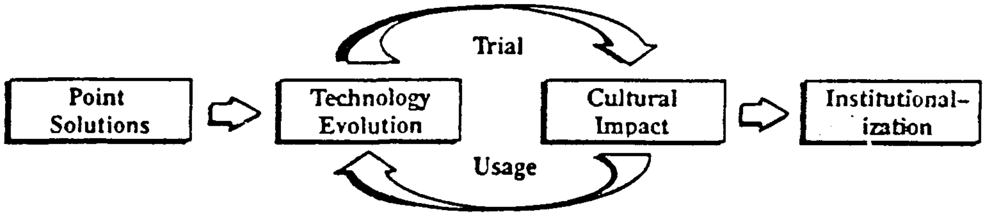
VG12 Title: Detailed Activities

Message: To promote usability of the instantiated SEE and improve its performance, the supporting prime will work closely with project staff to customize, adapt, and tailor the environment to the project's development environment.

As the projects are identified, the supporting prime will work with the projects to identify and integrate any domain-specific, DoD specific, and project unique tools supporting the project's development effort. The SEE instantiated for use on the demonstrations must be customized with respect to the tools integrated for use therein and for usability and performance. The SEE will be baselined for the selected project and project staff will be trained in its administration.

TECHNOLOGY SUPPORT

STARS SEE APPROACH



- Framework-based SEE test-beds
- Prototype information models

- Scalable open architecture
- COTS supplied tools
- Tool portability experiments across multiple frame-work-based SEEs
- Evaluate tool-to-tool interoperability
- Integration of process and reuse mechanisms
- Explore framework-to-framework interoperability

- Guidelines and techniques for extending core services
- Guidelines for adapting/tailoring SEEs to application domains/projects
- Identify and publicize cost benefits of framework-based approach

- Successful demonstration of framework-based approach
- Commercial technology for instantiating framework-based SEEs
- Frameworks as part of commercial platform

Technology Support/L.Frank/VG13

VG13 Title: STARS SEE Approach

Message: The STARS SEE development approach is an iterative one that seeks to evolve existing SEE capabilities based upon prototyping and experimentation. To ease the cultural impacts of adoption and to reduce technical risk, ongoing releases of SEE capabilities will be tested through trial usage on internal projects and by STARS affiliates.

Initial efforts at provisioning framework-based SEES have involved the instantiation of SEE testbeds and generating the supporting information models. These will evolve over time to form the integration platform upon which additional tools will be integrated to build out the SEEs which, in turn, will be the delivery vehicles for supporting the demonstration projects.

To realize the benefits of the framework-based SEE, it must be adopted and used by organizations involved in building systems for both government and industry. To ease the cultural impact of adoption and to reduce the associated technical risks, it is imperative that these cultural barriers be removed and the attendant risks mitigated.

We feel that this can best be accomplished by demonstrating the feasibility and utility of the framework-based approach and, by inference, successful demonstrations of development activities. To this end, the SEEs will be used on real DoD projects as well as internal projects. Additionally, we hope to enlist the aid of affiliates to further test their efficacy.

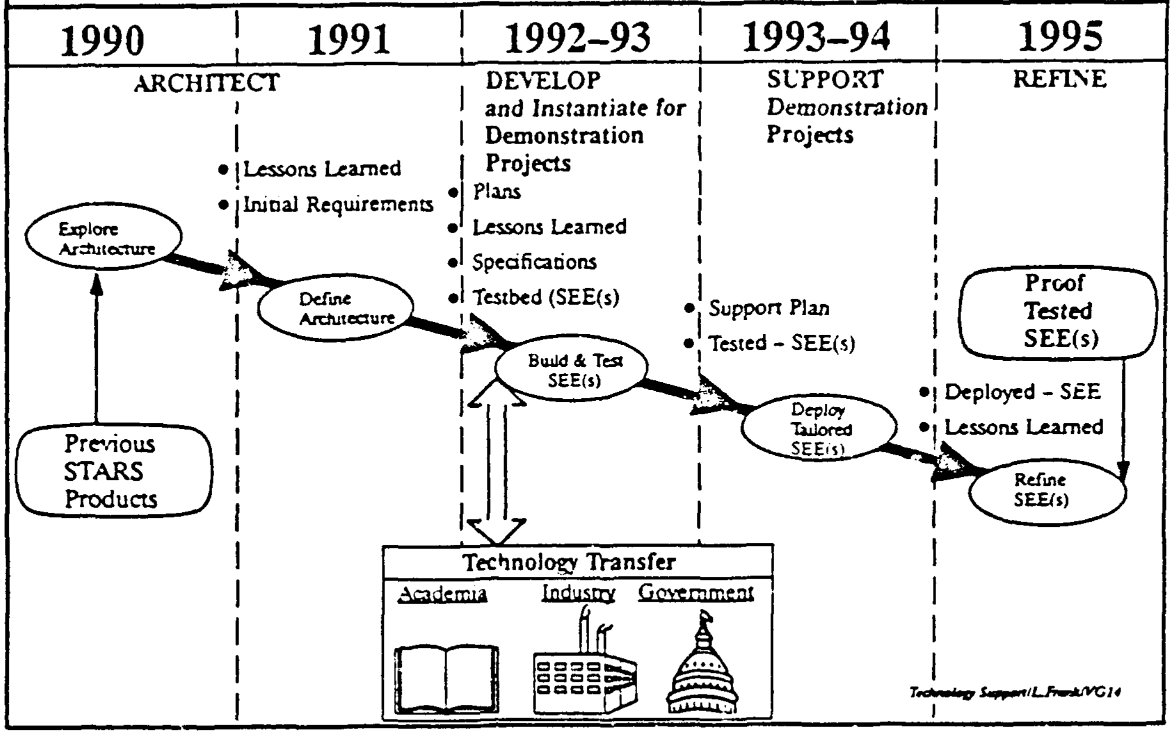
Even as these projects are actively using the SEEs to develop systems, feedback from them, continuing prime activities, and from affiliates will be used to refine and evolve the SEEs themselves. We plan to maintain active involvement of both the framwework providers and the tool vendors to evolve and mature their products to improve SEE utility and performance.

Guidelines for adapting and tailoring SEEs to application domains and projects will be documented along with lessons learned and usage guidelines. The results of actual usage, as well as attendant benefits, will be published and disseminated.

Ultimately, we expect frameworks to be offered by vendors and providers in much the same fashion as graphical user interfaces are currently. That is, they will become part of the commercial platform offerings.

TECHNOLOGY SUPPORT

SEE EVOLUTION



1990 | 1991 | 1992–93 | 1993–94 | 1995

ARCHITECT

- Lessons Learned
- Initial Requirements

DEVELOP and Instantiate for Demonstration Projects

- Plans
- Lessons Learned
- Specifications
- Testbed (SEE(s)

SUPPORT Demonstration Projects

- Support Plan
- Tested - SEE(s)

REFINE

- Deployed - SEE
- Lessons Learned

VG14 Title: SEE Evolution

Message: The development schedule supports the major, high-level activities in the SEE area.

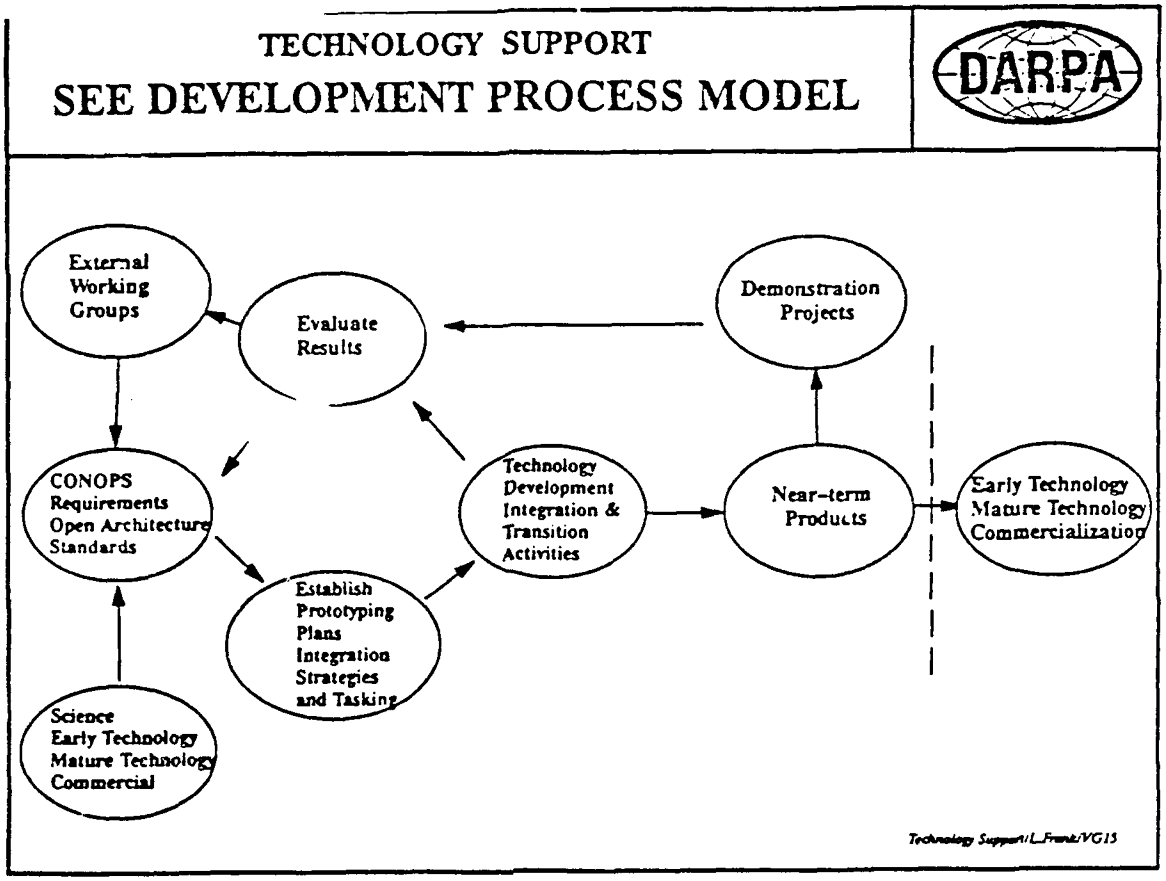
Early STARS foundation activities supported the technology exploration and definition of the SEE architecture. Requirements and criteria have been defined for the SEE infrastructure, the framework. The STARS SSP documents the open standards supporting these framework services.

SEE testbeds are currently being used to prototype and experiment with tooi portability, tool-to-framework integration, and SEE support services. Experience obtained therefrom will be used to further refine and evolve the SEE specifications and to provide feedback to the framework providers and tools builders.

Experience and lessons learned from the aforementioned prototyping activities will be used to instantiate SEEs for the demonstration projects. The target date is October, 1993.

The SEEs instantiated and deployed for use by these demonstration projects will be adapted to the particular application domain and tailored to the demonstration project within that domain. Domain and DoD specific tools will be integrated into these support environments as well as any project unique tooling that is driven by specific project development needs.

Even as the demonstration projects are utilizing the instantiated SEEs, further efforts will be expended on refining and evolving the framework-based SEEs. The projects may elect to take advantage of these refinement and maturation efforts by updating their then current baseline configurations. Results of these continuing efforts will be documented, published, and disseminated as part of the overall STARS evaluation efforts.



TECHNOLOGY SUPPORT

SEE DEVELOPMENT PROCESS MODEL

DARPA

External Working Groups

Evaluate Results

Demonstration Projects

CONOPS Requirements Open Architecture Standards

Technology Development Integration & Transition Activities

Near-term Products

Early Technology Mature Technology Commercialization

Establish Prototyping Plans Integration Strategies and Tasking

Science Early Technology Mature Technology Commercial

Technology Support/L_Frank/VG15

VG15 Title: SEE Development Process Model

Message: The SEE development process is an iterative one designed to take full advantage of evolving needs and maturing technologies as well as efforts by groups outside of STARS.

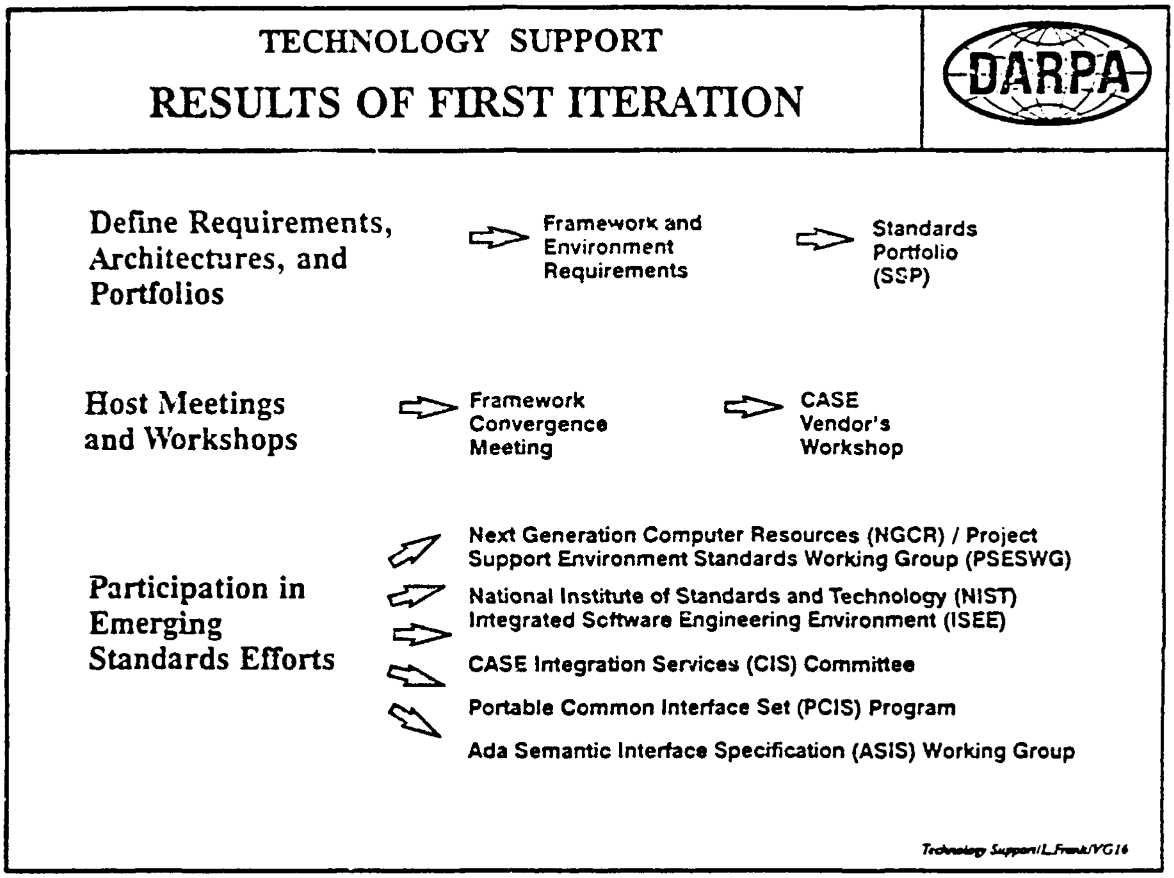
The model presented represents a top level view of the species of activities that are used by the STARS primes in evolving the SEE.

The model provides for technology insertion: basic science; early technology efforts both endogenous and exogenous to DARPA and STARS; and matute technology from industry and the commercial sector. Internal joint activity groups and external working groups also provided input to the definition of the SEE infrastructure in the form of requirements, concepts of operation, and standards analysis efforts.

The SEE Joint Activity Group (SJAG) has undertaken to define and establish prototyping plans, integration strategies, and tasking for evolving the SEE specifications, the testbed SEEs, and integration experiments within those testbeds. These activities, their results, and future plans will be addressed in presentations by each of the primes, entitled "STARS SEE Evolution Strategy." Results from the technology development, integration and transition activities will be evaluated resulting in refinement of the SEE specifications and planning activities.

The demonstration projects will also derive benefits from these development and integration activities. Near-term products as wellas commercial products will be used to instantiate SEEs for the demonstration projects. Actual usage of the SEEs on the projects will provide feedback to the STARS SEE evaluation task, and will be used to refine and evolve the SEEs.

Part of the technology transition effort will be directed towards the commercialization of the near-term STARS SEE products. The continuing evolution of commercial technology and tools are (re) inserted into the process via the technology insertion subtask.



TECHNOLOGY SUPPORT

RESULTS OF FIRST ITERATION

DARPA

Define Requirements, Architectures, and Portfolios → Framework and Environment Requirements → Standards Portfolio (SSP)

Host Meetings and Workshops → Framework Convergence Meeting → CASE Vendor's Workshop

Participation in Emerging Standards Efforts

- Next Generation Computer Resources (NGCR) / Project Support Environment Standards Working Group (PSESWG)
- National Institute of Standards and Technology (NIST) Integrated Scftware Engineering Environment (ISEE)
- CASE Integration Services (CIS) Committee
- Portable Common Interface Set (PCIS) Program
- Ada Semantic Interface Specification (ASIS) Working Group

Technology Support/L_Frank/VG16

VG16 Title: Results of First Iteration

Message: STARS has produced infrastructure documents supporting the SEE development efforts and begun the transitioning activities. STARS also actively supports similar efforts with external groups and organizations to aid in the two way transitioning of technology and infrastructure information.

STARS has been influenced by, and in turn, has influenced others in the SEE/framework area. These efforts will be addressed in detail by Jin. Hamilton and Bob Ekman in their respective presentations, "STARS Standards Portfolio (SSP)" and "STARS Role in Standards Maturation."

TECHNOLOGY SUPPORT

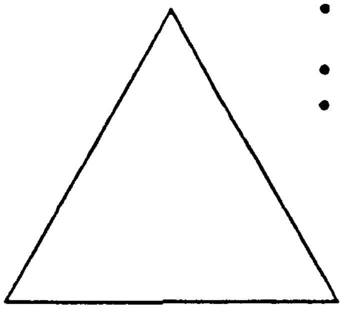
STARS ROLES AND RELATIONSHIPS

DARPA

STARS Primes (Boeing, IBM, Unisys)

- Test Beds/Open Architecture
- DoD-specific Adaptations

- Joint Technical Development: Process and Reuse
- Individual Technical Activities
- Technology transition–demo of evolving capabilities



Vendor Community

- Suppliers of S/W engineering tools populating SEE

Primes Commercial Counterparts (DEC, IBM, Unisys)

- Suppliers of tools and environments
- Supporters of STARS technical direction

Note: Each instantiated SEE consists of:
- Existing commercial capabilities
- Third party vendor capabilities
- Prime-specific technical extensions/adaptations

Technology Support/L.Frank/VG17

VG17 Title: STARS Roles and Relationships

Message: STARS and the primes have sought the active participation of the primes' commercial counterparts and the greater vendor community. We think that this participation is crucial to instantiating and deploying SEEs on the demonstration projects as well as the continuing maintenance and support of these SEEs.

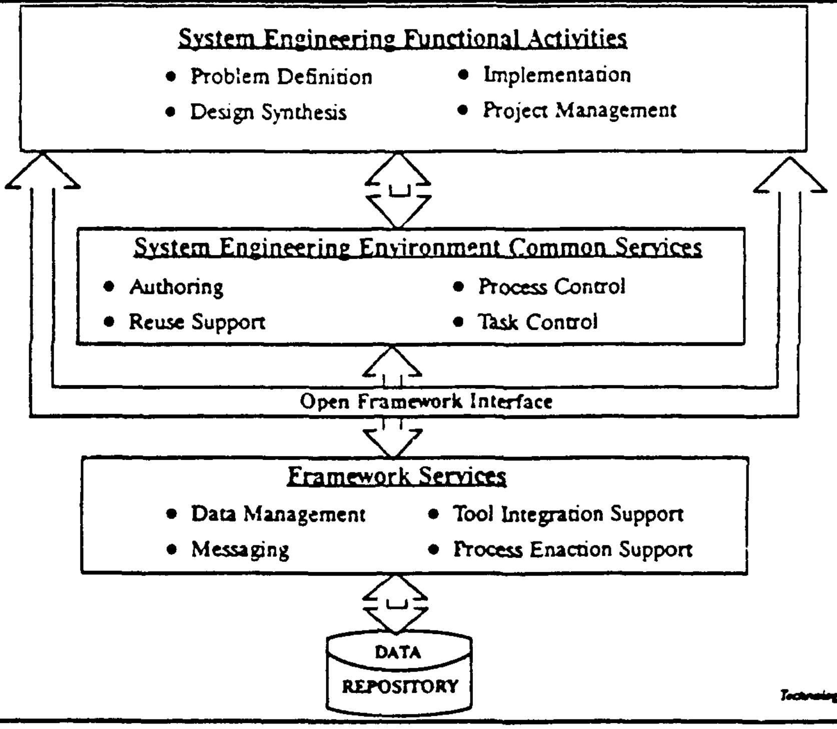
STARS continues to seek the active participation of the primes' commercial counterparts and the vendor community in the program. In July 91, the technical direction of the STARS program and the STARS Standards Portfolio was presented at the CASE Vendors Workshop (CVWS). Participants at the workshop were provided a forum in which to provide feedback to the program. Their reactions have been noted and incorporated into infrastructure documents and planning efforts.

The commercial counterparts have provided ongoing guidance (sanity checks) on environment activities and have indicated support of the technical directions of the STARS program. They underscored this support at the CVWS and have provided active support to the SJAG.

The primes continue to maintain responsibility for the joint and prime-specific technical activities. They, together with the STARS program management team have responsibility for technology transition to the commercial sector and other groups, agencies, and organizations outside of STARS. STARS has also defined an affiliates program to identify and actively involve technology receptors to assist the program in further transitioning activities.

TECHNOLOGY SUPPORT

SEE CONCEPTUAL ARCHITECTURE



System Engineering Functional Activities

- Problem Definition
- Design Synthesis
- Implementation
- Project Management

System Engineering Environment Common Services

- Authoring
- Reuse Support
- Process Control
- Task Control

Open Framework Interface

Framework Services

- Data Management
- Messaging
- Tool Integration Support
- Process Enaction Support

DATA REPOSITORY

Technology Support/L_Frank/VG18

VG18 Title: SEE Conceptual Architecture

Message: The SEE conceptual architecture provides the context for understanding the various functional capabilities required for supporting the development environment.

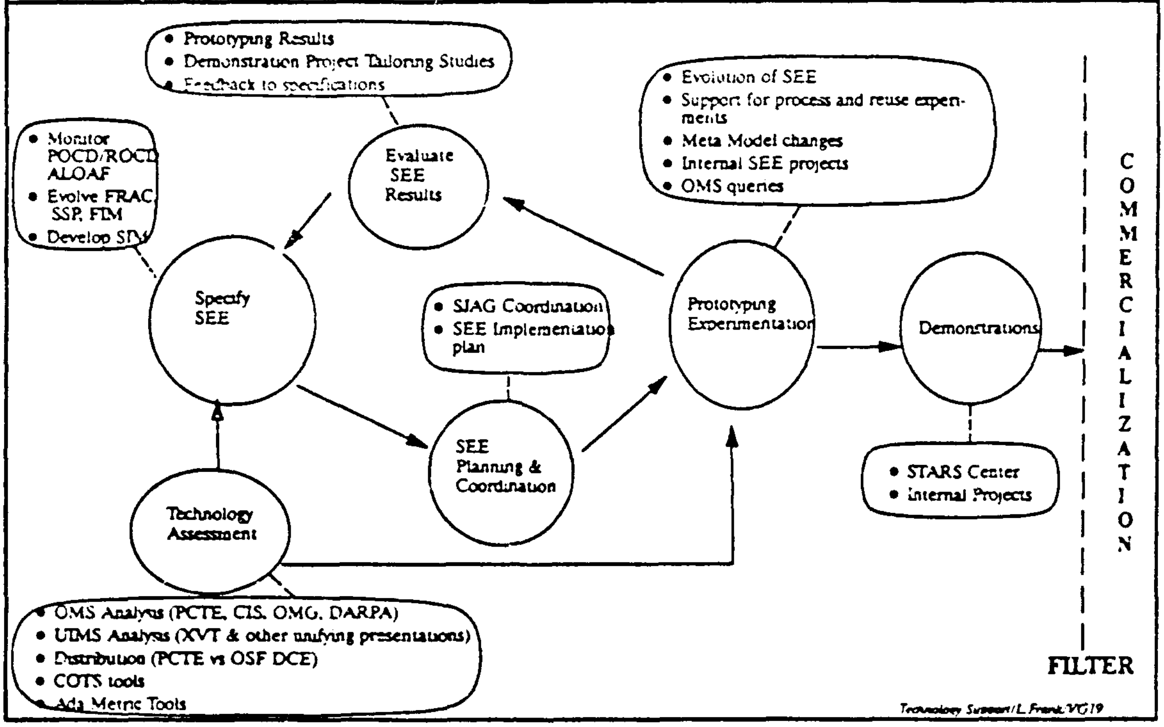
A given instance of a SEE does NOT, can NOT, and will NOT support any development project irrespective of the application domain and specific project and Dod unique needs. A context is needed for rationalizing the support requirements for a development project. The SEE conceptual architecture provides this context.

The model presented does not represent any manifestation of a physical architecture, nor is it complete with respect to functional capabilities needed to support the demonstration projects. It should be read notionally. It is presented here only to emphasize that the SEE is something more than a framework, and that there are other common services needed in the environment supporting software engineering activities in addition to those offered by the framework.

Peter Feiler in his presentaion, "Project Support Environment Services Reference Model", will further discuss additional environment services requisite to supporting software development.

TECHNOLOGY SUPPORT

FURTHER PROCESS ITERATIONS



- Prototyping Results
- Demonstration Project Tailoring Studies
- Feedback to specifications

- Monitor POCD/ROCD ALOAF
- Evolve FRAC, SSP, FIM
- Develop SDM

Evaluate SEE Results

Specify SEE

- SJAG Coordination
- SEE Implementation plan

SEE Planning & Coordination

Technology Assessment

- Evolution of SEE
- Support for process and reuse experiments
- Meta Model changes
- Internal SEE projects
- OMS queries

Prototyping Experimentation

Demonstrations

- STARS Center
- Internal Projects

COMMERCIALIZATION

FILTER

- OMS Analysis (PCTE, CIS, OMG, DARPA)
- UIMS Analysis (XVT & other unifying presentations)
- Distribution (PCTE vs OSF DCE)
- COTS tools
- Ada Metric Tools

Technology Support/L Frank VG19

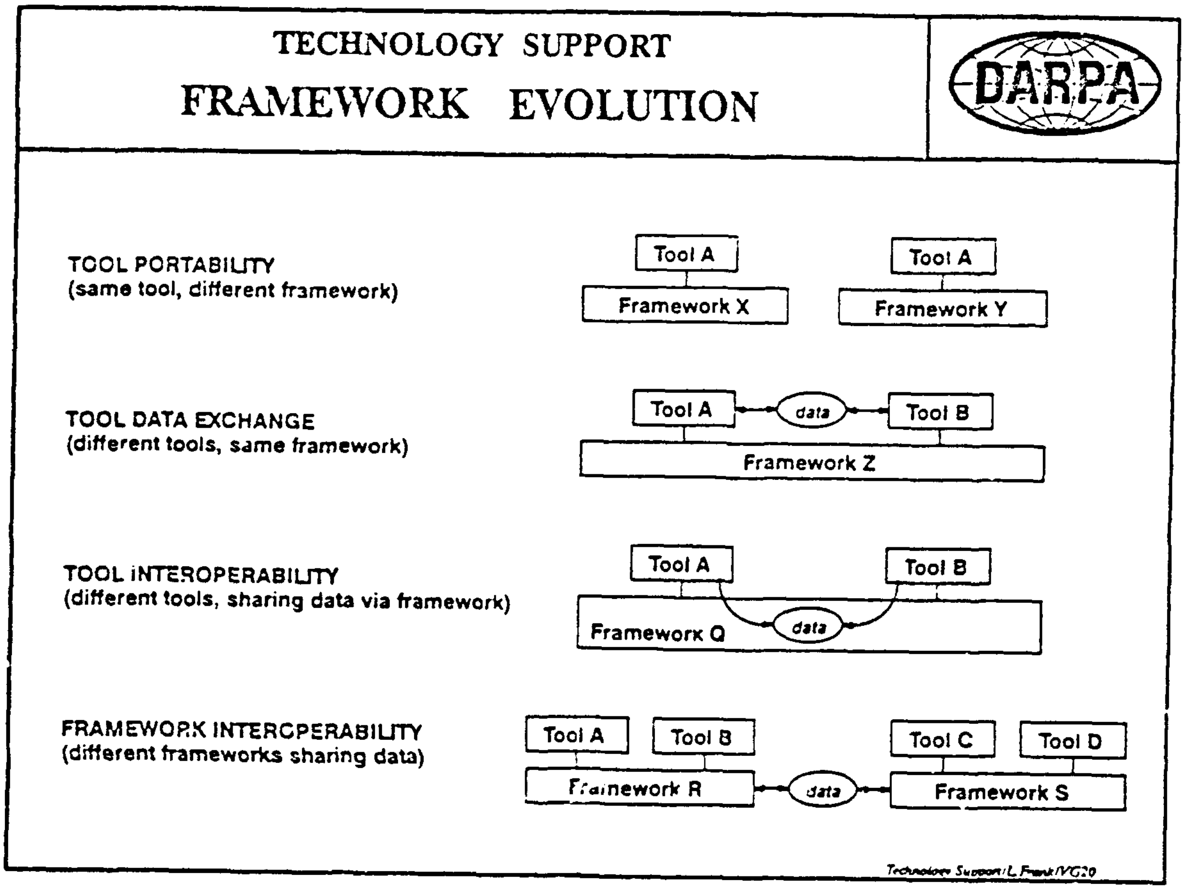
VG19 Title: Further Process Iterations

Message: Further iterations of the SEE development process will focus on evolving the SEE through prototyping efforts and through usage of the testbeds in supporting internal projects.

SEE testbeds ·vill be deployed on interanl projects ·within the primes' organizations and the results used to evolve both the SEE :pecifications and the capabilities supported in the testbeds. Reus and process mechanisms will be integrated (presentation, control, and data levels) within the SEE based on need, usage, and expected returns. Experiments will be conducted on accessing geographically dispersed reuse libraries, and on the exchange of reuse assets across these libraries.

Information modeling supporting deeper levels of integration and interoperability will be a prime concern. Experiments will be conducted in extending the information model supporting framework services. The commercial partners have indicated interest in pursuing these efforts and propose to actively become involved in such.

Testbeds will also be deployed at each prime location and the STARS Center. The latter will be used to demonstrate the evolving SEE capabilities to STARS affiliates, vendors, and other interested parties. These demonstrations are expected to commence inte first quarter, 1992.



TECHNOLOGY SUPPORT

FRAMEWORK EVOLUTION

DARPA

TOOL PORTABILITY
(same tool, different framework)

Tool A
Framework X
Tool A
Framework Y

TOOL DATA EXCHANGE
(different tools, same framework)

Tool A
data
Tool B
Framework Z

TOOL INTEROPERABILITY
(different tools, sharing data via framework)

Tool A
Tool B
Framework Q
data

FRAMEWORK INTEROPERABILITY
(different frameworks sharing data)

Tool A
Tool B
Tool C
Tool D
Framework R
data
Framework S

Technology Support/L.Frank/VG20

VG20 Title: Framework Evolution

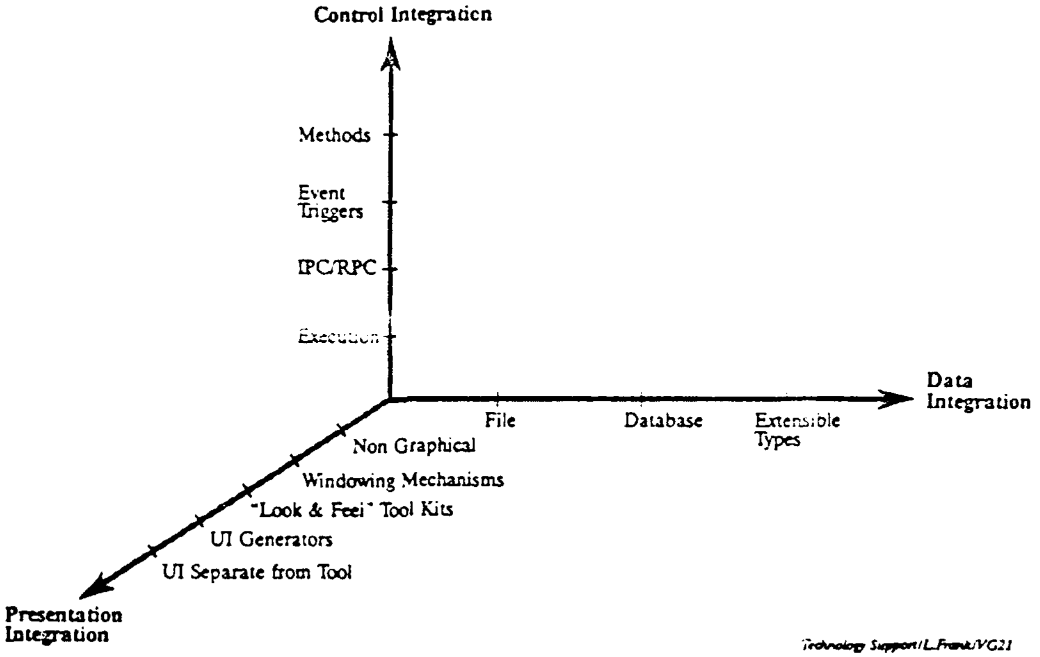
Message: Frameworks will continue to evolve. STARS will maintain an active role in furthering this evolution.

Tool portability is a chief concern of the STARS program. Some of the proposed prototyping efforts will address portability issues. Current efforts are centered on tool integration and interoperability. Each prime is focusing on issues of presentation, control, and data integration, and on the guidelines for integrating tools within their respective environments.

Further details will be addressed in the primes' presentation on "SFE Evolution Strategy."

TECHNOLOGY SUPPORT

INTEGRATION DIMENSIONS



Technology Support/L.Frank/VG21

VG21 Title: Integration Dimensions

Message: Tool integration is a focus of prototyping activities by each prime. The granularity (coarse versus fine) of integration one wishes to achieve across the dimensions of presentation, control, and data has implications for the types of services required.

The following are offered as concise definitions of presentation, control, and data integration together with what the potential implications of granularity might mean:

Presentation integration: the ability within the environment to provide a consistent "look and feel" across the tools within the environment. The effect is both visual as well as behavioural. At a coarse level, this is constrained by the man-machine interface implemented within the tool or tool suite. Finer levels of granularity necessitates the parametrization of the MMI across all functional interfaces within the tool architecture.

Control integration: the ability within the environment to control initiation, serialization, and synchronization of process or task execution. It also includes the notion of being able to establish pre-ambles and post-ambles for subprocesses or subtasks and to invoke these or other subprocesses/subtasks based on the evaluation of the pre-amble/post-amble. At the coarsest level, would be the ability to invoke a given process/task, a binary executable for example. At the finest level, it would imply the ability to embed methods, triggers, and control points within a process network and to subsequently control execution of subprocesses/subtasks within that network based on evaluation of the control structures (pre-ambles and post-ambles, among others).

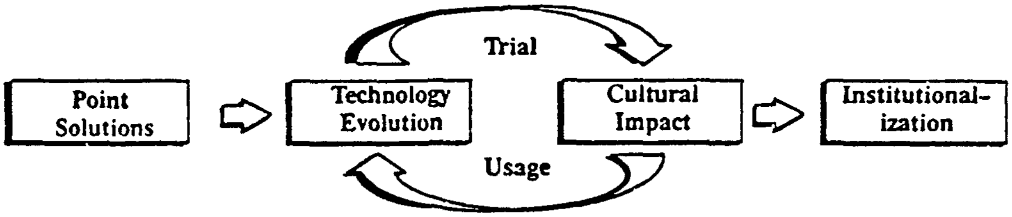
Data integration: the ability within an environment to share data referenced via a common representation across tools, services, and functional components. At the coarsest level, this would include the ability to reference files where the semantics of the underlying data representation lies within the tool itself. At the finest level, it would include the ability to reference finer grained objects within a defined class/type hierarchy where the semantics of the data within the hierarchy is publicly available and referenced by a common meta-model. (continued)

VG21 Title: Integration Dimensions (continued)

The primes will address their strategy for integration of tools within the SEE in their respective presentations on. "STARS SEE Evolution Strategy." Peter Feiler in the presentation, "Project Environment Services Reference Model" will argue for a fourth dimension, process integration.

TECHNOLOGY SUPPORT

ACHIEVEMENTS CONTEXT

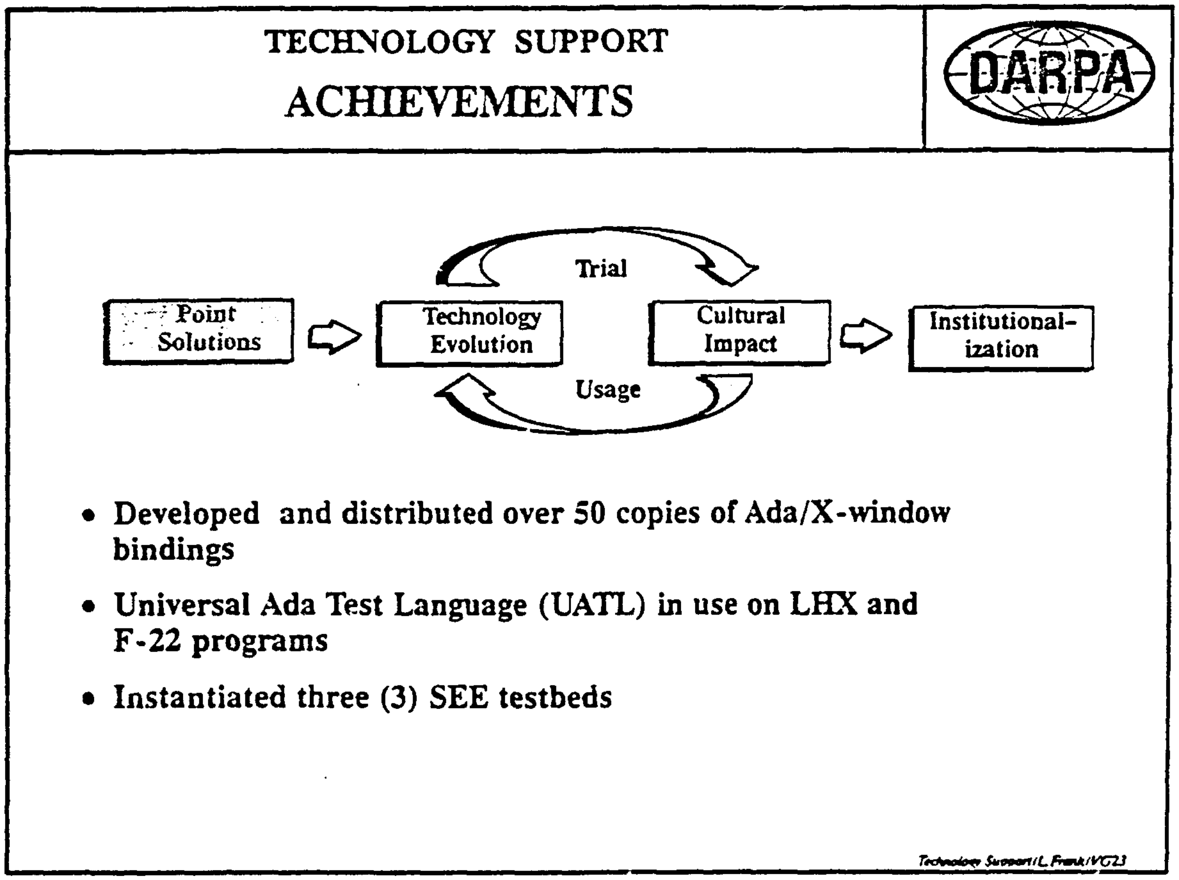


Technology Support/L Frank/VG22

VG22 Title: Achievements Context

Message: STARS has various achievements to its credit. These will be contexted in terms of the characterization of the STARS approach.

The achievements of the STARS program in the SEE area are contexted in terms of the STARS SEE Approach. They will be charecterized in terms of achievements that contribute to or are manifested as: Point Solutions, Technology Evolution, Cultural Impact, and Institutionalization.



TECHNOLOGY SUPPORT

ACHIEVEMENTS

DARPA

Point Solutions → Technology Evolution ⇄ Cultural Impact → Institutional-ization

Trial

Usage

- Developed and distributed over 50 copies of Ada/X-window bindings
- Universal Ada Test Language (UATL) in use on LHX and F-22 programs
- Instantiated three (3) SEE testbeds

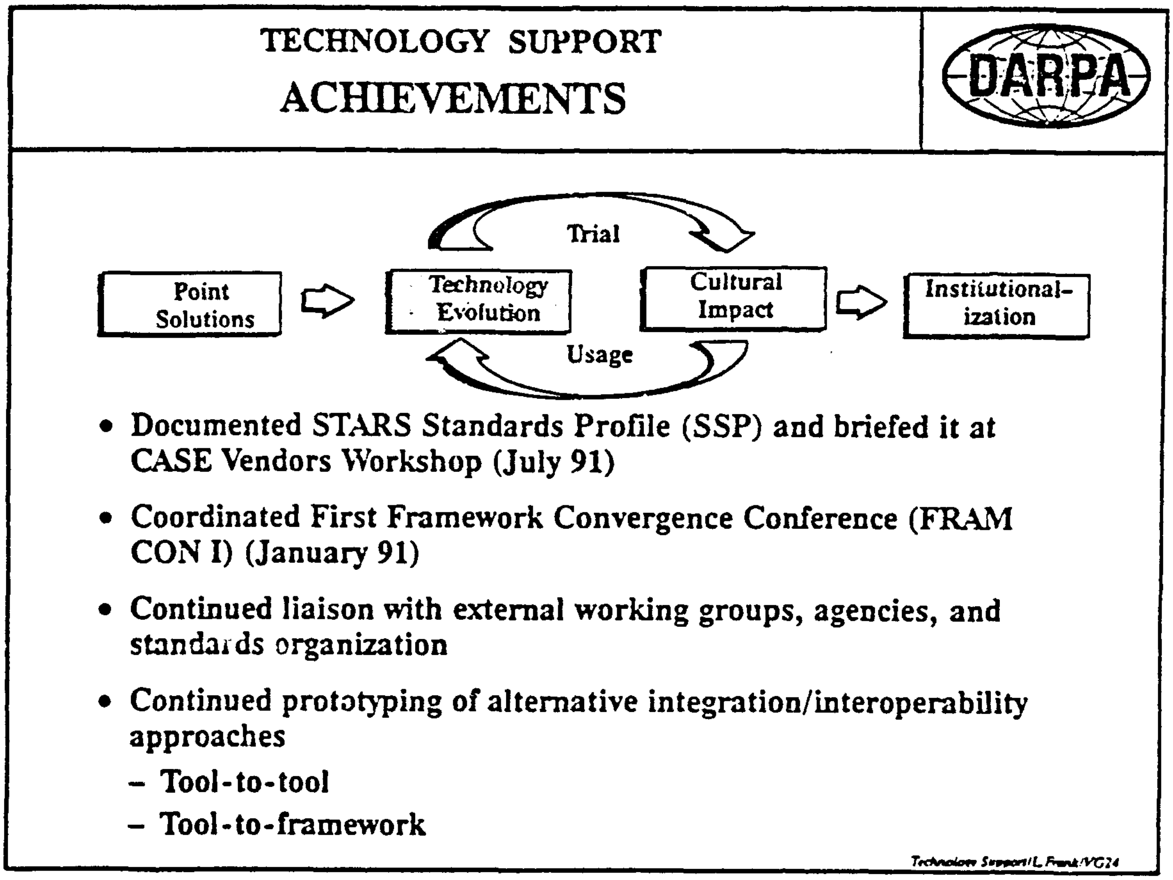
Technology Support/L Frank/VG23

VG23 Title: Achievements—Point Solutions

Over 50 copies of the Ada/X-window bindings have been delivered to various organizations. This effort has been picked up by commercial companies and are a basis for their product offerings.

The Universal Ada Test Language (UATL) is in use on the LHX and F-22 programs. It promises to reduce the development costs of testing by 30%.

Each of the three primes (Boeing, IBM, and Unisys) has instantiated SEE testbeds. Details will be presented by the primes' presentations on "SEE Evolution Strategy."



TECHNOLOGY SUPPORT

ACHIEVEMENTS

DARPA

Trial

Point Solutions

Technology Evolution

Cultural Impact

Institutional-ization

Usage

- Documented STARS Standards Profile (SSP) and briefed it at CASE Vendors Workshop (July 91)
- Coordinated First Framework Convergence Conference (FRAM CON I) (January 91)
- Continued liaison with external working groups, agencies, and standards organization
- Continued prototyping of alternative integration/interoperability approaches
  - Tool-to-tool
  - Tool-to-framework

Technology Support/L Frank/VG24

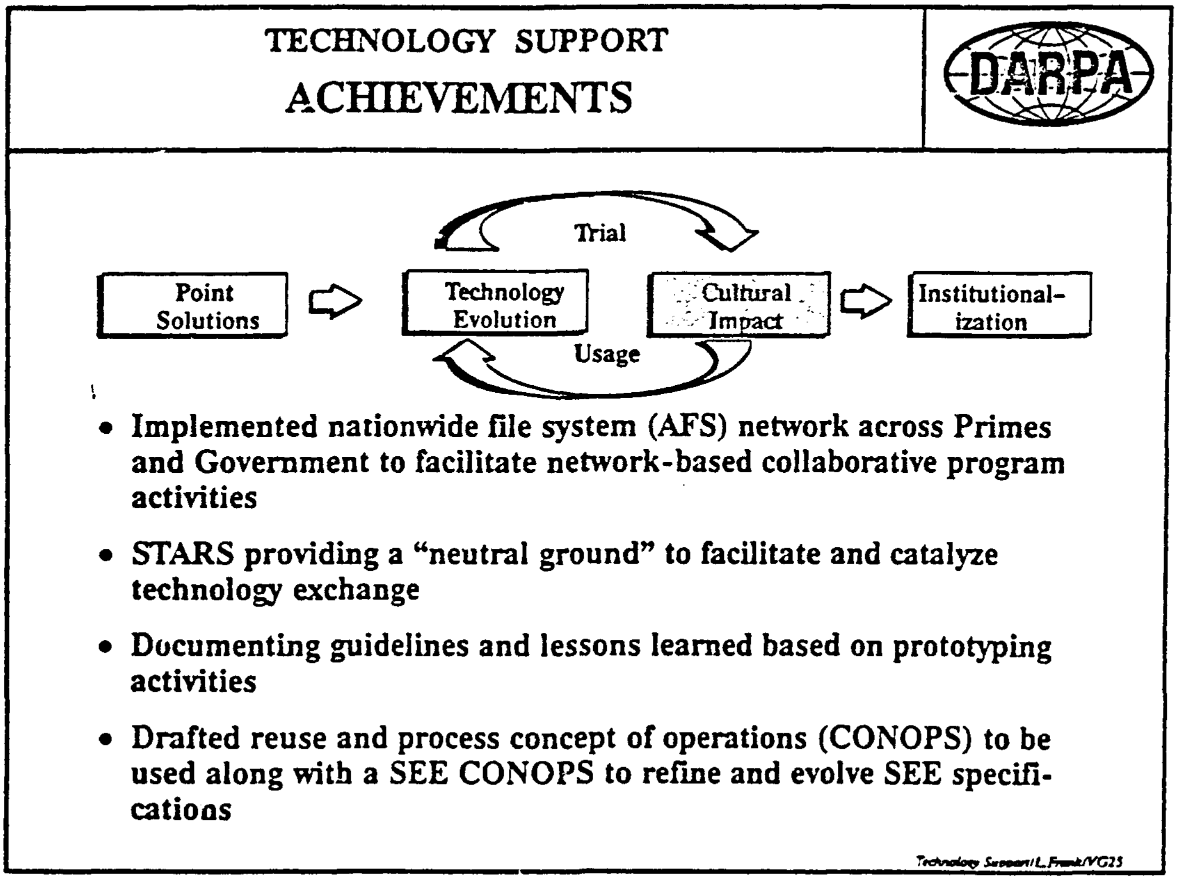
VG24 Title: Achievements—Technology Evolution

The STARS Standards Prortfolio (SSP) was briefed to the vendor community at the CASE Vendors Workshop, July 1991, for their reaction and feedback. Its reception was, generally, favorable.

STARS coordinated the first framework convergence conference (FRAMCON I) which was sponsored by NIST, January 1991. Various issues were identified and explored. Similarities and differences between ATIS and PCTE were noted and discussed. There seems to be sufficient interest in convening a second framework convergence conference to examine issues in greater detail.

Bob Ekman will address the STARS continuing efforts with external groups, agencies, and standards organizations.

As mentioned before, the STARS primes continue to prototype alternative integration and interoperability approaches with respect to tool-to-tool interoperability and tool-to-framework integration. Details will be addressed in the primes' "SEE Evolution Strategy."



TECHNOLOGY SUPPORT

ACHIEVEMENTS

DARPA

Trial

Point Solutions

Technology Evolution

Cultural Impact

Institutionalization

Usage

- Implemented nationwide file system (AFS) network across Primes and Government to facilitate network-based collaborative program activities
- STARS providing a "neutral ground" to facilitate and catalyze technology exchange
- Documenting guidelines and lessons learned based on prototyping activities
- Drafted reuse and process concept of operations (CONOPS) to be used along with a SEE CONOPS to refine and evolve SEE specifications

Technology Support/L.Frank/VG25

VG25 Title: Achievements—Cultural Impact

Within the STARS program, a nationwide file system (AFS) has been implemented to support the collaborative program activities. Several of the STARS documents have been distributed via this network.

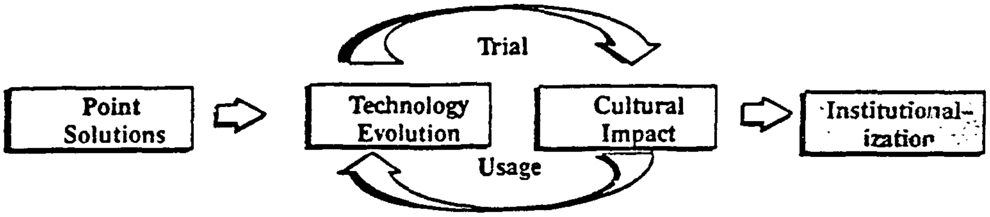
STARS will continue to provide a "neutral ground" to facilitate and catalyze technology exchange. FRAMCON I is an example of this activity. We are currently exploring other topics that might lend themselves to this method of technology exchange.

STARS primes are currently documenting their activities, guidelines, and lessons learned with respect to their prototyping and experimental activities.

The Reuse Joint Activity Group (RJAG) and Process Joint Activity Group (PJAG) have drafted concepts of operations (CONOPS) which will be used as input, along with the SEE concept of operations, to refine and evolve the SEE specifications.

TECHNOLOGY SUPPORT

ACHIEVEMENTS



- **Facilitated STARS Primes' commercial counterparts (DEC, IBM, Unisys) technical direction and support agreement**
- **Sponsored CASE Vendor Workshop and will maintain liaison through affiliates program**

Technology Support/L.Frank/VG26

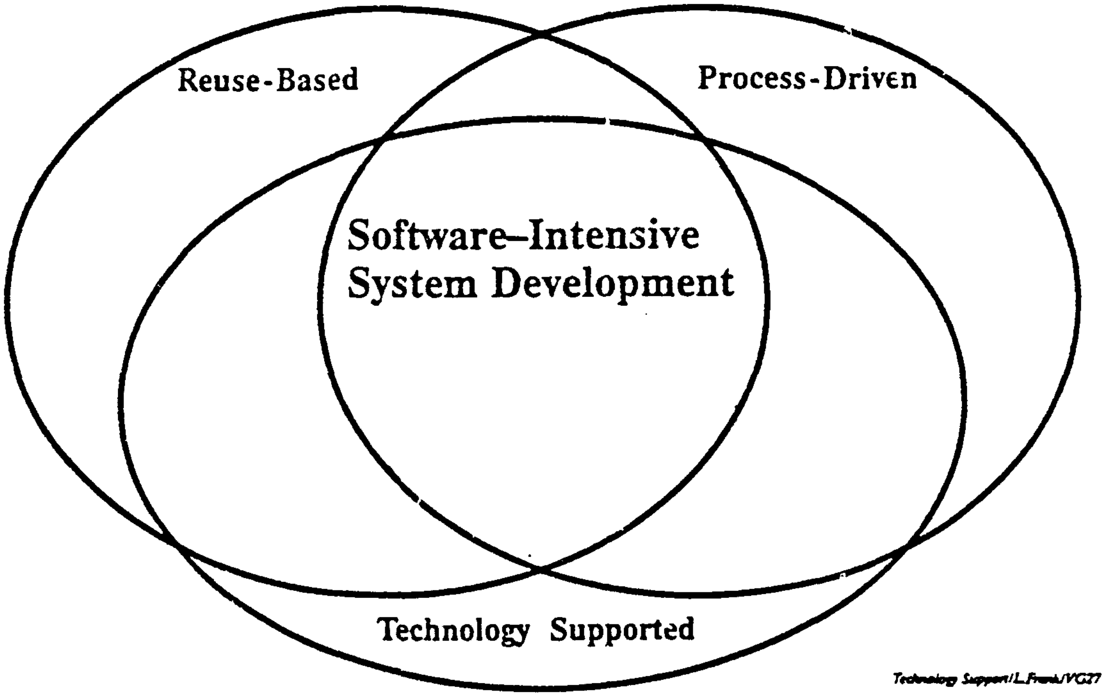
VG26 Title: Achievements—Institutionalization

STARS program management, primes' program managers, and the STARS SEE architect met with the primes' commercial counterparts (DEC, Nashua; IBM, Toronto; and Unisys, Roseville) to discuss the technical direction of the STARS program. The commercial counterparts supported the technical directions and have provided support to the program. They also participated in the first framework convergence conference, and have indicated interest in a proposed second conference.

STARS sponsored the CASE Vendors Workshop (CVWS), July 1991, to present to the vendor community the program's technical direction and to brief the SSP. Case vendors have been invited to STARS 91 to maintain currency with the STARS program. Continued liaison with the vendor community will be effected under the auspices of the affiliates program.

TECHNOLOGY SUPPORT

SUMMARY



Reuse-Based

Process-Driven

Software–Intensive System Development

Technology Supported

Technology Support/L.Frank/VG27

VG27 Title: Summary

Message: Productivity increases are achievable through the application of megaprogramming concepts, supporting processes, and reuse and SEE technologies.

In order to accelerate the shift to megaprogramming and to realize its potential benefits as applied to large–scale, complex, software–intensive system development projects, it will require thesynergistic effects represented by the confluence of technology thrusts in the reuse, process, and SEE areas.

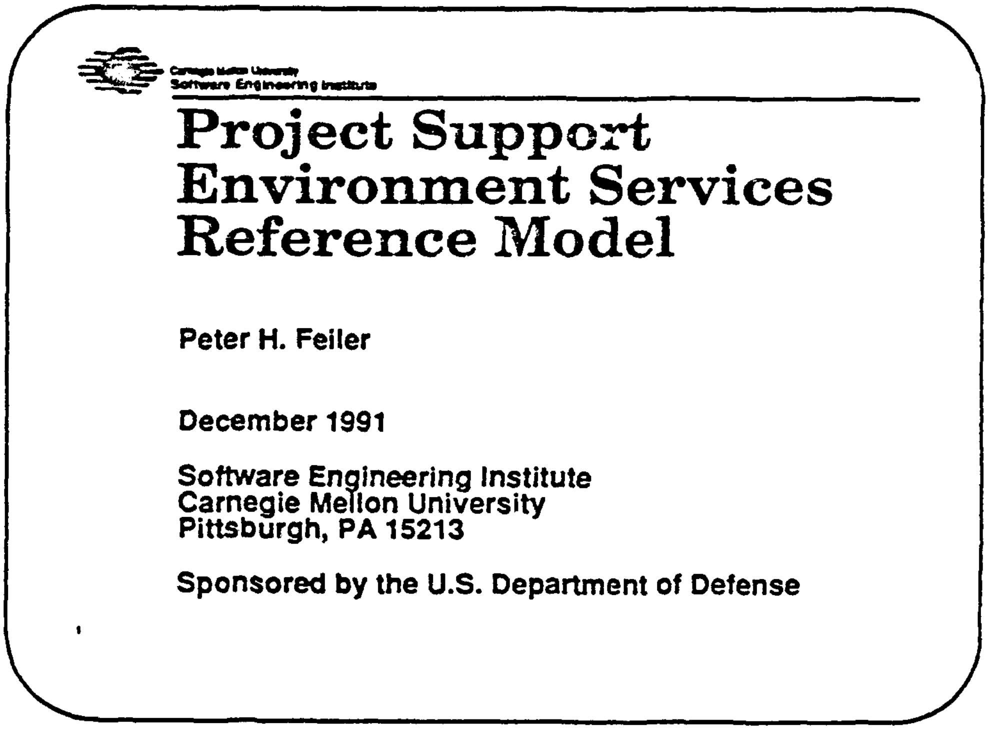
Megaprogramming will happen with or without the STARS efforts. The STARS role is to accelerate its pace. The technology underlying and supporting this envisioned paradigm is central to proving the feasibility of the approach. However, the crucial element in accelerating its pace is removal of the cultural barriers that impede its acceptance in both the government and the industrial organizations that will be impacted by this new way of doing business.

STARS can affect the rate of acceptance of the megaprogramming approach by demonstrating its effiency and efficancy on real DoD programs, and by working with both the government and industry sectors in improving the partnership needed to fully exploit its benefits. At the least, this will require the incentivization of industry to: define, evolve, and reuse domain–specific architectures; refine and evolve the processes supporting the megaprogramming approach; and incorporate new/emerging technologies in the supporting software engineering environments.

Acronym List

ABET	Ada Based Environment for Testing
AFS	A TRANSARC Product (Andrew File System)
ALOAF	Asset Library Open Architecture Framework
APP	Application Portability Profile
ASIS	Ada Semantic Interface Specification
ATIS	A Tool Integration Standard/Atherton Tool Integration Standard
CASE	Computer Aided Software Engineering
CIS	CASE Integration Services
COTS	Commercial Off-The-Shelf
DARPA	Defense Advanced Research Projects Agency
DCE	Distributed Computing Environment
DoD	Department of Defense
ECMA	European Computer Manufacturer's Association
FIM	Framework Information Model
FJAG	Framework Joint Activity Group
FRAC	Framework Requirements and Criteria
IM	Information Model
IPC	Inter-Process Communication
IRDS	Information Resource Dictionary System
ISEE	Integrated Software Engineering Environment
NGCR	Next Generation Computer Resources
NIST	National Institute of Standards and Technology
OAF	Open Architecture Framework
OMG	Object Management Group (continued)

OMS	Object Management System
OSF	Open Software Foundation
P1175	A Standard Reference Model for Computing System Tool Interconnection
PCIS	Portable Common Interface Set
PCTE	Portable Common Tool Environment
POCD	Process Operational Concept Document
POSIX	Portable Operating System Interface
PSESWG	Project Support Environment Standards Working Group
ROCD	Reuse Operational Concept Document
RPC	Remote Procedure Call
SEE	Software Engineering Environment
SEMATECH	A Consortium
SIM	SEE Information Model
SJAG	SEE Joint Activity Group
SOCD	SEE Operational Concept Document
SSP	STARS Standards Profile
UATL	Universal Ada Test Language
UI	User Interface
UIMS	User Interface Management System
XVT	Extensible Virtual Toolkit



Carnegie Mellon University
Software Engineering Institute

Project Support Environment Services Reference Model

Peter H. Feiler

December 1991

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Sponsored by the U.S. Department of Defense

NOTES

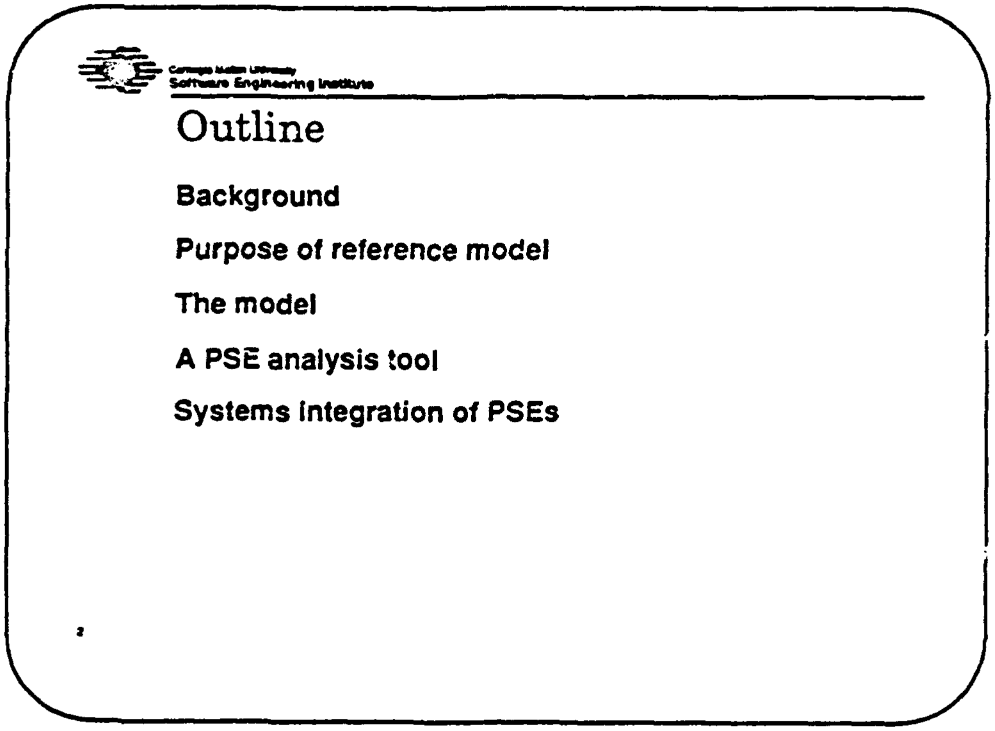
Peter H. Feiler
phf@sei.cmu.edu
(412) 268-7790

Acting program manager of Software Engineering Techniques program

Project leader of Software Development Environments project

Alan Brown Kurt Wallnau
Susan Dart Alan Christie
Howard Slomer (resident affiliate)

CASE Technology project
Dennis Smith project leader
Paul Zarrella, Ed Morris, Cliff Huff



Carnegie Mellon University
Software Engineering Institute

Outline

Background

Purpose of reference model

The model

A PSE analysis tool

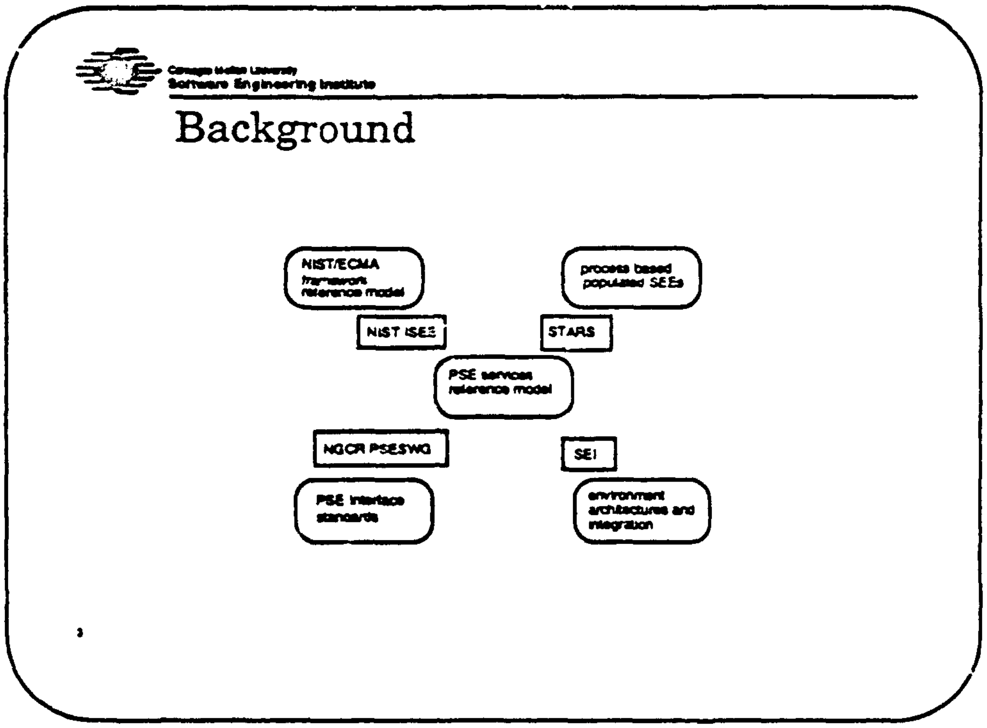
Systems Integration of PSEs

2

NOTES

For more information about the SEI's work on Software Engineering Environments see the information table or several present representatives

PSE	Project Support Environment
SEI	Software Engineering Institute



Carnegie Mellon University
Software Engineering Institute

Background

NIST/ECMA framework reference model

process based populated SEEs

NIST ISEE

STARS

PSE services reference model

NGCR PSESWG

SEI

PSE interface standards

environment architectures and integration

NOTES

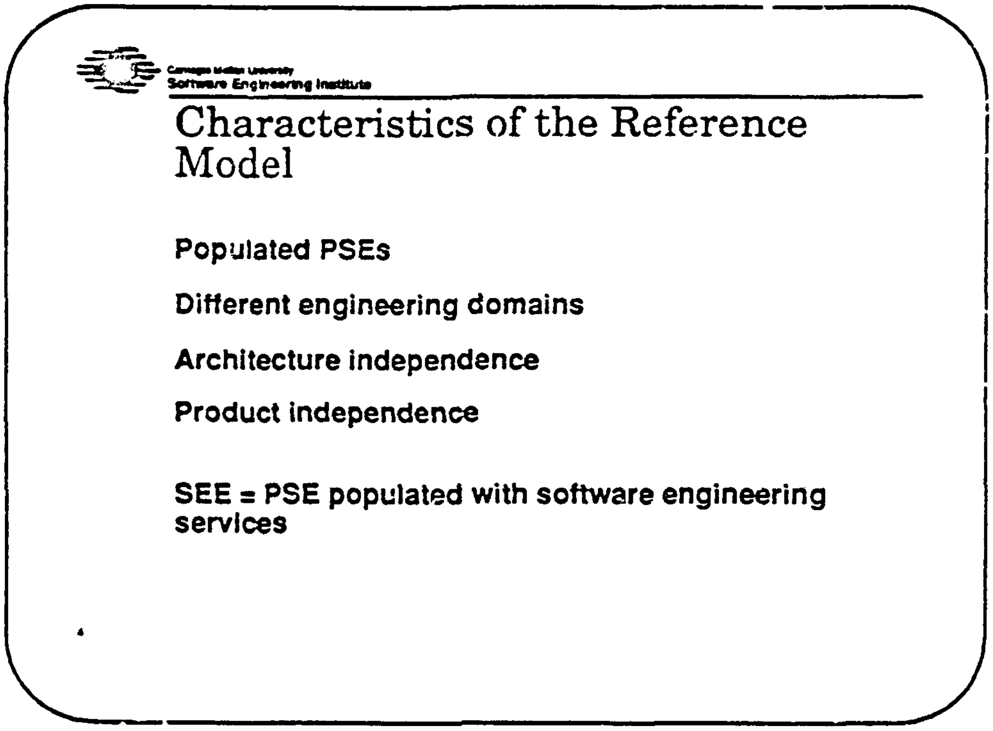
- Conceptual work on PSE services reference model at SEI in the context of environment work on architectures and integration approaches. Concept paper available as SEI technical report.

- SEI provides technical lead for the NGCR PSESWG reference model subgroup. This subgroup will produce a PSE service reference model report. NGCR PSESWG contact is: Tricia Oberndorf: (215/441-2737) tricia @NADC.NADC.NAVY.mil. See NGCR PSESWG flyer at information table.

- SEI actively contributes to NIST ISEE in both the process management subgroup and tool integration subgroup. NIST ISEE currently leads the international effort of refining the NIST/ECMA framework reference model. NIST ISEE contact is Bill Wong: (301/975-3341) wong@swe.ncsl.nist.gov

- SEI contributes to the coordination of these efforts as member of their executive committees.

NIST	National Institute of Standards and Technology
ISEE	Integrated Software Engineering Environments
ECMA	European Computer Manufacturing Association
NGCR	Next Generation Computer Resources effort by Navy
PSESWG	Project Support Environment Standards workshop Group



Carnegie Mellon University
Software Engineering Institute

Characteristics of the Reference Model

Populated PSEs

Different engineering domains

Architecture independence

Product independence

SEE = PSE populated with software engineering services

NOTES

The reference model covers a complete computer-based environment, i.e., an environment framework populated with tools.

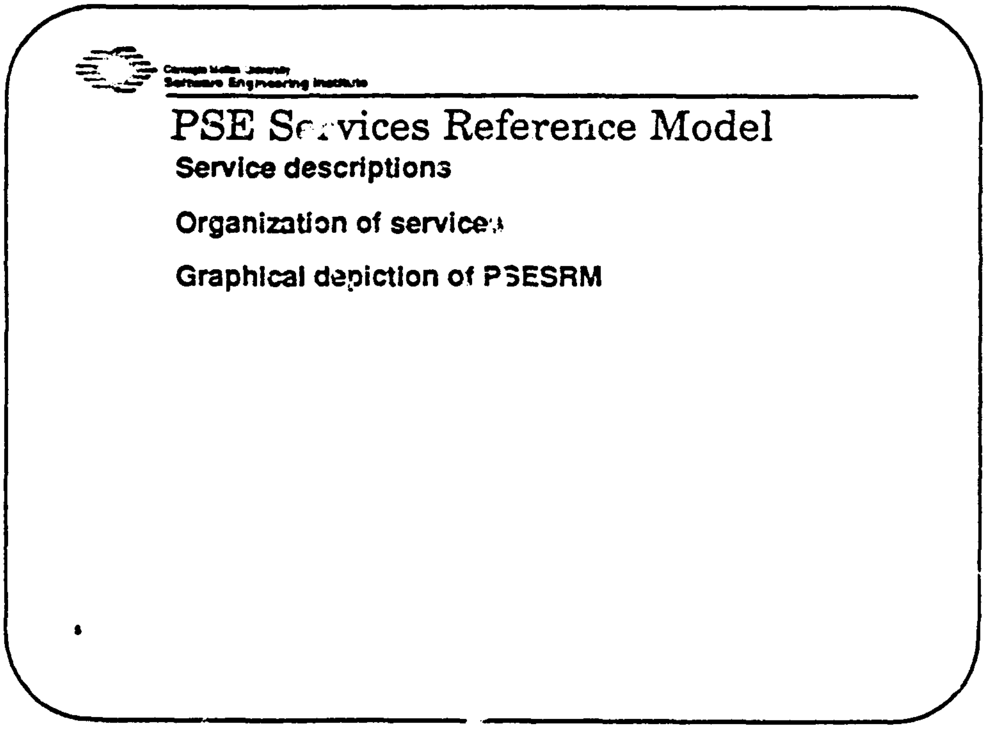
The model accommodates the domain of Software Engineering as well as other domains.

The term project support refers to the fact that the model accommodates both engineering activities and management activities.

The model does not impose a particular architecture--in fact one of its purposes is to characterize different architectures.

The term service indicates that the model does not reflect particular tool products, but is used to characterize them. One tool may provide a number of services.

SEE	Software Engineering Environment
PSE	Project Support Environment



Carnegie Mellon University
Software Engineering Institute

PSE Services Reference Model

Service descriptions

Organization of services

Graphical depiction of PSESRM

NOTES

The PSESRM is not a picture.

It consists of

- a set of service descriptions specified along several dimensions
- an organization of these services into a structure that is characteristic to this model
- a graphical depiction of this organization of services for the purpose of communication

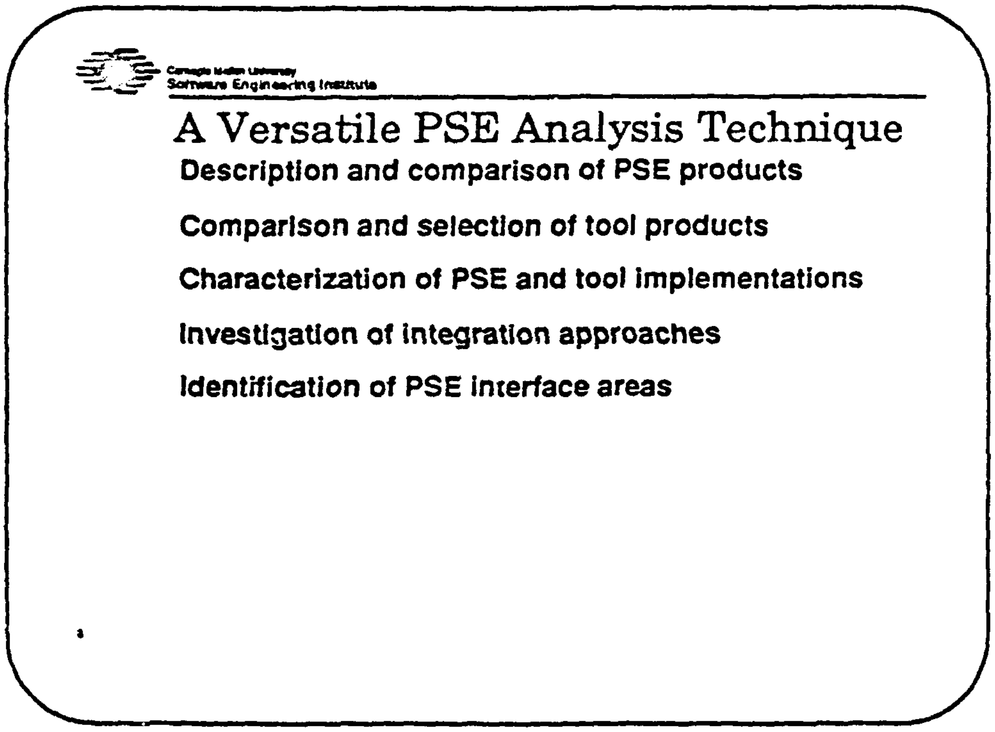
Reminder: The PSESRM document will be available as a NGCR PSESWG document in 1992.

A paper discussing the concepts of the PSESRM and its use will be available as an SEI technical report in January 1992.

Other papers on the objectives, goals, and strategy for PSESWG are available. See the NGCR PSESWG flyer.

NIST/ECMA framework reference model papers are available from NIST.

PSESRM Project Support Environment Services Reference Model



Carnegie Mellon University
Software Engineering Institute

A Versatile PSE Analysis Technique

Description and comparison of PSE products

Comparison and selection of tool products

Characterization of PSE and tool implementations

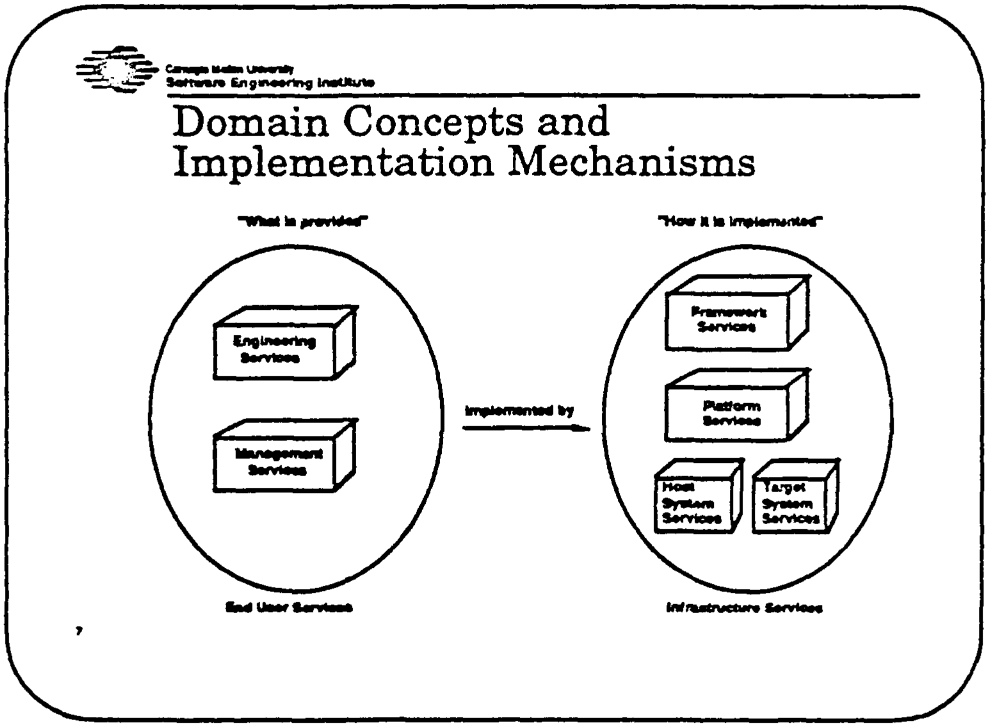
Investigation of integration approaches

Identification of PSE interface areas

NOTES

The PSESRM can be used as a basis for analysis of a variety of aspects of PSEs. This presentation will highlight the use of the PSESRM as an analysis technique for:

- describing desired populated PSEs
- comparing populated PSEs
- selecting tool to be placed in a PSE
- describing both functionality and implementation of PSEs and tools
- determining ways to integrate tools
- selecting relevant interface standards.



Carnegie Mellon University
Software Engineering Institute

Domain Concepts and Implementation Mechanisms

"What is provided"

"How it is implemented"

Engineering Services

Management Services

implemented by

Framework Services

Platform Services

Host System Services

Target System Services

End User Services

Infrastructure Services

7

NOTES

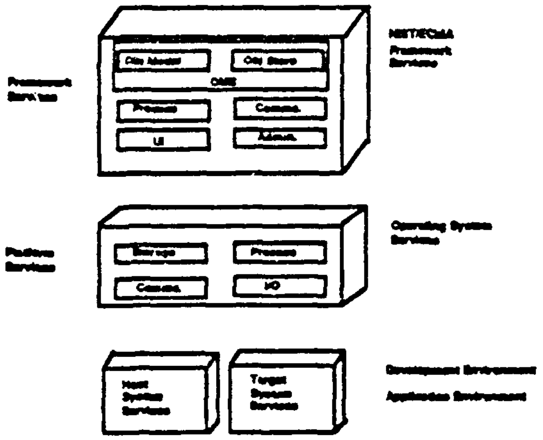
This illustration shows a separation of services into those provided to the end user of a populated PSE and those that are available in the *infrastructure* to implement the end user services. This separation allows us to examine PSEs and their integration at the conceptual level of the engineering and management domains independently from integration at the mechanism level of the implementation.

Tool products can be characterized through a combination of end user and infrastructure services.

End user services can be provided in terms of framework services and platform services, or can interface directly with the host or target system.

Carnegie Mellon University
Software Engineering Institute

Infrastructure Services



NOTES

The framework services represent those of the NIST/ECMA reference model. The framework services are considered to be part of a PSE product set. The illustration shows service groups. Each service group contains a number of services not shown here, but documented in the NIST/ECMA framework reference model report. The data services (OMS) are shown here as consisting of two subgroups, *object model* and *object storage*–a useful separation for our analysis.

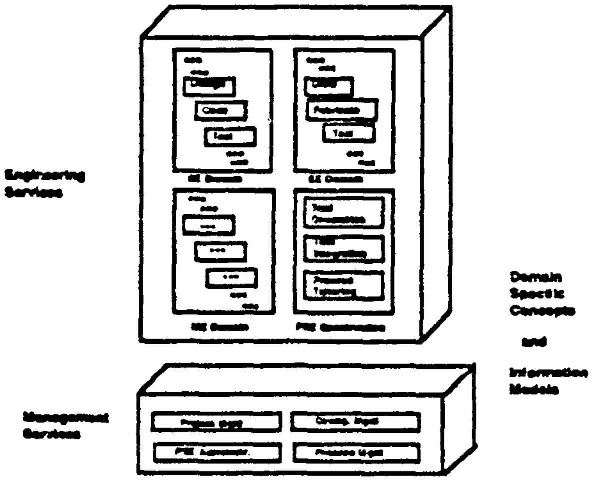
The platform services are expected to be provided by the underlying computing environment. Their interfaces are relevant, but not necessarily their implementation. The platform services are expected to correspond to POSIX.

The host and target system are shown as two services. The purpose of this is to raise awareness of the distinction between the development environment and the application environment. Tools that are part of a PSE may interface to the host environment via the framework and platform services or directly through the host system services, while potentially interfacing to the target environment via a different set of services.

OMS	Object Management System
I/O	Input/Output
UI	User Interface

Carnegie Mellon University
Software Engineering Institute

PSE End User Services



NOTES

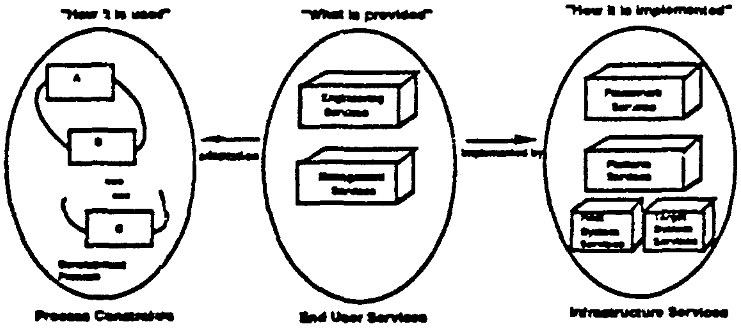
The engineering services are grouped into several (not necessarily exclusive) engineering domains. One of the domains is that of engineering a populated PSE, referred to in the figure as PSE construction. Each domain contains service groups which are refined into services

At this level of tha PSESRM we find (engineering) domain specific concepts and information models. For example, this is where configuration management concepts such as long transaction or information models such as multi-language symbol table formats can be found. Individual services as well as service groups may have information models associated.

SE	Software Engineering
EE	Electrical Engineering
ME	Mechanical Engineering

Carnegie Mellon University
Software Engineering Institute

Process and Project Support Environments



10

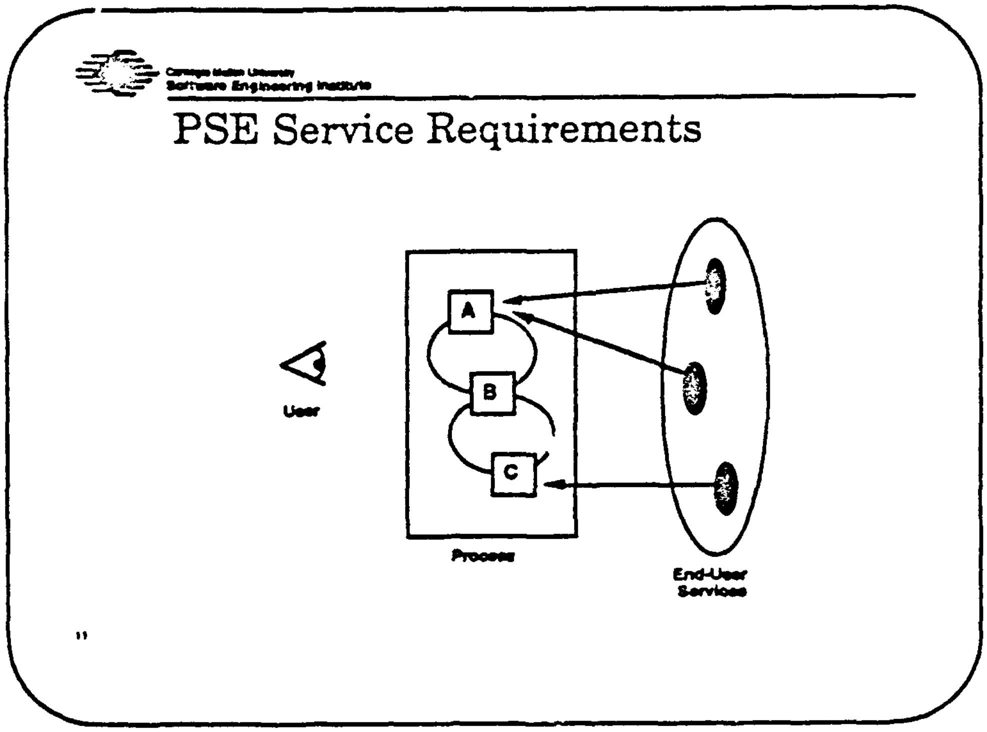
NOTES

The complete PSESRM includes a consideration for software processes. The need for certain services in a PPSE is determined by the process it is intended to support. Although the model accommodates process, computer-based support for process enactment is only emerging.

Typically, only elements of the process are reflected in end user services and their information models.

STARS is investigating process centered SEEs, which would lead us to PBPPSE(SE)s.

PPSE	Populated Project Support Environment
PBPPSE(SE)	Process Based Populated Project Support Environment for Software Engineering



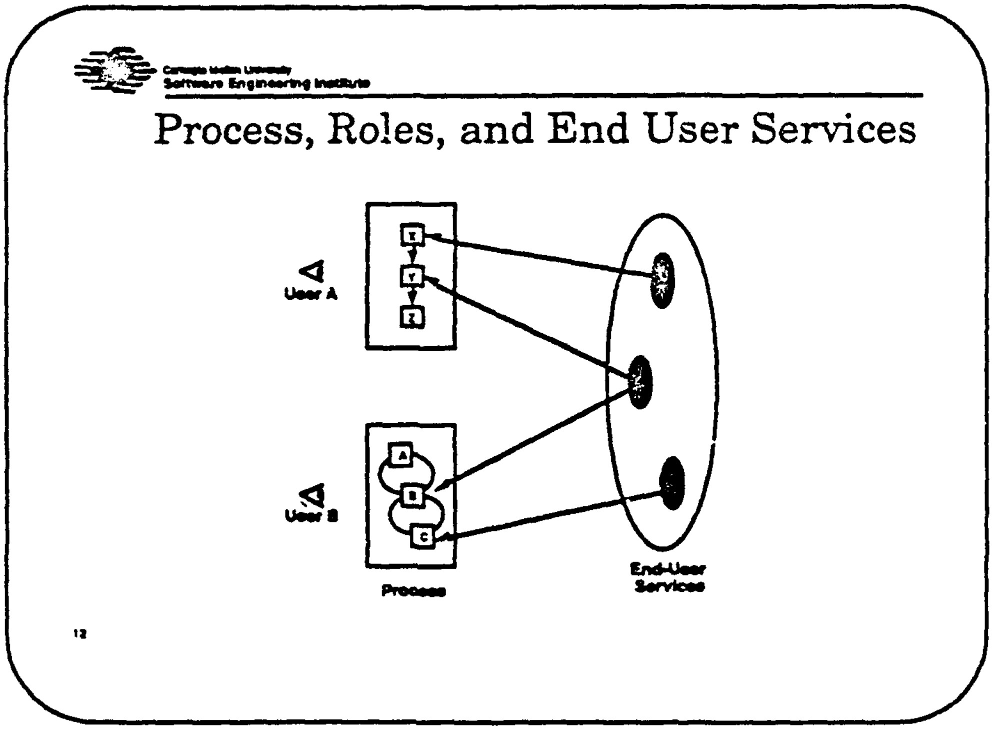
Carnegie Mellon University
Software Engineering Institute

PSE Service Requirements

NOTES

The requirements for services to be provided by a PPSE are determined by the process to be supported. This can be illustrated as a view the user of a PPSE has of the end user services in the PSESRM. This view uses the process level as a filter.

A particular service may be used in different parts of the process.



Carnegie Mellon University
Software Engineering Institute

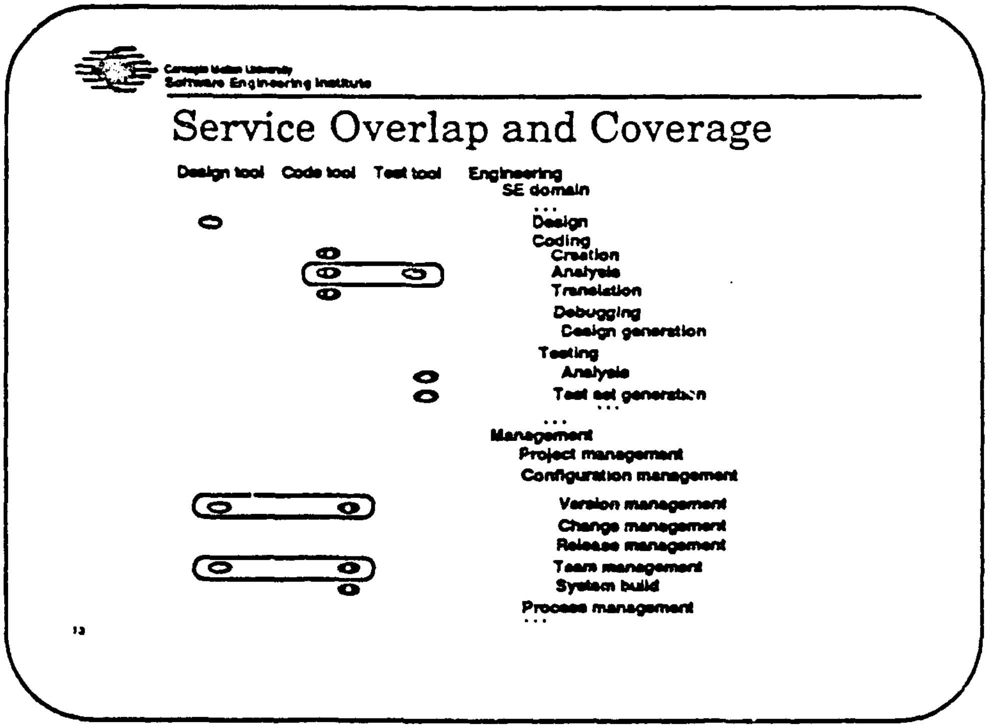
Process, Roles, and End User Services

12

NOTES

The view concept via the process level can be interpreted in several ways:

- different PPSEs supporting different processes can be characterized;
- a particular PPSE and its end user services can support several process variants;
- different users of a PPSE may have different roles, each represented by a different subprocess. These subprocesses interact and make up the complete process. Each subprocess acts as a filter to the subset of end user services appropriate to the role.



Carnegie Mellon University
Software Engineering Institute

Service Overlap and Coverage

Design tool Code tool Test tool

Engineering
SE domain
...
Design
Coding
Creation
Analysis
Translation
Debugging
Design generation
Testing
Analysis
Test set generation
...
...
Management
Project management
Configuration management
Version management
Change management
Release management
Team management
System build
Process management
...

13

NOTES

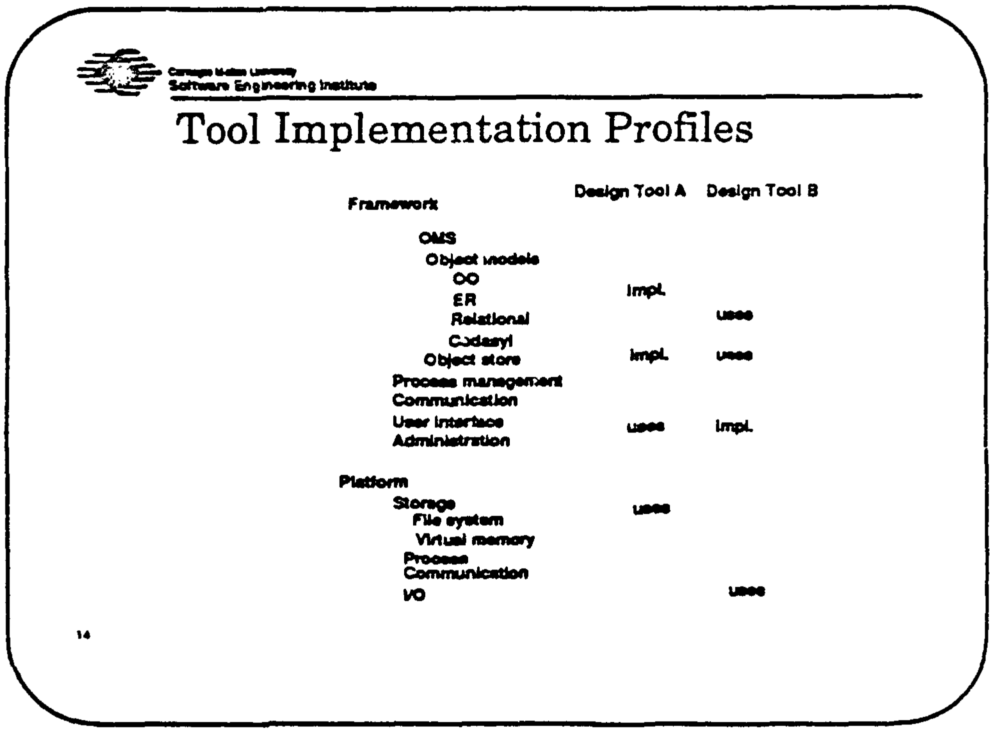
Analysis at the end user service level.

Different tools can be characterized by the services they provide.

This service profile can be used in two ways:

- to determine service overlap between tools to be combined in a PSE. Service overlaps may require attention, as incompatible concepts may be supported or common information models may be maintained redundantly.

- to determine service coverage by a given set of tools in order to satisfy the requirements of a desired PPSE.



Carnegie Mellon University
Software Engineering Institute

Tool Implementation Profiles

	Design Tool A	Design Tool B
Framework		
OMS		
Object models		
OO	Impl.	
ER		
Relational		uses
Codasyl		
Object store	Impl.	uses
Process management		
Communication		
User interface	uses	Impl.
Administration		
Platform		
Storage	uses	
File system		
Virtual memory		
Process Communication		
I/O		uses

14

NOTES

Tools implement the end user services they provide through infrastructure services. Some infrastructure services may be implemented by the tools themselves, while others are expected to be available in the execution environment of the tool.

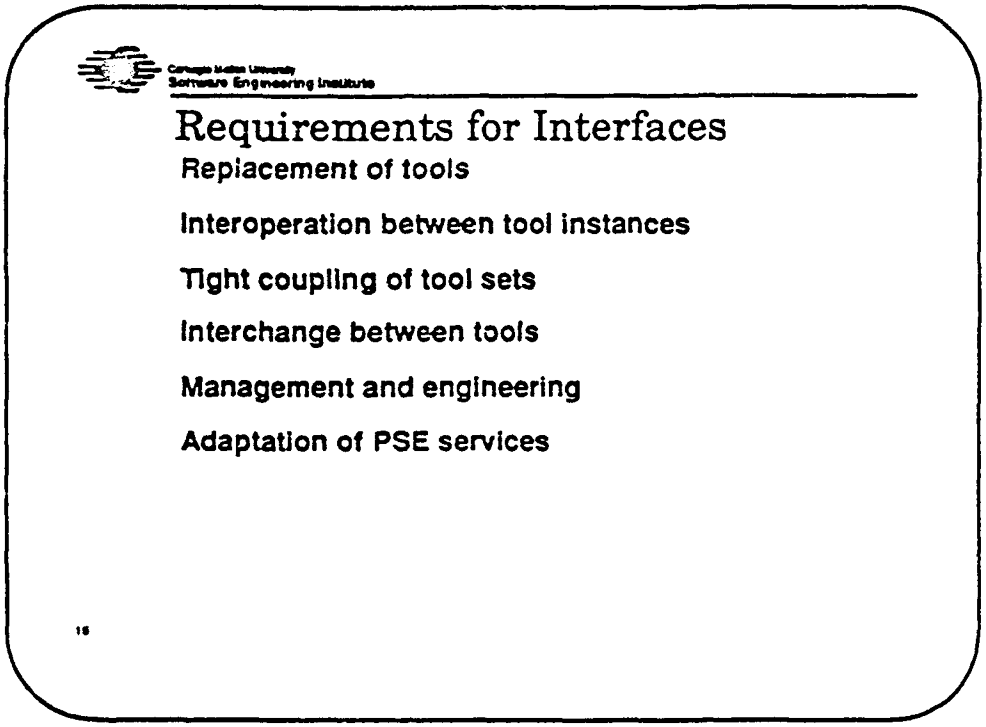
The illustration shows one tool implementing its data dictionary through its own data base while the other tool expects to use a separate data base product. At the same time, one tool implements its user interface itself on top of platform services, while the other tool uses higher level UI services, e.g., X-Motif.

When examining these implementation profiles, areas of conflict can be identified, including:

- difference in data (object) model used
- potential differences in "look and feel"

This illustration also highlights the need for tools to make interfaces to infrastructure services they implement publicly available in order to encourage different degrees of integration and interoperation.

OO	Object Oriented
ER	Entity Relationship



Carnegie Mellon University
Software Engineering Institute

Requirements for Interfaces

Repiacement of tools

Interoperation between tool instances

Tight coupling of tool sets

Interchange between tools

Management and engineering

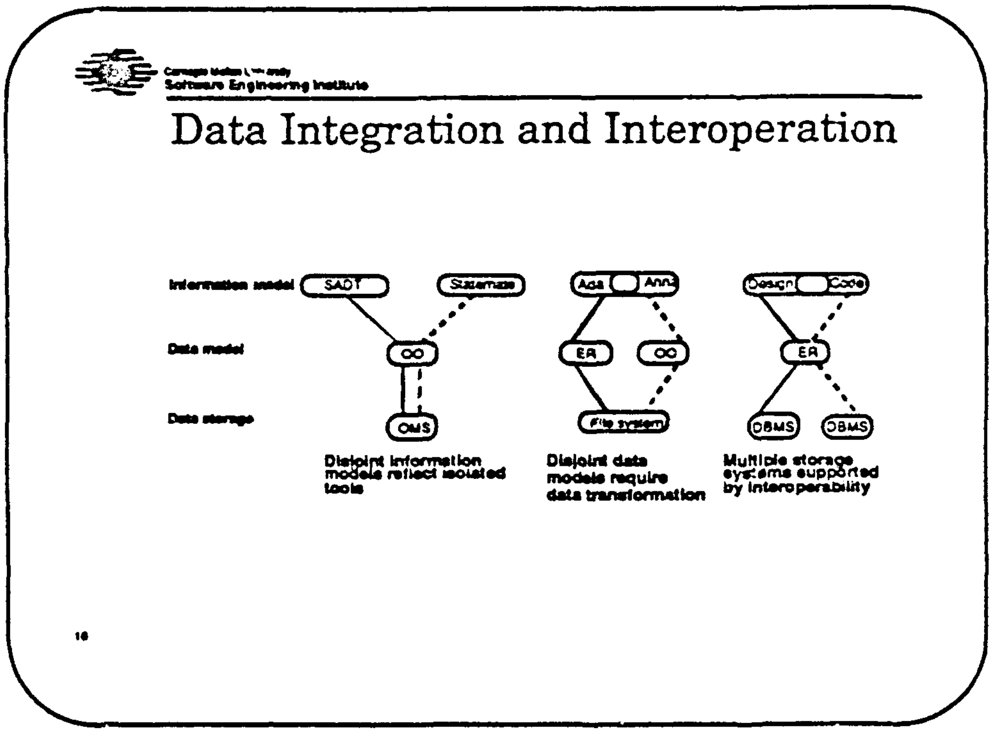
Adaptation of PSE services

15

NOTES

The need for interfaces can be determined by examining what can change in a PPSE.

- Debugger product X can be replaced by debugger product Y.
- Two instances of the same documentation system product or two different documentation systems products need to interchange information
- A design tool set consists of edition, analysis package, layout capability, and code generator. These components share information models and concepts. Subsets of these may be visible outside this to collection of services.
- Information may have to be transformed in order to be mapped between the information model of a design tool and a code tool.
- Management tools such as metric collection tools are interested in certain information from a number of engineering services.
- Many tools offer a layer which allows tailoring of the services. Examples are user interface tailoring and user profiles for tools.



Carnegie Mellon University
Software Engineering Institute

Data Integration and Interoperation

Information model
SADT
Statemate
Ada
Anna
Design
Code
Data model
OO
ER
OO
ER
Data storage
OMS
File system
DBMS
DBMS

Disjoint information models reflect isolated tools

Disjoint data models require data transformation

Multiple storage systems supported by interoperability

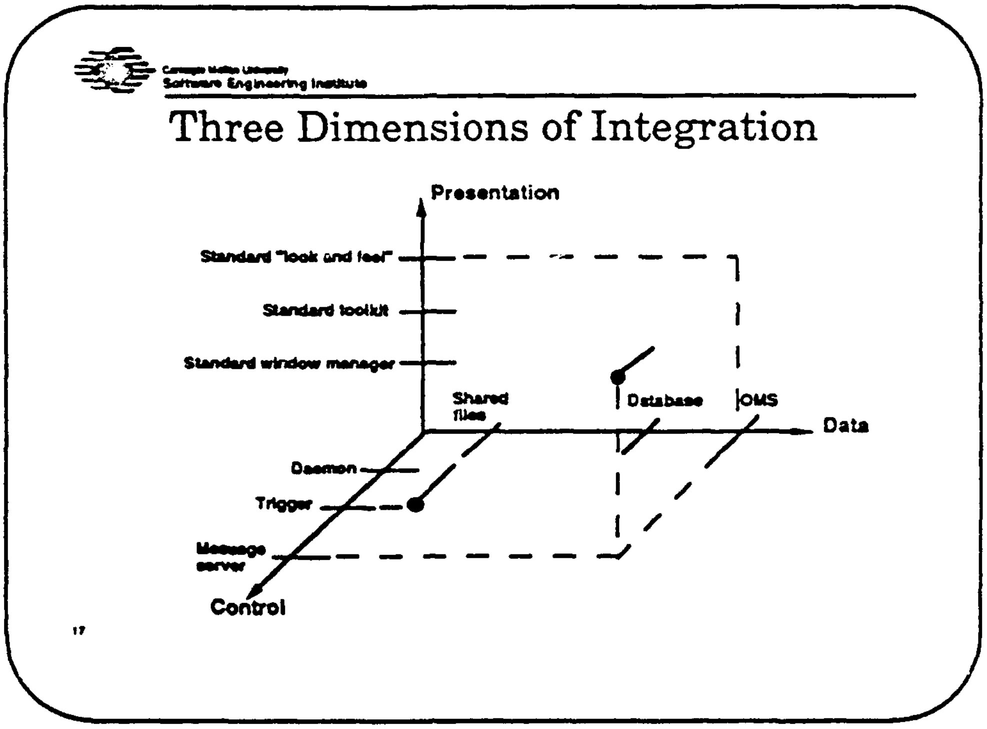
18

NOTES

Data integration is not a single fixpoint. There are degrees of integration and interoperation. Integration occurs at the level of data storage, data model, and information model.

Different tools with varying data storage, data models, and information models are amenable to different integration approaches. Each combination has its own cost/benefit tradeoff.

SADT	Structured Analysis and Design Technique
DBMS	Database Management System



Carnegie Mellon University
Software Engineering Institute

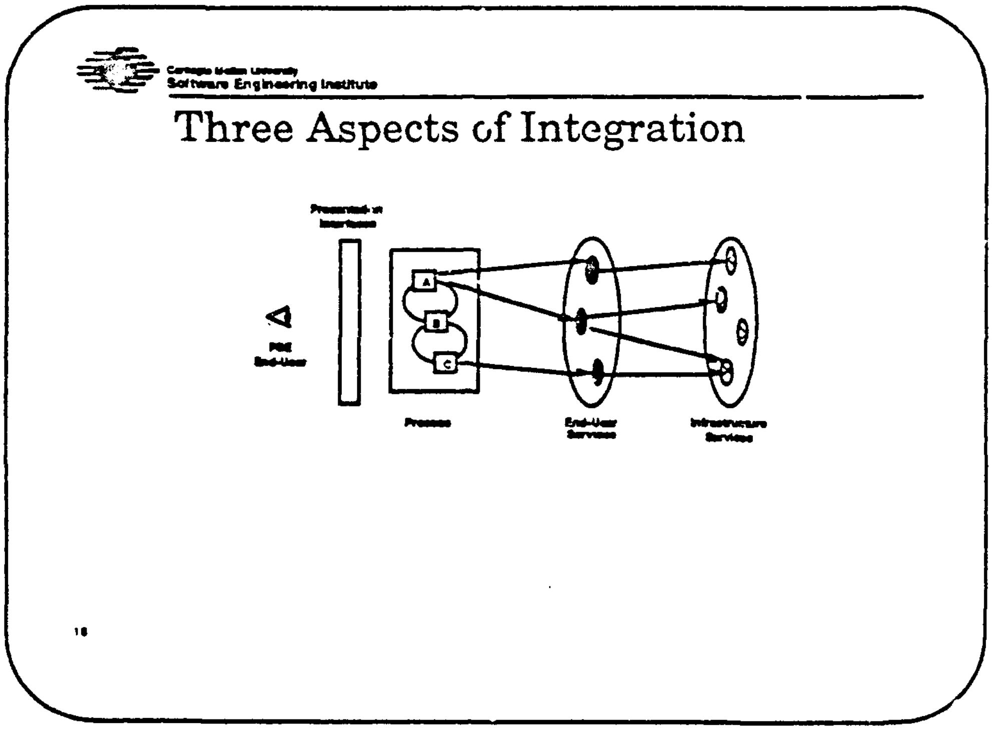
Three Dimensions of Integration

17

NOTES

This represents a view of integration commonly accepted by the CASE industry. Notice that a particular integration approach typically does not follow a single dimension.

It has been recognized that a process dimension has to be taken into account.



Carnegie Mellon University
Software Engineering Institute

Three Aspects of Integration

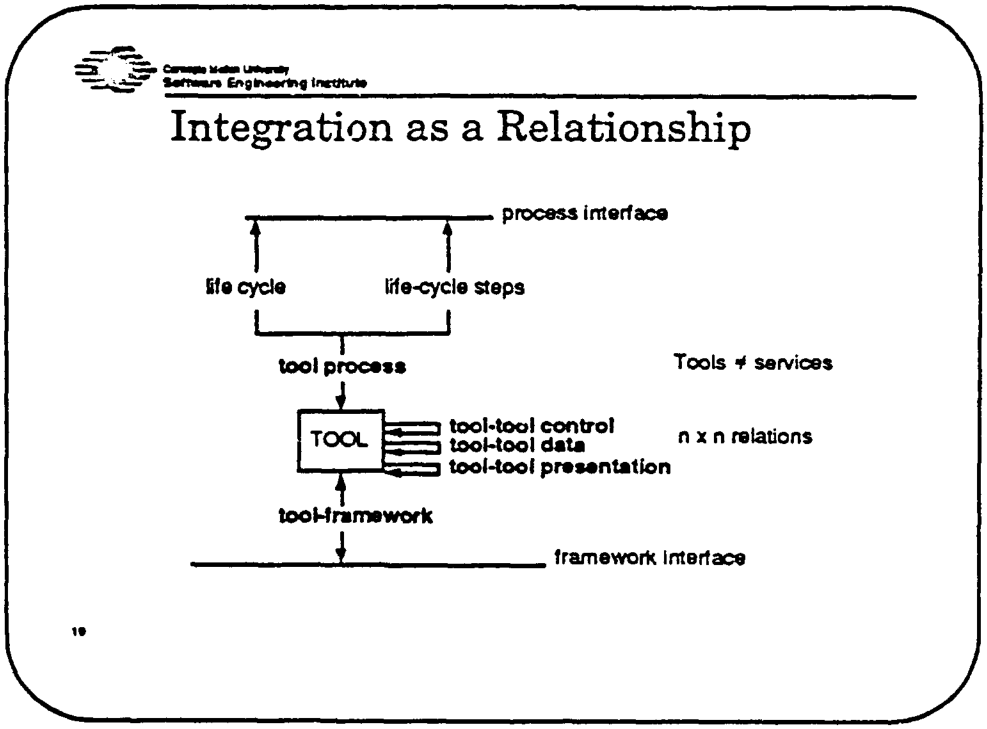
18

NOTES

This slide illustrates that there are three aspects to each of the three dimensions of integration. Presentation integration is used as example.

The PSE end user sees the system presented at the process, end user (concept), and infrastructure (mechanism) layer. Typically user interface services are discussed at the mechanism layer, e.g., much of Motif's "look and feel" is related to UI mechanisms such as menus.

The presentation interface at the end-user service layer addresses issues of how to present engineering domain concepts consistently. Similarly, the process to be followed by the user can be presented in a number of ways.



Carnegie Mellon University
Software Engineering Institute

Integration as a Relationship

process interface

life cycle

life-cycle steps

tool process

Tools ≠ services

TOOL

tool-tool control

tool-tool data

tool-tool presentation

n x n relations

tool-framework

framework interface

19

NOTES

Thomas and Nejmeh recognized that integration is a relationship between two entities, not a property of a single object (see SETA2 invited presentation).

Taking process into account as well, we have tool-process, tool-tool, and tool framework integration. Tool-process integration consists of large grain (life cycle) and small grain (life cycle step) processes being supported by tools.

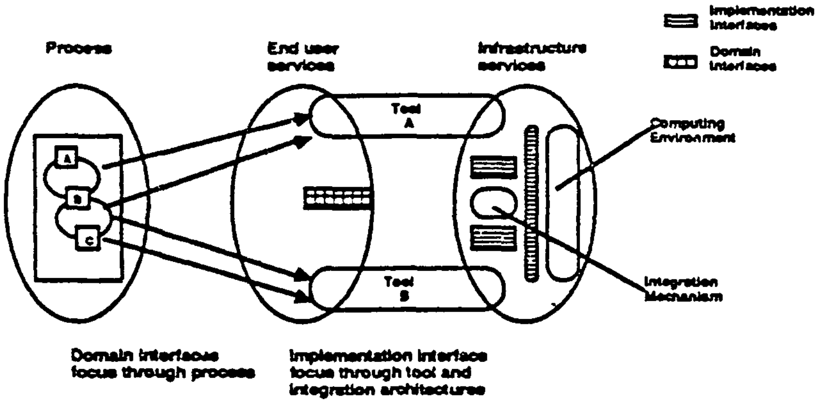
Two observations:

- Tool-tool is a binary relationship. For n tools there are nxn integration possibilities.
- Tools are not services. Tools provide end-user services, but may also implement framework services.

SETA2 Second International Symposium on Environments and Tools for Ada

Carnegie Mellon University
Software Engineering Institute

PSE Interface Areas


Implementation Interfaces
Domain Interfaces
Process
End user services
Infrastructure services
A
B
C
Tool A
Tool B
Computing Environment
Integration Mechanism
Domain interfaces focus through process
Implementation interface focus through tool and integration architectures

20

NOTES

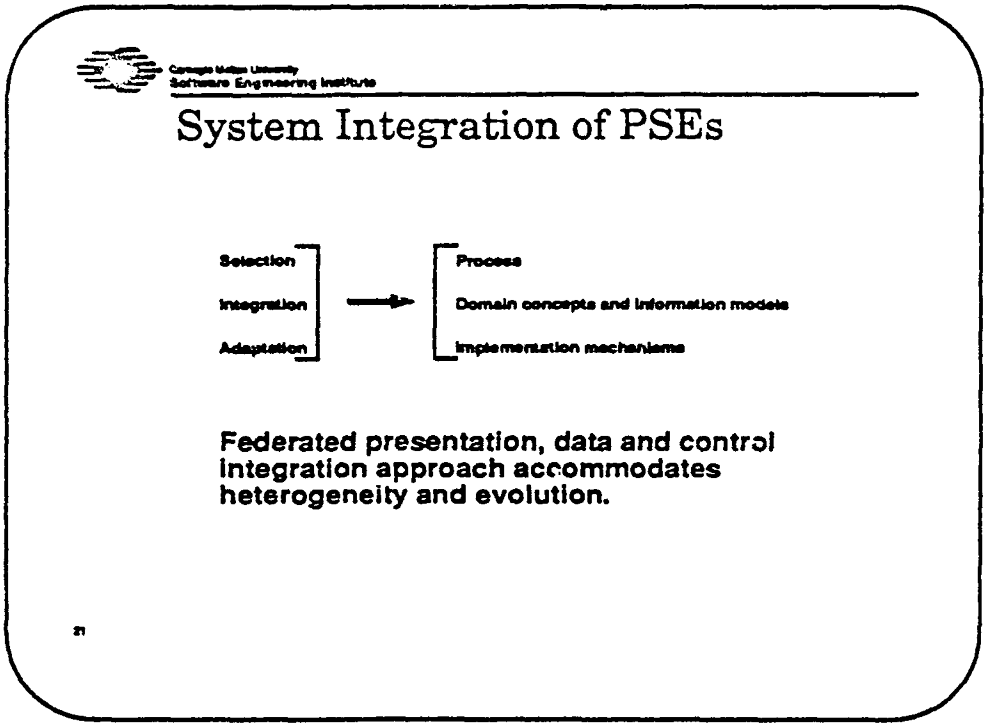
Tools are characterized by the end user services they provide, and the infrastructure services they implement and use.

The process view of end user services reduces the number of relevant tool-tool interfaces.

Tools have domain interfaces at the conceptual level, and implementation interfaces to the base computing environment (e.g., UNIX) and to the integration vehicle (e.g., a BMS).

Tools have certain classes of implementation architectures. Integration approaches use certain infrastructure services. This reduces the number of relevant infrastructure service interfaces. A particular tool, because of its architecture and its restriction to a certain integration paradigm may not need to adhere to the complete set of PSE infrastructure service interface standards.

BMS Broadcast Message Service



Carnegie Mellon University
Software Engineering Institute

System Integration of PSEs

Selection
Integration
Adaptation
→
Process
Domain concepts and information models
Implementation mechanisms

Federated presentation, data and control integration approach accommodates heterogeneity and evolution.

21

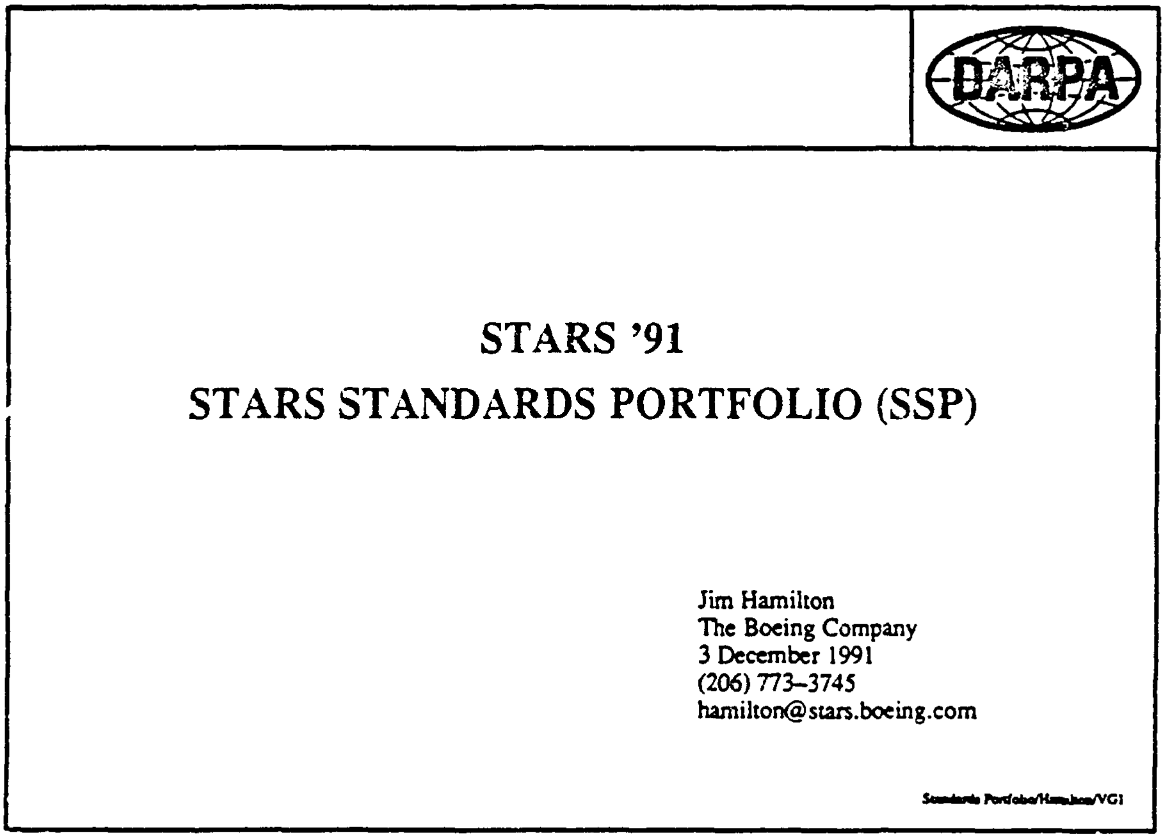
NOTES

PSEs are large systems and have to be treated from the systems integration perspective.

The three major steps of PSE systems integration (selection, integration, and adaptation) have to be performed at all three layers of the PSESRM.

Since PSE technology will continue to evolve rapidly, PSE architectures need to accommodate for the evolving technology, provide a migration path, and support heterogeneity of technology.

A federated systems approach utilizing a combination of control, data, and presentation integration technologies that can interoperate, will be demonstrated by STARS.



DARPA

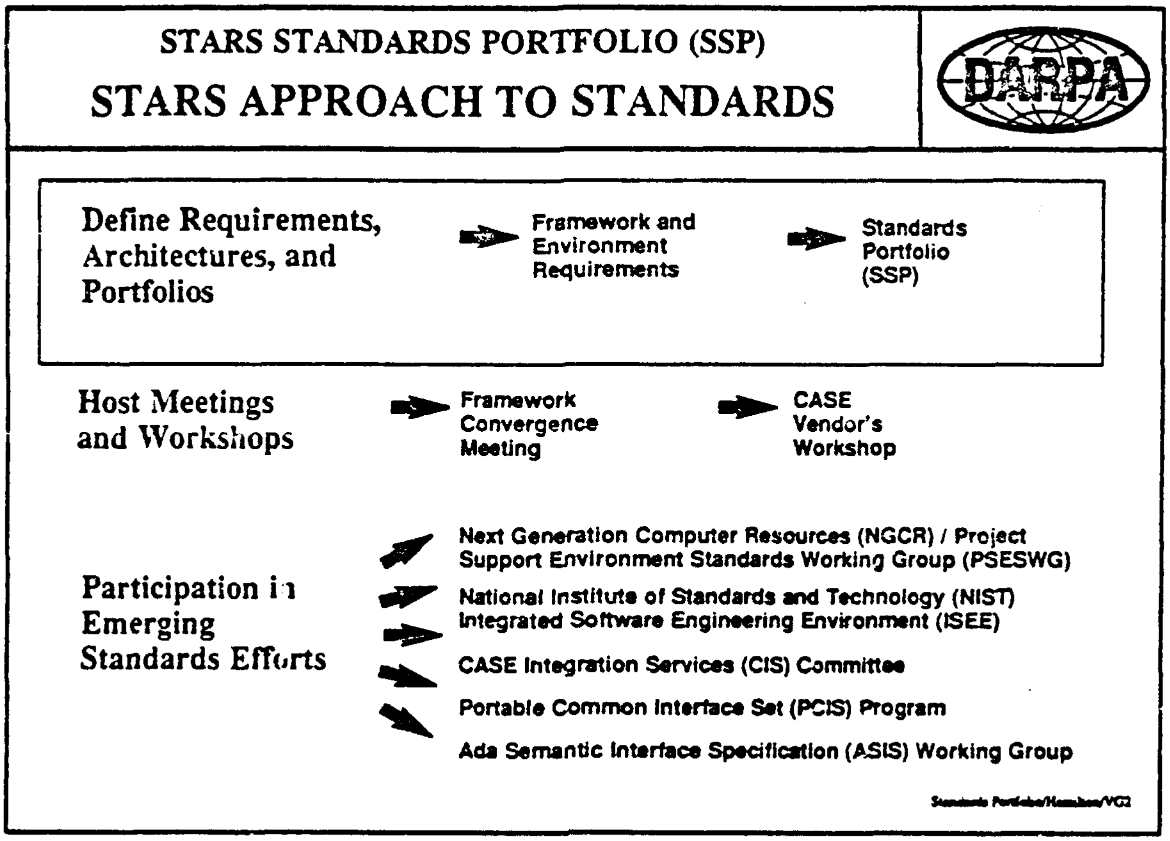
STARS '91

STARS STANDARDS PORTFOLIO (SSP)

Jim Hamilton
The Boeing Company
3 December 1991
(206) 773-3745
hamilton@stars.boeing.com

Standards Portfolio/Hamilton/VG1

Good afternoon. My name is Jim Hamilton from the Boeing STARS Team. I will be describing the STARS Standards Portfolio.



STARS STANDARDS PORTFOLIO (SSP)

STARS APPROACH TO STANDARDS

DARPA

Define Requirements, Architectures, and Portfolios → Framework and Environment Requirements → Standards Portfolio (SSP)

Host Meetings and Workshops → Framework Convergence Meeting → CASE Vendor's Workshop

Participation in Emerging Standards Efforts

- Next Generation Computer Resources (NGCR) / Project Support Environment Standards Working Group (PSESWG)
- National Institute of Standards and Technology (NIST) Integrated Software Engineering Environment (ISEE)
- CASE Integration Services (CIS) Committee
- Portable Common Interface Set (PCIS) Program
- Ada Semantic Interface Specification (ASIS) Working Group

Standards Portfolio/Hamilton/VG2

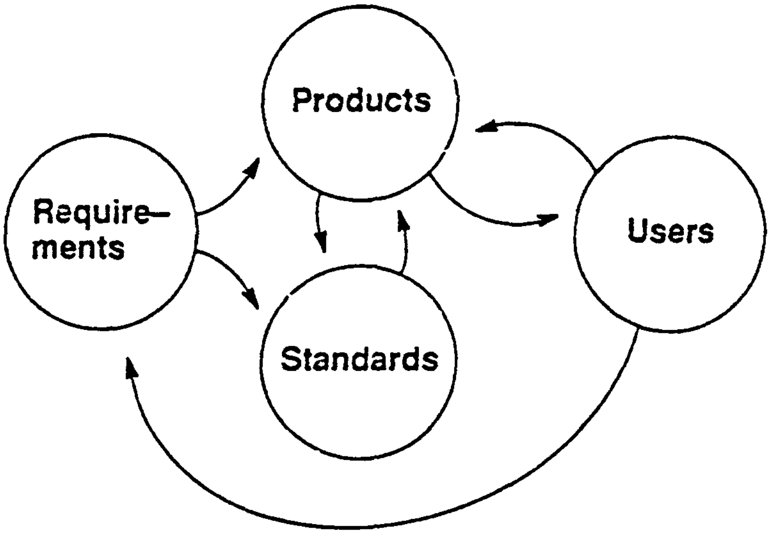
This chart is a common outline for my presentation and the one that follows (Bob Ekman - "STARS Role in Standards Maturation")

I'm going to talk about the STARS Standards Portfolio - the SSP - a list of standards that STARS is committed to. I will describe how these standards are related to STARS software engineering environments. I will present the SSP and it's evolution, and I will solicit your feedback.

The SSP was originally called the STARS Open Architecture Framework (OAF).

STARS STANDARDS PORTFOLIO (SSP)

THE ROLE OF STANDARDS



Standards Portfolio/Hutchison/VG5

Standards are an integral part of the process, but not directly observed by users. Standards provide a common point of reference and development. They may also act as a filter for product selection. Consensus is the key to the standards process.

This is how requirements, standards, products, and users relate.

1. "Requirements" are gathered from several places (RFPs, NIST, STARS, process technology, reuse technology).

2. "Products" are developed to meet needs as expressed in requirements.

3. "Standards" are developed to formalize existing technology or to push forward new technologies.

4. "Products" and "Standards" do a dance of new ideas and implementations.

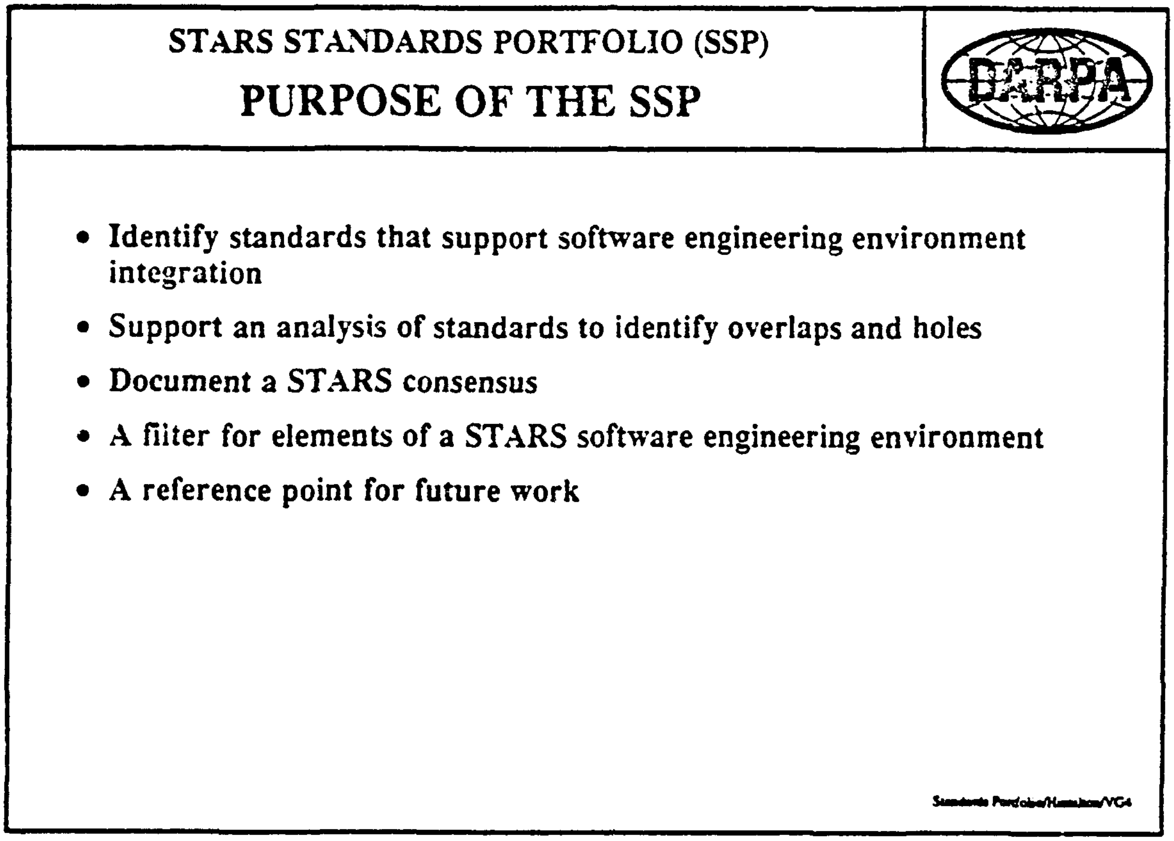
5. "U:. rs" use products – not standards. They feedback their comments on products to the product developers. And the; feedback their understanding of missing capabilities to the requirement gathers.

There are essentially two paths to the development of standards...

1. Use existing products to establish a common ground (examples: IBM PC, Adobe PostScript)

2. Establish a goal where no product exists (examples: Ada, ECMA PCTE).

Both methods are desirable, but STARS will focus more on the second path.



STARS STANDARDS PORTFOLIO (SSP)

PURPOSE OF THE SSP

DARPA

- Identify standards that support software engineering environment integration
- Support an analysis of standards to identify overlaps and holes
- Document a STARS consensus
- A filter for elements of a STARS software engineering environment
- A reference point for future work

Standards Portfolio/Hamilton/VG4

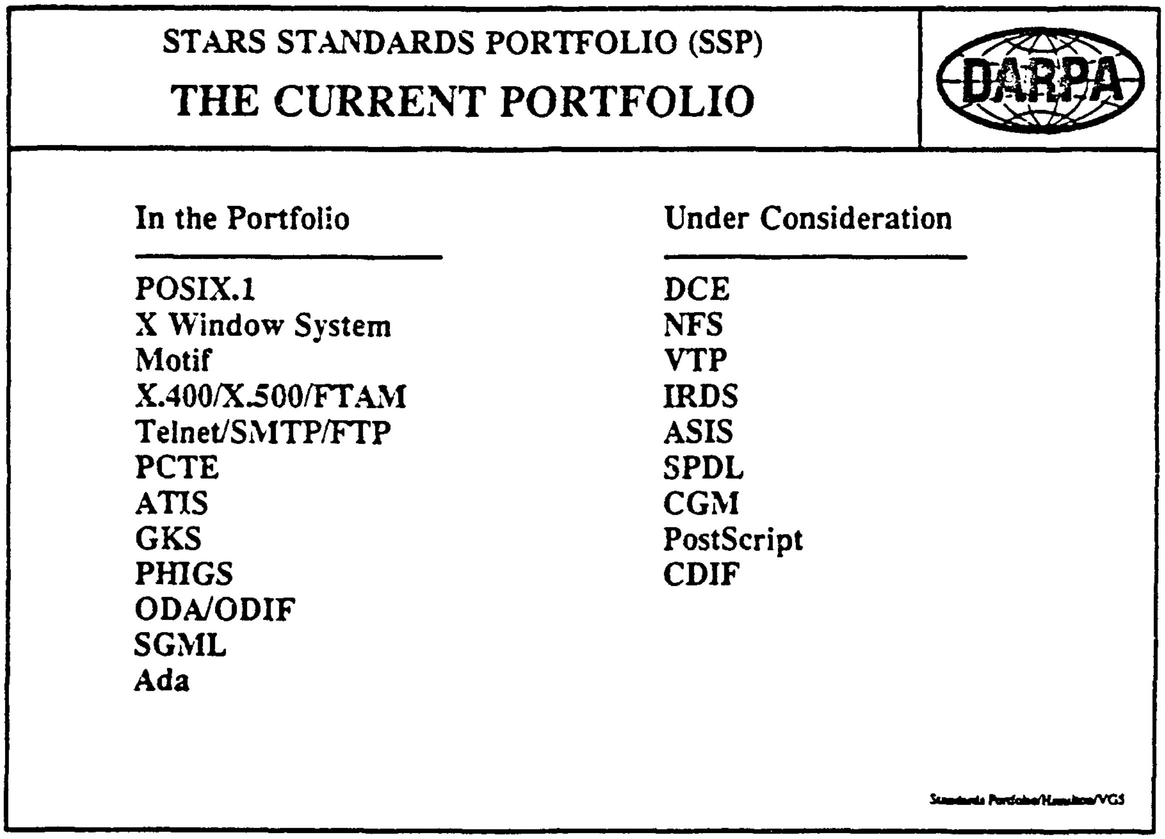
These are the objectives for the SSP.

The SSP is needed by STARS, and useful for other organizations.

POSIX and GOSIP is not all you need - you need more software engineering domain specific standards.

STARS is not developing standards.

You can use the SSP today.



STARS STANDARDS PORTFOLIO (SSP)

THE CURRENT PORTFOLIO

In the Portfolio

POSIX.1
X Window System
Motif
X.400/X.500/FTAM
Telnet/SMTP/FTP
PCTE
ATIS
GKS
PHIGS
ODA/ODIF
SGML
Ada

Under Consideration

DCE
NFS
VTP
IRDS
ASIS
SPDL
CGM
PostScript
CDIF

Standards Portfolio/Hamilton/VGS

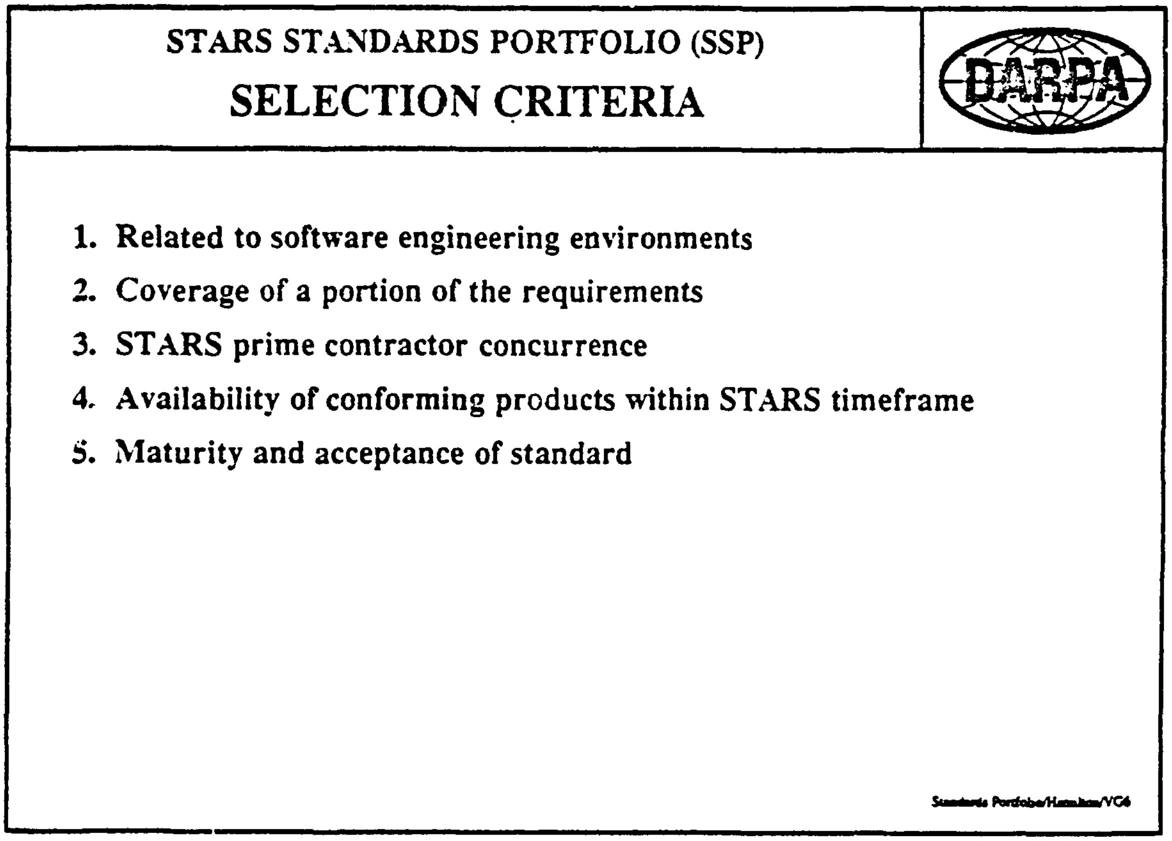
The portfolio is a maturing list. The titles for the columns are ...

"In the Portfolio" = the standard met all the criteria.

"Under consideration" = the standard meets some of the criteria.

standard - controlling organization - title

POSIX.1 - IEEE - Portable Operating System Interface
X - MIT X Consortium - X Window System
Motif - OSF - Motif User Interface
FTAM - CCITT - OSI File Transfer Access and Management
X.400 - CCITT - OSI Message handling system
X.500 - CCITT - OSI Network directory services
Telnet - DARPA - TCP/IP interactive session protocol
SMTP - DARPA - TCP/IP Simple Mail Transfer Protocol
FTP - DARPA - TCP/IP File Transfer Protocol
PCTE - ECMA - Portable Common Tool Environment
ATIS - CIS/X3H6 - A Tool Integration Standard
GKS - X3H3 - Graphical Kernel System
PHIGS - X3H3 - Programmer's Hierarchical Interactive Graphics System
ODA/ODIF - ISO/IEC JTC1 - Open Document Architecture/Interchange Format
SGML - ISO/IEC JTC1 - Standard Generalized Markup Language
Ada - AJPO - Ada programming language
DCE - OSF - Distributed Computing Environment
NFS - IEEE - Network File System
VTP - CCITT - OSI Virtual Terminal Protocol
IRDS - X3H4 - Information Resource Dictionary System
SPDL - ISO/IEC JTC1 - Standardized Page Descriptor Language
CGM - X3H6 - Computer Graphics Metafile
PostScript - Adobe - Page description language
CDIF - EIA - CASE Data Interchange Format
ASIS - STARS - Ada Semantic Interface Specification



STARS STANDARDS PORTFOLIO (SSP)

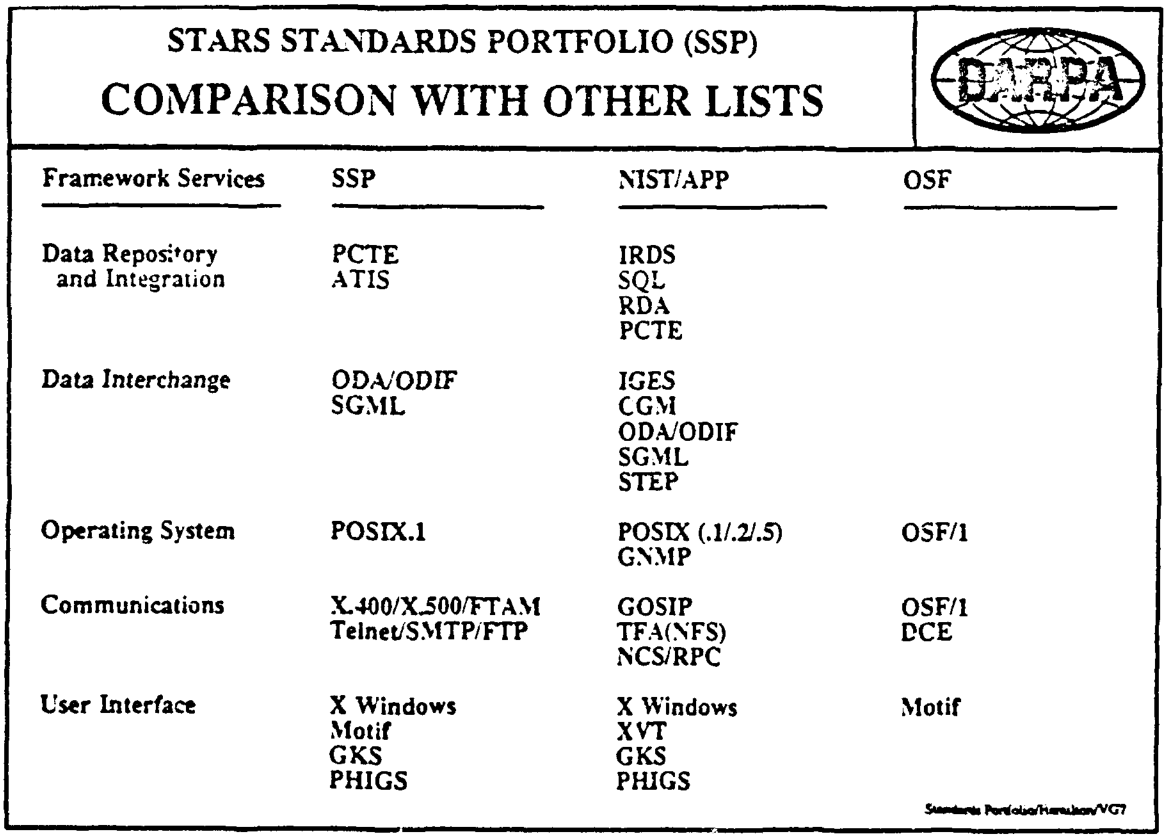
SELECTION CRITERIA

1. Related to software engineering environments
2. Coverage of a portion of the requirements
3. STARS prime contractor concurrence
4. Availability of conforming products within STARS timeframe
5. Maturity and acceptance of standard

Standards Portfolio/Hamilton/VG4

These are the criteria for selection of a STARS standard. The permit STARS to coordinate the assemblage of the STARS SEEs.

1. Software engineering relevance – generally applicable
2. Requirements coverage – this was an engineering effort
3. Commercial counterpart concurrence – market place dimension
4. Availability – this is the pragmatic filter
5. Maturity – consideration of source of standards – international, national, federal, de jure



STARS STANDARDS PORTFOLIO (SSP)

COMPARISON WITH OTHER LISTS

Framework Services	SSP	NIST/APP	OSF
Data Repository and Integration	PCTE ATIS	IRDS SQL RDA PCTE	
Data Interchange	ODA/ODIF SGML	IGES CGM ODA/ODIF SGML STEP	
Operating System	POSIX.1	POSIX (.1/.2/.5) GNMP	OSF/1
Communications	X.400/X.500/FTAM Telnet/SMTP/FTP	GOSIP TFA(NFS) NCS/RPC	OSF/1 DCE
User Interface	X Windows Motif GKS PHIGS	X Windows XVT GKS PHIGS	Motif

Standards Portfolio/Hamilton/VG7

Several other organizations are developing lists, usually with different intents and motivation, but it is interesting to compare...

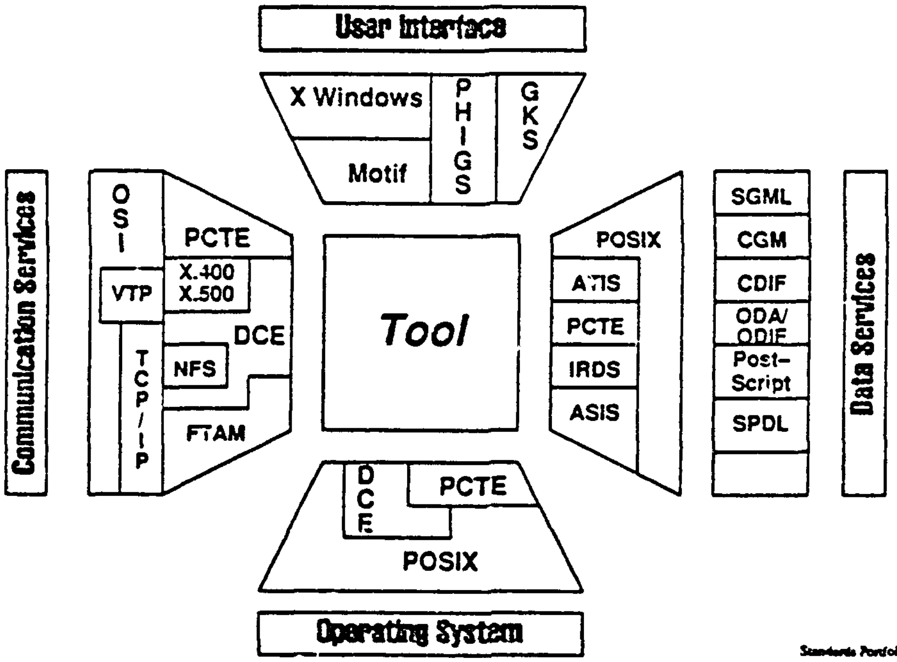
- "Framework Services" are general classes of requirements – similar to the "NIST/ECMA Framework Reference Model " sections.
- "NIST/APP" is a list from the "National Insitute of Standards and Technology (NIST) / Application Portability Profile(APP)".
- "OSF" is a list of the Open Software Foundation's products.

Related Organizations

AJPO – Ada Joint Program Office
ANSI – American National Standards Insitute
CCITT – Consultative Committee for International Telephone and Telegraph
CIS – CASE Integration Services Committee
DARPA – Defense Advanced Research Projects Agency
ECMA – European Computer Manufacturers Association
EIA – Electric Industries Association
IEEE – Insitute of Electrical and Electronics Engineers
ISO/IEC JTC1 – International Standards Organization/International Electrotechnical Commission – Joint Technical Committee
MIT – Massachusetts Insitute of Technology
NIST – National Insitute of Standards and Technology
OSF – Open Software Foundation
X3H3 – ANSI Technical Committee on Graphics
X3H4 – ANSI Technical Committee on IRDS
X3H6 – ANSI Technical Committee on CASE Tool Integration Models

STARS STANDARDS PORTFOLIO (SSP)

TOOL WRITER'S VIEW



Standards Portfolio/Hamilton/VG8

This is the tool writer's view, if all the STARS standards included in the SSP were followed, in a single collection of products.

Notice the overlay and alternatives.

It's not a pretty picture, but it is what they have to choose from. So how does a developer choose? Well, the answer is "pragmatically".

1. The most reliable service (a persistent objectbase must be persistent).
2. The easiest to use (if it is too complex then it will not be used).
3. Available on the developer's platform.
4. The least expensive to acquire and maintain.

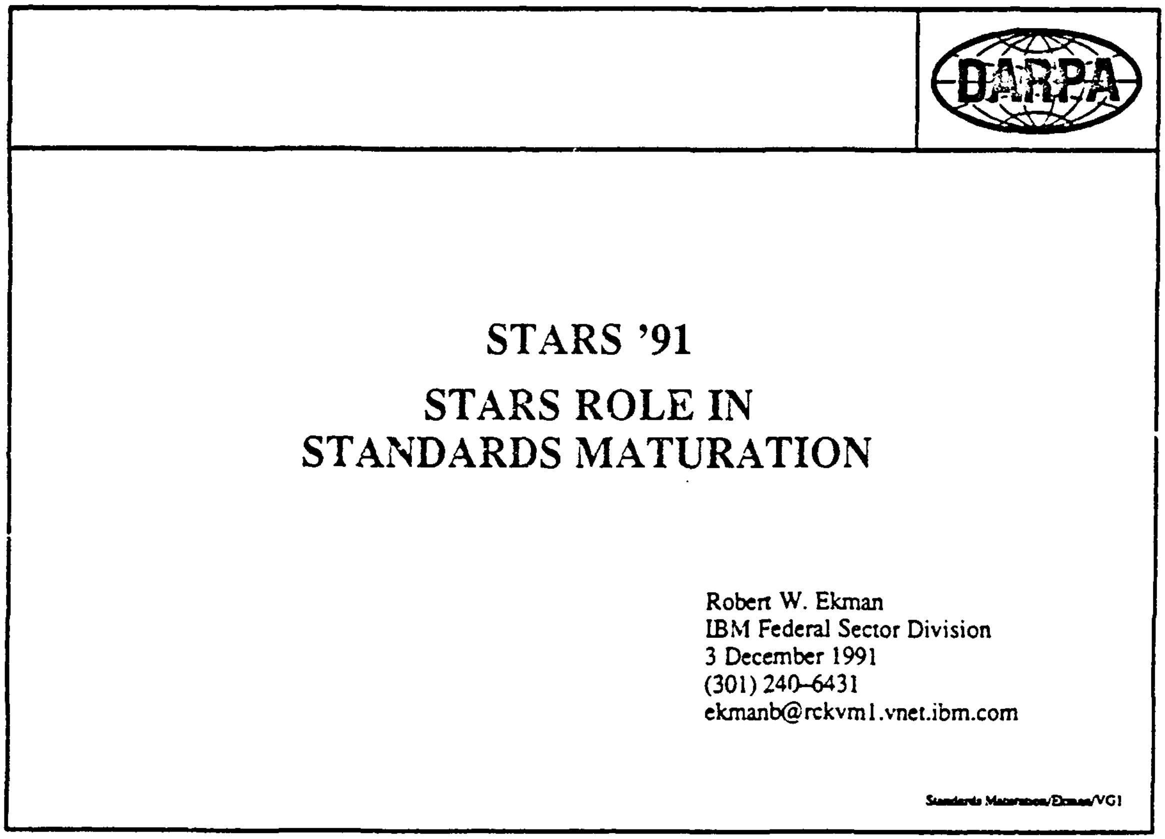
STARS STANDARDS PORTFOLIO (SSP)

SUMMARY

- Version 0.5 available
- Use of standardized framework services will require changes to existing tools
- Continued focus on object management system standards
- Planning evaluation and analysis of information models
- We want your inputs to further evolve the SSP

Standards Portfolio/Hamilton/VG9

In summary, the SSP is available, useful, and maturing. We want your participation.



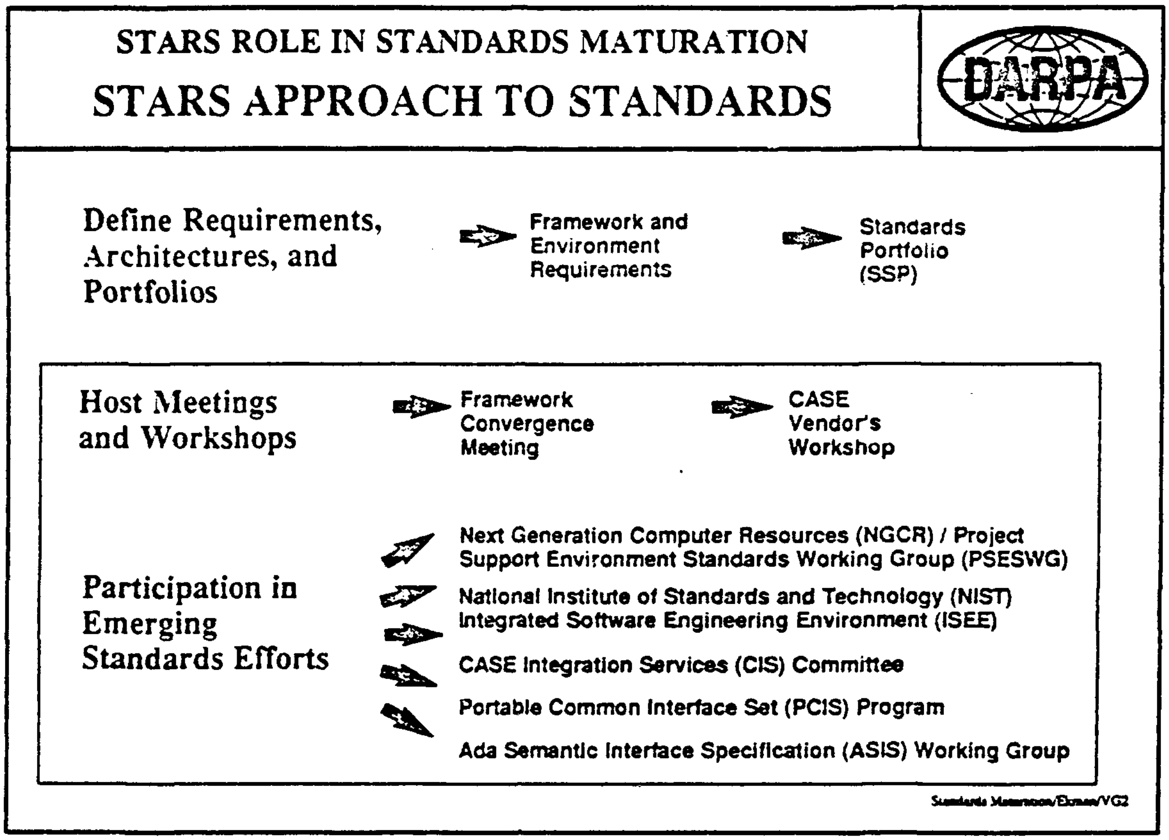
STARS '91

STARS ROLE IN STANDARDS MATURATION

Robert W. Ekman
IBM Federal Sector Division
3 December 1991
(301) 240-6431
ekmanb@rckvm1.vnet.ibm.com

Standards Maturation/Ekman/VG1

Good afternoon. My name is Bob Ekman from the IBM STARS Team, and I will be covering the STARS role in standards maturation.



STARS ROLE IN STANDARDS MATURATION

STARS APPROACH TO STANDARDS

DARPA

Define Requirements, Architectures, and Portfolios → Framework and Environment Requirements → Standards Portfolio (SSP)

Host Meetings and Workshops → Framework Convergence Meeting → CASE Vendor's Workshop

Participation in Emerging Standards Efforts →

- Next Generation Computer Resources (NGCR) / Project Support Environment Standards Working Group (PSESWG)
- National Institute of Standards and Technology (NIST) Integrated Software Engineering Environment (ISEE)
- CASE Integration Services (CIS) Committee
- Portable Common Interface Set (PCIS) Program
- Ada Semantic Interface Specification (ASIS) Working Group

Standards Maturation/Ekman/VG2

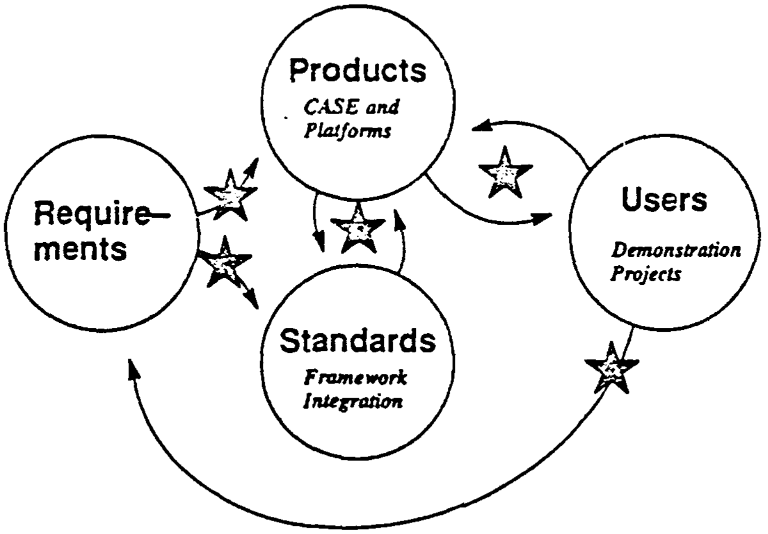
This presentation will further describe the STARS approach to standards.

I will continue the description of of the STARS approach to standards, which was started by Jim Hamilton and his explanation of the STARS Standards Portfolio (SSP). I will cover STARS hosted events, and how STARS has participated in emerging standards efforts.

You will see how STARS influences the direction of industry through outbound technology transfer. And you will see how STARS benefits through inbound technology transfer. This is how STARS assures itself it is in-line with industry direction.

STARS ROLE IN STANDARDS MATURATION

STARS AS A FACILITATOR



Products

CASE and Platforms

Requirements

Users

Demonstration Projects

Standards

Framework Integration

Standards Maturation/Flomen/VG3

Lets us first re-look at the role of standards in improving software engineering...

The process of standards maturation is a slow moving process. STARS is facilitating and accelerating the actions between requirements, products, standards, and users. STARS is placing emphasis on reference models, standards profiles, and Ada support. These actions have been recognized and welcomed by both industry and the standards groups. The STARS demonstration projects are the ultimate winner in this strategy.

The STARS role assumes (or implies) certain responsibilities from the product developers

Framework provider responsibilities:

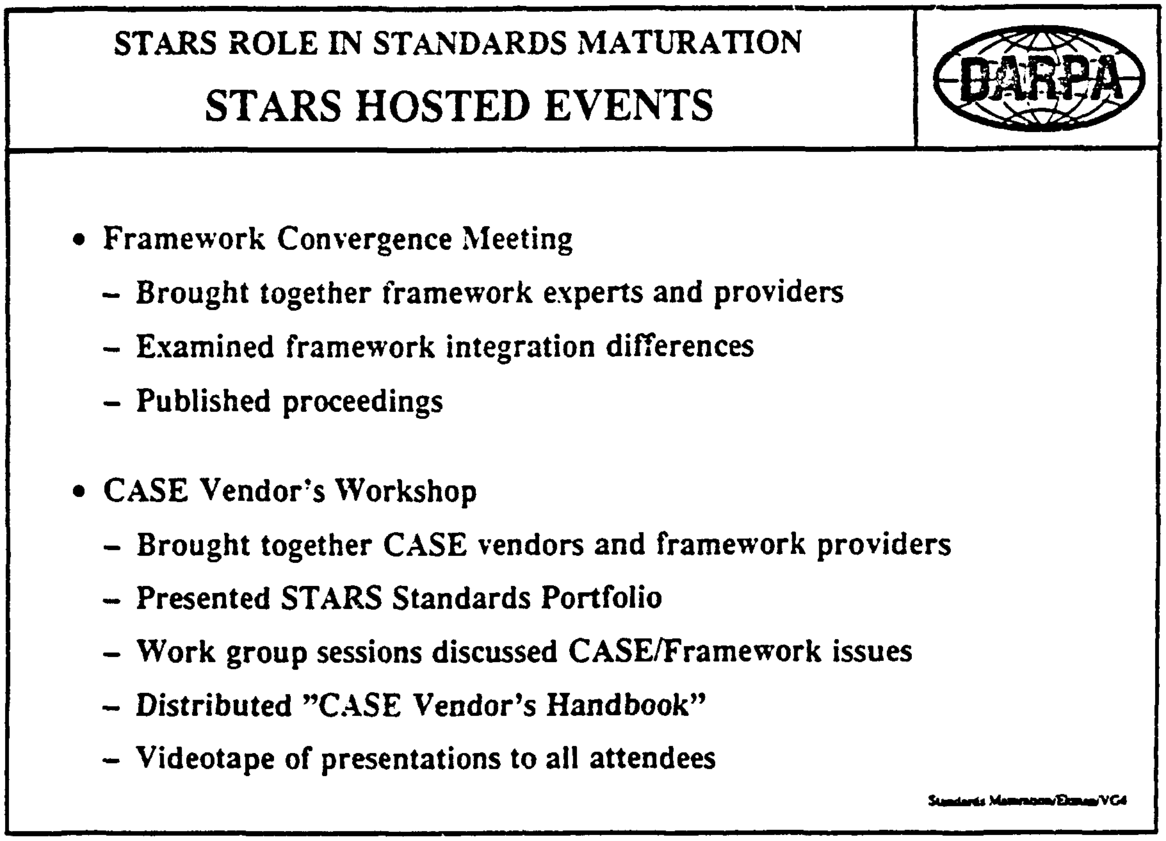
- build to standards
- architect frameworks
- productize the frameworks

Tool vendor responsibilities:

- populate the SEEs
- use the standardized frameworks
- meet special needs

STARS responsibilities:

- provide a neutral territory for tool vendors, framework providers, and standards groups
- experiment with and refine methods of framework integration
- establish demonstration projects



STARS ROLE IN STANDARDS MATURATION

STARS HOSTED EVENTS

- Framework Convergence Meeting
  - Brought together framework experts and providers
  - Examined framework integration differences
  - Published proceedings
- CASE Vendor's Workshop
  - Brought together CASE vendors and framework providers
  - Presented STARS Standards Portfolio
  - Work group sessions discussed CASE/Framework issues
  - Distributed "CASE Vendor's Handbook"
  - Videotape of presentations to all attendees

Standards Maturation/Events/VG4

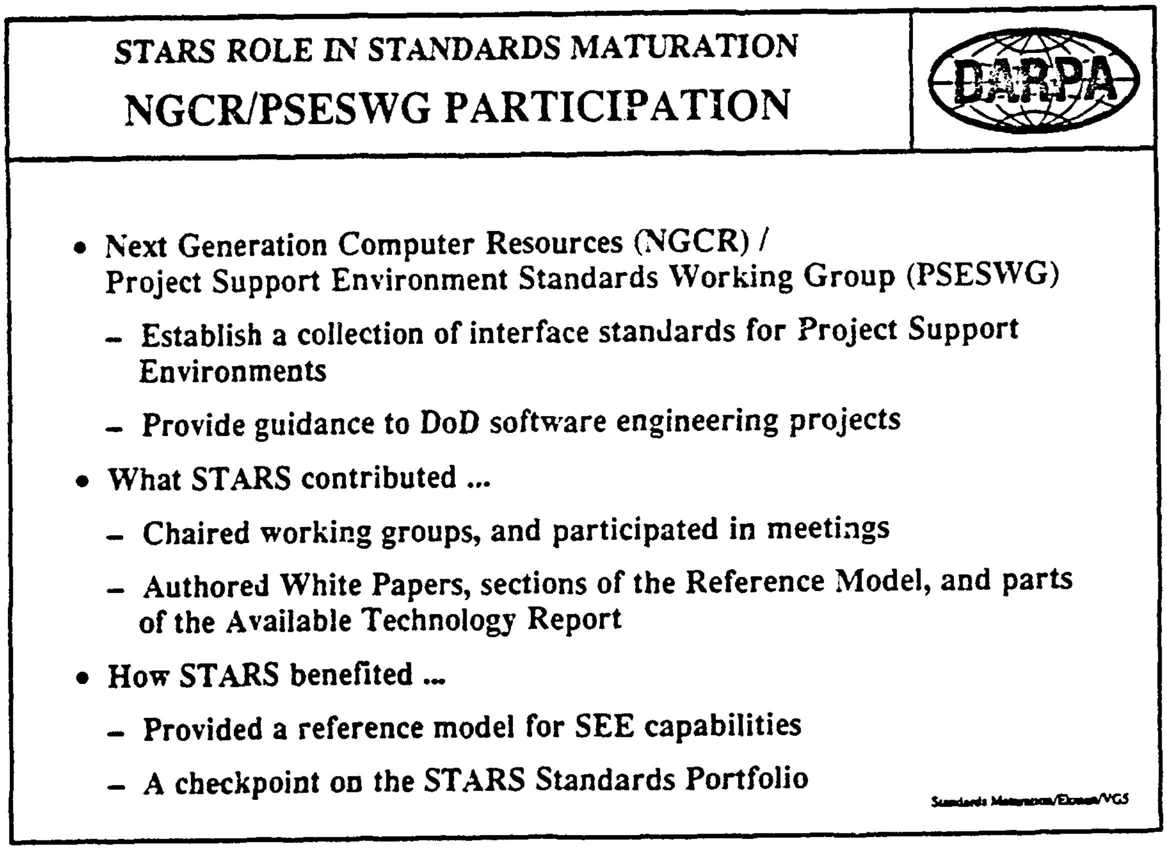
STARS has taken an active role in SEE technology

Framework Convergence Meeting

- Held January 22-23, 1991 at NIST in Gaithersburg
- Participants: STARS prime contractors, STARS commercial counterparts, PCTE, ATIS, and CAIS implementators.
- explored and documented
  - similarities and conflicts of type hierarchy and data models
  - possibilities of method dispatching for PCTE
  - different strategies of versioning and configuration management
- Proceedings are published in an IDA document (D-972), available from DTIC

STARS CASE Vendors Workshop (July 23, 1991)

- Held July 23-24, 1991 in Seattle
- Built a market consensus for the STARS standards and overall program
- Attendance: 135 vendors, DoD representatives, STARS participants
- Dataquest released a CASE tool market analysis - "CASE Vendor's Handbook"
- Videotaped the presentations and distributed to attendees



STARS ROLE IN STANDARDS MATURATION

NGCR/PSESWG PARTICIPATION

- Next Generation Computer Resources (NGCR) / Project Support Environment Standards Working Group (PSESWG)
  - Establish a collection of interface standards for Project Support Environments
  - Provide guidance to DoD software engineering projects
- What STARS contributed ...
  - Chaired working groups, and participated in meetings
  - Authored White Papers, sections of the Reference Model, and parts of the Available Technology Report
- How STARS benefited ...
  - Provided a reference model for SEE capabilities
  - A checkpoint on the STARS Standards Portfolio

Standards Maturation/[illegible]/VGS

NGCR/PSESWG ...

Sponsor: Navy

Membership: Open, by individual

Meetings: 4 per year, last at Newport RI in November

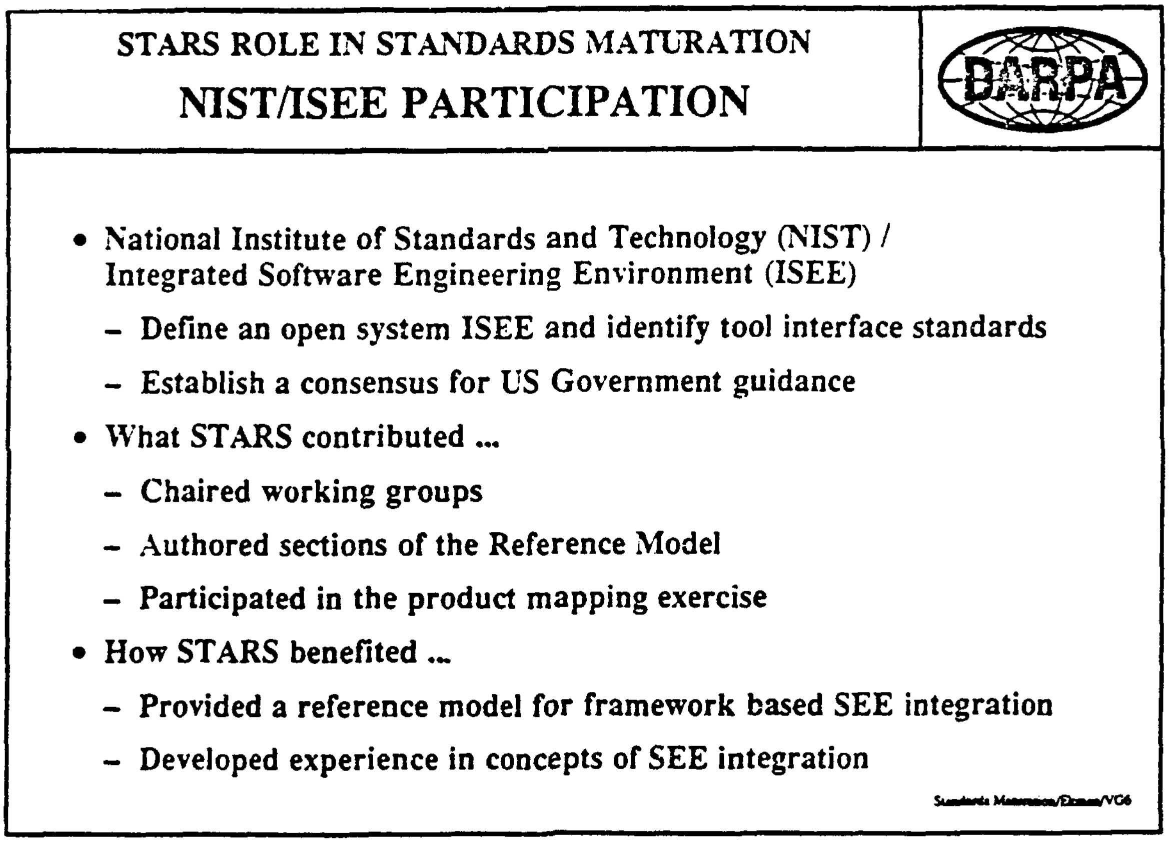
Publications:
"PSE Reference Model White Paper"
"Available Technology Report

PSESWG will provide STARS with a SEE reference model.

NGCR is looking for tri-service sponsorship.

Near term standards selection used the STARS Standards Portfolio as one input.

PSESWG is adopting the "Feiler" model for SEE services and capabilities.



STARS ROLE IN STANDARDS MATURATION

NIST/ISEE PARTICIPATION

- National Institute of Standards and Technology (NIST) / Integrated Software Engineering Environment (ISEE)
  - Define an open system ISEE and identify tool interface standards
  - Establish a consensus for US Government guidance
- What STARS contributed ...
  - Chaired working groups
  - Authored sections of the Reference Model
  - Participated in the product mapping exercise
- How STARS benefited ...
  - Provided a reference model for framework based SEE integration
  - Developed experience in concepts of SEE integration

Standards Maturation/Ekman/VG6

NIST/ISEE ...

Sponsor: NIST

Membership: Open, by individual

Meetings: Twice per year, last at Reston VA in November

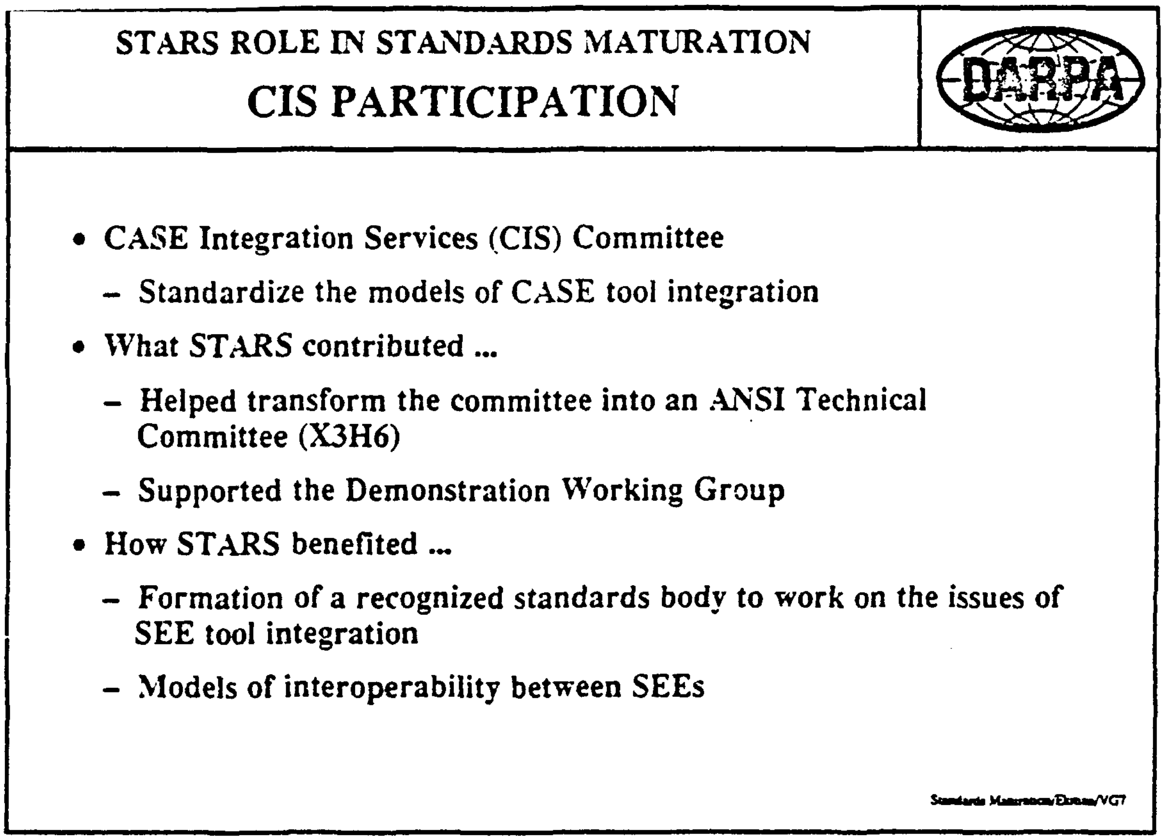
Publications:

"Reference Model for Frameworks of Software Engineering Environments"

NIST/ISEE will provide STARS with a framework services reference model.

NIST is producing a joint reference model with ECMA/TC33.

The ISEE work is coordinated with NGCR/PSESWG efforts.



STARS ROLE IN STANDARDS MATURATION

CIS PARTICIPATION

- CASE Integration Services (CIS) Committee
  - Standardize the models of CASE tool integration
- What STARS contributed ...
  - Helped transform the committee into an ANSI Technical Committee (X3H6)
  - Supported the Demonstration Working Group
- How STARS benefited ...
  - Formation of a recognized standards body to work on the issues of SEE tool integration
  - Models of interoperability between SEEs

Standards Maturation/[illegible]/VG7

CIS ...

Sponsor: Vendor Group

Membership: By Organization

Meetings: 4 per year

Publications:

"Project Proposal for the Development of an American National Standard for CASE Integration Services"
"CASE Integration Services Base Document"

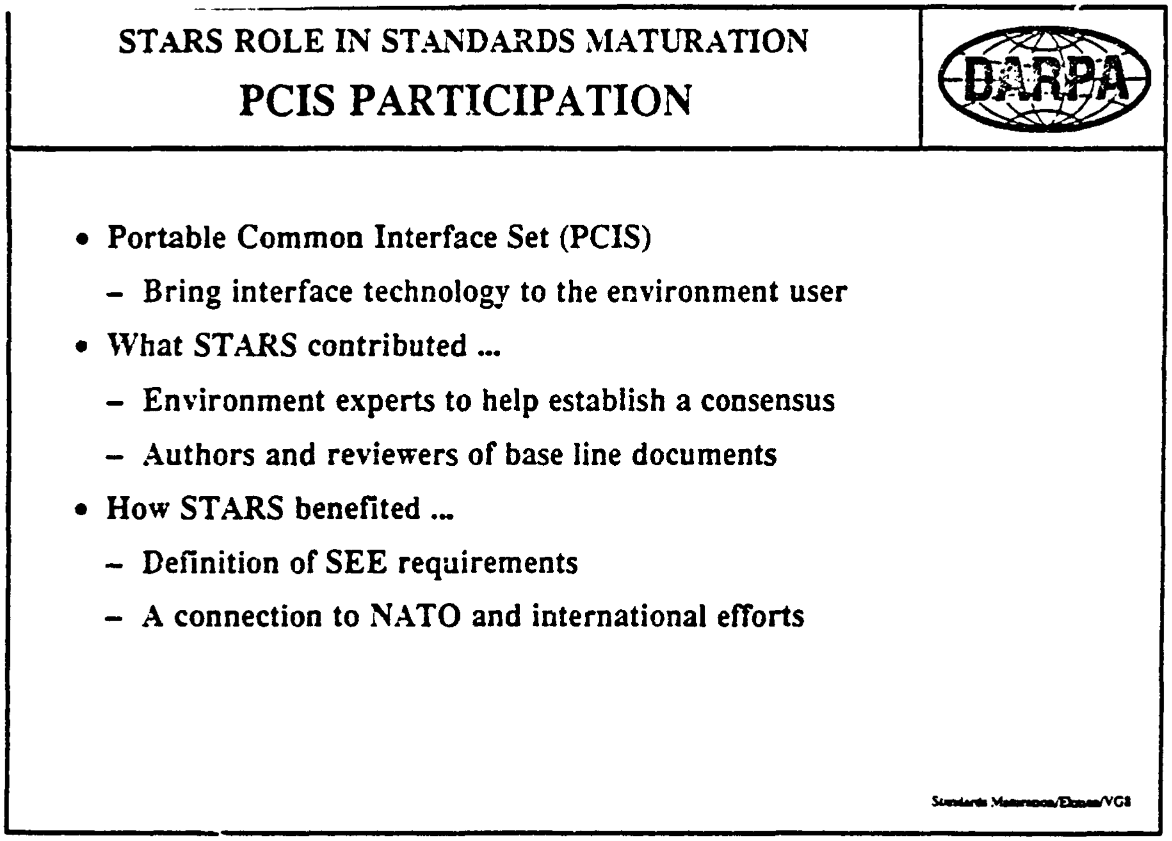
CIS will provide STARS with models of tool integration.

CIS is looking at approaches to tool integration, with emphasis on object-oriented approaches.

The CIS base document is derived from A Tool Integration Standard (ATIS).

Direction is non-divergence from PCTE, and looking for services from IRDS.

X3H6 will have initial meeting in January in Washington DC.



STARS ROLE IN STANDARDS MATURATION

PCIS PARTICIPATION

- Portable Common Interface Set (PCIS)
  - Bring interface technology to the environment user
- What STARS contributed ...
  - Environment experts to help establish a consensus
  - Authors and reviewers of base line documents
- How STARS benefited ...
  - Definition of SEE requirements
  - A connection to NATO and international efforts

Standards Maturation/[illegible]/VG8

PCIS ...

Sponsor: NATO Special Working Group on APSE (AJPO represents the US)

Membership: By AJPO Invitation

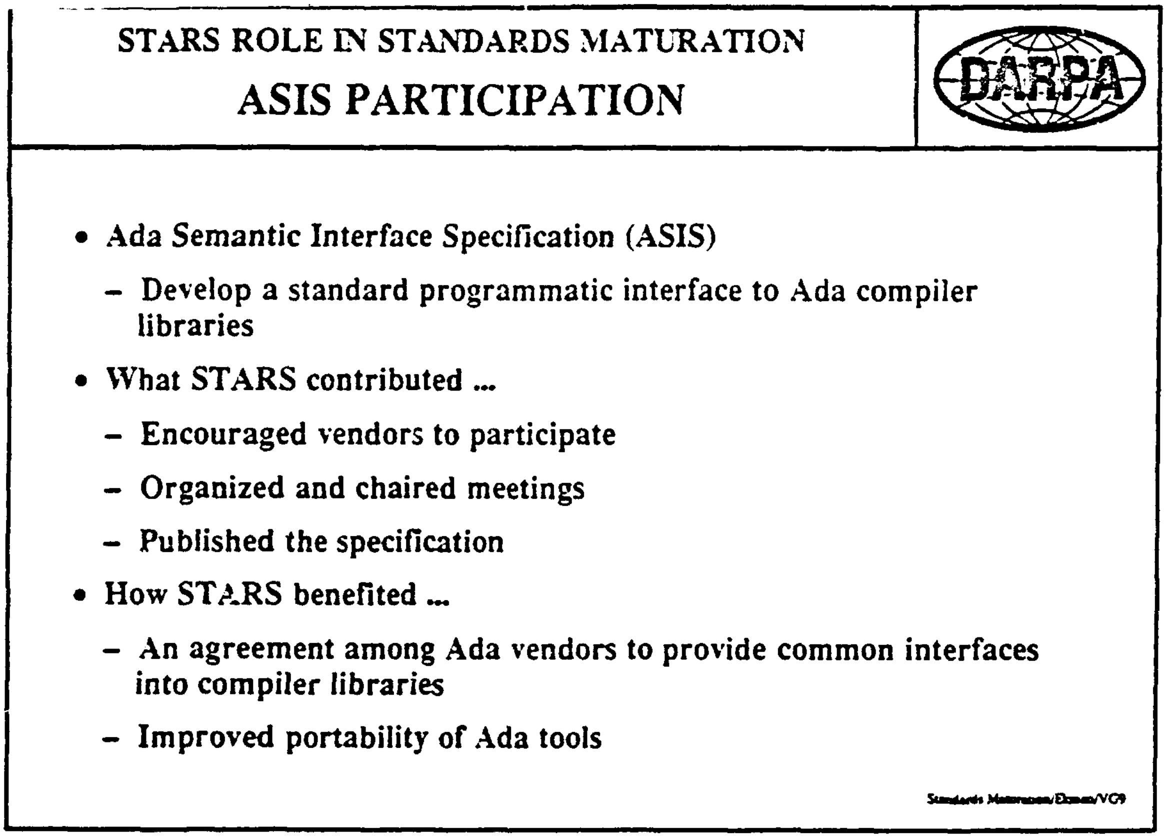
Meetings: Ad–hoc

Publications:

"International Requirements and Design Criteria (IRAC)"
"Interface Technology Analysis"
"Requirements for the PCIS Program"

PCIS is a source for SEE requirements for STARS.

PCIS is part of the STARS long term strategy.



STARS ROLE IN STANDARDS MATURATION

ASIS PARTICIPATION

- Ada Semantic Interface Specification (ASIS)
  - Develop a standard programmatic interface to Ada compiler libraries
- What STARS contributed ...
  - Encouraged vendors to participate
  - Organized and chaired meetings
  - Published the specification
- How STARS benefited ...
  - An agreement among Ada vendors to provide common interfaces into compiler libraries
  - Improved portability of Ada tools

Standards Maturation/[illegible]/VG9

ASIS ...

Sponsor: STARS

Membership: Ada Vendors

Meetings: Ad-hoc

Publications:
"ASIS, Version 0.4, Vendor Independent ASIS"

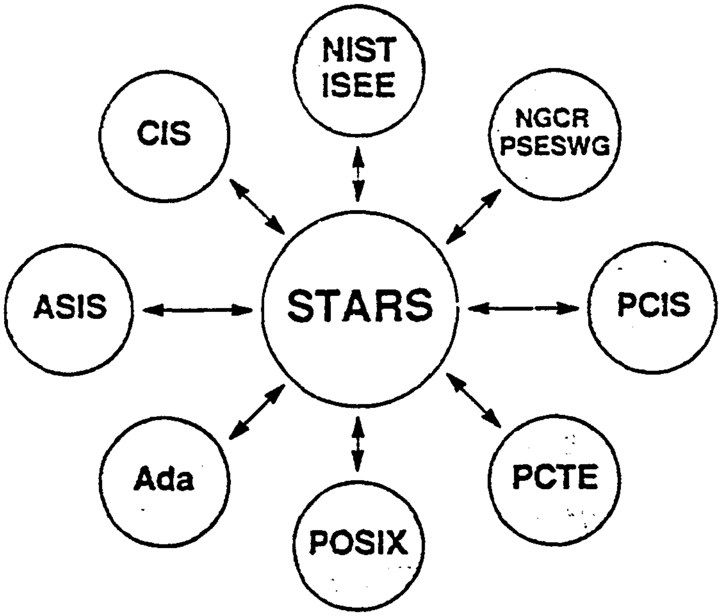
ASIS supports portability of Ada tools.

There is currently a buy-in by TeleSoft and Rational, with interest from Cadre, Verdix, and others.

STARS held a BOF at the October TRI-Ada Conference.

STARS ROLE IN STANDARDS MATURATION

SUMMARY

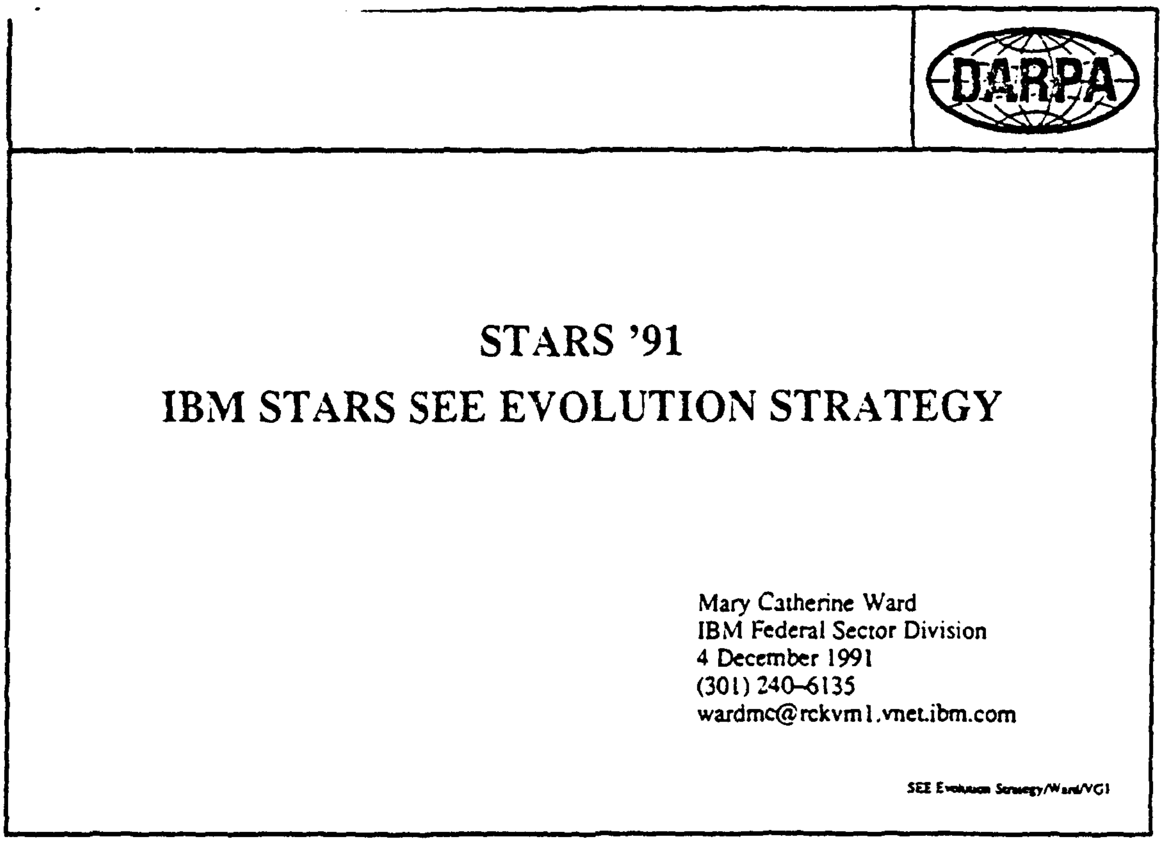


Standards Maturation/Ekman/VG10

STARS will continue to support NIST, NGCR, CIS, ASIS, and PCIS. But it needs to be recognized that both STARS and these organizations are maturing. STARS may be able to relax its focus because of success in seeding these efforts. We are able to trade–off doing standards work inside STARS because of the collection of organizations that are working on the issues outside STARS.

Information models, both abstract and specific implementations, are a growing focus area. STARS is interesting in understanding the issues. We will be working toward some form of convergence.

As a final word, I would emphasis that participation in these standards efforts requires considerable effort. Corporations contribute their expertise, the Government seeds and facilitates the efforts, and individuals professionals have supply ingenuity and personnel time.



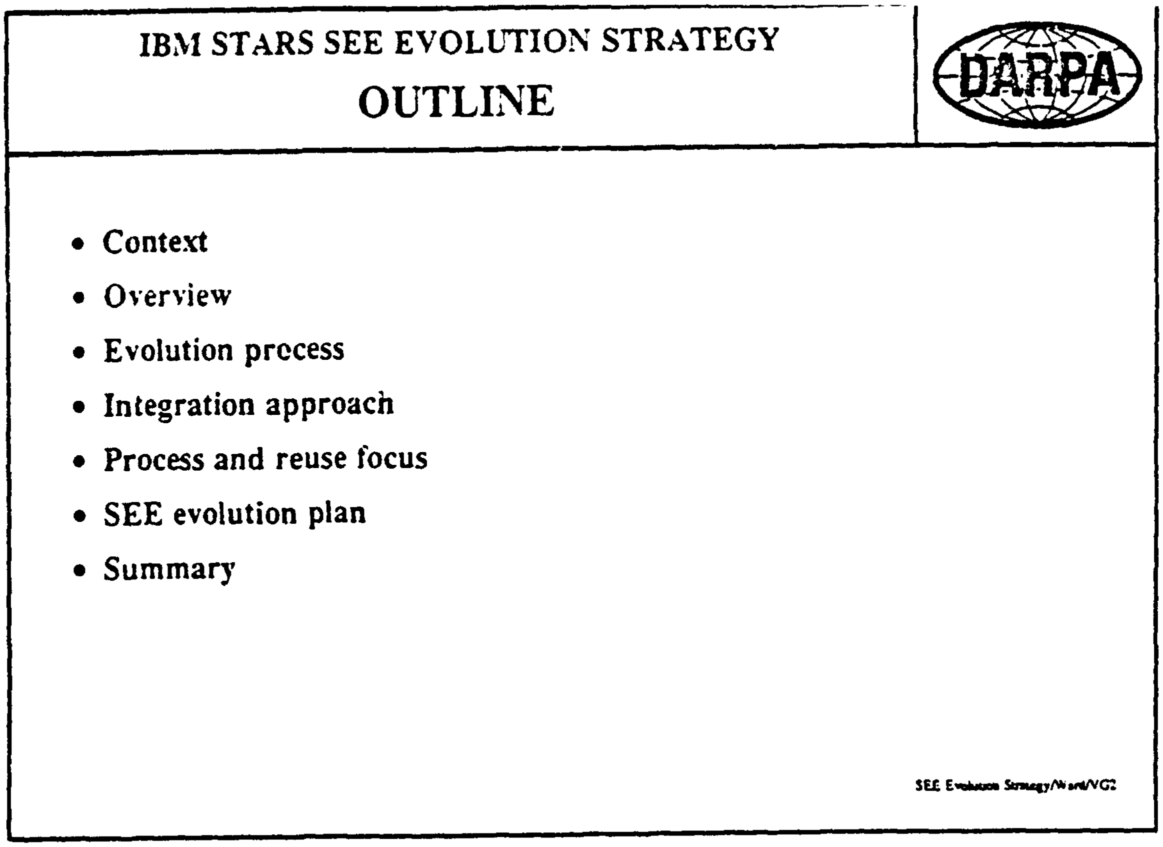
STARS '91

IBM STARS SEE EVOLUTION STRATEGY

Mary Catherine Ward
IBM Federal Sector Division
4 December 1991
(301) 240-6135
wardmc@rckvm1.vnet.ibm.com

SEE Evolution Strategy/Ward/VG1

This presentation describes the strategies of the IBM STARS Team for instantiating SEEs for use on the STARS Demonstration Frojects. The STARS goal is to evolve the SEE into a well-integrated, adaptable, tailorable environment supporting a process-driven, reuse-based engineering approach to megaprogramming.



IBM STARS SEE EVOLUTION STRATEGY
OUTLINE

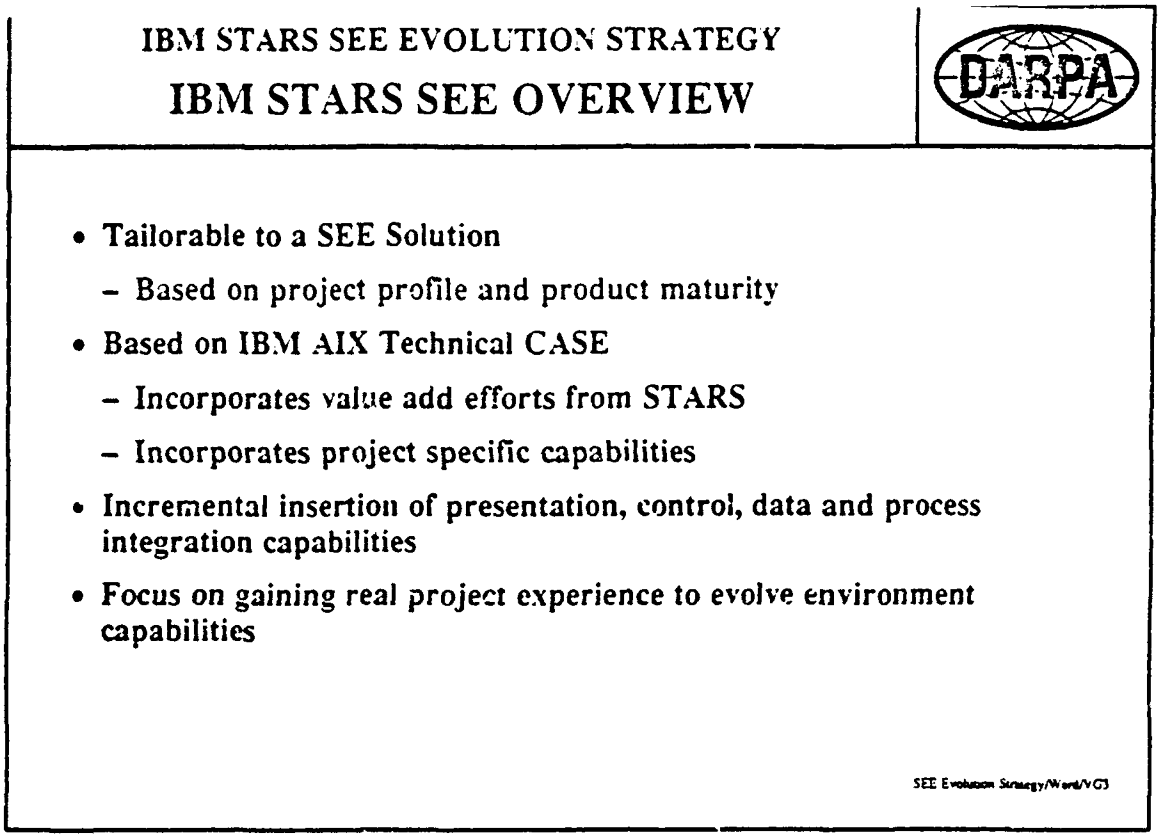
DARPA

- Context
- Overview
- Evolution process
- Integration approach
- Process and reuse focus
- SEE evolution plan
- Summary

SEE Evolution Strategy/Ward/VG2

This presentation addresses the IBM STARS SEE evolution strategy including

- the strategy for evolving to a framework-based, integrated, COTS supplied SEE;
- support for the various levels of integration: presentation, control and data;
- integration of reuse library mechanisms, and process definition and management capabilities; and
- supporting products, including automation of methods and technologies for SEE usage including tool integration and SEE system administration and management.



IBM STARS SEE EVOLUTION STRATEGY

IBM STARS SEE OVERVIEW

- Tailorable to a SEE Solution
  - Based on project profile and product maturity
- Based on IBM AIX Technical CASE
  - Incorporates value add efforts from STARS
  - Incorporates project specific capabilities
- Incremental insertion of presentation, control, data and process integration capabilities
- Focus on gaining real project experience to evolve environment capabilities

SEE Evolution Strategy/Word/VG3

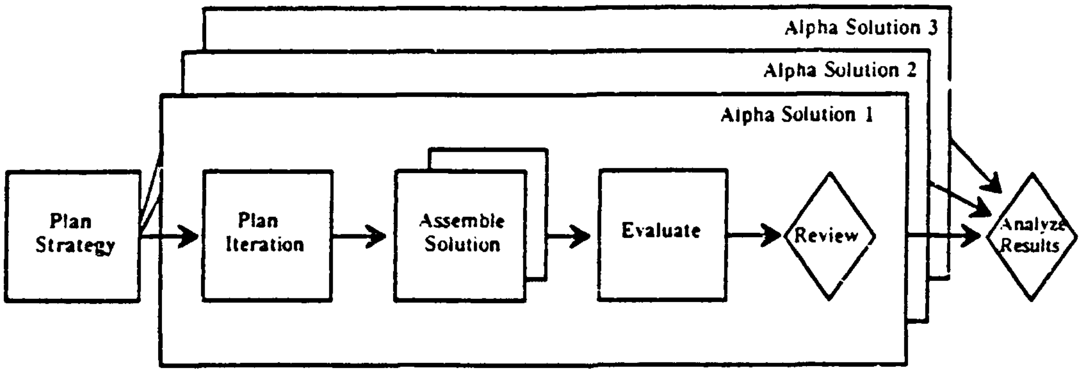
The IBM STARS SEE is a combination of hardware platforms and software tools which support Ada software development from requirements analysis through code generation, testing and maintenance. The SEE is adaptable, i.e., it is tailorable to a "SEE solution" which meets the specific needs of a project based upon the project profile and product maturity (SEE solutions are also assembled to evaluate solutions to particular problems.).

THE IBM STARS SEE is based on IBM's AIX Technical CASE Solutions and incorporates value add efforts from STARS (especially in the areas of process and reuse). Particular SEE solutions may incorporate project specific capabilities.

To date, SEE solutions have been assembled from COTS products and STARS prototypes providing a loosely-integrated environment. The goal is to evolve the SEE solutions into framework-based, integrated, COTS supplied SEE solutions. The IBM STARS Team plans to support the incremental insertion of presentation, control, data and process integration capabilities and their use in assembling SEE solutions. The IBM STARS Team places its SEE solutions into active project use to identify, validate and refine environment capabilities.

IBM STARS SEE EVOLUTION STRATEGY

EVOLUTION PROCESS



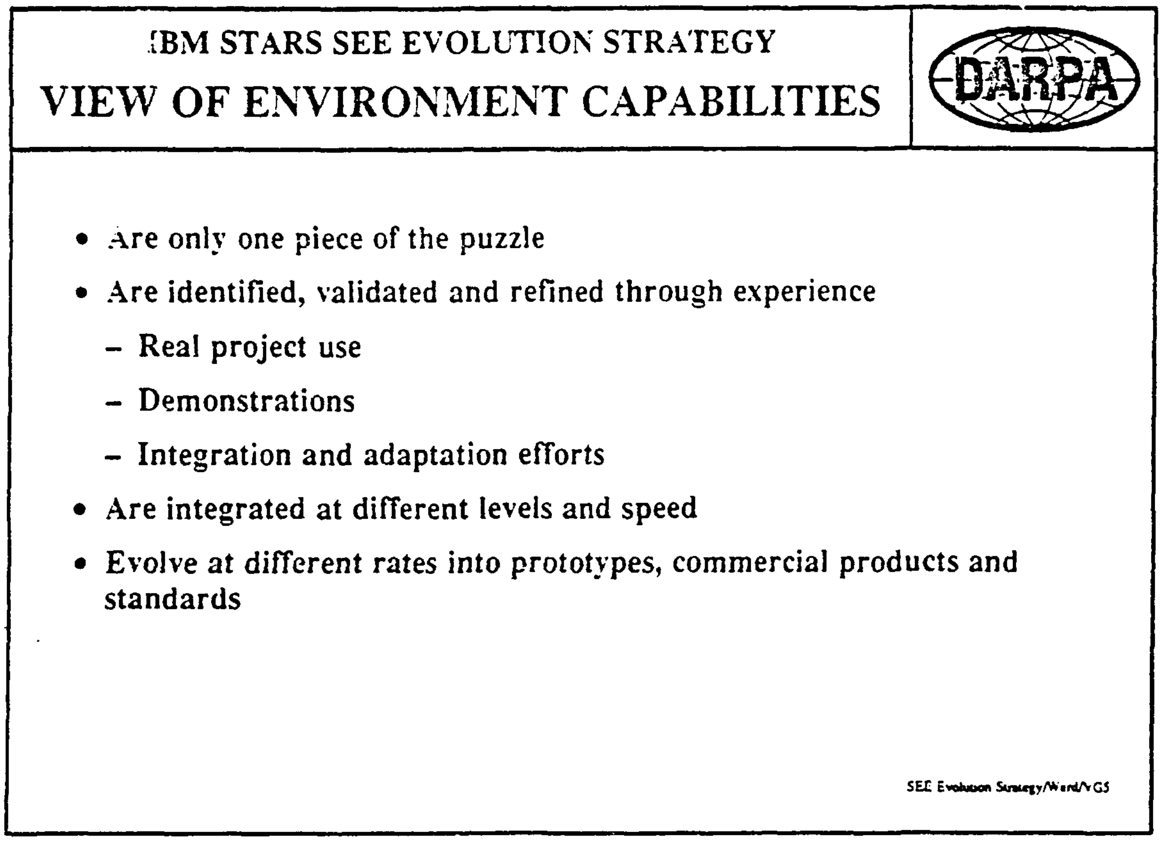
SEE Evolution Strategy/Ward/VG4

Integrating SEE solutions and inserting the technology into real projects requires effort. This effort must be applied over time to take advantage of new developments and experiences gained through active use of the SEE.

The IBM STARS Team uses a concurrent, yet iterative approach to evolve its environment capabilities. It is concurrent in the sense that multiple SEE solutions may be assembled and evaluated in parallel. It is iterative in the sense that these solutions evolve over time incorporating enhanced capabilities, new technologies and user feedback.

In planning, the IBM STARS Team outlines its objectives and evaluation criteria. A solution, (or multiple solutions) is assembled based on the goals and objectives. The solution also takes into account project needs, new technologies, product maturity, standards conformance etc.... The solution is demonstrated and/or placed into active project use. Feedback is captured and the solution is evaluated against evaluation criteria. The IBM STARS Team provides results to STARS, IBM commercial, CASE vendors and other technology organizations.

This approach (including the interim SEE solution and experience) will enable us to evolve today's environment capabilities to a framework-based environment supporting a process-driven, reuse-based engineering approach to megaprogramming for the Demonstration Projects. We will also use this approach to assemble and evolve the Demonstration Project SEE solution.



:BM STARS SEE EVOLUTION STRATEGY

VIEW OF ENVIRONMENT CAPABILITIES

DARPA

- Are only one piece of the puzzle
- Are identified, validated and refined through experience
  - Real project use
  - Demonstrations
  - Integration and adaptation efforts
- Are integrated at different levels and speed
- Evolve at different rates into prototypes, commercial products and standards

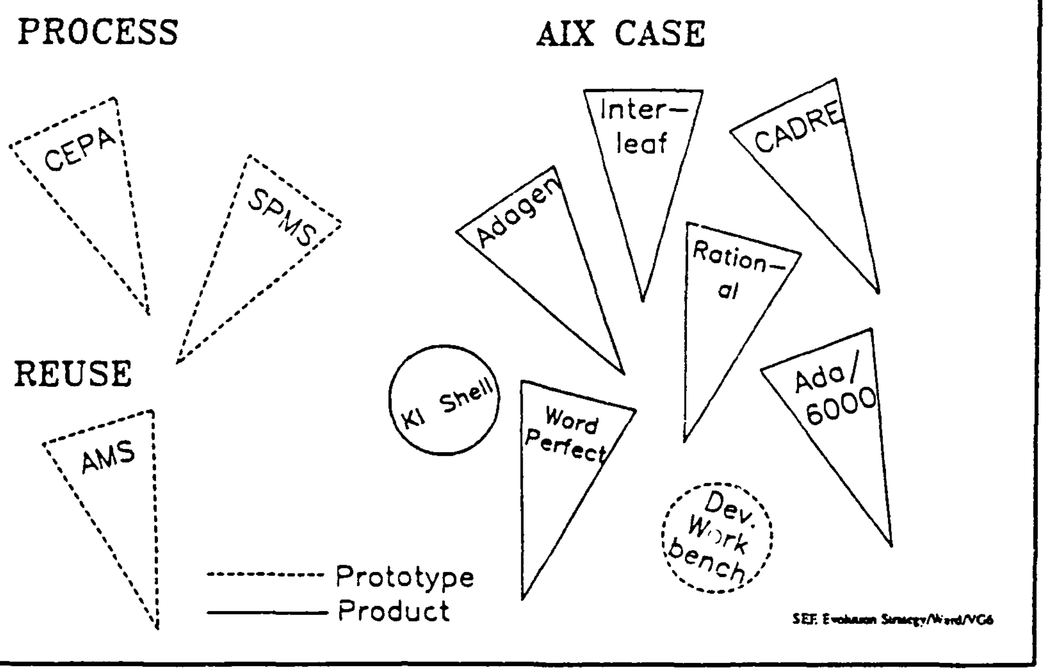
SEE Evolution Strategy/Word/VGS

The IBM STARS Team acknowledges the following as important observations and premises by which we work:

- environment capabilities are the technology support for the people and the processes and methods that they use to develop systems;
- environment capabilities are identified, validated and refined through experience. This experience is gathered from real project use, demonstrations and integration and adaptation efforts;
- environment capabilities can be integrated at different levels and still be effective. We recognize the its possible, and even desirable to have different tools integrated at varying levels; and
- environment capabilities evolve at different rates into prototypes, commercial products, and standards. Gone are the monolithic environments with all capabilities at equal development maturation.

IBM STARS SEE EVOLUTION STRATEGY

EXAMPLE CURRENT ELEMENTS

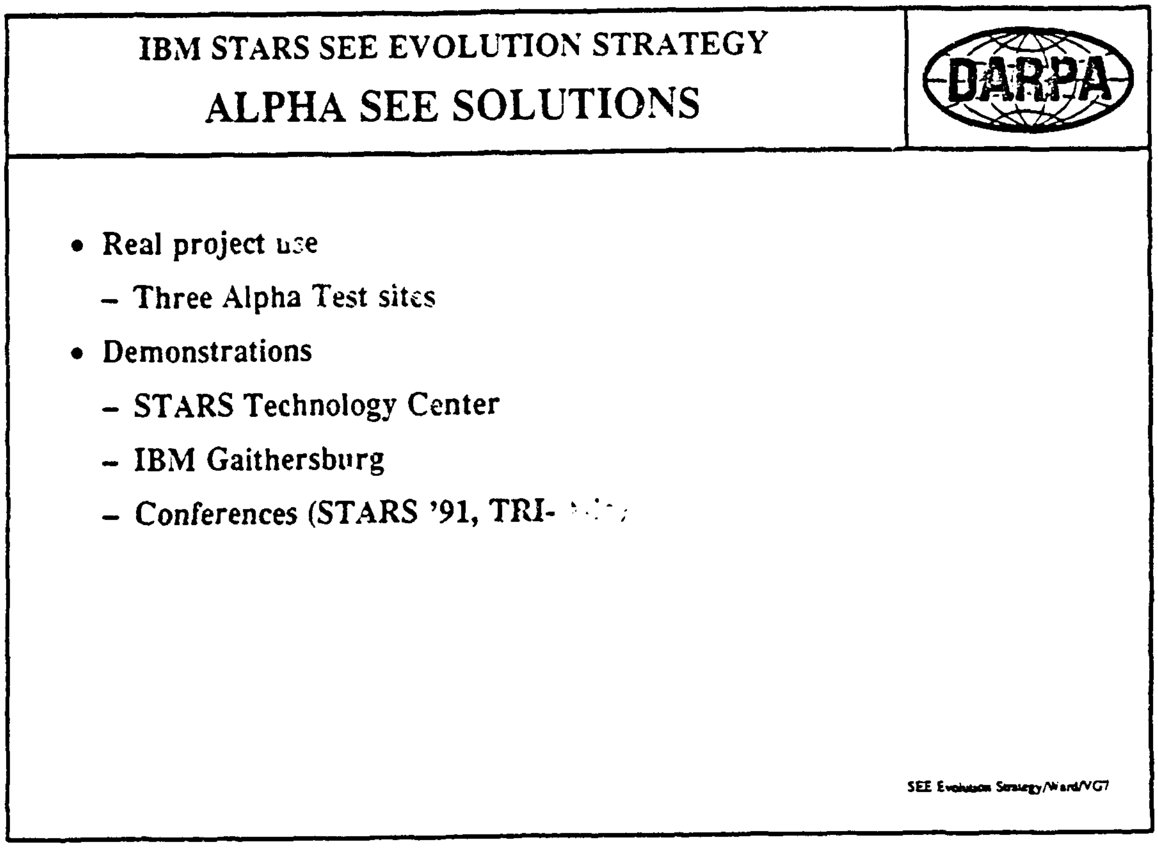


This diagram depicts example elements from the IBM STARS SEE. Note that the circles represent integration frameworks and the triangles represent technology support tools. The process and reuse support tools are prototypes developed by the IBM STARS Team and are described below. The IBM STARS SEE elements are assembled to form SEE solutions.

Cleanroom Engineering Process Assistant (CEPA) is a prototype that automates a portion of the Cleanroom process which allows engineers to follow the Cleanroom process more easily and effectively. Cleanroom uses a formal approach of specifying a pipeline of small, user executable increments, utilizing rigorous practices to create software that is of extremely high quality and using statistical quality control methods to certify the correctness of software. CEPA was implemented using KI Shell, an environment for implementing process models as an executable process–driven application.

Software Process Management System (SPMS) is a prototype tool set to support software process engineering in the analysis, modeling, design, development, evolution and support of organizational and project–specific software development process models. A process model represents the generic sequence of tasks, milestones, constraints and products necessary to produce a specific type of product.

Asset Management System (AMS) is an experimental reuse library mechanism supporting librarian activities, (for example, the definition and maintenance of classification schemes and the cataloging of assets) and subscriber activities (for example, searching, retrieving and browsing assets). AMS supports classification schemes defined by the IBM/SAIC Domain Analysis Process Model, in support of domain–specific software reuse. AMS is designed to run on multiple platforms and operate with other reuse library mechanisms and cooperating tools across a wide area network.



IBM STARS SEE EVOLUTION STRATEGY

ALPHA SEE SOLUTIONS

- Real project use
  - Three Alpha Test sites
- Demonstrations
  - STARS Technology Center
  - IBM Gaithersburg
  - Conferences (STARS '91, TRI-Ada)

SEE Evolution Strategy/Ward/VG7

In preparation for instantiating SEE solutions for the Demonstration Projects, the IBM STARS team established an "alpha test" subtask. The purpose of this subtask is as follows:

- learn how to deal with the cultural and technical barriers to transition to framework-based environments incorporating reuse and process support;
- gain early experience and feedback in the use of the SEE solutions;
- provide a vehicle for technology transfer; and
- be a precursor for the STARS Demonstration Activity in defining preliminary project selection criteria, how to support projects in using and evaluating a SEE, and how to capture information from evaluation projects.

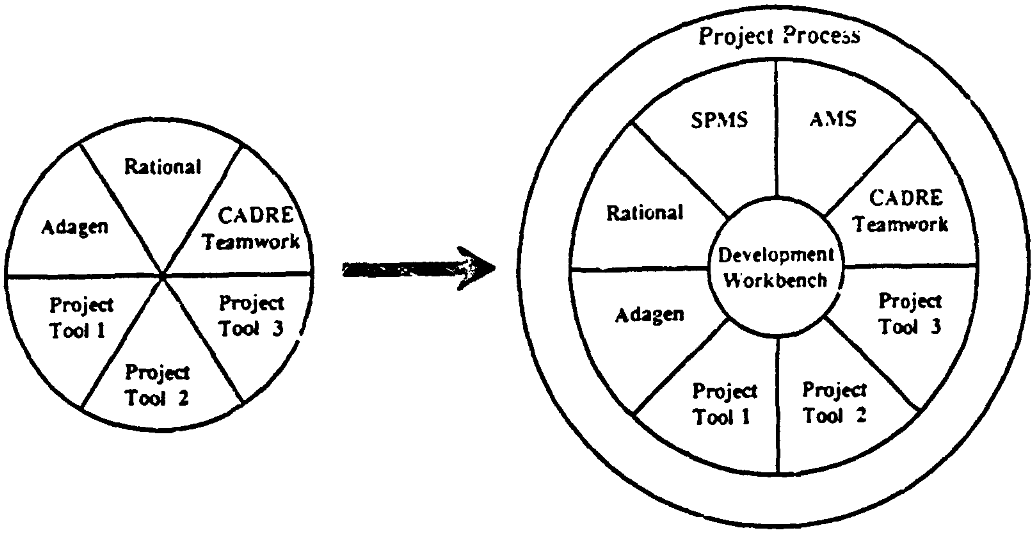
Currently we have three active alpha test projects. These projects are using, or are planning to use, a wide range of SEE tools and methods covering the system life cycle. However, to provide some focus to the alpha test efforts, each project was asked to concentrate on a specific aspect of the lifecycle (currently: software design, requirements traceability, reverse engineering and reusability). The plan is to increase the scope of the alpha test activity in 1992 by introducing frameworks, and process and reuse mechanisms into the alpha test projects.

The alpha test projects provide us with valuable feedback on our alpha SEE solutions. In addition, the IBM STARS Team actively participates in conferences such as STARS '91 and TRI-Ada '91 to demonstrate our solutions to a wider audience. IBM STARS SEE demonstration sites also include the STARS Technology Center and the IBM Gaithersburg facility.

The reader is referred to CDRL 03032, SEE Technical Report, for a complete discussion of our alpha SEE solutions, our alpha test projects and our lessons learned to date.

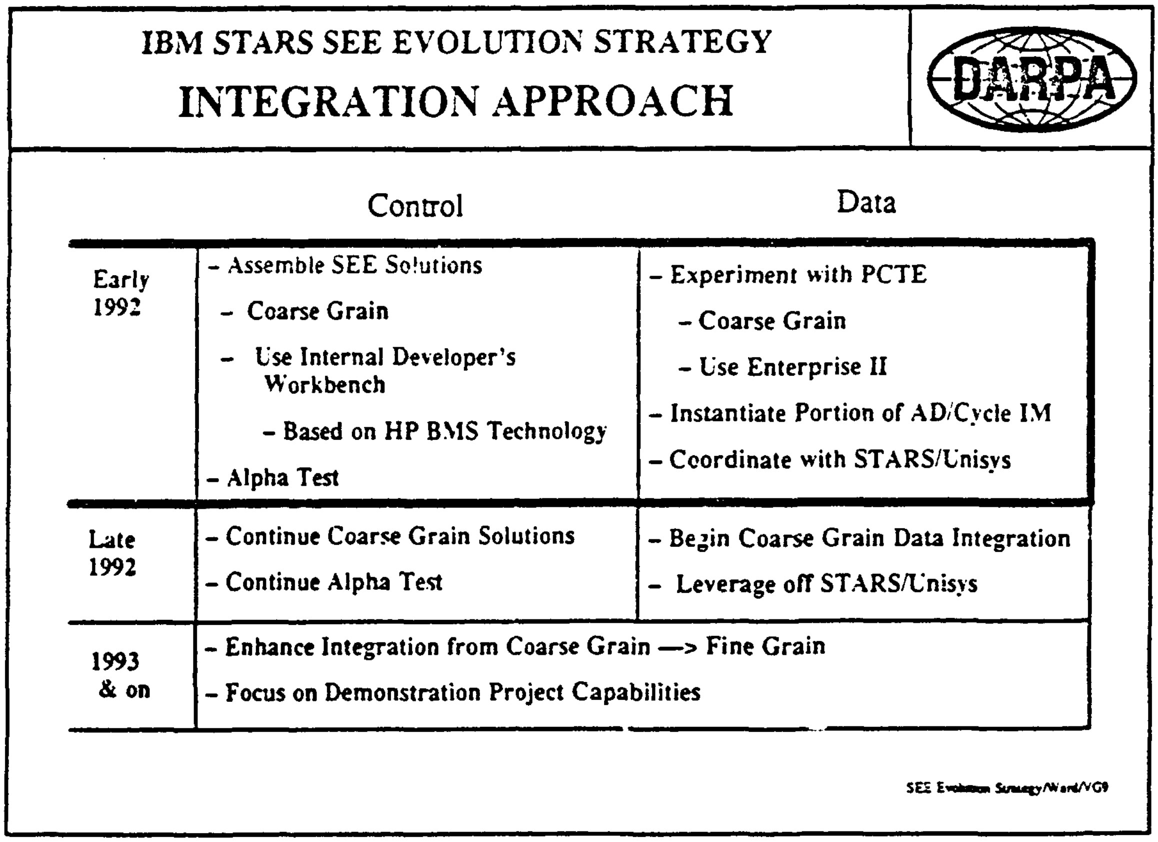
IBM STARS SEE EVOLUTION STRATEGY

EXAMPLE SOLUTION



See Evolution Strategy/Ward/VG8

The chart depicts an example of one of our planned activities during 1992. We plan to migrate one of our alpha test projects from their loosely-integrated tool set (represented by the left-hand-side of the picture) to a framework-based environment that incorporates reuse and process capabilities (represented by the right-hand-side of the picture). Not all of the capabilities will be integrated at the same level, and not all of the reuse and process capabilities will be exercised. The migration will be scoped by a defined set of goals and objectives.



IBM STARS SEE EVOLUTION STRATEGY

INTEGRATION APPROACH

DARPA

	Control	Data
Early 1992	- Assemble SEE Solutions - Coarse Grain - Use Internal Developer's Workbench - Based on HP BMS Technology - Alpha Test	- Experiment with PCTE - Coarse Grain - Use Enterprise II - Instantiate Portion of AD/Cycle IM - Coordinate with STARS/Unisys
Late 1992	- Continue Coarse Grain Solutions - Continue Alpha Test	- Begin Coarse Grain Data Integration - Leverage off STARS/Unisys
1993 & on	- Enhance Integration from Coarse Grain —> Fine Grain - Focus on Demonstration Project Capabilities	

SEE Evolution Strategy/Ward/VG9

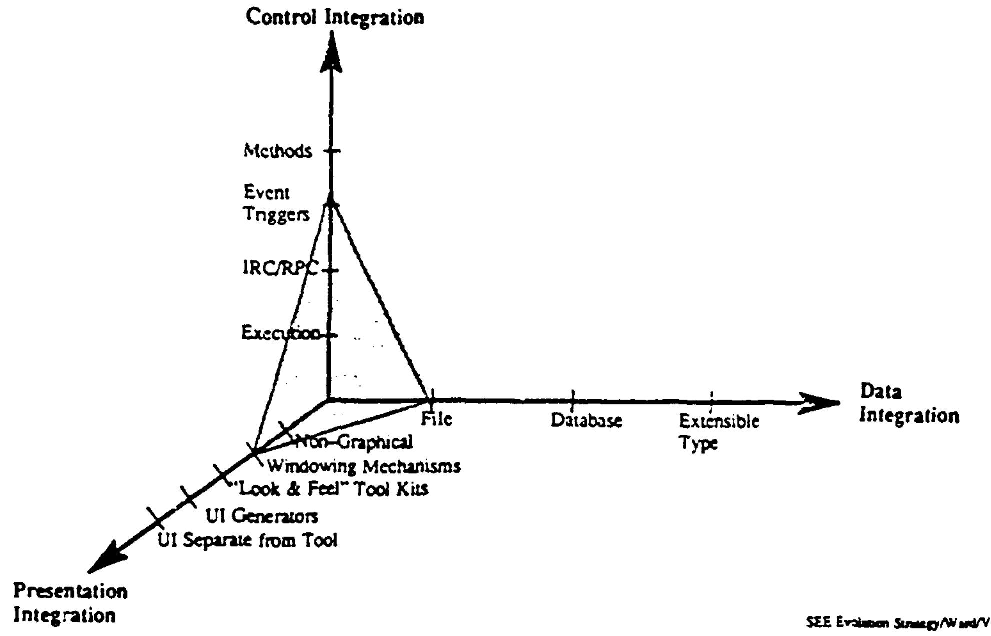
The next six charts outline the approach of the IBM STARS Team for incorporating control and data integration capabilities and their use in the SEE solutions. The IBM STARS Team will not actively focus on presentation integration issues aside from the support provided by the framework.

During early 1992, the IBM STARS Team will assemble SEE solutions using coarse grain control integration capabilities based on HP Broadcast Message Technology. HP Broadcast Message Technology supports close communication of independent tools, allowing the tools to communicate in a networked, heterogeneous environment. Message requests allow one tool to invoke the functionality of another tool and notification messages allow tools (or the user) to define triggers that respond to events and initiate other actions. The IBM STARS Team will use an internal developers workbench prototype as its integration framework. We will demonstrate and 'alpha test' the resulting SEE solutions.

In parallel, to augment our primary focus on control integration, we plan to experiment with coarse grain data integration using the Enterprise II framework. Enterprise II provides PCTE data integration services (currently at the coarse grain level). We plan to coordinate our efforts with the Unisys STARS Team in the data integration area, as their resources will initially focus on data integration. Our intent is to focus on the information model and bring our AD/Cycle experience in this area to the STARS Team.

PRESENTATION TITLE

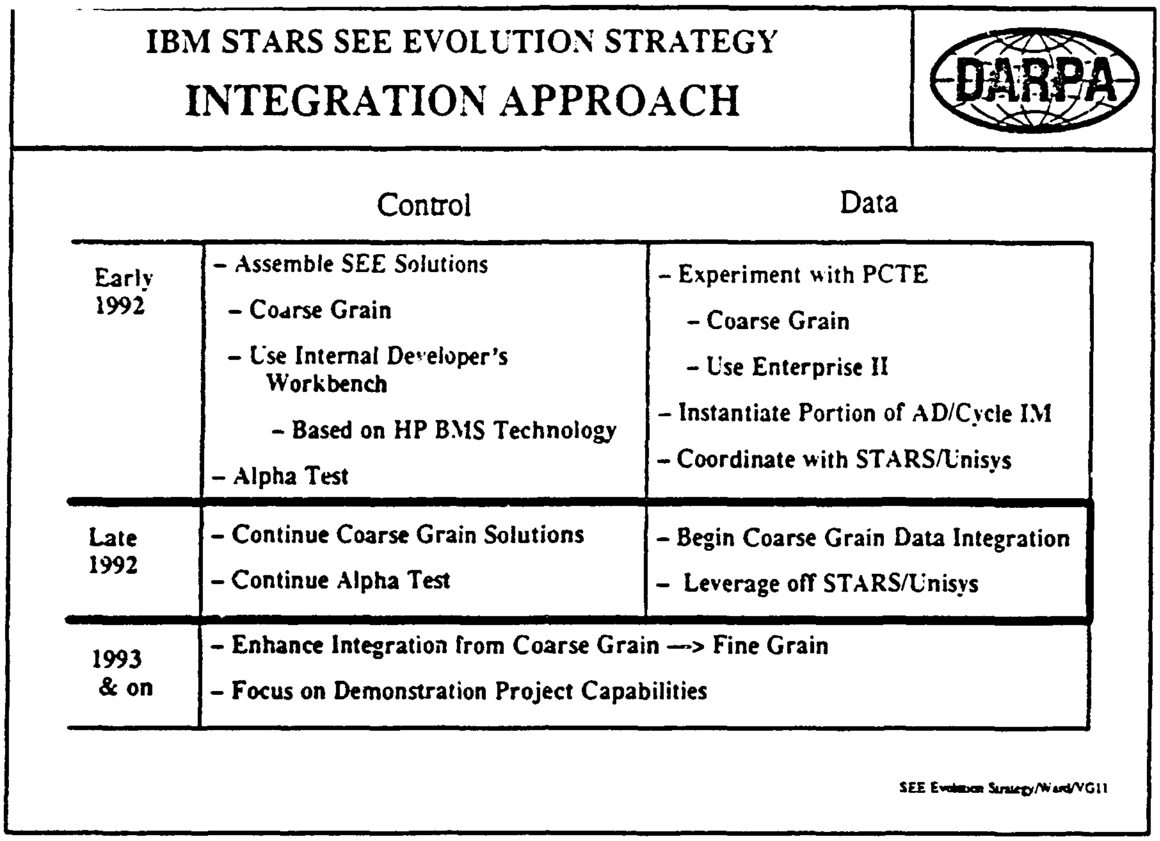
DIMENSIONS OF INTEGRATION EARLY 1992



SEE Evolution Strategy/Ward/VG10

This picture graphically represents the planned integration capabilities of the IBM STARS SEE in early 1992 as described in the previous chart.

Note that the measure on each axis is the 'granularity' of the data, control or presentation integration capabilities.



IBM STARS SEE EVOLUTION STRATEGY

INTEGRATION APPROACH

DARPA

	Control	Data
Early 1992	- Assemble SEE Solutions - Coarse Grain - Use Internal Developer's Workbench - Based on HP BMS Technology - Alpha Test	- Experiment with PCTE - Coarse Grain - Use Enterprise II - Instantiate Portion of AD/Cycle IM - Coordinate with STARS/Unisys
Late 1992	- Continue Coarse Grain Solutions - Continue Alpha Test	- Begin Coarse Grain Data Integration - Leverage off STARS/Unisys
1993 & on	- Enhance Integration from Coarse Grain —> Fine Grain - Focus on Demonstration Project Capabilities	

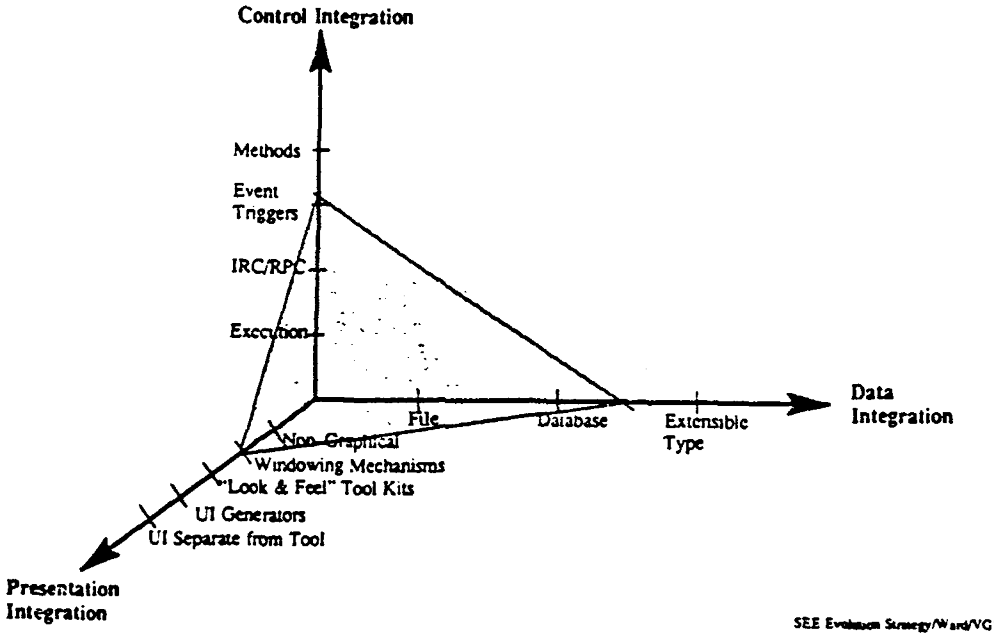
SEE Evolution Strategy/Ward/VG11

As 1992 progresses we plan to continue our focus on coarse grain control integration and begin balancing this strategy with more emphasis on providing and using coarse grain data integration services in the SEE solutions. We plan to rely heavily on the Unisys STARS Team and leverage off of their work in this area.

Solutions assembled using both control and data integration services will be demonstrated and placed into real projects for active use. Selection of the underlying framework(s) will be based on availability and prior experience gained by the IBM and Unisys STARS Teams.

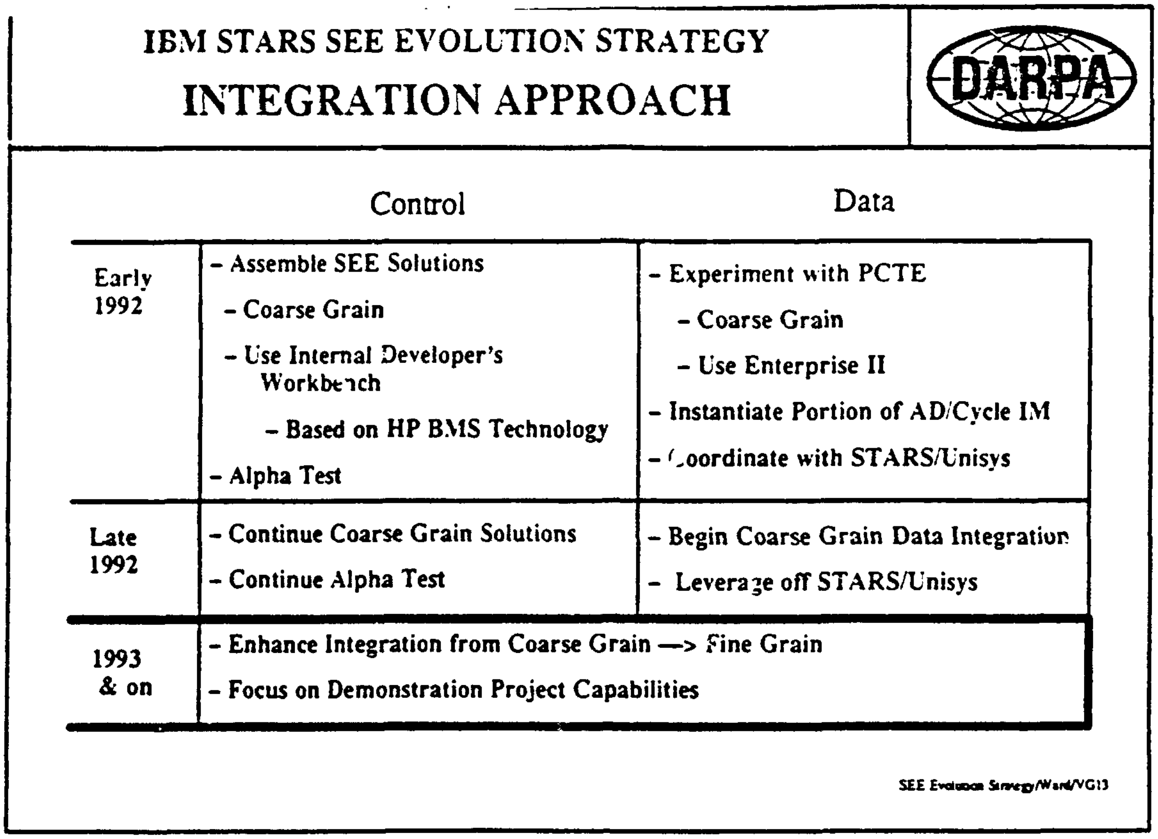
PRESENTATION TITLE

DIMENSIONS OF INTEGRATION LATE 1992



SEE Evolution Strategy/Ward/VG12

This picture graphically represents the planned integration capabilities of the IBM STARS SEE in late 1992 as described in the previous chart.



IBM STARS SEE EVOLUTION STRATEGY

INTEGRATION APPROACH

DARPA

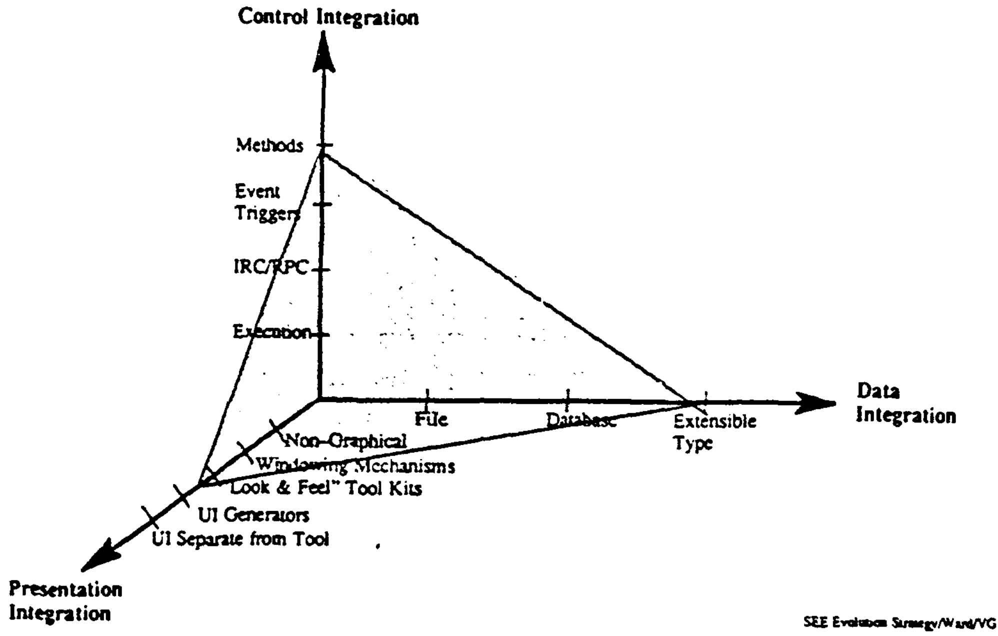
	Control	Data
Early 1992	- Assemble SEE Solutions - Coarse Grain - Use Internal Developer's Workbench - Based on HP BMS Technology - Alpha Test	- Experiment with PCTE - Coarse Grain - Use Enterprise II - Instantiate Portion of AD/Cycle IM - Coordinate with STARS/Unisys
Late 1992	- Continue Coarse Grain Solutions - Continue Alpha Test	- Begin Coarse Grain Data Integration - Leverage off STARS/Unisys
1993 & on	- Enhance Integration from Coarse Grain —> Fine Grain - Focus on Demonstration Project Capabilities	

SEE Evolution Strategy/Ward/VG13

During 1993 and throughout the demonstration time-frame, the IBM STARS Team plans to enhance the capabilities of the SEE solutions. In particular we plan to evolve from coarse grain control and data integration to fine grain. Heavy emphasis will be placed on supporting the Demonstration Project with a tailored, adaptable SEE solution and supporting work products such as training, and integration guidelines. We will support, adapt and evolve the Demonstration See solution over the lifetime of the demonstration.

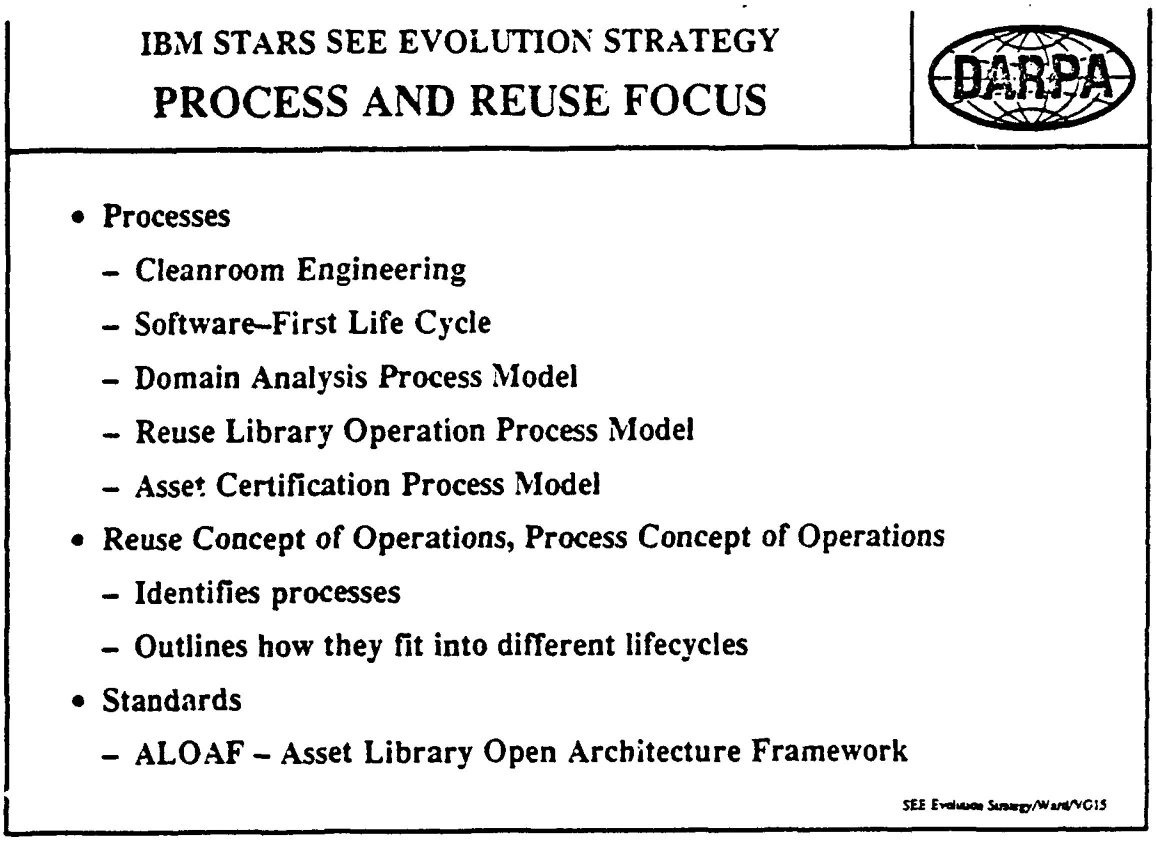
PRESENTATION TITLE

DIMENSIONS OF INTEGRATION 1993 AND ON



SEE Evolution Strategy/Ward/VG14

This picture graphically represents the planned integration capabilities of the IBM STARS SEE in 1993 and beyond as described in the previous chart.



IBM STARS SEE EVOLUTION STRATEGY

PROCESS AND REUSE FOCUS

- Processes
  - Cleanroom Engineering
  - Software–First Life Cycle
  - Domain Analysis Process Model
  - Reuse Library Operation Process Model
  - Asset Certification Process Model
- Reuse Concept of Operations, Process Concept of Operations
  - Identifies processes
  - Outlines how they fit into different lifecycles
- Standards
  - ALOAF – Asset Library Open Architecture Framework

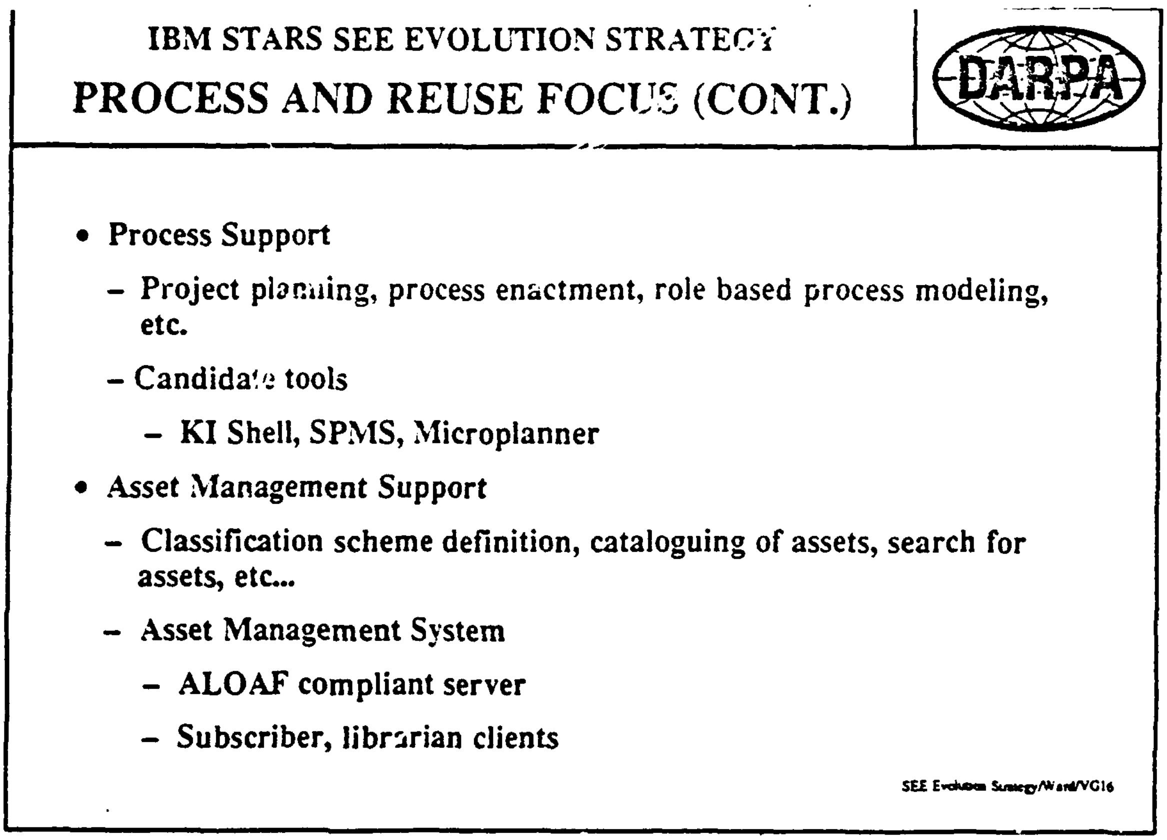
SEE Evolution Strategy/Ward/VG15

The next two charts highlight the IBM STARS Team efforts in the process and reuse areas in preparation for the Demonstration Projects. The first chart focuses on our efforts in identifying and defining processes and standards. The second chart outlines process and reuse mechanisms identified for incorporation into the IBM STARS SEE solutions and further evolution of their capabilities.

The joint activity groups (one member from each STARS prime, a STARS prime system architect and other members) for process and reuse focus heavily on identifying and defining the processes to support megaprogramming. Key processes developed or in development by the IBM STARS Team include:

- Cleanroom Engineering Process – process for creating software that is of extremely high quality and uses statistical quality control methods to certify the correctness of the software.
- IBM STARS Team Software–First Life Cycle – An IBM STARS Team instantiation of the class of life cycles known as Software–First Life Cycles.
- Domain Analysis Process Model – process for developing generic models and architectures that support multiple products across and application domain.
- Reuse Library Operation Process Model – process for operating an asset management system.
- Asset Certification Process Model – process for certifying assets in an asset management system.

Another important effort by these groups is the participation in defining an Asset Library Open Architecture Framework (ALOAF). The goal is to ensure seamless interoperablity between different reuse libraries on different platforms with possibly different data models.



IBM STARS SEE EVOLUTION STRATEGY

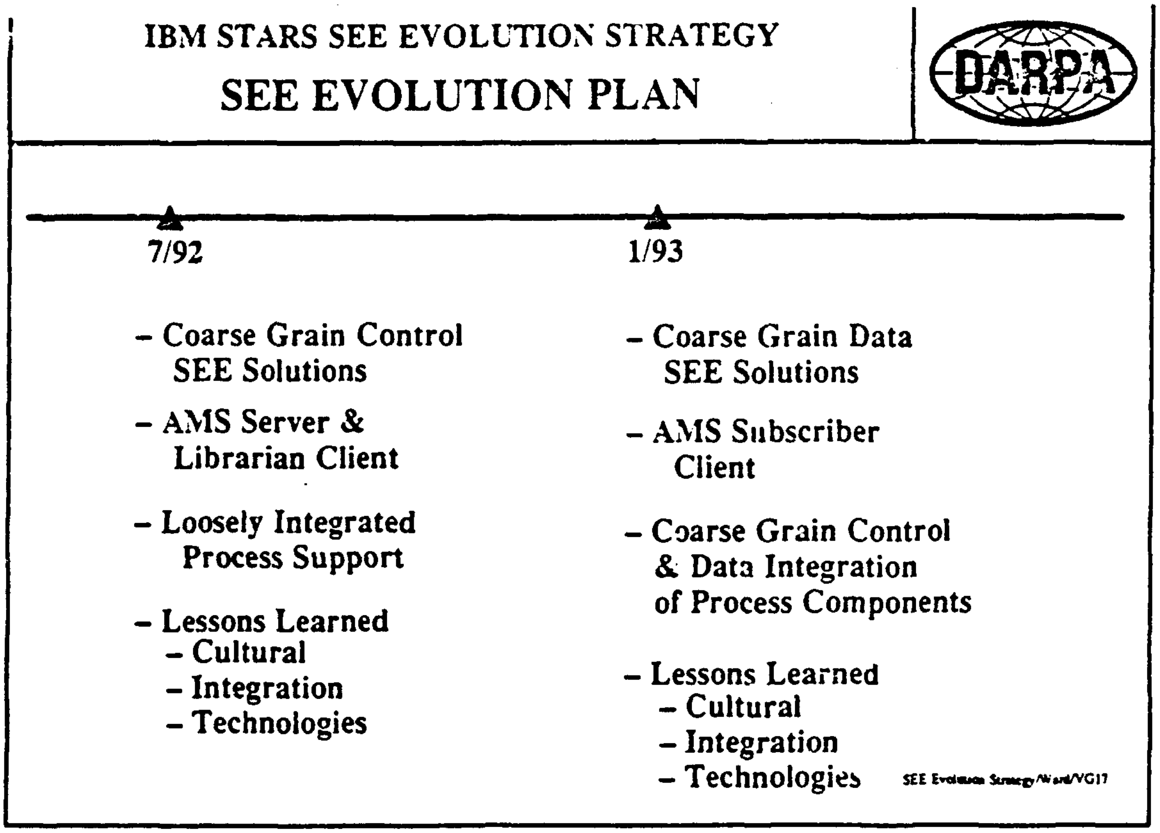
PROCESS AND REUSE FOCUS (CONT.)

- Process Support
  - Project planning, process enactment, role based process modeling, etc.
  - Candidate tools
    - KI Shell, SPMS, Microplanner
- Asset Management Support
  - Classification scheme definition, cataloguing of assets, search for assets, etc...
  - Asset Management System
    - ALOAF compliant server
    - Subscriber, librarian clients

SEE Evolution Strategy/Ward/VG16

To provide process capabilities in a SEE, one must provide support for process activities such as project planning, process enactment and role-based process modeling. The IBM STARS Team has identified a suite of candidate tools which cover the wide-range of process activities. Currently these tools are loosely integrated, but plans exist to evolve the tool capabilities and integration levels for better process support in the SEE solutions. The reader is referred to CDRL item 03705, Software Process Tools and Techniques Evaluation Report, for a complete discussion of the IBM STARS Team process capabilities and plans.

To provide asset management capabilities in a SEE, one must provide support for asset management activities such as classification scheme definition, asset cataloging, and searching. The IBM STARS Team has developed a prototype called Asset Management System (described previously) to provide asset management support in the SEE solutions. AMS will have an ALOAF compliant server and we plan to develop subscriber and librarian clients that use the server. Currently, AMS is in prototype form but SAIC and SPS have recently announced their intent to commercialize it. AMS will evolve to support enhanced data and control integration capabilities to support asset management in the SEE solutions.



IBM STARS SEE EVOLUTION STRATEGY

SEE EVOLUTION PLAN

DARPA

7/92

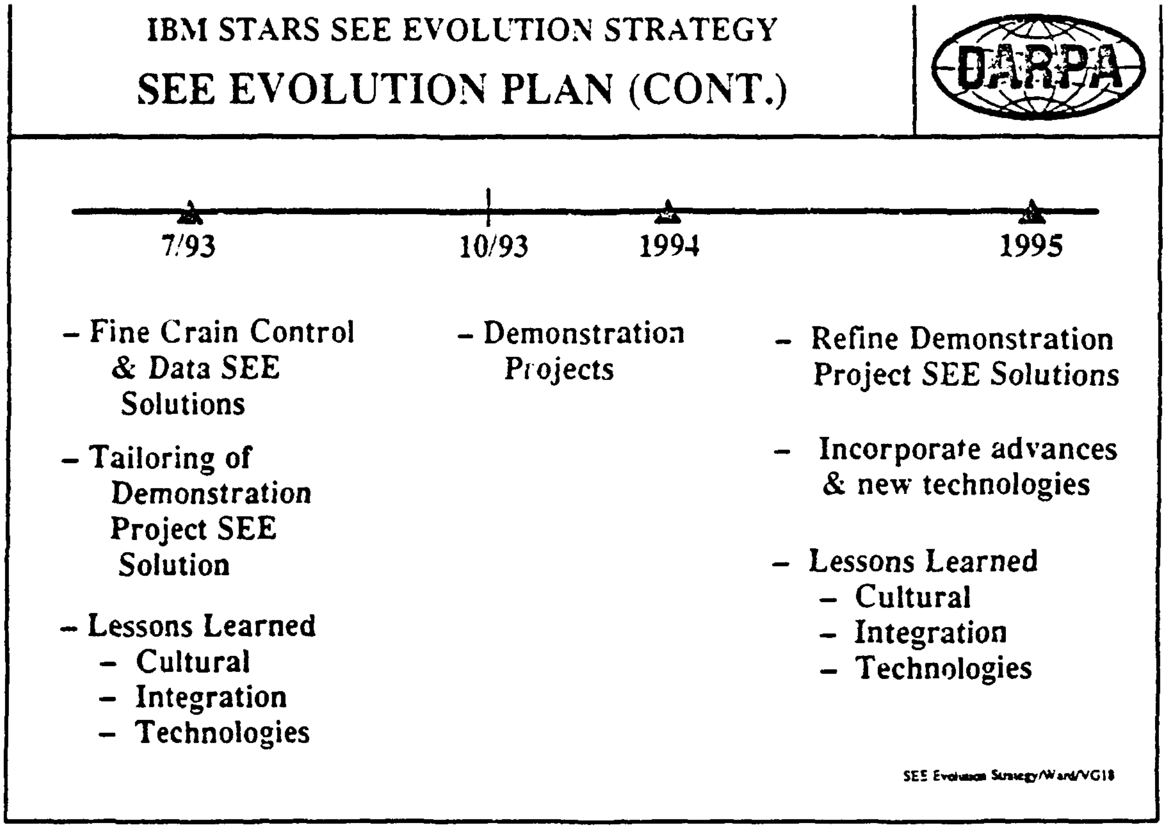
- Coarse Grain Control SEE Solutions
- AMS Server & Librarian Client
- Loosely Integrated Process Support
- Lessons Learned
  - Cultural
  - Integration
  - Technologies

1/93

- Coarse Grain Data SEE Solutions
- AMS Subscriber Client
- Coarse Grain Control & Data Integration of Process Components
- Lessons Learned
  - Cultural
  - Integration
  - Technologies

SEE Evolution Strategy/Ward/VG17

The next two charts recap the IBM STARS SEE evolution strategy and depict the expected products and major milestones. By 7/92 we expect to be able to demonstrate SEE solutions based on control integration capabilities and by 1/93 we expect to expand the integration capabilities to include coarse grain data integration. We expect the AMS Server and initial Librarian Client to be completed by 7/93 and the Subscriber Client shortly thereafter. By 7/92 we will still have a loosely integrated process tool suite, but this tool suite will be upgraded with both coarse grain control and data integration capabilities for incorporation into the SEE solutions by 1/93. Both the reuse and process capabilities will be placed in actual project use as soon as possible. We expect to gain lessons learned on cultural impacts, integration issues and new technologies.



IBM STARS SEE EVOLUTION STRATEGY

SEE EVOLUTION PLAN (CONT.)

7/93 — 10/93 — 1994 — 1995

7/93:

- Fine Crain Control & Data SEE Solutions
- Tailoring of Demonstration Project SEE Solution
- Lessons Learned
  - Cultural
  - Integration
  - Technologies

10/93:

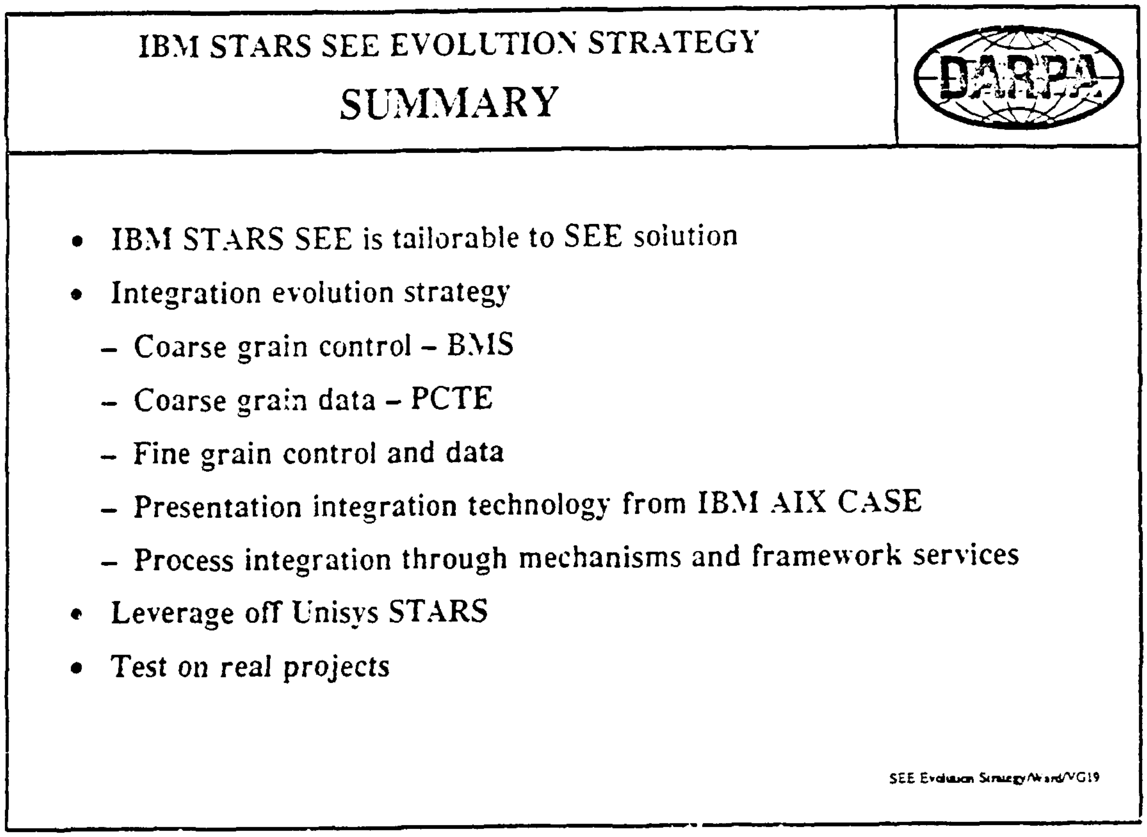
- Demonstration Projects

1994–1995:

- Refine Demonstration Project SEE Solutions
- Incorporate advances & new technologies
- Lessons Learned
  - Cultural
  - Integration
  - Technologies

SEE Evolution Strategy/Ward/VG18

10/93 marks the onset of the Demonstration Projects. We expect to work with them prior to this date to set goals, and objectives, understand and influence their processes, and to begin SEE solution tailoring and training. We will support the Demonstration Project, and adapt and evolve their SEE solution throughout the demonstration time frame. We will continue to focus on capturing lessons learned and developing support products such as integration guidelines for use by others who plan to transition to Megaprogramming.



IBM STARS SEE EVOLUTION STRATEGY

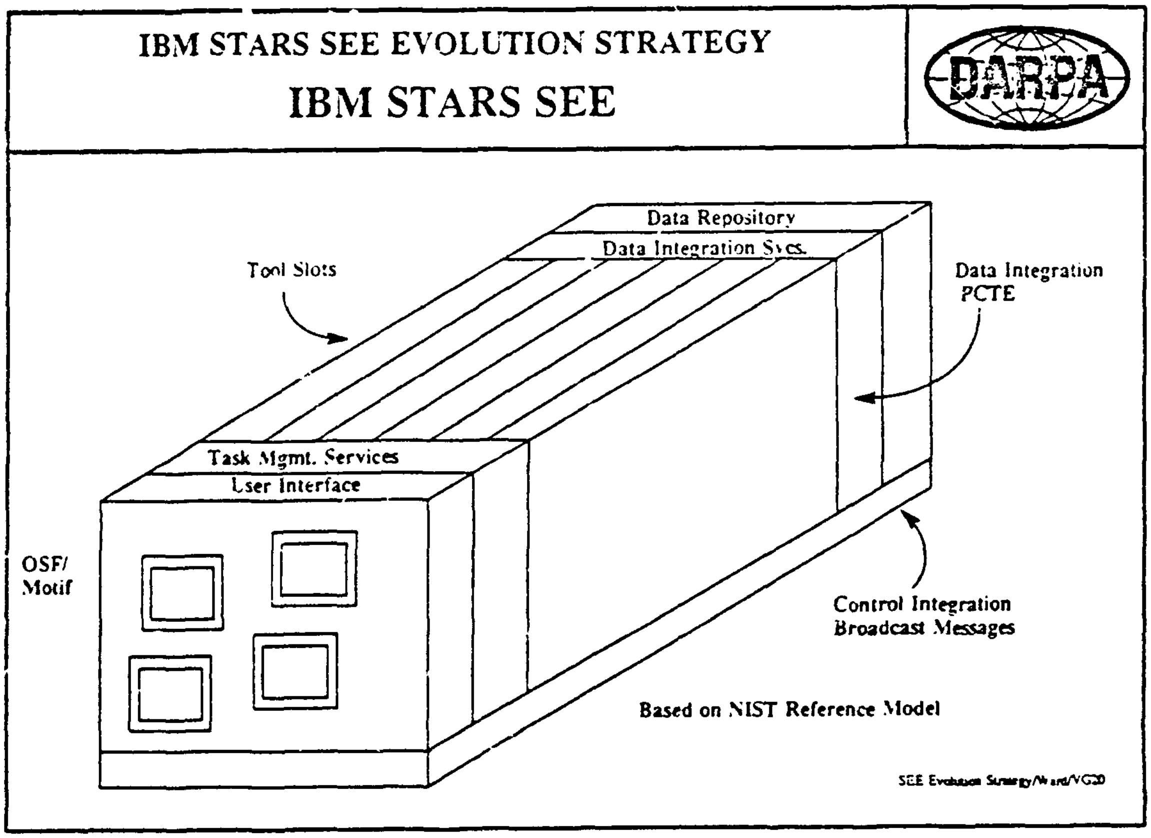
SUMMARY

- IBM STARS SEE is tailorable to SEE solution
- Integration evolution strategy
  - Coarse grain control – BMS
  - Coarse grain data – PCTE
  - Fine grain control and data
  - Presentation integration technology from IBM AIX CASE
  - Process integration through mechanisms and framework services
- Leverage off Unisys STARS
- Test on real projects

SEE Evolution Strategy/Ward/VG19

In summary, the IBM STARS Team plans to incrementally insert and use presentation, process, control and data integration capabilities in our SEE solutions, advancing from coarse grain to fine grain support and use. The IBM STARS Team and the Unisys STARS Team evolution strategies complement each other and we plan to leverage off each other's experience.

The IBM STARS SEE solutions will be placed in active projects to obtain feedback and experience on the cultural and technological obstacles in transitioning to a framework-based environment supporting a process-driven, reuse-based engineering approach to megaprogramming.



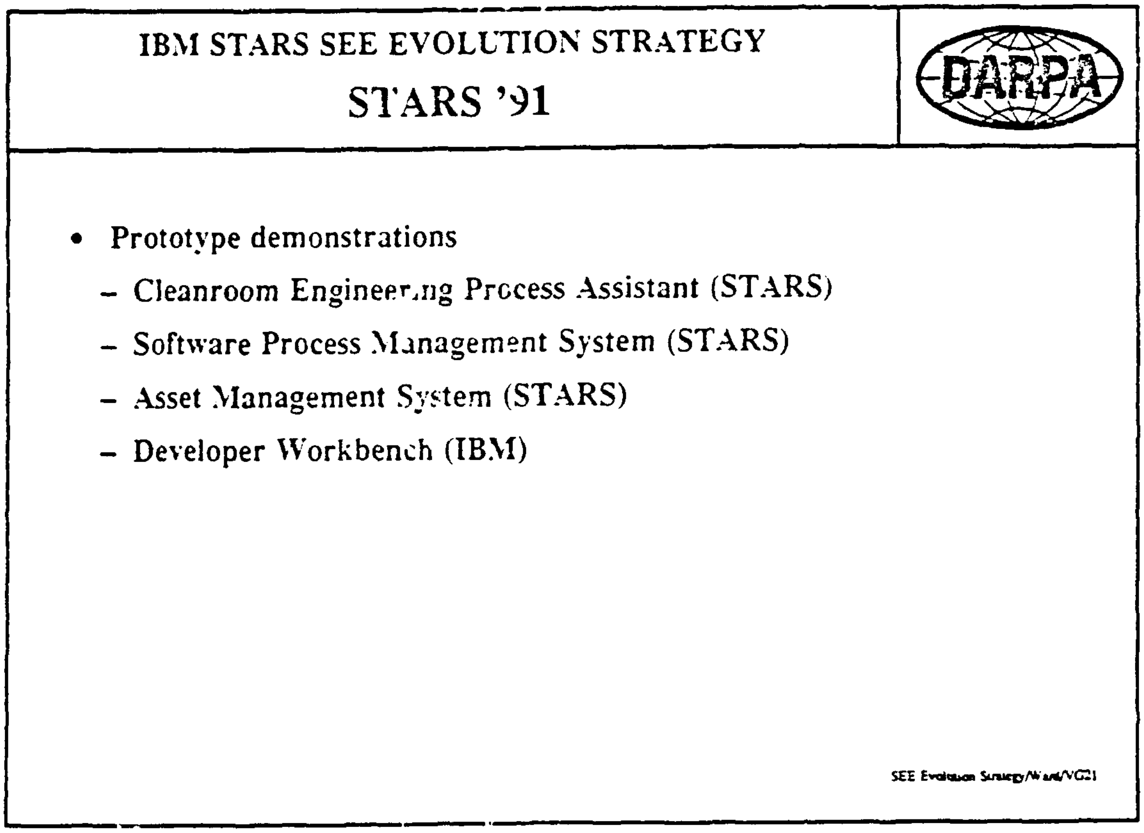
IBM STARS SEE EVOLUTION STRATEGY

IBM STARS SEE

Based on NIST Reference Model

SEE Evolution Strategy/Ward/VG20

This diagram maps the planned IBM STARS SEE integration services against the NIST Reference Model. The plans are for the IBM STARS SEE to use PCTE for data integration, BMS for control integration and OSF Motif or its successor for presentation integration.



IBM STARS SEE EVOLUTION STRATEGY
STARS '91

- Prototype demonstrations
  - Cleanroom Engineering Process Assistant (STARS)
  - Software Process Management System (STARS)
  - Asset Management System (STARS)
  - Developer Workbench (IBM)

SEE Evolution Strategy/Ward/VG21

The IBM STARS Team and the IBM Corporation exhibits at STAR '91 will include demonstrations of the following prototypes: the Cleanroom Engineering Process Assistant (CEPA), the Software Process Management System (SPMS), the Asset Management System (AMS) and the Developer Workbench. CEPA, SPMS and AMS have been previously described. The Developer Workbench is a technology demonstration of an internal development workbench that uses BMS technology for control integration of developer's tools including editing and configuration management capabilities.

ACRONYMS

AIX	Advanced Interactive Executive
AMS	Asset Management System
BMS	Broadcast Message Server
CASE	Computer Aided Software Engineering
CEPA	Cleanroom Engineering Process Assistant
HP	Hewlett Packard
IBM	International Business Machines Corporation
IM	Information Model
NIST	National Institute of Standards and Technology
PCTE	Portable Common Tool Environment
SAIC	Science Applications International Corporation
SEE	Software Engineering Environment
SPMS	Software Process Management System
SPS	Software Productivity Solutions, Inc.
STARS	Software Technology for Adaptable, Reliable Systems
UI	User Interface

Product trademarks referred to in this presentation.

Adagen is a trademark of Mark V

AIX is trademark of the IBM Corporation

Interleaf is a trademark of Interleaf

KI Shell is a trademark of Universal Energy Systems

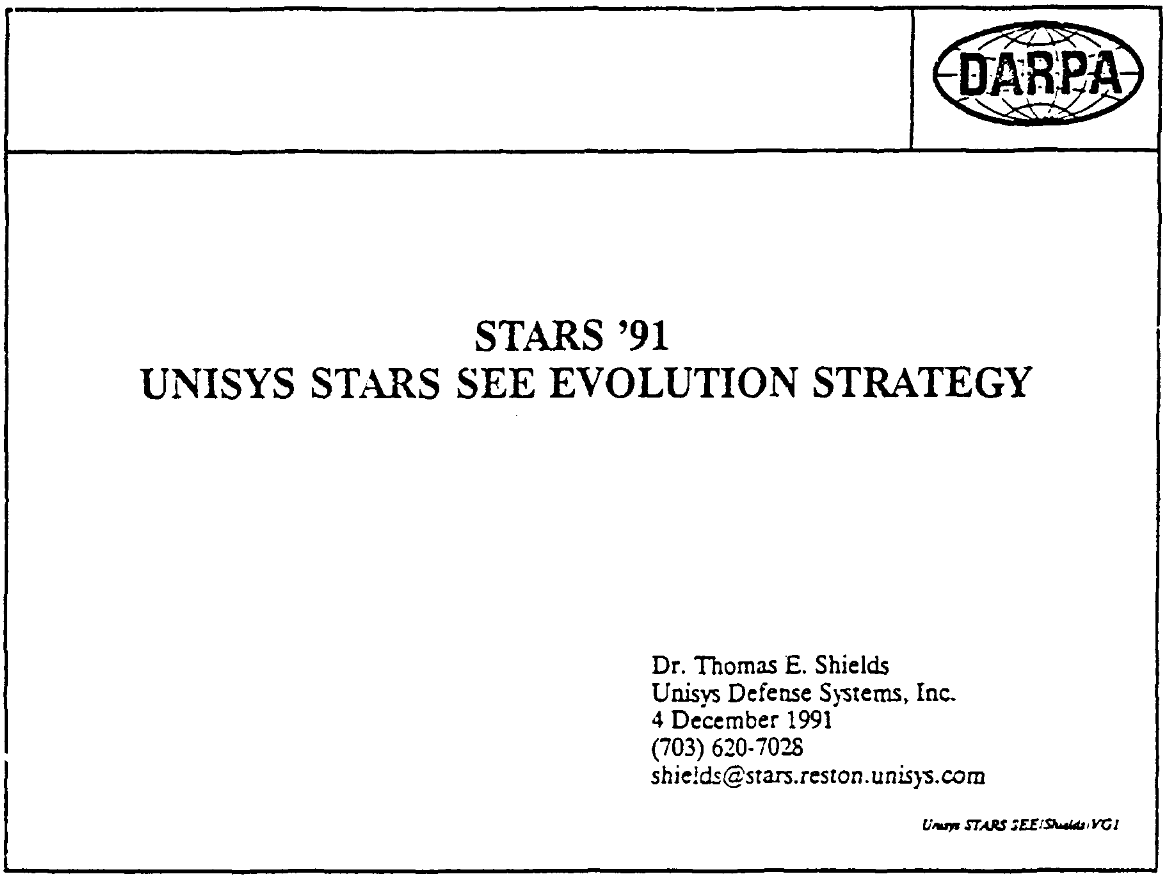
Microplanner Xpert is a trademark of Micro Planning Inc.

OSF/Motif is a trademark of the Open Systems Foundation

Rational is a trademark of Rational

Teamwork is a trademark of Cadre Technologies

WordPerfect is a trademark of WordPerfect Corporation

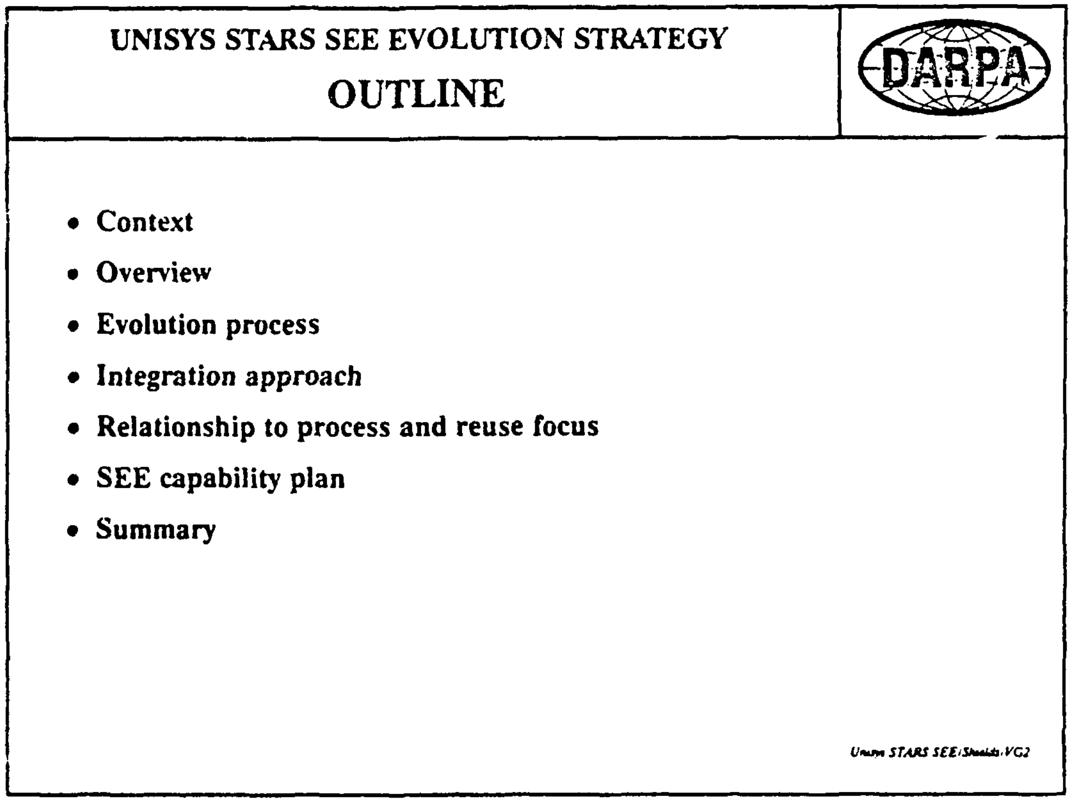


STARS '91
UNISYS STARS SEE EVOLUTION STRATEGY

Dr. Thomas E. Shields
Unisys Defense Systems, Inc.
4 December 1991
(703) 620-7028
shie!ds@stars.reston.unisys.com

Unisys STARS SEE/Shields/VG1

˥his presentation describes the strategy of the Unisys STARS team for instantiating a SEE for use on one of the STARS demonstration projects. The STARS goal is to evolve the SEE into a well-integrated, adaptable, tailorable environment supporting a process-driven, reuse-based engineering approach to Megaprogramming.



UNISYS STARS SEE EVOLUTION STRATEGY

OUTLINE

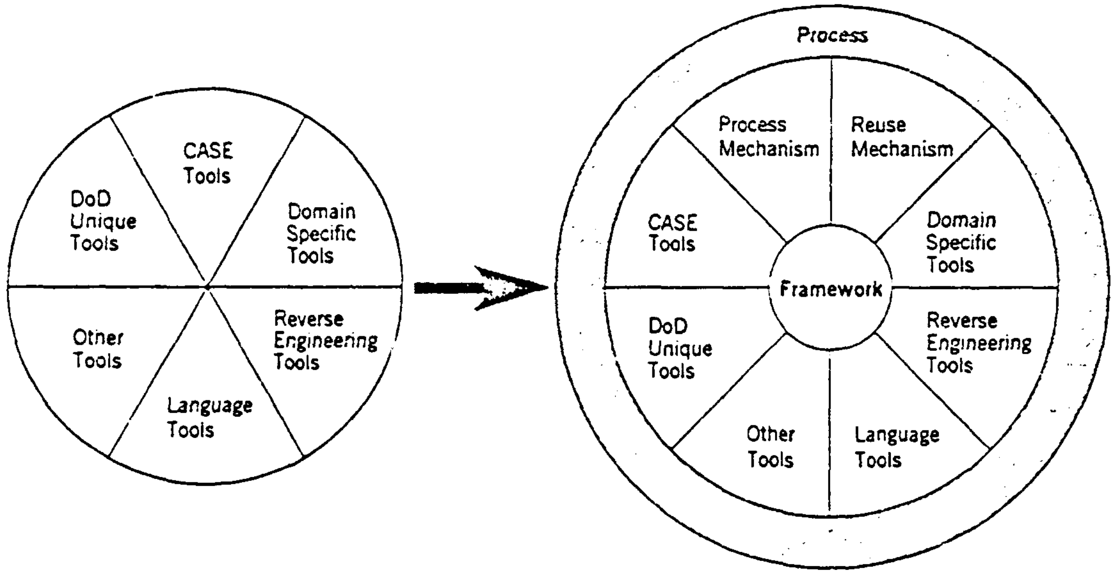
- Context
- Overview
- Evolution process
- Integration approach
- Relationship to process and reuse focus
- SEE capability plan
- Summary

Unisys STARS SEE/Shields/VG2

UNISYS STARS SEE EVOLUTION STRATEGY

EVOLUTION STRATEGY

- Transition to Framework-Based Environments Incorporating Process and Reuse Capabilities

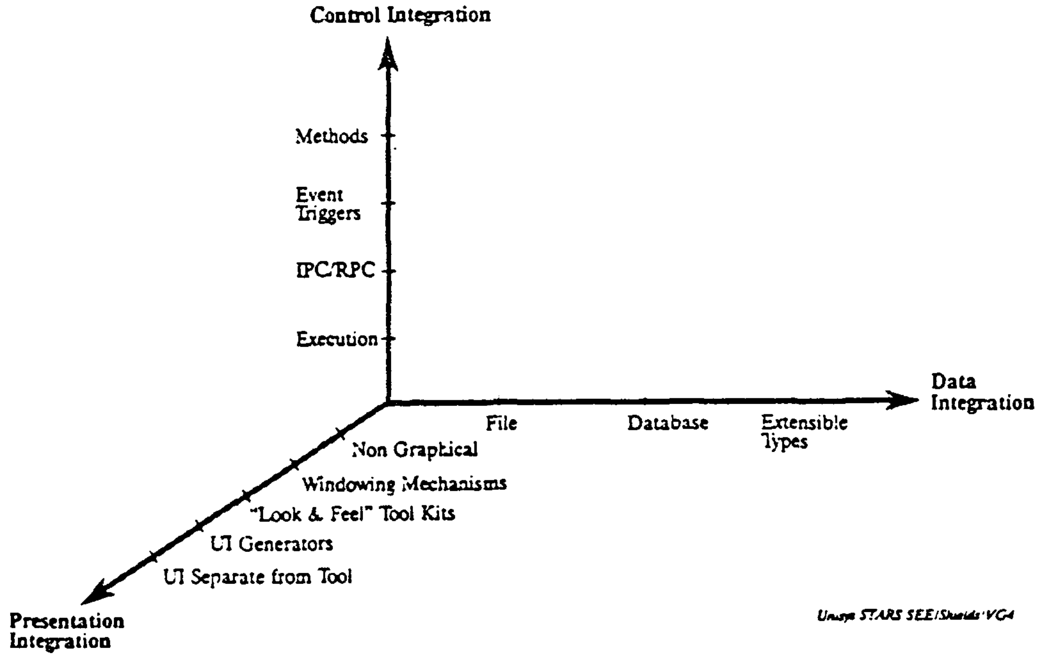


Unisys STARS SEE/Shields/VG3

The goal of the STARS Software Engineering Environment (SEE) evolution strategy is to demonstrate the benefits of a shift from point-to-point links between standalone tools to an open architecture approach based on an integrating framework. A STARS SEE incorporates reuse library technology tightly coupled with software development tools through use of this integrating framework, and it provides process integration through framework services that allow the environment to explicitly support project-specific software development life-cycle procedures and methodologies. In particular, a STARS SEE supports domain specific reuse-based software development processes.

UNISYS STARS SEE EVOLUTION STRATEGY

DIMENSIONS OF INTEGRATION

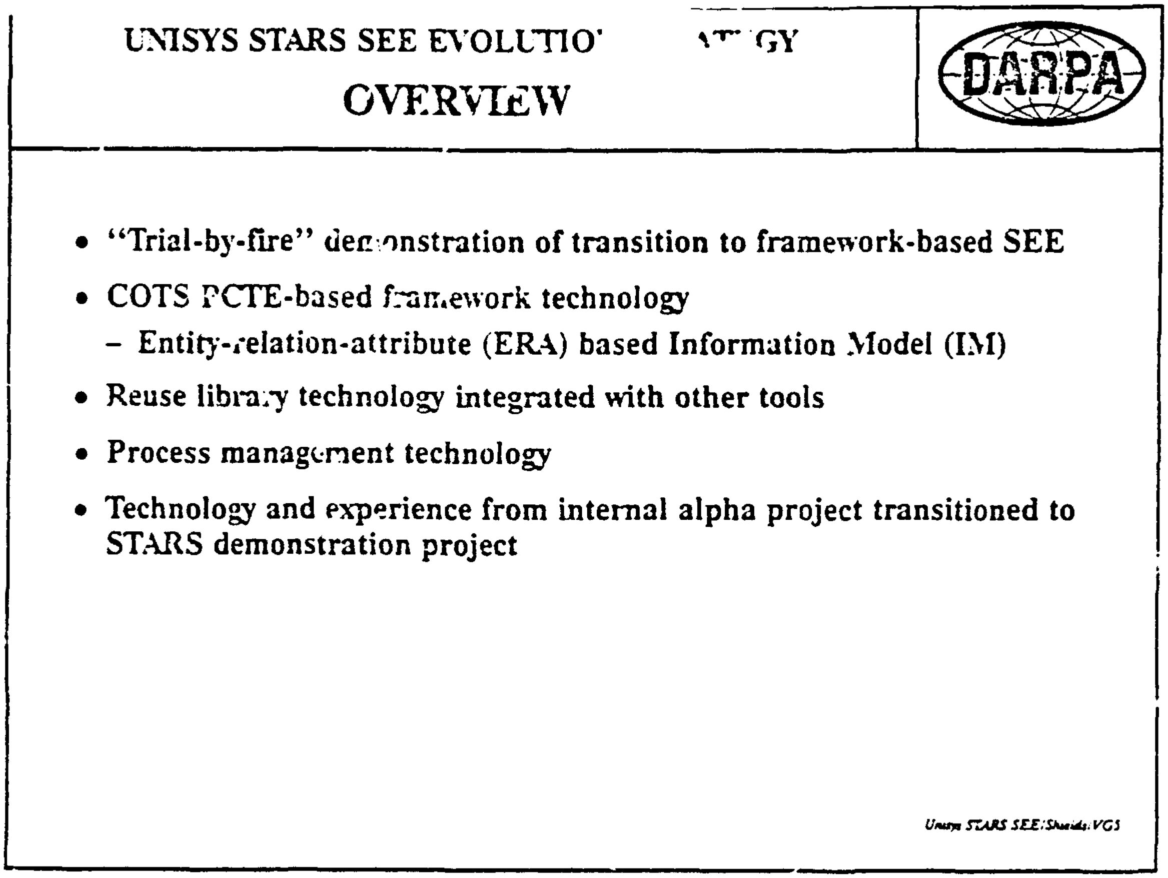


Unisys STARS SEE/Shields'VG4

"Control integration" refers to the ability to transfer control among a set of activities in a seamless, transparent manner. Examples of control integration mechanisms include shell scripts, remote execution, remote procedure calls (RPC), point-to-point messages, and broadcast messages. This dimension spans coarse-grained composition of tools into tool ensembles at the environment level to fine-grained composition of tool fragments into tools at the tool level.

"Data integration" refers to the ability to use common data formats permitting a tool to easily use the results produced by other tools. Examples of data integration mechanisms include Object Management Services (OMS), common data schemas, and tool interconnection languages. This dimension spans management of coarse-grained data (file-sized objects) at the environment level to management of fine-grained data (byte-sized objects) at the tool level.

"Presentation integration" refers to the ability for tools to provide a consistent (visual and behavioral) interface to the user. This is also referred to as "common look-and-feel".



UNISYS STARS SEE EVOLUTION STRATEGY

OVERVIEW

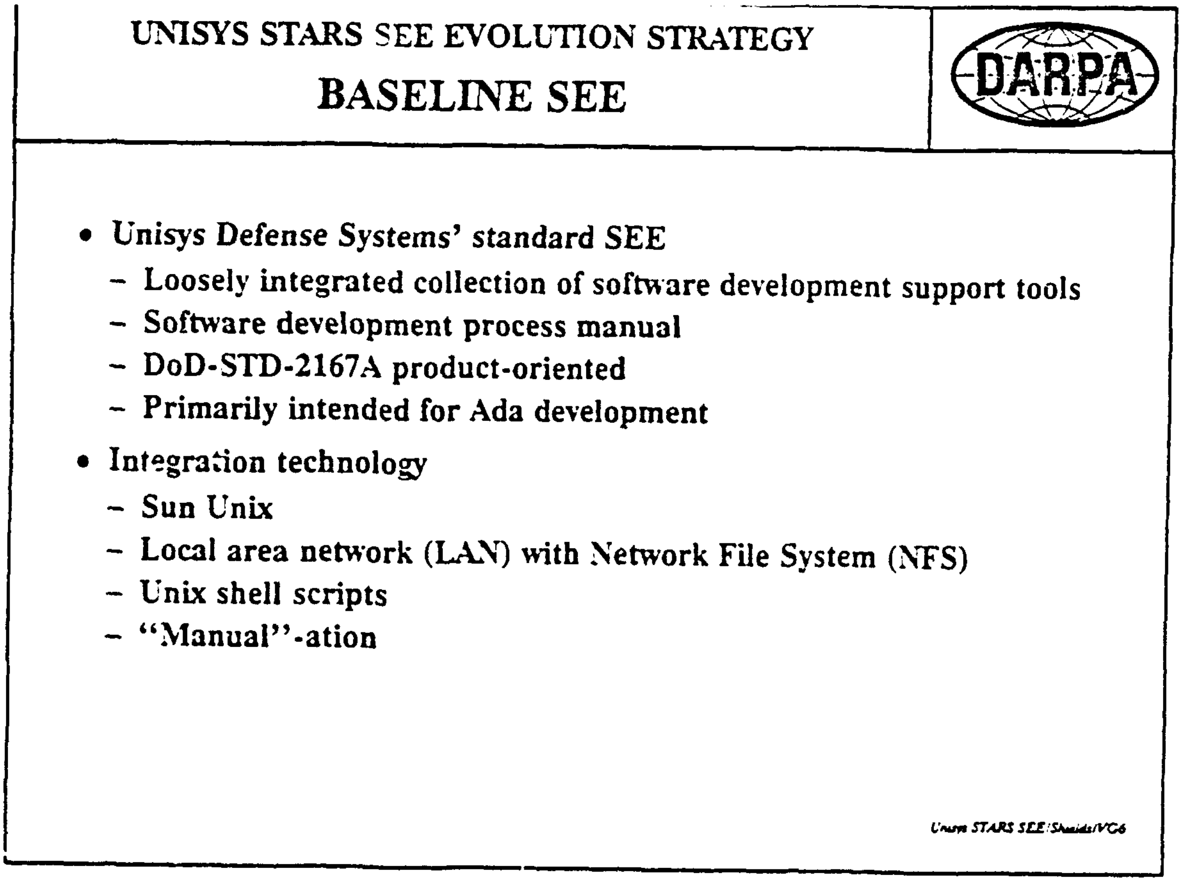
- "Trial-by-fire" demonstration of transition to framework-based SEE
- COTS PCTE-based framework technology
  - Entity-relation-attribute (ERA) based Information Model (IM)
- Reuse library technology integrated with other tools
- Process management technology
- Technology and experience from internal alpha project transitioned to STARS demonstration project

Unisys STARS SEE:Slides:VGS

The Unisys STARS approach is to gain experience with the application of SEE framework technology on relatively controlled, but realistic, alpha projects, and then transition that experience, along with proven framework technology, to one of the STARS sponsored demonstration projects beginning in October 1993. The plan is to incrementally insert commercially available data and control (and to a lesser extent, presentation) integration technology into active internal alpha projects within Unisys Defense Systems.

Unisys Defense Systems has standardized on a loosely coupled set of software development tools. This baseline will be transitioned to a framework-based SEE using the same (and similar) tools, integrated with reuse and process management technology. The lessons learned from this experience will be particularly important to the potential problems of transitioning a PDSS project to a STARS SEE.

Unisys STARS is using the Portable Common Tool Environment (PCTE) product from GIE Emeraude primarily for data integration, and intends to use HP's Softbench product for control integration. PCTE provides control integration mechanisms, as well. Unisys will also introduce other PCTE-based horizontal tool products, such as the European Advanced Software Technology (EAST) environment from SFGL.



UNISYS STARS SEE EVOLUTION STRATEGY

BASELINE SEE

- Unisys Defense Systems' standard SEE
  - Loosely integrated collection of software development support tools
  - Software development process manual
  - DoD-STD-2167A product-oriented
  - Primarily intended for Ada development
- Integration technology
  - Sun Unix
  - Local area network (LAN) with Network File System (NFS)
  - Unix shell scripts
  - "Manual"-ation

Unisys STARS SEE/Shields/VG6

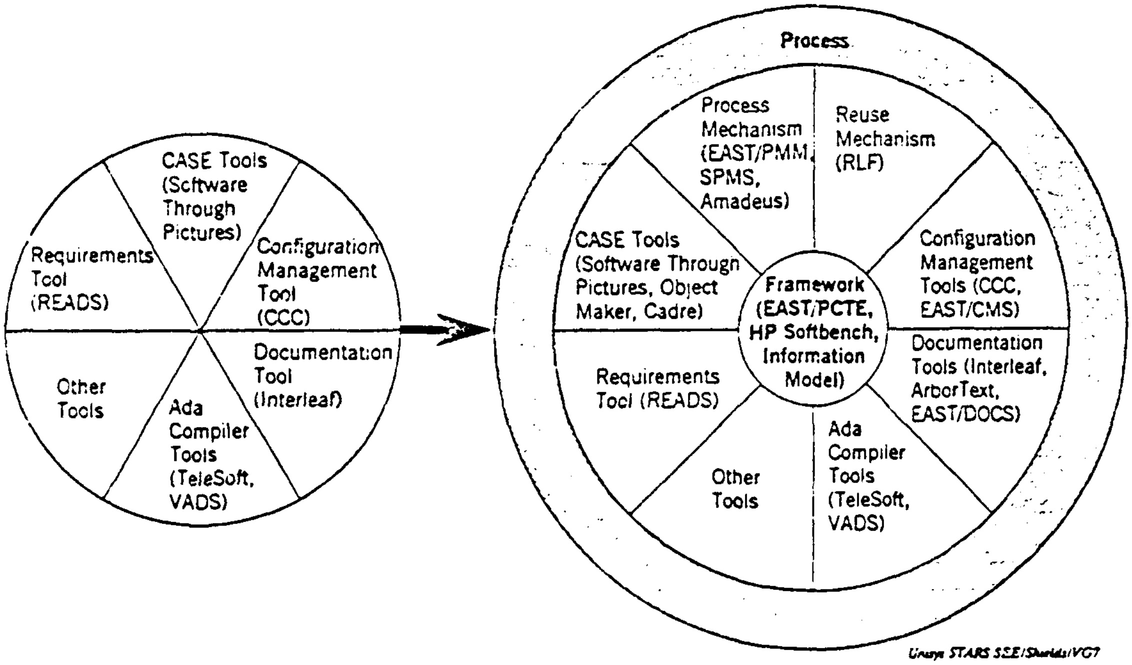
The Unisys Defense Systems standard software development environment is in use by both production and IR&D projects within the company.

The environment consists of a set of loosely integrated software development tools: IDE's Software through Pictures product suite; READS, a Unisys internally-developed requirements tool; CCC for configuration management; Interleaf for documentation production; the 20/20 spreadsheet product for metric analysis; and, in most cases, an Ada compiler (either TeleSoft or VADS).

A important aspect of this standard environment is that Unisys has developed a standard software development process manual, tailored to this choice of tools. This manual defines the processes to be (manually) executed during various phases of software development and maintenance projects generating DoD-STD-2167A compliant documentation. The manual also specifies product and process metrics to be (manually) collected; these metrics are analyzed using standard spreadsheets.

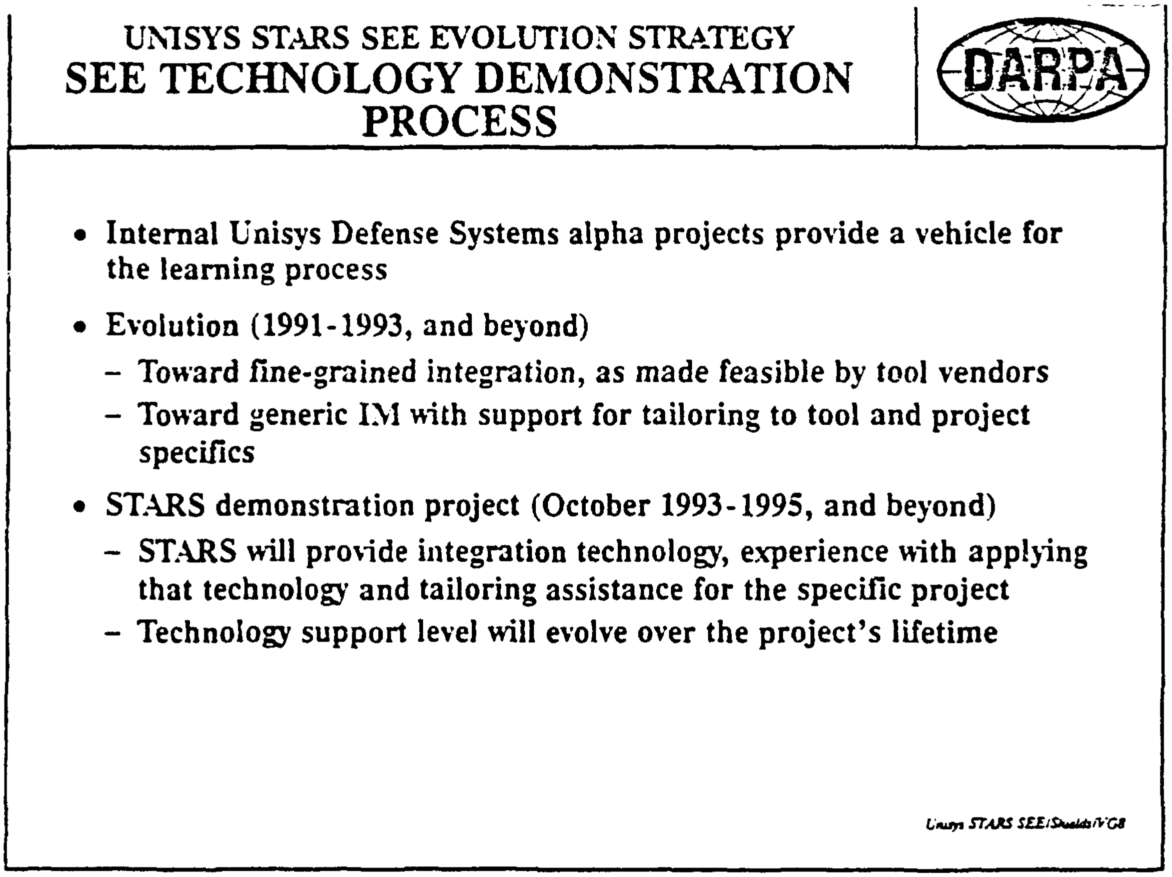
UNISYS STARS SEE EVOLUTION STRATEGY

UNISYS INTERNAL ALPHA PROJECTS


CASE Tools (Software Through Pictures)
Configuration Management Tool (CCC)
Documentation Tool (Interleaf)
Ada Compiler Tools (TeleSoft, VADS)
Other Tools
Requirements Tool (READS)
Process
Process Mechanism (EAST/PMM, SPMS, Amadeus)
Reuse Mechanism (RLF)
Configuration Management Tools (CCC, EAST/CMS)
Documentation Tools (Interleaf, ArborText, EAST/DOCS)
Ada Compiler Tools (TeleSoft, VADS)
Other Tools
Requirements Tool (READS)
CASE Tools (Software Through Pictures, Object Maker, Cadre)
Framework (EAST/PCTE, HP Softbench, Information Model)

Unisys STARS SEE/Slides/VG7

The goal of the Unisys internal alpha projects is to gain experience in the process of STARS SEE framework technology transition. The primary focus of these alpha projects will be to learn how to deal with the cultural and technical barriers to insertion of framework integration technology (as well as reuse and process management automation) into an existing project with an existing standard set of software development tools and processes. A secondary focus will be understanding the issues related to replacing a project's current "tools of choice" with equivalent tools that support finer-grained integration, and issues related to enhancing a project's current "processes of choice" with reuse-oriented process steps.



UNISYS STARS SEE EVOLUTION STRATEGY

SEE TECHNOLOGY DEMONSTRATION PROCESS

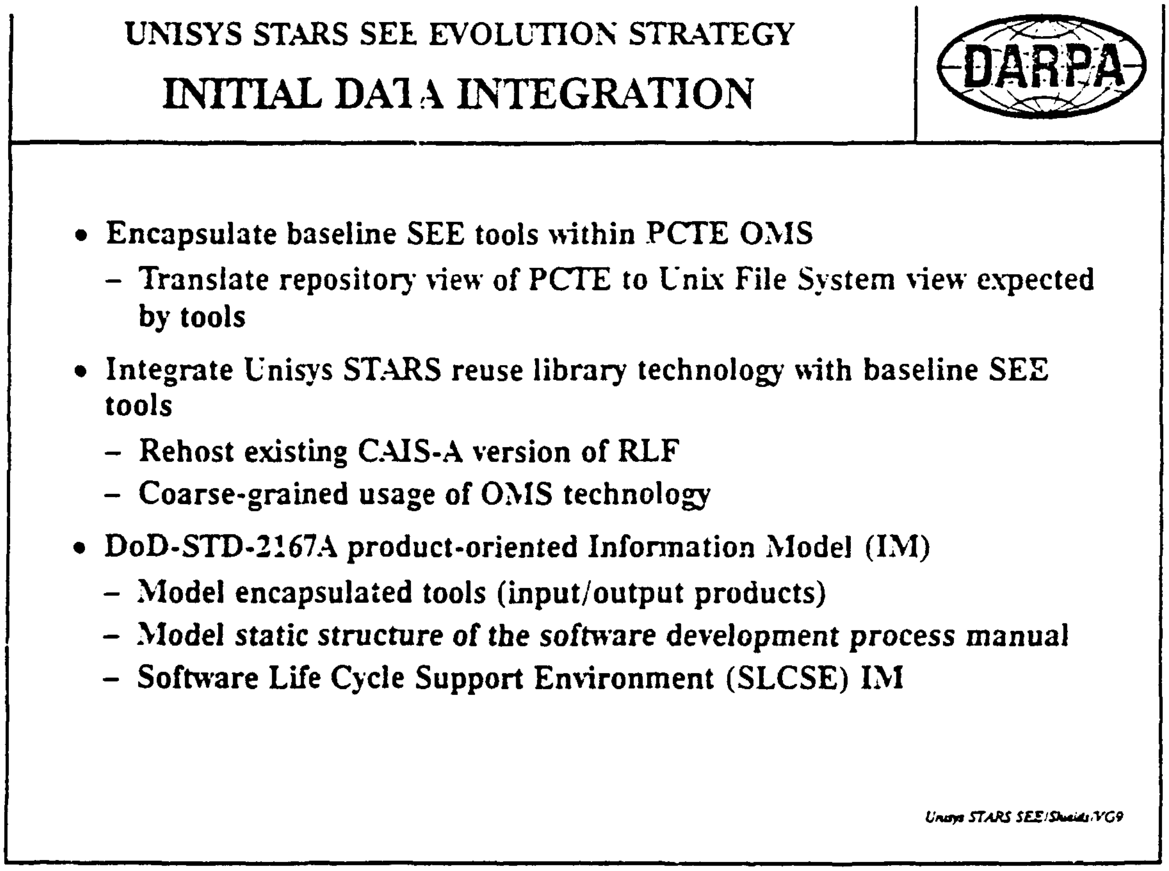
- Internal Unisys Defense Systems alpha projects provide a vehicle for the learning process
- Evolution (1991-1993, and beyond)
  - Toward fine-grained integration, as made feasible by tool vendors
  - Toward generic IM with support for tailoring to tool and project specifics
- STARS demonstration project (October 1993-1995, and beyond)
  - STARS will provide integration technology, experience with applying that technology and tailoring assistance for the specific project
  - Technology support level will evolve over the project's lifetime

Unisys STARS SEE/Shields/VG8

Internal Unisys Defense Systems alpha projects will provide a vehicle for learning how to (and, possibly, how not to) insert framework, reuse, and process automation technology into an existing project environment with defined and measured standardized processes. These internal alpha projects will focus on coarse-grained integration of COTS tools already in use by a project team, and on adapting and tailoring the SEE via tool- and project-specific information modeling.

The particular software development tools, reusable assets and processes used for the internal alpha projects may or may not be applicable to, or even available for, the STARS demonstration project.

The Unisys STARS team will bring selected framework integration technology, experience integrating software development tools with reuse and process management technology, and an adaptable, tailorable underlying generic information model baseline to the STARS demonstration project. The Unisys STARS team will support that demonstration project in the initial instantiation, tailoring, and training for use of a project-specific SEE, and in supporting, adapting and evolving that SEE over the lifetime of the demonstration. Both user feedback as well as availability of evolving COTS product technology will play a role in the SEE evolution process.



UNISYS STARS SEE EVOLUTION STRATEGY

INITIAL DATA INTEGRATION

- Encapsulate baseline SEE tools within PCTE OMS
  - Translate repository view of PCTE to Unix File System view expected by tools
- Integrate Unisys STARS reuse library technology with baseline SEE tools
  - Rehost existing CAIS-A version of RLF
  - Coarse-grained usage of OMS technology
- DoD-STD-2167A product-oriented Information Model (IM)
  - Model encapsulated tools (input/output products)
  - Model static structure of the software development process manual
  - Software Life Cycle Support Environment (SLCSE) IM

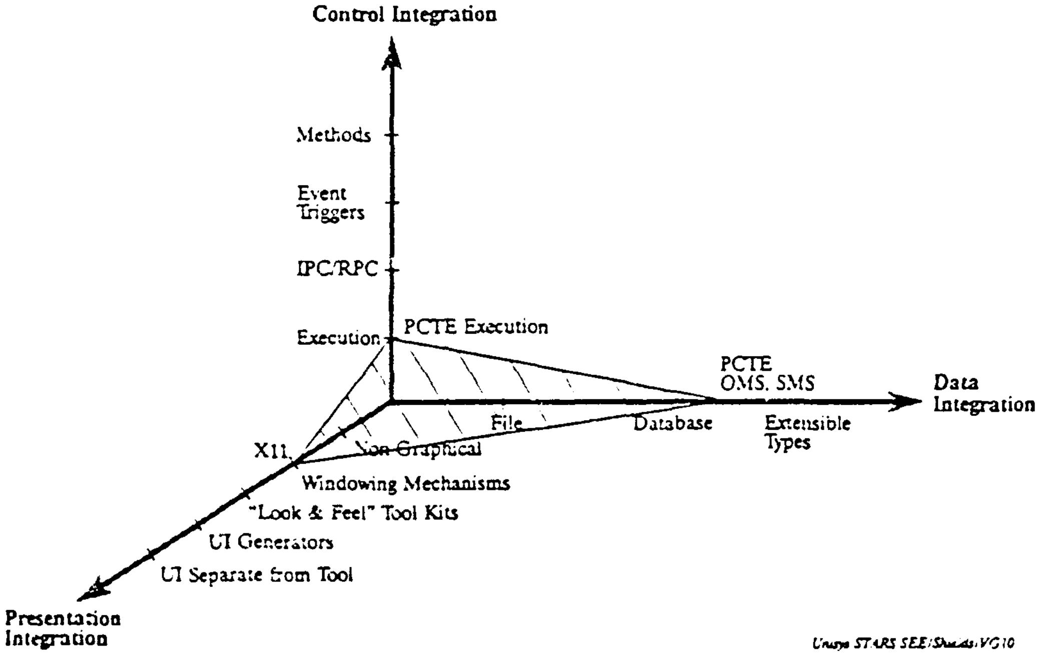
Unisys STARS SEE/Slides:VG9

During the 1990 timeframe, Unisys built an experimental SEE hosted on CAIS-A (DoD-STD-1838A), a portable tool integration environment incorporating an ERA-based OMS. Unisys is currently upgrading that environment under funding from NOSC. This experimental SEE focused on tool data integration. CAIS-A is not currently supported by any commercial products, so we have selected a PCTE-based product to form the baseline for the STARS demonstration project supported by Unisys. Since PCTE also incorporates an ERA-based OMS, Unisys will be able to leverage considerable experience with ERA-based tool data integration. Unisys expects to be able to transition this PCTE data integration experience to the IBM STARS team when they begin to address data integration issues.

Since the Unisys internal alpha projects will be producing DoD-STD-2167A compliant documentation products, the underlying ERA Information Model (IM) used with PCTE will need to reflect this focus. The Air Force Rome Laboratory has been sponsoring a project called the Software Life Cycle Support Environment (SLCSE), which includes a DoD-STD-2167A ERA-based IM. Unisys plans to start with the SLCSE data schema as the initial baseline for the PCTE IM, suitably tailored for the particular tools used within Unisys. The Rome Laboratory recently awarded a contract to ISSI for the SLCSE Enhancements and Demonstration Program. As a part of that program, Enhanced SLCSE (E-SLCSE) will probably adopt the PCTE standard OMS interface. Unisys and ISSI have begun discussions that should lead to coordinated efforts to move the SLCSE IM to a PCTE technology base.

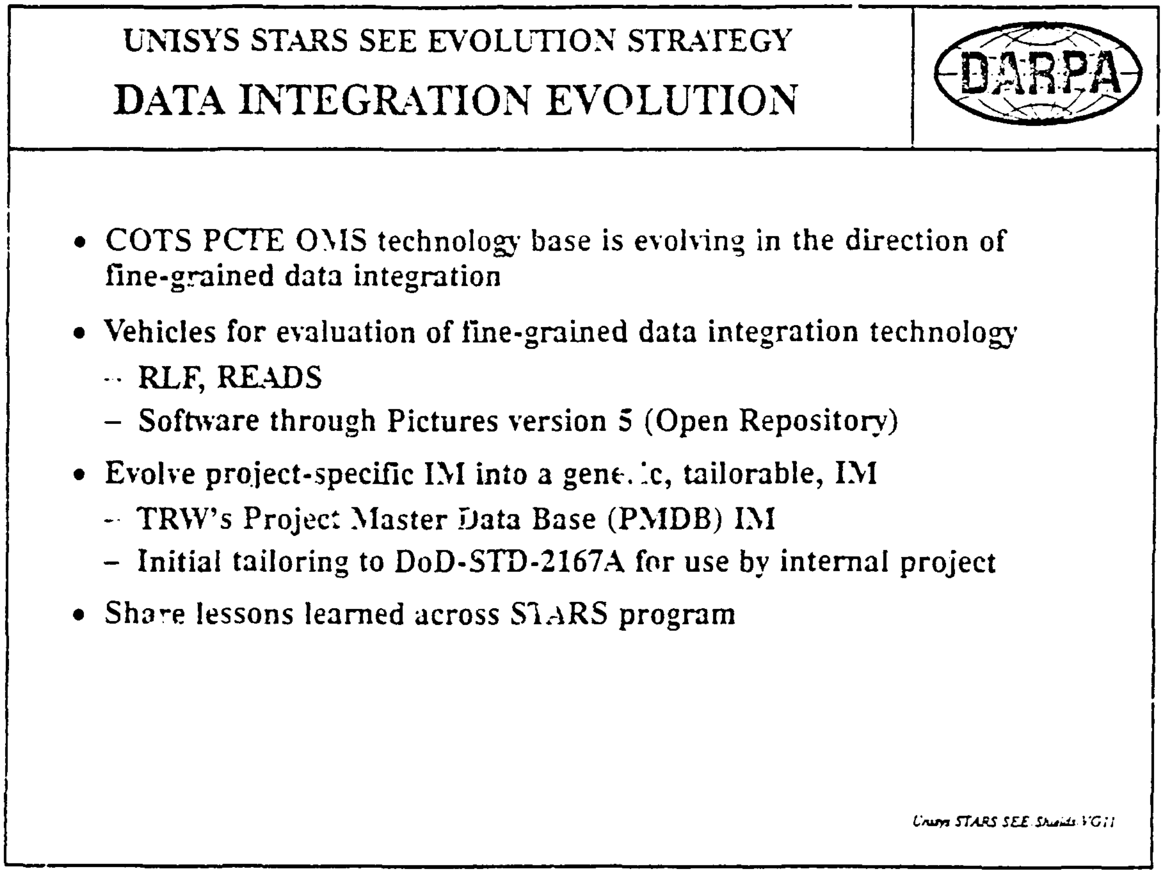
UNISYS STARS SEE EVOLUTION STRATEGY

INITIAL INTEGRATION



Unisys STARS SEE/Slides/VG10

This is a depiction of the framework integration space that will be used in the initial phase of SEE integration. Although the initial phase will concentrate on data integration, PCTE tool encapsulation makes use of the PCTE execution facilities. Presentation integration will not be explicitly addressed either, but there will be some commonality, at least at the X11 level.



UNISYS STARS SEE EVOLUTION STRATEGY

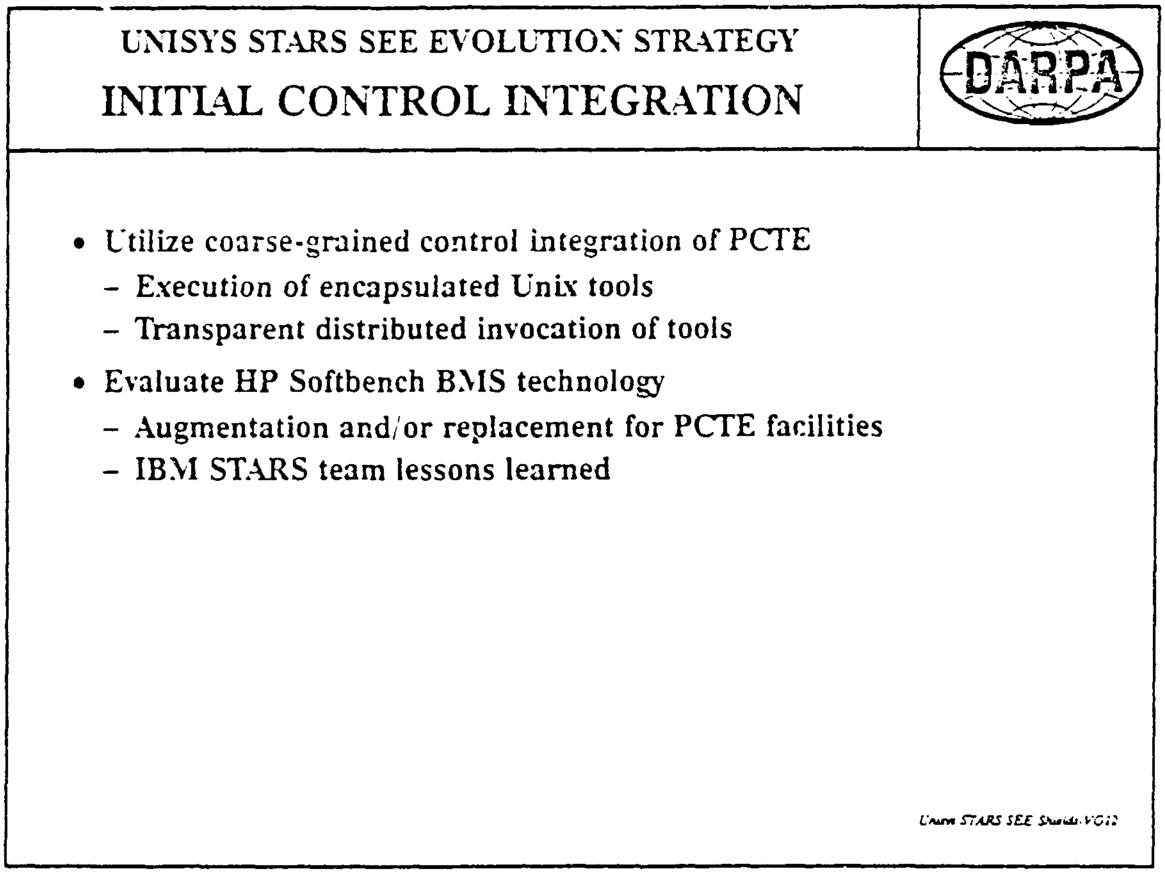
DATA INTEGRATION EVOLUTION

- COTS PCTE OMS technology base is evolving in the direction of fine-grained data integration
- Vehicles for evaluation of fine-grained data integration technology
  - RLF, READS
  - Software through Pictures version 5 (Open Repository)
- Evolve project-specific IM into a generic, tailorable, IM
  - TRW's Project Master Data Base (PMDB) IM
  - Initial tailoring to DoD-STD-2167A for use by internal project
- Share lessons learned across STARS program

Unisys STARS SEE Slides VG11

The Unisys STARS team will support the alpha and demonstration projects with whatever PCTE OMS technology base is made available by commercial vendors during the period of the demonstration projects. The anticipated progression of support of the PCTE standard from GIE Emeraude is PCTE 1.5, with some extensions from PCTE+ (Emeraude's V12 product baseline) now, PCTE+ compatibility in the spring of 1992, and ECMA PCTE compliance sometime in 1993 (Emeraude's V20 product baseline). Beyond this evolution to compliance with the current ECMA PCTE standard, PCTE is expected to evolve, prior to the end of the STARS demonstration project, in the direction of supporting (either directly or interoperating with) fine-grained data integration technology. Whether or not this is applicable to STARS depends both on vendors of PCTE-compliant framework products supporting the new functionality and on software development tool vendors "opening up" the data repository access facilities of their respective tools. For example, HP is proposing what they are calling Extended Tool Integration Services (XTIS) as a combination of the current ECMA PCTE facilities for coarse-grained objects with something as yet undefined (at least publicly) to support fine-grained objects as the product evolution direction for Softbench.

Unisys is planning to evaluate the ability of currently available PCTE OMS technology to support fine-grained objects. This investigation could be done with a number of vehicles. such as RLF, a READS conversion from the Ingres RDBMS, and (the not yet available) release 5 of IDE's CASE tool product suite, which will include an open interface to the underlying data repository, allowing integration of the IDE toolset on top of any data repository technology that can support IDE's open repository interface, such as PCTE.



UNISYS STARS SEE EVOLUTION STRATEGY

INITIAL CONTROL INTEGRATION

- Utilize coarse-grained control integration of PCTE
  - Execution of encapsulated Unix tools
  - Transparent distributed invocation of tools
- Evaluate HP Softbench BMS technology
  - Augmentation and/or replacement for PCTE facilities
  - IBM STARS team lessons learned

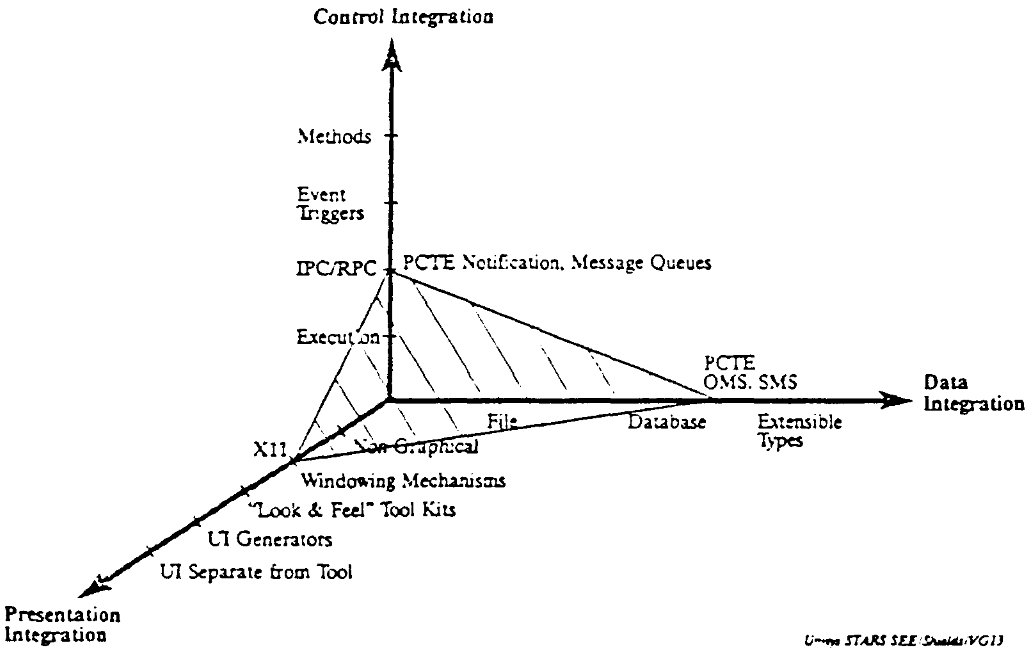
Unisys STARS SEE Slides-VG12

The initial use of control integration will be to support coarse-grained tool composition, providing straightforward automation of tool invocation. The ability to automatically invoke the appropriate tool to view a reusable asset just located within the reuse library (e.g., a syntax-directed text editor for source code, a graphical design editor for a design document) is an example of control integration.

The Unisys STARS team intends to focus the majority of its resources initially on data integration. We therefore plan to rely heavily on the work done by the IBM STARS team in the control integration area, as their resources will initially focus on control integration.

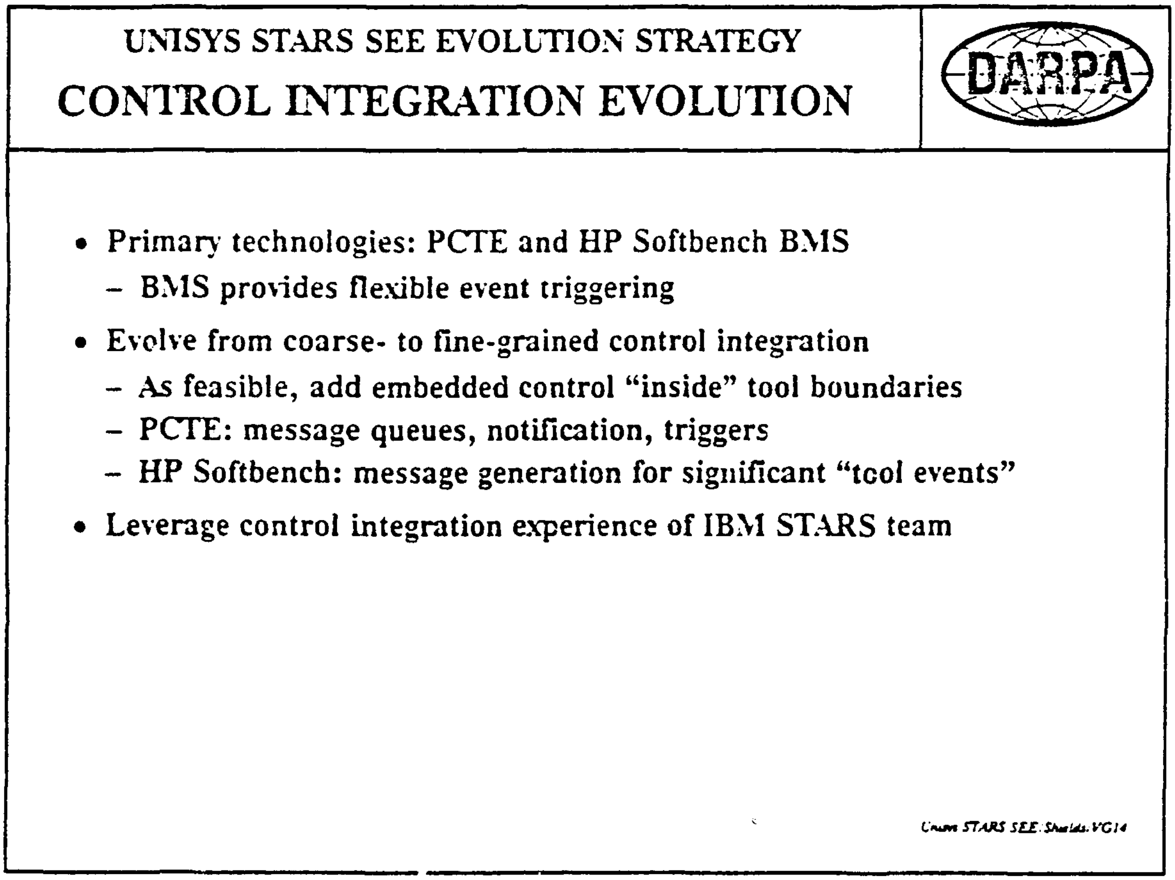
UNISYS STARS SEE EVOLUTION STRATEGY

PCTE INTEGRATION



Unisys STARS SEE/Slides/VG13

This is a depiction of the framework integration space provided by PCTE.



UNISYS STARS SEE EVOLUTION STRATEGY

CONTROL INTEGRATION EVOLUTION

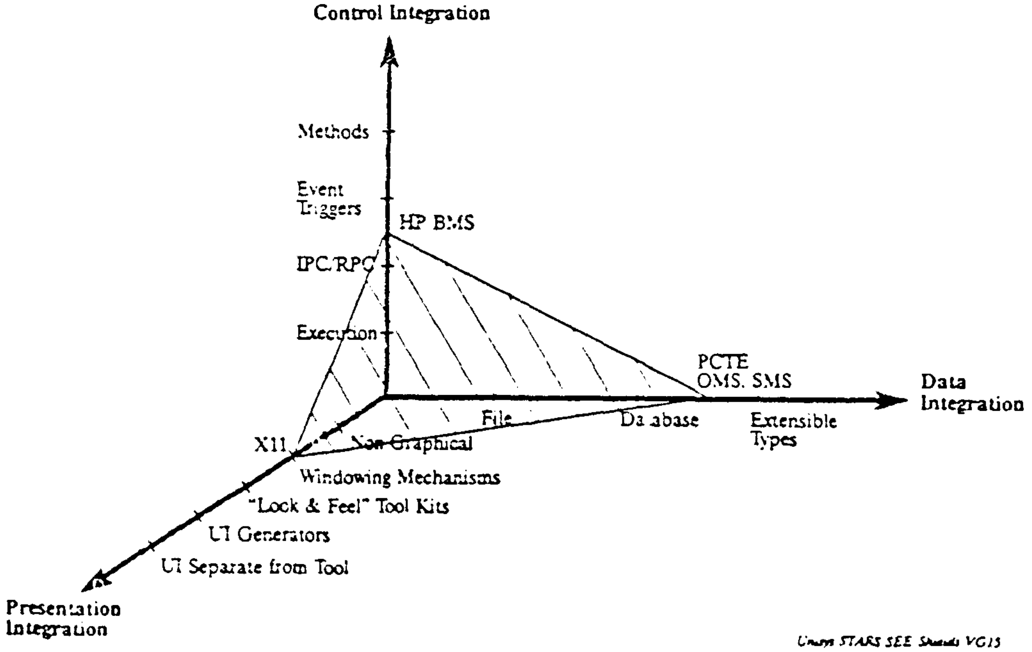
- Primary technologies: PCTE and HP Softbench BMS
  - BMS provides flexible event triggering
- Evolve from coarse- to fine-grained control integration
  - As feasible, add embedded control "inside" tool boundaries
  - PCTE: message queues, notification, triggers
  - HP Softbench: message generation for significant "tool events"
- Leverage control integration experience of IBM STARS team

Unisys STARS SEE: Shields.VG14

The Unisys STARS team plans to use RLF and/or READS as readily available platforms for incorporating fine-grained, embedded control integration. Additional opportunities for incorporation of fine-grained control integration between tools depends on the tool vendors "opening up" their tools at control point boundaries. As other tool vendors, such as IDE, provide support for HP Softbench and/or PCTE control integration, Unisys will incorporate those capabilities into the evolving SEE.

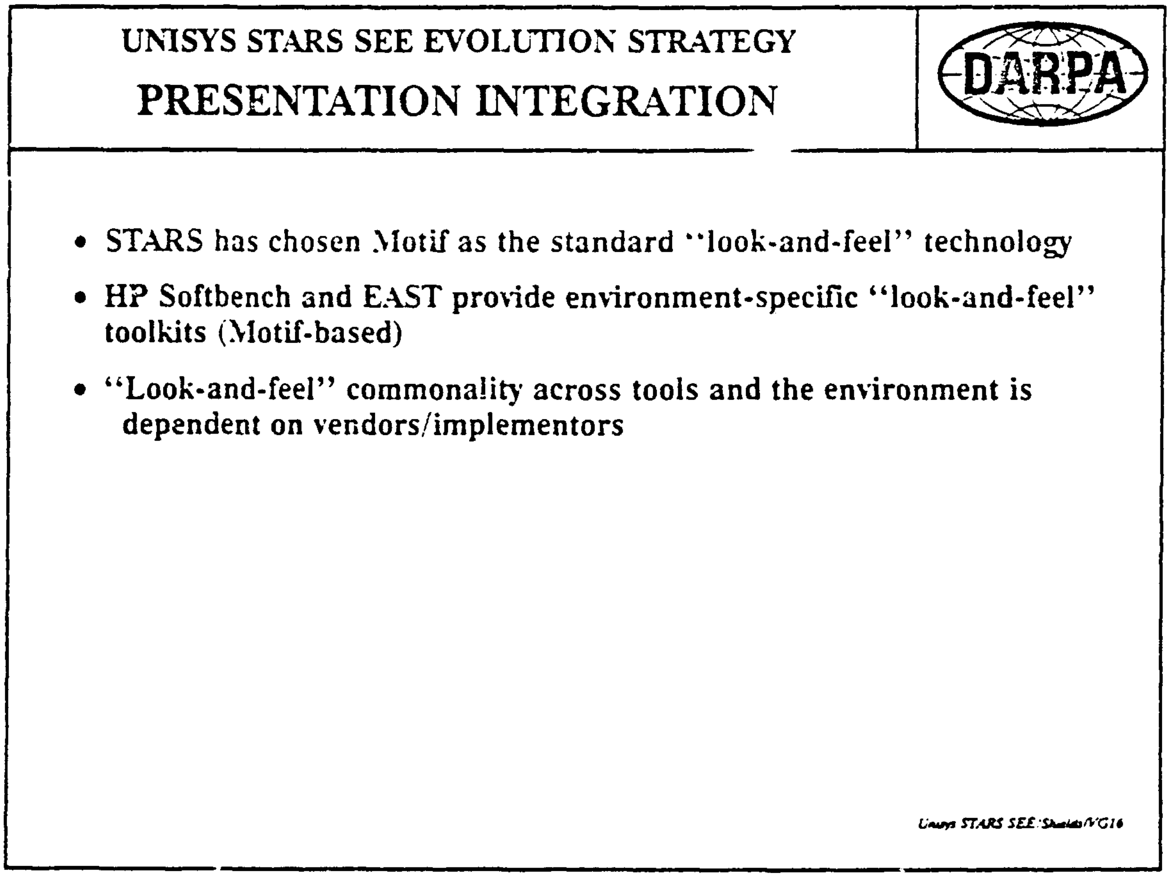
UNISYS STARS SEE EVOLUTION STRATEGY

INTEGRATION ADDING BMS



Unisys STARS SEE Shands VG15

This is a depiction of the framework integration space provided by PCTE in combination with HP's Softbench Broadcast Message Server (BMS) technology.



UNISYS STARS SEE EVOLUTION STRATEGY

PRESENTATION INTEGRATION

DARPA

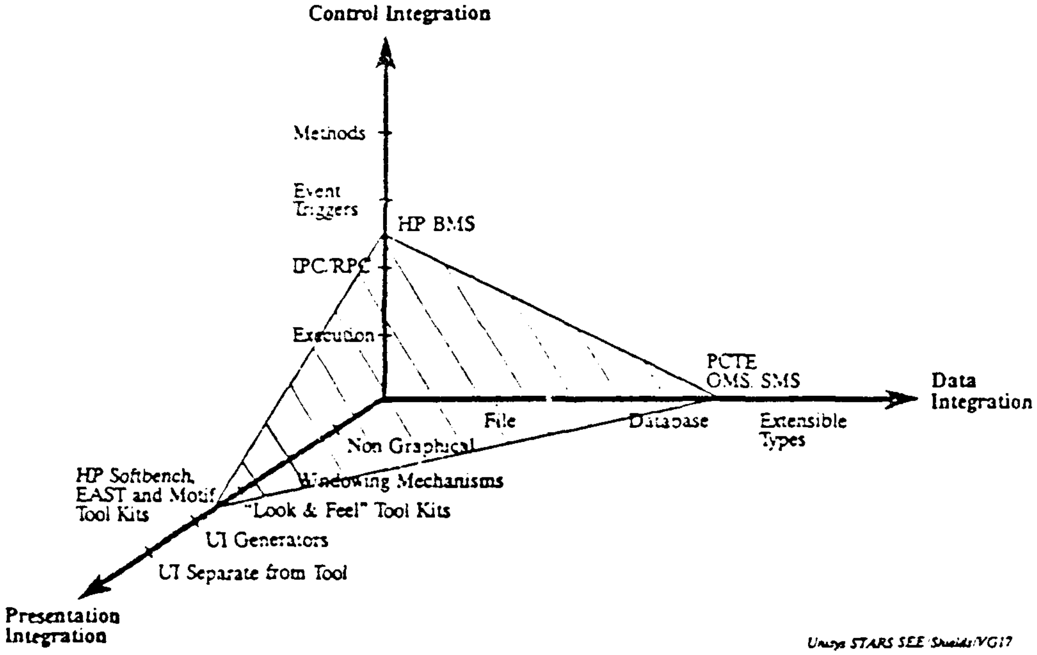
- STARS has chosen Motif as the standard "look-and-feel" technology
- HP Softbench and EAST provide environment-specific "look-and-feel" toolkits (Motif-based)
- "Look-and-feel" commonality across tools and the environment is dependent on vendors/implementors

Unisys STARS SEE:Slides/VG16

The Unisys SEE activities will not focus on presentation integration issues, other than as a byproduct of using tool encapsulation facilities provided by COTS framework products such as HP Softbench and EAST, which both include facilities for providing a common "look-and-feel."

UNISYS STARS SEE EVOLUTION STRATEGY

COTS INTEGRATION



Unisys STARS SEE Shields/VG17

This is a depiction of the framework integration space Unisys STARS expects to be using provided by COTS technology for the demonstration project. The COTS integration space may expand beyond this, depending on evolution of product technology.

UNISYS STARS SEE EVOLUTION STRATEGY

RELATIONSHIP TO PROCESS FOCUS

- Process integration through framework services
- Provide enabling technologies for:
  - Non-obtrusive collection of process/product metrics
  - Event-based task execution
  - Process/project management
- European Advanced Software Technology (EAST) environment
  - Evaluation for potential applicability to SEE
  - PCTE-based environment product
  - Process model driven environment
  - Horizontal tools: CALS-compliant documentation, CM, process/project management
- Unisys Defense Systems software development processes

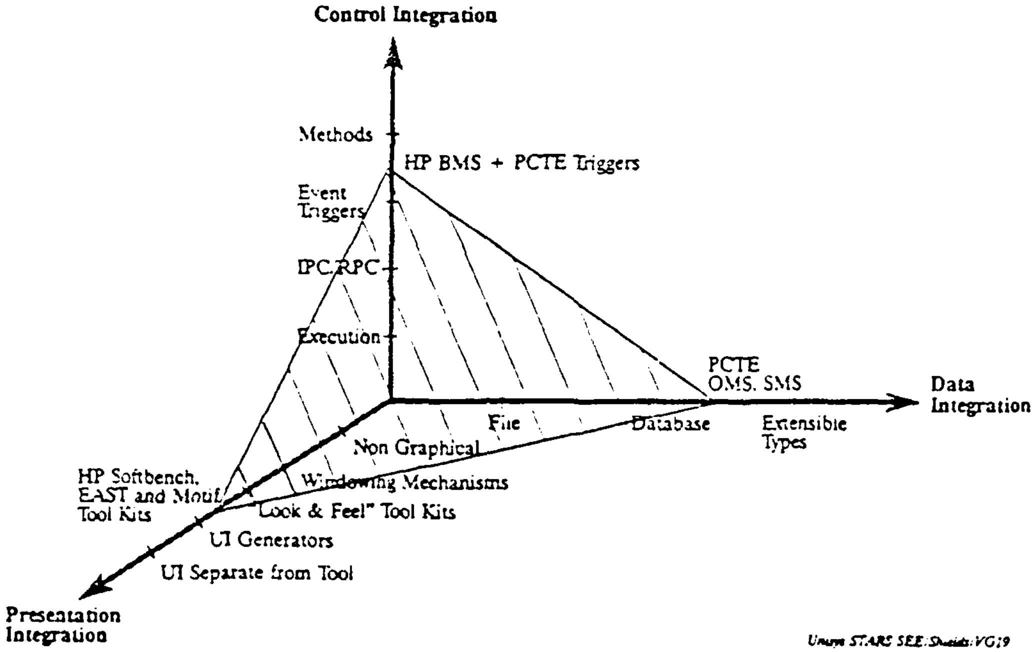
Unisys STARS SEE/Shields/VG18

The relationship to the process focus activities is to provide the underlying technology to support process integration. The Unisys STARS process focus is on process measurement (automated metric collection and analysis) to support measurement-driven feedback. The baseline technology for this capability is the Arcadia Amadeus prototype from Dr. Rick Selby at the University of California at Irvine.

Unisys will evaluate the EAST environment as a potential COTS baseline for a process-centered environment. EAST provides process and project management capabilities tightly integrated with documentation and configuration management tools. The IBM STARS team will, in parallel, be evaluating the Entreprise II environment, which is a competitor of EAST. Based on these two product evaluations, both teams may (or may not) decide to incorporate one or the other of these products in their respective SEE solutions provided to the STARS demonstration projects.

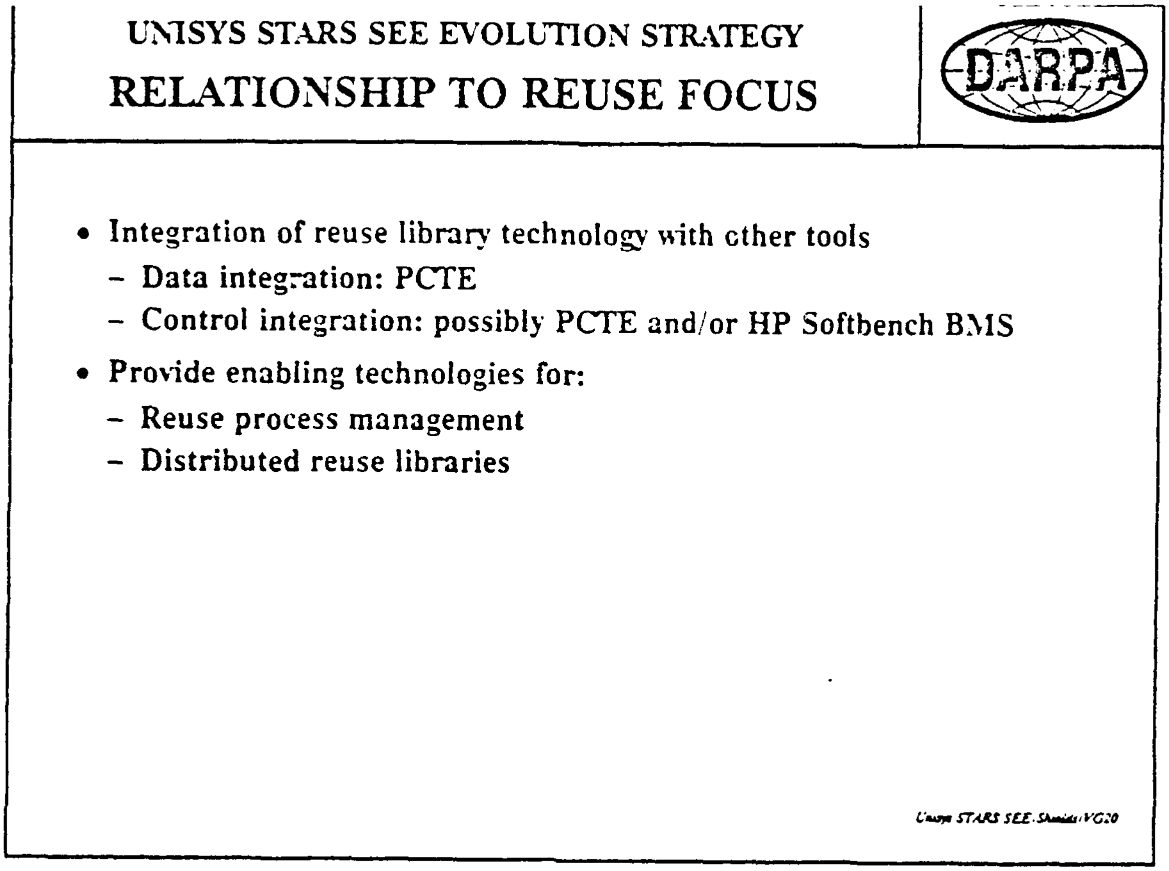
UNISYS STARS SEE EVOLUTION STRATEGY

PROCESS INTEGRATION



Unisys STARS SEE:Slides:VG19

This is a depiction of the framework integration space Unisys STARS will be consid~~ing for use for the STARS demonstration project provided by prototype technology. This may become COTS integration space by the start of, or sometime during, the STARS demonstration project. Object Management Service (OMS) triggers are not currently part of the PCTE standard, nor are they part of the GIE Emeraude PCTE product at this time. However, Emeraude has a prototype implementation of triggers in a variant of their product built for a research project, and Unisys has requested access to that variant of the product to support prototype process integration experimentation.



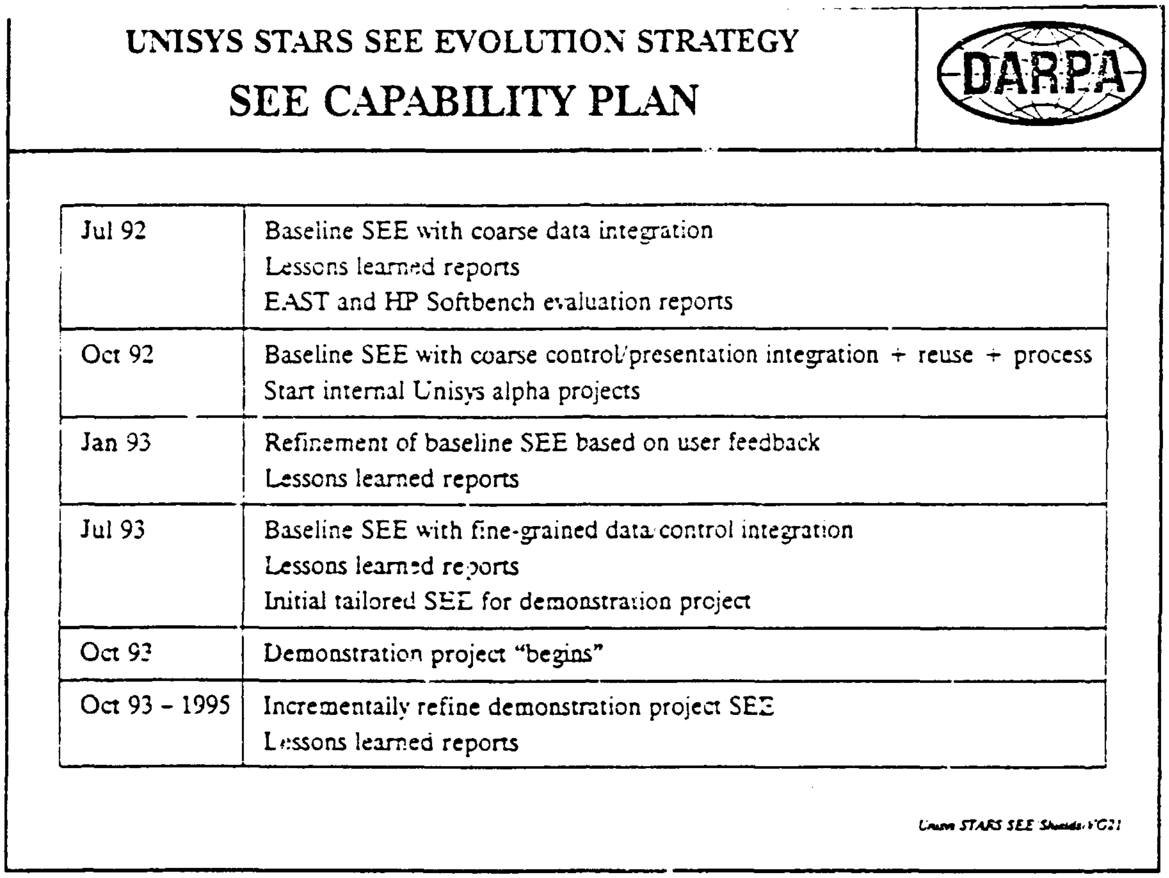
UNISYS STARS SEE EVOLUTION STRATEGY

RELATIONSHIP TO REUSE FOCUS

- Integration of reuse library technology with other tools
  - Data integration: PCTE
  - Control integration: possibly PCTE and/or HP Softbench BMS
- Provide enabling technologies for:
  - Reuse process management
  - Distributed reuse libraries

Unisys STARS SEE-Slides/VG20

The relationship to the reuse focus activities is to provide the underlying technology to support tight integration of reuse library technology with other software development tools. The purpose of this reuse/tool integration is, ultimately, to support the introduction of reuse-based processes into the SEE.



UNISYS STARS SEE EVOLUTION STRATEGY

SEE CAPABILITY PLAN

DARPA

Jul 92	Baseline SEE with coarse data integration Lessons learned reports EAST and HP Softbench evaluation reports
Oct 92	Baseline SEE with coarse control/presentation integration + reuse + process Start internal Unisys alpha projects
Jan 93	Refinement of baseline SEE based on user feedback Lessons learned reports
Jul 93	Baseline SEE with fine-grained data/control integration Lessons learned reports Initial tailored SEE for demonstration project
Oct 93	Demonstration project "begins"
Oct 93 - 1995	Incrementally refine demonstration project SEE Lessons learned reports

Unisys STARS SEE Slides VG21

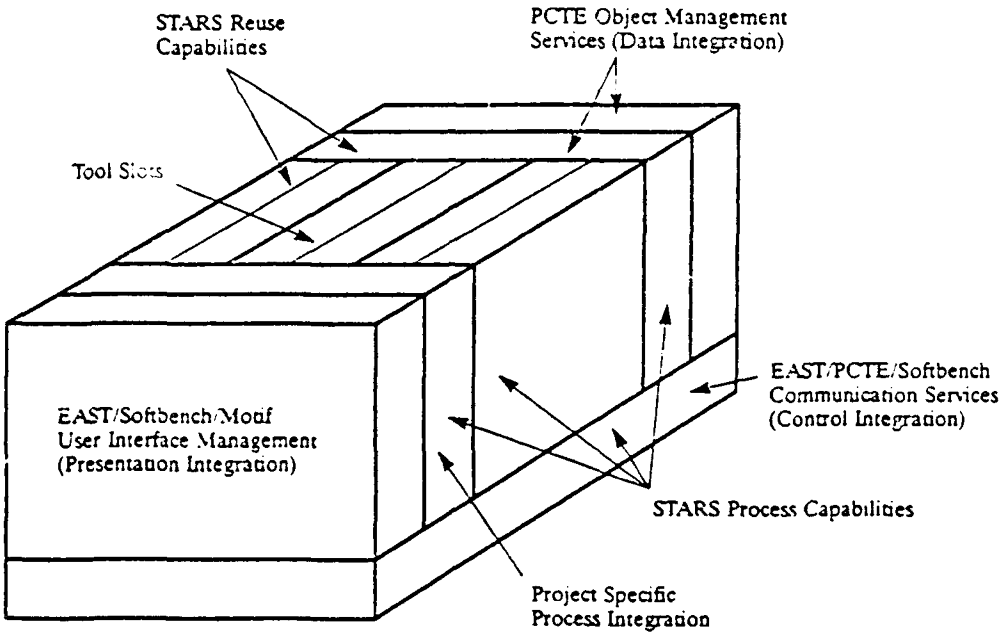
This chart depicts expected products and major planned milestones of the Unisys STARS SEE evolution process. Internal alpha projects must start not later than October 1992 in order to have adequate time to gain useful experience to ensure success of the STARS demonstration project beginning in October 1993.

Unisys expects to be able to demonstrate an initial baseline SEE at the STARS Technology Center by July 1992, incorporating coarse-grained data integration, adding coarse-grained control integration to that baseline by October 1992. Unisys will have an initial reuse and process management capability available for use by the internal alpha projects in October 1992 as well. There will be at least two incremental improvement releases of the baseline SEE delivered to the alpha projects, based on user experiences and on evolving technology availability.

Unisys expects to have identified the STARS demonstration project we will be supporting by the February 1993 timeframe. This will allow sufficient time to work with the project planning team to define an initial instantiation and tailoring of a project-specific SEE by July 1993. This lead time is needed to allow for initial training of the SEE capabilities for the project team prior to the start of the demonstration project in October 1993.

UNISYS STARS SEE EVOLUTION STRATEGY

SUMMARY



STARS Reuse Capabilities

PCTE Object Management Services (Data Integration)

Tool Slots

EAST/PCTE/Softbench Communication Services (Control Integration)

EAST/Softbench/Motif User Interface Management (Presentation Integration)

STARS Process Capabilities

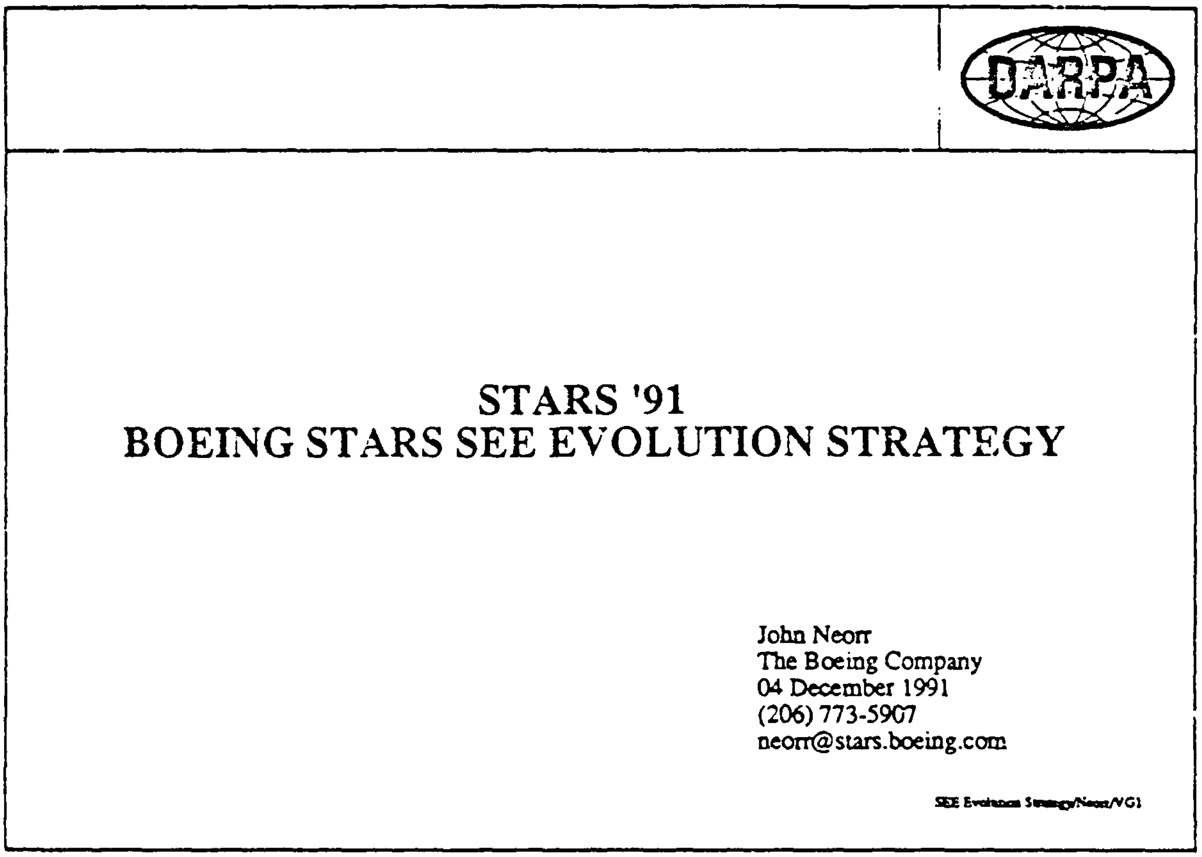
Project Specific Process Integration

Unisys STARS SEE/Shields/VG22

The Unisys STARS SEE evolution strategy focuses primarily on developing expertise with data integration technology, in particular with developing an adaptable, tailorable Information Model. The SEE integration framework will also incorporate control integration, but we plan to rely heavily on the lessons learned and experience of the IBM STARS team using HP's Softbench BMS product technology, rather than reinvent the experience from ourselves. The Unisys team will utilize the presentation integration technology provided by HP Softbench, and/or possibly by the EAST environment.

ECMA PCTE is the key component for coarse-grained data integration, but the current PCTE standard will likely need to be supplemented with additional services for management of fine-grained data. The current HP Softbench BMS technology supports coarse-grained tool composition, but likely will need to be supplemented by additional fine-grained tool-composition mechanisms. In both areas, the ability to capitalize on fine-grained integration mechanisms will depend on software development tool vendors "opening up" their products. The STARS program as a whole will be working with industry to facilitate product evolution in this direction.

The strategy is to "learn by doing" in preparation for supporting the STARS demonstration project, through incremental evolution of the Unisys Defense Systems' standard baseline SEE in the context of internal STARS-supported alpha projects. The end result will be adaptable, tailorable SEE integration framework technology and experience that can be transitioned to the STARS demonstration project.

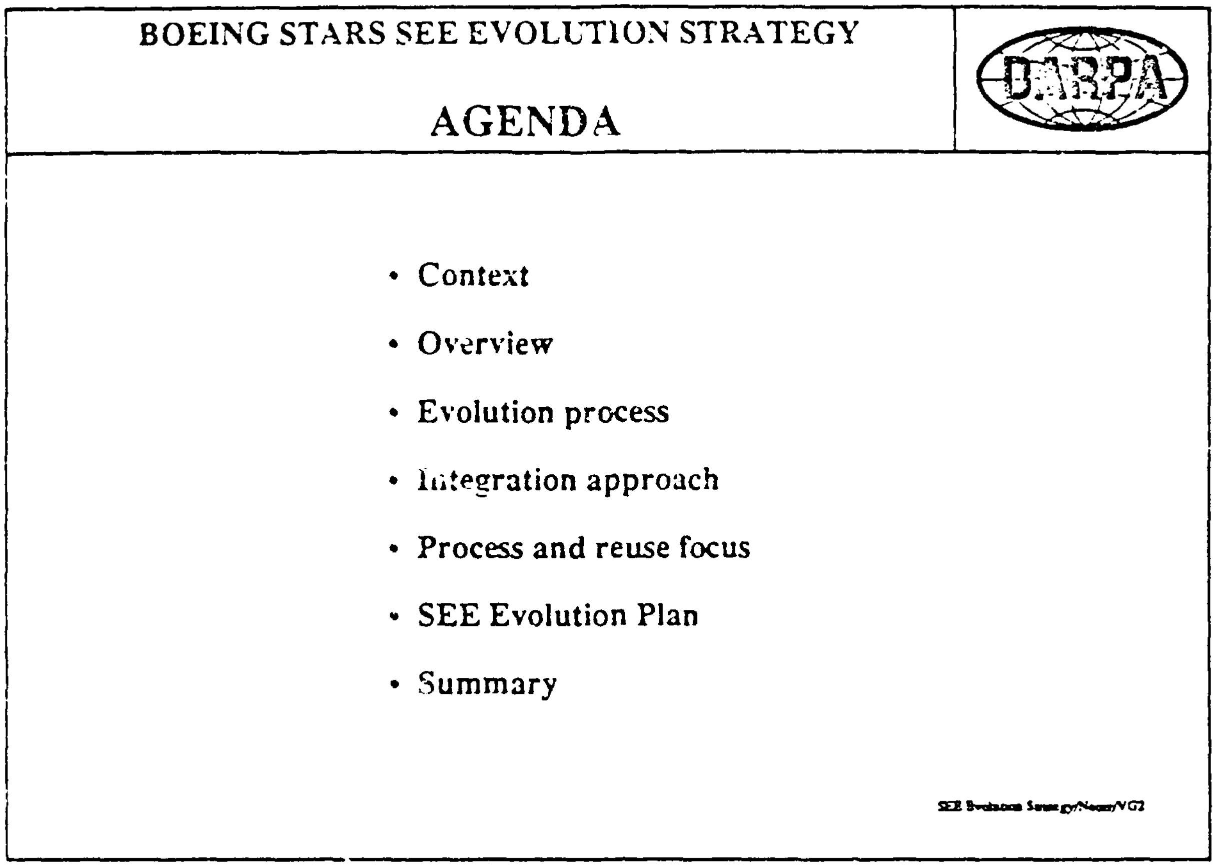


STARS '91
BOEING STARS SEE EVOLUTION STRATEGY

John Neorr
The Boeing Company
04 December 1991
(206) 773-5907
neorr@stars.boeing.com

SEE Evolution Strategy/Neorr/VG1

This presentation shows the Boeing strategy for evolving their STARS SEE. Several elements of our strategy are similar to those of IBM and Unisys. This is to be expected given the specifications developed jointly by the three primes and given the current state of technology. One unique element of our strategy is our alliance with the Digital Equipment Company (DEC) and our use of their COHESION product set.



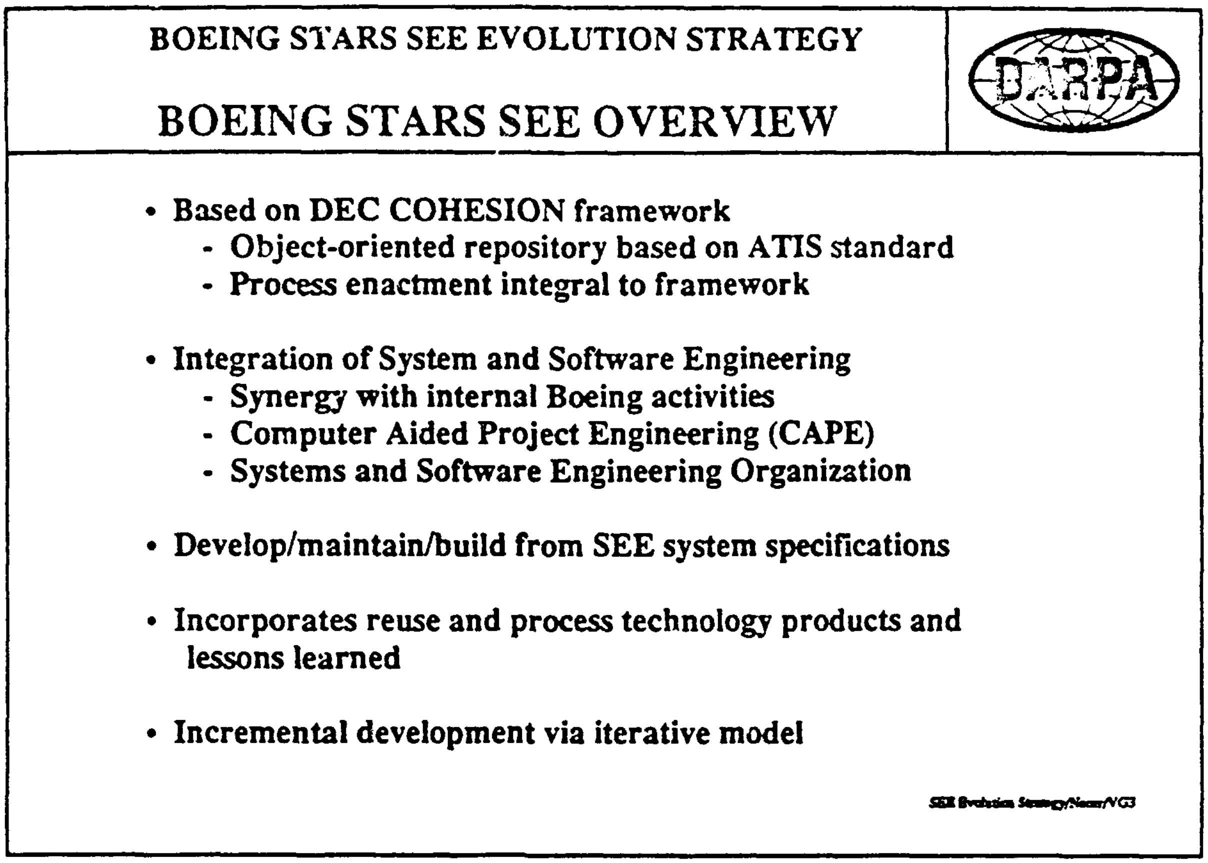
BOEING STARS SEE EVOLUTION STRATEGY

AGENDA

DARPA

- Context
- Overview
- Evolution process
- Integration approach
- Process and reuse focus
- SEE Evolution Plan
- Summary

SEE Evolution Strategy/Noon/VG2



BOEING STARS SEE EVOLUTION STRATEGY

BOEING STARS SEE OVERVIEW

- Based on DEC COHESION framework
  - Object-oriented repository based on ATIS standard
  - Process enactment integral to framework
- Integration of System and Software Engineering
  - Synergy with internal Boeing activities
  - Computer Aided Project Engineering (CAPE)
  - Systems and Software Engineering Organization
- Develop/maintain/build from SEE system specifications
- Incorporates reuse and process technology products and lessons learned
- Incremental development via iterative model

SEE Evolution Strategy/Name/VG3

As noted in the introduction, the Boeing STARS SEE is based on the DEC COHESION product set. This includes their framework product, CDD/Repository. CDD/Repository is object-oriented and based on "A Tool Integration Standard" (ATIS). This standard provides for a self-defining type hierarchy, a set of services for managing data within the repository, and a set of services dealing with method (or process) enactment.

Another feature of the Boeing SEE is its support to both systems and software engineering. This integrated approach to systems and software engineering is synergistic with Boeing's own approach to system development. For example, one effort underway at Boeing is creation of a support environment called the Computer Aided Project Engineering (CAPE) environment. Lessons learned on CAPE and domain specific requirements from CAPE will assist in the evolution of the Boeing STARS SEE.

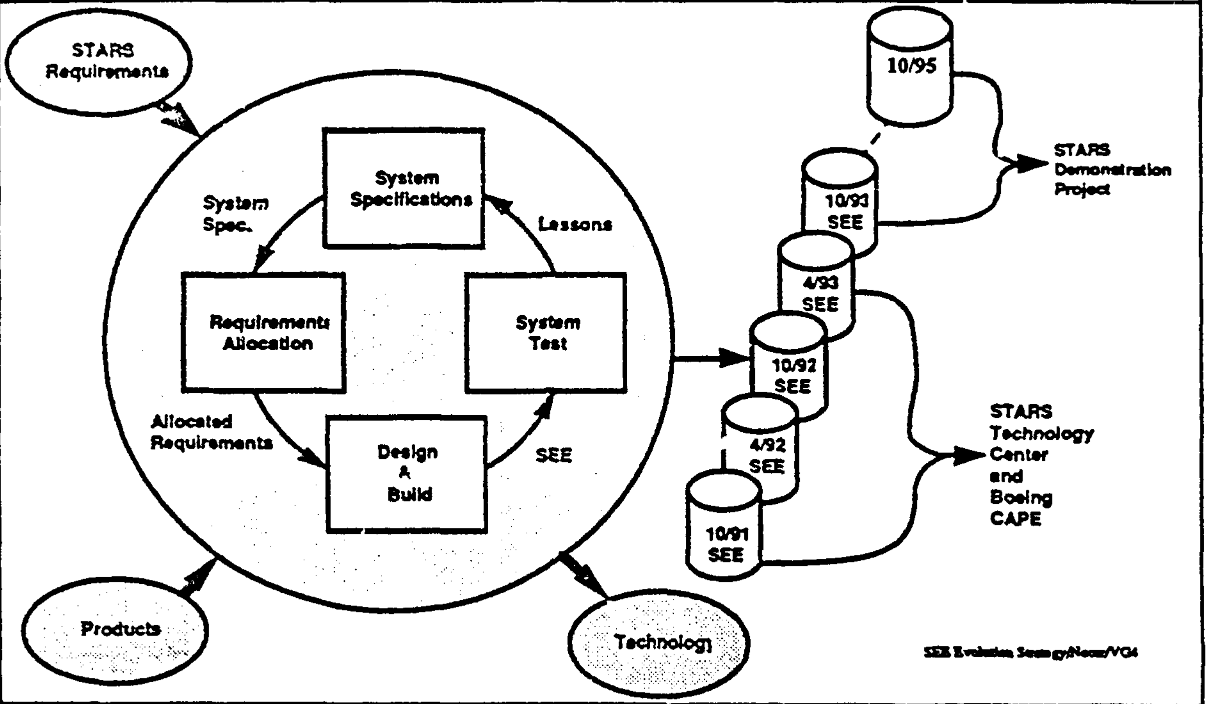
CAPE is being developed by the Boeing Defense and Space Group's Systems and Software Engineering Organization, the same organization that staffs STARS. This organization provides policies, procedures, and tools to support an integrated approach to systems and software engineering. These procedures are compatible with Government standards such as MIL-STD-499A and DoD-STD-2167A and, as such, are similar to procedures found in other large aerospace firms. They should prove useful as test cases for adapting the SEE to specific processes.

The Boeing SEE will be created based on Boeing SEE specifications derived from specification documents developed by STARS joint activity groups. A chart later in this presentation depicts the relationships among the various STARS documents.

In addition to commercial, off-the-shelf (COTS) tools, the Boeing STARS SEE will incorporate products and lessons learned from their own work on process and reuse and from the work of Unisys and IBM. Note that our strategy in building a SEE is to incorporate and integrate tools developed elsewhere. Our SEE will be built incrementally with periodic releases. Each new release will have additional functionality and will provide opportunities for demonstrations and feedback.

BOEING STARS SEE EVOLUTION STRATEGY

THE SEE EVOLUTION PROCESS



SEE Evolution Strategy/Notes/VG4

Like the IBM and Unisys SEEs, the Boeing STARS SEE will evolve with time. As you can see from this chart, we plan periodic releases as the SEE evolves. There will be a release every six months, each release containing more functionality. The release in October of 1993 will be used on a demonstration project to determine the effectiveness of a SEE embodying STARS technology. It should be fully functional at that time. Depending on the feedback, subsequent changes and new releases will be made through 1995.

The large circle in the middle of this chart represents our SEE development process. Note the two major inputs to this process. STARS requirements are all of those requirements emanating from the joint activity groups, from the demonstration project, and from ad-hoc sources. These requirements are distilled into a Boeing SEE specification document as shown on a later chart. Products used by this process come from the commercial sector and come from work being done on other STARS tasks in the area of reuse and process.

A very important component of the STARS program is technology transfer. It order to bring about significant productivity improvements in system development and maintenance we need to transition new products, ideas, and other technology into Government and Industry.

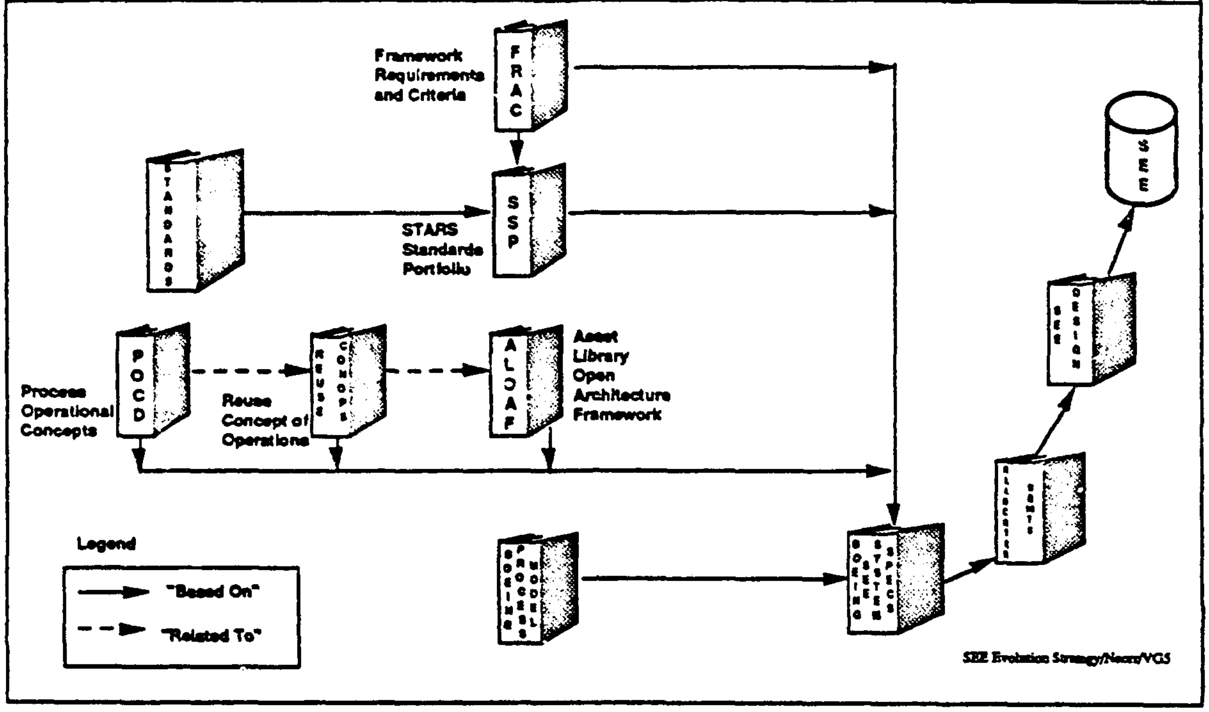
Inside the big circle, which represents the process for building the SEE, are four sub-processes. Beginning with the System Specification sub-process and proceeding counter-clockwise, each one of the four sub-processes tends to be sequentially executed. Of course, we don't live in a perfect world, so sometimes a subprocess is revisited during a single six-month increment. Major outputs from each phase (sub-process)are shown flowing from one phase to another.

A synopsis of the subprocesses is as follows:

1) The system specification process basically "distills" all of the various requirements into a single Boeing SEE system specification.
2) This specifications is analyzed and a set of requirements to be addressed in the design and build phase is allocated.
3) The design and build phase takes the allocated requirements, builds a SEE and hands it over for testing.
4) Once the system is tested, it is made available to the "outside world" (as depicted on the chart). Lessons are also made available for the next pass around the loop.

BOEING STARS SEE EVOLUTION STRATEGY

A SPECIFICATION-BASED DEVELOPMENT


Framework Requirements and Criteria
FRAC
STANDARDS
STARS Standards Portfolio
SSP
SEE
Process Operational Concepts
POCD
Reuse Concept of Operations
REUSE CONOPS
ALOAF
Asset Library Open Architecture Framework
SEE DESIGN
BOEING PROCESS MODEL
BOEING SEE SYSTEM SPECS
Legend
"Based On"
"Related To"

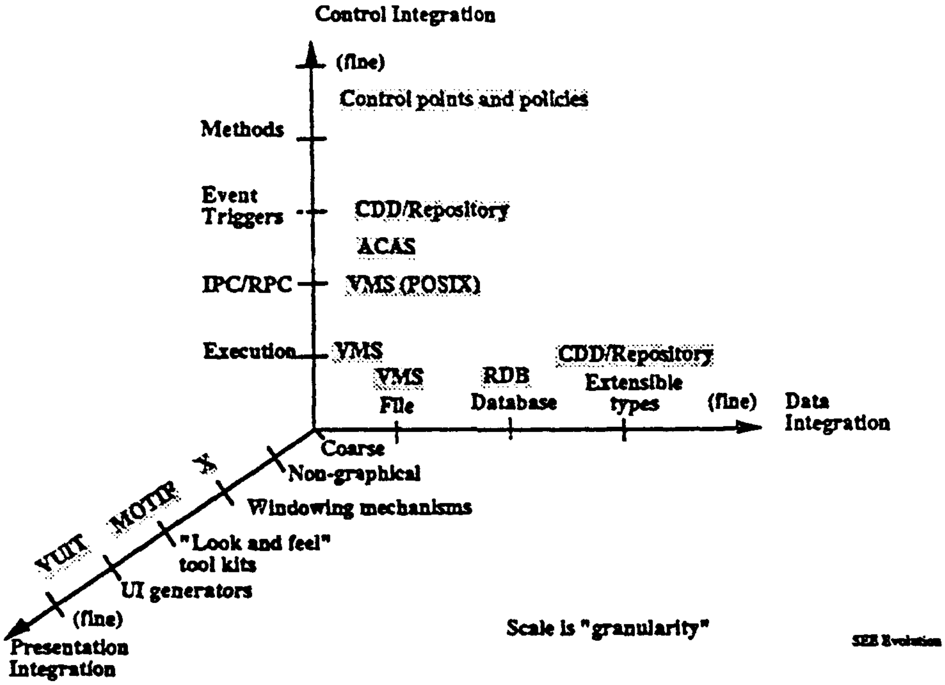
SEE Evolution Strategy/Neon/VGS

As illustrated on the preceding chart, the develop process is predicated on a set of requirements from various sources. This chart shows how those documents are used to direct the development of the STARS SEE. As noted on the legend, the solid arrow can be used to show what document is based on what other document, this typically has meant a precedence relationship. That is, the Framework Requirements and Criteria (FRAC) preceded the STARS Standards Portfolio (SSP); standards preceded their incorporation into the SSP. On the other hand, the Process Operational Concept Document (POCD) and the Reuse Concept of Operations (CONOPS) document, have been developed in parallel. The dashed arrow shows that there is some relationship between the two documents, but not a strict dependency.

The FRAC, SSP, Reuse CONOPS, and Asset Library Open Framework (ALOAF) documents have been developed by STARS joint activity groups. Standards been developed by various standards bodies, and the Boeing Process Model is specific to Boeing. The SEE System Specification is unique to the Boeing STARS SEE.

BOEING STARS SEE EVOLUTION STRATEGY

COHESION INTEGRATION APPROACH



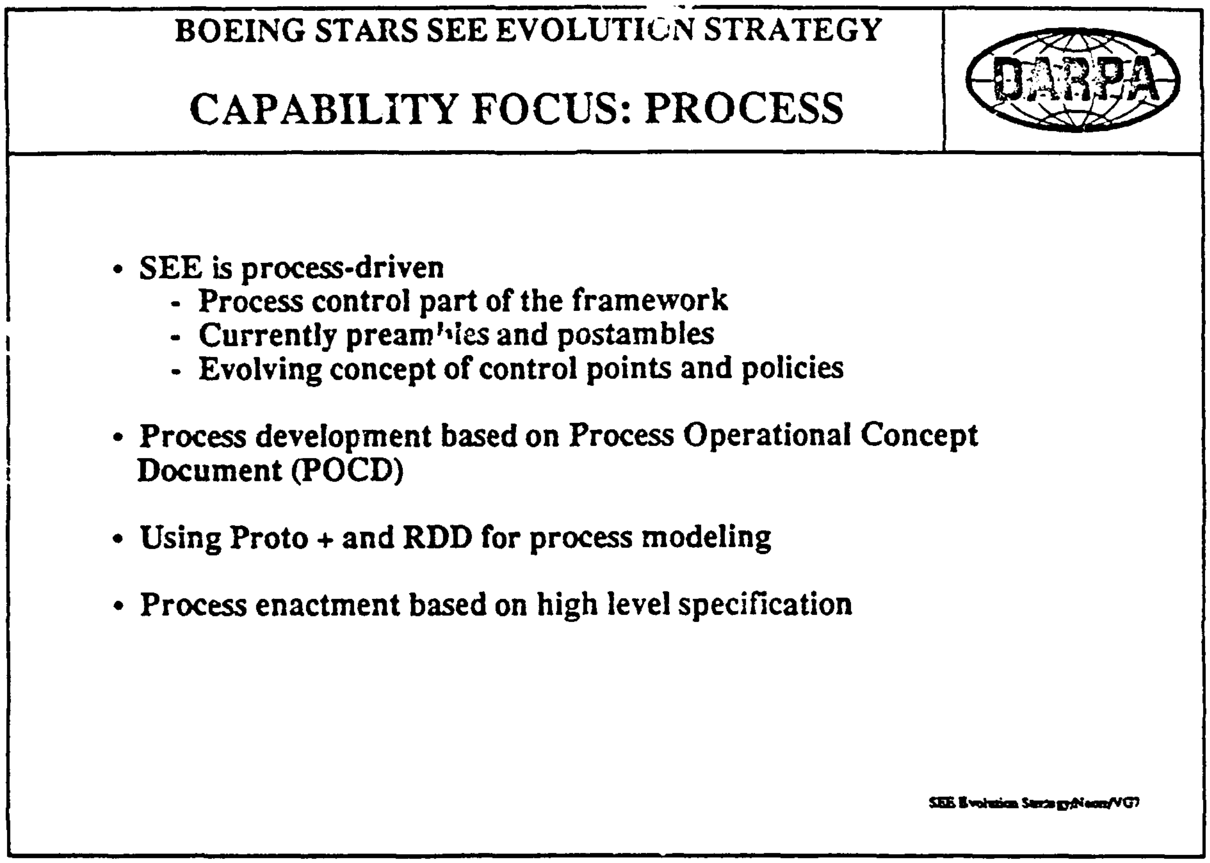
SEE Evolution Strategy/Nov/VG6

The chart, which shows the three dimensions of integration, also shows how COHESION products map to the various dimensions. It is important to note that the products shown here are FRAMEWORK products. That is, this chart does not depict tools. Rather, it depicts those services that enab‹ a SEE to be built and provide some granularity of integration. The concept of fine versus coarse granularity is reflected on the chart. For instance, data integration between two tools is considered to be of coarse granularity if they communicate by exchanging data via files. On the other hand, fine granularity data integration between two tools is realized when they share data via a common repository.

In terms of data integration, CDD/Repository provides the ability to have fine integration. This allows what we call a repository-centered SEE. A later chart will provide more information as to what types of tools will populate our SEE. Our current environment which is being demonstrated at STARS91 already supports fine-grained integration. The trick is continuing to populate our SEE with tools which will share a common information model and provide adequate performance.

Aspects of control integration are supported via the COHESION products, CDD/Repository and Application Control Architecture Services (ACAS). ACAS is designed to provide integration of third party tools that do not share a common repository. ACAS provide a means to integrate tools via message passing. CDD/Repository, which implements A Tool Integration Standard (ATIS), provides a feature called preambles and postambles. This capability will cause procedures respectively to be invoked prior to invocation of a method (preamble) and/or after invocation of a method (postamble).

Granularity of presentation integration is shown on the bottom left scale. Like IBM and Unisys, Boeing will be using the STARS standard interface, Motif. The DEC COHESION product, VUIT, is a user interface generator, that generates Motif-based interfaces. This product will be incorporated into the Boeing SEE as required.



BOEING STARS SEE EVOLUTICN STRATEGY

CAPABILITY FOCUS: PROCESS

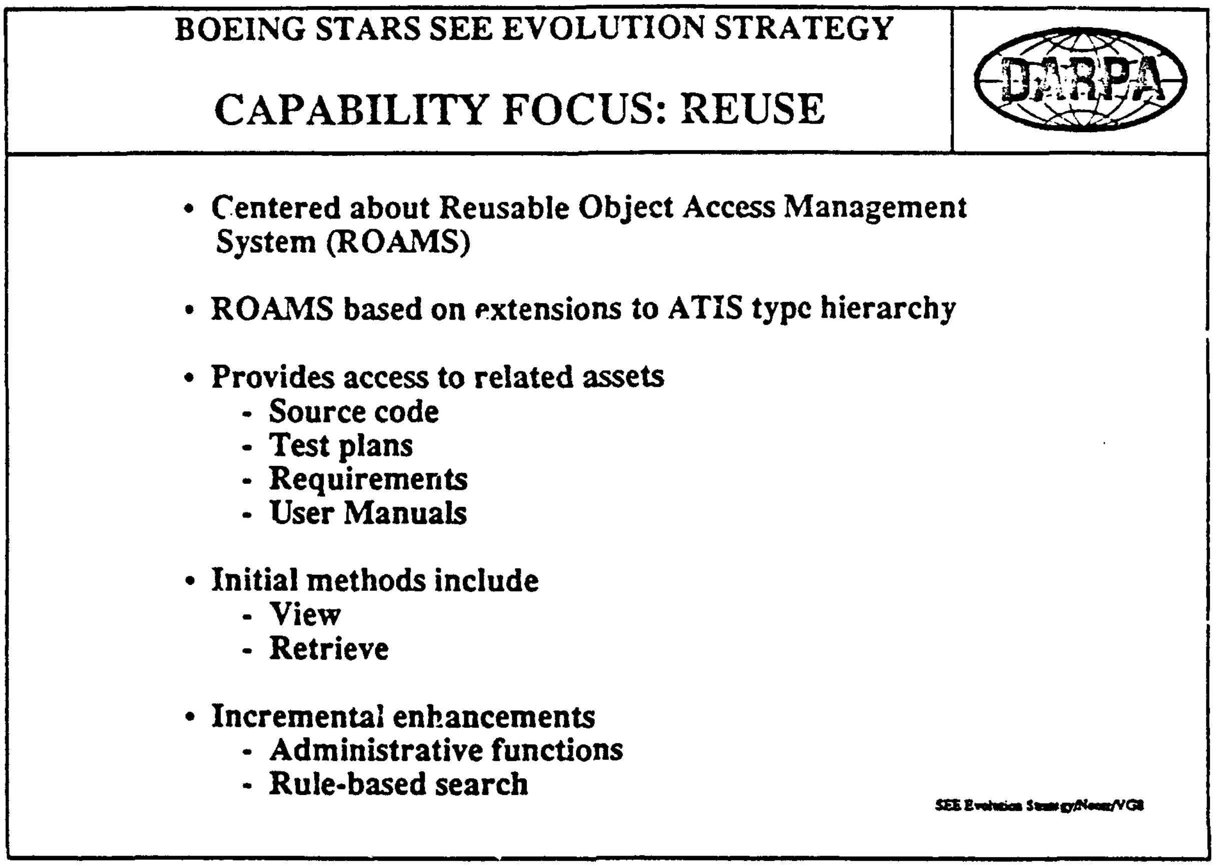
- SEE is process-driven
  - Process control part of the framework
  - Currently preamᵇˡes and postambles
  - Evolving concept of control points and policies
- Process development based on Process Operational Concept Document (POCD)
- Using Proto + and RDD for process modeling
- Process enactment based on high level specification

SEE Evolution StrategyNeon/VG7

One of the key features of the STARS SEEs is that they are process-driven. That is, the SEE has built into it, some knowledge of the user's process so that it behaves as the user would expect. This may be in the automatic invocation of processes, the ordering of processes, built-in audits, process metrics, and other process-related features. Through the use of COHESION, certain features of process control are built into the framework. That is, the CDD/Repository provides not only for the invocation of methods, it provides for preambles and postambles. A preamble is simply a user-defined program that executes prior to a method, and a postamble is a user-defined program that executes after a method. Currently, working as a subcontractor to Boeing, Honeywell is evolving a technique using control points and policies to control method invocation. Control points and policies provide a rich set of process control mechanisms that can be used to enhance the capabilities of CDD/Repository's preambles and postambles.

As shown on an earlier chart, the incorporation of process technology would be predicated on the specifications found in the Process Operational Concept Document (POCD). On functions specified in this document is process modelling. Currently experiments are being conducted using Proto+ and RDD for process modeiling. As these tools prove effective for process modelling, they will be incorporated into the Boeing SEE.

Currently control points and policies have to be written in C or Ada, based on a some sort of high level process model or specification. Thus the translation of a process model to enactable control points and policies is cumbersome and labor intensive. As a result, Honeywell not only is building mechanisms for process enactment using control points and policies, they are investigating ways in which a high level process specification can be compiled directly into control points and policies.



BOEING STARS SEE EVOLUTION STRATEGY

CAPABILITY FOCUS: REUSE

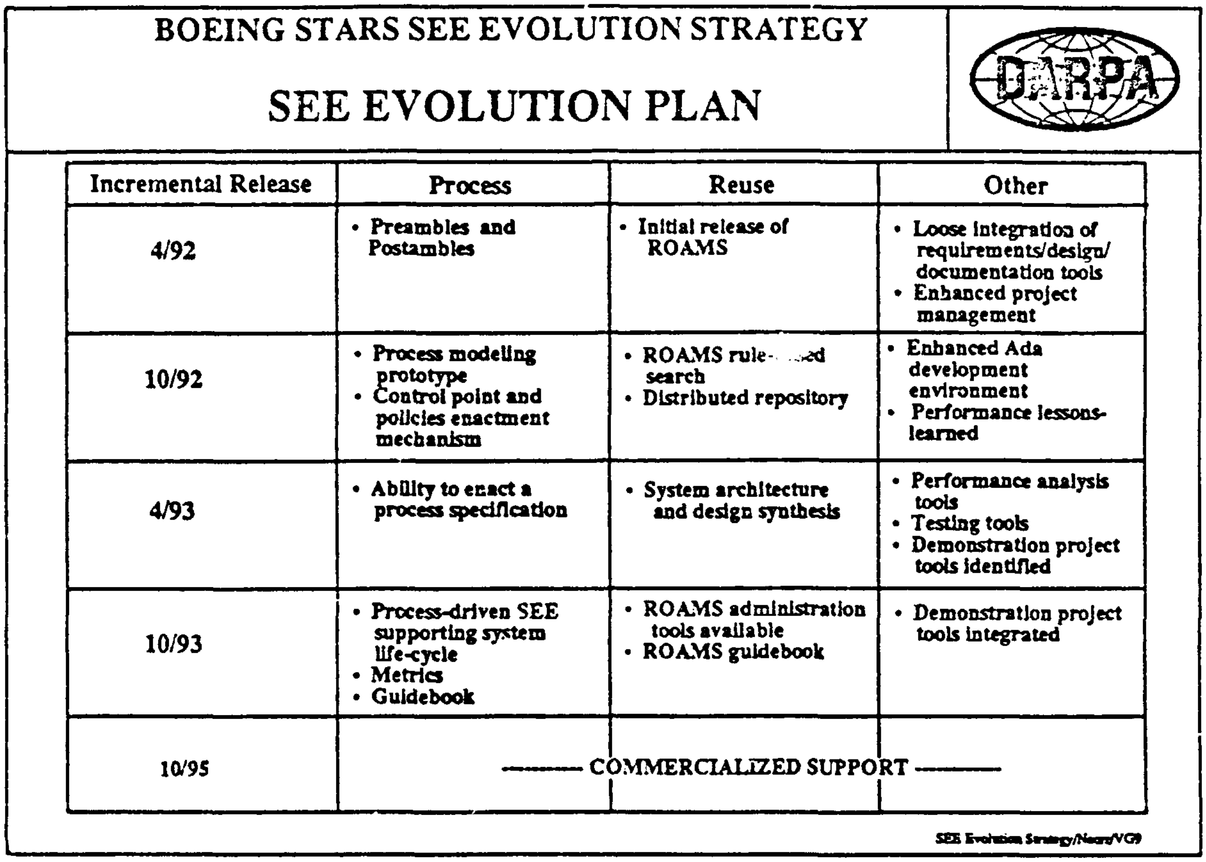
- Centered about Reusable Object Access Management System (ROAMS)
- ROAMS based on extensions to ATIS type hierarchy
- Provides access to related assets
  - Source code
  - Test plans
  - Requirements
  - User Manuals
- Initial methods include
  - View
  - Retrieve
- Incremental enhancements
  - Administrative functions
  - Rule-based search

SEE Evolution Strategy/Notes/VG8

Reuse capabilities provided by the Boeing SEE are centered about a tool called the Reusable Object Access Management System (ROAMS). This system takes advantage of the extensible ATIS type hierarchy in CDD/Repository. Using the concepts inherent the object-oriented database, KOAMS provides the capability to define an abstract object called an "asset" (refer to your proceedings in the Reuse Track for more information on reuse assets). An asset object then has several sub-types such as: source code, test plans, requirements, user manuals, and so on. Because the system is extensible it is a simple matter to extend the type hierarchy to create new types of related assets.

Our initial implementation, which can be seen during the demonstration period, supports the capability to view and retrieve assets. The look and feel of ROAMS is consistent with other framework tools because they are all based on a standard CDD/Repository navigator which uses Motif widgets.

ROAMS will continue to be enhanced as our SEE evolves. These enhancements will include administrative functions for asset check-in and check-out and functions for rule-based searching.



BOEING STARS SEE EVOLUTION STRATEGY

SEE EVOLUTION PLAN

Incremental Release	Process	Reuse	Other
4/92	• Preambles and Postambles	• Initial release of ROAMS	• Loose integration of requirements/design/ documentation tools • Enhanced project management
10/92	• Process modeling prototype • Control point and policies enactment mechanism	• ROAMS rule-based search • Distributed repository	• Enhanced Ada development environment • Performance lessons-learned
4/93	• Ability to enact a process specification	• System architecture and design synthesis	• Performance analysis tools • Testing tools • Demonstration project tools identified
10/93	• Process-driven SEE supporting system life-cycle • Metrics • Guidebook	• ROAMS administration tools available • ROAMS guidebook	• Demonstration project tools integrated
10/95	——— COMMERCIALIZED SUPPORT ———		

SEE Evolution Strategy/Neary/VG9

The plan for evolution of the Boeing SEE is shown on this chart. It shows the functions, features, and deliverable that will be available at six month intervals. It should be noted that this is the CURRENT plan. Our evolutionary development plan is intended to provide us with the flexibility to accommodate change as we discover more about process, reuse and SEE tools and technology.

In the area of process we are currently using preambles and postambles and will continue to do so in April of 1992. Subsequent releases cf the SEE will provide tools for process modeling (10/92) and process enactment directly from a high level specification (4/93). By October of 1993, the SEE will contain a process specification for the system development life cycle and tools will behave accordingly. Also at that time, metrics will be provided as well as a guidebook for tailoring the SEE's process.

As noted earlier, ROAMS will develop incrementally. The initial release of ROAMS (currently being demonstrated at STARS 91) will be included in our 4/92 SEE. It will probably have minor enhancements at that time depending on lessons-learned in the interim. Subsequent releases of ROAMS will support rule-based searches and distributed repositories (10/92). This will be followed (4/93) by integrated capabilities for synthesizing system architectures or designs based on domain analysis. By October 1993, tools will be available for evaluating assets for acceptance into ROAMS. Other capabilities for easily classifying assets and getting them into and out of the repository will also be available.

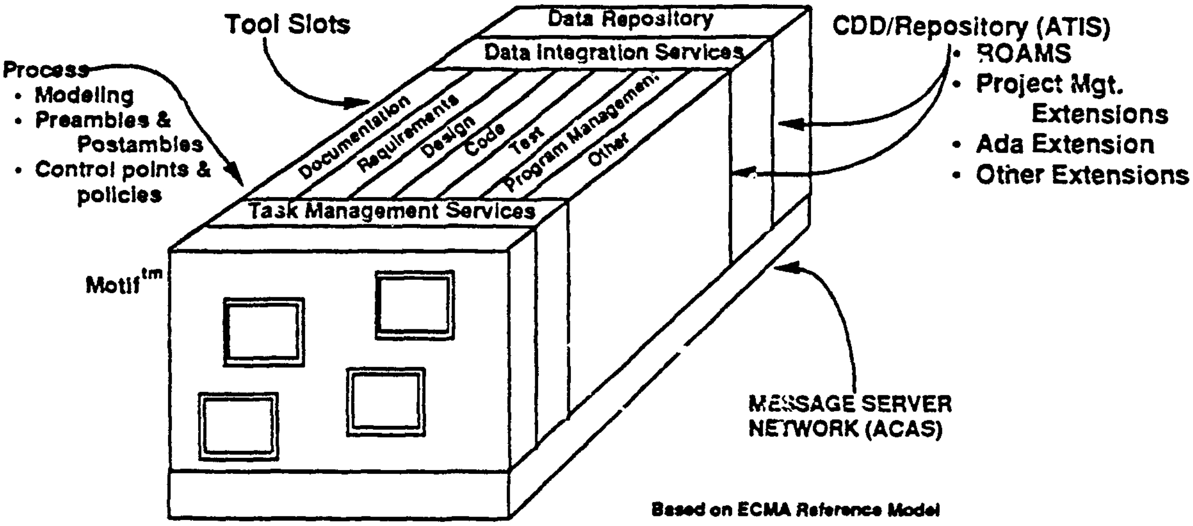
Tools in support of specific functional areas will be incrementally added to the SEE. This chart purposely does not identify specific products - rather, it identifies the general functional areas that will be supported across time. We believe the lessons learned about integration of tools into a repository-based is more important than the particular tool being integrated. We need to be able to develop strategies for integrating third party tools and for developing common information models for repository-centered SEEs. The lessons we learn and the technology we develop can then be transitioned into the commercial marketplace.

'Ve anticipate having a fully populated SEE by October 1993 in order to support a STARS demonstration project.

The demonstration project will continue through October 1995, and (as the chart indicates) the final objective is to have the tools and technologies in the SEE supported by the commercial marketplace.

BOEING STARS SEE EVOLUTI(ON STRATEGY

POPULATED SEE OVERVIEW



Based on ECMA Reference Model

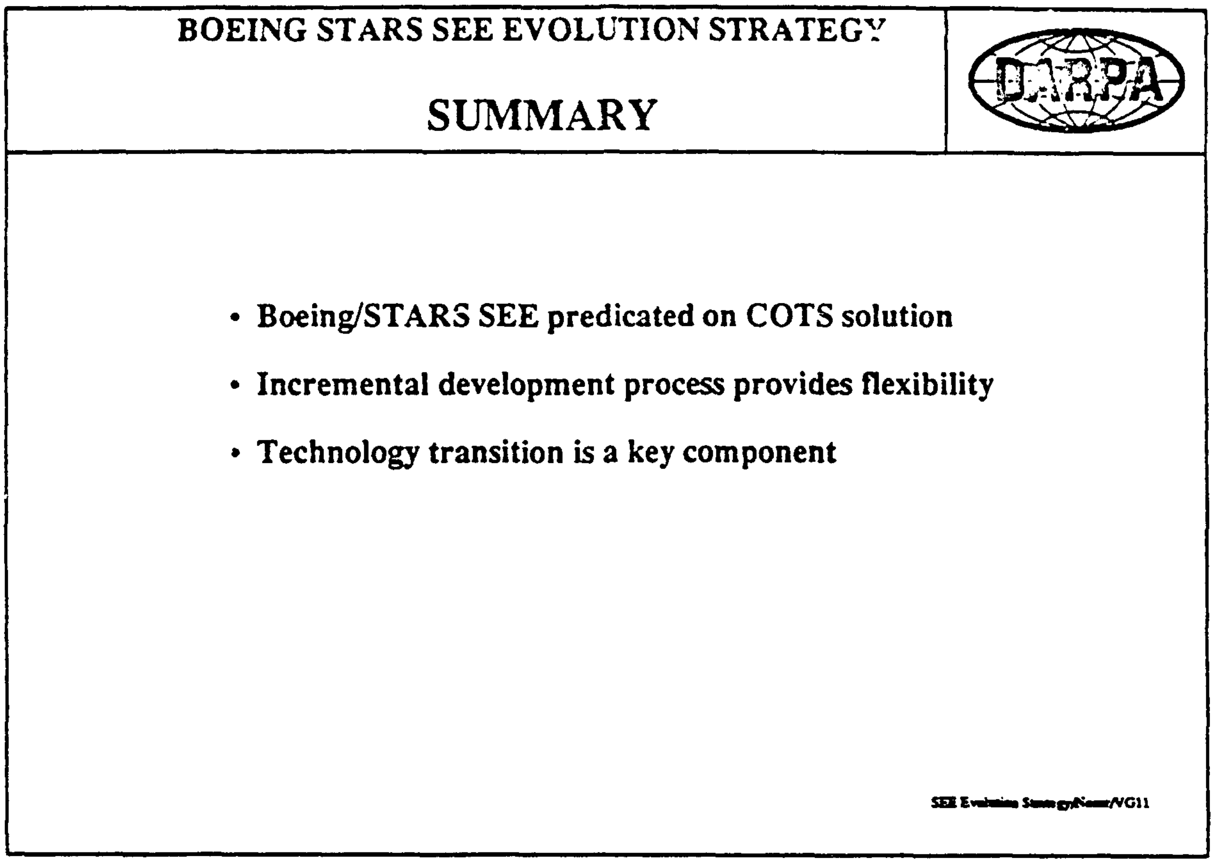
SEE Evolution Strategy/Neem/V010

This chart is a view of the fully populated Boeing STARS SEE, showing where DEC COHESION products fit and where other tools and products fit. This model is based on the European Computer Manufacturing Companies (ECMA) framework reference model and is used for depicting and describing framework services and their relationships.

A data repository and associated data integration services are provided by the CDD/Repository product which implements the ATIS standard. CDD/Repository provides a self-defining type hierarchy that allows the definition of data and meta-data alike. The type hierarchy can be extended to allow the definition of domain specific types together with their attributes, relationships, and methods. The current release of the Boeing SEE has been extended to accommodate the classifying of some types of assets and to allow the integration of project management and program development data.

Task management services are supported by both ACAS and CDD/Repository. These services will be those used to implement process invocation and control using preambles, po^tambles, control points, and policies. The user interface is supported by Motif.

This chart also shows tool slots populated with those types of tools that we plan on having in our environment. Again, we have avoided using specific product names. What we are primarily concerned about are the techniques used for tool integration and the common information model in the repository.



BOEING STARS SEE EVOLUTION STRATEGY

SUMMARY

- Boeing/STARS SEE predicated on COTS solution
- Incremental development process provides flexibility
- Technology transition is a key component

SEE Evolution Strategy/Name/VG11

In summary, our strategy is predicated on COTS solutions. Our alliance with DEC and our use of their COHESION framework products is a key part of this strategy. Our incremental development strategy will enable us to be flexible as requirements change, new lessons are learned, and new products become available. Finally, it is important to understand that creation of a SEE is not the primary purpose of our work. Rather, our work, like the rest of the STARS effort, is focused on technology transfer. Our intent is to provide Government and Industry technology for accelerating improvements in software development and maintenance - improvements that will enable us to build better systems quicker and cheaper.

Acronymns used in this Presentation

ACAS - Application Control Architecture Services

ALOAF - Asset Library Open Framework

ATIS - A Tool Integration Standard

CAPE - Ccmputer Aided Project Engineering

CDD - Common Data Dictionary

CONOPS - Concept of Operations

COTS - Commercial, Off-The-Shelf

DEC - Digital Equipment Company

ECMA - European Computer Manufacturers Association

FRAC - Framework Requirements and Criteria

IBM - International Business Machnies

POCD - Process Operational Concept Document

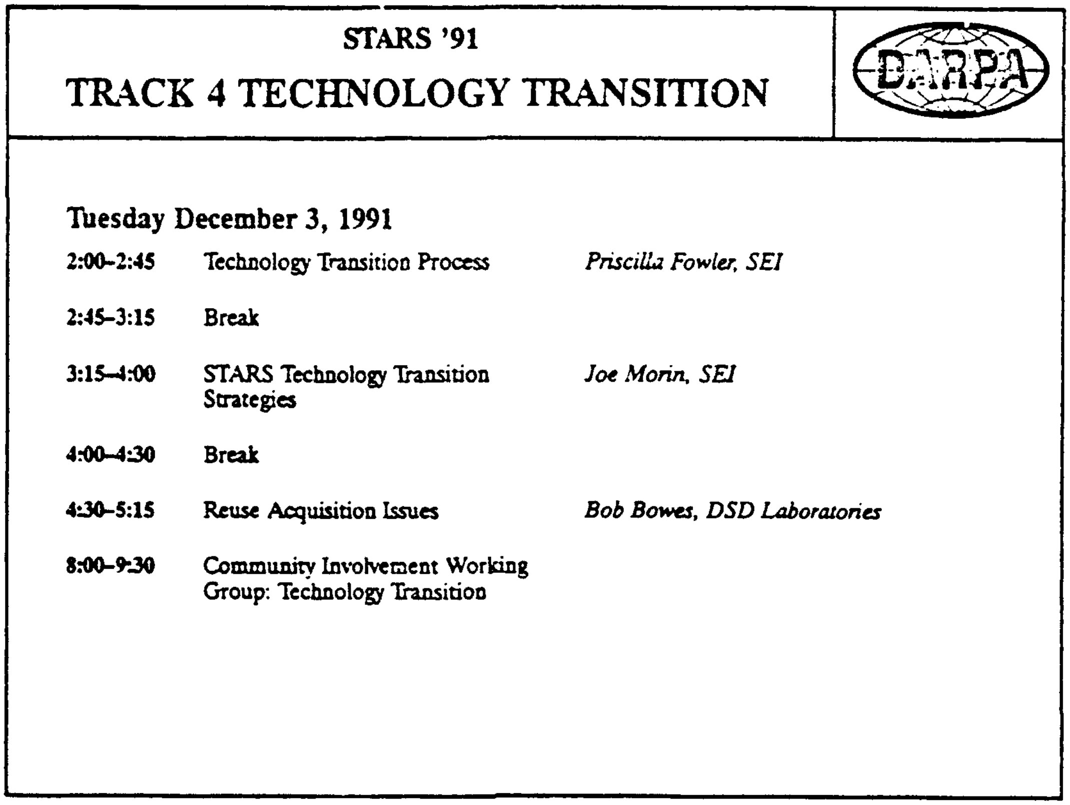
RDD - Requirements Driven Design

ROAMS - Reusable Object Access Management System

SEE - Software Engineering Environment

SSP - STARS Standards Portfolio

STARS - Software Technology for Adaptable, Reliable Systems

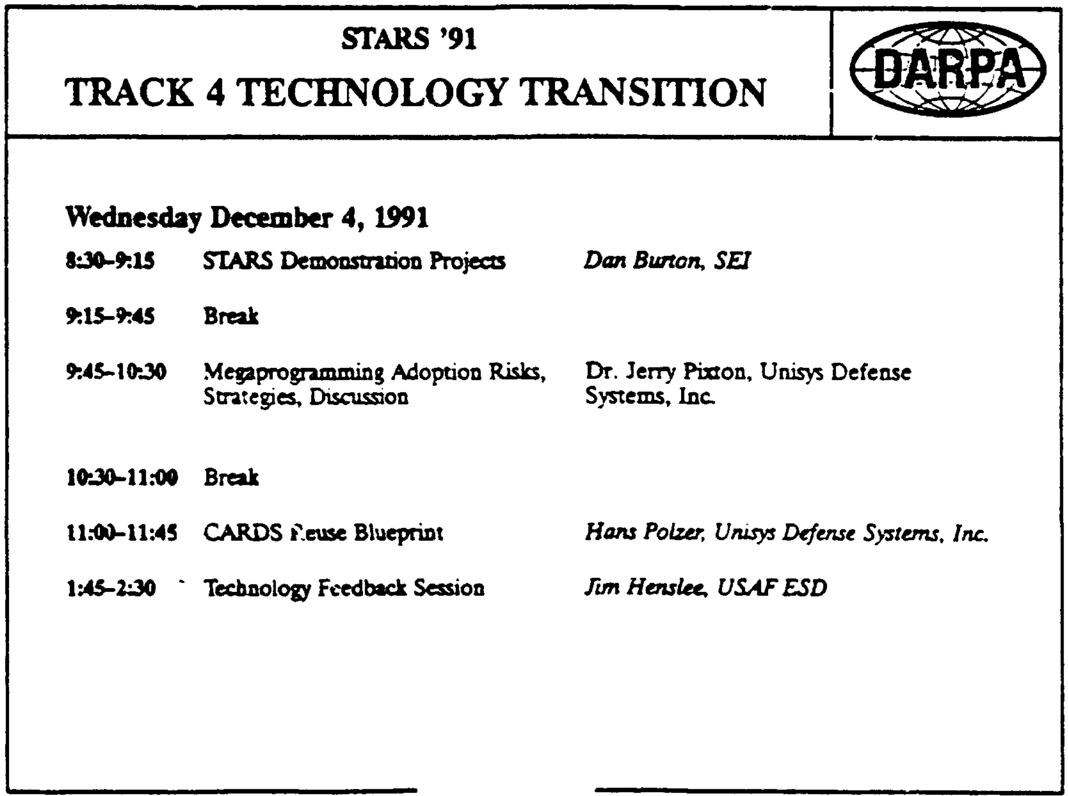


STARS '91

TRACK 4 TECHNOLOGY TRANSITION

Tuesday December 3, 1991

2:00–2:45	Technology Transition Process	*Priscilla Fowler, SEI*
2:45–3:15	Break	
3:15–4:00	STARS Technology Transition Strategies	*Joe Morin, SEI*
4:00–4:30	Break	
4:30–5:15	Reuse Acquisition Issues	*Bob Bowes, DSD Laboratories*
8:00–9:30	Community Involvement Working Group: Technology Transition	

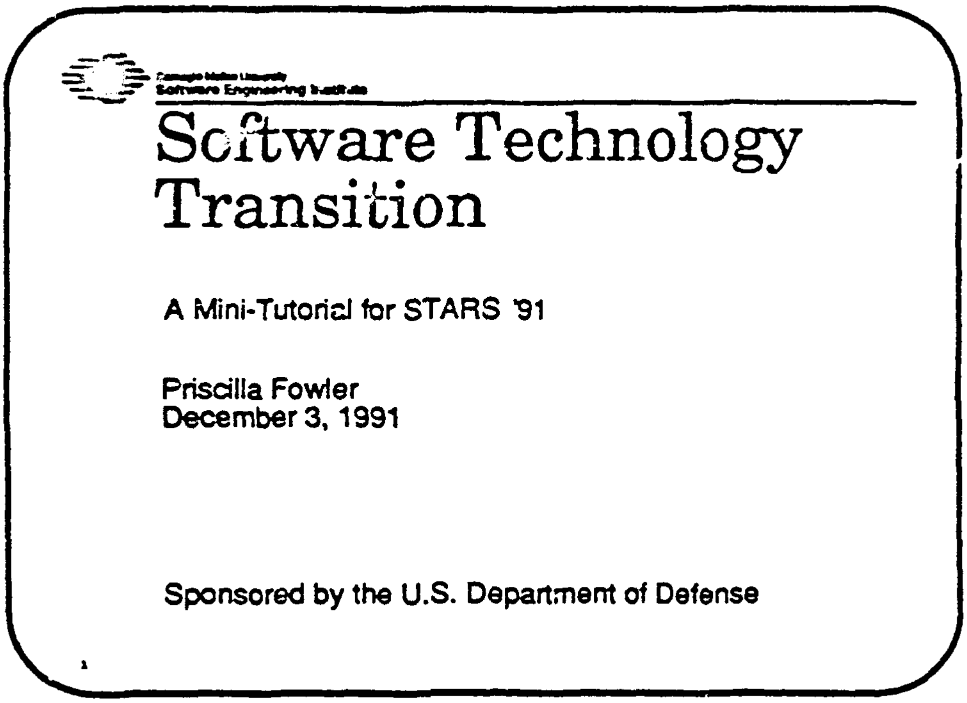


STARS '91

TRACK 4 TECHNOLOGY TRANSITION

Wednesday December 4, 1991

8:30–9:15	STARS Demonstration Projects	*Dan Burton, SEI*
9:15–9:45	Break	
9:45–10:30	Megaprogramming Adoption Risks, Strategies, Discussion	Dr. Jerry Pixton, Unisys Defense Systems, Inc.
10:30–11:00	Break	
11:00–11:45	CARDS Reuse Blueprint	*Hans Polzer, Unisys Defense Systems, Inc.*
1:45–2:30	Technology Feedback Session	*Jim Henslee, USAF ESD*



Carnegie Mellon University
Software Engineering Institute

Software Technology Transition

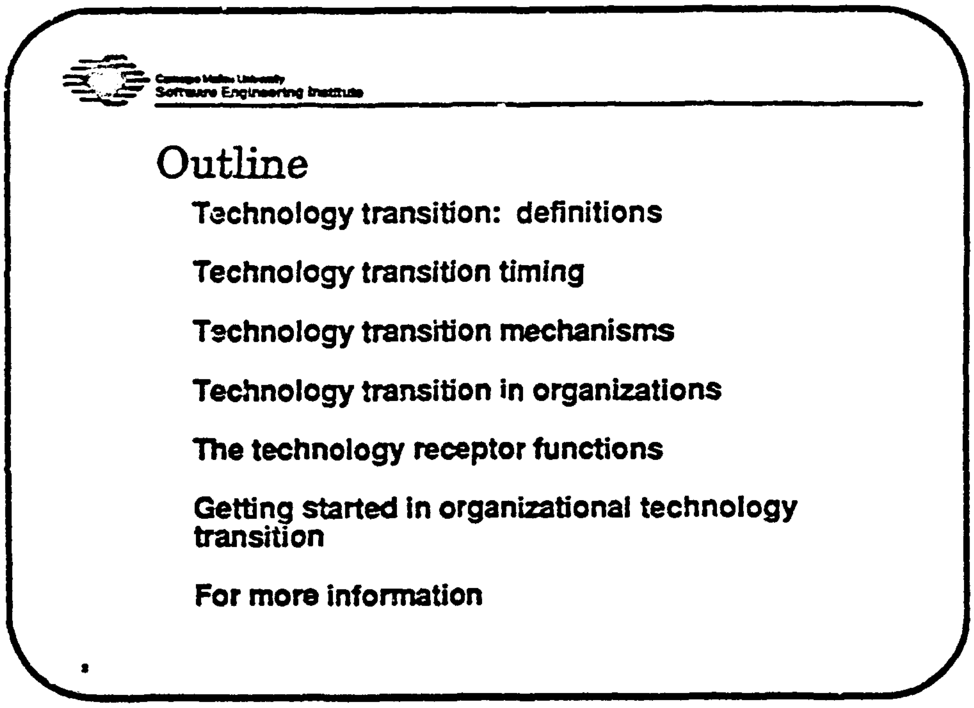
A Mini-Tutorial for STARS '91

Priscilla Fowler
December 3, 1991

Sponsored by the U.S. Department of Defense

1

Notes



Carnegie Mellon University
Software Engineering Institute

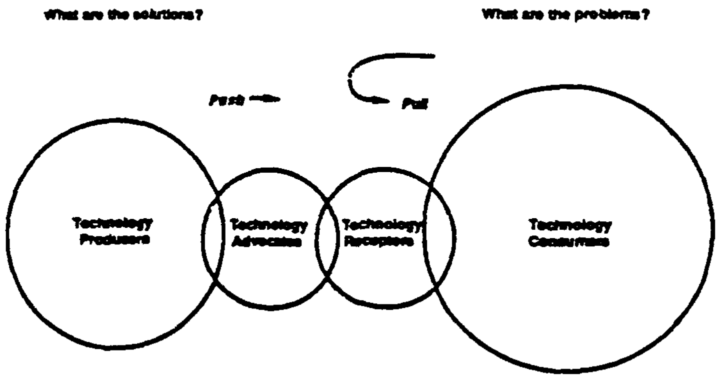
Outline

- Technology transition: definitions
- Technology transition timing
- Technology transition mechanisms
- Technology transition in organizations
- The technology receptor functions
- Getting started in organizational technology transition
- For more information

Notes

Carnegie Mellon University
Software Engineering Institute

A Maturation Transaction



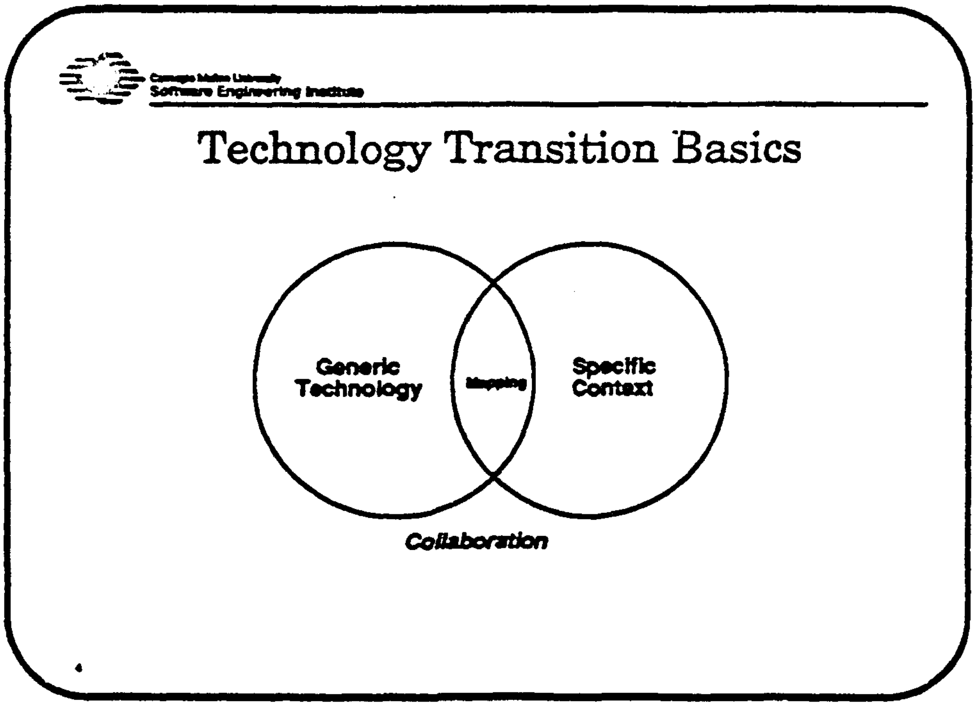
3

Notes

This model is derived from the change agent model used in diffusion research. Change agents provide a link between producers and consumers, helping to translate the meaning and implications of the technology to the potential users in the users' own terminology. This diagram depicts roles in one maturation transaction. Players in each bubble gather and process information, add value, and translate the results for their constituents. Roles are not always equated with specific individuals. The players within each bubble address issues relevant to their context:

- organizational issues
- innovation issues
- commitment issues

Most work to date focuses on the "push" side of the model. Advances in marketing and dissemination in general have led this effort. It is important to note that the models and issues are the same for entities across the diagram. The later sections of this tutorial focus on improving the transition capabilities on the "pull" side of the equation.



Carnegie Mellon University
Software Engineering Institute

Technology Transition Basics

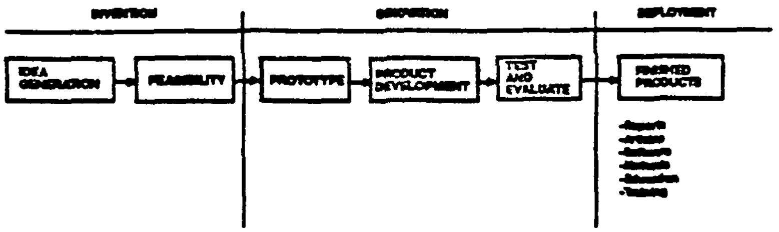
Notes

Any technology is "new" in a context where it hasn't been used before. Technology transition, at the most basic level, means taking a "generic" technology and mapping it into a specific organizational context. Technology is "generic" because of the assumptions made by its builders about the contexts of its potential users. Even with good market analyses or requirements analyses, there is never a perfect match between these assumptions and the specifics of the context in which it will be used. And those who know the technology seldom know enough about the specific context to do this mapping without help. Thus we use the model of collaboration between technology experts and context experts, i.e., users. Working together, people with these different perspectives can make the match, most likely adapting both context and technology in that process (this is the concept of "mutual adaptation", as described by Dorothy Leonard-Barton's work on innovations such as expert systems [Leonard-Barton 88]).

Another way of describing what happens here is to say that this mapping process requires orchestrating the actions that move people up the commitment curve.

Carnegie Mellon University
Software Engineering Institute

Technology Development Process

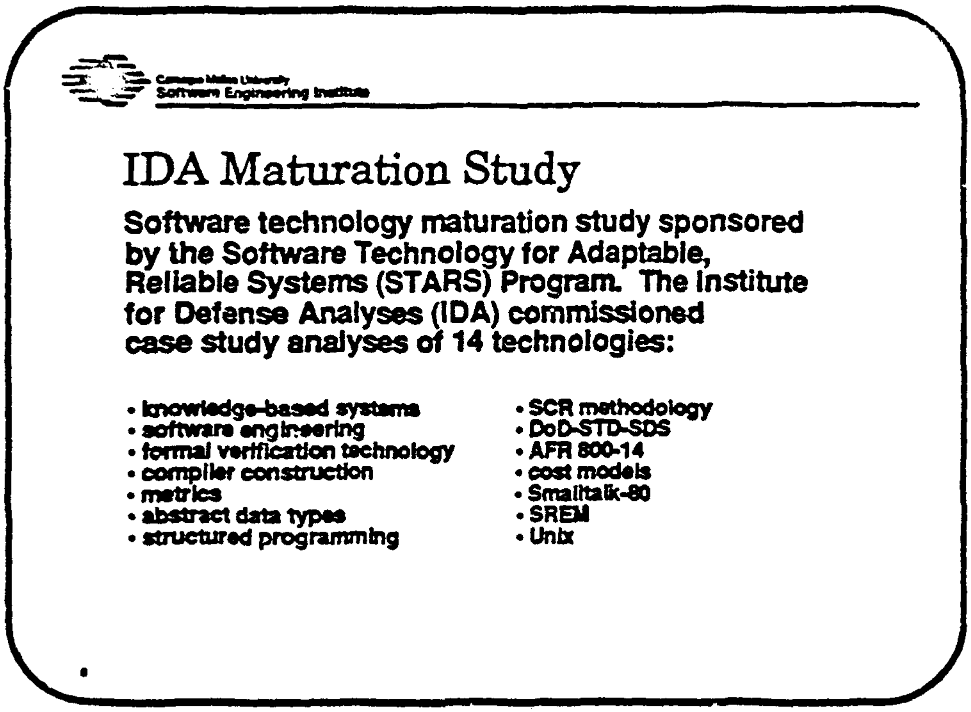


Notes

This generalized research and development process is typical of those found in the R&D management literature. In this model the feedback loops are omitted, just as they are often omitted in the "waterfall" model of software development.

Most often this model is used to describe the entire technology development process from raw idea to finished product. We are using the model a little differently. Few technologies evolve from basic science to finished product within one organization. We'd like to use this model as a process that recurs within different organizations throughout the technology maturation life cycle. The literature on the diffusion of innovations and computer-human interaction supports this idea of "reinvention." The stages in the diagram have different meanings depending on the missions, skills, and motivations in each organization. This process recurs until either the idea dies or it reaches some level of use in the outside world. One way to view the process is that each organization adopts technologies according to their risk profile and works to reduce the risk relevant to their missicn, adding value to the overall maturation of the technology.

How does this maturation occur?



Carnegie Mellon University
Software Engineering Institute

IDA Maturation Study

Software technology maturation study sponsored by the Software Technology for Adaptable, Reliable Systems (STARS) Program. The Institute for Defense Analyses (IDA) commissioned case study analyses of 14 technologies:

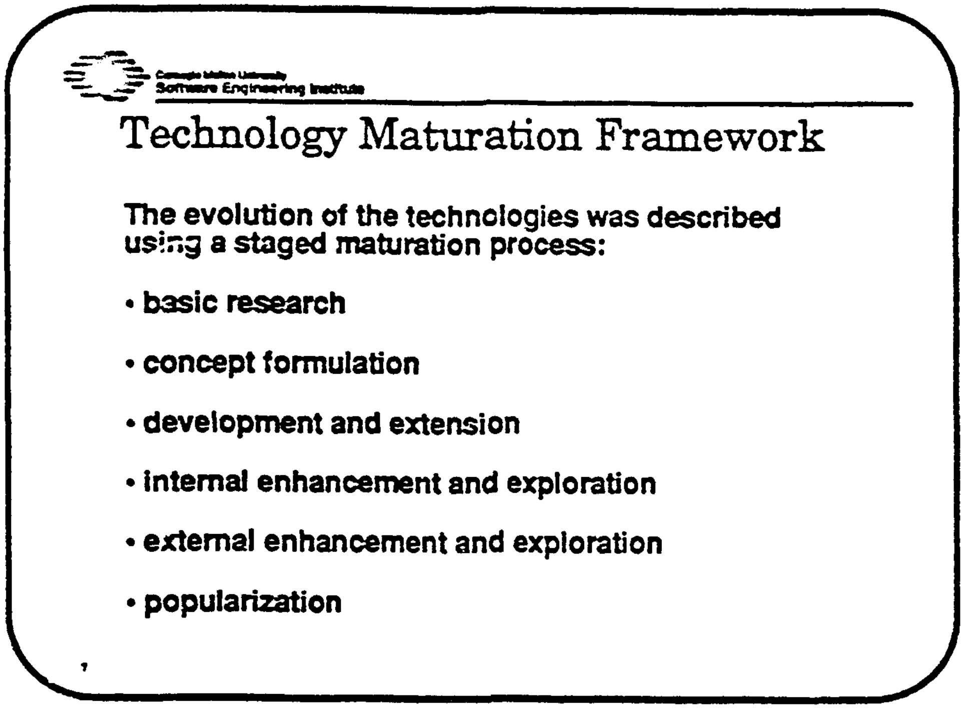
- knowledge-based systems
- software engineering
- formal verification technology
- compiler construction
- metrics
- abstract data types
- structured programming
- SCR methodology
- DoD-STD-SDS
- AFR 800-14
- cost models
- Smalltalk-80
- SREM
- Unix

Notes

[Redwine 84] was commissioned by the STARS Program. Between February and May 1984, in-depth case studies were created for each technology listed on the slide. The case studies focused on the technical activities performed at different organizations that helped to mature each technology. Some of this list may not be familiar to all in the audience:

- Naval Research Laboratory (NRL) Software Cost Reduction (SCR) Project - A-7E study done by Parnas et al.
- DoD-STD-SDS - precursor to DoD-STD-2167A
- AFR 800-14 - "Lifecycle Management of Computer Resources in Systems" - "This regulation establishes policy for the acquisition and support of computer resources."
- SREM - Systems Requirements Engineering Methodology developed by TRW, Huntsville, under the sponsorship of the Army's Ballistic Missile Defense Advanced Technology Center (BMDATC). Initially looked at requirements and specification - TRW has extended SREM for use by other customers.

The authors provided a timeline for the major events in the maturation of each technology so that the technologies could be "measured" using a common yardstick.



Carnegie Mellon University
Software Engineering Institute

Technology Maturation Framework

The evolution of the technologies was described using a staged maturation process:

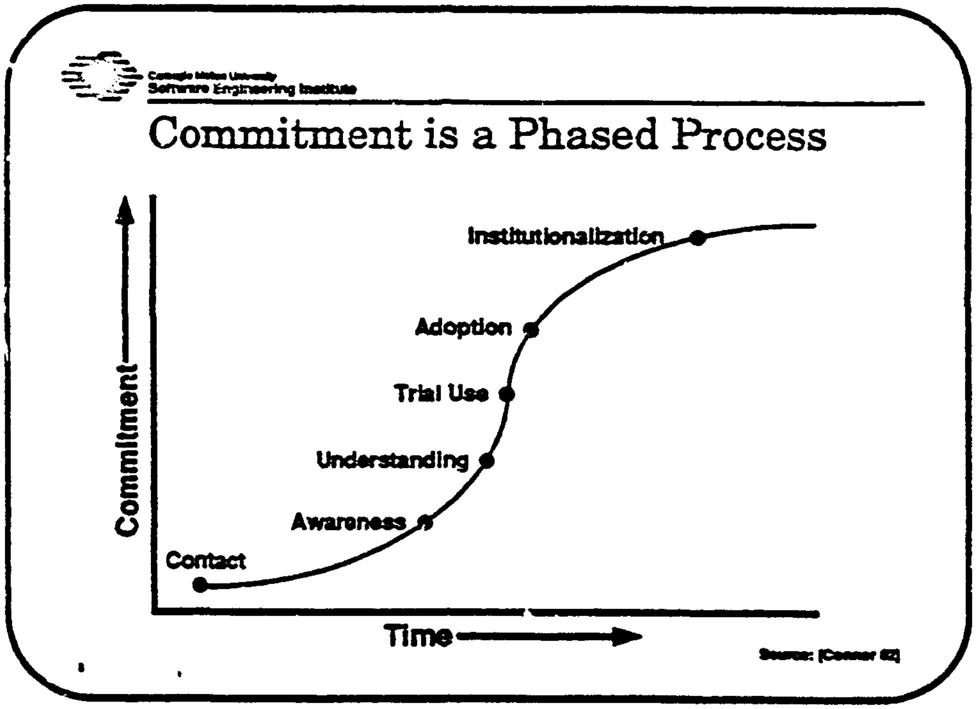
- basic research
- concept formulation
- development and extension
- internal enhancement and exploration
- external enhancement and exploration
- popularization

Notes

Using these timelines, the authors of the report mapped the technical and transition activities into this maturation framework:

- basic research: appearance of a key idea underlying the technology or a clear articulation of the problem
- concept formulation: clear definition of solution approach via a seminal paper or demo
- development and extension: usable capabilities become available
- enhancement and exploration (internal): shift to usage outside the development group
- enhancement and exploration (external): substantial evidence of value and applicability
- popularization: at 40% and 70% market penetration levels

A major point brought out in the case studies is that the lack of sharing of knowledge and experience with a technology across organizational boundaries greatly inhibited the transition of that technology. As we discussed earlier, this lack of sharing results in reinvention of the technology within the multiple contexts adopting the technology. We posit that this reinvention contributes to the authors' finding that technology maturation, as they defined it, takes 18 +/- 3 years.



Carnegie Mellon University
Software Engineering Institute

Commitment is a Phased Process

Source: [Conner 82]

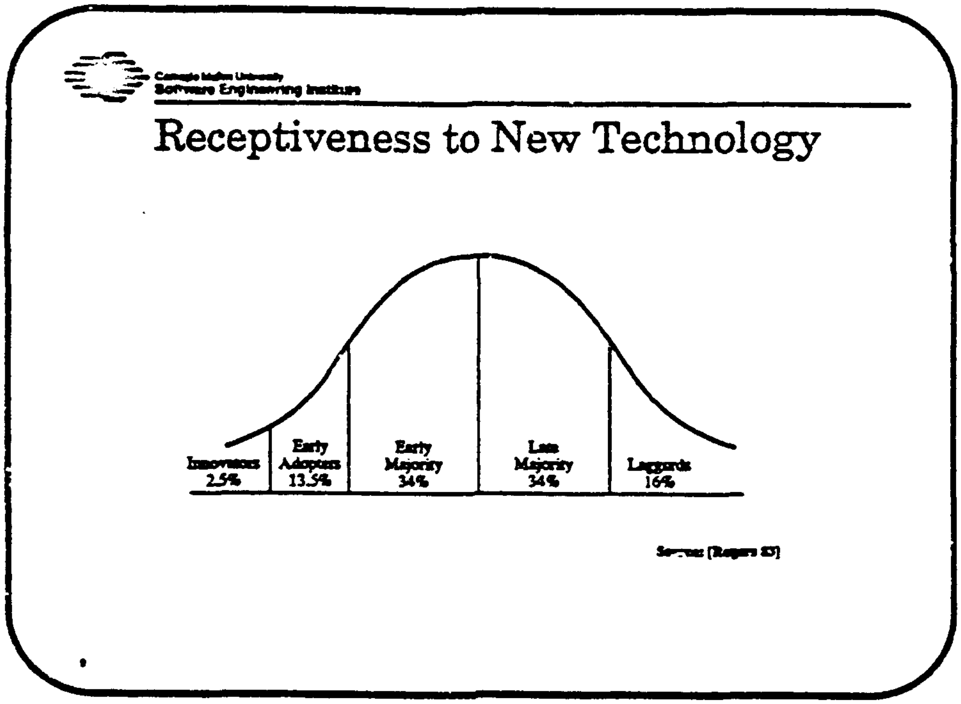
Notes

Both individuals and groups make commitments to the adoption of new technologies in a regular pattern:

- Con*ct - the transition target has had contact with the technology through some means, e.g., documents, briefings, marketing information, etc.
- Awareness - that contact (or others) make the target aware of the existence of the technology.
- Understanding - the target understands the technology well enough to be conversant in the relevant details.
- Trial use - the target agrees to use the technology for some purpose on a trial basis, e.g., a pilot project, prototype development, etc. This is often done to facilitate the "adoption" decision.
- Adoption - the target agrees to use the technology more widely within their organization for an application that is related to the target's business purporse
- Institutionalization - the use of the technology is made part of the standard practices of the organizations

There si an enterplay between the commitment curve and the models from Rogers and Curtis:

- Different information needs
- Different time frames
- Different success criteria



Carnegie Mellon University
Software Engineering Institute

Receptiveness to New Technology

Source: [Rogers 83]

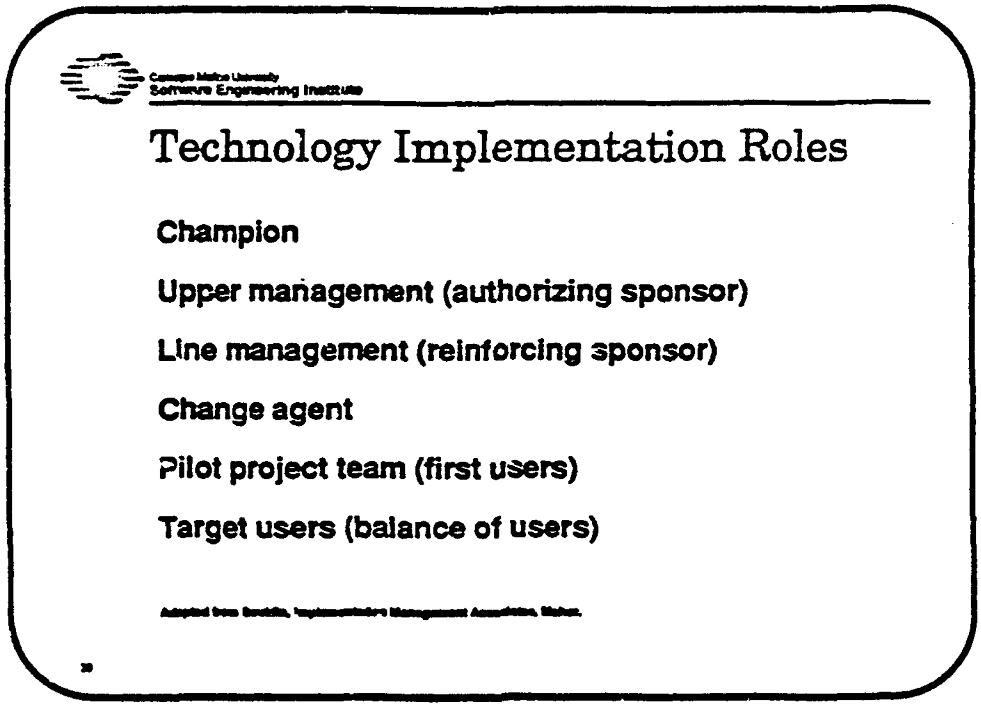
Notes

The study of the diffusion of innovations has been a major research area since the 1940s. "Diffusion is the process by which an *innovation* is *communicated* through certain channels *over time* among members of a *social system*" [Rogers 83, p.5.]. The bell curve represents classes of potential adopters:

- innovators — venturesome, cosmopolitan, technical expertise, often control financial resources
- early adopters — respectable, opinion leader, role model
- early majority — deliberate, seldom hold leadership positions
- late majority — skeptical, adopt in response to peers, risk averse
- laggards — traditional, often isolated

Membership in these "market segments" changes depending on a number of factors, including the results of previous change efforts, the type of technology, and an individual's role in the organization or change effort. This model can also be viewed as a surrogate measure for risk aversion, e.g., individuals on the left side of the model are more willing to take a chance on a new technology.

Maturation extends the diffusion concept to include value-adding activities performed by participants in the technology development life cycle.



Carnegie Mellon University
Software Engineering Institute

Technology Implementation Roles

Champion

Upper management (authorizing sponsor)

Line management (reinforcing sponsor)

Change agent

Pilot project team (first users)

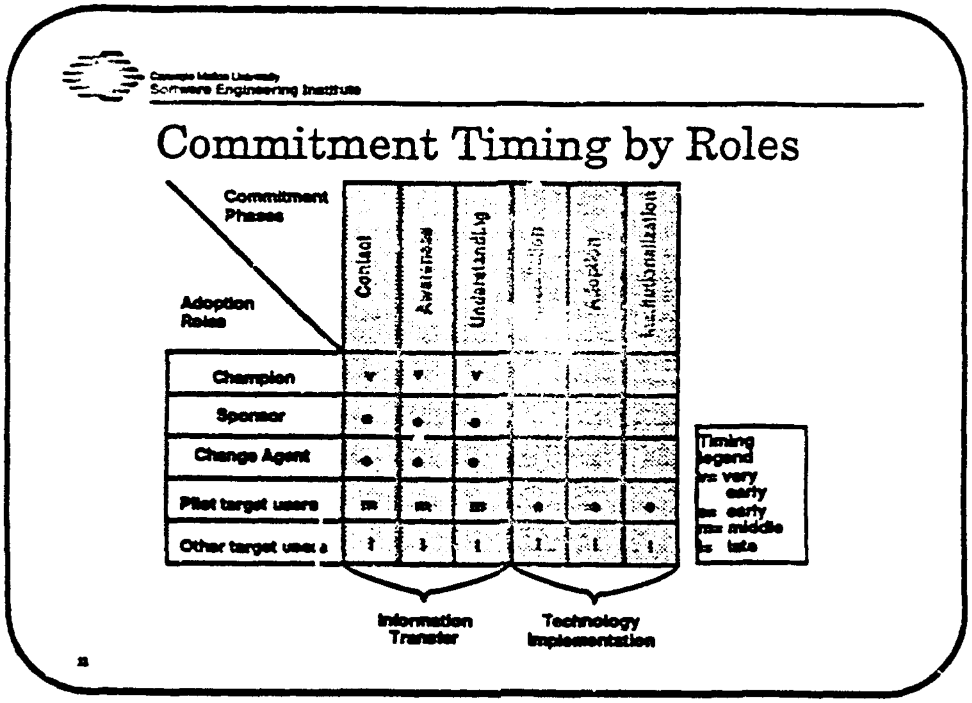
Target users (balance of users)

Adapted from [illegible]

Notes

People in each of these roles experience the commitment curve differently and at different times. For example, the champion proceeds up the curve ahead of everyone else (probably one of the innovators or early adopters described by Rogers). Let's take a look at what level of commitment is required when by which of these people (or groups of people).

The sponsor role at the upper management level provides resources, strategic and policy direction, and final approval to proceed with the implementation of a technology. At line management level, the sponsor may authorize resources and direct efforts toward planning for mplementation and trial use. The product champion is the individual who initially introduces the idea of a particular technology, and informally advocates it, calling it to the attention of others. The change agent is an individual or team, drawn from line management or software personnel, who does the detailed planning and implementation of the technology. The pilot project team tries the technology for the first time on behalf of the larger organization. The target users are the remainder of the organization who will eventually implement the technology. Routine, everyday use of a technology is called "institutionalization".



Carnegie Mellon University
Software Engineering Institute

Commitment Timing by Roles

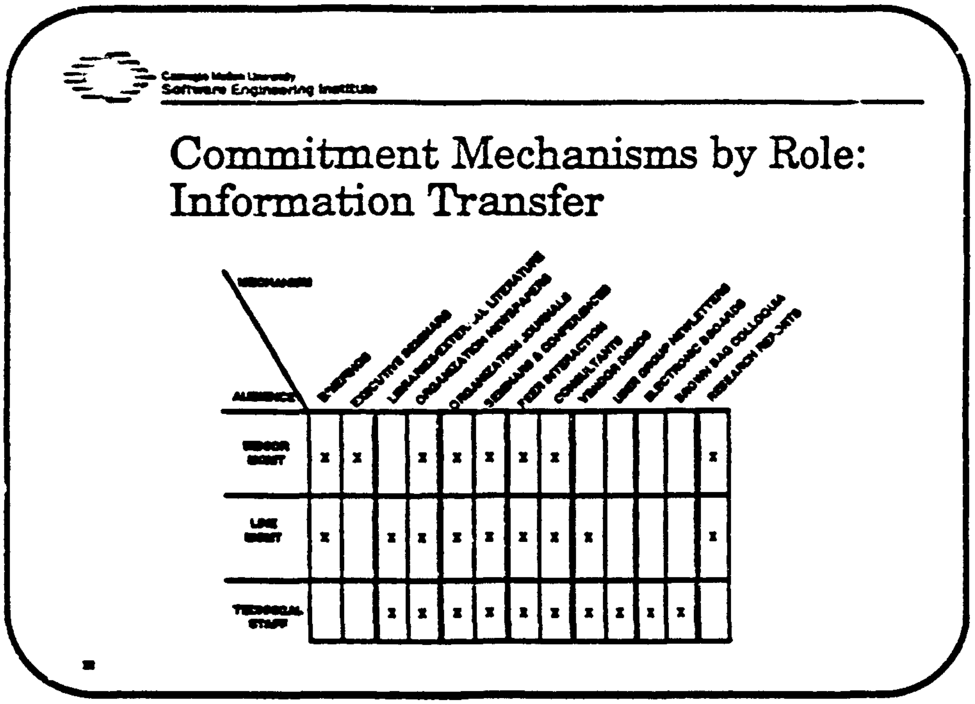
Adoption Roles \ Commitment Phases	Contact	Awareness	Understanding	Installation	Adoption	Institutionalization
Champion	▼	▼	▼			
Sponsor	●	●	●			
Change Agent	●	●	●			
Pilot target users	■	■	■	●	●	●
Other target users	↕	↕	↕	↕	↕	↕

Information Transfer
Technology Implementation

Timing legend
▼ very early
● early
■ middle
↕ late

Notes

Very generally, the timing of commitment is related to roles within the transition process. This picture gives a rough idea of how different participants in the transition process proceed through the stages represented on the commitment curve. For purposes of managing the transition process, it is helpful to think in terms of two categories of activies: information transfer, and technology implementation. The mechanisms that fall into the former category are most likely familiar to you; those that fall into the latter category may be less so.



Carnegie Mellon University
Software Engineering Institute

Commitment Mechanisms by Role: Information Transfer

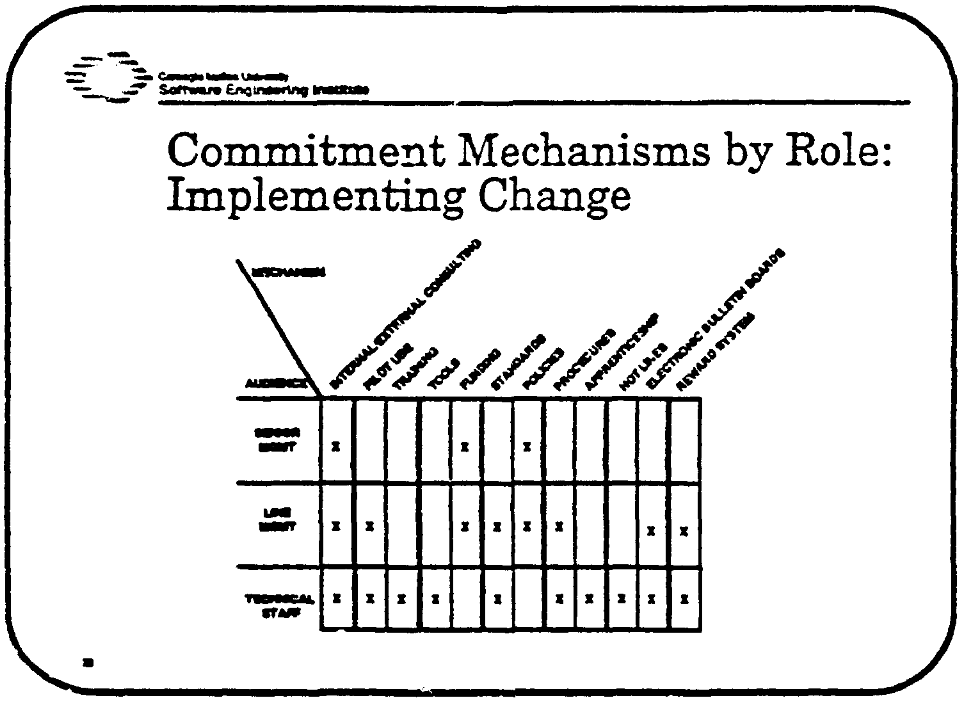
AUDIENCE \ MECHANISMS	BRIEFINGS	EXECUTIVE SEMINARS	LIBRARIES/EXTERNAL LITERATURE	ORGANIZATION NEWSPAPERS	ORGANIZATION JOURNALS	SEMINARS & CONFERENCES	PEER INTERACTION	CONSULTANTS	VENDOR DEMOS	USER GROUP NEWSLETTERS	ELECTRONIC BOARDS	BROWN BAG COLLOQUIA	RESEARCH REPORTS
SENIOR MGMT	x	x		x	x	x	x	x					x
LINE MGMT	x		x	x	x	x	x	x	x				x
TECHNICAL STAFF			x	x	x	x	x	x	x	x	x	x	

Notes

Information transfer is often confused with technology transition, whereas in reality it is only *part* of technology transition. It is an important part, because it is what effects contact, awareness, and understanding, and can be managed much more systematically and strategically that it typically is. So we'll spend a few minutes on it.

Sponsors, at both senior and middle management levels, need information as much as the engineers and practitioners who will be champions, change agents, and users. They are often forgotten, or their need for information is not attended to early enough. One way to develop a group of *potential* sponsors is to provide to management more broadly, through mechanisms such as company newspapers, or presentations at annual meetings of corporate technical committees, the opportunity to hear about new technologies and how they might apply to the organization. In addition to describing the technologies themselves, it can be helpful to describe how other organizations are using them, and how.

Mechanisms for technical personnel are very familiar, including informal colloquia such as brown bag lunches; libraries; demonstrations, conferences, and so on. Most people are familiar with these, and have some skill in using these. What is less carefully considered is to whom these activities are targeted and in what sequence. Use of the commitment curve along with some analysis of potential users can help here.



Carnegie Mellon University
Software Engineering Institute

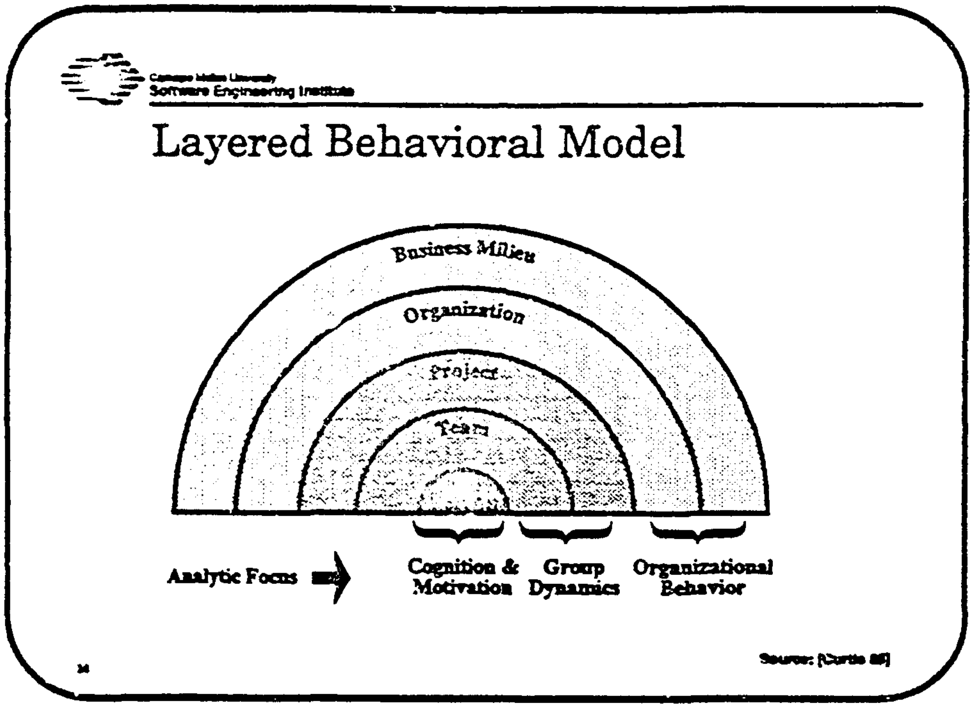
Commitment Mechanisms by Role: Implementing Change

MECHANISM / AUDIENCE	INTERNAL/EXTERNAL CONSULTING	PILOT USE	TRAINING	TOOLS	FUNDING	STANDARDS	POLICIES	PROCEDURES	APPRENTICESHIP	HOT LINES	ELECTRONIC BULLETIN BOARDS	REWARD SYSTEM
SENIOR MGMT	X				X		X					
LINE MGMT	X	X			X	X	X	X			X	X
TECHNICAL STAFF	X	X	X	X		X		X	X	X	X	X

Notes

Mechanisms for implementing change are more labor-intensive than those for information transfer, and thus are more likely to be used once sponsorship is obtained and resources for transition are allocated. These are the mechanisms that support the use of new technology in practice. For example, without skills from training, new users are often frustrated and waste considerable effort attempting to learn a new technology from peers, documentation, or experimentation. Without proactive standards revision or waivers, new technology which is being piloted will hit roadblocks, and the additional resources required to communicate with standards personnel add overhead to the pilot use, muddying the evaluation of the technology. Without pilots themselves, premature attempts are made to use new technologies with no "shake down" period, and problems of technology transition are often blamed on the technology itself, which then gets discarded.

There is another consideration here often omitted from technology transition planning. Management may need to do its job differently. The classic example of this in a software engineering context is how management reacts when no code is immediately produced on a new software project. Management needs to be educated not just about the technology content, but also about the changes they need to make in their own practices, such as how they track indicators that technology implementation is proceeding successfully. Sometimes management needs new skils as well as new information.



Carnegie Mellon University
Software Engineering Institute

Layered Behavioral Model

Source: [Curtis 88]

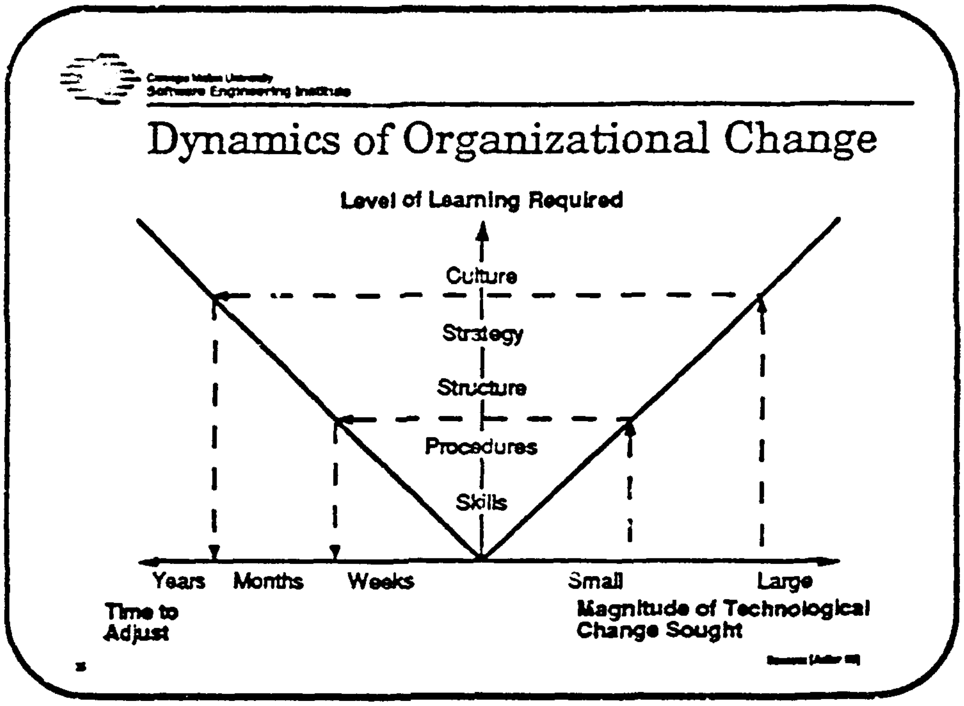
Notes

Curtis, Krasner, and Iscoe conducted a field study of the software design process. Between May and August 1986, the research team conducted interviews with personnel on 19 application development projects in 9 companies. The applications ranged in size from 24 to 1000 KLOC and included embedded systems, operating systems, Computer-Aided Design (CAD), and telephony.

While their research focused on "creation," we can argue that most technologies are "reinvented" in each context. Any analysis of the technology maturation and adoption process must recognize differences in orientations, motivations, and responsibilities.

Depending on the context, needs, and possible impacts of the technology, those seeking to implement the technology may have to address multiple levels within the organization. John will later discuss the "cascading sponsorship" across organizational levels that is often required to implement such technologies.

Technology maturation and adoption is a learning and knowledge transfer process. [Curtis 88] suggests that we look at cognition and motivation to understand the process. The next models to be discussed examine issues at this level.



Carnegie Mellon University
Software Engineering Institute

Dynamics of Organizational Change

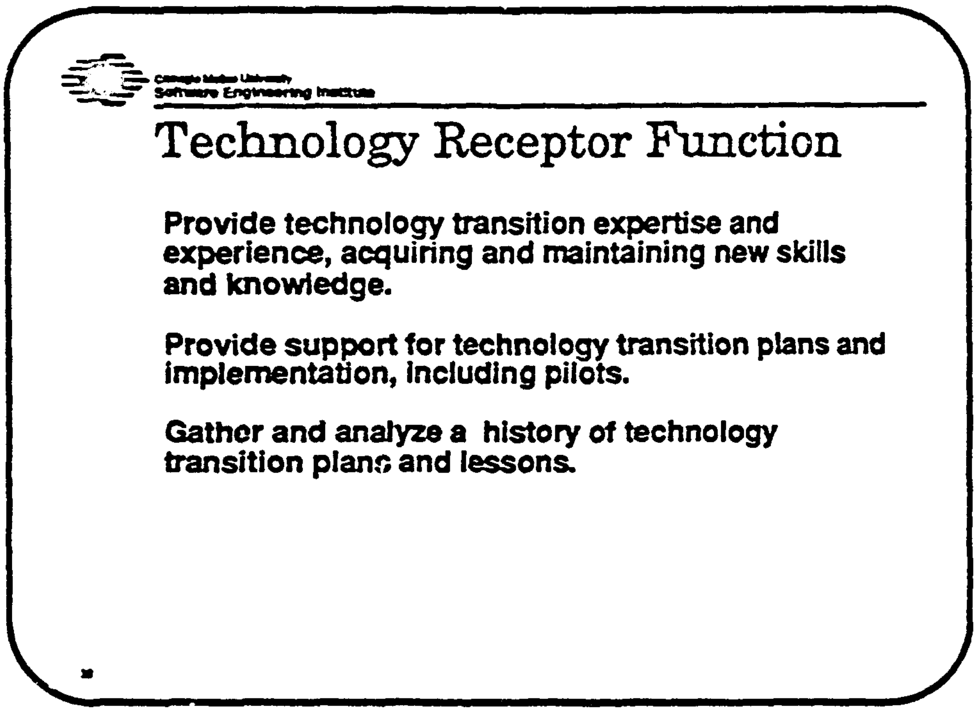
Source: [Adler 89]

Notes

The magnitude of a technology-driven change depends on the overall impact of the technology on the organization. A new design method, for example, may be part of a larger effort to change the way an organization does business as part of a quality improvement program.

Conversely, a new CASE tool may only be used in a small part of the organization, and will have little impact beyond a specialized application.

More often than not, the advent of multiple small technologies can be seen from a broader perspective as a larger effort, with greater impact on the organization.



Carnegie Mellon University
Software Engineering Institute

Technology Receptor Function

Provide technology transition expertise and experience, acquiring and maintaining new skills and knowledge.

Provide support for technology transition plans and implementation, including pilots.

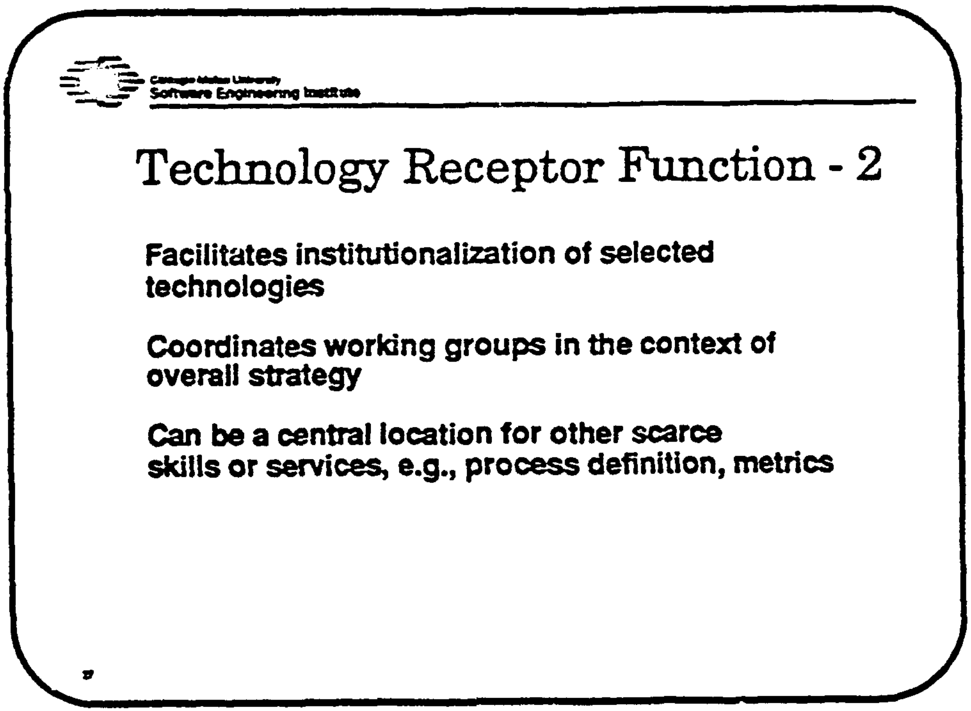
Gather and analyze a history of technology transition planc and lessons.

Notes

Let's now look more closely at the technology transition function proposed here.

What expertise might we look for in the technology transition function? Candidates need not have an MBA in technology management, but we do suggest the following:

1. Both people and technical skills
2. Credibility with technical people and with management
3. Experience outside this particular organization
4. Several years experience inside this organization
5. Some knowledge of technology transition (material from this tutorial, and from the SEPG Guide [Fowler 90] would be helpful)
6. A strong interest in strategic planning



Carnegie Mellon University
Software Engineering Institute

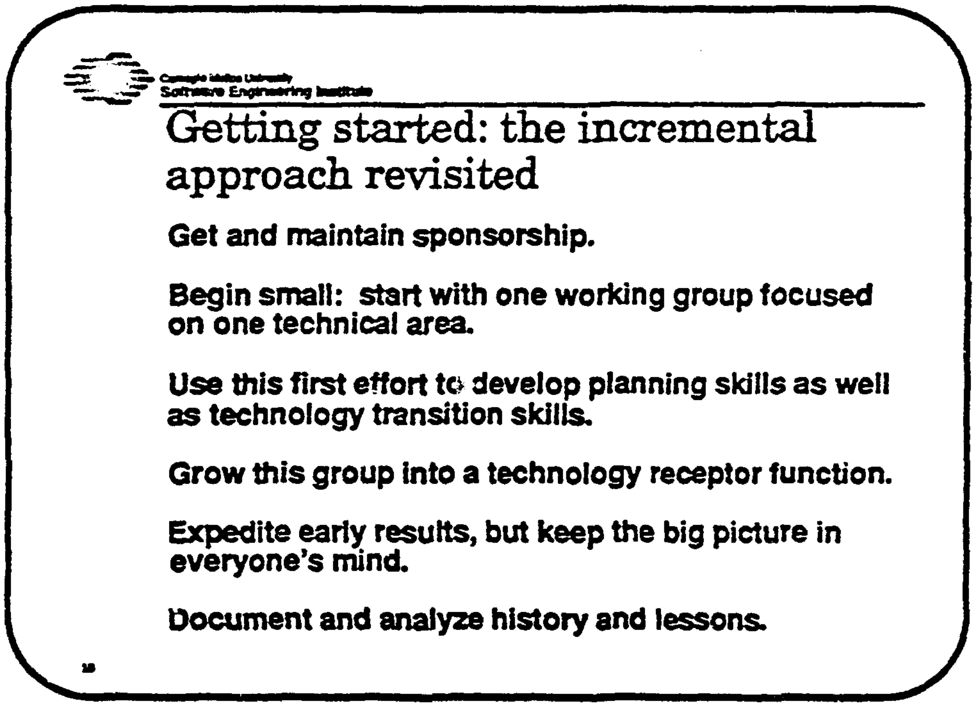
Technology Receptor Function - 2

Facilitates institutionalization of selected technologies

Coordinates working groups in the context of overall strategy

Can be a central location for other scarce skills or services, e.g., process definition, metrics

Notes



Carnegie Mellon University
Software Engineering Institute

Getting started: the incremental approach revisited

Get and maintain sponsorship.

Begin small: start with one working group focused on one technical area.

Use this first effort to develop planning skills as well as technology transition skills.

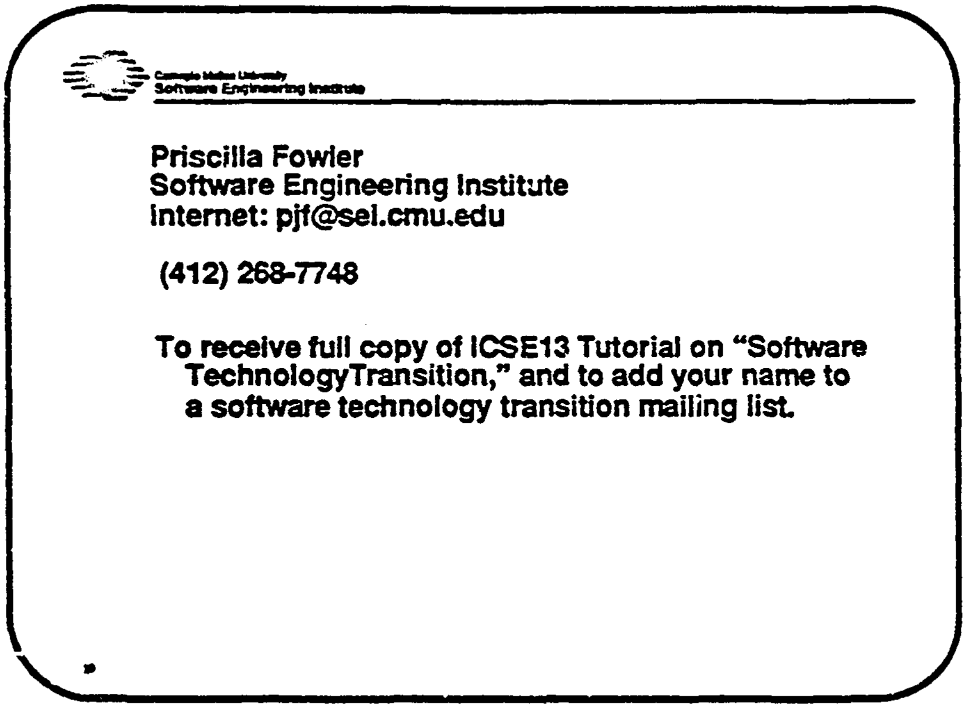
Grow this group into a technology receptor function.

Expedite early results, but keep the big picture in everyone's mind.

Document and analyze history and lessons.

Notes

The principles we've discussed for technology transition also apply to putting all the recommended elements into place. The set of three key elements-cyclical approach, plan hierarchy, and organizational architecture-comprise a major innovation for most organizations. Unless your organization has most of these elements already in place, we recommend you start small. Use a single working group as a prototype. Don't immediately establish a formal steering committee, but rather work informally with one or two key managers who agree to act as sponsors. Develop your planning and technology tranistion skills by trying these approaches on small-scale change efforts where you can manage the risk and limit the visibility of your mistakes. When you are successful, talk up the results and sho how they support progress toward the ultimate goal, but be very careful not to promise too much too soon. Document your lessons and use your initial results and experience to bootstrap a larger effort, evolving your way towards a full-fledged and systematic approach to ongoing technology transition.

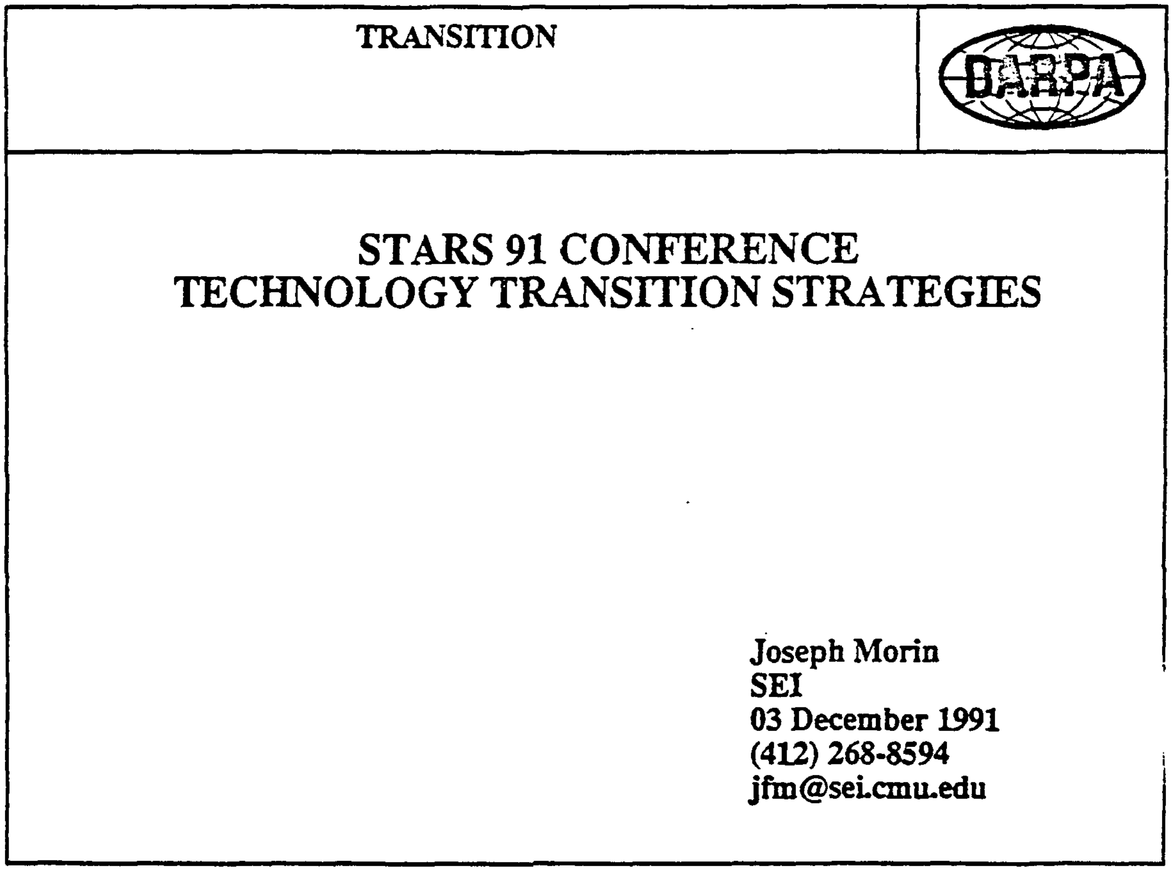


Priscilla Fowler
Software Engineering Institute
Internet: pjf@sei.cmu.edu

(412) 268-7748

To receive full copy of ICSE13 Tutorial on "Software TechnologyTransition," and to add your name to a software technology transition mailing list.

Notes



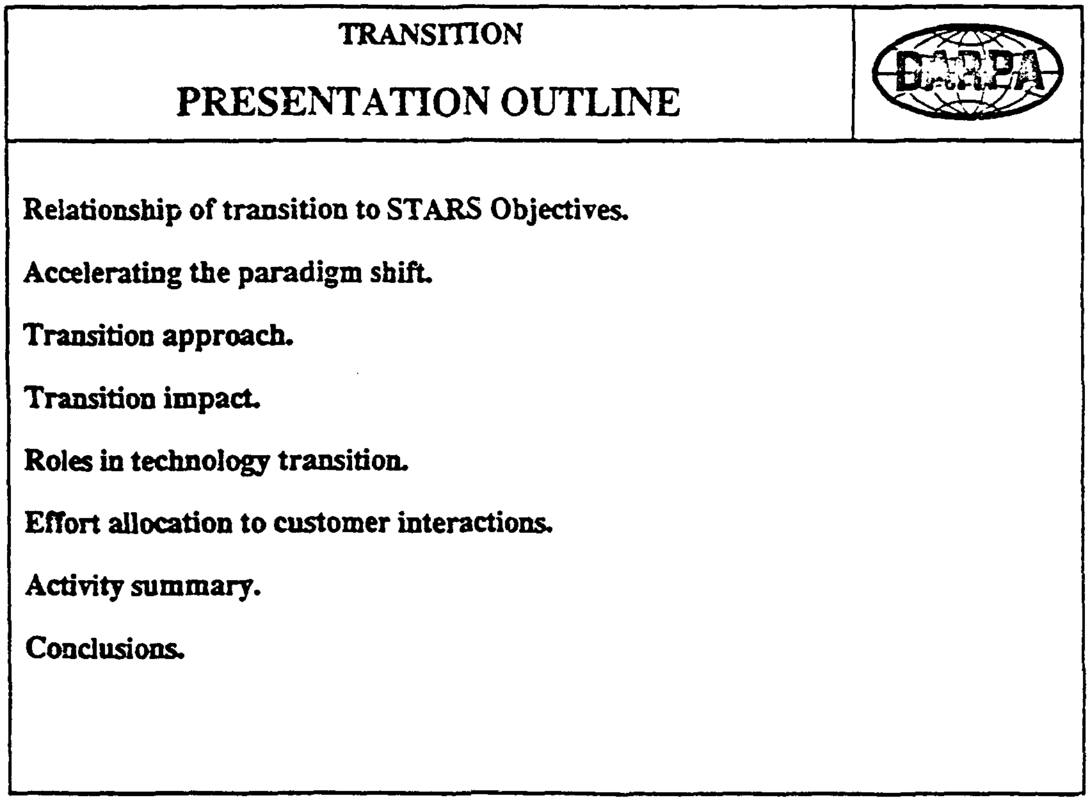
TRANSITION

STARS 91 CONFERENCE
TECHNOLOGY TRANSITION STRATEGIES

Joseph Morin
SEI
03 December 1991
(412) 268-8594
jfm@sei.cmu.edu

Transition/Morin/VG-1

This presentation will discuss the approach STARS is taking with respect to technology transition. It will discuss the STARS approach to information dissemination as well as the STARS approach to working with receptor groups in order to accelerate the installation and adoption of megaprogramming support products.



TRANSITION

PRESENTATION OUTLINE

Relationship of transition to STARS Objectives.

Accelerating the paradigm shift.

Transition approach.

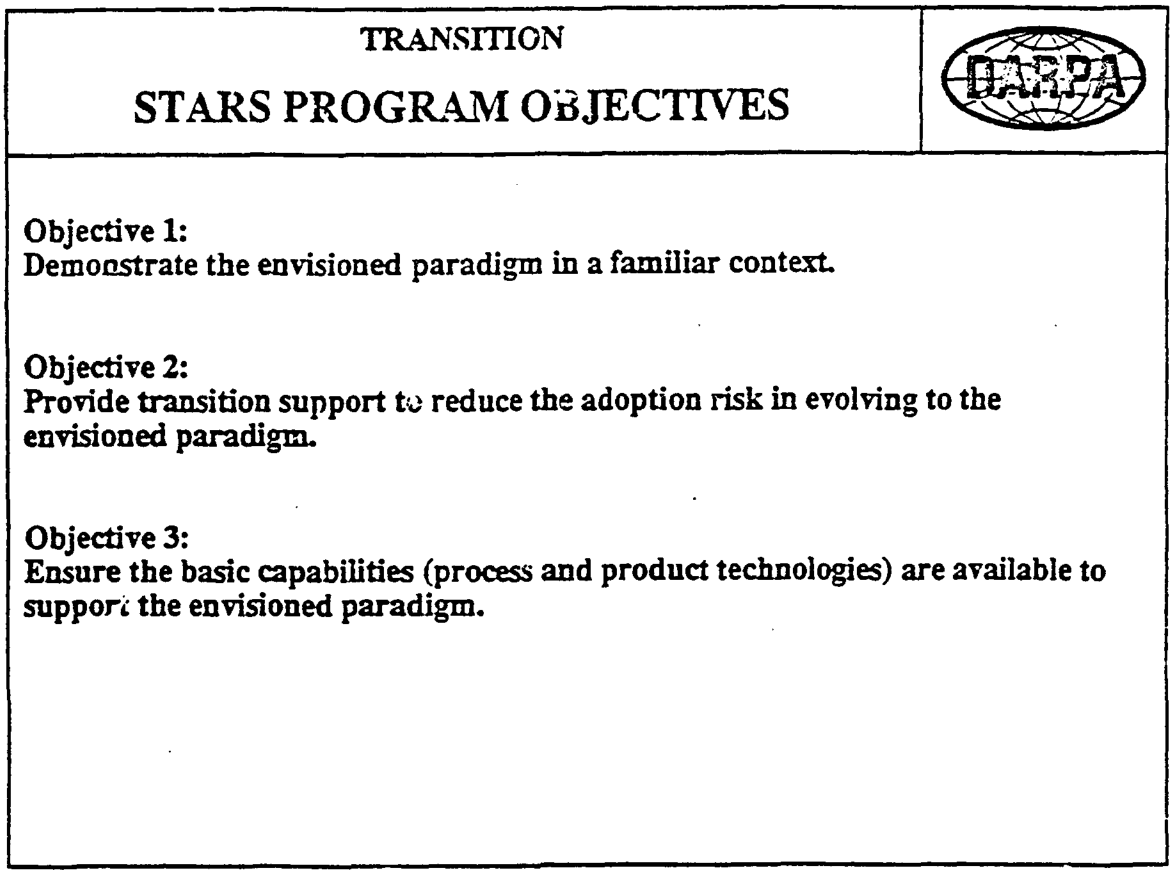
Transition impact.

Roles in technology transition.

Effort allocation to customer interactions.

Activity summary.

Conclusions.



TRANSITION

STARS PROGRAM OBJECTIVES

DARPA

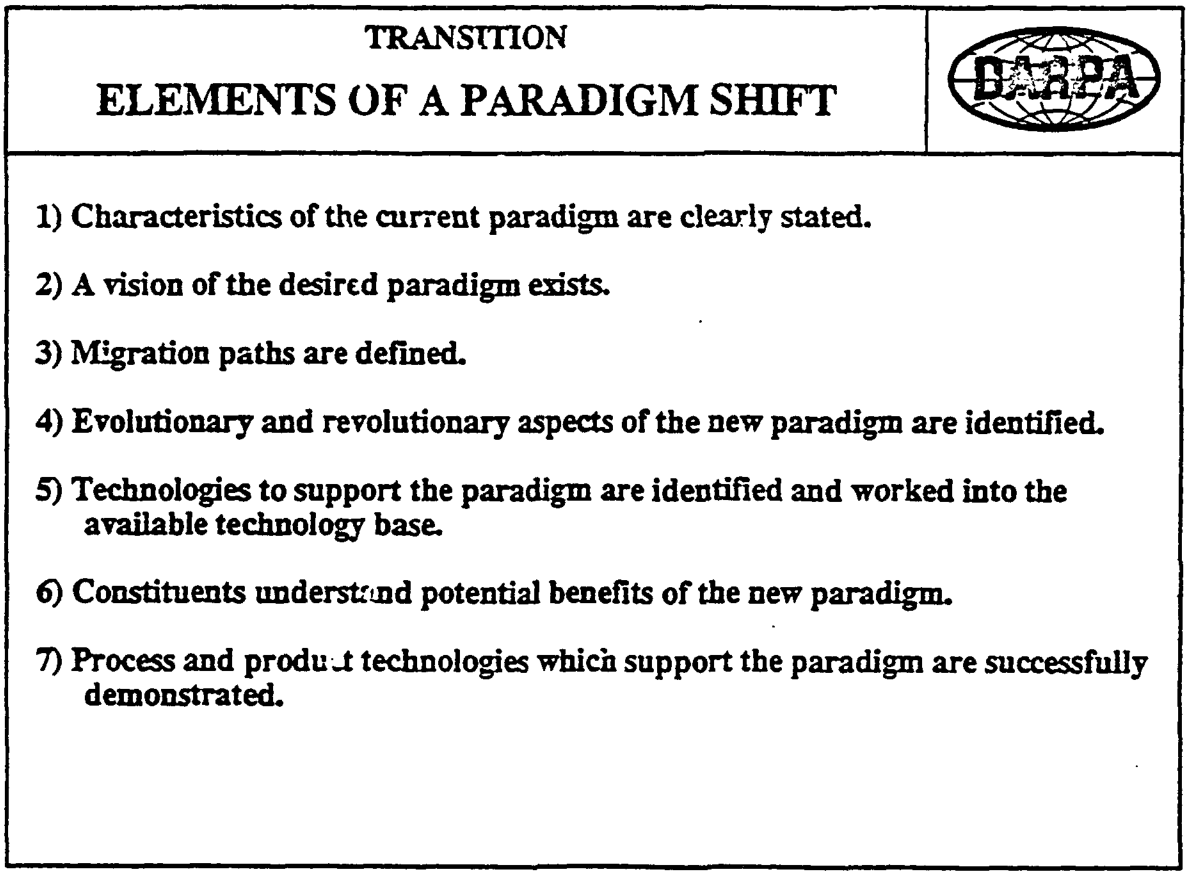
Objective 1:
Demonstrate the envisioned paradigm in a familiar context.

Objective 2:
Provide transition support to reduce the adoption risk in evolving to the envisioned paradigm.

Objective 3:
Ensure the basic capabilities (process and product technologies) are available to support the envisioned paradigm.

Transition/Marin/VG-3

paradigm shift. The program's objectives are designed to successfully transfer technology; however, they are not in themselves the activities of technology transition. The previous speaker has outlined models of technology transition. Now we will discuss the STARS strategy for applying those models to actual transition activities which support the program's objectives. For our purposes, the focus will be on objectives 2 and 1 in that order. Objective 3 is, for the most part, a precondition to transition activities and we will not dwell on it in this track.



TRANSITION

ELEMENTS OF A PARADIGM SHIFT

DARPA

1) Characteristics of the current paradigm are clearly stated.

2) A vision of the desired paradigm exists.

3) Migration paths are defined.

4) Evolutionary and revolutionary aspects of the new paradigm are identified.

5) Technologies to support the paradigm are identified and worked into the available technology base.

6) Constituents understand potential benefits of the new paradigm.

7) Process and product technologies which support the paradigm are successfully demonstrated.

Transition/Morin/VG-4

NOTES

TRANSITION

ACCELERATING THE SHIFT

1) Paradigm comparisons in DoD, DARPA, and STARS documents. Other documentation of identified and latent DoD needs.

2) DARPA Software Technology Strategy; STARS adaptation of Megaprogramming Vision.

3) SWAP; SDP 2000; CARDS blueprint; SEI CMM; JLC Reuse Committee;...

4) Build on industry *standards and commercial base* to facilitate *Evolution*. Demonstrate viability and benefit of revolutionary aspects (minimize).

5) DARPA Software programs; STARS process / reuse / technology support thrusts; coordination among DoD software technology programs.

6) IDA cost modeling work; Cost / Benefit data from demo projects.

7) Alpha & Beta usage; TT affiliates; STARS demonstrations and lessons learned.

Transition/Mom/VG-5

NOTES

The current paradigm and the envisioned paradigm are described in a variety of existing or planned documents.

Migration paths are being defined by STARS and others:
DoD's Software Action Plan (SWAP);
STARS Software Development Plan (SDP) of the year 2000;
SEI's Capability Maturity Model (CMM);
etc.

The emphasis is on evolutionary rather than revolutionary change. Use of commercial standards and technologies is preferred.

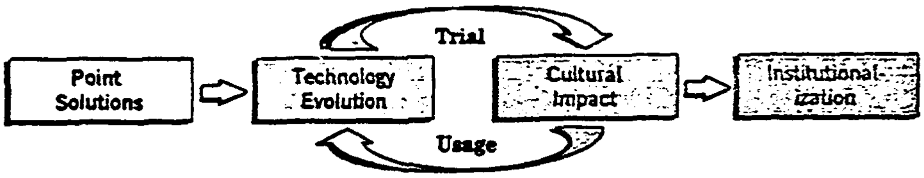
The technology base is being extended to support the new paradigm via STARS work (discussed in the other tracks) and via other DoD software programs.

Cost / Benefit determinations and a base of success stories are key factors which we are providing.

TRANSITION AND COMMUNITY INVOLVEMENT

TECHNOLOGY TRANSITION APPROACH

DARPA



• Multiple Disjoint Technology Transfer efforts – Newsletters – Conferences – Bulletin Board	• Integrated Technology Transition Strategy – Identification of receptor groups – Alpha/beta Usage by Primes – Affiliates Program – STARS 'XX	• Adoption Barrier Risk Reduction • CARDS reuse blue-print • Lessons Learned • Migration Paths	• General Community Adoption
• STARS Catalog/ DTIC distribution • Shotgun distribution of STARS Point solu-tions	• ASSET • Packaging interim products	• ASSET • Demo Projects – Instantiated solutions	• Commercialized solutions

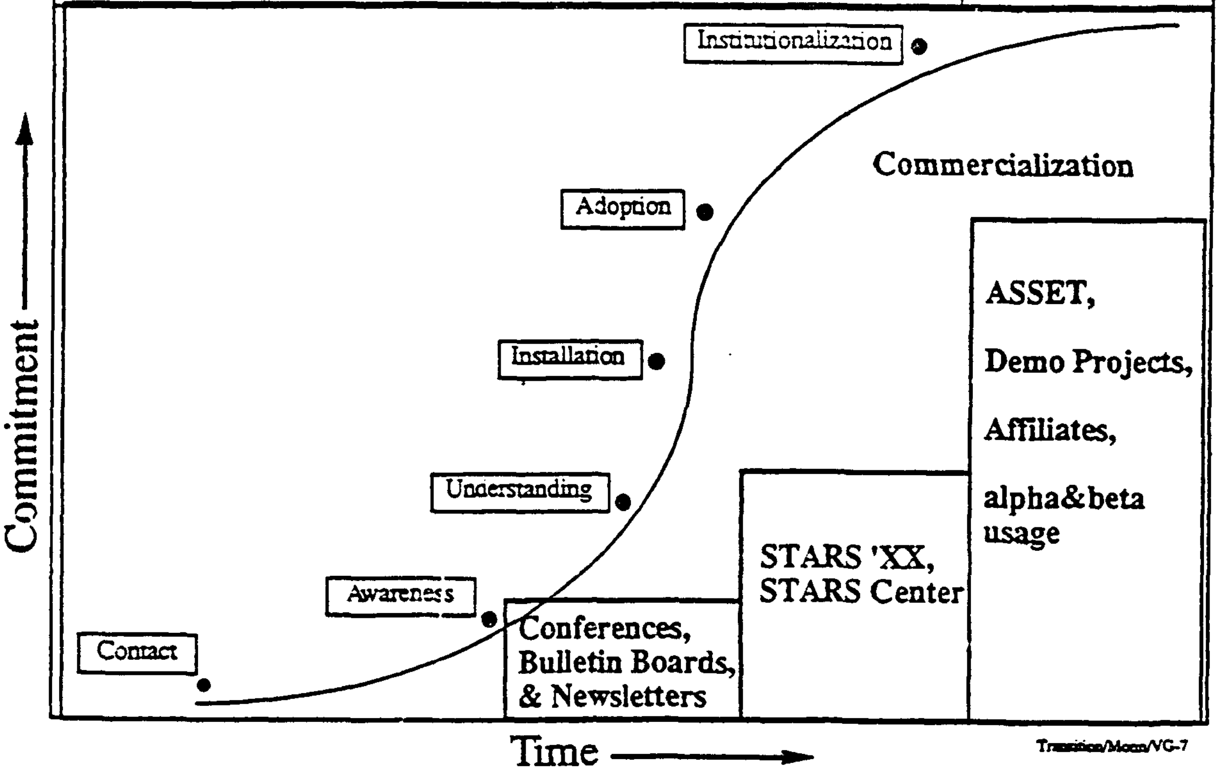
Transition/Marin/VG-4

NOTES

As you can see, part of our strategy is to apply transition principles to evolving our transition activities. We are currently at the point of moving from point solution transition activities to an integrated transition strategy under the guidance of a transition coordinator to be appointed shortly. The initial strategy and activity definition will be evolved through use and in light of it's cultural impact. Dissemination of transition lessons learned will in itself become a valuable transition activity supporting eventual institutionalization.

TRANSITION

STARS TRANSITION IMPACT



NOTES

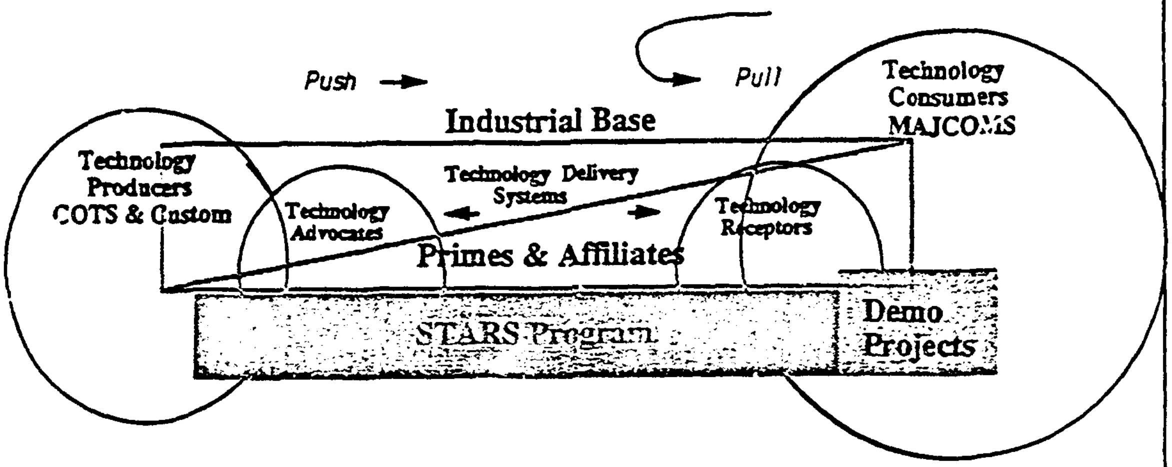
STARS transition activities are intended to move the community up the adoption curve as quickly as possible. Different members of the community will move up the curve at different times and rates. Not surprisingly, activities designed to move up through higher levels are more resource intensive than those at lower levels. This will be dealt with in part through feedback from higher level activities into the information base being disseminated as part of the activities promoting Awareness and Understanding. The program will not in and of itself carry through to institutionalization. However, the way will be paved for commercialization as a path to institutionalization.

TRANSITION

ROLES IN TECHNOLOGY TRANSITION

What are the solutions?

What are the problems?



Adapted from Fowler

NOTES

This adaptation of a chart introduced in the previous talk attempts to show the roles of STARS program participants in the transition process.

The program itself is a producer and consumer of technology. It is also concerned with establishing the delivery vehicles and forums needed to bring producers and consumers together.

The primes, subs, and affiliates are a subset of the overall DoD software industrial base with which we are concerned. They include members focused on production, advocacy, reception, and consumption to varying degrees. The involvement of this group and the lessons learned by this group will be directly relevant to motivating the larger industrial base.

The Demo projects will involve the major commands which are the ultimate consumers of the technology as well as the contractors and support organizations which represent them.

TRANSITION

TRANSITION ROLES ELABORATED

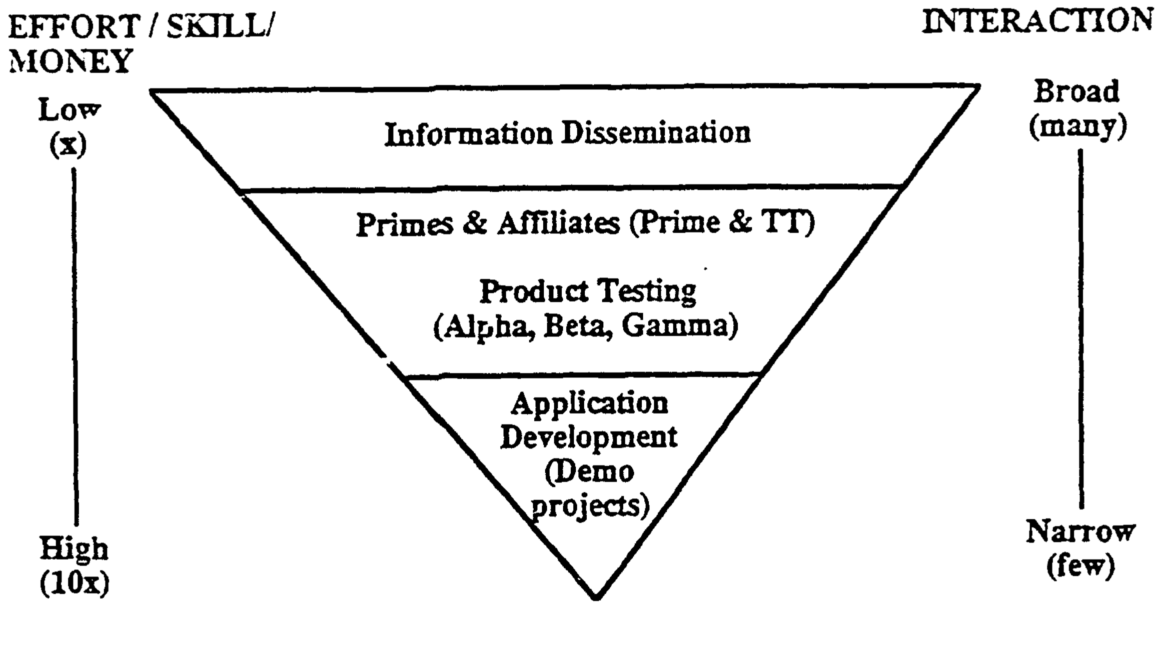
WHO	WHAT	WHY
STARS PROGRAM	Vision, Direction, Resource, Forum	Motivation, Sponsorship
PRIMES & SUBs	Technology maturation and integration	Technology Base
COMMERCIAL COUNTERPARTS & AFFILIATES	Technology production and advocacy	Commercialization (supply side)
AFFILIATES	Technology reception and consumption	Commercialization (demand side)
DEMO PROJECTS	Application development case studies	Adoption Barriers "Validation"

NOTES

This chart elaborates the roles just introduced to one more level of detail. It is not all encompassing; however, it does help associate the activities (what) of the participants (who) with a transition objective (why).

TRANSITION

CUSTOMER INTERACTIONS



EFFORT / SKILL/ MONEY

Low (x)

High (10x)

Information Dissemination

Primes & Affiliates (Prime & TT)

Product Testing (Alpha, Beta, Gamma)

Application Development (Demo projects)

INTERACTION

Broad (many)

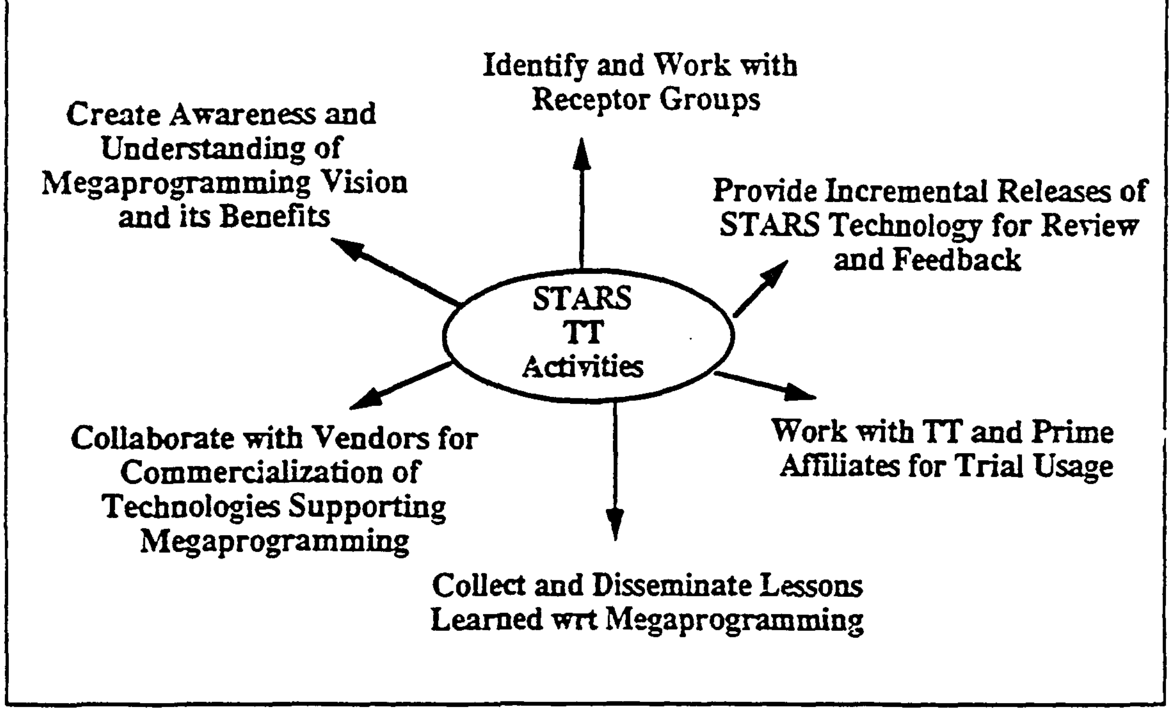
Narrow (few)

NOTES

Although there are only three application development projects, there will be a significant number of people involved in each and as mentioned the results will become part of the general information distribution activities.

TRANSITION

TT SUPPORT ACTIVITIES SUMMARY



STARS TT Activities

Identify and Work with Receptor Groups

Create Awareness and Understanding of Megaprogramming Vision and its Benefits

Provide Incremental Releases of STARS Technology for Review and Feedback

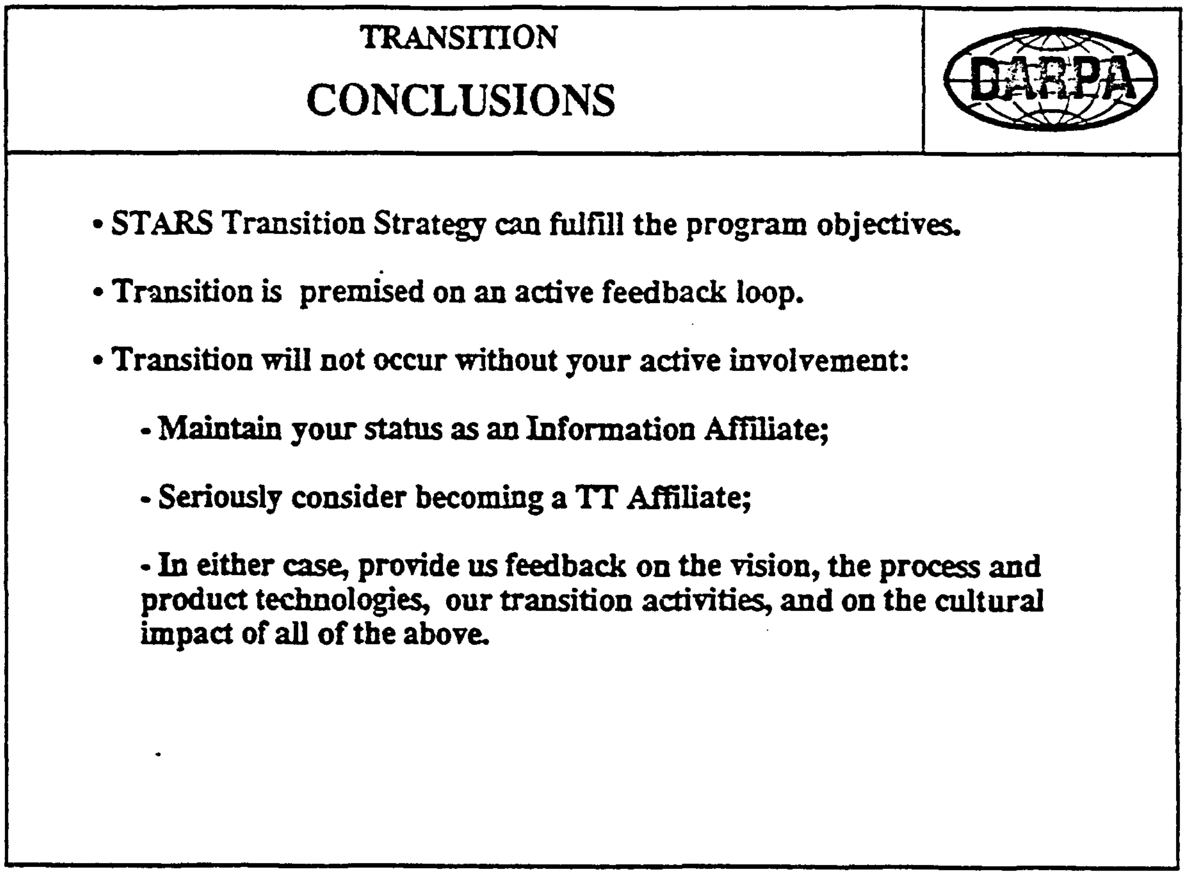
Collaborate with Vendors for Commercialization of Technologies Supporting Megaprogramming

Work with TT and Prime Affiliates for Trial Usage

Collect and Disseminate Lessons Learned wrt Megaprogramming

Transition/Mono/VG-11

NOTES



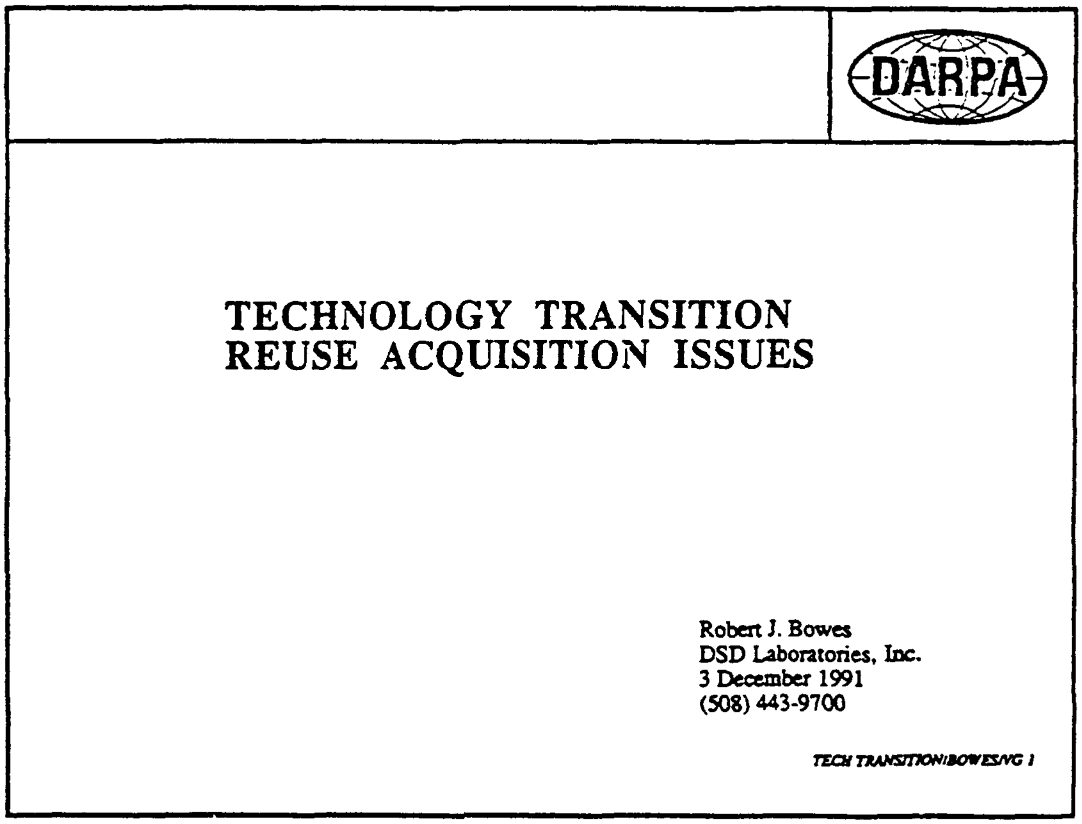
TRANSITION

CONCLUSIONS

- STARS Transition Strategy can fulfill the program objectives.
- Transition is premised on an active feedback loop.
- Transition will not occur without your active involvement:
  - Maintain your status as an Information Affiliate;
  - Seriously consider becoming a TT Affiliate;
  - In either case, provide us feedback on the vision, the process and product technologies, our transition activities, and on the cultural impact of all of the above.

Transition/Mocia/VG-12

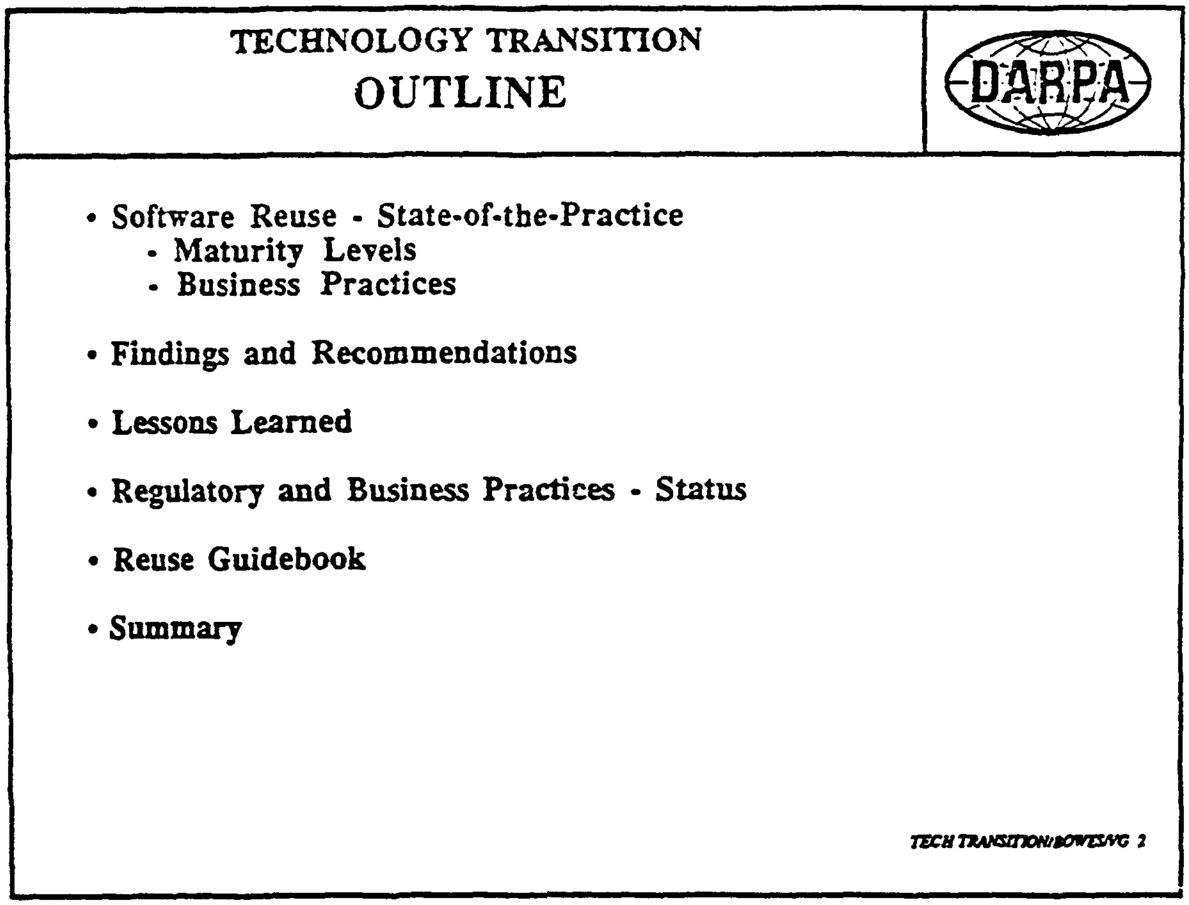
NOTES



TECHNOLOGY TRANSITION REUSE ACQUISITION ISSUES

Robert J. Bowes
DSD Laboratories, Inc.
3 December 1991
(508) 443-9700

TECH TRANSITION/BOWES/VG 1



TECHNOLOGY TRANSITION

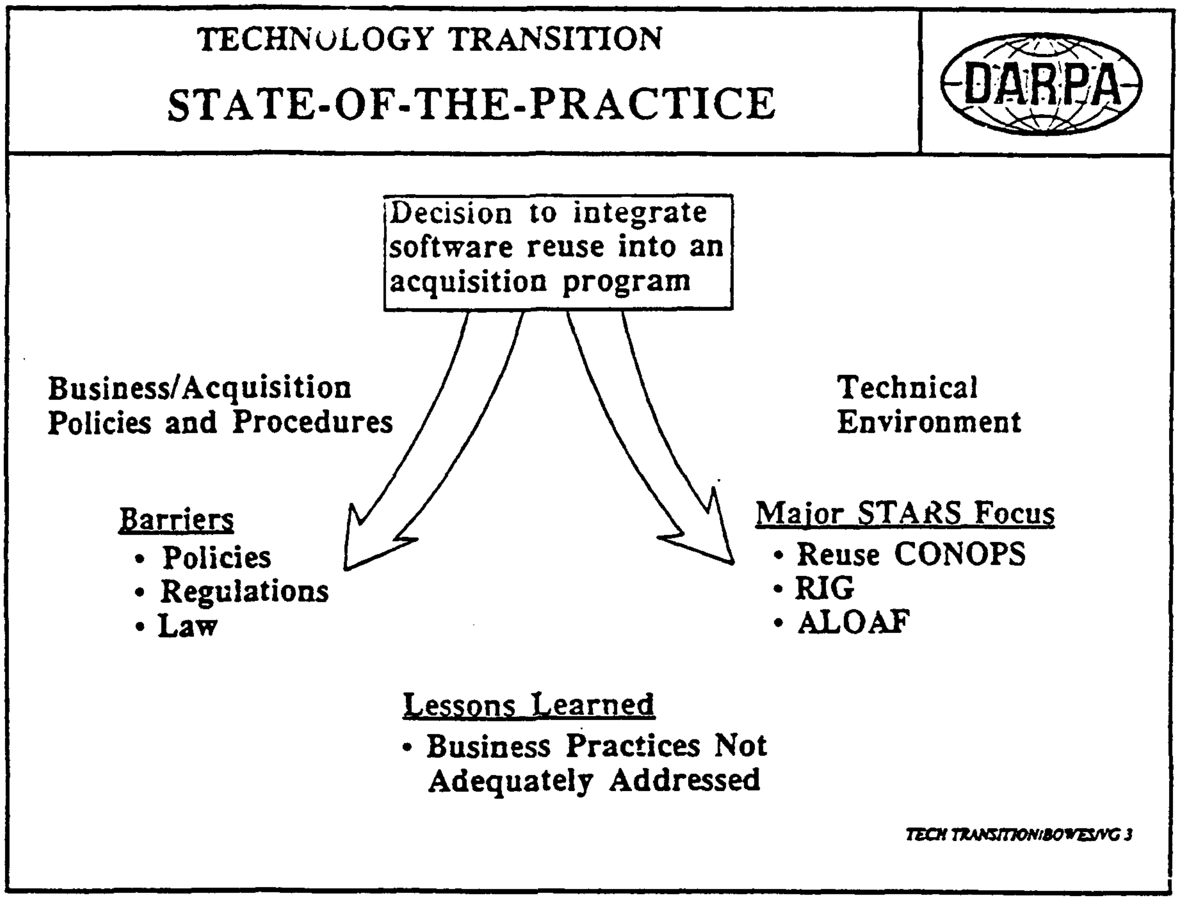
OUTLINE

DARPA

- Software Reuse - State-of-the-Practice
    - Maturity Levels
    - Business Practices
- Findings and Recommendations
- Lessons Learned
- Regulatory and Business Practices - Status
- Reuse Guidebook
- Summary

TECH TRANSITION/BOWES/VG 2

Provides an overview of what issues will be covered in this presentation.



TECHNOLOGY TRANSITION

STATE-OF-THE-PRACTICE

DARPA

Decision to integrate software reuse into an acquisition program

Business/Acquisition Policies and Procedures

Technical Environment

<u>Barriers</u>

- Policies
- Regulations
- Law

<u>Major STARS Focus</u>

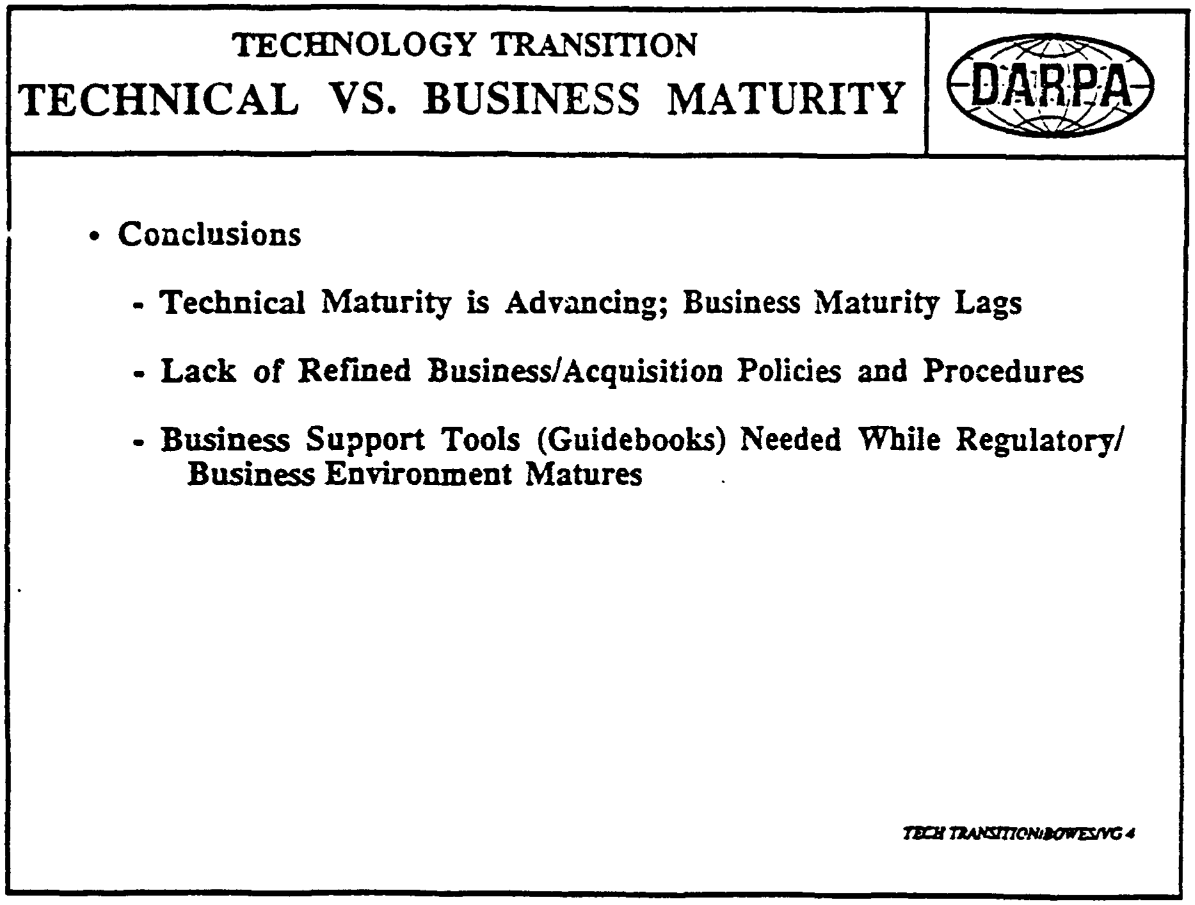
- Reuse CONOPS
- RIG
- ALOAF

<u>Lessons Learned</u>

- Business Practices Not Adequately Addressed

TECH TRANSITION/BOWES/VG 3

Focuses on the dual need for a set of business/acquisition policies and procedures and an appropriate technical environment to enable integration of software reuse into the acquisition process. Conclusion: In the past, business practices have been inadequately addressed to achieve successful integration.



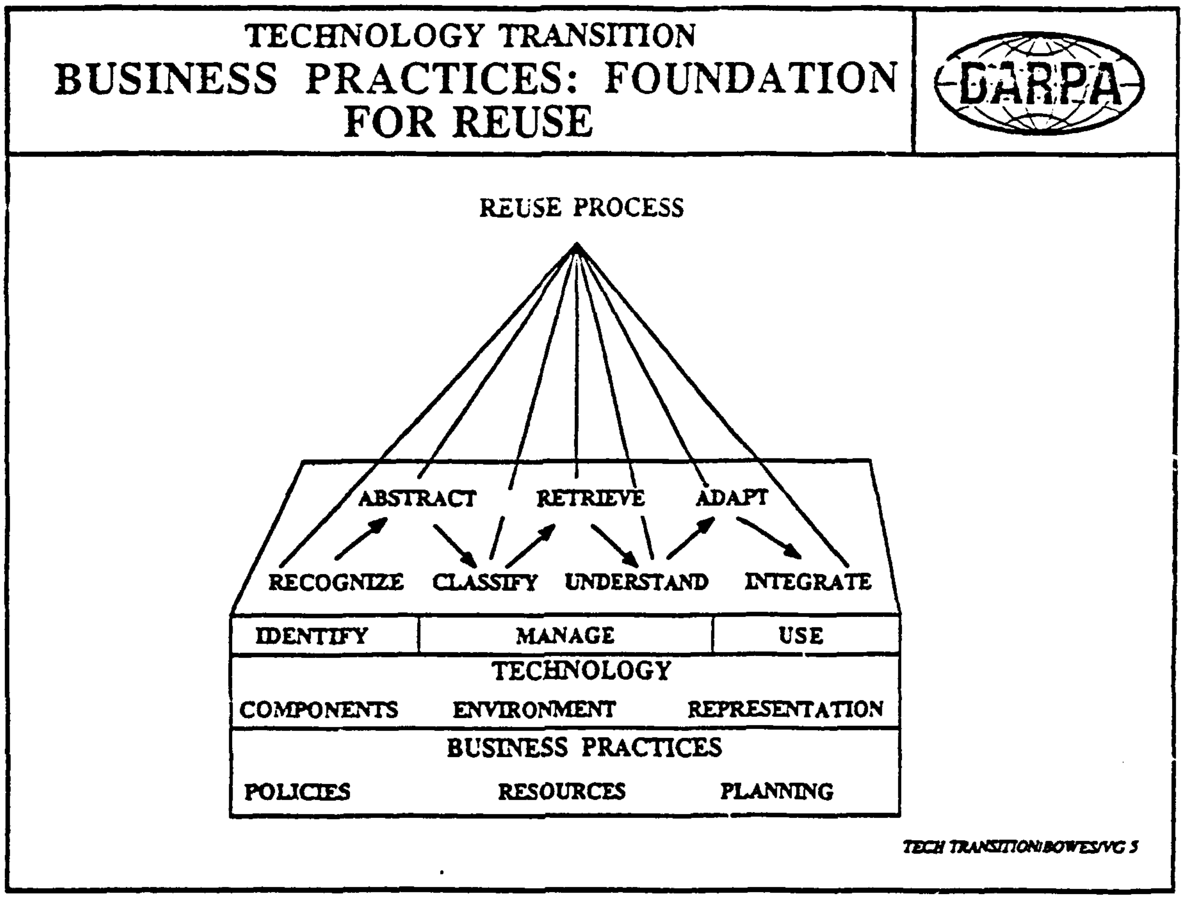
TECHNOLOGY TRANSITION

TECHNICAL VS. BUSINESS MATURITY

- Conclusions
  - Technical Maturity is Advancing; Business Maturity Lags
  - Lack of Refined Business/Acquisition Policies and Procedures
  - Business Support Tools (Guidebooks) Needed While Regulatory/ Business Environment Matures

TECH TRANSITION/BOWES/VG 4

Expands the theme that business practices maturity in software reuse lags far behind the technical advances. Also lays the groundwork for asserting the necessity for additional work and products to improve this significant and inappropriate discrepancy.



TECHNOLOGY TRANSITION

BUSINESS PRACTICES: FOUNDATION FOR REUSE

DARPA

REUSE PROCESS

ABSTRACT RETRIEVE ADAPT

RECOGNIZE CLASSIFY UNDERSTAND INTEGRATE

IDENTIFY	MANAGE	USE

TECHNOLOGY

COMPONENTS ENVIRONMENT REPRESENTATION

BUSINESS PRACTICES

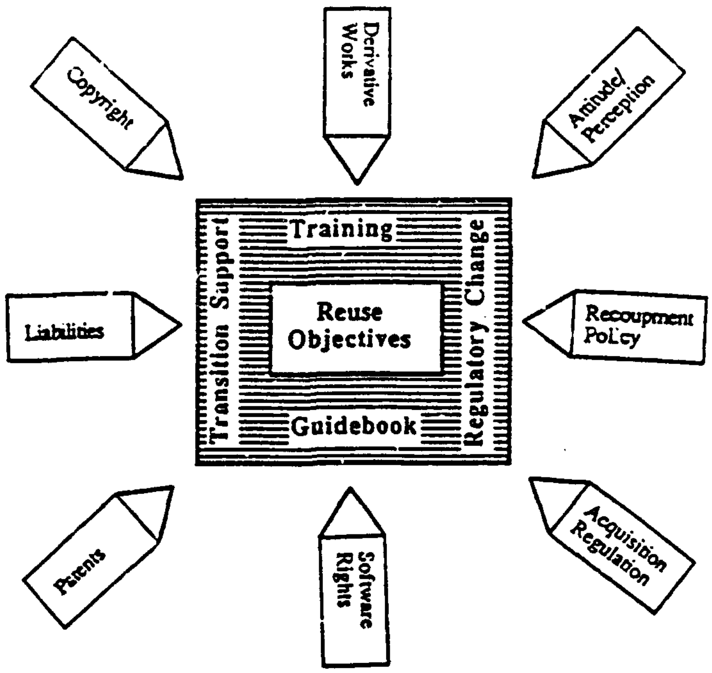
POLICIES RESOURCES PLANNING

TECH TRANSITION/BOWES/VG 5

In a diagrammatic form, subdivides the reuse structure into the realms of the reuse process, technology, and non-technical business practices. The diagram is designed to demonstrate the critical role played by business practices in establishing the foundation to support all technical initiatives. Without adequate efforts to address these essential underpinnings, the entire reuse structure could collapse.

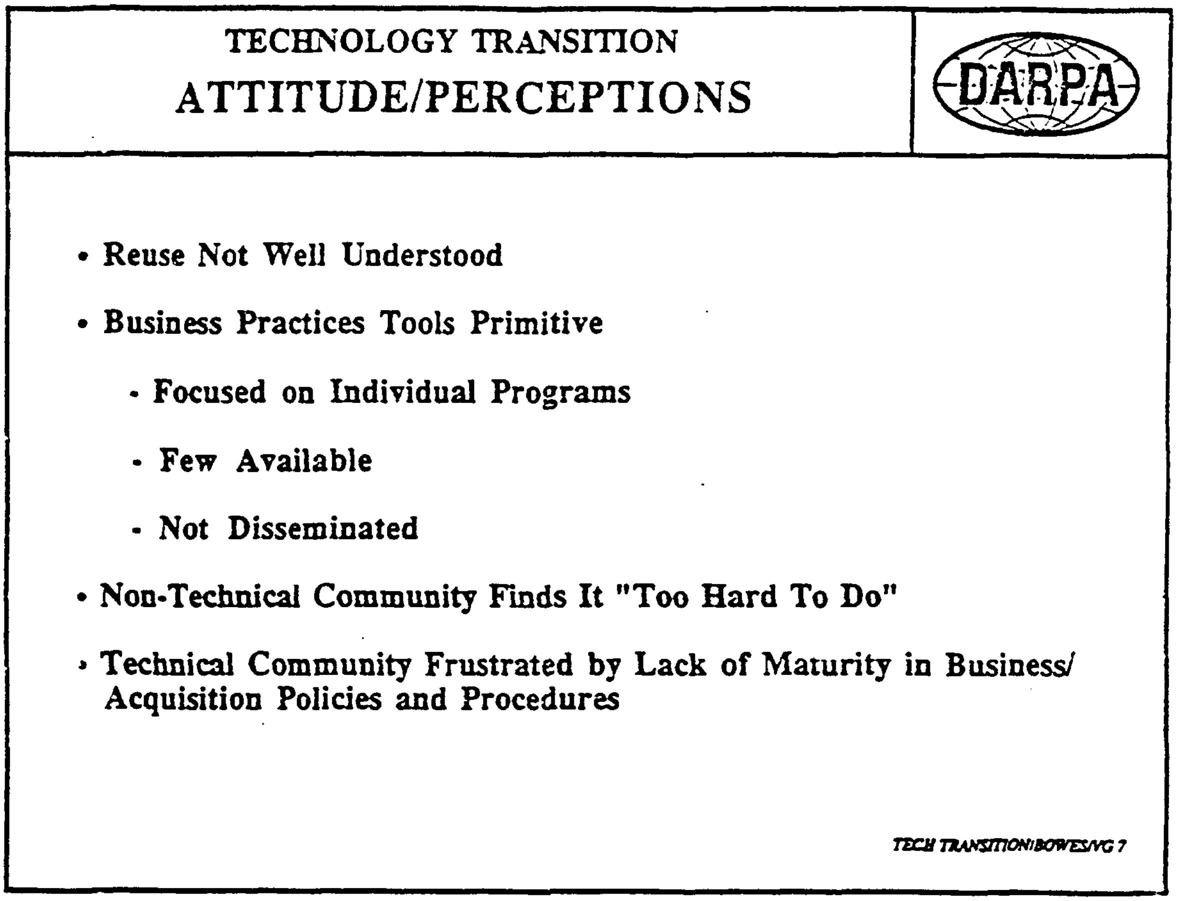
TECHNOLOGY TRANSITION

FINDINGS AND RECOMMENDATIONS



TECH TRANSITION/BOWES/VG 6

Expands the state-of-the-practice in reuse through use of a pictorial representation of a number of inhibitors which have the potential to undermine the achievement of present reuse objectives. The interior box highlights those proposed tools that may assist in neutralizing the inhibitors. Provides another illustration of the critical importance of business practices ir successful reuse implementation.



TECHNOLOGY TRANSITION

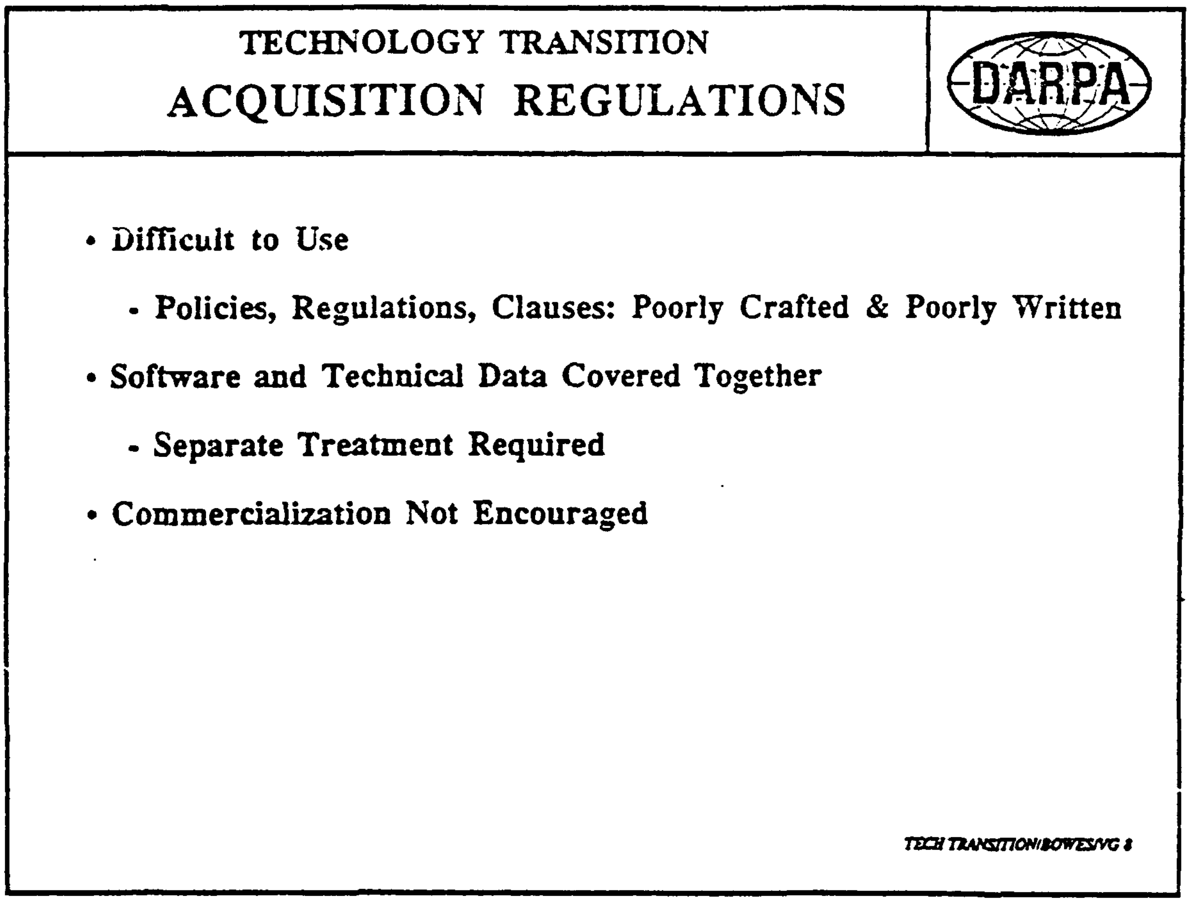
ATTITUDE/PERCEPTIONS

DARPA

- Reuse Not Well Understood
- Business Practices Tools Primitive
  - Focused on Individual Programs
  - Few Available
  - Not Disseminated
- Non-Technical Community Finds It "Too Hard To Do"
- Technical Community Frustrated by Lack of Maturity in Business/ Acquisition Policies and Procedures

TECH TRANSITION/BOWES/VG 7

Addresses some of the organizational behavicr issues that impact reuse.



TECHNOLOGY TRANSITION

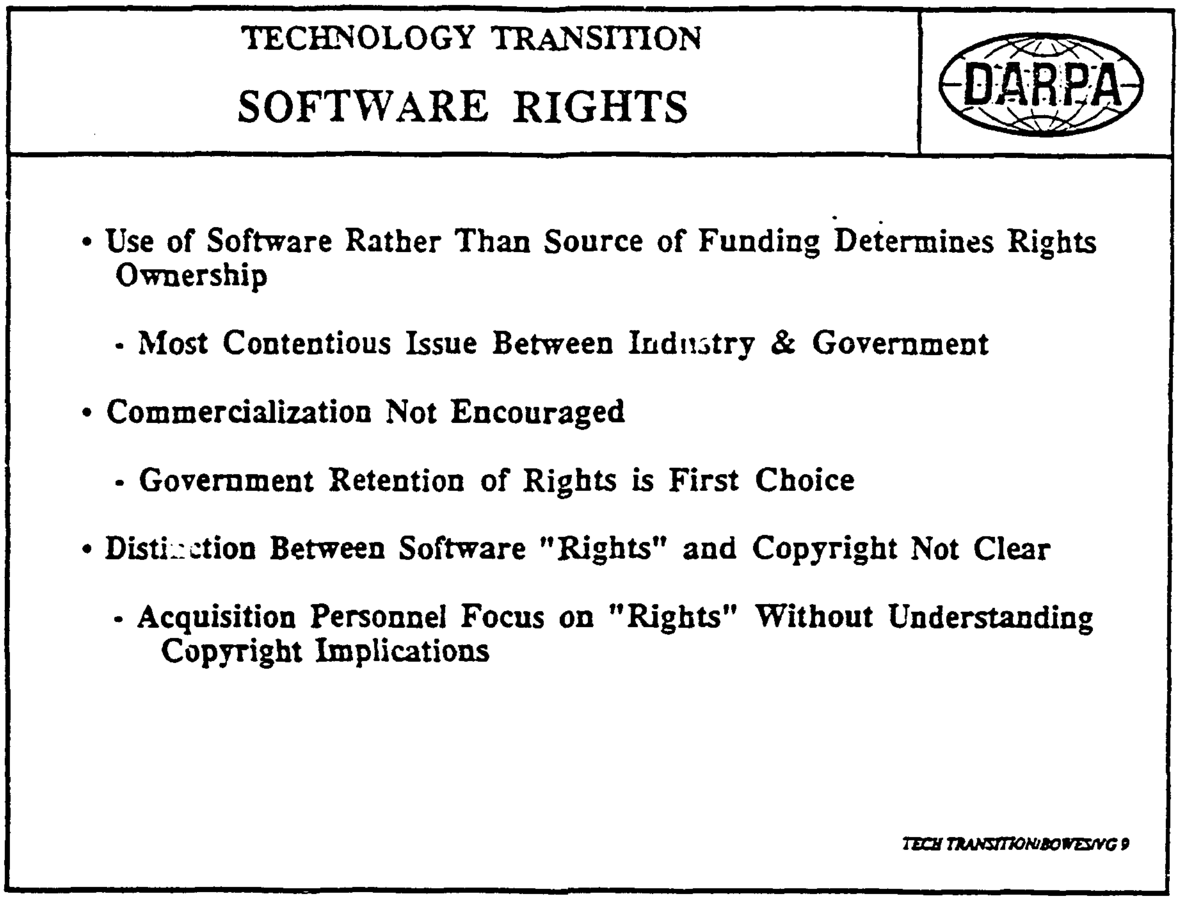
ACQUISITION REGULATIONS

DARPA

- Difficult to Use
  - Policies, Regulations, Clauses: Poorly Crafted & Poorly Written
- Software and Technical Data Covered Together
  - Separate Treatment Required
- Commercialization Not Encouraged

TECH TRANSITION/BOWES/VG 8

Highlights currently existing deficiencies within the acquisition regulations that govern the acquisition of software.



TECHNOLOGY TRANSITION

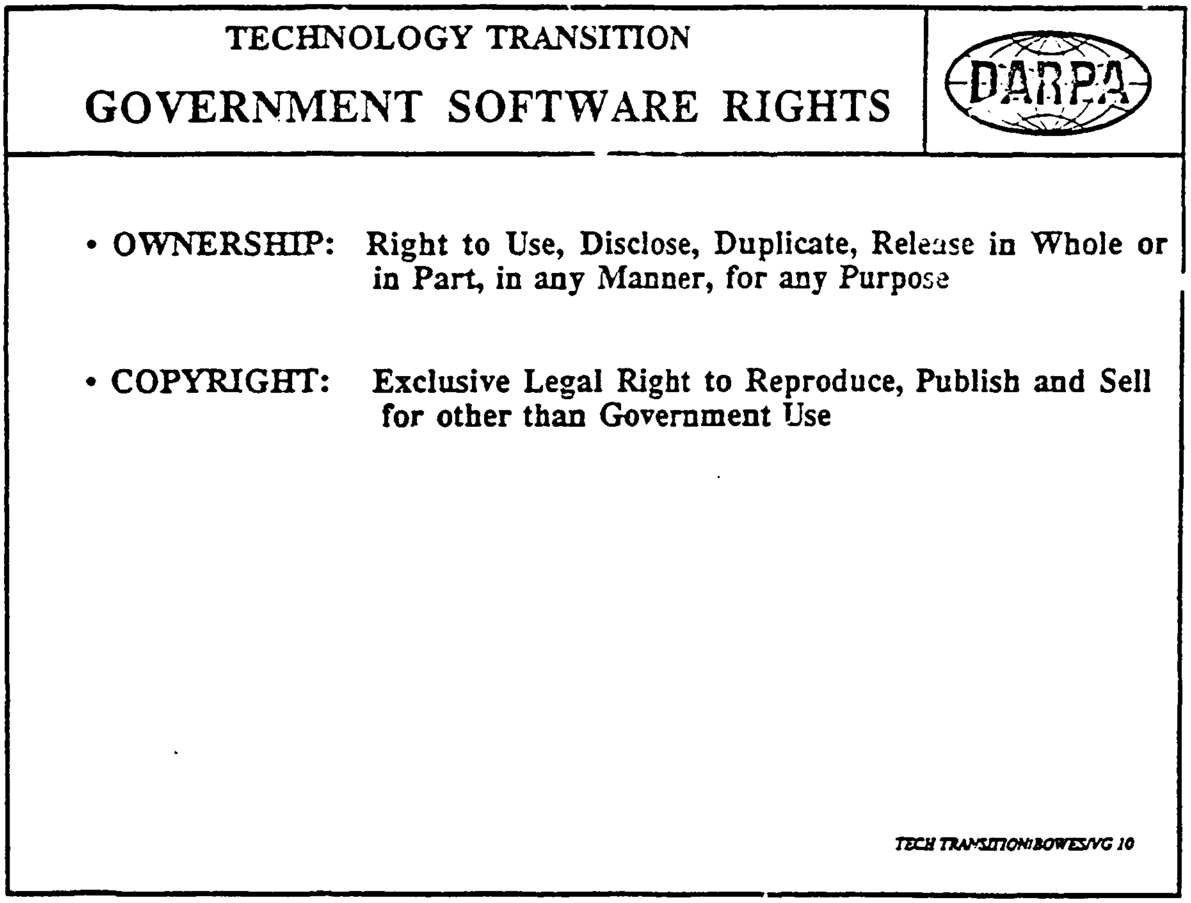
SOFTWARE RIGHTS

DARPA

- Use of Software Rather Than Source of Funding Determines Rights Ownership
  - Most Contentious Issue Between Industry & Government
- Commercialization Not Encouraged
  - Government Retention of Rights is First Choice
- Distinction Between Software "Rights" and Copyright Not Clear
  - Acquisition Personnel Focus on "Rights" Without Understanding Copyright Implications

TECH TRANSITION/BOWES/VG 9

Describes the confusion surrounding this issue, and the difficulties in resolving current contention between industry and the Government.



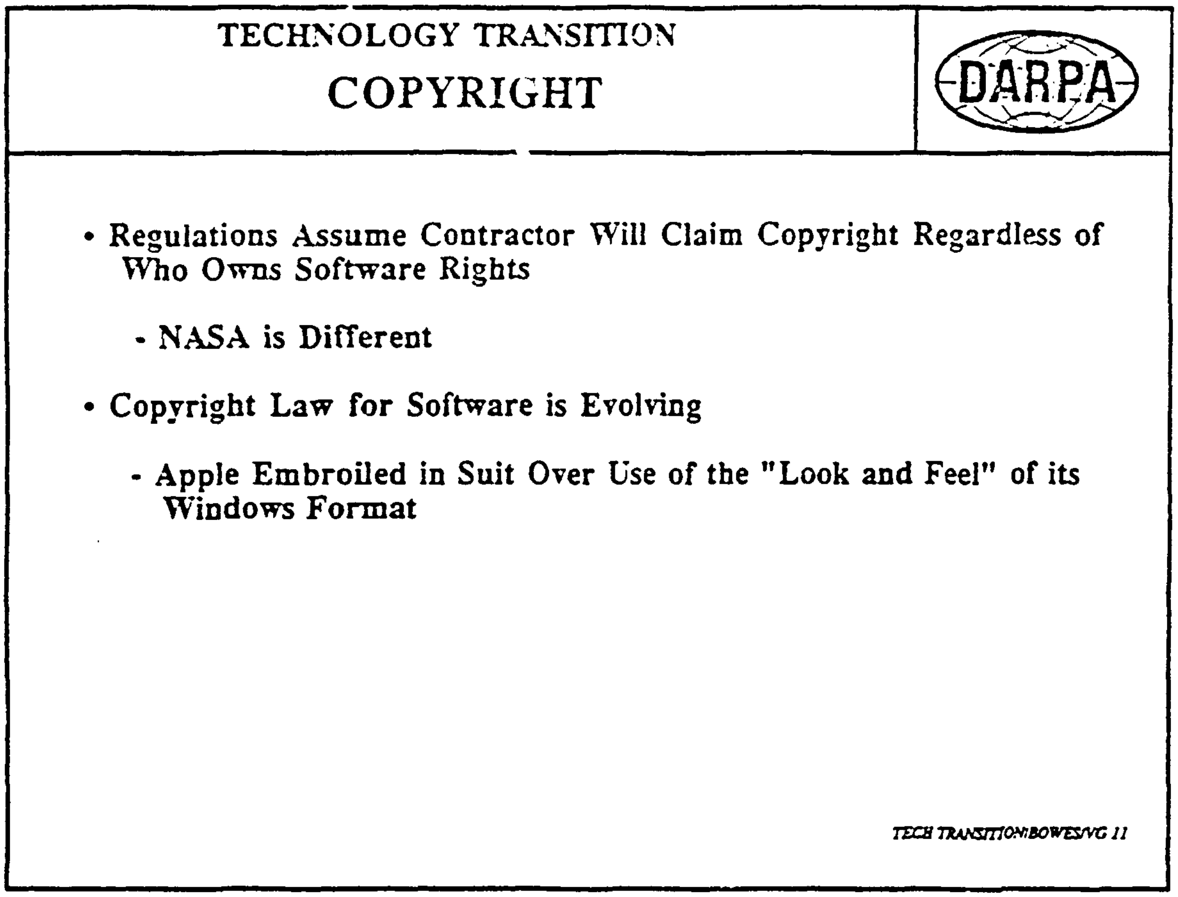
TECHNOLOGY TRANSITION

GOVERNMENT SOFTWARE RIGHTS

- OWNERSHIP: Right to Use, Disclose, Duplicate, Release in Whole or in Part, in any Manner, for any Purpose
- COPYRIGHT: Exclusive Legal Right to Reproduce, Publish and Sell for other than Government Use

TECH TRANSITION/BOWES/VG 10

Defines the terms ownership and copyright, to enhance understanding of the issues involved in their use. Highlights the confusion that arises, since the inherent rights specified herein often conflict.



TECHNOLOGY TRANSITION

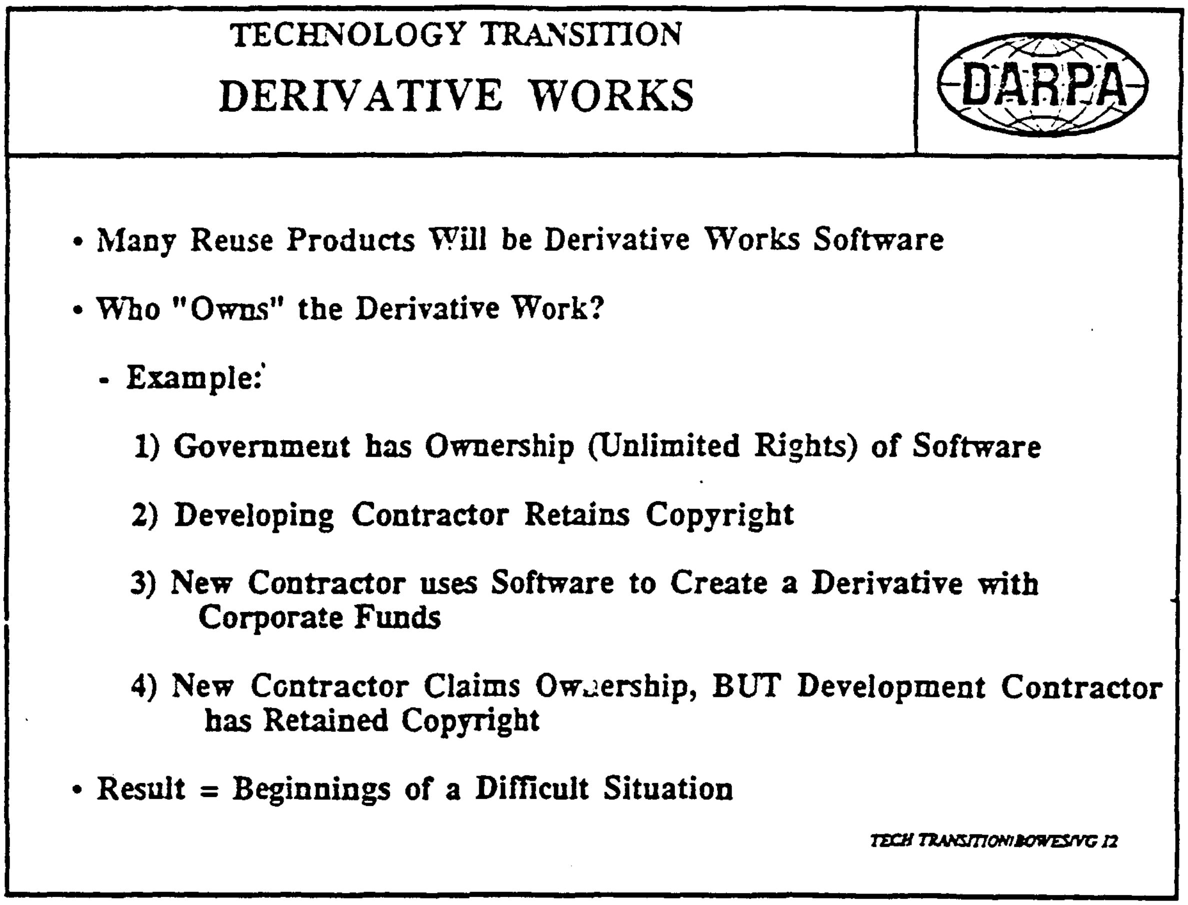
COPYRIGHT

DARPA

- Regulations Assume Contractor Will Claim Copyright Regardless of Who Owns Software Rights
  - NASA is Different
- Copyright Law for Software is Evolving
  - Apple Embroiled in Suit Over Use of the "Look and Feel" of its Windows Format

TECH TRANSITION/BOWES/VG 11

Use of copyrights is described, and an example of current litigation is explored.



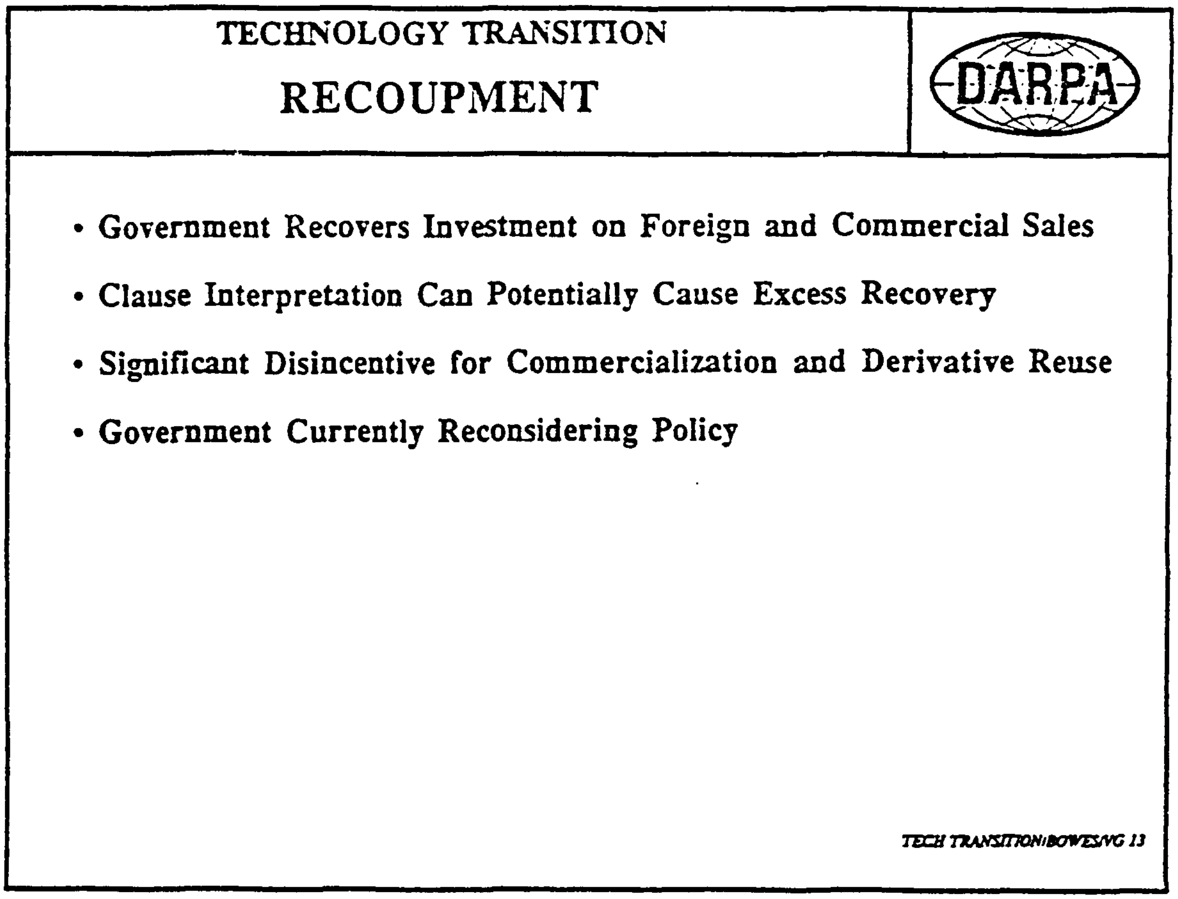
TECHNOLOGY TRANSITION

DERIVATIVE WORKS

- Many Reuse Products Will be Derivative Works Software
- Who "Owns" the Derivative Work?
  - Example:
    1) Government has Ownership (Unlimited Rights) of Software
    2) Developing Contractor Retains Copyright
    3) New Contractor uses Software to Create a Derivative with Corporate Funds
    4) New Contractor Claims Ownership, BUT Development Contractor has Retained Copyright
- Result = Beginnings of a Difficult Situation

TECH TRANSITION\BOWES\VG 12

A further complication of the ownership issue is addressed.



TECHNOLOGY TRANSITION

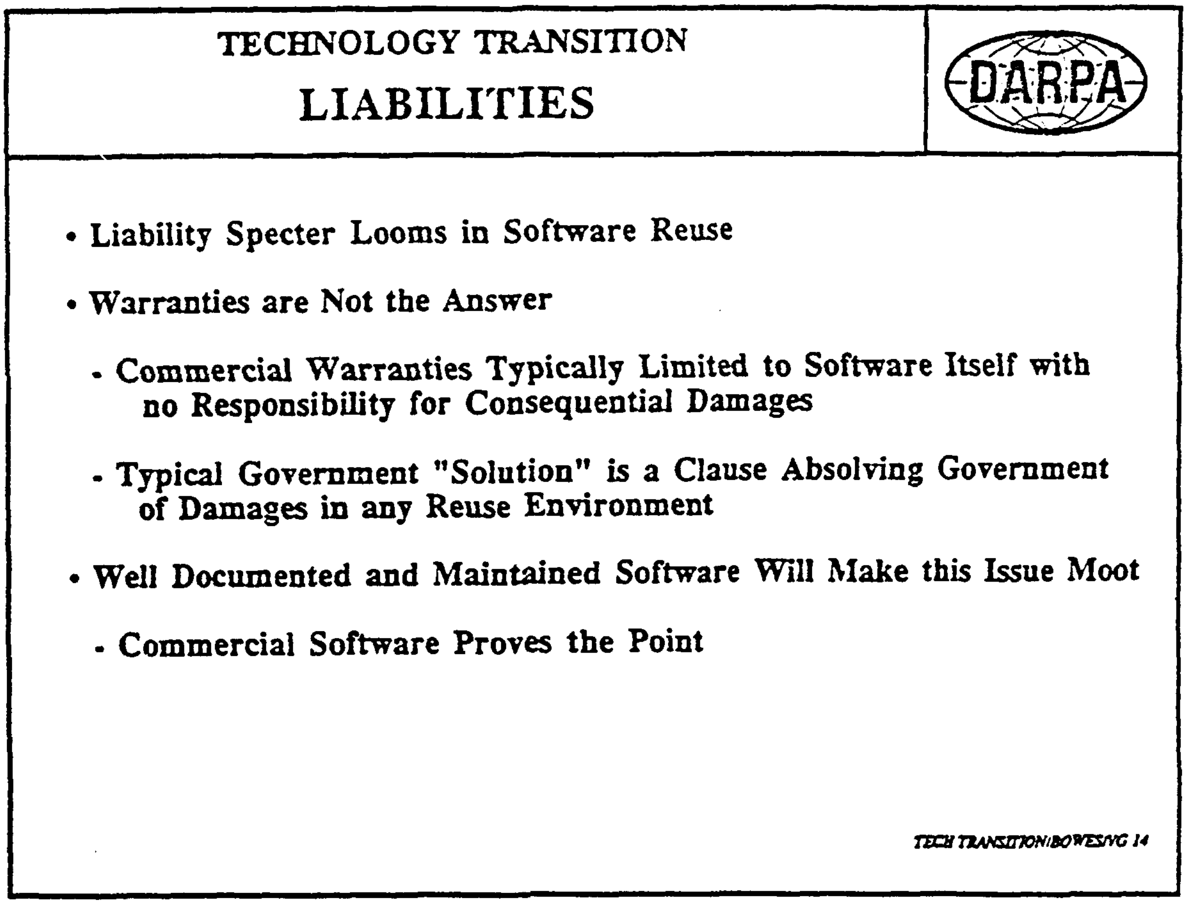
RECOUPMENT

DARPA

- Government Recovers Investment on Foreign and Commercial Sales
- Clause Interpretation Can Potentially Cause Excess Recovery
- Significant Disincentive for Commercialization and Derivative Reuse
- Government Currently Reconsidering Policy

TECH TRANSITION/BOWES/VG 13

Highlights the difficulties in commercializing reusable software due to recoupment, and the potential for the Government to recover more than its original investment in software development.



TECHNOLOGY TRANSITION

LIABILITIES

DARPA

- Liability Specter Looms in Software Reuse
- Warranties are Not the Answer
  - Commercial Warranties Typically Limited to Software Itself with no Responsibility for Consequential Damages
  - Typical Government "Solution" is a Clause Absolving Government of Damages in any Reuse Environment
- Well Documented and Maintained Software Will Make this Issue Moot
  - Commercial Software Proves the Point

TECH TRANSITION/BOWES/VG 14

Issues relating to the liability impediment are addressed.

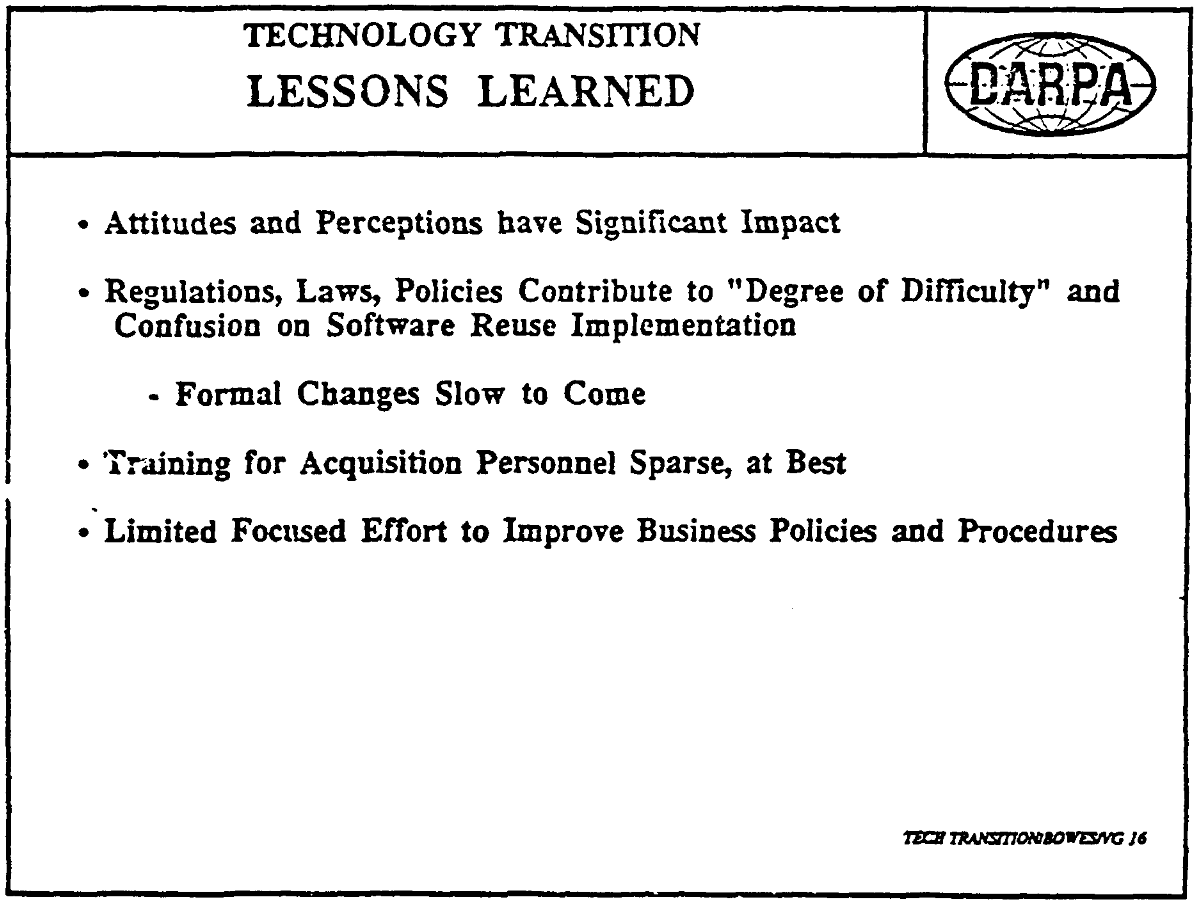
TECHNOLOGY TRANSITION

PATENTS

- Software Patents are Relatively New Phenomena
  - Use is Increasing
- Significant Disagreement on "Patentability" of Software
- Further Complications for Derivative Works

TECH TRANSITION/BOWES/VG 15

Addresses uncertainties associated with applying patent protection to software, and inherent difficulties for future users of such software.



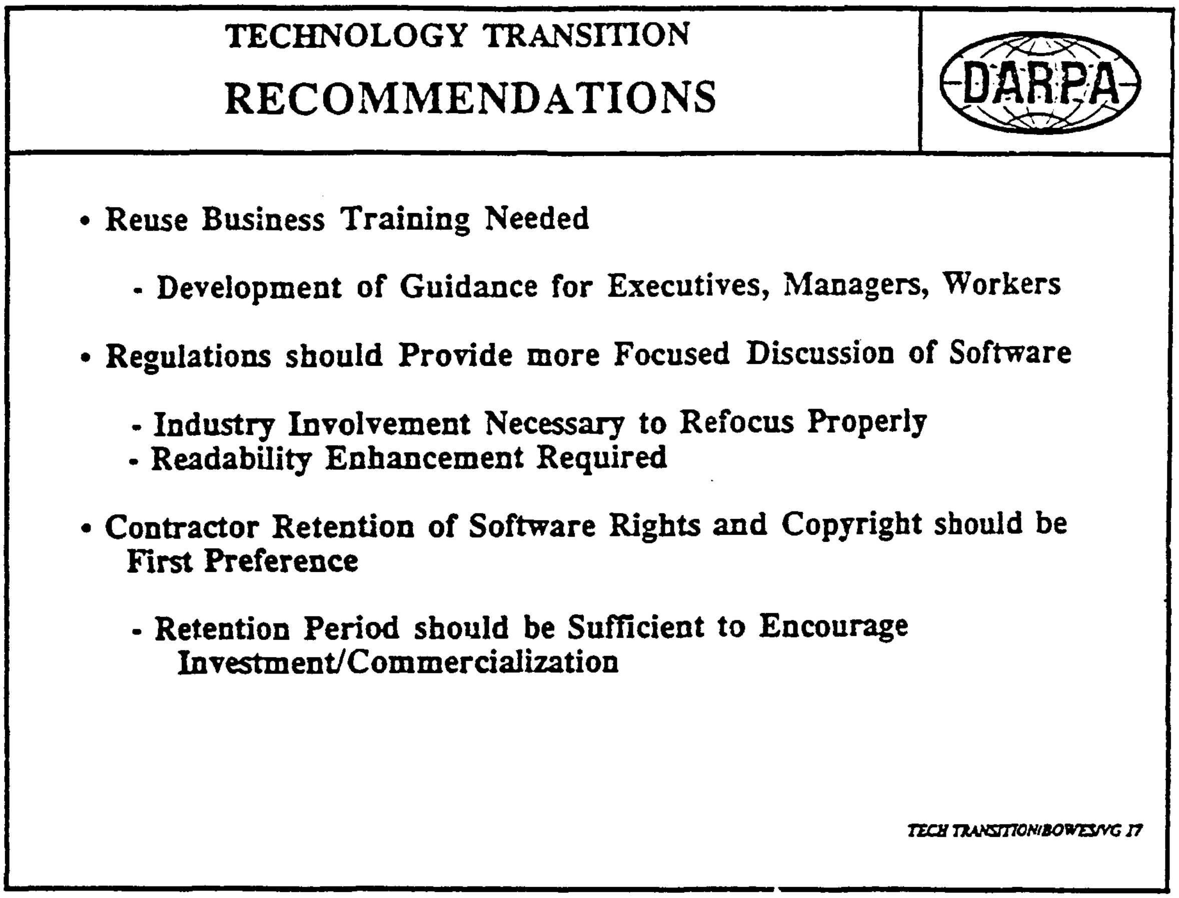
TECHNOLOGY TRANSITION

LESSONS LEARNED

- Attitudes and Perceptions have Significant Impact
- Regulations, Laws, Policies Contribute to "Degree of Difficulty" and Confusion on Software Reuse Implementation
  - Formal Changes Slow to Come
- Training for Acquisition Personnel Sparse, at Best
- Limited Focused Effort to Improve Business Policies and Procedures

TECH TRANSITION/BOWES/VG 16

Addresses some of the lessons learned with respect to each of the impediments previously outlined, based on our work to date. A subset of these is described, with subsequent recommendations to support corrections/improvements.



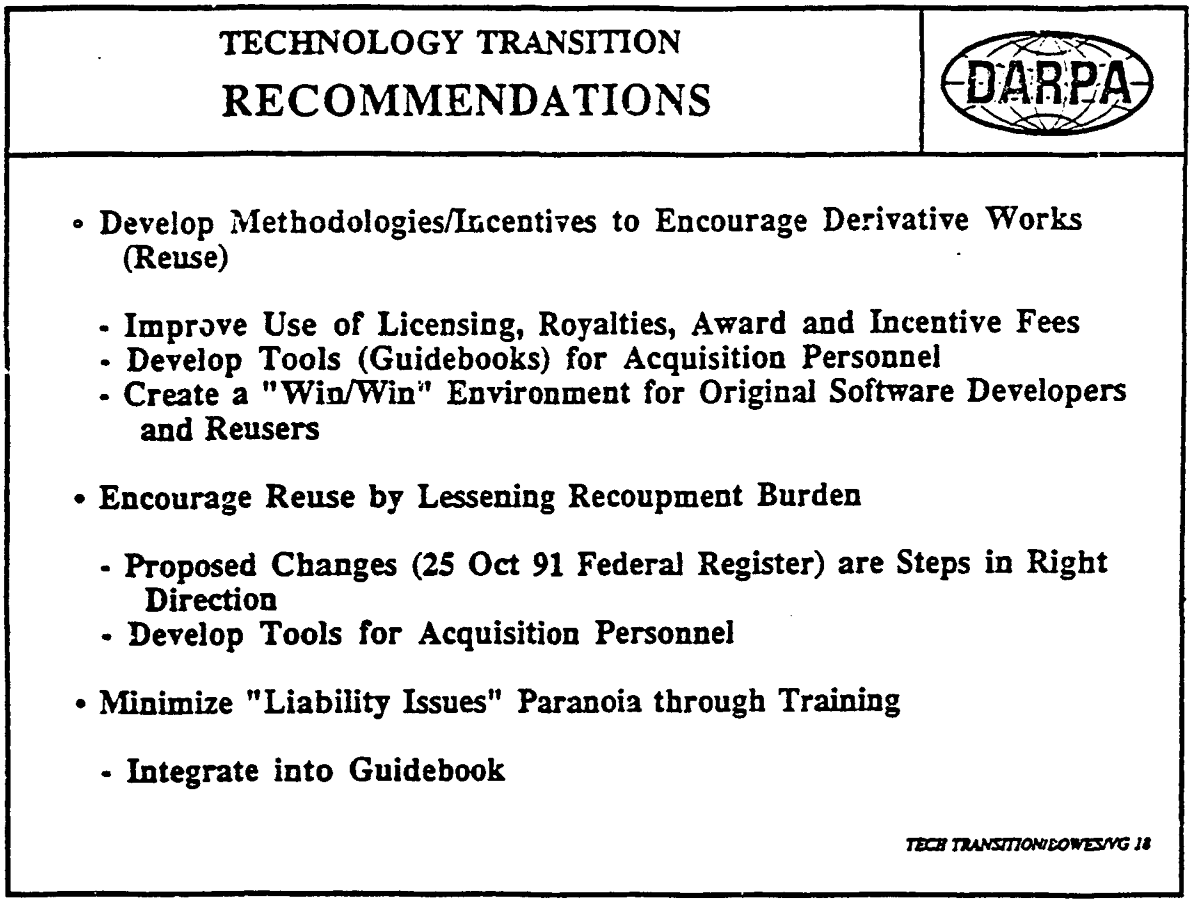
TECHNOLOGY TRANSITION

RECOMMENDATIONS

- Reuse Business Training Needed
  - Development of Guidance for Executives, Managers, Workers
- Regulations should Provide more Focused Discussion of Software
  - Industry Involvement Necessary to Refocus Properly
  - Readability Enhancement Required
- Contractor Retention of Software Rights and Copyright should be First Preference
  - Retention Period should be Sufficient to Encourage Investment/Commercialization

TECH TRANSITION/BOWES/VG 17

Discusses issues pertaining to training, modification of regulations, and contractor retention of rights and copyrights.



TECHNOLOGY TRANSITION

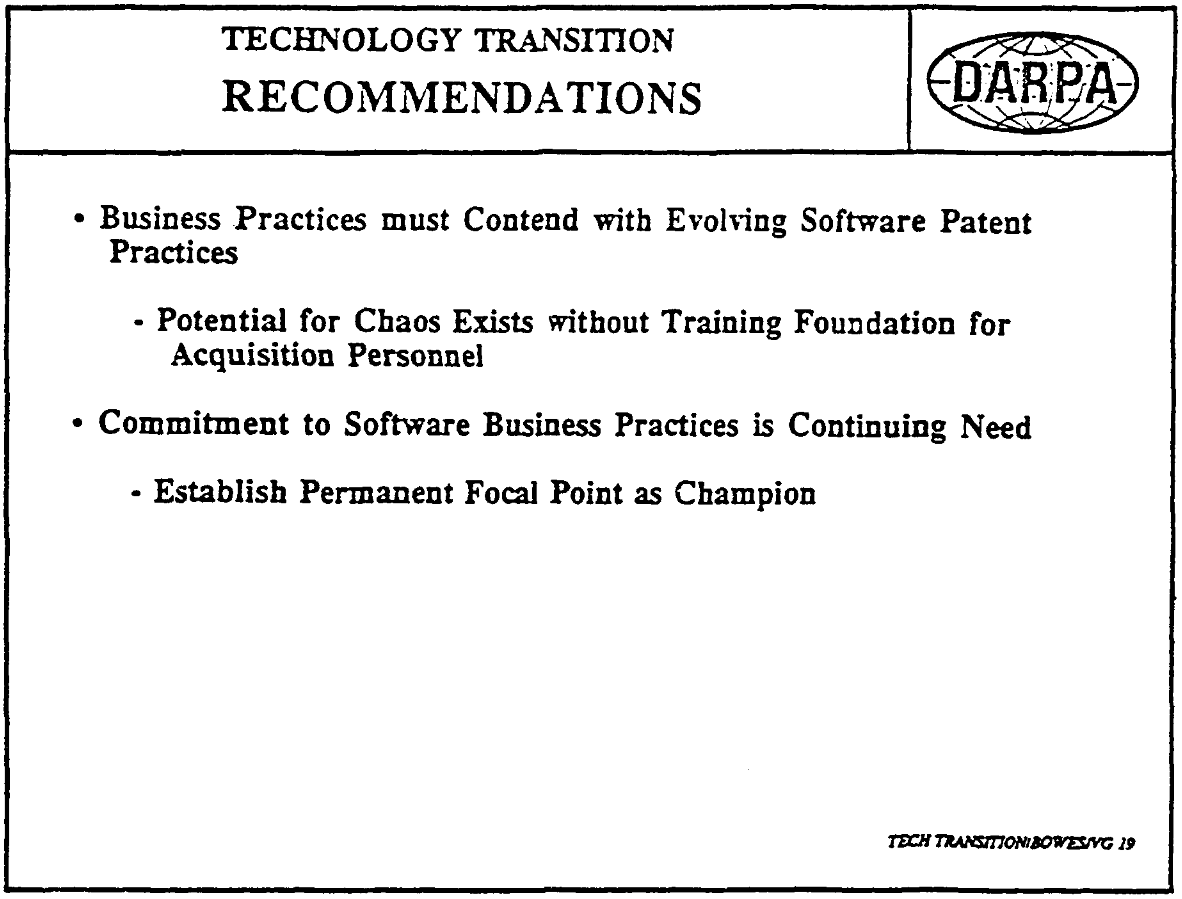
RECOMMENDATIONS

DARPA

- Develop Methodologies/Incentives to Encourage Derivative Works (Reuse)
  - Improve Use of Licensing, Royalties, Award and Incentive Fees
  - Develop Tools (Guidebooks) for Acquisition Personnel
  - Create a "Win/Win" Environment for Original Software Developers and Reusers
- Encourage Reuse by Lessening Recoupment Burden
  - Proposed Changes (25 Oct 91 Federal Register) are Steps in Right Direction
  - Develop Tools for Acquisition Personnel
- Minimize "Liability Issues" Paranoia through Training
  - Integrate into Guidebook

TECH TRANSITION/BOWES/VG 18

Discusses means of stimulating reuse.



TECHNOLOGY TRANSITION

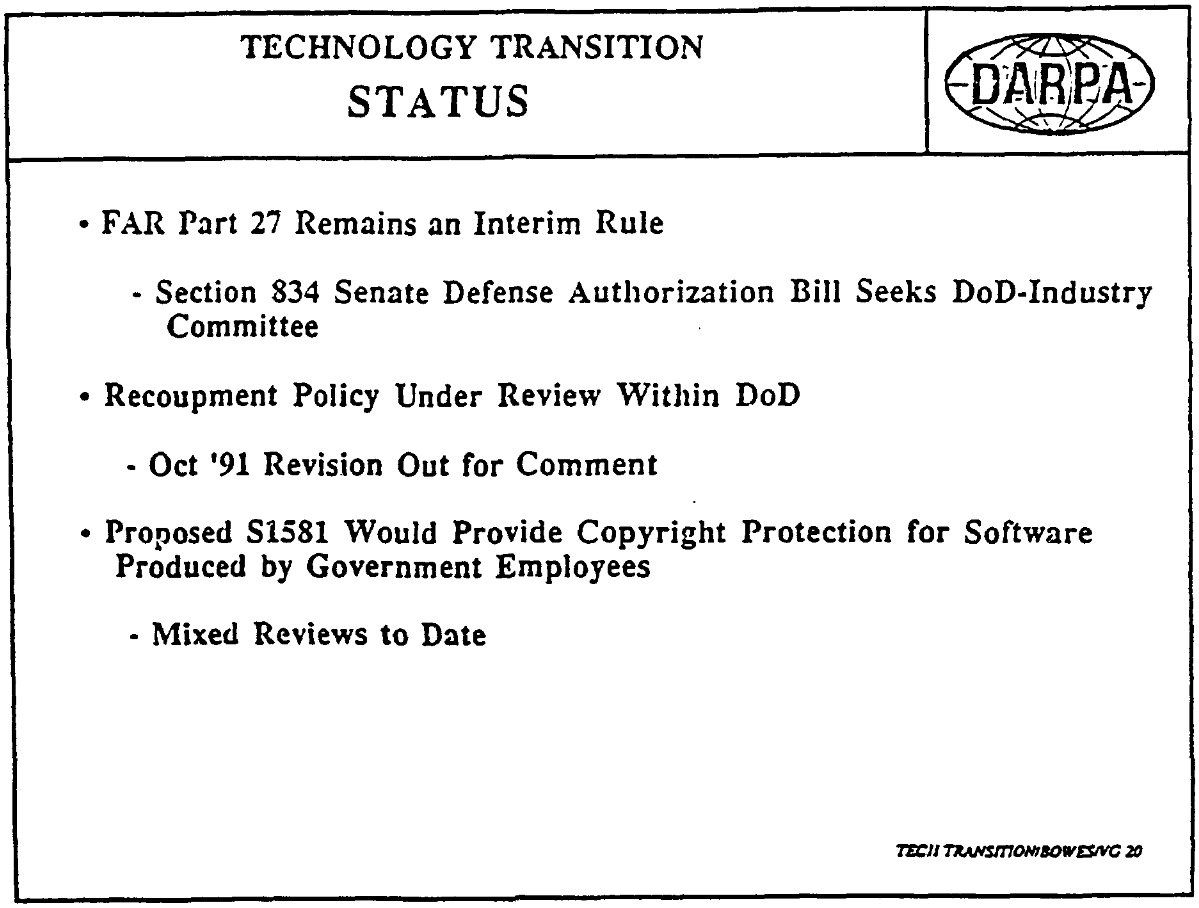
RECOMMENDATIONS

DARPA

- Business Practices must Contend with Evolving Software Patent Practices
  - Potential for Chaos Exists without Training Foundation for Acquisition Personnel
- Commitment to Software Business Practices is Continuing Need
  - Establish Permanent Focal Point as Champion

TECH TRANSITION/BOWES/VG 19

Highlights some critical issues pertaining to business practices.



TECHNOLOGY TRANSITION

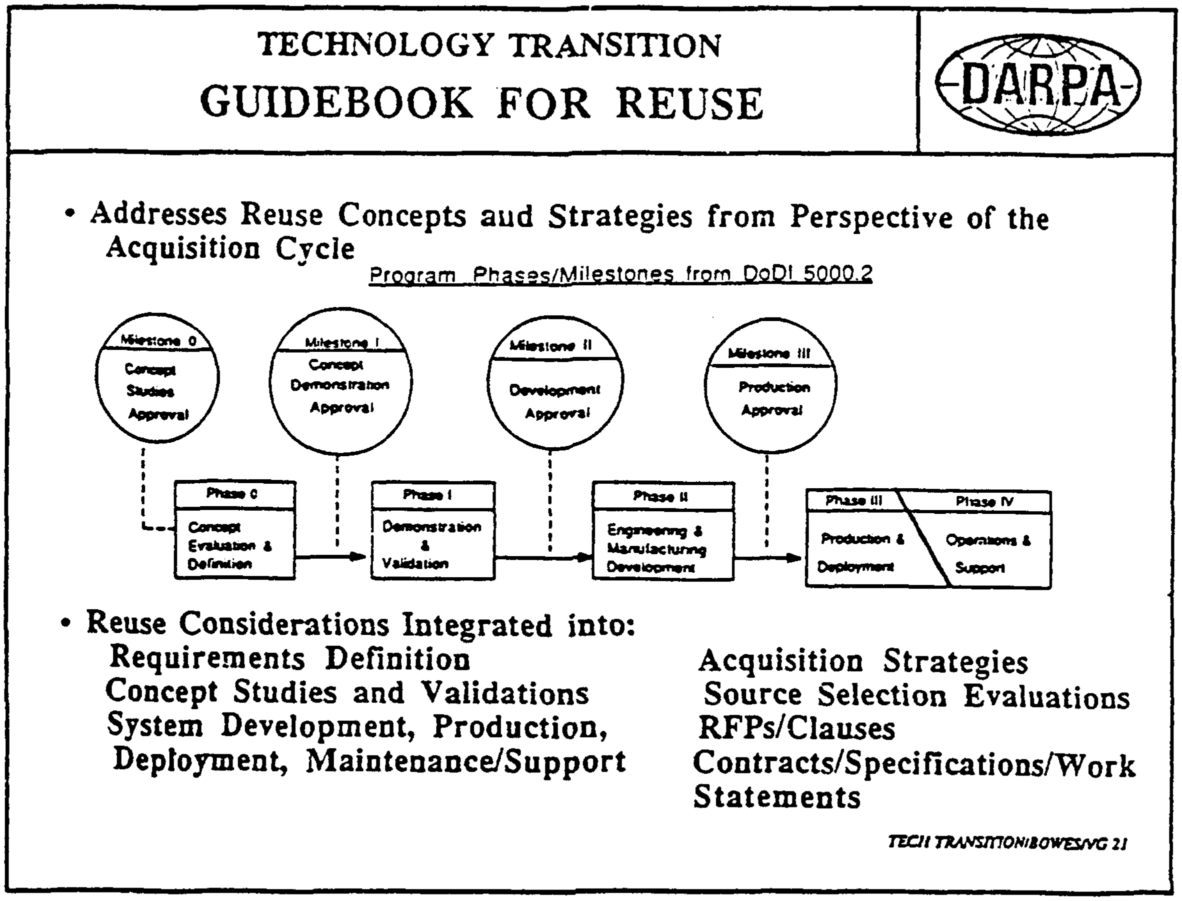
STATUS

DARPA

- FAR Part 27 Remains an Interim Rule
  - Section 834 Senate Defense Authorization Bill Seeks DoD-Industry Committee
- Recoupment Policy Under Review Within DoD
  - Oct '91 Revision Out for Comment
- Proposed S1581 Would Provide Copyright Protection for Software Produced by Government Employees
  - Mixed Reviews to Date

TECH TRANSITION/BOWES/VC 20

After noting areas of concern, lessons learned, and appropriate recommendations, we provide the current status of regulatory initiatives.



TECHNOLOGY TRANSITION

GUIDEBOOK FOR REUSE

DARPA

- Addresses Reuse Concepts aud Strategies from Perspective of the Acquisition Cycle

Program Phases/Milestones from DoDI 5000.2

Milestone 0
Concept Studies Approval

Milestone I
Concept Demonstration Approval

Milestone II
Development Approval

Milestone III
Production Approval

Phase 0
Concept Evaluation & Definition

Phase I
Demonstration & Validation

Phase II
Engineering & Manufacturing Development

Phase III
Production & Deployment

Phase IV
Operations & Support

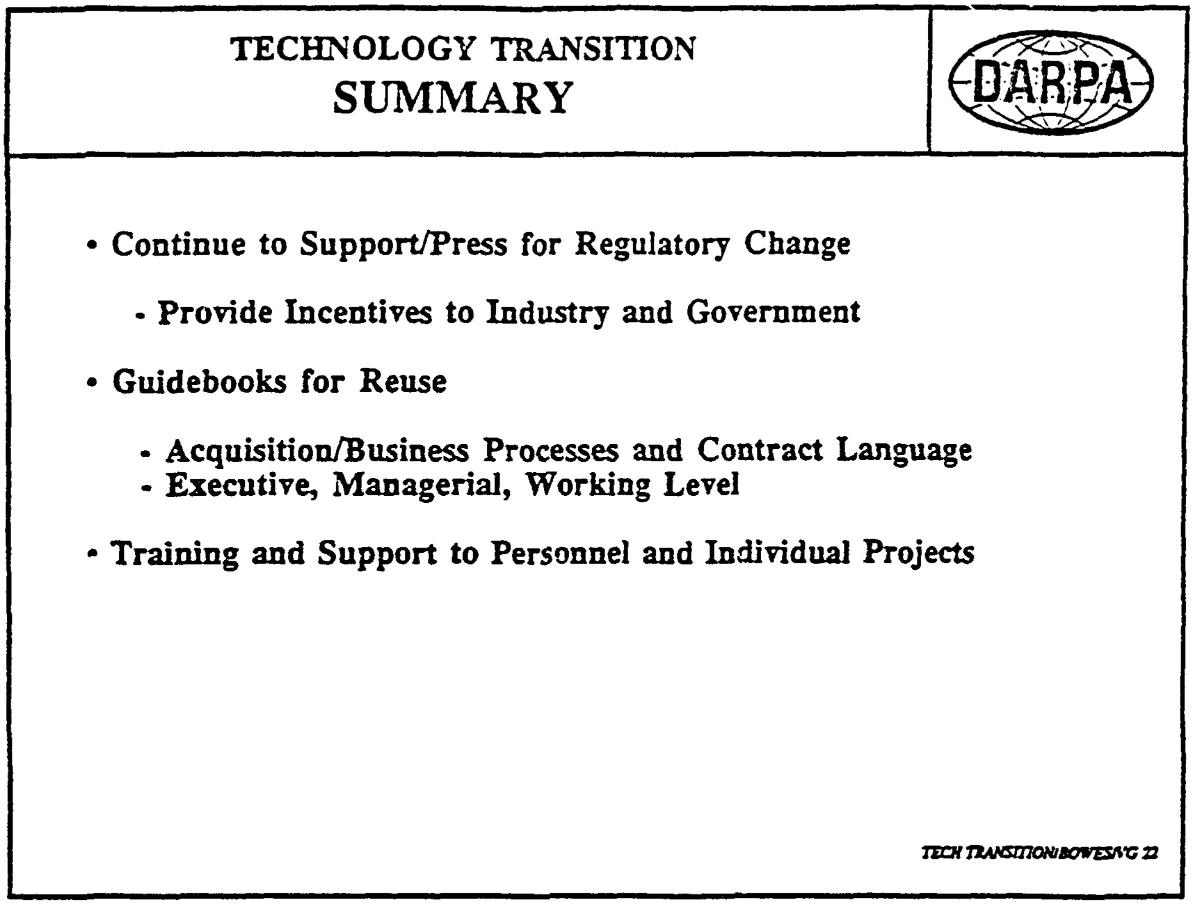
- Reuse Considerations Integrated into:

Requirements Definition
Concept Studies and Validations
System Development, Production, Deployment, Maintenance/Support

Acquisition Strategies
Source Selection Evaluations
RFPs/Clauses
Contracts/Specifications/Work Statements

TECH TRANSITION/BOWES/VG 21

We have established a basis for identifying key voids in business practices issues. Consequently, in order to support software reuse objectives, we describe our ongoing work aimed toward developing a guidebook to address the needs of high-level acquisition executives.



TECHNOLOGY TRANSITION

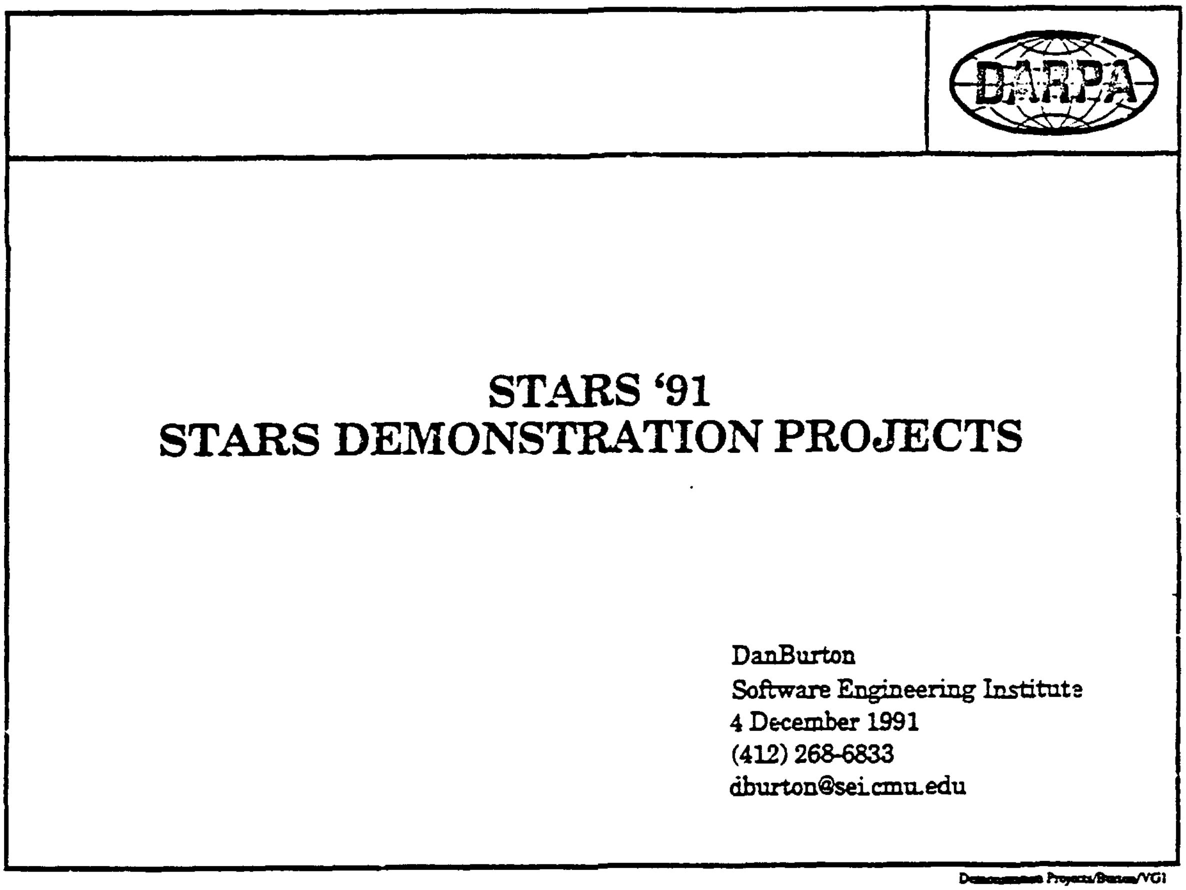
SUMMARY

DARPA

- Continue to Support/Press for Regulatory Change
  - Provide Incentives to Industry and Government
- Guidebooks for Reuse
  - Acquisition/Business Processes and Contract Language
  - Executive, Managerial, Working Level
- Training and Support to Personnel and Individual Projects

TECH TRANSITION/BOWES/VG 22

Recaps importance of addressing key business practices issues, in order to ensure future success of reuse initiatives.



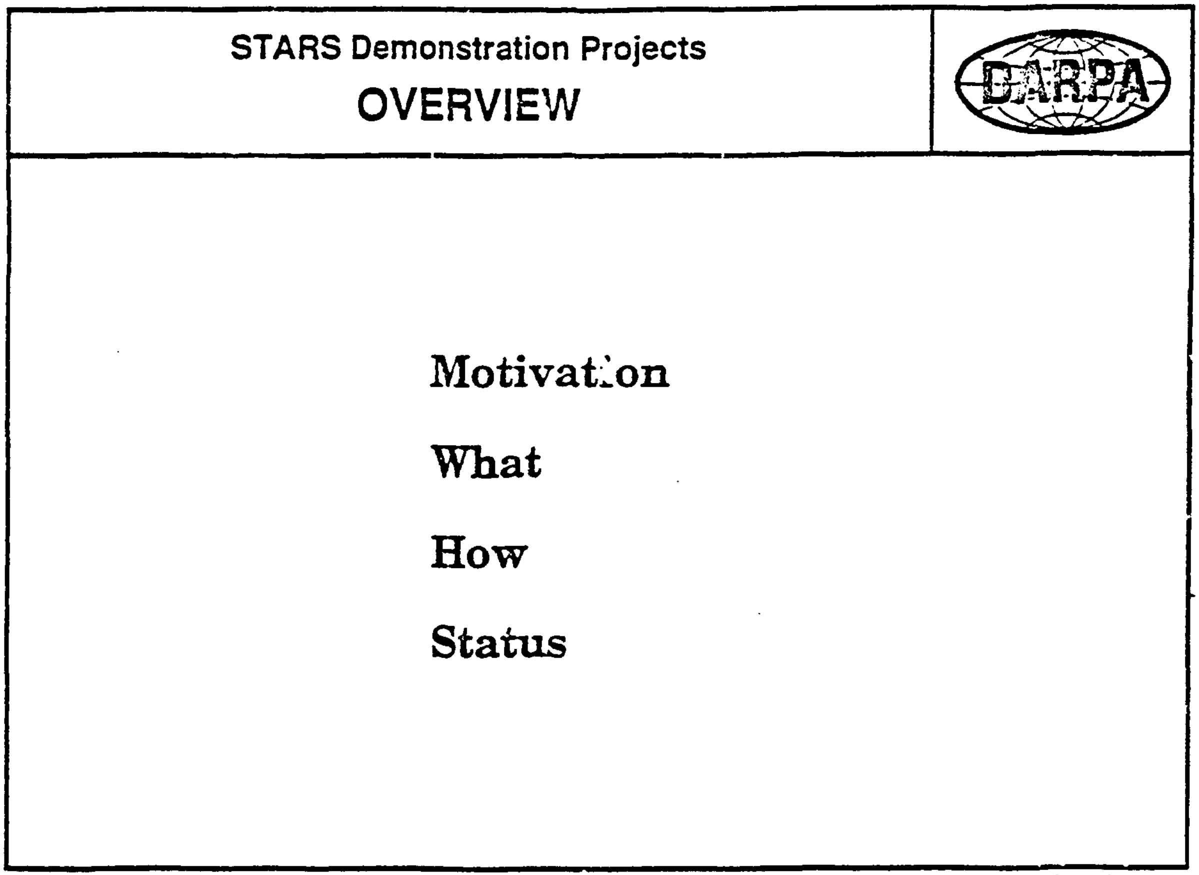
DARPA

STARS '91
STARS DEMONSTRATION PROJECTS

DanBurton
Software Engineering Institute
4 December 1991
(412) 268-6833
dburton@sei.cmu.edu

Demonstration Projects/Burton/VG1

In earlier presentations, STARS demonstration projects have been mentioned. This presentation provides a more in-depth look at STARS demonstration projects.



STARS Demonstration Projects

OVERVIEW

Motivat:on

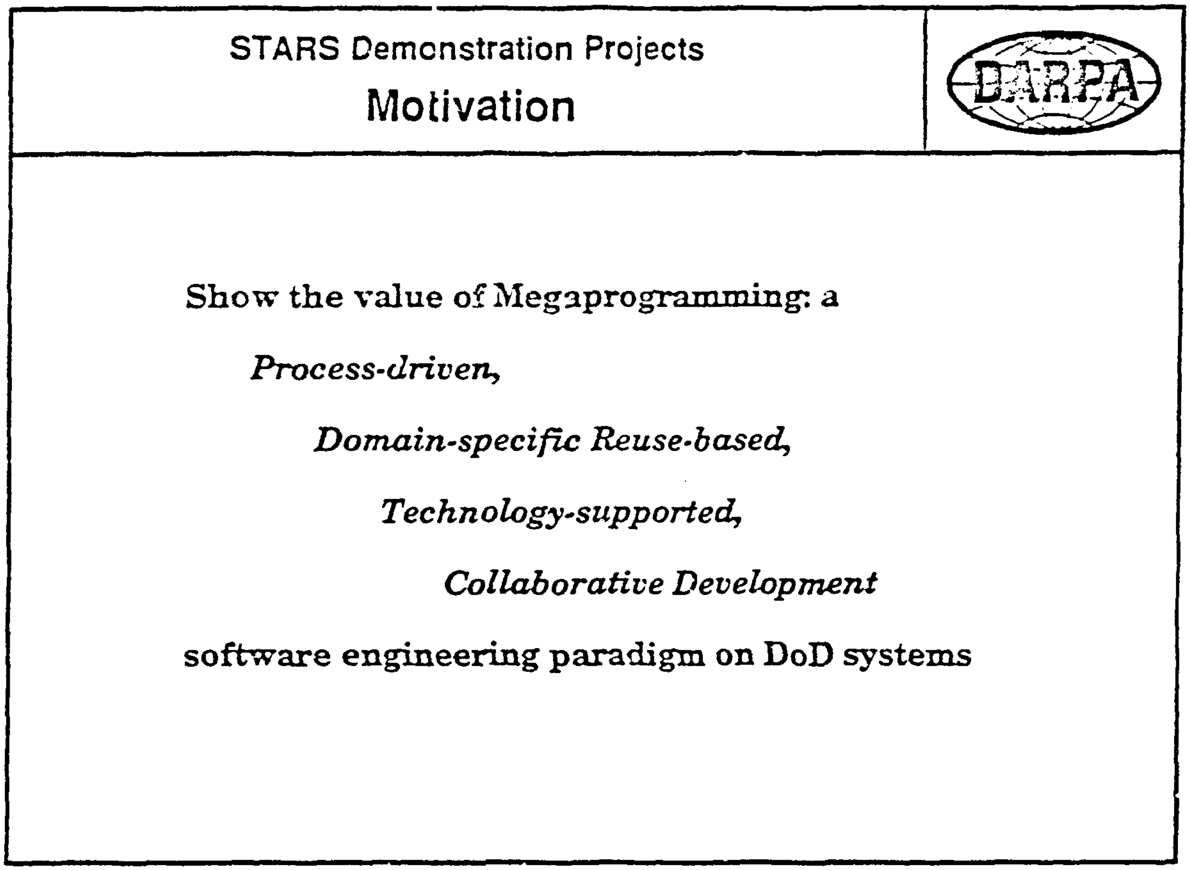
What

How

Status

Demonstration Projects/Barton/VG2

I'll talk about why STARS is doing demonstration projects; what we see as demonstration projects; how we plan to identify, select, and run the demonstration projects; and where we are in this process today.



STARS Demonstration Projects

Motivation

Show the value of Megaprogramming: a

Process-driven,

Domain-specific Reuse-based,

Technology-supported,

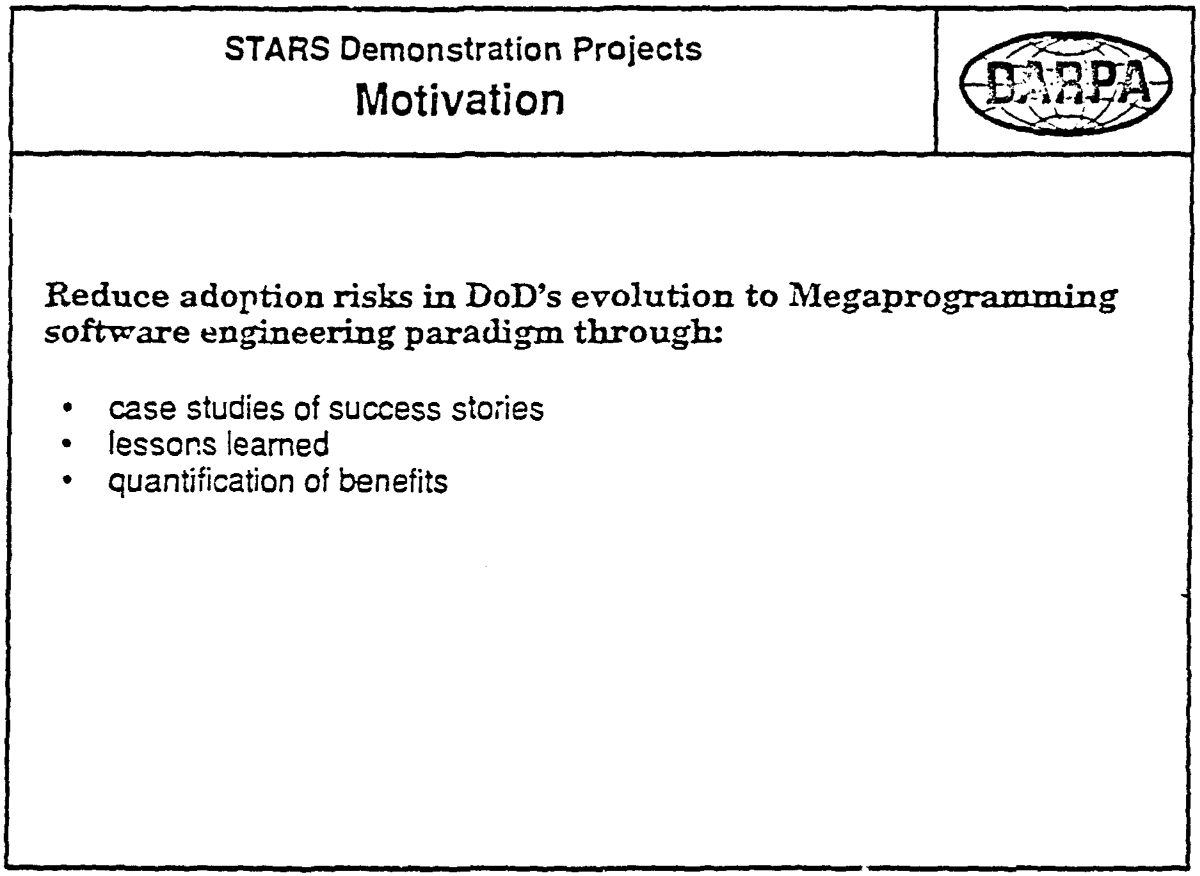
Collaborative Development

software engineering paradigm on DoD systems

Demonstration Projects Barnes/VG3

As you have seen from earlier presentations, STARS' vision is that megaprogramming is an emerging new software engineering paradigm and that the STARS' mission is to accelerate the adoption of megaprogramming concepts and technologies within the DOD community. The strategy STARS has adopted to achieve this mission is threefold: ensure the basic technologies and capabilities are available to support employing the megaprogramming paradigm, show that the megaprogramming paradigm can be used successfully on DOD systems, and to provide transition support to reduce the risks of adopting the megaprogramming paradigm.

The demonstration projects primarily support the second part of this strategy by applying the megaprogramming paradigm to some actual DOD systems.



STARS Demonstration Projects

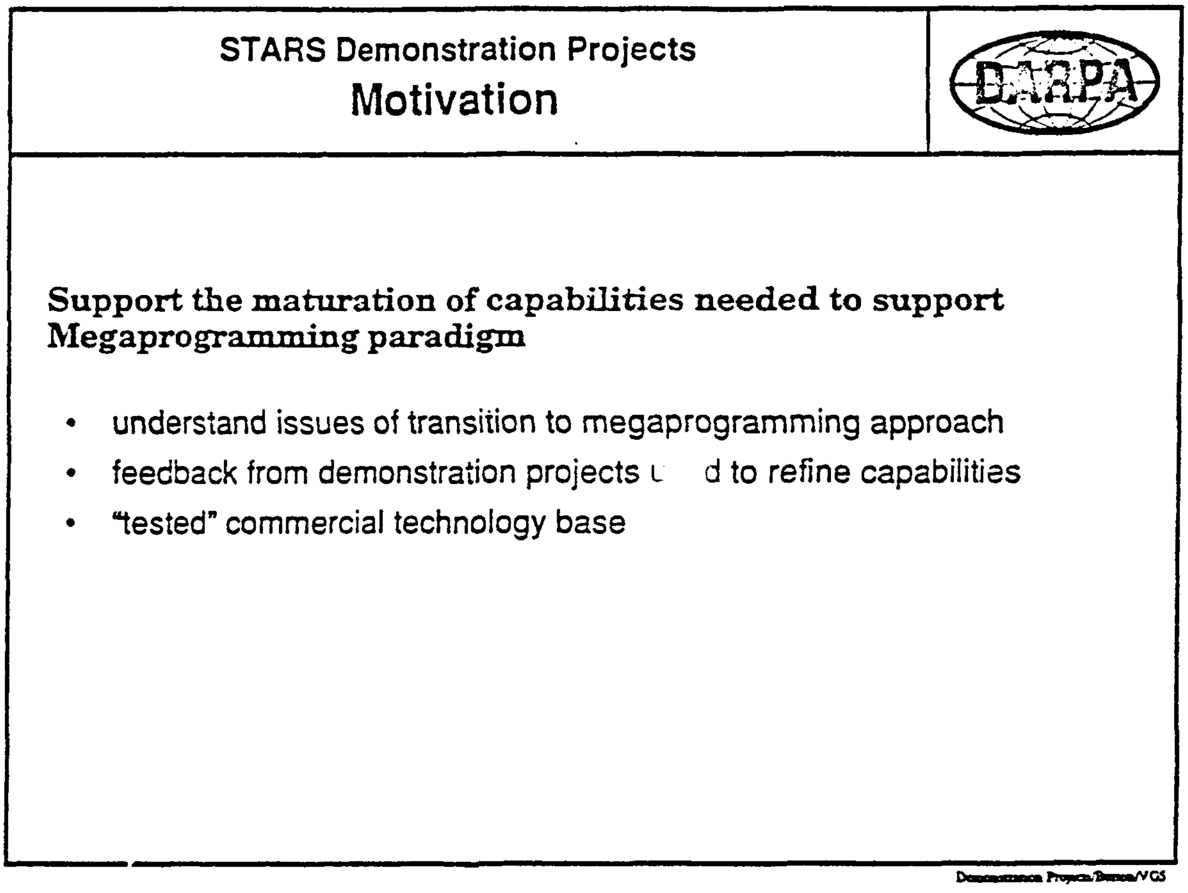
Motivation

Reduce adoption risks in DoD's evolution to Megaprogramming software engineering paradigm through:

- case studies of success stories
- lessons learned
- quantification of benefits

Demonstration Projects/[illegible]/VG4

The demonstration projects also support the third part of the strategy by providing the opportunity to document with case studies the application of the megaprogramming approach to some actual DOD systems, collect lessons learned in transitioning to and applying megaprogramming, and quantify some of the benefits of using the megaprogramming approach.



STARS Demonstration Projects

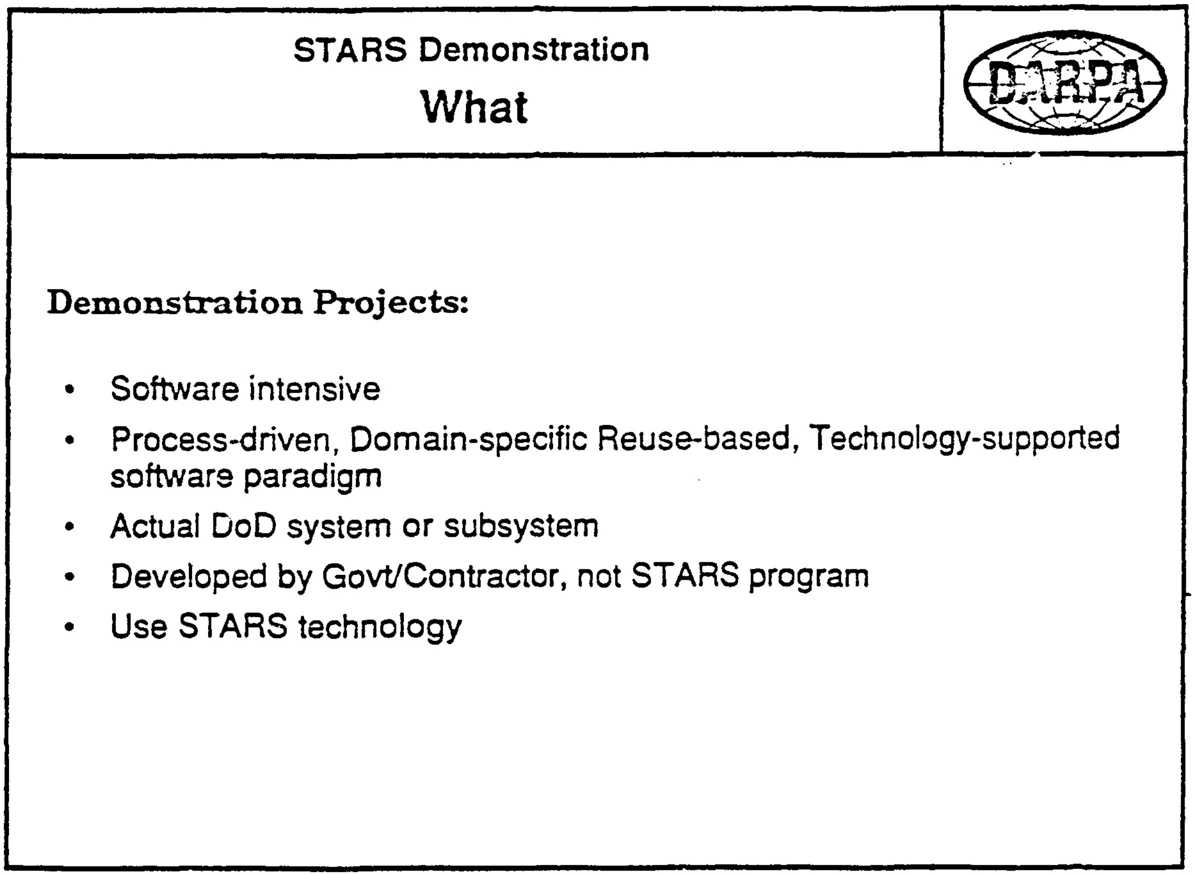
Motivation

Support the maturation of capabilities needed to support Megaprogramming paradigm

- understand issues of transition to megaprogramming approach
- feedback from demonstration projects used to refine capabilities
- "tested" commercial technology base

Demonstration Projects/Boston/VGS

The demonstration projects also support the first part of the strategy by providing an opportunity to test many of the megaprogramming technologies in the context of a real DOD application. The feedback provided by the demonstration projects will help in refining these technologies and tools so that more mature capabilities will be available.



STARS Demonstration

What

Demonstration Projects:

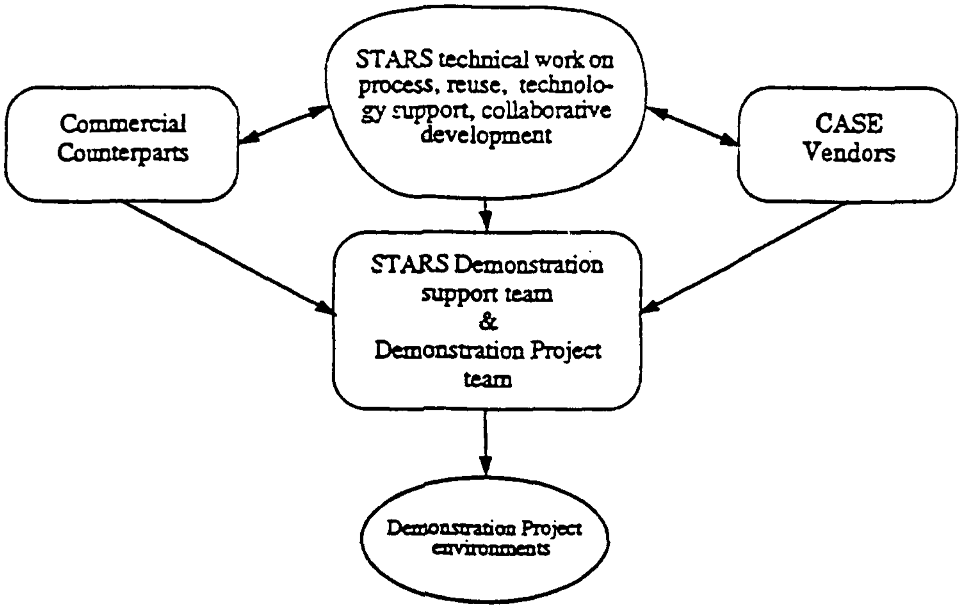
- Software intensive
- Process-driven, Domain-specific Reuse-based, Technology-supported software paradigm
- Actual DoD system or subsystem
- Developed by Govt/Contractor, not STARS program
- Use STARS technology

[illegible] Projects/[illegible]/VG6

Our goal is to demonstrate the value of the megaprogramming paradigm on actual DOD systems. In order to have a credible demonstration and to understand the transition issues, the actual developer needs to be separate from the STARS program. It could be an in-house government organization or DOD contractor. Over the last couple of years, the STARS program has developed technologies and capabilities to specifically support the megaprogramming paradigm. Most of these capabilities will be embodied in a Software Engineering Environment (SEE) that each STARS Prime contractor is assembling. These SEEs and other STARS developed technologies will be used in the demonstration projects.

STARS Demonstration

What



STARS technical work on process, reuse, technology support, collaborative development

Commercial Counterparts

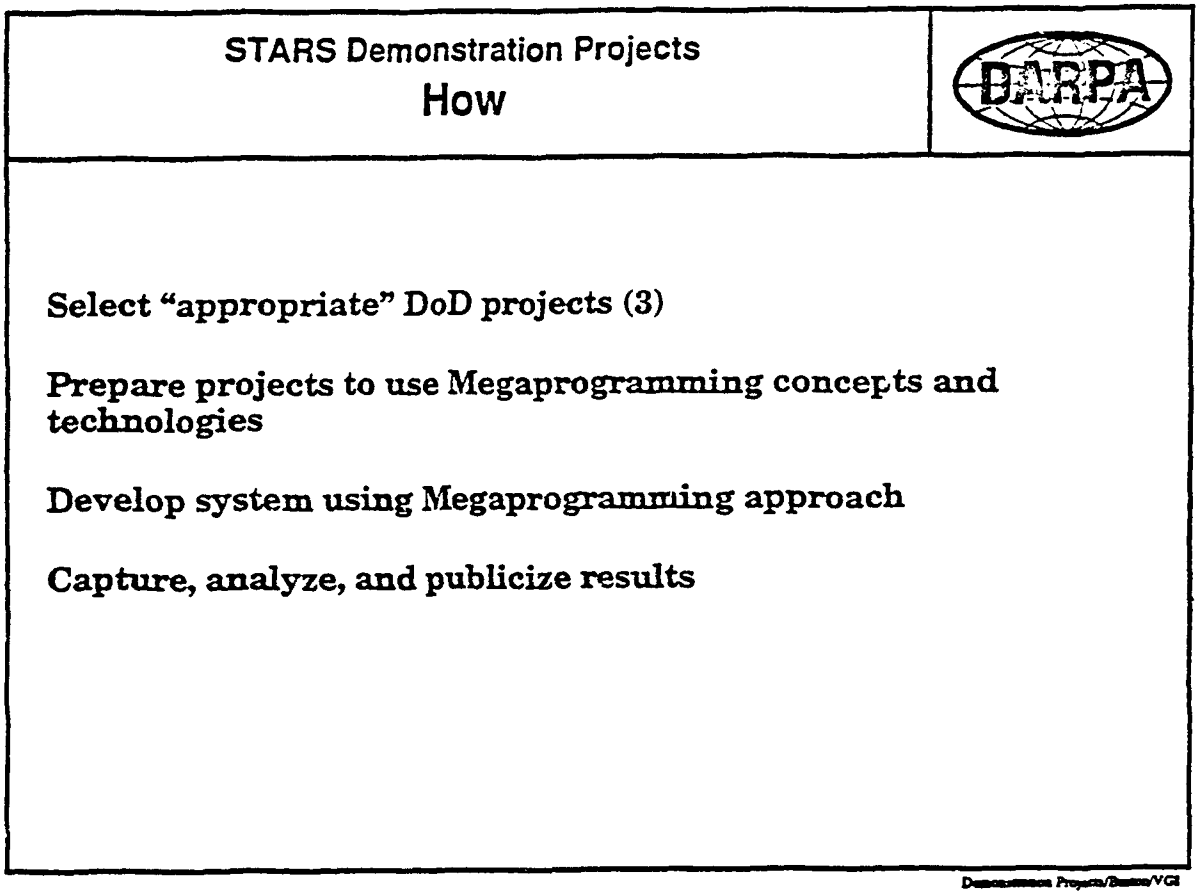
CASE Vendors

STARS Demonstration support team & Demonstration Project team

Demonstration Project environments

Demonstration Projects/Burton/VG7

This chart shows some of the relationships involved in putting together the SEE that will be used in the demonstration projects. Each STARS Prime contractor has a commerical counterpart that will supply the basic underlying framework for the SEE. Most of the tools integrated with the SEE will be provided by commercial CASE tool vendors. STARS will provide some special purpose tools and integrate all the pieces to provide and integrated SEE for each demonstration project to use.



STARS Demonstration Projects

How

Select "appropriate" DoD projects (3)

Prepare projects to use Megaprogramming concepts and technologies

Develop system using Megaprogramming approach

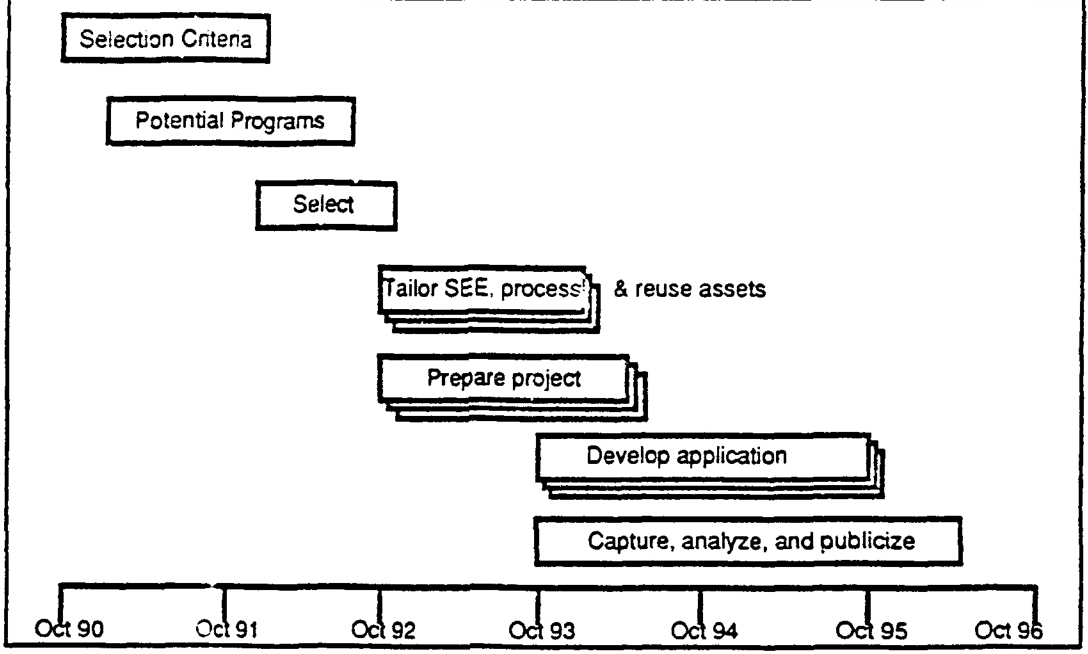
Capture, analyze, and publicize results

Demonstration Projects/Boston/VGI

The overall process for doing the demonstration projects is outlined here: selecting appropriate projects, preparing each project for the demonstration, developing the demonstration project application, and capturing the results.The next chart shows a timeline for these activiries and the steps are described in the following slides.

STARS Demonstration Projects

TIME LINE



Selection Criteria

Potential Programs

Select

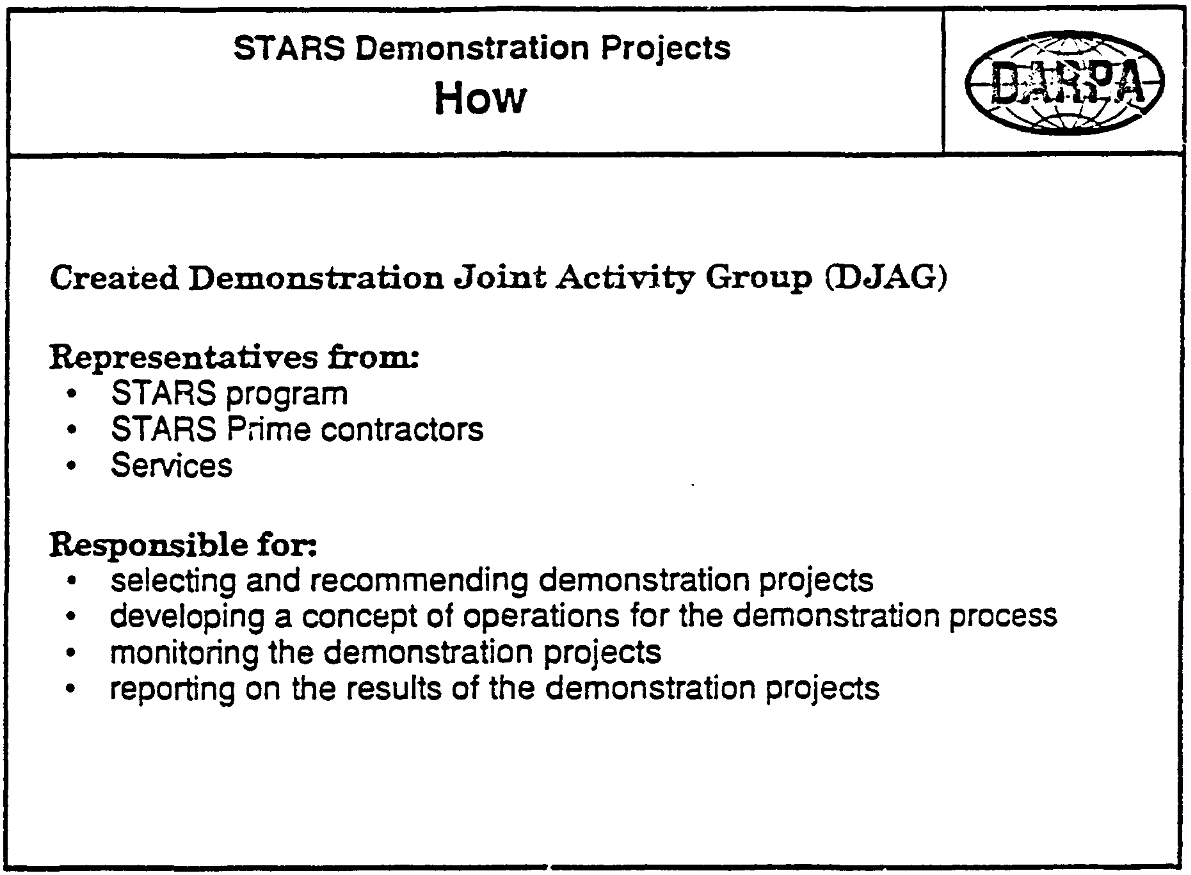
Tailor SEE, process & reuse assets

Prepare project

Develop application

Capture, analyze, and publicize

Demonstration Projects/[illegible]/VG9



STARS Demonstration Projects
How

Created Demonstration Joint Activity Group (DJAG)

Representatives from:
- STARS program
- STARS Prime contractors
- Services

Responsible for:
- selecting and recommending demonstration projects
- developing a concept of operations for the demonstration process
- monitoring the demonstration projects
- reporting on the results of the demonstration projects

Demonstration Projects/Barton/VG10

In order to plan, select, and oversee the demonstration projects, a working group (DJAG) with representatives from the STARS program, STARS Prime contractors, and the Services was formed. The DJAG is responsible for doing the project selection activities, doing the high level program planning for the demonstration projects, monitoring the demonstration projects as they are conducted, and reporting on the results of demonstration projects.

STARS Demonstration Projects
How

"Ideal" projects:

- size sufficient to be valid demonstration
- schedule compatible with STARS
- organization/management receptive to change
- willing and able to utilize new technology and processes
- domain appropriate for architecture-based reuse
- reuse assets exist or can be obtained
- organization has software process awareness
- security classification won't get in way
- Ada

Demonstration Projects/Burton/VG11

We are currently in the process of identifying candidate demonstration projects. These are some of the traits of the "ideal" demonstration project.

A project should be large enough that programming-in-the-large is demonstrated. We are looking for projects that would require in the range of 16 to 20 software engineers over a 24 month period.

The schedule for the project must be compatible with ours. The developing organization must be able to interact with STARS beginning early in FY93 to get prepared for applying the megaprogramming paradigm to the project. The demonstration project development should begin early in FY94 and complete by early FY96.

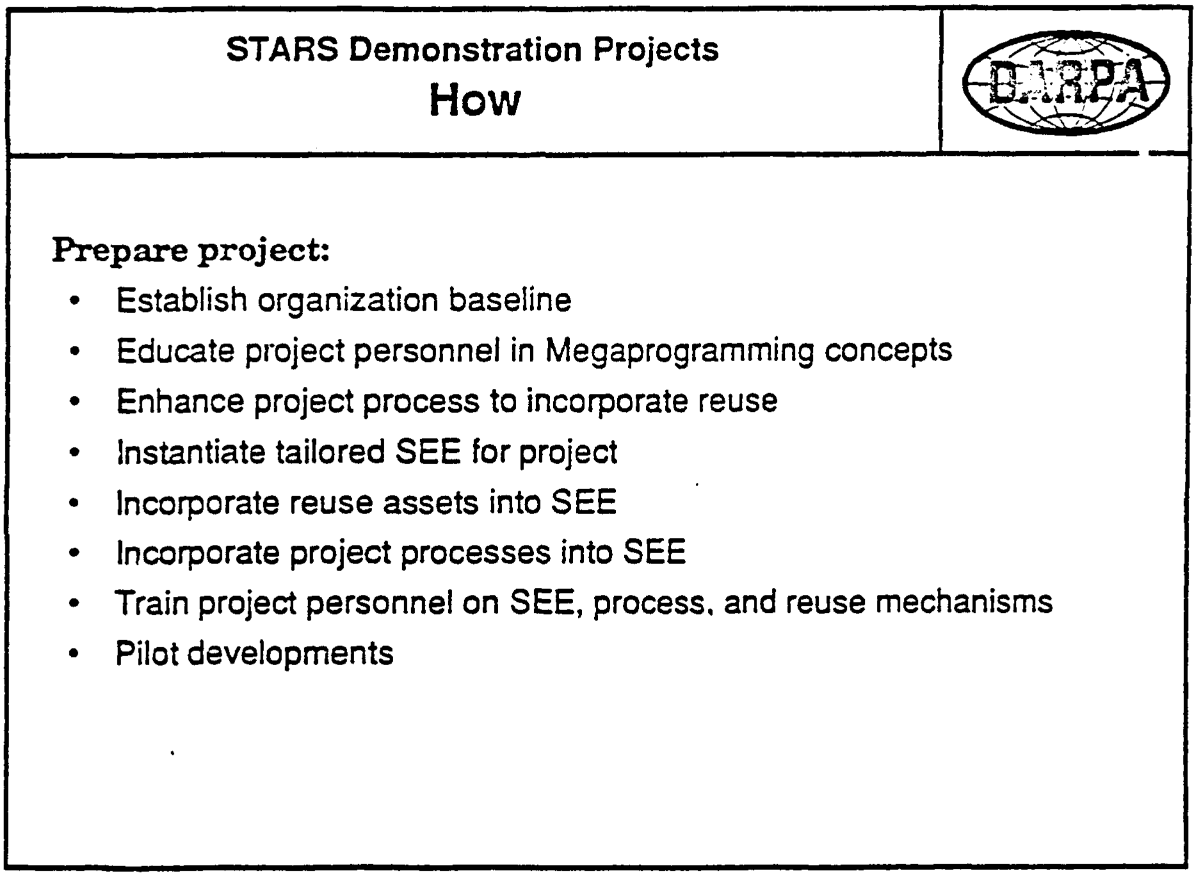
Adopting the megaprogramming paradigm will require a cultural change in the development organization that can only effectively take place if that organization fully supports the change. Included in this change is the adoption of the SEE supplied by the STARS Prime associated with the demonstration project.

Domain-specific reuse is a key element of the megaprogramming paradigm, so the demonstration project must be in a domain appropriate for domain-specific reuse and a set of domain-specific reusable assets should be available to use in the demonstration project.

Another key component of the megaprogramming paradigm is software process. The develpment organization should have a process awareness and a documented software process.

One of the major goals of these demonstration projects is to provide success stories of adopting and using the megaprogramming paradigm on actual DOD applications. Therefore, we want to work with projects that can allow the results to be made available openly and that will be able to talk about their experiences openly.

The project should be an Ada project.



STARS Demonstration Projects

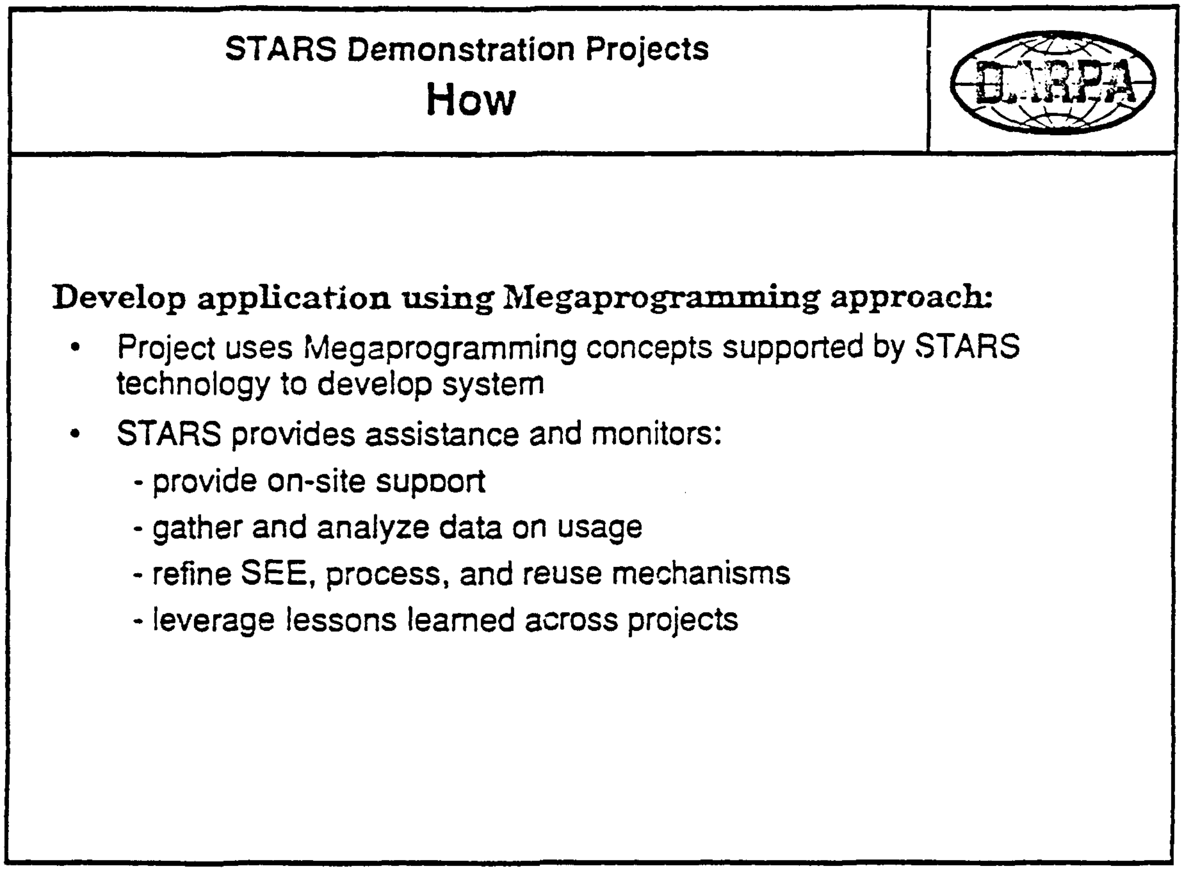
How

Prepare project:

- Establish organization baseline
- Educate project personnel in Megaprogramming concepts
- Enhance project process to incorporate reuse
- Instantiate tailored SEE for project
- Incorporate reuse assets into SEE
- Incorporate project processes into SEE
- Train project personnel on SEE, process, and reuse mechanisms
- Pilot developments

Demonstration Projects/[illegible]/VG12

When a "promising" project has been identified, we will then see if we can form a partnership with that project to "craft" a demonstration project that meets both STARS' and the project's objectives. Once a partnership has been formed, STARS will then begin preparing the project to adopt the megaprogramming paradigm. The initial steps of this phase will be to educate the development organization personnel in megaprogramming concepts and to baseline the development organization in terms of process maturity, productivity, and quality. After understanding the development organization's software process and supporting tools, STARS will work with the development organization to incorporate reuse into their software process and to determine the tool suite needed in their SEE. The SEE will then be assembled, the modified software processes will installed in the SEE, the reuse assets incorporated in the SEE, and integration testing performed on the SEE. Development organization personnel will be trained in the use of the SEE, process, and reuse mechanisms. The develpment organization may then do some small pilot projects to refine their software process before starting the demonstration project.



STARS Demonstration Projects

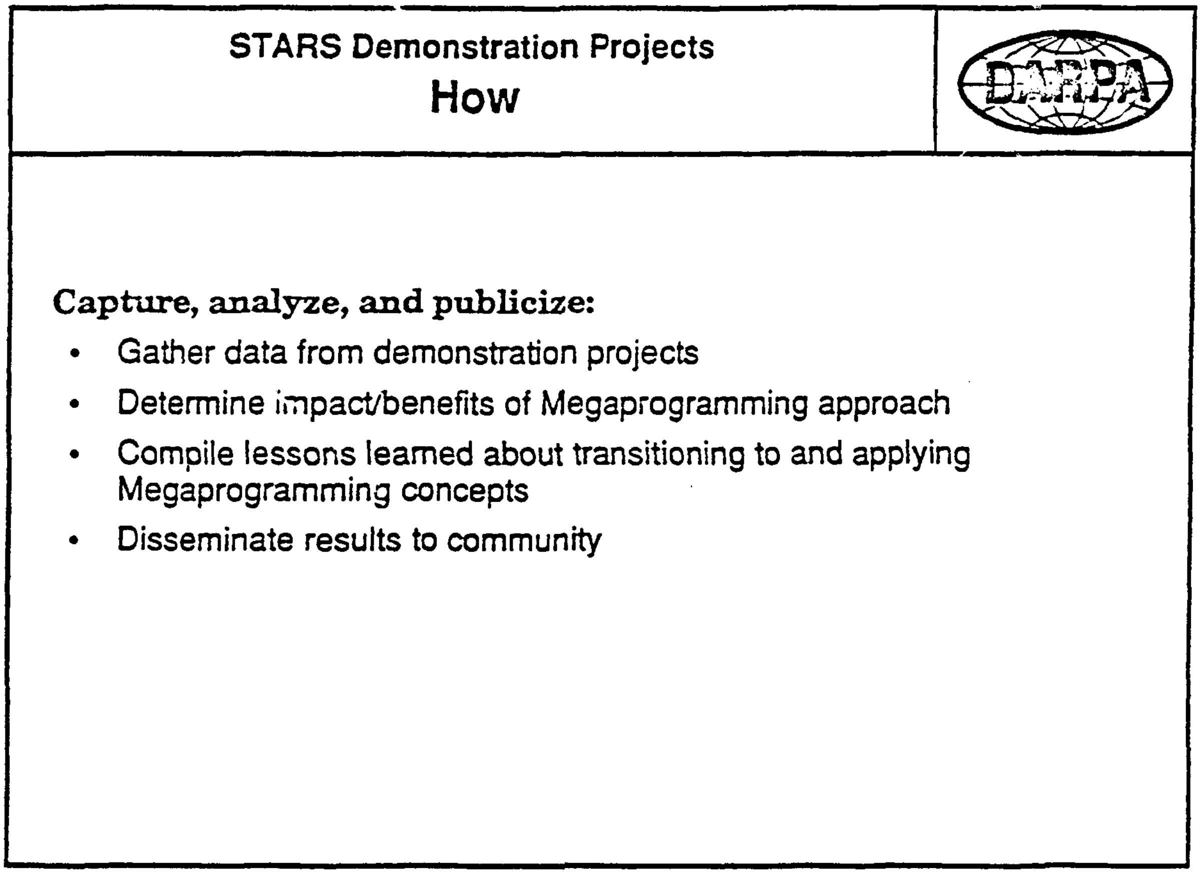
How

Develop application using Megaprogramming approach:

- Project uses Megaprogramming concepts supported by STARS technology to develop system
- STARS provides assistance and monitors:
  - provide on-site support
  - gather and analyze data on usage
  - refine SEE, process, and reuse mechanisms
  - leverage lessons learned across projects

Demonstration Projects/[illegible]/VG13

As the development organization works on the demonstraton project, the STARS Prime associated with the demonstration project will provide on-site support to the development organization and monitor how the project's usage of the megaprogramming paradigm is going. The STARS Prime may assist the development organization to refine their processes and may tune the SEE to better support the demonstration project. As things are learned in one demonstration project that may benefit others, they will be spread to the other projects.



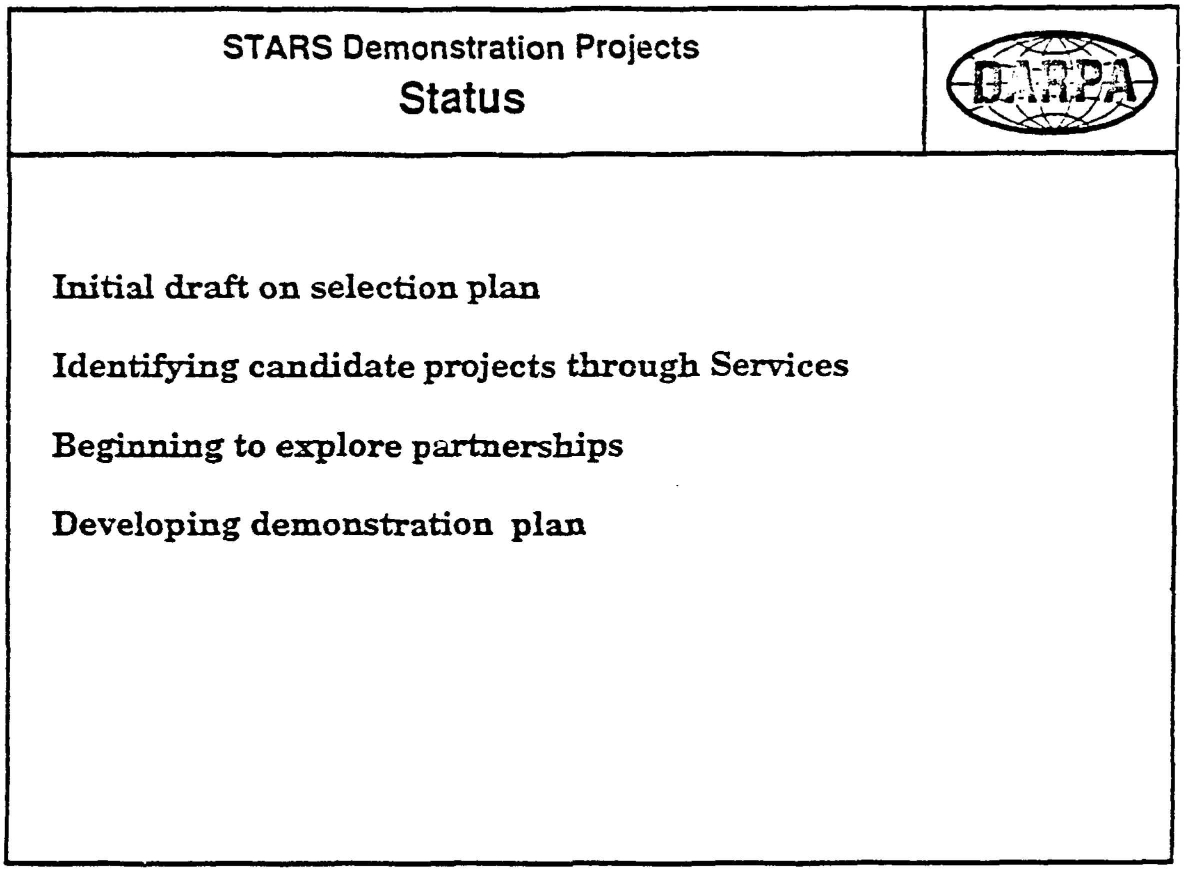
STARS Demonstration Projects
How

Capture, analyze, and publicize:

- Gather data from demonstration projects
- Determine impact/benefits of Megaprogramming approach
- Compile lessons learned about transitioning to and applying Megaprogramming concepts
- Disseminate results to community

Demonstration Projects/Demos/VG14

Before, during, and afterwards data will be collected from the development organization for the demonstration project. During the demonstration project this data will be used to determine how well the megaprogramming paradigm is working and to make mid-course corrections if necessary. At the completion of the demonstration projects, the information will be compiled into lessons learned about transitioning to and applying the megaprogramming paradigm, and to determine the impact of using the megaprogramming paradigm. These will be put into reports that will be made available to the DOD community.



STARS Demonstration Projects

Status

Initial draft on selection plan

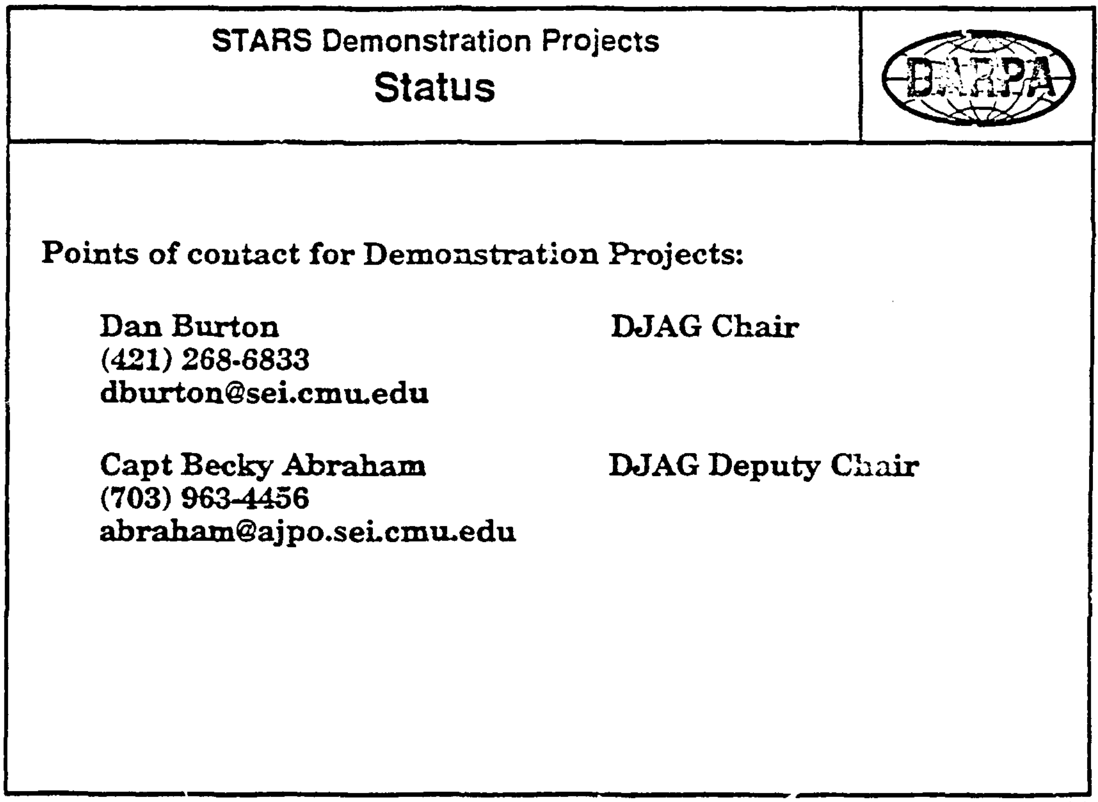
Identifying candidate projects through Services

Beginning to explore partnerships

Developing demonstration plan

Demonstration Projects/Status/VG15

The DJAG is currently concentrating on project selection. We have initiated activities with each of the Services to identify good candidate projects and expect to finish this activity by the end of this fiscal year with agreements for three demonstration projects. In parallel to this, we are developing an overall plan for the demonstration process that will define the demonstration process, identify the data to be collected during the demonstration projects, and describe how the data will be analyzed and reported.



STARS Demonstration Projects

Status

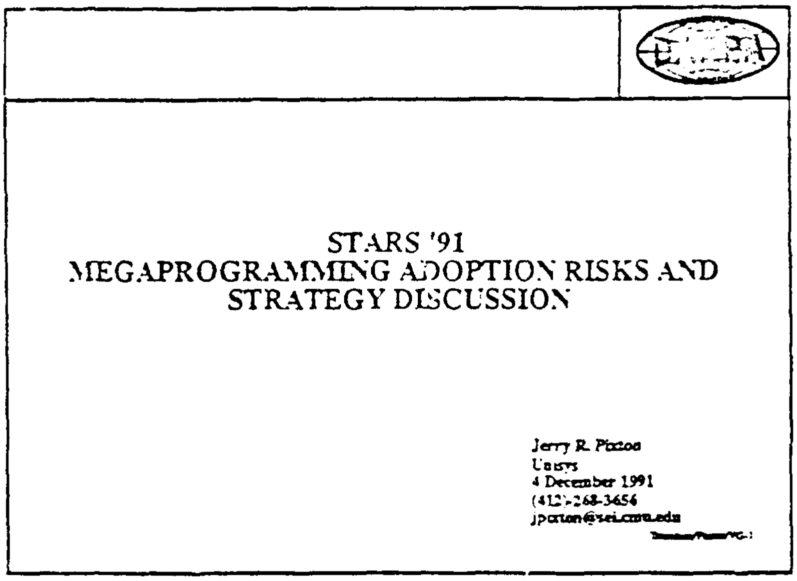
DARPA

Points of contact for Demonstration Projects:

Dan Burton DJAG Chair
(421) 268-6833
dburton@sei.cmu.edu

Capt Becky Abraham DJAG Deputy Chair
(703) 963-4456
abraham@ajpo.sei.cmu.edu

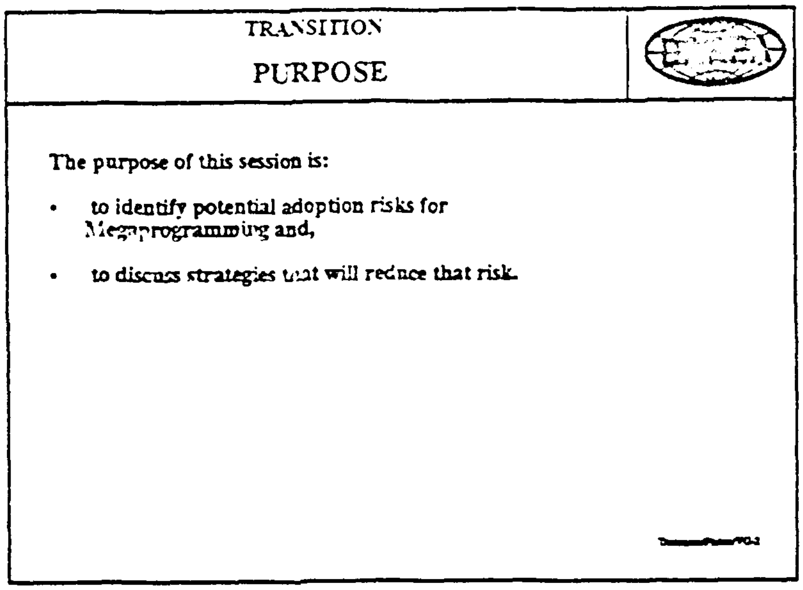
Demonstration Projects/Burton/VG16



STARS '91
MEGAPROGRAMMING ADOPTION RISKS AND
STRATEGY DISCUSSION

Jerry R. Pixton
Unisys
4 December 1991
(412)-268-3656
jpixton@sei.cmu.edu

Good morning, I am Dr Jerry Pixton from the Unisys STARS Program. I am currently a Resident Affiliate at the Software Engineering Institute working on software process acquisition.



TRANSITION

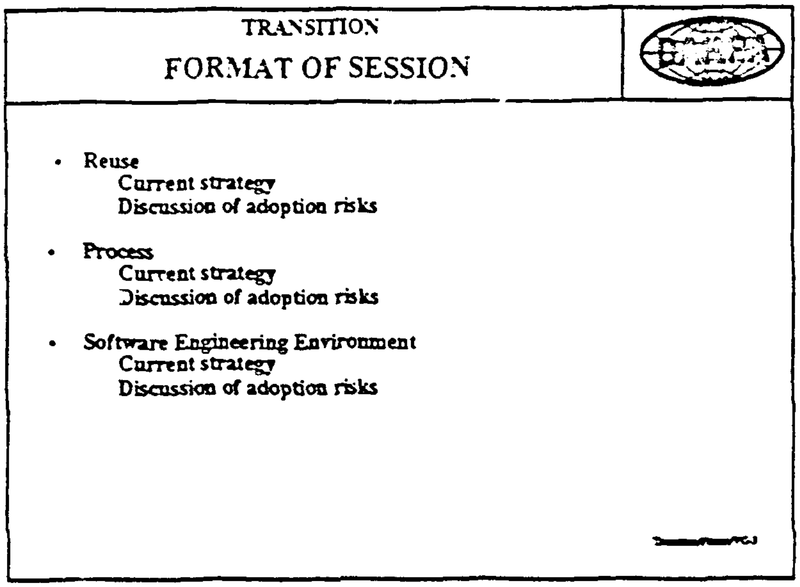
PURPOSE

The purpose of this session is:

- to identify potential adoption risks for Megaprogramming and,
- to discuss strategies that will reduce that risk.

This session is designed to enlist your help in looking for potential barriers to the adoption of the Megaprogramming paradigm.

And to start some discussion of possible corrective methods that might be used for risk reduction.

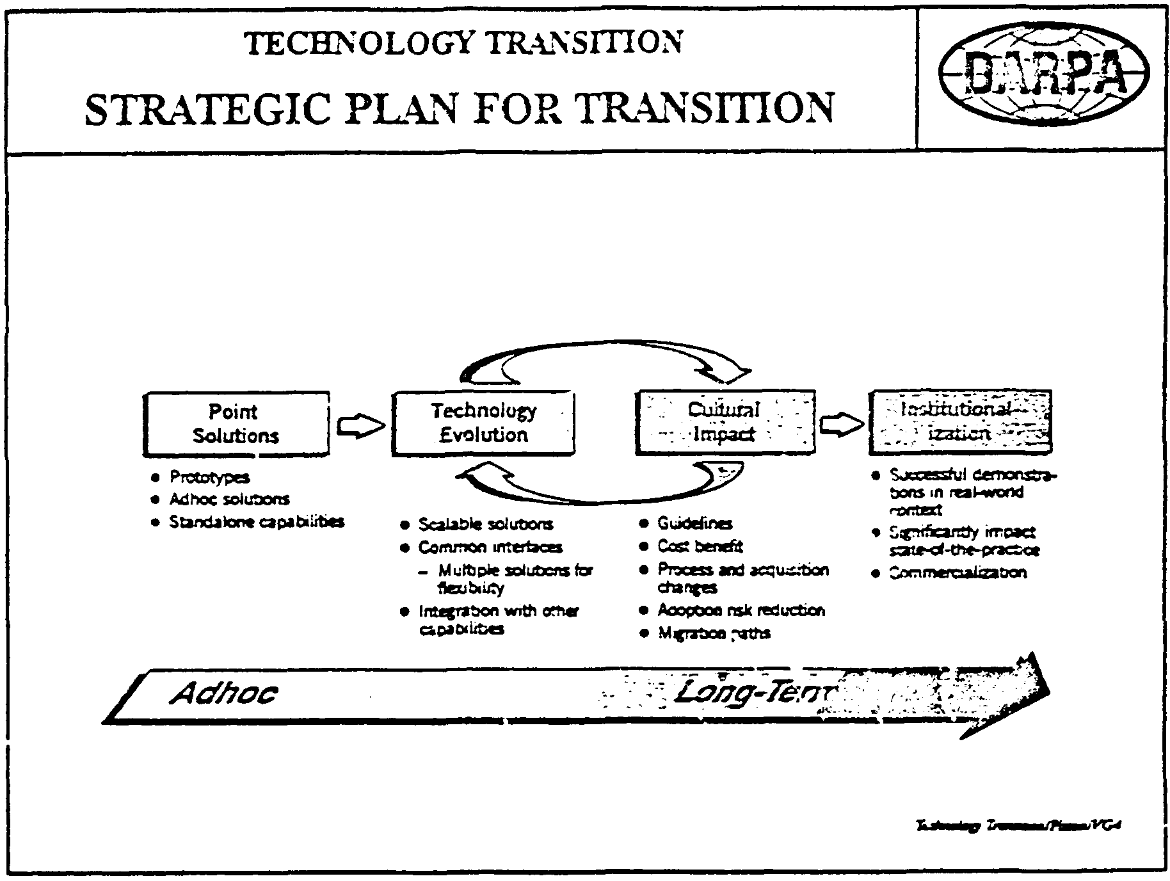


TRANSITION

FORMAT OF SESSION

- Reuse
  Current strategy
  Discussion of adoption risks
- Process
  Current strategy
  Discussion of adoption risks
- Software Engineering Environment
  Current strategy
  Discussion of adoption risks

The format to be used in this session is slightly different than other sessions. What I am going to do is break our 45 minutes (of briefing and questions) into 3 segments. Each segment (reuse, process and environment) will start with a brief introduction of the current approach that STARS is following in this particular technical transfer area. Then, I will moderate an 8 minute discussion period on other risks that the audience thinks are important and solicit some methods to reduce that risk. The discussion will be recorded in outline form as we go along. We will repeat this for each of the 3 segments. Following that, there will be a few minutes to discuss any issues that did not fit into these 3 areas.



TECHNOLOGY TRANSITION

STRATEGIC PLAN FOR TRANSITION

Point Solutions

- Prototypes
- Adhoc solutions
- Standalone capabilities

Technology Evolution

- Scalable solutions
- Common interfaces
  - Multiple solutions for flexibility
- Integration with other capabilities

Cultural Impact

- Guidelines
- Cost benefit
- Process and acquisition changes
- Adoption risk reduction
- Migration paths

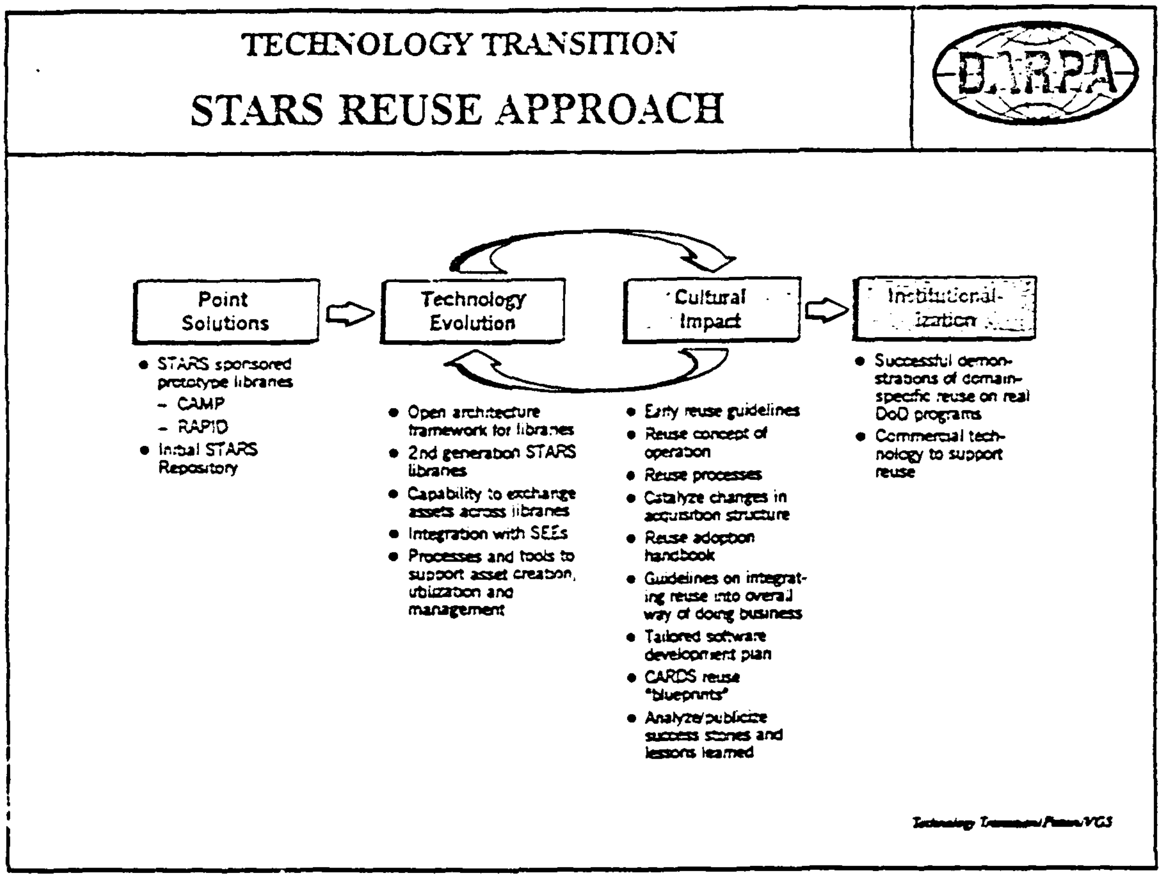
Institutional-ization

- Successful demonstrations in real-world context
- Significantly impact state-of-the-practice
- Commercialization

Adhoc Long-Term

Technology Transition/Plans/VG4

You will see this template, of the four transition steps that STARS is usi ; to introduce new ideas into an organization's software development environment, on each of the 3 approaches that I explain. The template represents stages of maturity for the introduction of new methods into an organization. On the left, the new ideas are being prototyped in an ad hoc, limited manner. As you decide that this is a good method that deserves wider use, you start to mature the method by evolving the supporting technology and by managing the organizational change caused by the introduction of the new method. This is an iterative cycle, of technology evolution and cultural change, which may take several rounds before the method becomes widely accepted by the organization. Once this acceptance occurs, the method becomes institutionalized with a broad base of usage.



TECHNOLOGY TRANSITION

STARS REUSE APPROACH

Point Solutions

- STARS sponsored prototype libranes
  - CAMP
  - RAPID
- Initial STARS Repository

Technology Evolution

- Open architecture framework for libranes
- 2nd generation STARS libranes
- Capability to exchange assets across libranes
- Integration with SEEs
- Processes and tools to support asset creation, utilization and management

Cultural Impact

- Early reuse guidelines
- Reuse concept of operation
- Reuse processes
- Catalyze changes in acquisition structure
- Reuse adoption handbook
- Guidelines on integrating reuse into overall way of doing business
- Tailored software development plan
- CARDS reuse "blueprints"
- Analyze/publicize success stories and lessons learned

Institutionalization

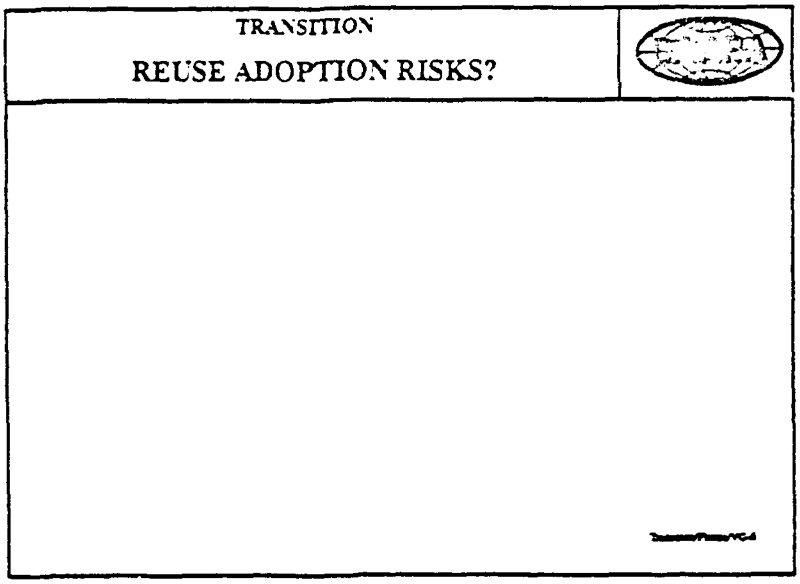
- Successful demonstrations of domain-specific reuse on real DoD programs
- Commercial technology to support reuse

Technology Transition/Pattern/VGS

For the STARS Reuse area, the approach to technology transfer that is being taken consists of starting with prototype reuse librar:es, such as CAMP and RAPID, and the initial STARS repository technology. This is being evolved into 2nd generation reuse libraries build around an open architecture framework so that organizations can implement libraries on multiple platforms. These libraries will be integrated into their environments. Processes are being developed to help organizations manage reusable assets.

The cultural impact is being reduced by the early introduction of the concept of software component reuse with guidelines on how a company can develop reusat:le assets as a business strategy. Changes to acquisition policy are necessary to encourage this trend. Blueprints for reuse libraries are being developed, e.g., by CARDS, and success stories, of the benefits to projects of doing reuse, are being publicized.

The STARS effort will measure the benefits of reusable assets as a business strategy during the 3 demonstration projects, starting in October 1993. This will be followed by commercialization of this technology for wide spread use.



TRANSITION

REUSE ADOPTION RISKS?

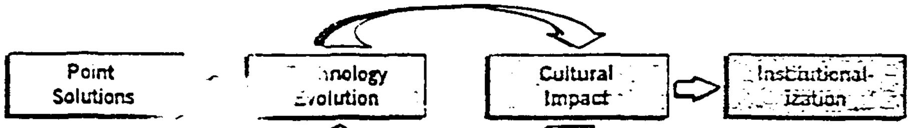
Now we have about 8 minutes for discussion of some areas that you feel are important adoption risks for us to consider. I will moderate the discussion.

Some potential risks are included here, as samples, to give you some ideas for discussion:

- Acquisition policy (benefit from reusing, shift to manage families of systems)
- Investments in application domain architectures and components
- Change Management
- Dichotomy of community consensus on architecture vs company competitive advantage
- Lack of reuse base and practical experience
- Lack of market size

TECHNOLOGY TRANSITION

STARS PROCESS APPROACH



- Concept prototypes
- Identify and document experience-tested software processes
- Process modeling experiments
- Process enaction experiments

- Library of fundamental process assets
- Capability to compose a software process from reusable process assets
- Capability to guide and support the performance and measurement of a defined process

- Guidelines and techniques for process definition and measurement
- Guidelines for tailoring process to specific development standards and acquisition policy
- Identify and publicize cost benefit models for process-driven development

- Successful demonstrations of process definition and management on DoD projects
- Commercial technology for process definition
- Integrated SEE support for process management

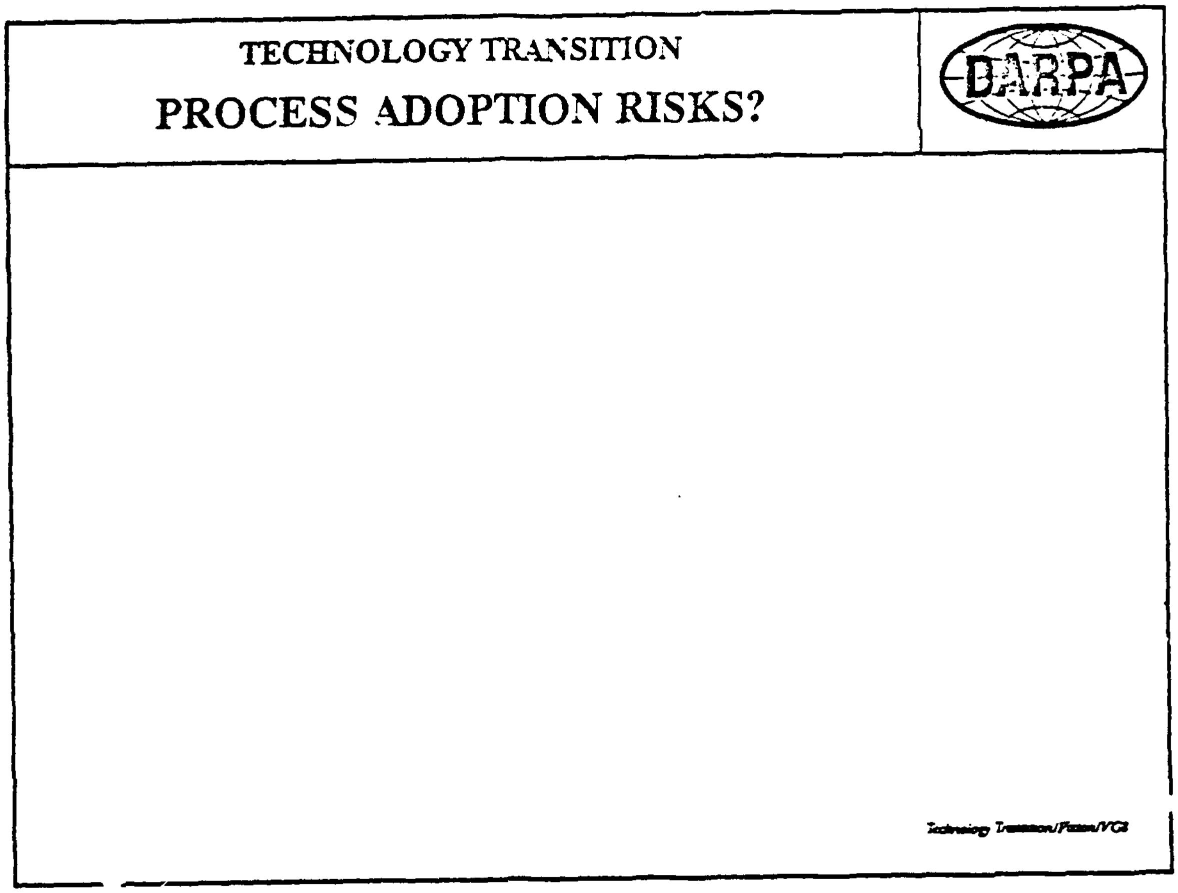
Technology Transition/Process/VG7

The technology transfer approach for process starts with the work that the Software Engineering Institute and others have done to identify and document experience-tested software processes. Also, various process modeling and process measurement experiments have been conducted, such as the Arcadia project at the University of Southern California.

This base is being evolved by STARS to construct a Process Asset Library of experience-tested software processes. The capability will be present to support measurement and performance analysis of process components. This will provide support for early attempts at Statistical Process Control of software processes. The Process Asset Library will eventually have the capability to compose new software processes from reusable process assets.

Since this is a relative new subject for industry (the Software Engineering Process Group is the closest concept today), guidelines and techniques for process definition, tailoring, measurement and management will be developed to aid businesses in improving their software development processes.

The cost benefits of process-driven development will be determined during the demonstration projects.



TECHNOLOGY TRANSITION

PROCESS ADOPTION RISKS?

Technology Transition/[illegible]/VG3

Some potential risks are included here, as samples, to give you some ideas for discussion:

- How do process engineers appear, get trained and operate in organizations?

- How do we prevent process engineers from appearing as excess staff, like Configuration Management and Quality Assurance have been treated in past?

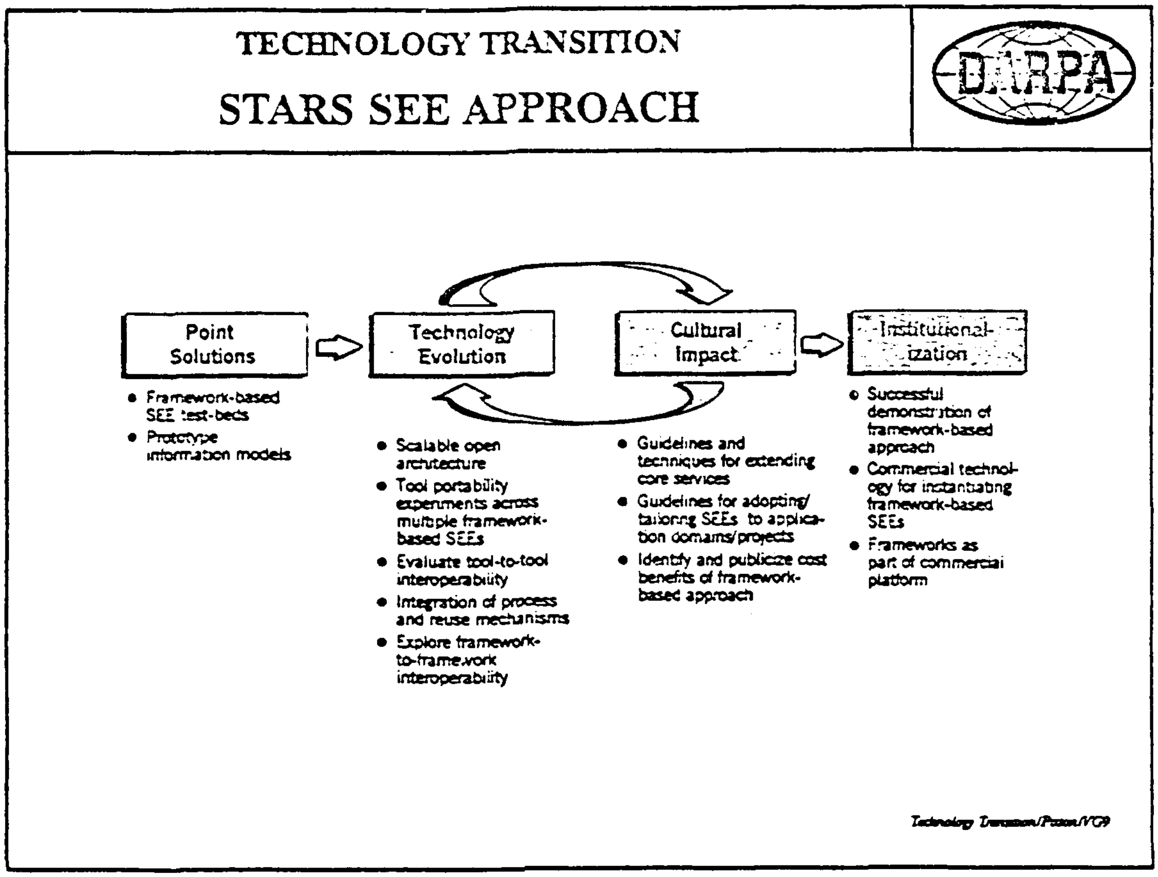
- DoD STD 2167A is document driven vice process driven. Will it get in the way, like it has for parallel development in "builds" or Ada?

- How maturity do organizations need to be before benefiting?

- Increased investment in training and education

- Lack of Policies (e.g., metrics...)

- Can we take the concept of process metrics (which industry has been struggling with for some time) and expand it into the next generation concept of process definition with automated measurements?



TECHNOLOGY TRANSITION

STARS SEE APPROACH

Point Solutions

- Framework-based SEE test-beds
- Prototype information models

Technology Evolution

- Scalable open architecture
- Tool portability experiments across multiple framework-based SEEs
- Evaluate tool-to-tool interoperability
- Integration of process and reuse mechanisms
- Explore framework-to-framework interoperability

Cultural Impact

- Guidelines and techniques for extending core services
- Guidelines for adopting/tailoring SEEs to application domains/projects
- Identify and publicize cost benefits of framework-based approach

Institutionalization

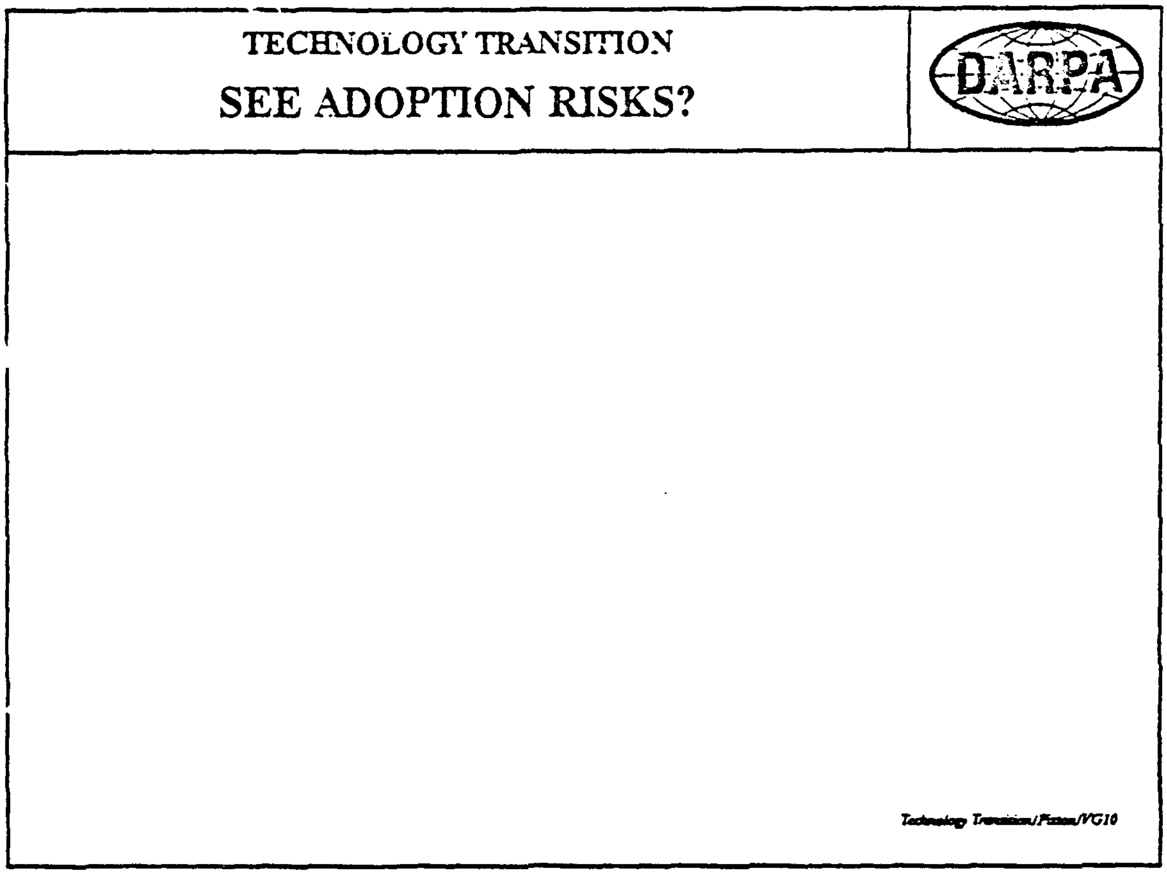
- Successful demonstration of framework-based approach
- Commercial technology for instantiating framework-based SEEs
- Frameworks as part of commercial platform

Technology Transition/Poston/VG9

Work on prototype frameworks and initial information models are being expanded to open architectures which will allow environments to be sized to the project and for tools to be moved from one environment to another. This is the delivery vehicle for the process and reuse technology. For the first time, process and reuse aspects will be integrated into the environment.

Guidelines will be produced so that businesses can adapt, and tailor, the Software Engineering Environment for their application domains and projects. Another important cultural impact area to be addressed is how to capture legacy processes, from your organization, into the new environment. Guidelines will be developed on successful ways to do this.

Commercial components will be used, by the Commercial Counterparts, to develop the environment. Successful demonstration of framework-based environments will aid in their commercial acceptance.



TECHNOLOGY TRANSITION

SEE ADOPTION RISKS?

DARPA

Technology Transition/Fixes/VG10

Some potential risks are included here to give you some ideas for discussion:

- How will legacy tools, which represent investments still on the books, be included in the new environment?

- The levels of investment in software tools has never been very high considering the complexity of the task. Other complex tasks, such as mechanical design (which have been automated using Computer Aided Design) and shop floor tooling (which have been automated using numerical control), have seen large investments because management perceives that the job can be done more cost effective. How can this perception be changed for software?

TECHNOLOGY TRANSITION

MEGAPROGRAMMING ADOPTION RISKS?

Technology Transition/Panel/VG11

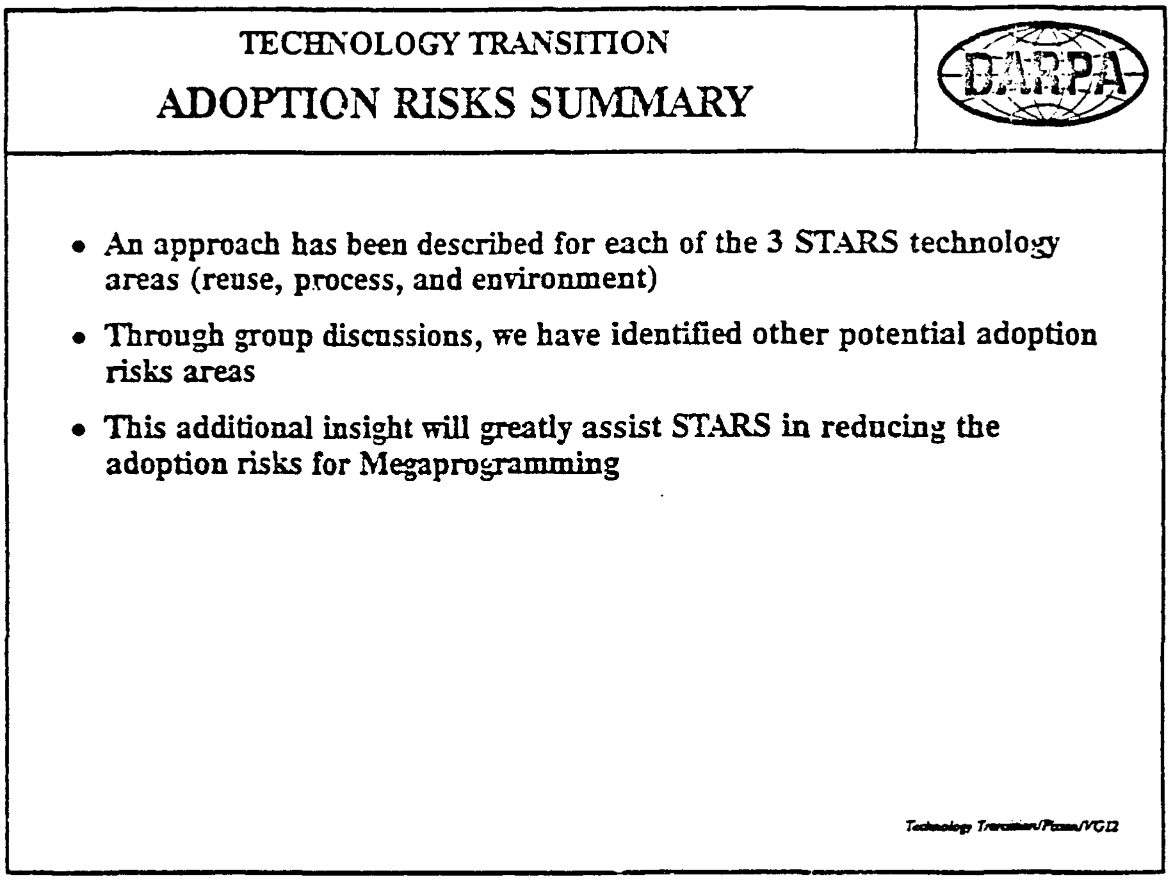
I realize that the three areas, of reuse, process and environment, that we have discussed might not have been sufficient for you to discuss some other aspects. Are there any other barriers or adoption risks associated with Megaprogramming that you want to mention?

For example:

- Are there other activities going on that could help, or accelerate, the adoption of Megaprogramming?

- Demonstration projects?

- Technology transfer?



TECHNOLOGY TRANSITION

ADOPTION RISKS SUMMARY

- An approach has been described for each of the 3 STARS technology areas (reuse, process, and environment)
- Through group discussions, we have identified other potential adoption risks areas
- This additional insight will greatly assist STARS in reducing the adoption risks for Megaprogramming

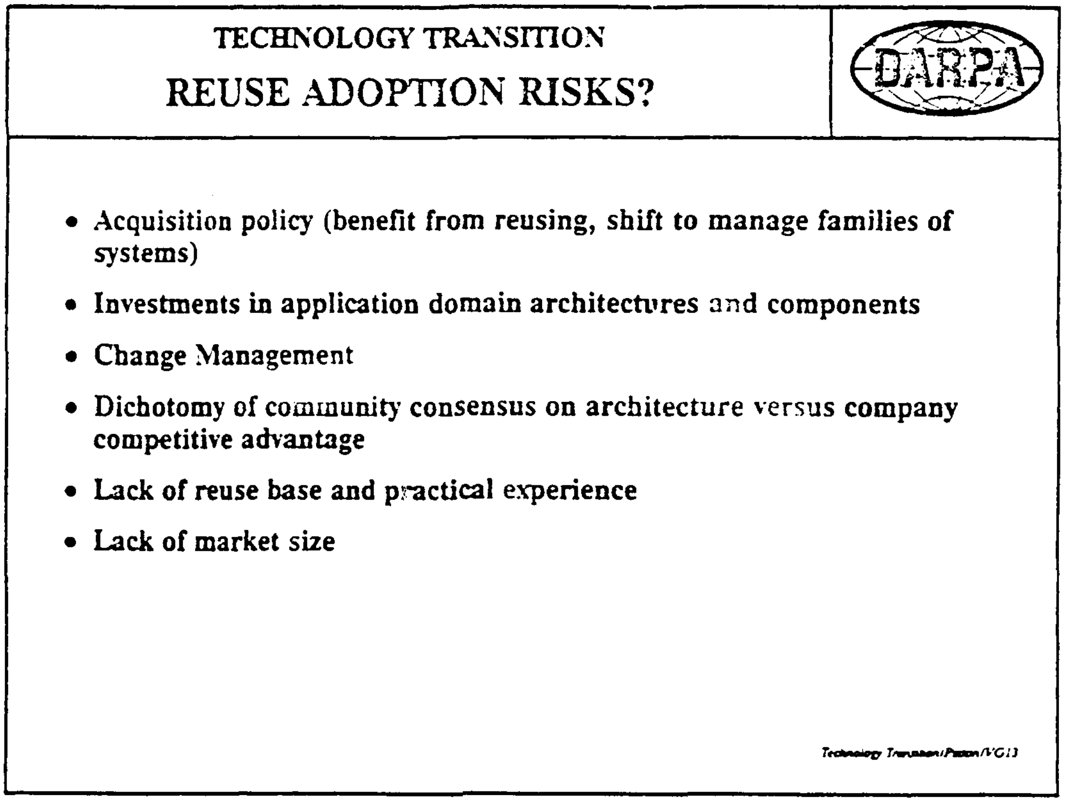
Technology Transition/Pixton/VG12

An approach to technology transfer has been described for each of the 3 STARS technology areas (reuse, process and environment). Through group discussions, we have identified other potential adoption risks areas. This additional insight will greatly assist STARS in reducing the adoption risks for Megaprogramming.

The discussion can continue during the breaks with all of us, or at a later time.

Thank you for your ideas. For further information, contact:

Dr. Jerry R. Pixton, 412-268-3656, jpixton@sei.cmu.edu

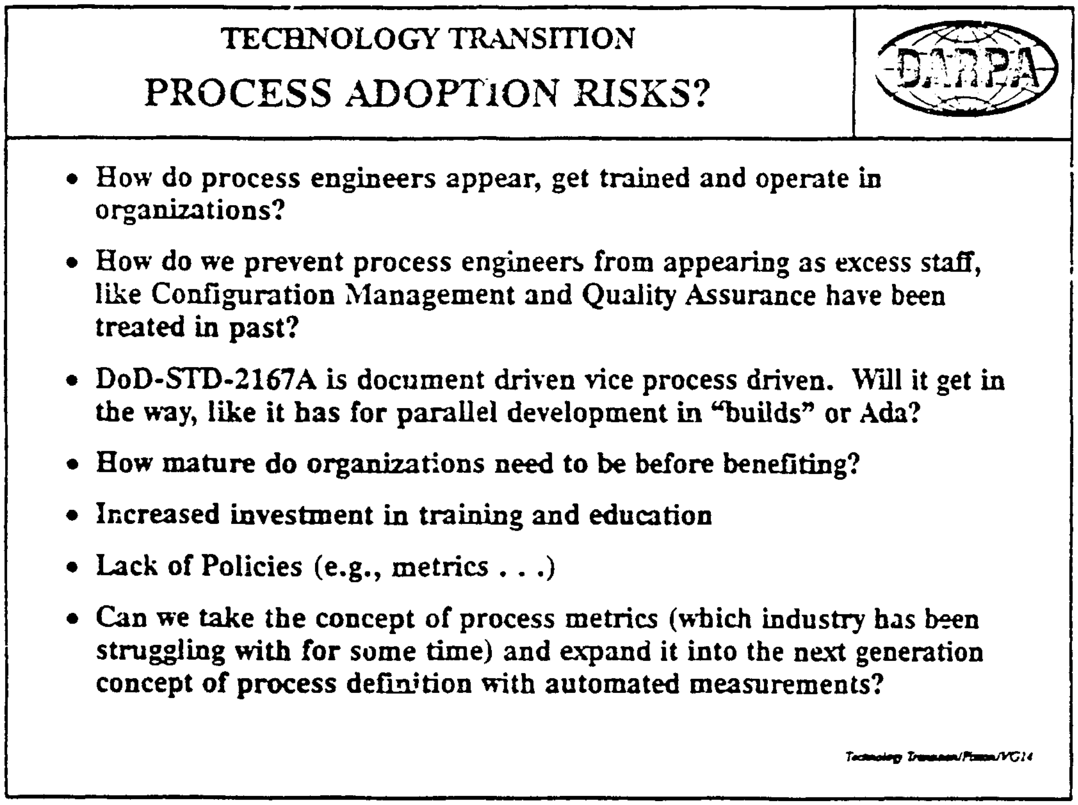


TECHNOLOGY TRANSITION

REUSE ADOPTION RISKS?

- Acquisition policy (benefit from reusing, shift to manage families of systems)
- Investments in application domain architectures and components
- Change Management
- Dichotomy of community consensus on architecture versus company competitive advantage
- Lack of reuse base and practical experience
- Lack of market size

Technology Transition/Paxton/VG13

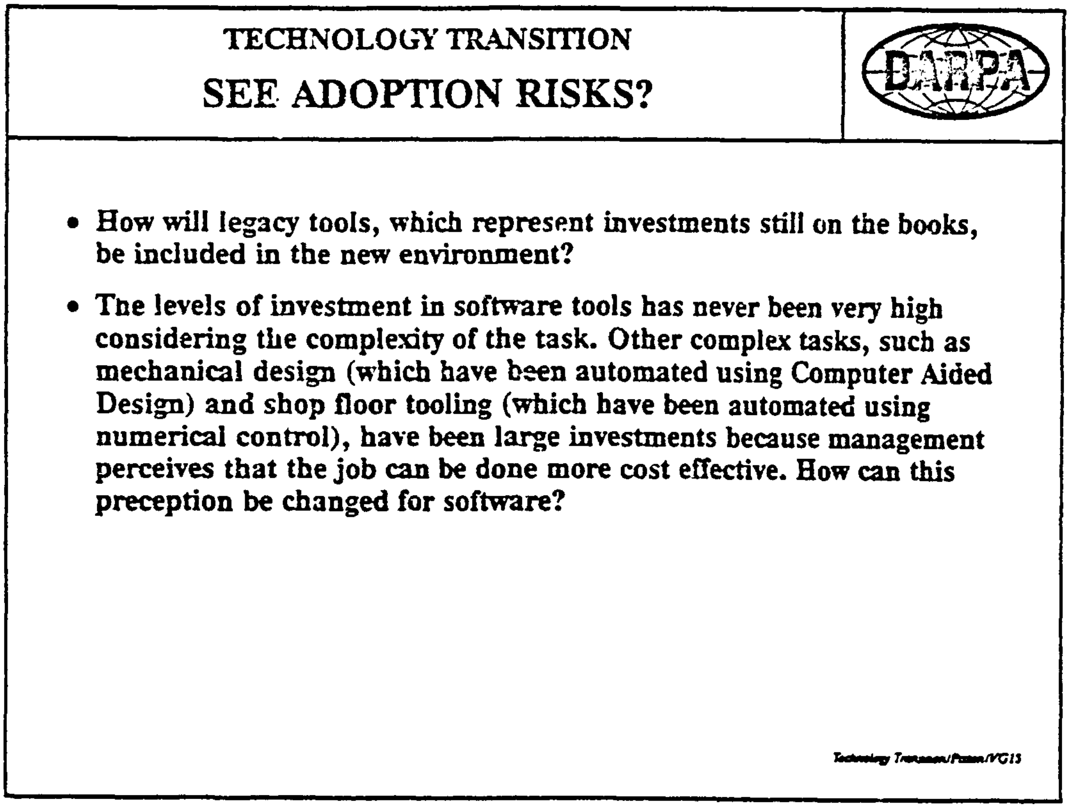


TECHNOLOGY TRANSITION

PROCESS ADOPTION RISKS?

- How do process engineers appear, get trained and operate in organizations?
- How do we prevent process engineers from appearing as excess staff, like Configuration Management and Quality Assurance have been treated in past?
- DoD-STD-2167A is document driven vice process driven. Will it get in the way, like it has for parallel development in "builds" or Ada?
- How mature do organizations need to be before benefiting?
- Increased investment in training and education
- Lack of Policies (e.g., metrics . . .)
- Can we take the concept of process metrics (which industry has been struggling with for some time) and expand it into the next generation concept of process definition with automated measurements?

Technology Transition/Process/VG14

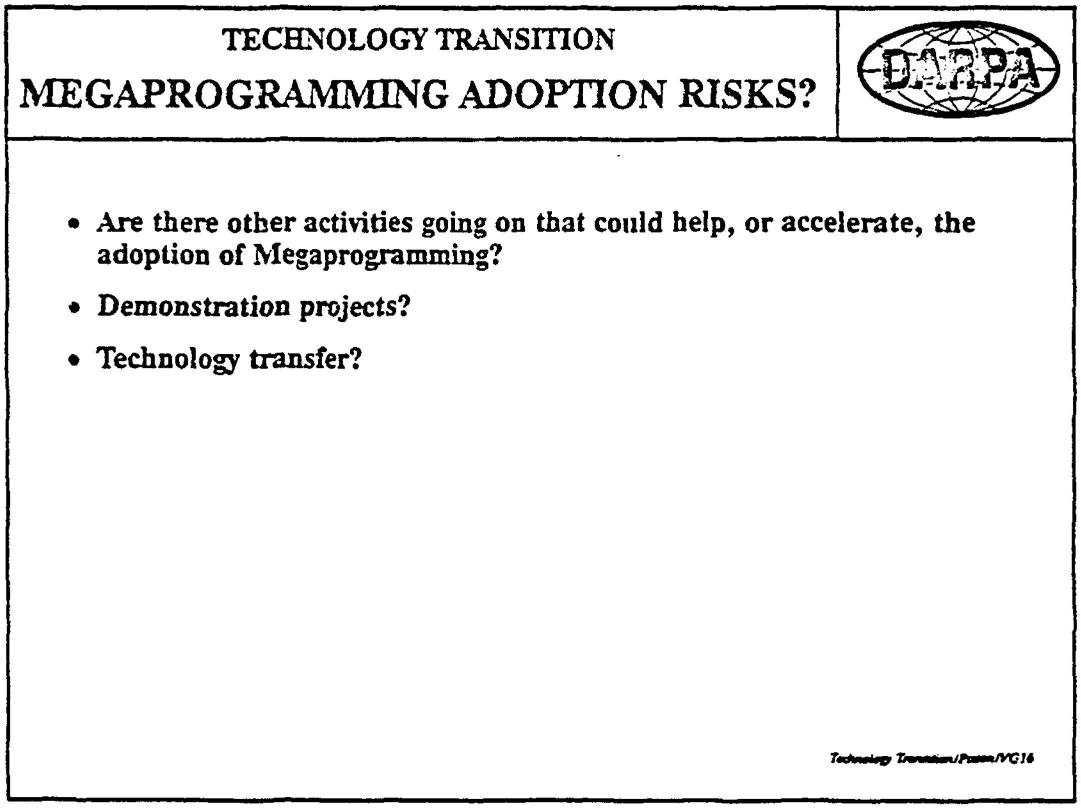


TECHNOLOGY TRANSITION

SEE ADOPTION RISKS?

- How will legacy tools, which represent investments still on the books, be included in the new environment?
- The levels of investment in software tools has never been very high considering the complexity of the task. Other complex tasks, such as mechanical design (which have been automated using Computer Aided Design) and shop floor tooling (which have been automated using numerical control), have been large investments because management perceives that the job can be done more cost effective. How can this preception be changed for software?

Technology Transition/[illegible]/VG13

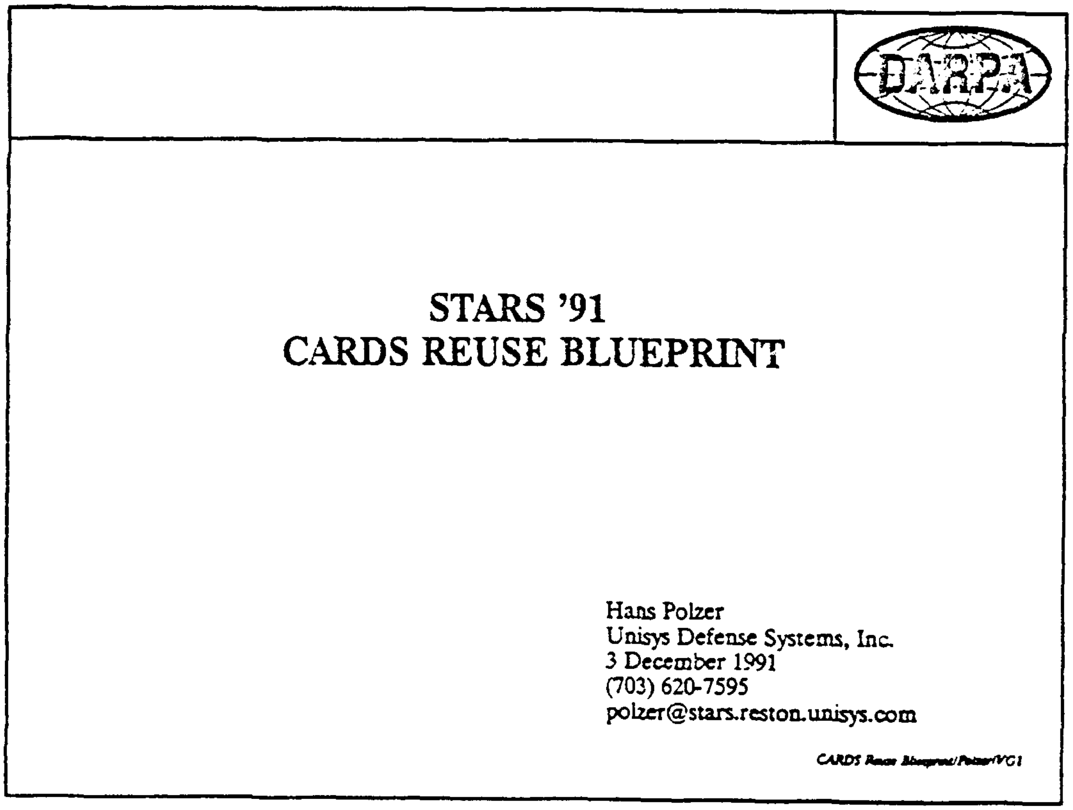


TECHNOLOGY TRANSITION

MEGAPROGRAMMING ADOPTION RISKS?

- **Are there other activities going on that could help, or accelerate, the adoption of Megaprogramming?**
- **Demonstration projects?**
- **Technology transfer?**

Technology Transition/Poston/VG16

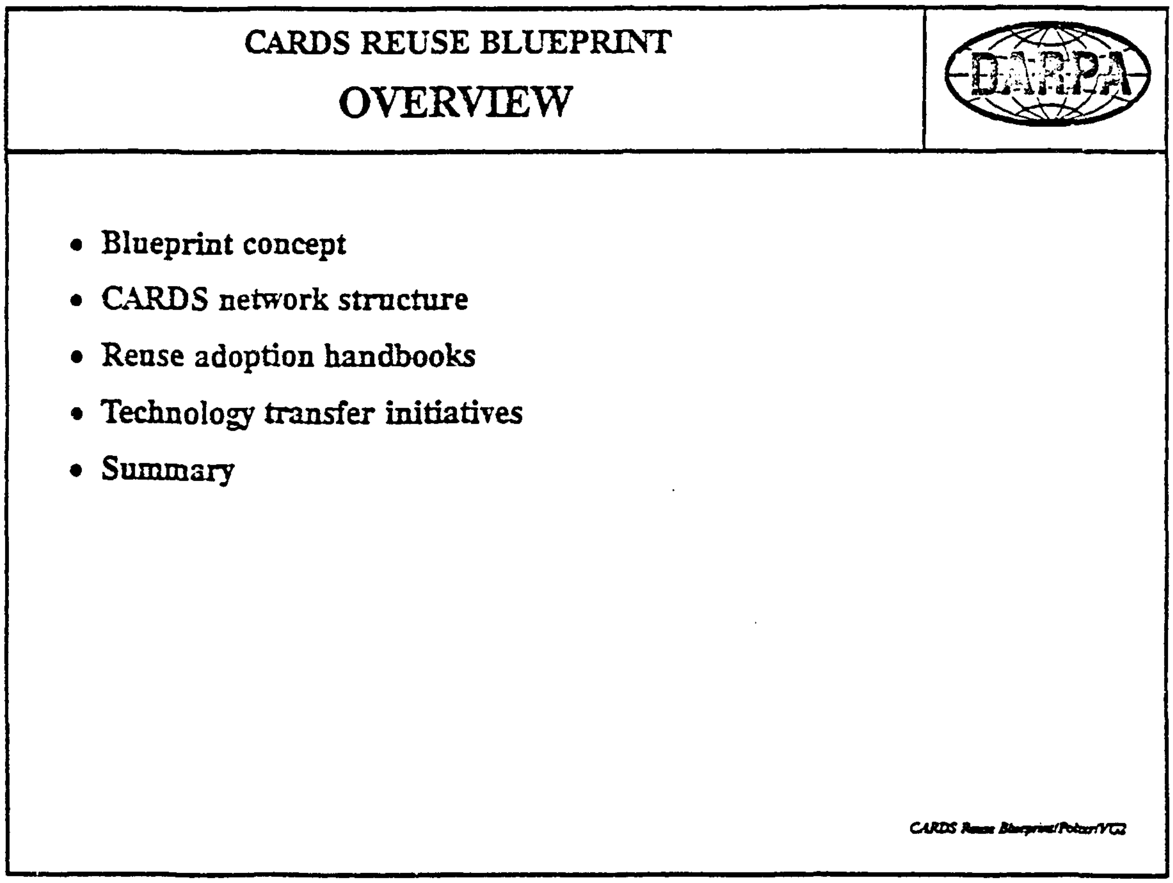


DARPA

STARS '91
CARDS REUSE BLUEPRINT

Hans Polzer
Unisys Defense Systems, Inc.
3 December 1991
(703) 620-7595
polzer@stars.reston.unisys.com

CARDS Reuse Blueprint/Polzer/VG1



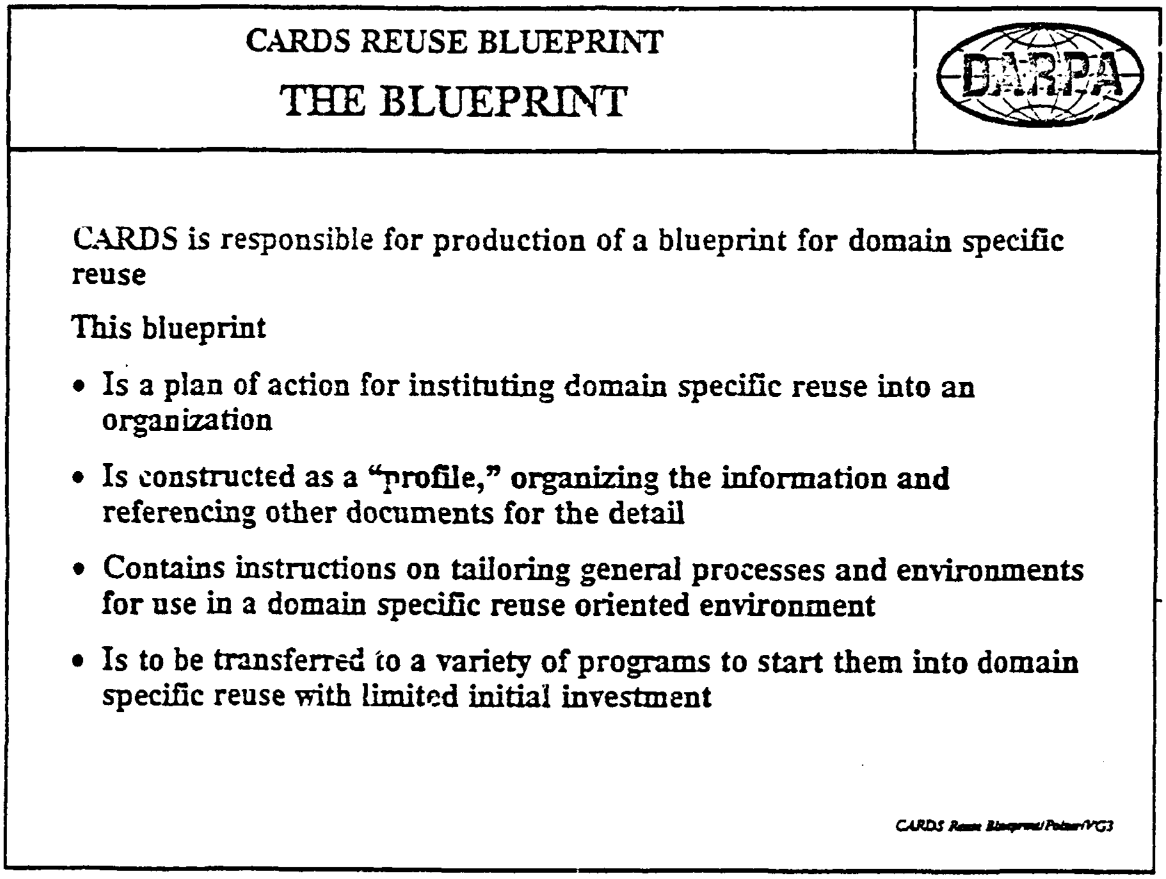
CARDS REUSE BLUEPRINT

OVERVIEW

- Blueprint concept
- CARDS network structure
- Reuse adoption handbooks
- Technology transfer initiatives
- Summary

CARDS Reuse Blueprint/Polzer/VG2

This presentation discusses the blueprint for domain-specific reuse being developed by the CARDS (Central Archive for Reusable Defense Software) program under the purview of the USAF Electronic Systems Division (ESD) and the STARS program. We will discuss the concept of the "knowledge blueprint" for reuse; the physical structure of the prototype reuse library that serves as a base for validation of the blueprint; the set of reuse adoption handbooks that form the heart of the blueprint; the other technology transfer initiatives being undertaken by CARDS; and, finally, a summary of the CARDS blueprint.



CARDS REUSE BLUEPRINT

THE BLUEPRINT

CARDS is responsible for production of a blueprint for domain specific reuse

This blueprint

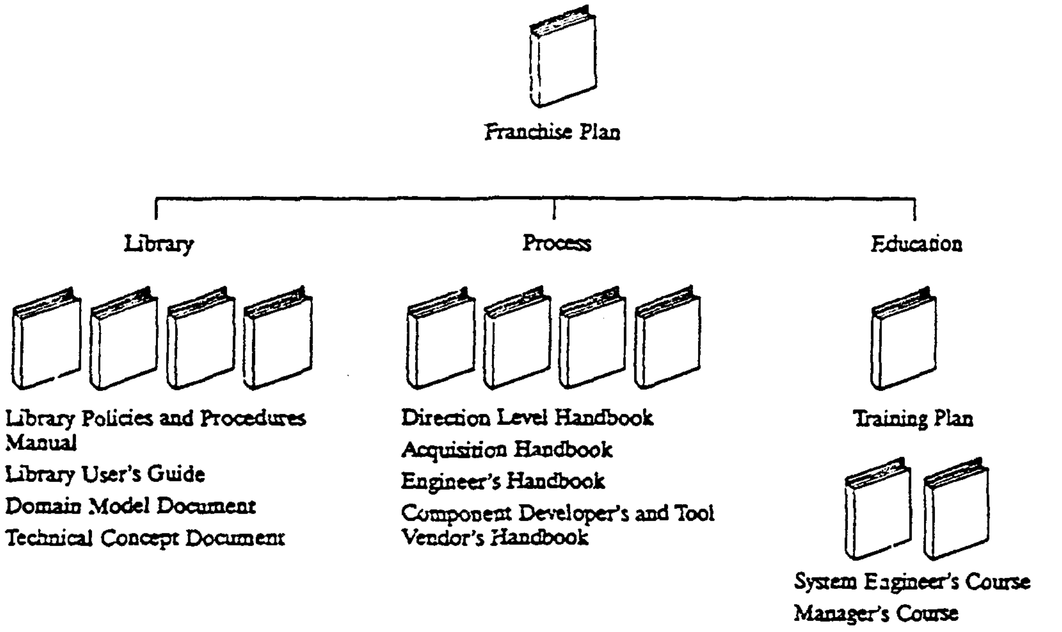
- Is a plan of action for instituting domain specific reuse into an organization
- Is constructed as a "profile," organizing the information and referencing other documents for the detail
- Contains instructions on tailoring general processes and environments for use in a domain specific reuse oriented environment
- Is to be transferred to a variety of programs to start them into domain specific reuse with limited initial investment

CARDS Reuse Blueprint/Poster/VG3

The primary mission of CARDS is to develop a "knowledge blueprint" for domain-specific reuse.

CARDS REUSE BLUEPRINT

BLUEPRINT CONCEPT



Franchise Plan

Library

Process

Education

Library Policies and Procedures Manual

Library User's Guide

Domain Model Document

Technical Concept Document

Direction Level Handbook

Acquisition Handbook

Engineer's Handbook

Component Developer's and Tool Vendor's Handbook

Training Plan

System Engineer's Course

Manager's Course

CARDS Reuse Blueprint/Poster/VG4

CARDS has three main goals:

- o Develop a prototype library for the command center domain as a testbed for the reuse concepts
- o Develop a "knowledge blueprint" for domain-specific reuse
- o Develop a training pian to teach domain-specific reuse to others

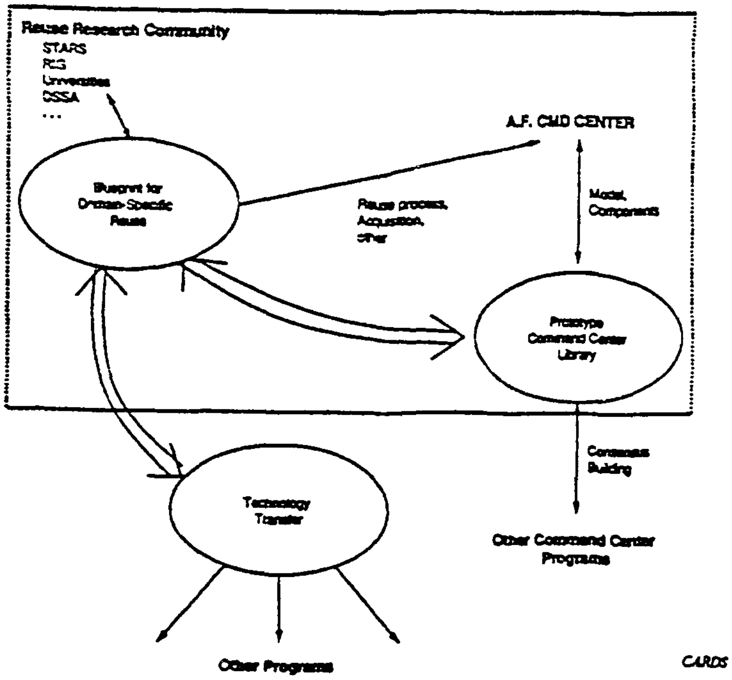
The prototype library work is developing a set of documents that describe the process of setting up and populating this library. These documents can be used by others interested in setting up domain-specific reuse libraries.

The knowledge blueprint is made up of documents describing the whole process of introducing and making effective use of domain-specific reuse.

The training portion of the project is explicitly tasked with teaching domain-specific reuse to others.

CARDS REUSE BLUEPRINT

BLUEPRPINT EVOLUTION



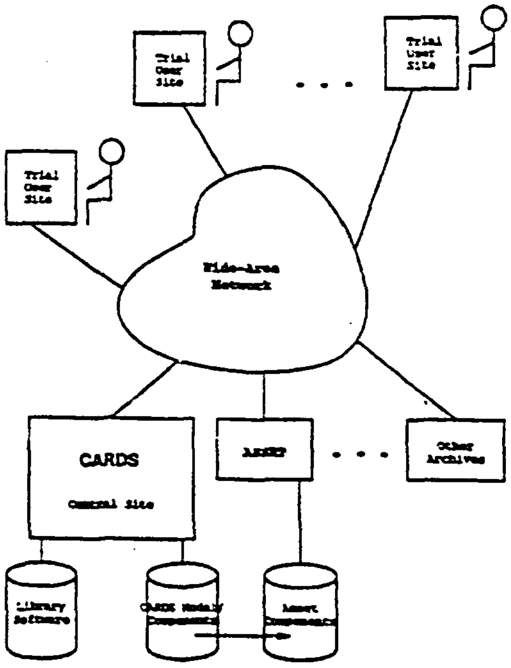
CARDS Reuse Blueprint/Polzer/VG5

The blueprint is based on results from the reuse research community, including work done by STARS, RIG (Reuse library Interoperability Group), universities, etc. These initial concepts have been used in setting up a prototype domain-specific library for Command Centers. Through interaction among the blueprint developers, the prototype library, and the commanċ ɔenter program at ESD, the blueprint is refined. The ,rototype library is reaching out to other command center programs in order to achieve a consensus on the structure and contents of a command center reuse library.

A main goal of the blueprint is to transfer the technology learned via its development into other programs.

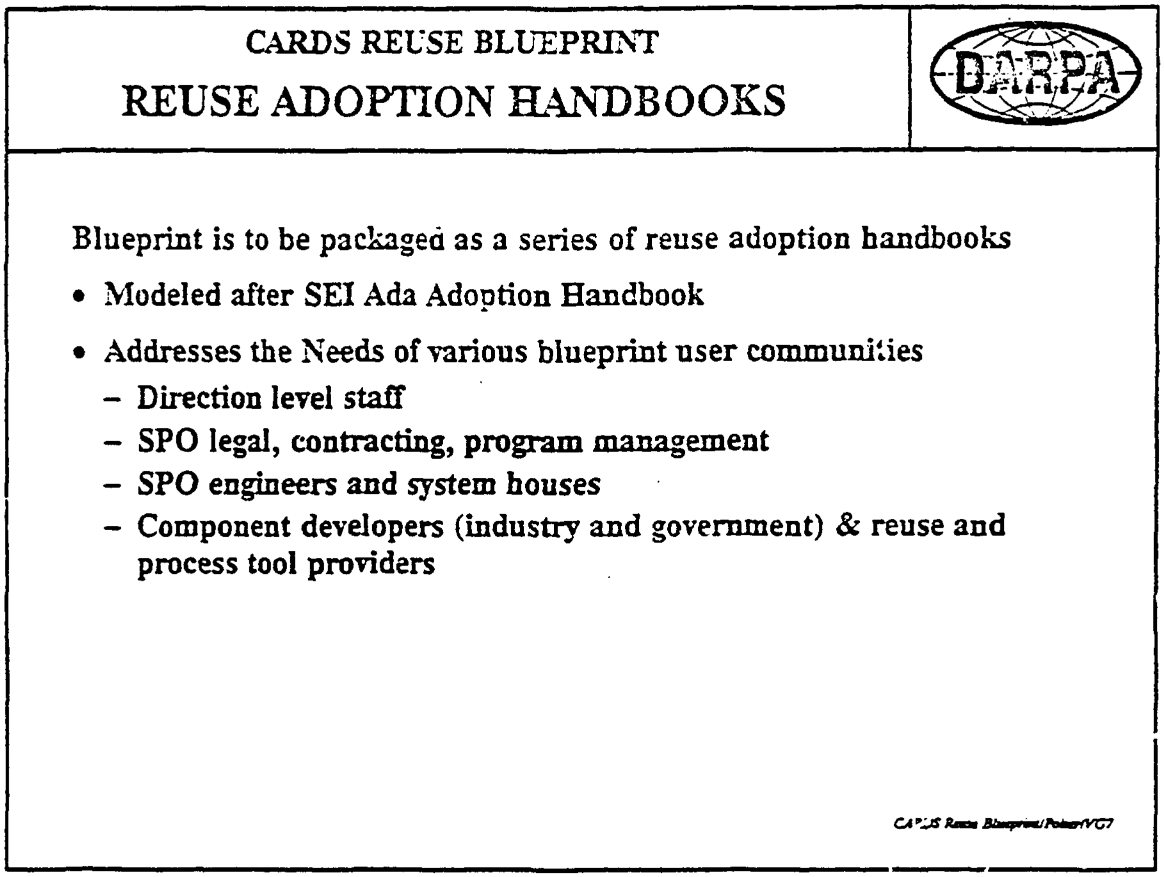
CARDS REUSE BLUEPRINT

CARDS NETWORK STRUCTURE



CARDS Reuse Blueprint/Poster/VG6

- o Central library management and maintenance site in Fairmont, WV
- o Four initial remote access sites, accessing the library via Internet
- o Can be extended to more sites as desired; additional trial users for the command center reuse library are solicited
- o Access to library is identical at all sites
- o Places access to library at the fingertips of the command center developers
- o Cooperating with ASSET library in experiments on library component interchange and interoperability



CARDS REUSE BLUEPRINT

REUSE ADOPTION HANDBOOKS

Blueprint is to be packaged as a series of reuse adoption handbooks

- Modeled after SEI Ada Adoption Handbook
- Addresses the Needs of various blueprint user communities
  - Direction level staff
  - SPO legal, contracting, program management
  - SPO engineers and system houses
  - Component developers (industry and government) & reuse and process tool providers

CARDS Reuse Blueprint/Poster/VG7

The heart of the knowledge blueprint for domain-specific reuse is a series of Reuse Adoption Handbooks. They are modeled after the Ada Adoption Handbook produced by Carnegie Mellon University's Software Engineering Institute.

In order for reuse to become widespread, various groups of people will have to be convinced of its advantages and taught how to implement it. These handbooks will address the issues from the perspective of

o Direction level staff, who oversee many programs

o Contracting and program management personnel, who will be issuing RFP's for specific programs

o Engineers who will be performing contracts involving reuse

o Component developers, who will develop assets that will be reused, and

o Tool vendors, who will provide software tools to aid the reuse process.

CARDS REUSE BLUEPRINT

HANDBOOK CONTENTS

Blueprint subject areas addressed across the handbooks are:

- Process—drives all other aspects of the blueprint: how the process affects each level
- Acquisition—system acquisition rules that have significant effect on reuse
- Benefits—investment costs and expected return on investment for reuse for each level
- Training—what training is required for this level to effect a reuse program
- Security—protection of secure components in the recommended reuse environment
- Consensus—obtaining multi-organizational consensus on domain models

CARDS Reuse Blueprint/Poster/VG8

There are several issues that cut across several of the different categories of people involved with reuse, and will thus be addressed in several of the Handbooks.

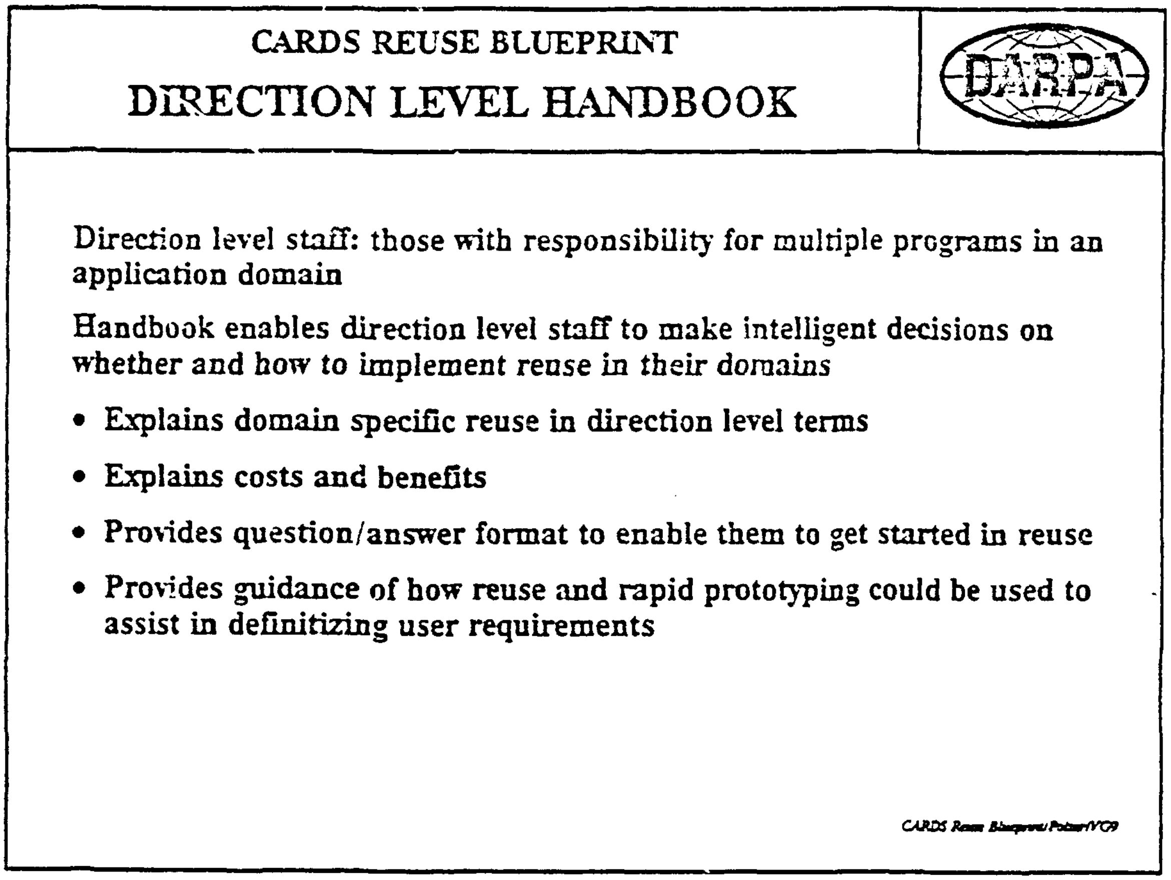
o Process—affects all categories

o Acquisition—direction, acquisition, and engineers are all affected by rules that encourage or discourage reuse

o Benefits—all categories need to know, "What's in it for me?"

o Security—direction level and program managers want to be assured their programs are secure; engineers and tool vendors have to know how to ensure the security

o Consensus—all those involved must reach consensus on reuse



CARDS REUSE BLUEPRINT

DIRECTION LEVEL HANDBOOK

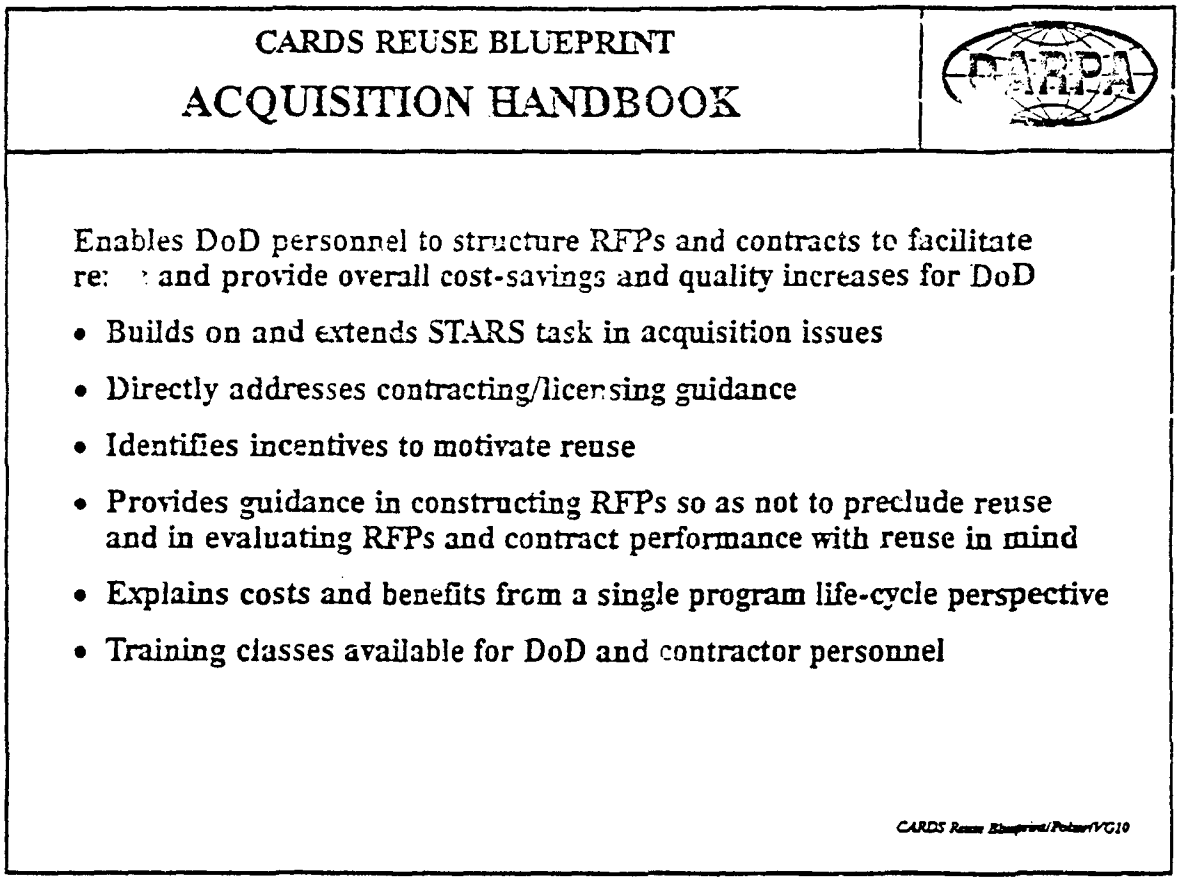
Direction level staff: those with responsibility for multiple programs in an application domain

Handbook enables direction level staff to make intelligent decisions on whether and how to implement reuse in their domains

- Explains domain specific reuse in direction level terms
- Explains costs and benefits
- Provides question/answer format to enable them to get started in reuse
- Provides guidance of how reuse and rapid prototyping could be used to assist in definitizing user requirements

CARDS Reuse Blueprint/Poster/VG9

The direction level handbook addresses reuse issues from the perspective of people who oversee many programs. They are in a position to foster reuse and also stand to gain a lot with the adoption of reuse. A single program, because of the added expense and aggravation, may resist developing components that can be used by other programs, but a director can see the overall benefits across programs.



CARDS REUSE BLUEPRINT

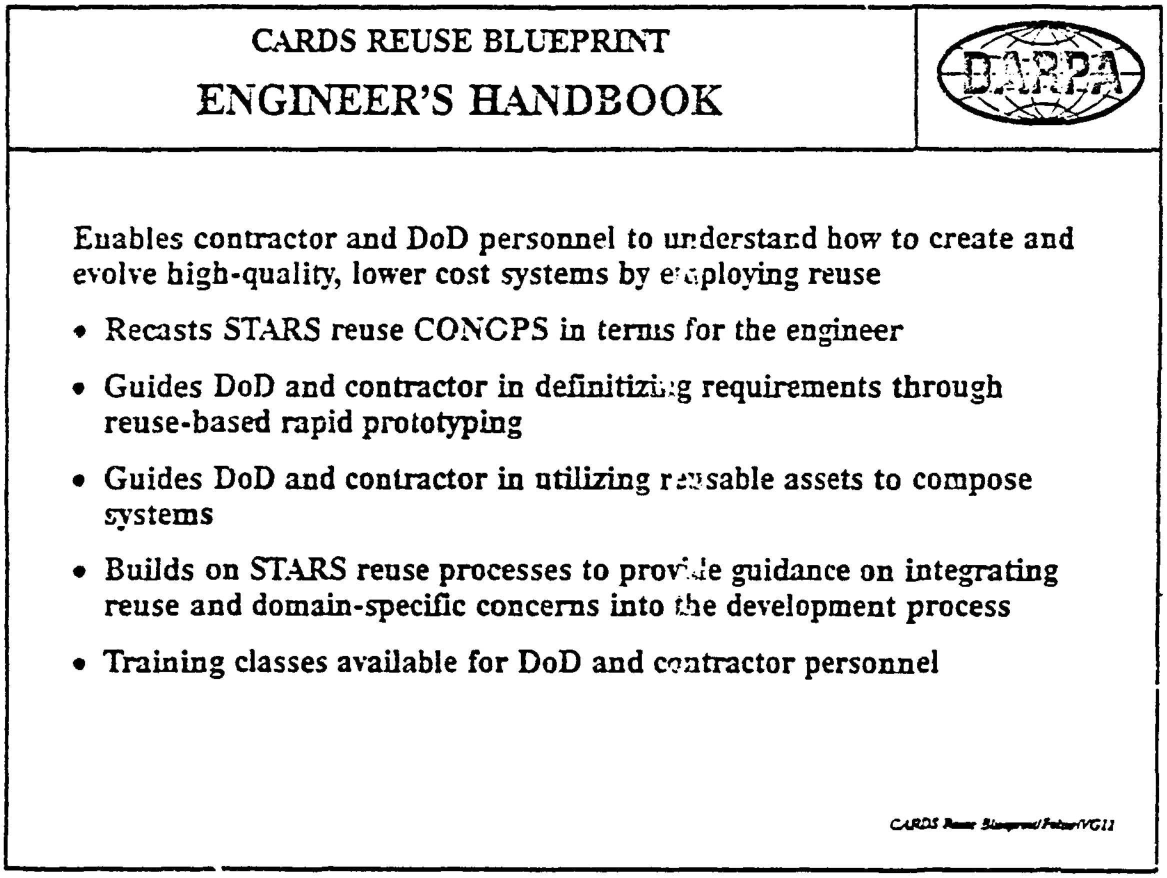
ACQUISITION HANDBOOK

Enables DoD personnel to structure RFPs and contracts to facilitate reuse and provide overall cost-savings and quality increases for DoD

- Builds on and extends STARS task in acquisition issues
- Directly addresses contracting/licensing guidance
- Identifies incentives to motivate reuse
- Provides guidance in constructing RFPs so as not to preclude reuse and in evaluating RFPs and contract performance with reuse in mind
- Explains costs and benefits from a single program life-cycle perspective
- Training classes available for DoD and contractor personnel

CARDS Reuse Blueprint/Poster/VG10

Many current RFPs are structured to actively discourage reuse. This handbook will explain why reuse is beneficial to a single program and how to write RFPs that encourage the contractor to adopt reuse.



CARDS REUSE BLUEPRINT

ENGINEER'S HANDBOOK

Enables contractor and DoD personnel to understand how to create and evolve high-quality, lower cost systems by employing reuse

- Recasts STARS reuse CONCPS in terms for the engineer
- Guides DoD and contractor in definitizing requirements through reuse-based rapid prototyping
- Guides DoD and contractor in utilizing reusable assets to compose systems
- Builds on STARS reuse processes to provide guidance on integrating reuse and domain-specific concerns into the development process
- Training classes available for DoD and contractor personnel

CARDS Reuse Blueprint/Foils/VG11

The engineers who are actually carrying out system development are vital to the success of reuse; their passive resistance or lack of understanding could sabotage the possibility of successful reuse. This handbook explains the concepts and mechanisms for reuse adoption by the system developers across the entire life-cycle, from requirements definition through prototype to delivered system and maintenance.

CARDS REUSE BLUEPRINT

COMPONENT AND TOOL HANDBOOK

DARPA

Enables component developers and tool vendors to understand how to create reusable components and tools for reuse

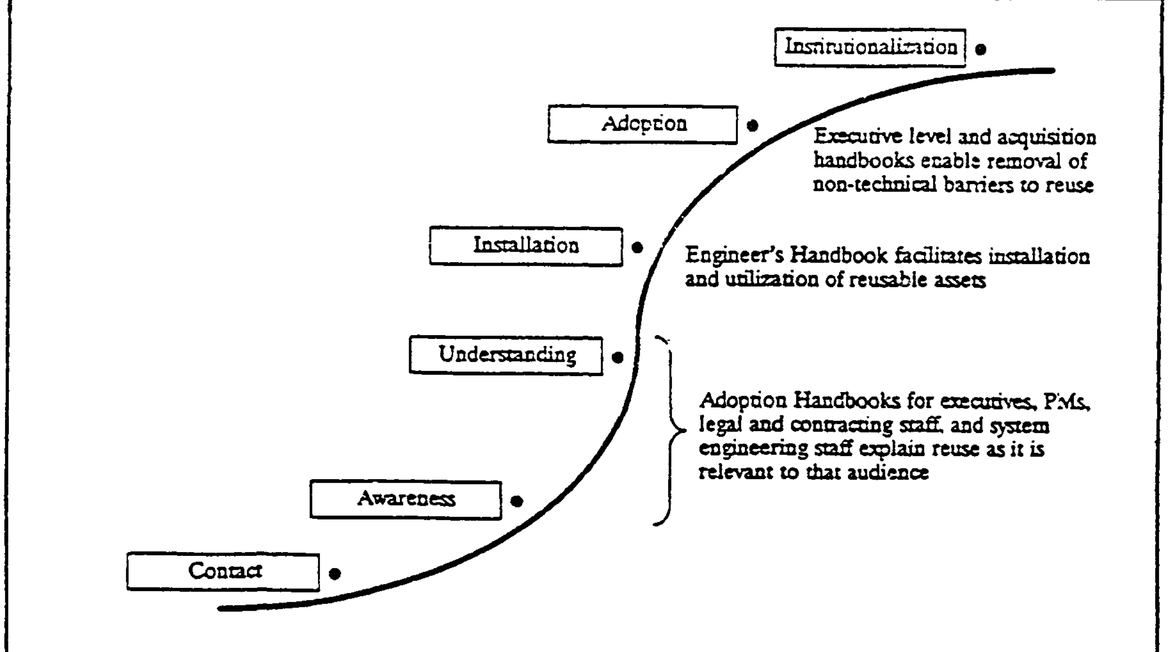
- Guidance for developing reusable components
- Specifies requirements for tools to support reuse-based system development and maintenance
- Explains incentives for developing these components and tools

CARDS Reuse Blueprint/Poster/VG12

For reuse to be successful, commercial reusable components and software tools that assist reuse will have to be readily available for purchase. This handbook informs potential vendors of the needs and the benefits to them of meeting those needs.

CARDS REUSE BLUEPRINT

SUPPORTS ASPECTS OF TT MODEL



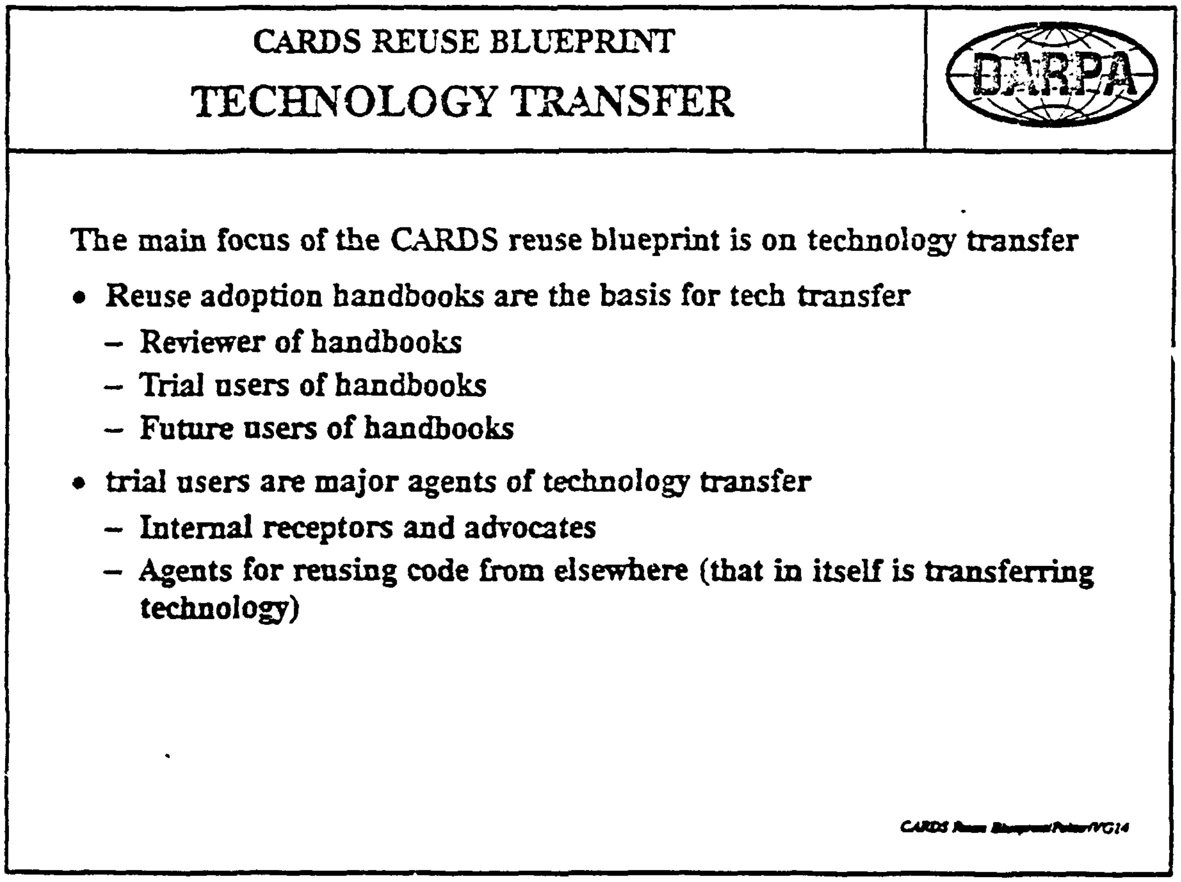
Executive level and acquisition handbooks enable removal of non-technical barriers to reuse

Engineer's Handbook facilitates installation and utilization of reusable assets

Adoption Handbooks for executives, PMs, legal and contracting staff, and system engineering staff explain reuse as it is relevant to that audience

CARDS Reuse Blueprint/Poster/VG13

The Software Engineering Institute (SEI) has developed the above S-curve indicating the stages involved along the way toward the institutionalization of reuse. We show where these handbooks will fit on this path. They will aid an organization in moving from awareness of reuse, to understanding, to installation, and then to adoption of reuse by the organization.



CARDS REUSE BLUEPRINT

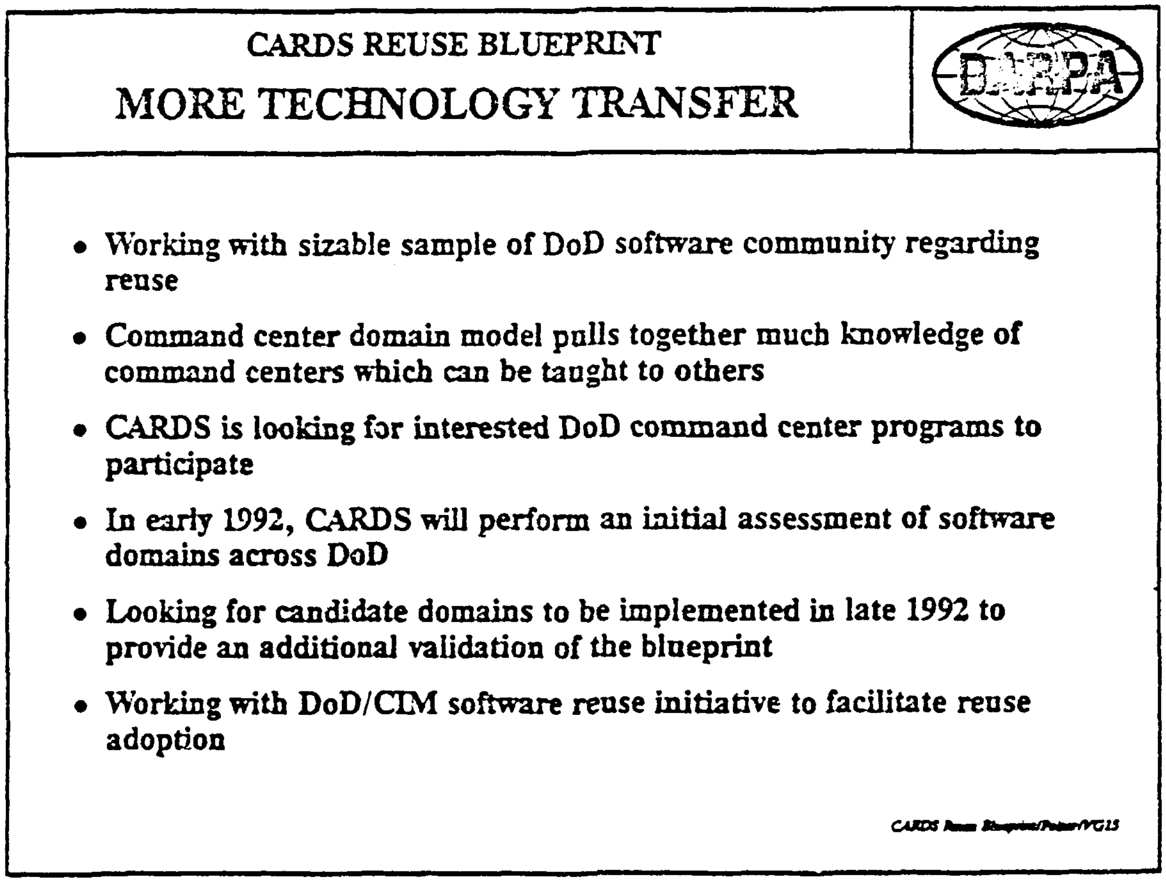
TECHNOLOGY TRANSFER

The main focus of the CARDS reuse blueprint is on technology transfer

- Reuse adoption handbooks are the basis for tech transfer
  - Reviewer of handbooks
  - Trial users of handbooks
  - Future users of handbooks
- trial users are major agents of technology transfer
  - Internal receptors and advocates
  - Agents for reusing code from elsewhere (that in itself is transferring technology)

CARDS Reuse Blueprint/Poster/VG14

The primary mission of CARDS is to learn about application of domain specific reuse through implementation and use of a prototype library, and then to disseminate the information as widely and effectively as possible. The Reuse Adoption Handbooks are the center of this strategy, but not the only aspect of the program addressing technology transfer. The trial users of the prototype itself are learning reuse, and will be agents of change within their organizations. Furthermore, the very act of reusing components from other organizations is transferring technology.



CARDS REUSE BLUEPRINT

MORE TECHNOLOGY TRANSFER

DARPA

- Working with sizable sample of DoD software community regarding reuse
- Command center domain model pulls together much knowledge of command centers which can be taught to others
- CARDS is looking for interested DoD command center programs to participate
- In early 1992, CARDS will perform an initial assessment of software domains across DoD
- Looking for candidate domains to be implemented in late 1992 to provide an additional validation of the blueprint
- Working with DoD/CIM software reuse initiative to facilitate reuse adoption

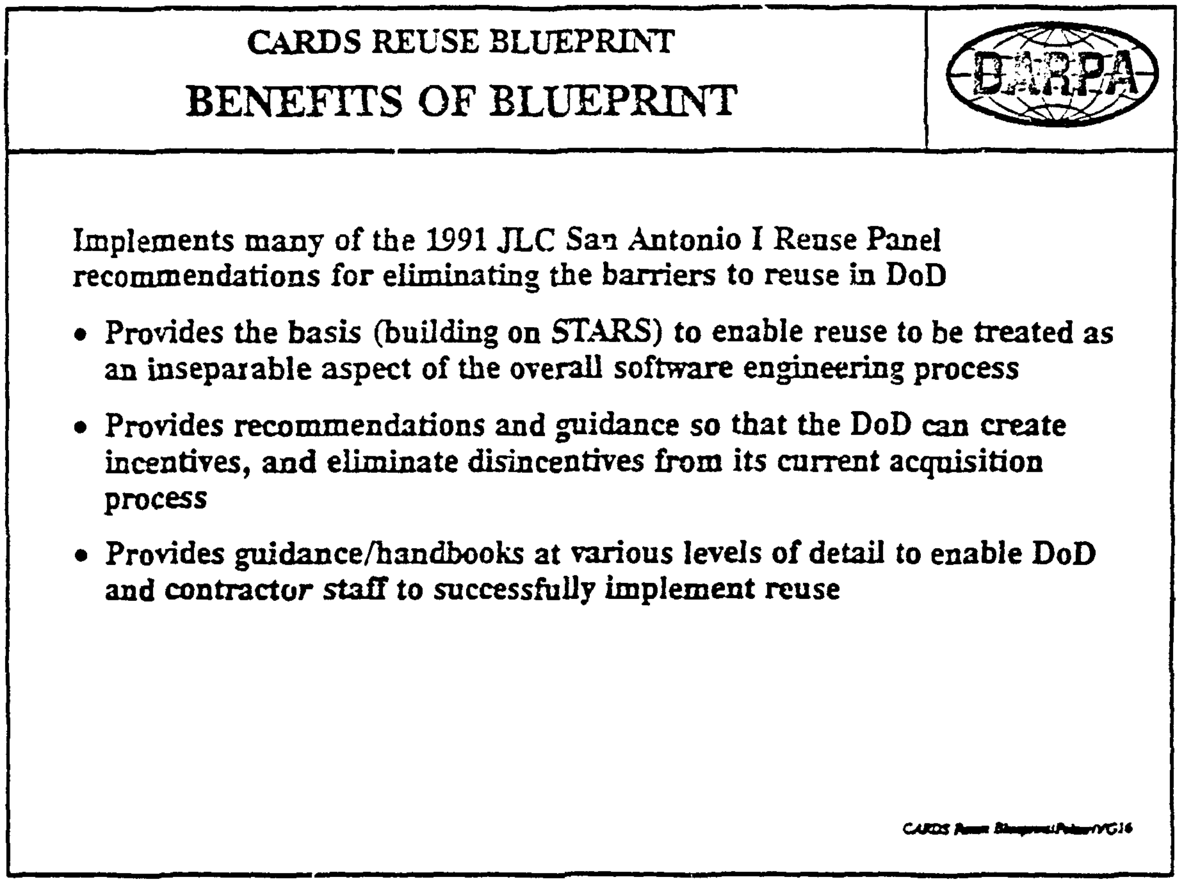
CARDS Reuse Blueprint/Poster/VG15

In addition to the blueprint and the prototype library users, there are other aspects to technology transfer by CARDS:

o We are working with other DoD groups interested in reuse

o We are looking for other interested command center groups to participate with the prototype command center library, and

o We are looking for a second domain for which to begin implementation of a domain-specific library in late 1992.



CARDS REUSE BLUEPRINT

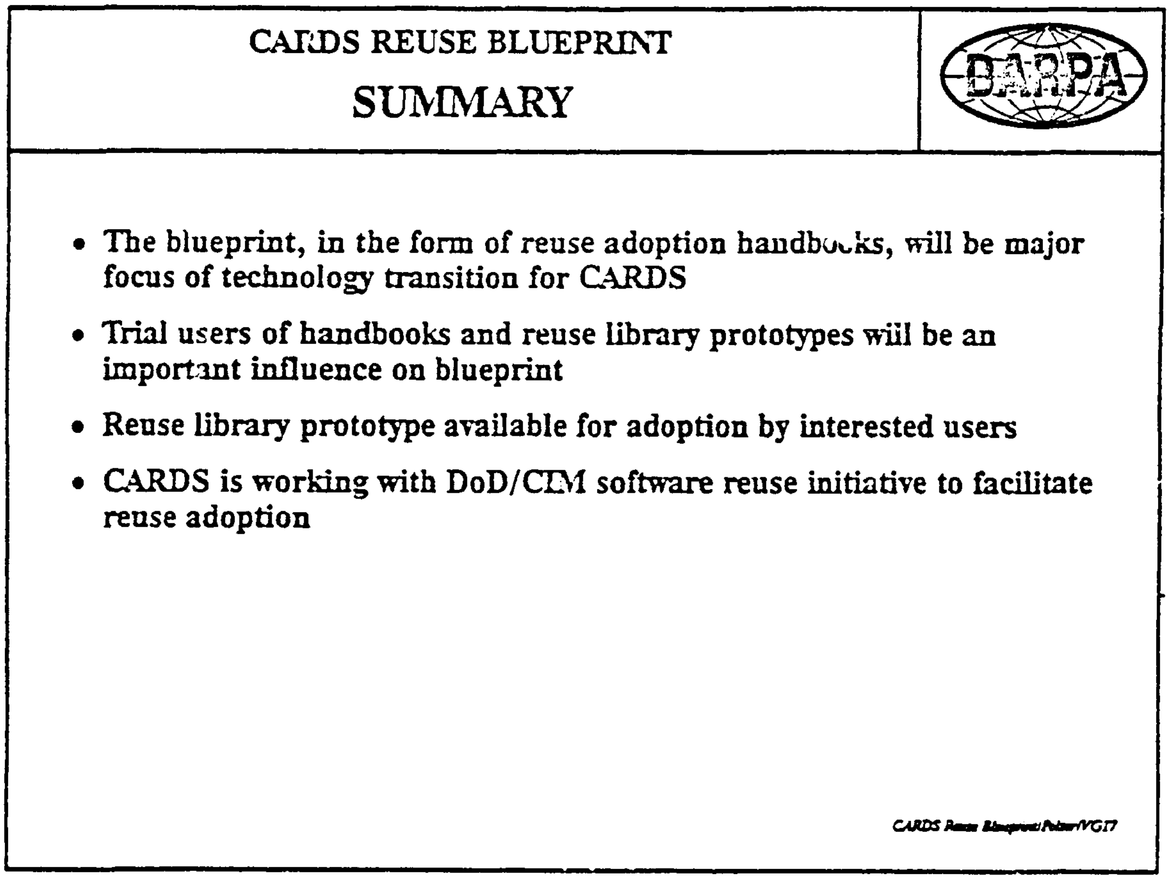
BENEFITS OF BLUEPRINT

Implements many of the 1991 JLC San Antonio I Reuse Panel recommendations for eliminating the barriers to reuse in DoD

- Provides the basis (building on STARS) to enable reuse to be treated as an inseparable aspect of the overall software engineering process
- Provides recommendations and guidance so that the DoD can create incentives, and eliminate disincentives from its current acquisition process
- Provides guidance/handbooks at various levels of detail to enable DoD and contractor staff to successfully implement reuse

CARDS Reuse Blueprint/Poster/VG16

In autumn 1991 a reuse panel sponsored by the Joint Logistics Command convened in San Antonio to discuss the barriers that are keeping reuse from being widely adopted within the DoD. The handbooks being produced will address several of these obstacles (listed above) and encourage their elimination.



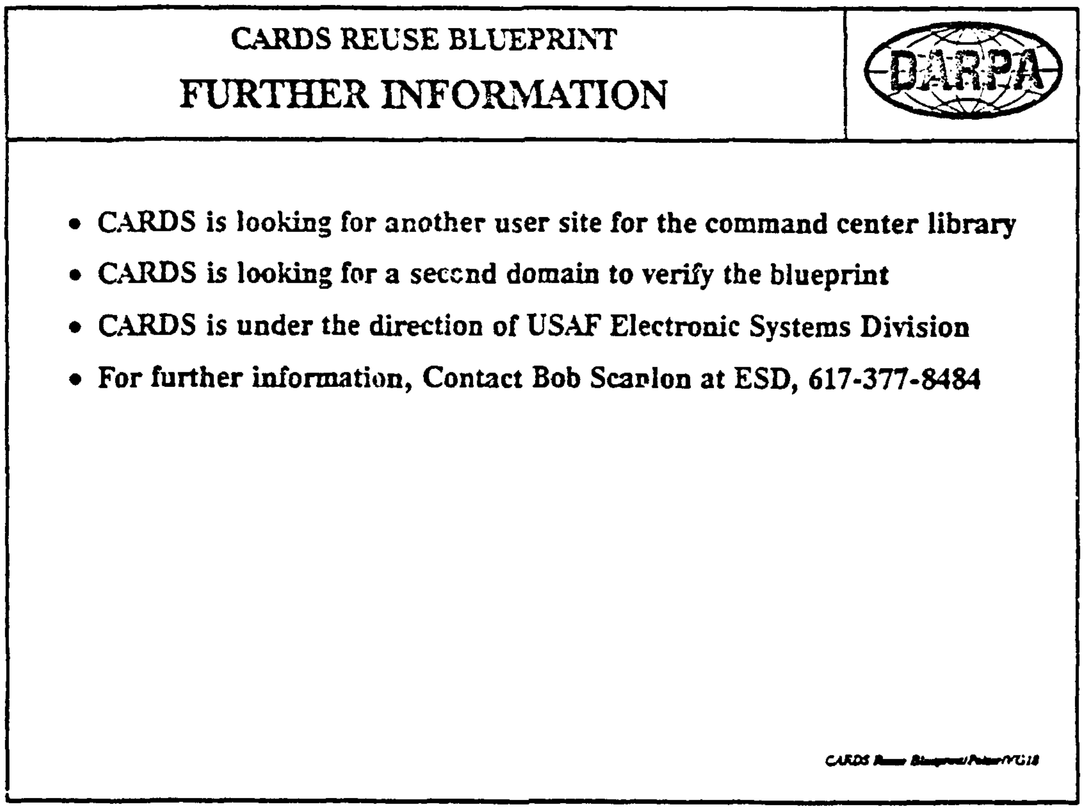
CAI:DS REUSE BLUEPRINT

SUMMARY

- The blueprint, in the form of reuse adoption handbooks, will be major focus of technology transition for CARDS
- Trial users of handbooks and reuse library prototypes will be an important influence on blueprint
- Reuse library prototype available for adoption by interested users
- CARDS is working with DoD/CIM software reuse initiative to facilitate reuse adoption

CARDS Reuse Blueprint/Palmer/VG17

Those interested in participating as a trial user site for the Command Center library, or those from another domain interested in setting up a reuse library for that domain, should contact Mr. Scanlon of USAF ESD at 617-377-8484.



CARDS REUSE BLUEPRINT

FURTHER INFORMATION

- CARDS is looking for another user site for the command center library
- CARDS is looking for a seccnd domain to verify the blueprint
- CARDS is under the direction of USAF Electronic Systems Division
- For further information, Contact Bob Scanlon at ESD, 617-377-8484

CARDS Reuse Blueprint/Poster/VG18

STARS AFFILIATES PROGRAM

STARS has established an affiliates program. This program provides an opportunity for the software community to participate in the technology activities associated with the STARS Program and to join with STARS in accelerating the paradigm shift to megaprogramming.

STARS affiliates are individual representatives of organizations involved in software development for the government, including government contractors, universities, government agencies, and environment/tool vendors.

Three levels of affiliates have been established:

* Information Affiliates: Information Affiliates have access to information regarding the STARS program such as newsletters; are included on the STARS mailing list; have access to the bulletin board; and may participate in the monthly briefings and demonstrations at the STARS Technology Center.

* Technology Transfer Affiliates: This level of affiliate is expected to cooperate with the STARS program on a consistent basis to aid in technology transition to/from STARS and the DoD community. Technology Transfer affiliates are expected to appoint a single point of contact within their organization who will participate actively in technology exchange working group meetings. These working group meetings would be coordinated by STARS and would meet periodically with network interaction between meetings. Sub-groups might be established to focus on specific technology areas. These affiliates would become familiar with the STARS Program and participate in all working group meetings. They may also be asked to be alpha test sites for new STARS products.

* Prime Affiliates: A Prime Affiliate works directly with one of the STARS Prime Contractors (Boeing, IBM, Unisys) in technology activities relevant to the STARS Program, such as product evaluation, technology transition, technology integration, and tool development. In addition to participation in periodic workshops as described above, Prime Affiliates may also participate in prime team meetings. Joint activities with any of the Prime Contractors are arranged directly with that prime on a case-by-case basis.

Participation in the Technology Transfer or Prime Affiliate program will require that you complete an affiliates questionaire and sign a non-disclosure agreement.

Labor, travel and trial usage expenses associated with participation in the affiliates program is the responsibility of each affiliate's parent (sponsoring) organization. STARS provides meeting accommodations and network access.

If you are interested in becoming a STARS Affiliate, please indicate the level of affiliation and address your request to the STARS Technology Center, Attn: Affiliates Desk, Suite 400, 801 North Randolph, Arlington, VA 22203 or call (703) 243-8655 or Email to "affiliates-desk@stars.rosslyn.unisys.com".

STARS AFFILIATE
NONDISCLOSURE AGREEMENT

The Software Technology for Adaptable, Reliable Systems (STARS) program is directed out of the Defense Advanced Research Projects Agency (DARPA) with contract administration provided by the Air Force Electronic Systems Division and sponsored by the Department of Defense (DoD) through the DoD Consolidated Software Initiative. This Agreement is entered into as of this __________ day of ___________, 19___, by and between the STARS program and ________________________________ (Affiliate). For purposes of this Agreement, the term Affiliate is used generically to include any entity of any kind, or its employees or agents.

The parties agree:

1. The STARS program in discharging its obligations to the DoD may have access to proprietary information belonging to third parties. The STARS program's mission is to accelerate the paradigm shift to megaprogramming - a process-driven, domain-specific reuse-based, technology supported ways of doing business. Instances may occur where the aforementioned information or technology will be held as proprietary to the U.S. Government.

2. The Affiliate, in connection with the work of STARS, may have access to STARS, third party proprietary information, or to Government information designated "For Official Use Only". The foregoing described information or technology shall be disclosed within the STARS program on a "need to know" basis and shall not be disclosed outside of STARS without specific written authorization from the STARS program office and the Electronic Systems Division (ESD) Public Affairs Office.

3. Any information disclosed to the Affiliate shall not be deemed to be confidential or proprietary and the Affiliate shall have no obligation with respect to any such information which:
   a. was known to STARS or to the Affiliate and unrestricted at the time it was submitted by a third party, or
   b. was previously cleared for public release through the DARPA STARS Program Manager or the STARS program office (ESD/AVS) or cleared through the ESD Public Affairs, or
   c. is in the process of being cleared for public release through ESD/AVS or the ESD Public Affairs, or
   d. is received by the Affiliate from another third party without restrictions and without breach of the agreement with the initial disclosure.

4. In the case of STARS information designated "Distribution Statement C," the Affiliate will be relieved of the limitations imposed herein upon receipt of specific written authorization or removal of the said designation by the STARS Program Office (ESD/AVS).

5. Upon ending affiliation with STARS, the Affiliate will not, either in whole or part, take or keep any drawings, blueprints, documents, computer programs, compilations or technical data, specifications or other records of any nature (whether written or in machine readable form) proprietary to STARS or to any third party, or any Government information designated "For Official Use Only", or any reproductions of any of the foregoing described information.

Affiliate
By:______________________
Date:____________________

STARS
By:____________________
Date:__________________

Software Technology for Adaptable Reliable Systems (STARS)
Affiliates Questionnaire

Note: If you need additional space, use the back of this questionnaire, or additional sheets.

1. Company/Organizaticn Name____
   Division/Group/Organization____

2. Primary Contact:
   Name:____
   Address:____
   ____
   Phone:____
   FAX:____
   Internet:____

3. What level of STARS affiliation do you want?
   a. Technology Transfer
   b. Prime (Circle preferred Prime - Boeing, IBM, Unisys)

4. What is your Company's/Organization's primary business area?
   (Mark all that arr applicable)
   a. Sys'ems
   b. Software
   c. Hardware
   d. Manufacturing
   e. Consulting
   f. Systems Integration
   g. Other (please specify)

5. Does your company/organization (please specify which) have an active technology receptor organization?

6. If yes to question 5, please provide contact information:

Name:____

Address:____

____

Phone:____

7. Please describe your company's/organization's primary area of interest in software engineering technology.

____

____

8. Please describe how being a STARS Affiliate can best serve the needs of your organization (i.e., in what area are you interested in working with STARS)

____

____

____

9. Comments (please provide any additional information, views, or questions you may have):

____

____

____

10. Please attach a one-page position paper describing area(s) of greatest interest where cooperative development would be mutually beneficial and a short resume' for each participant.

12/03/9[illegible] Page:

PARTICIPANTS BY LAST NAME
For Meeting Code: stars91
STARS '91 CONFERENCE ATTENDEE LIST

Last Name	First Name	Reg Id	Company
Abraham	Rebecca	500406	Defense Info. Systems Agency
Adams	Dale	5000[illegible]4	Rockwell International
Adhikari	Sam	500166	AT&T BELL LABS
Ahearn	Tom	500211	Atherton Technology
Albert[illegible]	Glen	500237	Central Intelligence Agency
Alexander	Robert	500297	Electronic Warfare Assoc.
Anderson	Jim	500439	Hewlett Packard
Anderson	Larry	500306	Defense Intelligence Agency
Andrews	Phil	500470	Dept of the Navy
Angier	Bruce	500241	Inst. for Defense Analyses
Aridas	James	500190	UNISYS Defense Systems, Inc.
Armitage	James	500054	Software Engineering Inst.
Armstrong	[illegible]	500031	LWA
Arnold	Paul	500264	IBM
Arnold	Robert	500498	Software Productivity Consort.
Arrington	Curtis	500464	A.F. Military Personnel Ctr
Audette	Donald	500473	NCTS Washington
Austin	Antoinette	500103	INEL/EG & G Idaho
Bailey	Elizabeth	500[illegible]40	Inst. for Defense Analyses
Baldo	James	500497	I.D.A.
Ballentine	William	500187	Digital Equipment Corp.
Bansel	Daniel	500116	Electronic Warfare Assoc.
Barbosa	Edson	500507	Brazilian' Software Plant Proj.
Barcham	D.E.	500506	Systematica
Barksdale	Gerry	500179	HUGHES AIRCRAFT CO.
Bassman	Mitchell	500400	Computer Sciences Corp.
Baudoin	Claude	500163	NSC
Beals	Gilbert	500475	Lockheed
Bebb	Joan	500021	TRW, Inc.
Becker	Cynthia	500201	Digital Equipment Corporation
Beckhardt	Arnold	500344	Software Engineering Tech.
Belz	Frank	500352	TRW
Berggren	Peter	500279	SAIC
Beser	Eric	500404	Westinghouse
Biancaniello	Steve	500265	IBM
Biggar	Donald	500104	UNISYS
Bird	Mark	500434	U.S. General Acctg. Office
Bjornestad	Mo	500511	Mark V Systems
Bladen	Jim	500320	TeleSoft
Bloodgood	Charles	500167	SEMATECH
Blue	David	500350	CTA, Inc.
Blumberg	Maurice	500266	IBM
Boehm	Barry	500340	DARPA
Bond	Paul	500025	ASSET
Bowes	Robert	500202	Acquisition Enter.
Boyer	Wayne	500180	EG & G
Braun	Linda	500057	MountainNet, Inc.
Braunstein	Mary Ann	500267	UNISYS Defense Systems
[illegible]	[illegible]	500487	USTRANSCOM
Brewster	[illegible]	[illegible]	IBM
Bristow	David	500476	SWSC
Brodbeck	Wayne	500170	AT & T
Brodman	Judith	[illegible]	[illegible]

PARTICIPANTS BY LAST NAME
For Meeting Code: stars91
STARS '91 CONFERENCE ATTENDEE LIST

Last Name	First Name	Reg Id	Company
Brown	George	500343	General Research Corporation
Brown	Linda	500467	OSD (C3I)
Brown	Mike	500480	ArborText
Bryant	Marchita	500191	IBM
Buchman	Caroline	500188	Allied Signal Aerospace
Bulat	Brian	500267	IBM
Burns	Eva	500307	Department of Defense
Burton	Daniel	500389	Carnegie Mellon University
Cate	C. Spencer	500014	HQ USAF/SCXS
Capizzi	Joseph	500171	UNISYS
Cappellini	Andrea	500105	AT&T Bell Labs
Carlson	Susan	500062	Ada IC
Carlstrom	Donald	500193	FCDSSADN (USN)
Carney	David	500462	Inst. for Defense Analyses
Ceely	Dave	500268	IBM
Chao	David	500257	U.S. General Accounting Office
Chapman	Jack	500124	UNISYS
Chappell	Alan	500492	Naval Oceanographic Office
Chen	Cliff	500159	Texas Instruments
Cherinka	Robert	500095	USAF
Christie	Alan	500432	Software Engineering Institute
Ciomei	L.	500142	UNISYS
Cirbus	Amy	500125	UNISYS
Clark	Eileen	500236	The MITRE Corporation
Clark	Mike	500423	Cadre Technologies
Clark	Paul	500355	Trusted Information Sys. Inc.
Cleveland	Scott	500037	Systems Engineering Research
Cobb	David	500510	Naval Underwater Systems Ctr
Cobb	Richard	500365	Software Engineering Tech.
Coheeney	Thomas	500337	FCDSSADN (US Navy)
Cohen	Howard	500215	The MITRE Corporation
Coleman	Willie	500440	Hewlett Packard
Comer	Ed	500182	Software Productivity Solut.
Conley	Richard	500382	VERITAS Software
Converse	Robert	500489	Computer Sciences Corp.
Cool	TW	500212	Atherton Technology
Copping	Rick	500213	Atherton Technology
Courtwright	Terry	500194	CEA
Coutant	Ray	500374	Contel
Craig	Alan	500109	Computer Sciences Manager
Crawford	Brad	500391	TASC
Crepps	Dick	500126	UNISYS Defense Systems, Inc.
Croak	Thomas	500400	HQ USAF/SCX
Crutchfield	Brenda	500269	IBM
Cunius	Robert	500199	Control Data Corporation
Daniels	Jack	500508	USAISSC
Danner	Bonnie	500286	TRW Systems Division
Dart	Susan	500433	Software Engineering Institute
Davis	Margaret	500412	Boeing
Day	John	500270	IBM
De Baun	Gerald	500304	VERITAS Software
Deller	Steve	[illegible]	[illegible]

PARTICIPANTS BY LAST NAME
For Meeting Code: sts91
STARS '91 CONFERENCE ATTENDEE LIST

Last Name	First Name	Reg Id	Company
DeNoi	Edward	500316	IBM Corporation
DePalma	Evelyn	500245	Center for Information Mgmt.
DeRiso	Michael	500088	Software Engineering Institute
Dern	Bob	500059	Air Force STSC
DiBona	Robert	500111	Softech
Dikel	David	500385	Applied Expertise Inc.
Distin	David	500348	DOD/Defense Info. Syst. Agency
Doland	Jerry	500127	UNISYS
Donaldson	Bruce	500453	DSD Laboratories
Doudican	Zacodia	500154	Litton Data Systems
Dougherty	Marie	500121	UNISYS Defense Systems
Dowson	Mark	500513	Marlstone Software Technology
Drake	Dick	500271	IBM
Druffel	Larry	500446	Software Engineering Institute
Duncan	Ray	500040	Simms Industries, Inc.
Dzubay	Antoinette	500305	Defence Intelligence Agency
Economou	Dorian	500479	IBM
Elman	Bob	500272	IBM
Ellenhorst	Daniel	500203	Martin Marietta DS&C
Engstrom	Eric	500239	Honeywell
Epperson	Thomas	500199	DISA/DSSO/JN
Erb	Dona	500370	The MITRE Corporation
Eschallier	Fred	500120	UNISYS Defense Systems
Essex	David	500033	DSD Laboratories
Estes	Gene	500428	UNISYS Defense Systems
Ett	Bill	500273	IBM
Farrell	William	500022	DSD Laboratories Inc.
Feiler	Peter	500431	Software Engineering Institute
Fischer	Herman	500162	MARK V SYSTEMS
Fischer	Kurt	500329	DOD/DDI
Fleming	Barbara	500247	Center for Information Mgmt.
Flemings	Karen	500056	MountainNet, Inc.
Foreman	John	500341	DARPA/SISTO
Foster	David	500358	Martin Marietta
Fowler	Priscilla	500436	Software Engineering Institute
Fowler	Tim	500118	PRC Inc.
Fox	Joe	500421	Software Arch. & Engineering
Fox	Joe	500450	SAE
Frank	Lawrence	500416	Boeing
Frankl	Adam	500444	Interactive Develop. Environ.
Frantz	Albert	500401	US Air Force
Frazier	Thomas	500343	Inst. for Defense Analyses
Freer	Rob	500463	Tele Soft
Frigon	Pierre	500028	Paramax Electronics
Froscher	Judith	500100	Naval Research Laboratory
Fuhrer	James	500310	UNISYS
Fuhrer	James	500471	UNISYS
Fulton	James	500104	US Geological Survey
Gallagher	Edward	500030	U.S. Army CECOM RD&E Center
Gan	Judy	500505	IBM
Gardner	Jean	500128	UNISYS
Gehrmann	Paul	500294	IBM
Gicca	Greg	[illegible]	[illegible]

PARTICIPANTS BY LAST NAME
For Meeting Code: stars91
STARS '91 CONFERENCE ATTENDEE LIST

Last Name	First Name	Reg Id	Company
Gill	Terry	500240	Carnegie Mellon Univ.
Girard	Tom	500129	UNISYS
Glick	Norman	500430	NSA
Gogia	B.L.	500347	SAIC
Goodell	Thomas	500250	Fleet Combat Direction
[illegible]	Bob	500319	[illegible]
Gordon	Del	500130	UNISYS
Graham	Marc	500307	SEI/CMU
Green	Ronald	500006	U.S. Army Strategic Def. Cmnd.
Grieve	David	500160	IBM
Grillo	Louis	500222	GTE
Grimes	Ray	500274	IBM
Grindley	Alexander	500461	SPAWAR
Grobicki	Tom	500181	ADASOFT, Inc.
Gross	Richard	500185	USAF
Guffey	Moira	500077	USAF
Guy	Ed	500131	UNISYS
Haddad	[illegible]	500227	Aerospace Corporation
Hall	John	500008	Dept of Defense - NSA
Hamilton	James	500414	Boeing
Hamilton	Paul	500402	Paramax Electronics Inc.
Hamstreet	Robert	500369	US Air Force
Hanrahan	Bob	500061	Air Force STSC
Harmon	Don	500132	UNISYS
Harmon	Stig	500103	AT&T Bell Labs
Harris	Robert	500091	Wright Laboratory
Hart	Gibbie	500443	Software Engineering Institute
Hart	Hal	500377	TRW
Hartmoller	Roger	500356	Trusted Information Syst. Inc.
Hartwell	Craig	500422	Software Arch. & Engineering
Hathorn	Fred	500501	DOD
Hazle	Marlene	500153	MITRE Corp.
Heartwell	Craig	500451	SAF
Hebenstreit	Earl	500087	Logicon, Inc.
Hefferon	Henry	500376	EDP News Service
Heller	Trevor	500195	SAIC
Henault	Richard	500030	Systems Engineering Research
Henslee	James	500009	USAF/ESC/AVS
Herman	Frank	500005	Vitro Corporation
Hewitt	Michael	500500	Oracle Corp.
Hinchliffe	Anita	500477	[illegible]
Hinton	Dawn Maria	500231	Transarc Corporation
Hodges	William	500410	Boeing
Hodyle	Andrew	500417	US Air Force
Hoffman	Andy	500165	TEXAS INSTRUMENTS
Holibaugh	Bob	500388	Carnegie Mellon University
Hollowell	Glenn	500458	Texas Instruments
Holmes	Robert	500035	Naval Ocean Systems Center
Holtzblatt	Lester	500230	MITRE Corporation
House	Ronald	500224	Naval Underwater Syst. Cntr.
Houser	Anita	500396	IBM
Huey	C.W.	500112	IBM Corporation
[illegible]	[illegible]		

PARTICIPANTS BY LAST NAME
For Meeting Code: stars91
STARS '91 CONFERENCE ATTENDEE LIST

Last Name	First Name	Reg Id	Company
Hunninghake	David	500142	USAF
Hwang	Phillip	500469	NAVSWC
Ierardi	Frank	500324	IBM Corporation
Ivy	Nathan	500164	INTEL CORPORATION
Jacowitz	Larry	500275	IBM
Jenkins-Bonta	Jovita	500427	TRW
Jepson	Robert	500303	VERITAS Software
Johnson	Denise	500478	FED SSAS
Johnson	Denise	500372	US Navy
Johnson	Donna	500064	JCS Systems, Inc.
Jones	Mark	500373	US Navy
Jones	Ronald	500371	Boeing Computer Services
Jordano	Tony	500094	IBM/Federal Sector Division
Juttelstad	Daniel	500379	Naval Underwater Sys. Ctr.
Kantor	Jeff	500157	SEMATECH
Kaposta	David	500197	USAF/ESD/AVE
Kasunic	Mark	500410	Boeing
Kelly	Paul	500344	IBM
Kenner	Jim	500363	Software Engineering Tech.
Ker	Keith	500357	Trusted Information Syst. Inc.
Kessinger	Richard	500326	Lockheed Missiles & Space Co.
Kihara	Mike	500415	Boeing
King	James	500111	Boeing
King	Konrad	500417	Boeing
Klein	Judith	500491	IBM
Kleinwaks	Jeff	500353	CAE LINK Corporation
Klingler	Carol	500123	TRW
Klucas	Casper	500335	HQUSAF/SCXS
Kodeih	Mahmoud	500078	IBM
Koeser	Ron	500325	VERDIX Corporation
Koger, Jr.	Perry	500074	Electronic Warfare Associates
Konieczny	Frank	500342	General Research Corporation
Kopp	Alan	500010	TeleSoft
Kouchakdjian	Ara	500362	Software Engineering Tech.
Kovalcik, Jr.	Edward	500327	IBM
Kraly	Tom	500080	IBM
Kramer	John	500339	DARPA
Krantz	Don	500032	Alliant Techsystems, Inc.
Krasner	Herb	500012	SAIC STC
Krause	Keith	500155	Intermetrics
LaForme	Deborah	500472	USAISSDCW, RAPID Ctr.
Lai	Robert	500302	Software Productivity Consort.
Lambert	Chris	500359	Galaxy Global Corporation
Landry	Pat	500502	Digital Equipment Corp.
Langdon	Kim	500196	CECOM Signals Warfare Div.
Larson	Ronald	500361	Navy Dept.
Law	Margaret	500041	NIST
Lawrence	Tanya	500133	UNISYS
Layman	Gene	500307	Naval Research Lab
Leary	John	500395	Software Engineering Institute
LeFevre	Jeanne	500393	UNISYS Defense Systems
LePinto	Frank	500330	Computer Sciences Corp

PARTICIPANTS BY LAST NAME
For Meeting Code: stars91
STARS '91 CONFERENCE ATTENDEE LIST

Last Name	First Name	Reg Id	Company
Lillie	Charles	500201	[illegible]
Lin	Cathy	500072	ESD/AVS Hanscom Air Force Base
Lin	Chi	500220	Jet Propulsion Laboratory
Lindenberger	Nicholas	500140	USAF
Listvinsky	Gary	500455	Rockwell International
Lock	Evan	500170	COMPUTER COMMAND & CONTROL CO.
Lonardo	Guido	500069	Raytheon
Lopez	Rosa	500441	[illegible] Technology
Lott	Christopher	500102	TRW/Univ. of Maryland
Love	Kim	500154	UNISYS
[illegible]	Joe	500204	[illegible] International
MacDonald	Richard	500161	Univ. of Virginia
Macdonell	[illegible]	500515	RATIONAL
Madere	Sharon	500457	US Naval Oceanographic Office
Makowski	Jerome	500304	Defense Intelligence Agency
Malcolm	Nadine	500301	Ascent Logic Corporation
Marceau	Gilles	500205	Dassault Aviation
Martell	Jeff	500517	RATIONAL
Martin	Rhonda	500016	Purdue University
Mason	Eugenie	500135	UNISYS
Maurer	Donna	500397	Local Command & Control Sys.
May	Tom	500150	Texas Instruments
Maymir-Ducharme	Fred	500256	Grumman - DSD
McCormick	Allan	500073	[illegible] & Associates
McInnes	Colin	500425	Paramax Electronics
McInnis	Ray	500233	UNISYS
McIntyre	David	500219	UNISYS Defense Systems, Inc.
McKay	Charles	500042	Software Engineering Rsch Ctr.
McKinney	Tom	500512	UNISYS
McManus	John	500448	DSD Laboratories
McPherson	Michael	500106	USAF
McWhorter	Richard	500407	McDonnell Douglas Space Systems
Melhado	Warren	500460	UNISYS Defense Systems
Mellor	Frank	500429	US Air Force
Mellott	Robert	500452	Paramax Syst. Corp.
[illegible]	L.A.	500464	UNISYS
Meredith	L.M.	500354	Control Data Corporation
Micol	David	500321	IBM
Milett	Irene	500334	GTE
Miller	Deron	500291	[illegible]
Millett	Irene	500070	GTE
Mitchell	Brenda	500003	Vitro Corporation
Moody	Jim	500456	DISA
Moore	Jim	500276	IBM
Morin	Joseph	500300	Software Engineering Institute
Morris	Ed	500300	Software Engineering Institute
Moseley	Warren	500315	Texas Instruments
Mumm	Hans	500214	Naval Ocean System Center
Munck	Robert	500136	UNISYS
Murphy	Jack	500442	DISA
Myers	Gilbert	500420	Naval Ocean Systems Ctr.
Myles	David	500317	IBM Corporation
Neiberg	[illegible]	500485	The MITRE Corporation

PARTICIPANTS BY LAST NAME
For Meeting Code: stars91
STARS '91 CONFERENCE ATTENDEE LIST

Last Name	First Name	Reg Id	Company
Neary	John	500415	Boeing
Nieng	Kenneth	500055	AT&T Bell Laboratories
Nix	Kevin	500338	Softool Corporation
O'Leary	Kate	500260	Arbor Text
O Neel	Bruce	500465	Hughes / STX
Obranovich	Ron	500460	DOD/USAF
Obranovich	Ron	500332	USAF
Olivo	Martha	500229	Univ. of Maryland
Orgren	Carl	500137	UNISYS
Orville	Douglas	500208	USAF
Osborne	Joan	500151	IBM
Over	Jim	500437	Software Engineering Institute
Parker Gates	Linda	500497	SEI
Parks	Donald	500001	ASSET
Parsons	Virginia	500496	US Army
Parter	Marshall	500259	NISMC
Payton	Teri	500139	UNISYS
Pease	Dale	500019	GTE
Perolta	Carl	500280	MITRE Corporation
Peterson	Jodi	500017	TRW
Pethe	Vishwas	500068	UNISYS-UGS
Petrucci	Dennis	500138	UNISYS
Phillips	Pat	500175	PEO STAMIS, US ARMY
Piper	Joanne	500246	Center for Information Mgmt.
Pixton	Jerry	500140	UNISYS
Plebani	Eileen	500161	AT&T
Polzer	Hans	500141	UNISYS
Poore	T.H.	500228	
Pope	George	500261	Arbor Text
Poppel	Paul	500360	IBM
Porter	Roz	500310	CINCOM
Potter	Marshall	500495	Dept of the Navy
Poulin	Jeffrey	500115	IBM Reuse Tech. Support Center
Powers	Matt	500399	IBM
Prasel	Sandra	500217	Oki Corporation
Price	Andrew	500327	VERDIX Corporation
Prieto Diaz	Ruben	500438	Reuse, Inc.
Pyster	Arthur	500011	Software Product. Consortium
Quiel	Mark	500253	DN American
Racine	Glenn	500346	AIRMICS
Raveling	Jerry	500204	JR Software
Ready	Lynn	500325	Lockheed Missiles & Sp. Co.
Reddy	Jay	500065	Strictly Business
Redwine, Jr.	Samuel	500218	Software Productivity Consor.
Reed	Lowell	500100	GE Aircraft Engines
Reid	Thomas	500043	VA Tech N. VA Graduate Center
Reilly	Lauren	500172	UNISYS
Renfro	John	500306	IBM
Rhue	Lee	500127	UNISYS Defense Systems
Riddle	Bill	500007	Software Design & Analysis
Riggs	Marcus	500474	US Army
Robinette	Jim	500403	DISA/JIEO/TVCF
Robinson	Jeff	500169	MOTOROLA

PARTICIPANTS BY Last NAME
For Meeting Code: stars91
STARS '91 CONFERENCE ATTENDEE LIST

Last Name	First Name	Reg Id	Company
Ruby	Clyde	500234	Inst. For Defense Analyses
Rosenberg	Julia	500119	UNISYS Defense Systems
Rotter	Sharon	500036	Naval Ocean Systems Center
Rouff	Chris	500504	NASA GSFC
Ruegsegger	Theodore	500113	GE Control Federal Systems
Rush	Darlene	500435	U.S. General Acctg. Office
Russell	Keith	500210	Naval Trng. Syst. Ctr./Nav Air
Russell	Kevin	500143	UNISYS
Sabbagh	Ronald	500307	UNISYS
Saisi	Robert	500034	DSD Laboratories
Salerno	Alex	500232	Inst. for Defense Analyses
Sarah	Agnes	500070	Raytheon
Saunders	Larry	500503	Digital Equipment Corp.
Savage	James	500514	Software A & E
Schmidt	Richard	500516	RATIONAL
Schmiedekamp	Carl	500075	Naval Air Development Ctr.
Scott	Randall	500023	Software Productivity Consortium
Seaver	David	500395	P.E.I.
Seidewitz	Ed	500494	NASA Goddard Space Center
Shaffer	Clyde	500101	ESC
Shields	Thomas	500144	UNISYS
Shwetz	Andrew	500168	DIGITAL EQUIPMENT CORP
Sielaff	Robert	500354	UNISYS Defense Systems
Sigler	Steve	500025	USAF
Sili	Jeffery	500037	Simms Industries, Inc.
Simonson	Marty	500406	HQ AFSC/SCTD
Sims	Calvin	500349	US Army
Smith	Dennis	500299	Software Engineering Institute
Smith	Dennis	500026	USAF
Smith	Dudley	500307	Smiths Industries
Smith	Kathryn	500292	NASA Langley Research Center
Smith	Larry	500060	Air Force STSC
Soule	Albert	500306	Software Engineering Inst.
St. Dennis	Carol	500146	USAF
Stanfill	R. James	500071	Digital Equipment Corporation
Steffey	Ray	500084	Logicon, Inc.
Stellar	Frank	500400	Technology Mgmt. School
Stephenson	David	500447	Computer Sciences Corporation
Stuart	Antoinette	500115	US Navy
Stump III	Charles	500226	VITRO Corporation
Sweet	William	500390	Software Eng. Institute
Szmanski	Raymond	500109	USAF
Tarver	Sandra	500159	USAF
Tarver	Tim	500131	USAF
Taylor	Tim	500345	IBM
Terrell	Jim	500321	SAIC
Thelen	Karen	500[illegible]	Honeywell
Tobin	Vickie	500010	CTA Inc.
Trimble	Joel	500013	Software Engineering Inst.
Tupper	Kevin	500300	UNISYS
Turner	Gary	500[illegible]	IBM
Vajda	Dan	500360	TRW
Valdez	Paul	500454	USAF ESD/AVS

PARTICIPANTS BY LAST NAME
For Meeting Code: stars91
STARS '91 CONFERENCE ATTENDEE LIST

Last Name	First Name	Reg Id	Company
Van Groesbeek	James	500067	Northrop Aircraft Division
Van Valkenburg	Randy	500251	NASA Langley Research Center
VanBuren	James	500177	C.S. DRAPER LAB
Varno	Raul	500298	IBM
Vasbury	Newman	500255	UNISYS
Vassallo	Geoffrey	500311	UNISYS
Vertillo	Tom	500173	UNISYS
Vitaletti	William	500110	SofTech, Inc.
Vladavsky	Luba	500086	Logicon, Inc.
Vucurevich	Ronald	500083	IBM Corporation
Wald	Elizabeth	500376	Office of Naval Technology
Walker	Carolyn	500262	IBM
Waller	Phil	500293	SAIC
Walters	Neal	500152	IBM
Ward	Elaine	500235	The MITRE Corporation
Ward	Mary Catherine	500278	IBM
Warshaw	Susan	500027	Defense Info. Systems Agency
Watts	Jerry	500409	US Air Force WR ALC
Weber	Doug	500216	BDM Corporation
Weeks	Joe	500424	Cadre Technologies
Weiberg	Maurine	500426	The MITRE Corporation
Weiler	Paul	500263	IBM
Weinberger	Alan	500145	UNISYS
Weisman	David	500394	UNISYS Defense Systems
Weszka	Joan	500420	IBM FSD
White	Donald	500207	IBM
White	Douglas	500312	USAF
White	Melissa	500405	Azimuth
Whitmore	Ken	500092	Dual & Associates
Wilkinson	Charles	500288	Textron Defense Systems
Williams	Charles	500445	Systematica
Williams	Roger	500303	Software Productivity Consort.
Williams	Steve	500290	IBM
Williamson	James	500296	U.S. Air Force
Willis	John	500375	U.S. Army, CECOM
Wilson	Eddie	500128	2500 CG
Wilson	Kathryn	500344	Inst. for Defense Analyses
Wilson	Tom	500029	IBM FSD
Winfield	David	500030	IBM
Winters	Jack	500045	IBM
Wong	William	500070	NIST
Wright	Charles	500174	UNISYS
Wright	Elena	500405	USAF/ESDC W
Yaeger	Anthony	500117	PRC Inc.
Yanney	Ralf	500289	TRW SEDD Division
Yaramanoglu	Melih	500085	Logicon, Inc.
Yu	Donald	500444	UNISYS
Zayas	Ed	500237	Transarc Corporation
Ziebe	Ted	500156	SEMATECH
Zimmerman	John	500398	IBM
Zimmerman	Raymond	500302	ITT Gilfillan

Total Participants in this Report: 475

STARS NEWSLETTER
MAILING LIST

The STARS Newsletter continually updates its mailing list with new additions and changes of place of employment or residence. To be added to the mailing list or to have your mailing address changed, please fill out this form and mail it to:

STARS Technology Center
Suite 400
401 North Randolph Street
Arlington, Va. 22203

□ New Addition to List □ Change of Address

□ Mr. □ Mrs. □ Ms. □ Other

Name______________________________________

Organization/Company_____________________________

Address____________________________________

City ________________ State ________ Zip________

Telephone ____-____-______

E-Mail Address ____________________________

DEMONSTRATIONS

STARS Technology Center (STC) Demonstrations

Beginning early in 1992, demonstrations at the STC will be held on the fourth week of every month.

For information on what is to be demonstrated, as well as on which day of the week it will be demonstrated, call the STC Demonstrations Coordinator at (703) 243-8655. Initially, demonstrations will be provided in response to requests.

Refer to the attached map for the location of the STC.

HOW TO OBTAIN ITEMS FROM THE DEFENSE TECHNICAL INFORMATION CENTER (DTIC) OR FROM THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS)

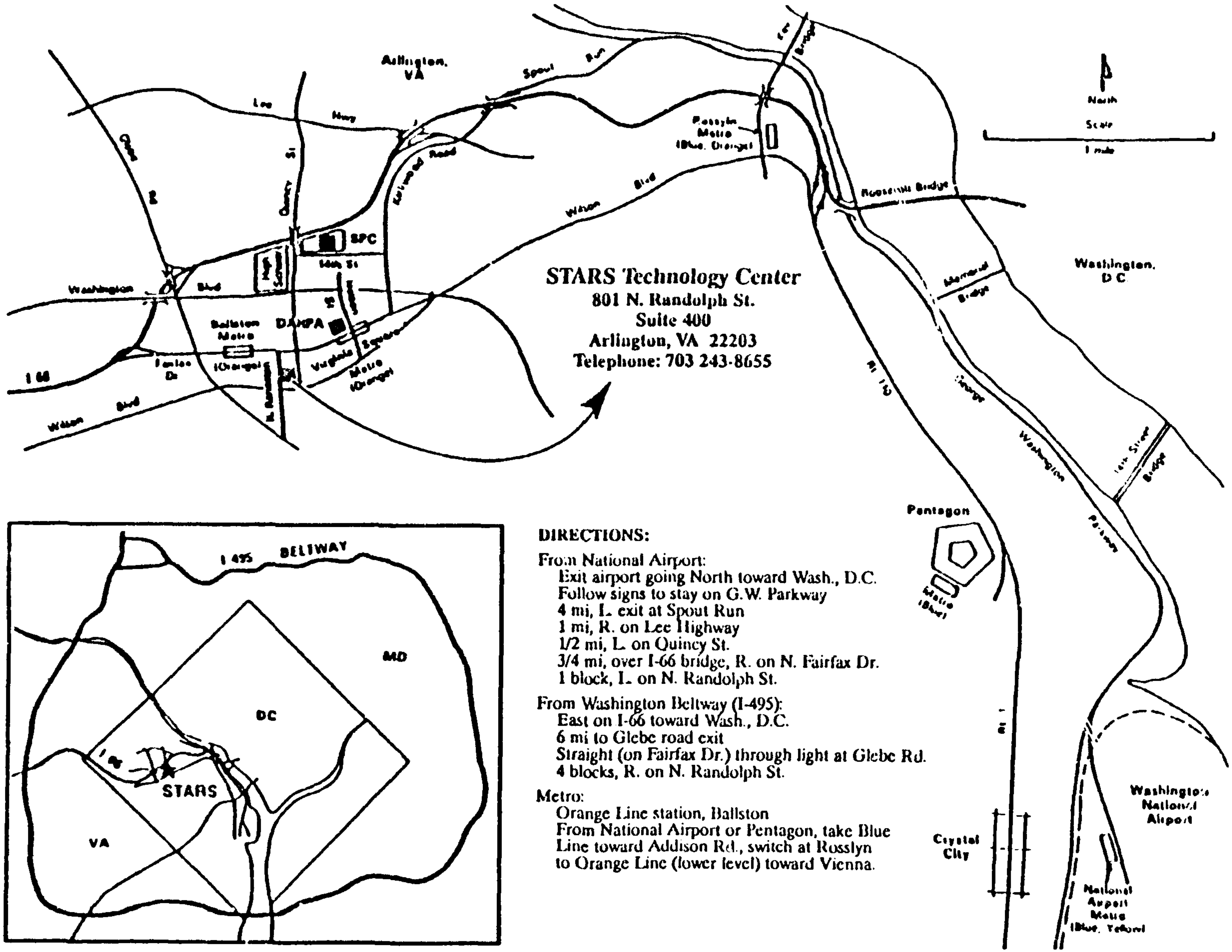
DTIC contains items that are not releasable to the general public. To obtain items from DTIC, you must be government or a contractor and be a registered DTIC user. Please refer to the information provided below or contact DTIC for details.

Defense Technical Information Center
Cameron Station
Alexandria, Va. 22304-6145
Tel: (703) 274-7633

The general public can obtain items approved for general release from NTIS. Items can be ordered through the mail or over the phone. A variety of payment options is available.

NTIS
5285 Port Royal Road
Springfield, Va. 22161-0001
Tel: (800) 553-NTIS (553-6847)
(703) 487-4650

Directions to STARS Technology Center



STARS Technology Center
801 N. Randolph St.
Suite 400
Arlington, VA 22203
Telephone: 703 243-8655

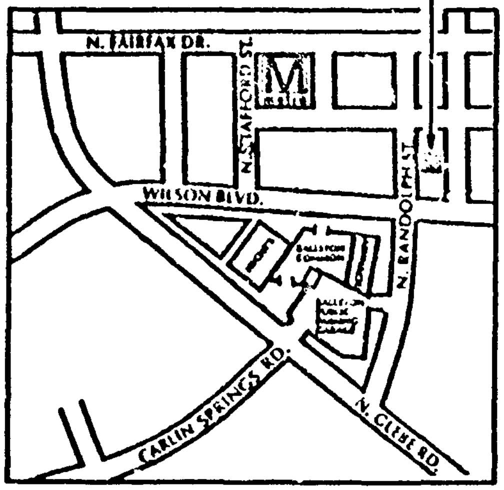
DIRECTIONS:

From National Airport:
- Exit airport going North toward Wash., D.C.
- Follow signs to stay on G.W. Parkway
- 4 mi, L. exit at Spout Run
- 1 mi, R. on Lee Highway
- 1/2 mi, L. on Quincy St.
- 3/4 mi, over I-66 bridge, R. on N. Fairfax Dr.
- 1 block, L. on N. Randolph St.

From Washington Beltway (I-495):
- East on I-66 toward Wash., D.C.
- 6 mi to Glebe road exit
- Straight (on Fairfax Dr.) through light at Glebe Rd.
- 4 blocks, R. on N. Randolph St.

Metro:
- Orange Line station, Ballston
- From National Airport or Pentagon, take Blue Line toward Addison Rd., switch at Rosslyn to Orange Line (lower level) toward Vienna.

Directions to STARS Technology Center



STARS Technology Center
801 N. Randolph St.
Suite 400
Arlington, VA 22203
Telephone: 703 243-8655

ACRONYMS USED IN THESE PROCEEDINGS

AAA	Artifacts, Agents, and Activities
ACAS	Application Control Architecture Services
ADA	Programming language
AF	Air Force
AFS	A distributed product of Tramsarc
AIB	Action Item Browser
AIL	ASSET Interchange Language
AIX	IBM Advanced Interactive Executive operating system
AIX	Advanced Interactive operating system
AJPO	Ada Joint Program Office
ALOAF	ASSET Library Open Architecture Framework
AMS	ASSET Management System
ANS	Artificial Neural Systems; American National Standard
ANSI	American National Standards Institute
APP	Application Portability Profile
APPL/A	Ada Process Programming Language based on Aspen
ARCS	Automated Reusable Components System
ASIS	Ada Semantic Interface Specification
ASSET	ASSET Source for Software Engineering Technology
ASW	Anti-Submarine Warfare
ATIS	A Tool Integration Standard (DEC); Atherton Tool Integration Service
BMS	Broadcast Message Service
C2	Command and Control
C3	Command, Control and Communication
CAIS-A	Common APSE Interface Set - Revision A
CALS	Computer-Aided Acquisition and Logistics Support
CAMP	Common Ada Missile Packages
CAPE	Computer Aided Project Engineering
CARDS	Central Archive for Reusable Defense Software
CASE	Computer Aided Software Engineering
CC	Configuration Control
CCC	Change and Configuration Control (Product name)
CCPDSR	Command Center Processing and Display System Replacement
CCITT	Consultative Committee for ITT Corp.
CDD	Common Data Dictionary
CDIF	CASE Data Interchange Format
CDRL	Contract Data Requirements List
CECOM	Communications and Electronics Command (U.S. Army)
CEPA	Cleanroom Engineering Process Assistant
CGM	Computer Graphics Metafile
CIM	Corporate Information Management
CIS	CASE Integration Services committee
CLIPS	A NASA expert system shell - C language
CM	Configuration Management
CMM	Capability Maturity Model
CODASYL	Conference On Data Systems Languages
COHESION	A digital software environment product

CONOPS	CONcept of Operations
COTS	Commercial, Off-The-Shelf
CRDA	Cooperative Research and Development Agreements
DARPA	Defense Advanced Research Projects Agency
DC	Data Collection
DCE	Distributed Computing Environment
DDR&E	Design Development Research & Engineering
DEC	Digital Equipment Corporation
DISA/CIM	Defense Information Systems Agency/Center for Information Management
DoD	Department of Defense
DSSA	Domain Specific Software Architecture
DTIC	Defense Technical Information Center
E-SLCSE	SLCSE commercialization effort
EAST	European Advanced Software Technology
ECMA	European Computer Manufacturers Associations
ECP	Engineering Change Proposal
EE	Electrical Engineering
EIA	Electrical Industries Association
EIS	Engineering Information System
ER	Entity Relationship
ERA	Entity-Relation-Attribute
FAR	Federal Acquisition Regulation
FCDSSA	Fleet Combat Directorate Systems Support Activity
FFRDC	Federally Funded Research and Development Center
FIM	Framework Information Model
FJAG	Framework Joint Activity Group
FRAC	Framework Requirements and Criteria
FS	Fraction of Savings
FT	Fraction of Time
FTAM	OSI File Transfer Access and Management
FTP	TCP/IP File Transfer Protocol
GCC	Generic Command Center
GIE	Groupement d'Interets Economiques
GKS	Graphical Kernel System
GNMP	Government Network Management Program
GOSIP	Government Open System Interconnection Protocol
HP	Hewlett Packard
HPCC	High Performance Computers and Communication
I&T	Integration & Test
I/O	Input/Output
ID	IDentification
IDA	Institute for Defense Analysis
IDE	Interactive Development Environments
IEEE	Institute of Electrical and Electronics Engineers
IGES	Initial Graphics Exchange Specification
IM	Information Management
IM	Information Model
IOC	Initial Operational Capability
IMCON	Information Modeling CONvergence

IR&D	Independent Research & Development
IRAC	International Requirements And Criteria
IRDS	Information Resource Dictionary System
ISAM	Indexed Sequential Access Method
ISEE	Integrated Software Engineering Environment
ISO/IEC JTC1	International Standards Organization/International Electrotechnical Commission Joint Technical Committee
ISP . 6	International Software Process Working group No. 6
ISSI	International Software Systems, Inc.
IW CASE	International Working group on CASE
JLC	Joint Logistics Commanders
KDSI	Thousands of Delivered Source Instructions
KI	Knowledge Information
LAN	Local Area Network
LOC	Lines Of Code
MCCR	Mission-Critical Computer Resources
ME	Mechanical Engineering
MFPL	Message Format Processing Language
MIL-STD-2167A	Military standard for defense systems software development
MIS	Management Information Systems
MIT	Massachusetts Institute of Technology
MLS	MultiLevel Secure
MM	Man Months
MTV	Message Translation and Validation
MVP	MultI-View Process modeling project
MVP-L	MVP Language
NASA	National Aeronautics and Space Administration
NATO	North Atlantic Treaty Organization
NAVAIR	NAVal AIR Systems
NBCD	Network-Based Collaborative Development
NCCPM	Navy Command and Control Process Model
NCS/RPC	Network Computing Services/Remote Procedure Call
NFS	Network File System
NGCR	Next Generation Computer Resources (navy program)
NIST	National Institute of Standards and Technology
NOSC	Naval Ocean Systems Center
NRAC	NATO Requirements and Criteria
NTIS	National Technical Information Service
NTSC	Navy Training Systems Center
OAF	Open Architecture Framework
OCD	Operational Concept Document
ODA/ODIF	Office Document Architecture/Interchange Format
OMG	Object Management Group
OMS	Object Management System
OS	Operating System
OSF	Open Software Foundation
OSI	Open Systems Interconnection (ISO standard)
OTS	Off-The-Shelf

P1175	A standard reference model for computing system tool interconnection
PAL	Process Asset Library
PCIS	Portable Common Interface Set
PCTE	Portable Common Tools Environment
PDAG	Process Definition Advisory Group
PDES	Product Data Exchange Specification
PDSS	Post-Deployment Software Support
PHIGS	Programmer's Hierarchical Interactive Graphics System
PM	Process Management
PM	Process Model
PMDB	Project Master DataBase (TRW IR&D project name)
POCD	Process Operational Concept Document
POSIX	Portable Operating System Interface
PPL	Process Programming Language
PREIS	PRototype Engineering Information System
PREIS	Policy Representation using PREIS
PRISM	Portable Reusable Integrated Software Modules
PSE	Project Support Environment
PSESWG	Project Support Environment Standards Working Group
PSL/PPL	Process Support Language/Process Programming Language
Q/I	Question/Issue
QA	Quality Assurance
RAASP	Reusable Ada Avionics Software Package
RAPID	Reusable Ada Products for Information systems Development
RDA	Remote Database Access
RDBMS	Relational DataBase Management System
READS	Requirements Entry Allocation Decomposition System
RIG	Reuse library Interoperability Group
RJAG	Reuse Joint Activities Group
RLF	Reusability Library Framework
RMP	Risk Management Plan
RNTDS	Restructured Navy Tactical Data System
ROAMS	Reusable Object Access Management System
ROCD	Reuse Operational Concept Document
ROI	Return On Investment
RPC	Remote Procedure Call
S/W	SoftWare
SADT	Structured Analysis and Design Techniques
SAIC	Science Applications International Corporation
SBIR	Small Business Innovative Research
SCPM	STARS Composite Process Model
SDE	Software Development Environment
SDIO	Strategic Defense Initiative Office
SDP	Software Development Plan
SE	Software Engineering
SEE	Software Engineering Environment
SEI	Software Engineering Institute
SEMATECH	A consortium
SEPG	Software Engineering Process Group
SETA 2	Second international Symposium on Environments and tools for Ada
SFGL	Name of French software company

SGML	Standard Generalized Markup Language
SIM	SEE Information Model
SJAG	SEE Joint Activities Group
SLCSE	Software Life Cycle Support Environment
SMTP	TCP/IP Simple Mail Transfer Protocol
SOCD	SEE Operational Concepts Document
SOP	Standard Operating Procedures
SPC	Software Productivity Consortium
SPDL	Standardized Page Descriptor Language
SPMS	Software Process Management System
SPO	Systems Program Office
SPS	Software Productivity Solutions
SQL	Standard Query Language; SQL database language
SRL	Software Reuse Library
SSP	STARS Standards Portfolio
STARS	Software Technology for Adaptable. Reliable Systems
STEP	Spec To Executable Program; Standard for Exchange of Product model data
SW	Software
SWAP	SoftWare Action Plan
SWTP	SoftWare Technology Plan

TCP/IP	Transmission Control Protocol/Internet Protocol
TELNET	TCP/IP interactive session protocol
TFA(NFS)	Transparent File Access
TT	Technology Transfer

UI	User Interface
UIMS	UI Management System

V/M/S	Vision/Mission/Strategy
VAX	DEC product
VMS	VAX Virtual Moritor System
VTP	OSI Virtual Terminal Protocol

WAN	Wide Area Network

X	X Window System
X3H3	ANSI technical committee on graphics
X3H4	ANSI Technical committee on IRDS
X3H6	ANSI Technical committee on CASE tool integration models
X.400	OSI message handling system
X.500	OSI network directory services
XTI	X/open Transport Interface
XVT	X Virtual Terminal; eXtensible Virtual Toolkit

AUTHOR INDEX